programmez!
#202 - décembre 2016
le magazine des développeurs


# Apprendre le code aux enfants

## NOUVEAU !

Chaque mois, tout savoir sur la **Réalité virtuelle**

## Je maîtrise GitHub

## Choisir son fournisseur de Cloud Computing

## Faire du Responsive Design avec Drupal 7

M 04319 - 202 - F: 6,50 € - RD

Printed in EU - Imprimé en UE - BELGIQUE 7 € - Canada 9,80 $ CAN - SUISSE 13,10 FS - DOM Surf 7,50 € - TOM 1020 XPF - MAROC 55 DH

OPÉRATION

# POUR 1 EURO DE PLUS

JUSQU'AU 12 DÉCEMBRE

Aucun abonnement à souscrire

## COMMANDEZ WINDEV 22

**OU** WEBDEV 22 **OU** WINDEV MOBILE 22

## ET RECEVEZ LE NOUVEL iPhone 7

POUR 1 EURO DE PLUS

Actuellement sur les routes:
WinDevTour 22. 11 villes. Gratuit.
Inscrivez-vous sur pcsoft.fr

iPhone 7 Plus

iPhone 7 Plus **128GB**.

Ou choisissez :

- iPhone 7 **256GB**
- iPad Pro **32GB** 12,9"
- MacBook Air **128GB** 11,6"
- ...

(voir le détail sur www.pcsoft.fr)

# WINDEV®

Atelier de Génie
Logiciel Professionnel
cross-plateformes


Pour bénéficier de cette offre exceptionnelle, il suffit de commander WINDEV 22 (ou WINDEV Mobile 22, ou WEBDEV 22) chez PC SOFT au tarif catalogue avant le 12 Décembre 2016. Pour 1 Euro de plus, vous recevrez alors le ou les magnifiques matériels que vous aurez choisis. Offre réservée aux sociétés, administrations, mairies, GIE et professions libérales, en France métropolitaine. **L'offre s'applique sur le tarif catalogue uniquement**. Voir tous les détails et des vidéos sur : **www.pcsoft.fr** ou appelez-nous (04.67.032.032).

Le Logiciel et le matériel peuvent être acquis séparément. Tarif du Logiciel au prix catalogue de 1.650 Euros HT (1.980,00 TTC). Merci de vous connecter au site www.pcsoft.fr pour consulter la liste des prix des matériels et les dates de disponibilité. Tarifs modifiables sans préavis.

**WWW.PCSOFT.FR**

# Une dose de réalité (virtuelle) ?

Depuis les débuts de Programmez!, nous vous parlons régulièrement de nouvelles technologies, plus ou moins nouvelles et cet esprit ne change pas. Sauf si votre connexion Internet était en panne depuis un an, vous savez sans doute que les casques de réalité virtuelle commencent à se multiplier, et qu'une petite guerre est en train d'être livrée par HTC, Facebook, Sony, et un peu plus loin Google et Microsoft. Pour ne citer que les principaux acteurs. Nous avons décidé d'ouvrir une rubrique dédiée à la réalité virtuelle / augmentée / mixte.

Il nous semble important d'en parler régulièrement car pour le développeur, de nombreuses technologies et opportunités s'ouvrent à lui. Les compétences VR / VA ne sont pas à négliger surtout si vous souhaitez travailler dans le monde des jeux. La dernière Paris Games Week a été l'occasion de voir de nombreux jeux en démonstration sur les stands. Même si les compétences ne sont pas encore très recherchées ailleurs que chez certains éditeurs de jeux et studios, que cela ne vous empêche pas de regarder ce qu'il s'y passe, de tester les matériels et les modèles de développement.

Mais avant d'aller vous frotter à ces casques, nous avons beaucoup de choses dans notre référentiel : GitHub, choisir son fournisseur de Cloud, le langage Rust, Spark, Drupal, Microsoft Azure, notamment avec la partie serverless d'Azure et un peu de Java.

Il y a 12-18 mois, les medias et le gouvernement parlaient beaucoup d'apprentissage du code, de la programmation à l'école. Le sujet s'est tassé mais ce n'est pas une raison pour l'oublier. Au-delà de l'école, aujourd'hui, de nombreux outils et matériels permettent d'apprendre, de faire découvrir facilement la programmation aux enfants et aux ados. Notre dossier vous fera découvrir de nombreuses expériences sur le sujet. Nous reviendrons sur une journée dédiée dans le Sud de la France. Même si le côté ludique est important, il ne faut pas oublier que la programmation peut-être frustrante. Un des défis est ne pas rebuter les jeunes et de monter en difficulté progressivement. Car parmi ces jeunes, il y a les développeurs de demain.

*Toute l'équipe de Programmez! vous souhaite de bonnes fêtes de fin d'année (eh oui, déjà).*

ftonic@programmez.com
Dresseur de codes.



**Dans le prochain numéro !**
**Programmez! #203, dès le 2 janvier 2017**

**Spécial Web !**
Progressive Web Apps : le Web mobile enfin mobile !
WebGL : un moteur 3D natif dans les navigateurs

**Les technologies et les profils techniques à suivre en 2017**

**Dossier Bot**
Pourquoi et comment utiliser un Bot ? Créer et déployer en quelques minutes vos bots.

décembre

## INTEGRATION SUMMIT @MICROSOFT

### 5 décembre, Paris

La digitalisation croissante des entreprises, faisant la part belle à l'adoption du Cloud, des applications mobiles et de l'Internet des objets, implique un bouleversement dans la façon de penser l'intégration des assets. Dans le même temps, l'avènement de ces mêmes technologies offre l'opportunité à ces entreprises de collecter et traiter l'information plus vite et à moindre coût pour en faire un avantage concurrentiel.

Le monde de l'intégration d'application, pleinement impacté, est ainsi en pleine mutation. Les solutions Microsoft BizTalk Server, Azure Logic App, API Management et Flow offrent toutes les fonctionnalités qui permettent de décliner cette transformation digitale sur l'aspect ô combien critique qu'est l'intégration.

Informations : http://blog.cellenza.com/cloud-2/azure/savethedate-0811-integration-summit-microsoft/

## NUI DAY 2016

### 13 décembre, Paris

Suite au succès de l'édition 2015, la conférence NUI Day revient le 13 décembre prochain avec un programme encore une fois dédié aux expériences émergentes. L'équipe NUI Day (Natural User Interface Day) vous propose une journée dédiée à l'innovation et aux nouvelles interactions. Articulée autour de deux expériences : d'un côté 8 conférences animées par des professionnels de différents horizons et de l'autre une zone expérientielle afin de permettre à tous de rencontrer ceux qui font l'innovation et de tester les dernières nouveautés en termes de nouvelles technologies. Vous découvrirez comment les entreprises, les geeks, les artistes, les entrepreneurs et les makers s'approprient ces nouveaux outils d'interaction afin de créer les expériences de demain.

Durant cette journée dans le centre de conférence Microsoft (Issy-Les-Moulineaux), les speakers aborderont les sujets suivants :

- Panorama et usages des technologies de la réalité virtuelle et augmentée ;
- Technologies cognitives au service des expériences utilisateur ;
- NUI en entreprise : enjeux et potentiels business ;
- Les pratiques de créativité en entreprise, comment et pourquoi ?
- Vision croisée d'un projet au travers des métiers NUI ;
- Vision prospective de la robotique : usages et technologies ;
- IoT : Quels outils pour les makers et les artistes ?

Suite à l'annonce de la disponibilité du casque de réalité mixte Microsoft HoloLens en France, l'équipe NUI Day vous propose une session dédiée. N'oubliez pas de vous inscrire gratuitement sur http://bit.ly/nuiday2016

Pour plus d'information, consultez le site http://www.nuiday.com/

### Quelques rendez-vous des communautés

***Paris Jug :***
13 décembre : CQRS / Event Sourcing / DDD
10 janvier 2017 : soirée Young Blood VI

***Paris MongoDB user Group :***
13 décembre : MongoDB 3.4 est là

***Hybride Mobile Paris :***
6 décembre : spécial DotJS

***Chartres.JS***
13 décembre : le premier meetup de la communauté JS de Chartres

***CocoaHeads Paris :***
8 décembre : pas de sujet défini pour le moment. Attention ce meetup se tiendra si une salle est trouvée

***Meet your dev Lyon :***
15 décembre : meetup speed recrutement développeurs – intégrateurs web

***Meetup .Net Toulouse :***
19 décembre : coding for fun
16 janvier 2017 : TypeScript 2, le futur de JS et les transpileurs

***MUG Lyon :***
8 décembre : formation Azure PaaS

OPÉRATION

# POUR 1 EURO DE PLUS

JUSQU'AU 12 DÉCEMBRE

# COMMANDEZ WINDEV 22

**OU** WEBDEV 22 **OU** WINDEV MOBILE 22

# ET RECEVEZ UN SUPERBE MATÉRIEL

POUR 1 EURO DE PLUS

Actuellement sur les routes:
WinDevTour 22. 11 villes. Gratuit.
Inscrivez-vous sur pcsoft.fr


Smart TV **Samsung** Full HD **152cm**
ou Smart TV Samsung **4K** 138 cm

Ou choisissez :
- Samsung Galaxy S7 Edge 32+128 Go
- 2x Samsung Galaxy Tab S2 9,7"

Voir le détail sur www.pcsoft.fr

ASUS
IN SEARCH OF INCREDIBLE

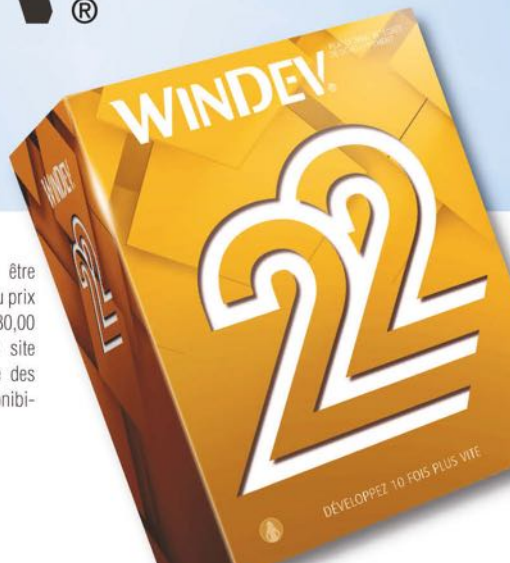
Ultraportable **Asus Zenbook** Aluminium
Windows 10, 13,3", Core i5, 512Go SSD-M2

Ou choisissez :
- Asus Portable 17,3" Core i7 1To
- Asus Mini Tour Core i7 1 To + Ecran 24"

Voir le détail sur www.pcsoft.fr

# WINDEV®

Atelier de Génie
Logiciel Professionnel
cross-plateformes




Pour bénéficier de cette offre exceptionnelle, il suffit de commander WINDEV 22 (ou WINDEV Mobile 22, ou WEBDEV 22) chez PC SOFT au tarif catalogue avant le 12 Décembre 2016. Pour 1 Euro de plus, vous recevrez alors le ou les magnifiques matériels que vous aurez choisis. Offre réservée aux sociétés, administrations, mairies, GIE et professions libérales, en France métropolitaine. **L'offre s'applique sur le tarif catalogue uniquement**. Voir tous les détails et des vidéos sur : **www.pcsoft.fr** ou appelez-nous (04.67.032.032).

Le Logiciel et le matériel peuvent être acquis séparément. Tarif du Logiciel au prix catalogue de 1.650 Euros HT (1.980,00 TTC). Merci de vous connecter au site www.pcsoft.fr pour consulter la liste des prix des matériels et les dates de disponibilité. Tarifs modifiables sans préavis.

**WWW.PCSOFT.FR**

Logiciel professionnel. Version 22 annoncée.


**Elon Musk** continue de se diversifier. Fin octobre dernier, il a annoncé des panneaux solaires qui ne ressemblent pas à des panneaux solaires, mais à des tuiles, des ardoises, etc. Ces panneaux seront le compagnon idéal du Powerwall 2.0, batterie autonome dédiée à la maison.

• **Visual Studio for Mac** a été annoncé le mi-novembre. Il s'agit de Xamarin Studio avec l'interface largement inspirée de Visual Studio et de nouvelles fonctions héritées de la version Windows.

## SÉRIE

La saison 2 de "the man in the high castle" arrive enfin ! Produite par Amazon, la s2 sera là le 16 décembre. L'attente a été longue et la production de cette suite n'a pas été simple. La s1 nous avait énormement plu, une des meilleures séries de 2015 ! La série est basée sur le livre de Philip K. Dick.


• **Les constructeurs français** de systèmes audio sont à la pointe. Devialet a levé 100 millions $.

• **Azure Bot Service** n'aime pas Safari. Nous avons eu ce bug durant nos premiers tests. L'équipe de dév en charge du projet a rapidement précisé à la rédaction que le problème sera réglé rapidement (le cookie du service dépasse les capacités autorisées par Safari). Par contre, aucun problème avec Chrome…

• **OSCC.** Tu veux développer un système autonome pour les voitures ? Le projet Open Source Car Control peut t'aider ! Pour seulement 649 $. Choix de la voiture très limité. www.oscc.io

• **Bug !** Facebook a connu une mésaventure étonnante : 2 millions de morts parmi les inscrits… Tout est redevenu normal.

• Silicon Valley attend la politique du nouveau président des Etats-Unis, **Donald Trump.**

• **Final Fantasy 15 :** seulement 4 ans de développement.

• **Super Mario Run** disponible sur mobile mi-décembre. Attention : 9,99 €

## MOZILLA VEUT SE RELANCER SUR FIREFOX

Chrome a été un rouleau compresseur, Firefox en fut la victime la plus spectaculaire. WebKit, malgré les aléas, reste le moteur de rendu de référence. Mozilla se prépare à changer de moteur. Gecko a été un bon moteur mais n'a pas pu évoluer aussi rapidement que d'autres. Aujourd'hui, le projet Quantum apparaît comme une chance pour relancer Firefox auprès des développeurs et utilisateurs. Cependant, ce moteur n'arrivera que dans un an. Quantum intègrera des modules du projet Servo, un moteur de parallélisme dans le navigateur, soutenu par Mozilla. Les attentes sont grandes et Mozilla joue là une des dernières cartes pour redonner de la vigueur à son navigateur.
Wiki du projet : https://wiki.mozilla.org/Quantum

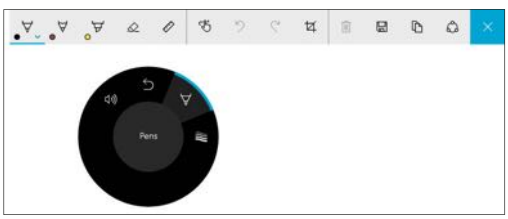
• **Google** a mis fin, officiellement, aux Eclipse Android Developer Tools. Désormais, l'unique IDE officiel est Android Studio.

• **La conférence PWNFest** est sans pitié pour les logiciels et les navigateurs. Cette année, Safari est tombé en 20s, le Google Pixel a résisté 60 secondes. MS Edge est lui aussi tombé.

• **Samsung** rachète le constructeur Harman pour 8 milliards $.

• Les nouveaux **MacBook Pro 2016** embarquent un petit morceau de processeur ARM pour gérer les barres OLED et la sécurité TouchID.

• **Surface Dial : un peu de doc technique pour en savoir plus**




Sur MSDN, on trouve des informations techniques intéressantes sur le Surface Dial, l'objet interactif présenté dans le Surface Studio. Sur les interactions, avec exemples d'implémentations : https://msdn.microsoft.com/en-us/windows/uwp/input-and-devices/windows-wheel-interactions

## INDEX TIOBE DU MOIS

| 11/16 | 11/15 | Tendance | Langage | % | Evolution |
|---|---|---|---|---|---|
| 1 | 1 | | Java | 18.755% | -1.65% |
| 2 | 2 | | C | 9.203% | -7.94% |
| 3 | 3 | | C++ | 5.415% | -0.78% |
| 4 | 4 | | C# | 3.659% | -0.66% |
| 5 | 5 | | Python | 3.567% | -0.20% |
| 6 | 8 | ^ | VB .NET | 3.167% | +0.94% |
| 7 | 6 | v | PHP | 3.125% | -0.12% |
| 8 | 7 | v | JavaScript | 2.705% | +0.23% |
| 9 | 11 | ^ | Assembleur | 2.441% | +0.56% |
| 10 | 10 | | Perl | 2.361% | +0.33% |

Encore une fois peu de changements. PHP chute un peu. Objective-C reprend des couleurs et passe 11e, suivi de très près par Swift, 12e.

ikoula
HÉBERGEUR CLOUD

VOTRE SERVEUR DÉDIÉ XEON À LA CARTE

# OFFRE DÉCOUVERTE -50 %*

DOMAINE ▼ | HÉBERGEMENT ▼ | SERVEUR DÉDIÉ | SERVEUR VIRTUEL VPS ▼ | CLOUD ▼

CODE PROMO
**SERV50**

**14,99 €*** HT/MOIS
~~29,99 €~~

SETUP **10 €**
~~60 €~~

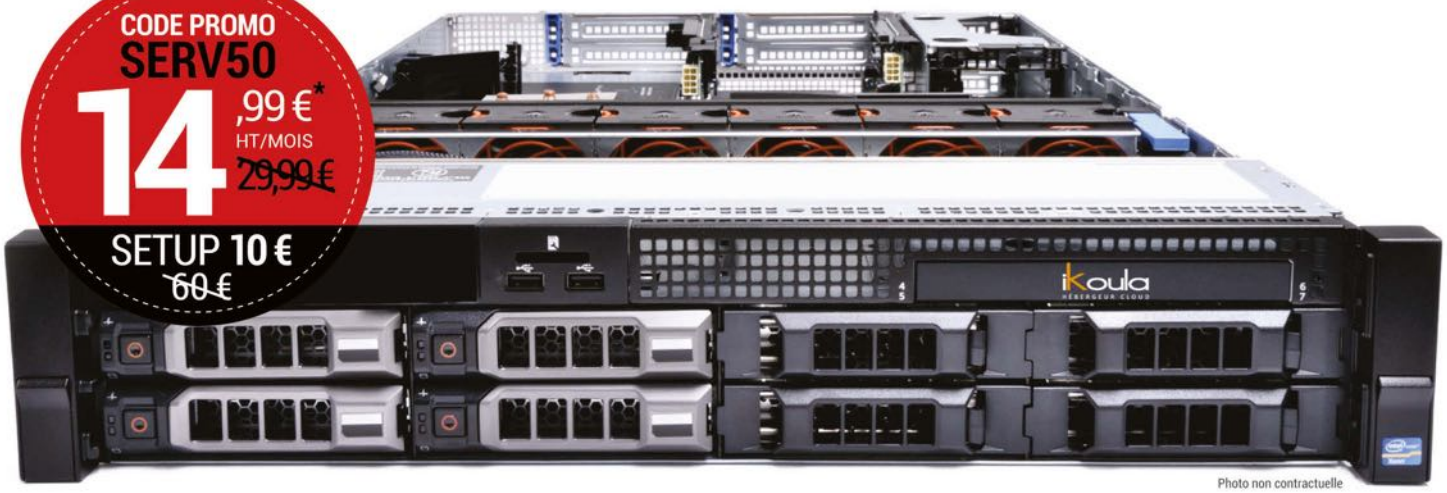



Photo non contractuelle

debian ubuntu CentOS Windows Server 2012

- Intel® Xeon® **E3 1220v5**
- 1 CPU (4C/4T) **@3 Ghz**
- **16 Go** RAM DDR4
- **1 To** SATA
- **GeForce** GT 710 1 Go
- **100 Mbs** Full duplex

**COMMANDEZ SUR**
**https://express.ikoula.com**

*Offre découverte -50 % sur la première période de souscription avec un engagement de 1 ou 3 mois et setup à 10 € HT (valable uniquement sur le plan Xeon® 1220v5 et Xeon® 1230v5, hors options et hors renouvellement). Voir toutes les conditions sur le site.

## CONFIGUREZ VOTRE SERVEUR DÉDIÉ XEON

**PROCESSEUR ▼**

- ● Intel® Xeon® **E3 1220v5 4C/4T @3 GHz**
- ○ Intel® Xeon® **E3 1230v5 4C/8T @3,4 GHz**

**MÉMOIRE ▼**

- ● **16 Go** DDR4
- ○ **32 Go** DDR4
- ○ **64 Go** DDR4

**DISQUE DUR ▼**

- ● **1 To** SATA
- ○ **2 To** SATA
- ○ **4 To** SATA
- ○ **240** Go SSD
- ○ **480** Go SSD
- ○ Disque secondaire

**COULEUR ▼**

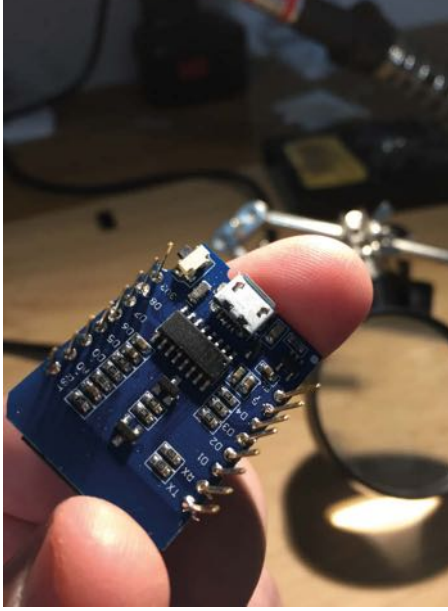
○ ○ ●

ikoula
HÉBERGEUR CLOUD

sales@ikoula.com

01 84 01 02 50

# Wemos D1 Mini :
# 5$ de plaisirs maker intenses !



François Tonic

*Il y a un mois, dans Programmez! 201, nous vous avions présenté très rapidement cette minuscule board : la D1 mini de Wemos. Récemment, la version pro est sortie, au même prix, 5$, mais avec des ressources supplémentaires. L'ESP8266 est plus que jamais une valeur sûre du maker et des IoT !*

Grâce à Sébastien Warin, nous avons découvert les ESP8266, mais ce n'est qu'avec la D1 Mini que nous nous sommes réellement mis à tester la bête. Ce qui saute aux yeux, outre le prix, c'est la légèreté de la carte et sa taille ridiculement petite. Des atouts précieux quand vous faites des IoT. Les ESP ont l'avantage de prendre peu de place et ils sont WiFi par défaut. La D1 Mini garde tous les avantages des ESP et rajoute des ressources matérielles peu communes dans le monde Maker, hormis des cartes de type Raspberry Pi.

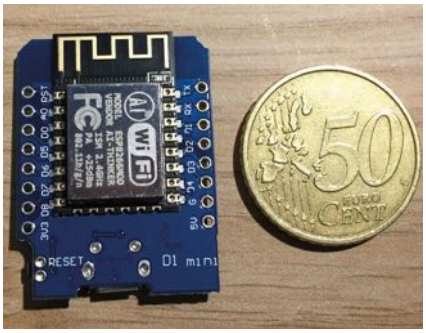
| | D1 Mini | D1 Mini Pro |
|---|---|---|
| microcontrôleur | ESP8266 | ESP8266 |
| broches digitales | 11 | 11 |
| broches analogiques | 1 | 1 |
| Voltage | 3,3V | 3,3V |
| Mémoire flash | 4 Mo | 16 Mo |
| Poids | -5g | 2,5g |

Actuellement, Wemos conseille la D1 Mini Pro, la D1 Mini sera dépréciée dans les prochains mois, mais cette version constitue toujours une excellente base de travail. Une des contraintes fortes de la carte est le voltage opérationnel : 3,3V. Il faut donc être prudent durant les montages. Nous avons grillé des composants et la D1 a connu des surchauffes à cause de mauvais branchements. Elle est moins tolérante qu'une Arduino.

## Une carte nue

Par défaut, la D1 Mini / Mini Pro est livrée avec les hearders non soudés, à vous de le faire. Si vous avez un peu d'expérience en soudure, cela ne posera aucune difficulté. Le câble USB n'est pas livré en standard. Un câble avec un connecteur micro-USB est indispensable.

La Wemos est pleinement compatible avec l'outil Arduino IDE. Si vous êtes habitué(e) à cet IDE, pas de souci, une fois passée l'installation des pilotes…




## Installation de la partie logicielle

Wemos propose une installation des fichiers hardware sur son site (https://www.wemos.cc/tutorial/get-started-arduino.html). Mais nous préférons une autre méthode d'installation :

1 Afficher le panneau préférences de votre Arduino IDE ;
2 Dans le champ URL de gestionnaire de cartes supplémentaires, rentrer l'adresse suivante : http://arduino.esp8266.com/stable/package_esp8266com_index. Json ;
3 Cliquer sur le bouton OK ;
4 Quitter et relancer Arduino IDE ;
5 Dans la fenêtre Gestionnaire de carte (menu Outils -> Gestionnaire de carte) : sélectionner esp8266 (tout en bas) puis cliquer sur installer. L'opération prendra plusieurs minutes.

Les cartes ESP8266 sont maintenant disponibles dans le menu Outils -> Type de Carte, secteur ESP8266 Modules. Il vous suffit de sélectionner WeMos D1. N'oubliez pas de sélectionner le port de connexion. Sur notre poste, il s'agit de COM3. Par défaut, la configuration (CPU, Flash Size, programmateur, Upload Speek) est correcte. Vous pouvez tester la carte immédiatement par exemple avec un simple Blink. Le chargement du code est un peu plus long qu'avec une carte Arduino.

## En pratique

Comme pour les Arduino et les Pi, tout dépendra de vos projets et idées. L'avantage est de posséder une carte WiFi par défaut, facilitant la communication de l'objet vers le Cloud, une app mobile, etc.

Bien entendu, pour certaines utilisations bas niveau, il faudrait utiliser les librairies ESP et non Arduino pures. Et vous allez pouvoir travailler entièrement en sans-fil, évitant d'avoir des fils et surtout des shields réseau encombrants et pas toujours très fiables.

Hackster.io propose quelques montages : https://www.hackster.io/esp/products/wemos-d1-mini

WeMos propose aussi plusieurs shields dédiés : batterie, DHT, moteur, protoboard (nano planche à pain), OLED, carte SD, Led RGB. À cela se rajoute l'ensemble des capteurs du marché. Mais là-encore, attention au voltage !

La D1 mini souffre d'une communauté restreinte en France et parfois le manque de documentation dédiée même si la communauté ESP8266, au sens large, est très active.

Les ESP sont encore marginales en France et auprès des développeurs et c'est bien dommage ! •

## WEMOS D1 : UNE ARDUINO UNO SAUCE ESP

Même si WeMos communique peu autour de cette carte, le constructeur propose aussi une carte qui ressemble à une Arduino Uno, la D1. La D1 reprend les mêmes ressources et GPIO que les minis. Les différences se font sur les dimensions, le poids (25 grammes) et l'alimentation (connecteur dédié).

# Abonnez-vous à programmez!

le magazine des développeurs

## Nos classiques

**1 an** — 11 numéros — **49€***

**2 ans** — 22 numéros — **79€***

**Etudiant** — 1 an - 11 numéros — **39€***

** Tarifs France métropolitaine*

## Abonnement numérique

**PDF** — 1 an - 11 numéros — **35€***

**Souscription uniquement sur**
**www.programmez.com**

**Option :**
**accès aux archives 10€**

## Nos offres spéciales 2017

**1 an offre 2017 — 65€***

11 numéros + clé USB***
+ 1 livre numérique (au choix)
+ 4 n° vintage****

** Au lieu de 126,42 € (1 an : 49 €, clé USB : 34,99 €, livre numérique : 22,43 €, n° vintage : 20 €) / Limitée à France métropolitaine*


**2 ans offre 2017 — 95€****

22 numéros + clé USB***
+ 1 livre numérique (au choix)
+ 4 n° vintage****

***Au lieu de 156,42 € (2 ans : 79 €, clé USB : 34,99 €, livre numérique : 22,43 €, n° vintage : 20 €) / Limitée à France métropolitaine*

**** Clé USB Programmez! 4 Go contenant tous les numéros depuis le n°100*
***** Selon les stocks disponibles. Antérieur au N°168*

**Toutes nos offres sur www.programmez.com**

# Oui, je m'abonne

*ABONNEMENT à retourner avec votre règlement à :*
*Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex.*

☐ *Abonnement 1 an au magazine :* 49 €
☐ *Abonnement 2 ans au magazine :* 79 €
☐ *Abonnement étudiant 1 an au magazine :* 39 €
Photocopie de la carte d'étudiant à joindre

☐ *Abonnement 1 an au magazine :* 65 €
11 numéros + clé USB
+ 1 livre numérique (au choix)
☐ **Scratch** ou ☐ **Arduino**
+ 4 n° vintage

☐ *Abonnement 2 ans au magazine :* 95 €
22 numéros + clé USB
+ 1 livre numérique (au choix)
☐ **Scratch** ou ☐ **Arduino**
+ 4 n° vintage

**PROG 202**
*Offre limitée, valable jusqu'au 2 janvier 2017*

☐ M. ☐ Mme Entreprise : Fonction :

Prénom : Nom :

Adresse :

Code postal : Ville :

*email indispensable pour l'envoi d'informations relatives à votre abonnement*

E-mail : @

☐ Je joins mon règlement par chèque à l'ordre de Programmez !

☐ Je souhaite régler à réception de facture

** Tarifs France métropolitaine*

# GitHub Universe : plongée dans l'univers GitHub

Aurélien GALTIER
*TechLead Software Craftman*
*Manager Studio chez* **Cellenza**

*Du 14 au 15 septembre se tenait la conférence GitHub Universe à San Francisco. Pour l'occasion j'ai pu rencontrer des gens de GitHub et visiter leurs locaux. Beaucoup d'annonces et une philosophie orientée ouverture d'esprit.*

GitHub est une société dont son CEO est Chris Wanstrath. Il en est co-fondateur, il y a maintenant 9 ans. Pendant la visite des locaux de GitHub, nos guides nous expliquent que GitHub est une société qui cherche à démocratiser le code, à partager le code et à échanger. C'est une culture importante et encrée dans la philosophie de GitHub. Avant tout GitHub oriente ces développements pour les développeurs. La stratégie est les développeurs en premier. Ce sont les développeurs qui sont majoritaires dans les sociétés. Mais GitHub n'oublie pas les entreprises avec par exemple GitHub Enterprise et les nouveautés sur la sécurité. GitHub prouve qu'elle a aussi sa place dans le monde de l'entreprise. C'est aussi une société soucieuse du détail. Quand on utilise leur plateforme, on remarque énormément de détails pour améliorer la productivité, pour aider les développeurs, … Si je devais résumer l'esprit de GitHub ce serait : c'est une société ouverte, sur le partage et soucieuse du détail.

## La mascotte : Octocat

Pendant notre visite, Mark Otto (Director of Design chez GitHub) nous a expliqué, que la mascotte est un mélange de pieuvre et de chat. Elle a donc été appelée « Octocat ». Cela décrit parfaitement ce qui peut représenter un merge entre deux commit.

On nous explique qu'avec cette forme l'emblème de GitHub est trans-pays. Ainsi il s'adapte aussi bien en Europe, aux Etats-Unis, … Ce qui fait de cette Octocat une créature mythique connue dans tous les pays.

GitHub a travaillé sur la personnalité de l'OctoCat. « Yes, she is she » nous dit Mark

Otto. OctoCat est une fille. L'éditeur a travaillé toutes les facettes de la mascotte (devant, derrière, assis, debout, etc.). Et une multitude de produits dérivés est disponible. A la conférence il y avait un Octoshop, ainsi ils écoulaient des figurines, posters, t-shirts, vestes, pendant tout l'événement. Et on peut acheter tout cela sur le site https://github.myshopify.com/.

## Les locaux de GitHub

Les locaux de GitHub sont à leur image. Des locaux dans un esprit de partage et d'ouverture. Pendant la visite, un terme dit par Katelyn Bryant m'a marqué : "Social Space". On parle d'espace social. Car les employés de GitHub ne travaillent pas toujours sur place. Ils peuvent travailler de chez eux. Les locaux de GitHub sont vus comme un emplacement pour se rencontrer. Ils peuvent aussi travailler dans différents espaces aménagés pour l'occasion. Mais les locaux de GitHub sont vus comme un espace social, d'échange et de partage. Les locaux représentent la philosophie de GitHub. C'est là que l'on découvre que leur plateforme virtuelle d'échange et de partage est retranscrite dans leur organisation. C'est à mon avis leur grande force. On ressent un plaisir de venir travailler. Un plaisir de venir échanger, créer et construire des applications.

Mais commençons la visite. De l'extérieur on découvre un bâtiment propre avec un logo de GitHub. Cela ressemble à une petite usine dans un quartier rassemblant des sociétés. De l'extérieur les locaux de GitHub semblent simples. La sensation d'un bâtiment basique mais propre. Puis en entrant on découvre une décoration plus sympa : un style industriel.

Puis on est accueilli par Katelyn Bryant (PR Manager chez GitHub) qui nous fait la visite. On nous demande, par respect des employés, qu'aucune photo ne soit prise avec eux.

On commence par le rez-de-chaussée, ou l'on découvre un bar, un billard, une table de ping-pong, des bornes d'arcades, … Mais aussi des tables pour manger, pour faire des réunions, pour les invités. Un rez-de-chaussée orienté social. Une partie de ce grand espace est réservée pour faire des points avec toute l'équipe le vendredi. On découvre une grande statue du penseur en forme d'Octocat. Une grande statue imposante comme on peut en trouver dans certaines grandes sociétés, mais avec un côté comique. La sensation en voyant cela est que GitHub est une grosse société, mais aussi une société fun. Je crois que ce qui motive les employés c'est le fun et la qualité du détail.

Puis on arrive au premier et l'on découvre la boutique GitHub, t-shirts, mugs, … Ainsi on peut repartir avec un souvenir. Le premier étage est constitué d'un grand open space orienté pour les développeurs. Pas beaucoup de monde dans les bureaux, car ils préparent l'évènement. Dissimulés dans des coins il y a aussi une salle de sport, une bibliothèque, des espaces pour téléphoner, des espaces pour les mères et leurs enfants, … On voit donc que tout le confort est là.

On est rejoint par Mark Otto (Director of Design chez GitHub) pour la suite de la visite.

Ensuite on passe au deuxième niveau. On découvre des dessins d'enfants, des petits scénarios de l'Octocat, des illustrations. On tourne à droite sur un grand open space. Plus

orienté, marketing, designer, … Cette open space a la même philosophie que le premier niveau. Puis on suit nos guides dans les escaliers pour atteindre le toit. Et on découvre un endroit sous forme d'îlot perché sur le toit, pour manger, travailler, échanger se reposer. On a la vue sur les buildings de San Francisco d'un côté et de l'autre sur la baie.
Les employés ne travaillent pas toujours sur place et les locaux sont un lieu de rencontre plus qu'un lieu de travail.

## Keynote d'ouverture

La Keynote d'ouverture débute par un dessin animé mettant en scène l'Octocat et des amis de l'Octocat. Car oui on découvre que l'Octocat se prend pour un cosmonaute en sautant sur un trampoline. Et qu'il joue avec un autre Octocat. En regardant ce dessin animé on voit l'Octocat prendre vie. Puis Chris Wanstrath (CEO & Cofounder, GitHub) entre en scène. [1]
L'impression d'un CEO proche des développeurs. Il monte sur scène de façon décontractée. Il nous explique que cela fait 9 ans que le premier commit a été fait sur GitHub. Il insiste sur le fait que GitHub travaille pour les développeurs. Que GitHub est une plateforme d'échanges pour les développeurs.
Il émet un focus sur « GitHub Enterprise ». Il explique la philosophie de GitHub en entreprise. Ce n'est pas GitHub Enterprise vs GitHub open source, mais open source en entreprise. Ainsi le discours est de l'open source interne à l'entreprise, et de l'open source externe à l'entreprise. GitHub Enterprise est la plateforme qui permet de démocratiser l'open source dans une entreprise. GitHub Enterprise possède toutes les fonctionnalités de GitHub pour de l'open source. L'idée est de capitaliser sur les développements faits sur la plateforme open source. Ainsi la plateforme publique permet de capitaliser sur les performances, les fonctionnalités et les développeurs. La stratégie de GitHub est d'évangéliser les développeurs avec sa plateforme publique pour ensuite que ces mêmes développeurs utilisent la plateforme en entreprise.
Ensuite il nous parle de différents projets communautaires. Parle du CryEngine, de DOOM. Mais aussi de Machine Learning, Tensorflow, Chart.js, … Avec tous ces projets Chris montre que GitHub n'est pas fermée à un seul type de projet. On a des jeux, des algos, des outils, … En somme GitHub est une plateforme universelle pour héberger du code open source.
On sent que Chris est fier que GitHub permette de partager autant de codes sources importants et utilisés. Il fait même un focus sur Apollo. Le code source des premiers projets lunaires est sur GitHub. Des projets qui ont permis de faire avancer les recherches dans le monde de l'aéronautique.
Entre GitHub Enterprise et tous les projets communautaires montrés par Chris on se rend compte que GitHub est une plateforme incontournable. Tous les développeurs ont aujourd'hui utilisé la plateforme. Que l'on utilise la plateforme pour consulter, partager ou échanger, GitHub est une plateforme indispensable.
Quelques chiffres :

- 5,8 millions d'utilisateurs actifs
- 19,4 millions de repository
- Microsoft possède le plus de contributeurs : 16 419

## GitHub for student et les annonces

Chris nous parle maintenant de la vision de GitHub pour les années à venir. Il nous explique que GitHub veut aider à démocratiser le développement. Ainsi il veut que tout le monde soit capable de développer des applications. Pour cela il parie sur l'avenir et propose donc une plateforme pour les étudiants. La plateforme « Student Developper pack », permet à tous les étudiants d'avoir accès à des outils de manière gratuite (https://education.github.com/pack). Ce pack est une stratégie importante pour l'avenir de GitHub. Ils veulent former les développeurs de demain, les développeurs qui utiliseront GitHub pour partager du code.
GitHub met aussi en place une plateforme pour donner des cours en ligne. Avec « GitHub Classroom » (https://classroom.github.com/). On peut créer un repos Git et faire rejoindre des gens pour leur distribuer des exercices. Ce qui permet de distribuer des cours via ces exercices enregistrés sur des repos.
Puis pendant la conférence Chris nous présente un projet important « Black Girls CODE ». Kimberly Bryant rentre sur scène pour nous parler de son projet. L'idée de « Black Girls CODE » part du constat que très peu de femmes sont développeuses. Et qu'encore très peu de femmes noires sont développeuses. Ainsi Kimberly souhaite accompagner les femmes à devenir développeuses. D'augmenter le nombre de femmes dans les technologies. Pour que dans l'avenir on soit à parts égales dans le monde du développement logiciel. Encore un beau projet soutenu par GitHub et qui rejoint la stratégie de former les futurs développeurs. On passe aux annonces, Chris commence par nous expliquer qu'il y a trois annonces. Il commence par nous parler d'intégration. Ensuite il passe aux nouveautés buisiness, et termine par parler des Workflow. Puis commence par nous parler de la conclusion. A ce moment il nous annonce qu'il a une autre chose à nous dire. Et il nous parle des nouveautés sur les pull requests. Il recommence en nous faisant le même coup de la conclusion et nous annonce les nouveautés sur les profils.


1

## Code review

Les pull requests sont une très grande force de la plateforme. GitHub a amélioré les pull requests en ajoutant des notions de code review. Ce qui permet de spécifier un petit commentaire associé à un bout de code.
Au moment de rajouter un commentaire à cette pull request, on peut maintenant préciser si on veut approuver la pull request en même temps ou si l'on souhaite que la pull request soit améliorée avant de faire un merge dans la branche principale.
Les revues de codes contiennent aussi du markdown et des émoticônes. On a donc des fonctionnalités de revues de codes avancés, ce qui renforce le côté social de la plateforme.
Les merges ont eux aussi des évolutions. Maintenant un merge sur une branche peut être protégé. Avec de l'intégration continue, de la review, ou d'autres outils. Ce qui permet de renforcer le contrôle avant de faire un merge automatique. Chris précise que bien d'autres fonctionnalités arriveront dans le code review.
Le système de pull request de GitHub est à mon sens la partie la plus importante de GitHub. On voit bien que la société n'abandonne pas la fonctionnalité et ne se repose pas sur ce qui est déjà fait. Mais tout au contraire ren-

force ses acquis. Ce qui place GitHub toujours en première position dans le monde de l'open source.

## Projects

Un nouvel onglet apparaît dans GitHub. L'onglet Projects permet de gérer ses projets avec un board. Ainsi on retrouve nos « issues » sous forme de cartes que l'on peut déplacer d'une colonne à une autre. Avec cette fonctionnalité GitHub commence à combler son manque sur les issues par rapport à des outils externes. Toutes les issues peuvent être affichées dans les colonnes. Les colonnes sont modifiables et peuvent être réorganisées. Il est facile d'adapter le board à son propre process de travail. [2]

C'est une très belle nouveauté. Ce board permet à GitHub de montrer que sa gestion des issues évolue. Que l'on peut maintenant planifier nos issues. Ce qui permet à nos projets d'avoir une planification.

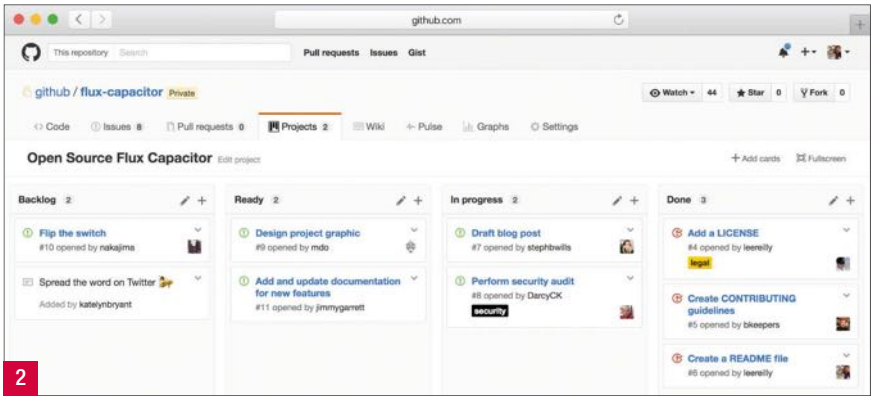
## Profil

Les profils évoluent. Chris nous explique que GitHub est une communauté. Et à ce titre le profil de ses membres est important. On retrouve maintenant une petite bio pour nous décrire. La possibilité d'accrocher nos repos de code préférés et de les réorganiser. Ainsi on customise comme on le souhaite notre profil. La timeline a été revue. Elle est plus complète et plus détaillée. On découvre une page de profil orientée social. Elle affiche nos premiers push, notre premier jour dans une société. Ainsi ce nouveau profil raconte notre histoire de développeur.

## Platform

En utilisant GitHub, on peut utiliser tous les outils que l'on souhaite. Ainsi Chris insiste sur le fait que GitHub peut être utilisé avec n'importe quel outil. Il explique que 60% des organisations possèdent des intégrations. C'est à nous de choisir notre serveur de build. A nous de choisir les outils que l'on connaît ou préfère le plus. Puis il parle de la nouvelle API de GitHub, GitHub GrapQL API. GitHub possède maintenant une API GrapheQL. GitHub évolue et propose une API évoluée. GrapheQL est un projet créé par Facebook. GitHub a décidé de l'intégrer comme une part importante dans leur architecture. Ainsi on peut presque tout faire grâce à cette API. [3]

On sent bien que GitHub cherche à s'ouvrir. La société cherche à faire de sa plateforme un environnement majeur, compatible et ouvert avec



2

d'autres produits. En offrant cette nouvelle API, on sent une plateforme très axée sur l'ouverture.

## Business

GitHub n'oublie pas les sociétés. Ainsi GitHub comble ses manques en proposant deux nouveautés au tour de la sécurité. La première est l'ajout du Two-factor authentification. Ce qui permet de sécuriser nos comptes avec l'envoi d'un code par SMS ou TOTP app (Time-based One-time Password Algorithm). [4]

Et la deuxième partie pour les entreprises est pour moi encore plus importante. GitHub va proposer un single sign-on pour pouvoir intégrer les authentifications des entreprises au sein de GitHub. Ce n'est pas encore disponible, mais à l'avenir on pourra se connecter avec notre compte d'entreprise pour accéder à notre espace GitHub. L'utilisation de SAML permettra d'intégrer toutes les solutions d'authentifications possibles. C'est un point très fort pour GitHub ainsi on pourra continuer à utiliser notre compte d'entreprise standard pour se connecter à GitHub. Et l'entreprise pourra gérer l'accès au GitHub de manière centralisée.

## Community

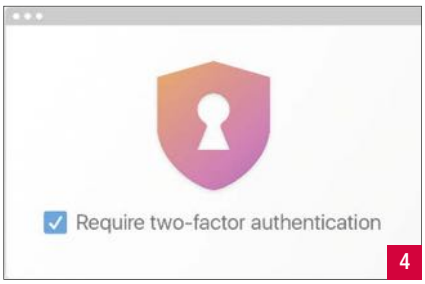
GitHub annonce des forums. La société va proposer début 2017, 3 forums pour échanger sur l'éducation, la plateforme et les communautés.

Avec « Education Forum » GitHub renforce sa volonté de faire apprendre le développement à tout le monde. C'est une stratégie sur du long terme qui permet à GitHub de promouvoir sa plateforme et d'accrocher les futurs étudiants.

Le « Github platform forum » va permettre à tous les passionnés d'échanger avec les développeurs de GitHub. Là encore GitHub fait preuve d'ouverture. Avec ce forum il va être possible d'échanger directement avec les développeurs derrière GitHub.

Le « Community Forum » permettra à tous les développeurs de discuter et d'échanger entre eux. GitHub renforce avec ses forums la notion de développeur social. Ainsi le monde de l'open source va prendre un tournant social.


3



4

## Keynote d'ouverture jour 2

La deuxième journée s'ouvre avec une plénière animée par Nicole Sanchez (VP, Social Impact chez GitHub). Elle nous raconte une histoire comme quoi elle est surprise que des musées soient ouverts et gratuits. Elle fait la relation avec GitHub. Après nous avoir montré le pouvoir de l'open source et du code open source, elle nous explique que GitHub veut investir sur les nouveaux développeurs : on peut investir sur les jeunes mais aussi sur les personnes qui une fois adultes souhaiteraient apprendre à coder. Ainsi Nicole nous parle de différents projets qui permettent de démocratiser le développement d'applications.

Puis elle évoque le projet Operation Code (https://operationcode.org/). Il propose à tous les gens qui ont servi leur pays d'avoir accès au code. David Molina (Founder & Executive Director, Operation Code) nous raconte son histoire : après avoir servi son pays il est retourné à la vie civile, et a appris à coder. Maintenant, il souhaite aider tous les gens qui se trouvent dans le même cas que lui. Il explique aussi qu'il a compris que le monde de l'open source était très important. Nicole Sanchez termine en expliquant pourquoi le monde de l'open source est

important. Et pourquoi GitHub est important. GitHub permet de connecter les gens. Permet de connecter les nouveaux développeurs. Cela permet de créer le futur des logiciels.

## Nos rencontres

### Kakul Srivastava : évangéliser l'open source en entreprise

Kakul Srivastava (@kakuls) est VP of Product Management chez GitHub. Elle a la responsabilité de vendre la plateforme. Quand on lui pose la question, qu'appelle-t-on produit chez GitHub ? Elle répond que GitHub est un produit. C'est un produit pour l'open source et les entreprises. GitHub est vue comme un seul et même produit disponible de deux façons différentes.

Comment faire pour démocratiser l'open source en entreprise ? L'explication selon elle est d'utiliser GitHub en entreprise. Une fois que les développeurs auront l'habitude d'utiliser GitHub en entreprise, ils seront plus amenés à utiliser GitHub en open source.

Comment faire pour que GitHub soit utilisé en entreprise ? Elle explique qu'Il y a deux façons d'aborder le problème. Cela peut passer par des référents techniques ou chefs de projets qui souhaitent utiliser GitHub. Mais aussi les développeurs. Les développeurs sont majoritaires dans les entreprises. C'est surtout eux que GitHub vise. La plateforme est adaptée pour les développeurs. Car ce sont eux qui vont pousser les entreprises à adopter GitHub.

GitHub est fait pour les développeurs. GitHub est fait pour l'open source. Mais les développeurs en utilisant GitHub en entreprise vont évangéliser les sociétés à l'open source.

En entreprise ou chez soi, on utilise les mêmes outils. GitHub et GitHub Enterprise sont donc un seul et même produit. Ce produit répond aussi bien au monde de l'open source, mais aussi très bien à l'entreprise avec des concepts d'inner-sourcing.

### Brandon KEEPERS, mission open source

Brandon KEEPERS (@bkeepers) est un défenseur de l'open source. Il est un ingénieur de GitHub et croit en l'open source. Malgré cela il n'est pas fermé au code privé. Pour lui, l'open source n'est pas incompatible avec du code privé en entreprise.

Brandon croit en l'open source car l'avantage des projets open source est la disponibilité de milliers de développeurs. Cela permet par exemple à GitHub d'initier des projets. Ces projets seront repris et les contributeurs se multiplieront. Beaucoup de sociétés ne souhaitent pas mettre leur code en open source. Il explique sa stratégie pour qu'une société fasse de l'open source. La première chose à faire est de leur expliquer les avantages. Il faut qu'elles comprennent que l'important n'est pas le code du produit, mais dans le métier et les données. On n'est pas obligé de mettre tout le code en open source, certains morceaux de codes peuvent rester fermés. Lorsqu'une société réalise un projet en open source, c'est un signe de stabilité, de performance et de transparence. GitHub Enterprise permet de faire de l'open source en interne d'une entreprise. On appelle cela de l'Inner-sourcing. L'Inner-sourcing est une ouverture des projets à l'intérieur d'une entreprise. Il n'y a pas de contradiction entre du code privé et du code public. D'après Brandon il faut les deux pour qu'une entreprise rencontre un succès. Les entreprises qui utilisent GitHub en interne auront tendance à l'utiliser en externe. Et la place du mouvement craft pour GitHub ? Il répond que la qualité d'un code est plus importante lorsque l'on fait de l'open source. Votre code est disponible et lu par des centaines de personnes. On doit avoir un code lisible et une bonne documentation. Quelle proportion de code open source versus code privé chez GitHub ? Il répond que la taille de code open source de GitHub est inconnue. Mais un maximum de projets passent en open source. Vous pouvez modifier, utiliser les composants visuels de GitHub, les bases de données, les outils, … Le cœur de GitHub est dans les données, le savoir-faire et la plateforme, …

### Sam LAMBERT : l'importance des performances

Sam Lambert(@isamlambert) est Senior Director of Infrastructure Engineering chez GitHub. Depuis deux ans, il travaille sur les performances et la disponibilité de la plateforme GitHub. Pour réussir ce travail, il se focalise sur l'organisation des équipes. Il pense qu'il faut avant tout s'entourer de bons profils.

GitHub utilise des serveurs physiques et ses propres datacenters. Ceci leur permet de contrôler les performances de la plateforme. Le code est stocké directement sur disque, tandis que le reste est stocké sur MySQL. Par exemple l'authentification pour accéder au code utilise une base MySQL pour vérifier les droits d'accès. GitHub utilise le projet interne Spoke. Il permet de répliquer l'ensemble des projets GitHub : un projet GitHub est répliqué trois fois. Deux réplications en lecture/écriture et la dernière en lecture seule.

Toutes les parties de GitHub sont importantes. Le hardware comme le software. Il explique qu'ils ont une très bonne équipe pour les bases de données MySQL et que les performances sont un problème à tous les niveaux. L'équipe ne fait pas un focus sur un composant en particulier mais sur l'ensemble. Comment garantir les performances sur la plateforme entreprise ? Il y a deux possibilités, la première est GitHub Enterprise dans les datacenters de GitHub. Dans ce cas GitHub suit les performances régulièrement et peut intervenir. La deuxième solution est un hébergement dans les locaux du client. Du coup le client remonte régulièrement les informations de performances à GitHub. Ceci leur permet de vérifier et garantir que leur plateforme fonctionne correctement même dans un environnement dédié.

## Résumés de sessions techniques

### Git Flow : le workflow selon GitHub

On commence par nous montrer les icônes qui représentent le workflow. Dans la salle peu de gens sont capables de dire à quoi sert chaque icône… Partant de ce constat Matt Desmond (Trainer, GitHub) et Eric Hollenberry (Trainer, GitHub) nous expliquent chaque icône et donc le workflow traditionnel qui fait la force de GitHub en open source.

Pour expliquer, ils choisissent l'exemple d'une chaîne de restaurants dont l'activité est de faire le burger du jour. Dans l'exemple la personne souhaite ouvrir un autre restaurant qui fait aussi un burger du jour, mais pas forcément le même. Donc elle force le restaurant à modifier la carte et à proposer son propre burger du jour. Elle souhaite soumettre cette idée au premier restaurant. Elle fait donc un pull request. Le premier restaurant fait des remarques, une discussion est entamée entre les deux restaurants pour déterminer lequel des deux burgers est le burger du jour, c'est la phase de review. A l'issue de cette phase, après discussions, on peut mettre à jour le burger du jour, c'est la phase de merge.

#### *Fork*

La première étape consiste à faire un fork du projet que l'on souhaite modifier. La notion de fork consiste à dupliquer tout le repository git pour en créer un nouveau.

Dans le repos source, pas le droit de faire des modifications. Dans le nouveau repository, il est possible de faire des modifications. Ce nouveau repos nous appartient.

### *Commit*

La seconde étape est de faire une ou plusieurs modifications dans notre nouveau repos. Ils nous expliquent le workflow de Git. On peut travailler à plusieurs avec des branches différentes. Ainsi dans ce nouveau repos on peut utiliser ce repos comme s'il était le nôtre.

### *Pull Request*

La création d'un pull request permet d'ajouter un sujet et une description lors de la demande de modification. Ainsi l'idée est de commencer une discussion avec le ou les responsables du repos originel. Elle permet aussi pour les propriétaires du repos d'avoir des indications automatiques sur la « Pull Request ». On peut savoir si la modification crée des conflits avec la branche actuelle. Si l'usine de build a été exécutée et si les tests passent.

### *Merge*

Une fois toutes les discussions terminées sur la demande et son code associé, les contributeurs peuvent décider d'intégrer la modification dans la branche principale. Il y a deux solutions. Soit l'intégration se fait sans problèmes et le nouveau code est disponible directement. Soit il faut corriger le code ; on va donc extraire la branche du pull request, et avoir la main sur le code en local. Une fois que l'on a fait une correction dans le code celui-ci devient disponible au merge dans la branche principale.

## Implementing and Using GraphQL at GitHub

On commence par une présentation de GrapheQL. GrapheQL est un format d'API fourni par Facebook. L'idée est d'éviter que notre API ne renvoie trop d'informations à chaque fois que l'on appelle cette API. Avec un langage particulier on va déterminer quelles informations l'API doit renvoyer. Par exemple si l'on veut renvoyer un utilisateur avec son nom, il suffit de faire la requête suivante :

```
{
  me {
   name
  }
}
```

Ce qui aura pour effet de produire le résultat :

```
{
  "me": {
    "name": "Luke Skywalker"
  }
}
```

GitHub fournit une API GraphQL. Cette API est utilisée en interne. Ainsi toutes les fonctionnalités de GitHub peuvent être accédées via l'API.
La documentation de l'API est disponible à l'adresse https://developer.github.com/early-access/graphql/

## Making Electron Development Simpler, More Pleasant, and More Productive

Machisté Quintana, Software Engineer chez Slack, nous explique comment faire des applications avec Electron. Et comment ces applications peuvent être performantes.
On nous rappelle le principe d'Electron. L'idée est d'embarquer un viewer Web dans une application de type client lourd. Puis ensuite de faire tout notre code dans du JavaScript. Ce même JavaScript peut utiliser des dépendances qui lui permettront d'avoir l'accès à des composantes systèmes. Electron permet de donner des supers pouvoirs à une application JavaScript. Pour avoir une application Electron performante, ils ont développé une couche « electron-compile ». L'intérêt de cette couche permet de transpiler à la volée les fichiers JavaScript utilisés. Si l'on fait du ReactJS avec les dernières normes JavaScript et les includes, le JavaScript sera transpilé dans une version plus simple qui, elle sera interprétée par Electron. Mais cela pose la question des performances pour cette techno. Il nous explique que le code transpilé est gardé en cache. Tant que l'on ne change pas le code du fichier, le compilateur utilise la version en cache.

## Tips & Tricks: Gotta Git Them All

Pendant cette session, Jamie Strusz et Brent Beer nous présentent des trucs et astuces sur les commandes Git et GitHub.
Par exemple la commande « git status -sb » qui donne une version plus simplifiée des modifications faites dans le repos. C'est une commande qu'ils utilisent sous forme d'alias qui leur permet d'avoir une visualisation des branches en ligne de commandes via « git log --graph …» .
Il y a aussi une commande « git config --global help.autocorrect 10 » qui permet de changer la config de git pour faire de l'autocorrection sur les commandes git. Ainsi, si on tape « git comit » il va corriger en « git commit » automatiquement. « git add -p » qui permet de revoir les modifications que l'on a effectuées avant de les commit. Ensuite on nous parle de « git lfs » (Large File Storage). Cela nous permet de stocker des gros fichiers dans notre repos git. Mais ces fichiers vont être mis sur un serveur dédié aux gros fichiers. Cela permet de gérer les fichiers de grandes tailles dans un repos git. On nous fait remarquer que GitHub marche aussi avec du SVN. La possibilité de faire le différentiel entre deux releases. Spécifier une pull request avec de l'auto complétion dans les commentaires. Quelques démonstrations des filtres de recherche sur des pull request et du code. Un déballage de trucs et astuces pour utiliser Git et GitHub. On apprend aussi que Gists, le copier-coller de code de GitHub, crée un repos complet pour chaque bout de code. Ainsi on peut le cloner, le modifier et repousser les modifications.

## Building Reliability with Spokes

Spokes répond à plusieurs objectifs :

- Un serveur plante ne donne aucun impact ;
- Deux serveurs plantent ne donnent qu'un impact minimal ;
- Balance disk et CPU ;
- Automatique resynchronisation et réparation ;
- Scale horizontale ;
- Opérations de maintenance avec un impact zero ;

Pour répondre à cette problématique Spokes réplique les répertoires git. Chaque répertoire git est répliqué trois fois. Un en lecture seule et les deux autres en écriture. Les répertoires disposent de hash, ce qui permet de savoir si le répertoire est corrompu ou s'il est à synchroniser. Si le hash n'est pas correct le répertoire est entièrement recopié. S'il est correct, il suffit de faire le différentiel. Ensuite lorsqu'une requête est faite par un utilisateur, l'infrastructure de GitHub va faire un tri sur la liste des réplications du répertoire. Une fois cette liste triée on utilise le serveur qui est en premier. Le tri de cette liste de serveurs permet de garantir que le serveur utilisé est celui qui est disponible.
Pour gérer les serveurs occupés, leur stratégie est de les gérer comme s'ils étaient hors ligne. Ainsi si un serveur met trop de temps à répondre, il est considéré comme hors ligne. Ce qui permet de réduire la charge sur ce serveur. Puis, du coup, il revient en ligne à ce moment-là il sera de nouveau disponible dans la liste des réplications de serveurs.Pendant les démonstrations pour faire une interruption de service il utilise slack pour suivre le graphique d'interruptions. Il nous démontre que GitHub est capable de suivre l'évolution de leur plateforme. C'est un gage de qualité pour GitHub. Pendant cette session on comprend que leur solution Spokes répond aussi bien à la plateforme en ligne qu'à la plateforme GitHub Enterprise.
Si l'on veut en savoir plus on peut consulter le site http://githubengineering.com/. •

# Découvrir GitHub

Partie 1

Paquet Judicaël
*DSI chez Batiwiz, Judicaël est spécialiste dans le coaching agile et l'architecture logicielle/devops*

*Devenu un incontournable du monde de l'open source, Github est une plateforme de contrôle de version et de collaboration basée sur l'excellent Git, gratuit pour tous développeurs souhaitant contribuer à leur manière au monde open source.*

## Qu'est-ce que Git ?

Pour les personnes n'étant pas encore familières à ce monde open source, Git est un logiciel de gestion de versions décentralisé en ligne de commande. Il a été créé par Linus Torvalds en 2005 qui était déjà le créateur du noyau de Linux en 1991. Ce logiciel libre est distribué en GNU GPL v2. En 2016, Git est devenu le logiciel de versionning le plus utilisé dans le monde avec 12 millions d'utilisateurs, passant devant SVN qui était très populaire il y a 10 ans.

Git a su s'imposer par rapport à SVN pour plusieurs raisons :

- Son système de branche est intelligent et permet de changer de branche instantanément. Les IDE du moment profitent de cette fonction pour améliorer le changement de branche (SVN ne le permettait pas).
- Contrairement à SVN, Git fonctionne de façon décentralisée (vous comprendrez plus loin pourquoi).

## Qu'est-ce que Github ?

Github est une interface graphique qui simplifie sa gestion et qui propose de nombreux ajouts complémentaires très utiles que nous verrons plus tard : https://github.com/. Il se base sur le logiciel de versionning Git et propose un ensemble d'outils collaboratifs utiles pour vos projets. Mais Github c'est bien plus que ça, c'est une mine d'or de projets open source que toute personne en informatique connaît : Nodejs, redis, Mongo… Vous pourrez devenir vous-même un contributeur acharné de projets open source car Github simplifie le travail des contributeurs.

### Travail collaboratif

Github va vous permettre de créer des dépôts de codes (appelés repository) afin de travailler en groupe sur un même projet ; contrairement à un dossier classique d'un système d'exploitation, Github permet grâce à l'intelligence de Git de ne pas effacer le travail d'un collègue mais de fusionner l'ensemble des travaux réalisés par l'équipe contributrice.

Cela permettra de garder également un historique de l'ensemble des modifications réalisées par l'équipe.

### Gérer des remontées de contributeurs

Github propose également des « issue » pour déclarer des problèmes de code et communiquer dessus entre développeurs. On peut vulgairement le comparer à un forum simplifié utilisé comme gestionnaire de bug.

### Le pull-request

Cet outil de Github permet aux éventuels contributeurs de demander une revue de code afin d'intégrer les modifications au sein du projet. C'est très complet car il est possible de demander aux différents développeurs de valider ou non et de faire également des validations automatisées avec des outils d'intégration continue tels que Travis ou Style CI.

### Le wiki pour chaque repository

Vous pourrez créer une petite documentation pour vos dépôts afin de mettre un peu d'aide en ligne pour les utilisateurs de vos projets.

### D'autres options utiles

Github propose également d'autres choses comme Gist qui permet de partager des morceaux de code, des pages de statistiques sur les dépôts et d'avancement sur les pull requests.

Nous verrons également plus tard des fonctionnalités telles que les webhooks qui permettent de déclencher des actions automatiquement vers d'autres outils comme Packagist, Travis ou Sensiolab Insight.

### Github propose des solutions payantes

Si vous utilisez Github pour faire des dépôts publics, l'ensemble de la solution est gratuit. C'est cette solution qui a permis à l'open source d'exploser ses 5 dernières années.

Pour les personnes désirant avoir également des dépôts privés, Github propose un pack à 7$/mois. Cela peut vous permettre de mélanger des projets privés et vos projets publics.

Pour obtenir des comptes plus avancés avec une gestion d'équipe, il faudra débourser 9€/mois/utilisateur (25€ pour les 5 premiers utilisateurs). Ce type de compte est plutôt fait pour des entreprises que pour un particulier. Un pack encore plus complet est proposé pour les entreprises qui ont beaucoup de développeurs pour 21$/mois/utilisateur contenant des outils complémentaires intéressants tels que les authentifications LDAP, SAML ou CAS, un support 7 jours sur 7, des outils d'administration avancés et d'autres configurations avancées.

## Quelques manipulations possibles

### Logiciel Github Desktop

Sur Windows et Mac, vous pouvez installer le logiciel de Github pour gérer vos différents dépôts en local. Vous pouvez le télécharger sur ce lien : https://desktop.github.com/

Celui-ci permet de gérer vos commit/push, vos synchronisations et permet également de gérer vos dépôts qui sont hébergés chez d'autres plateformes équivalentes comme BitBucket.

D'autres outils comme tortoise SVN ou les plugins Git intégrés aux IDE (celui de PhpStorm est excellent) peuvent vous aider dans vos développements. Cependant, utiliser Git en ligne de commande n'est pas si complexe que ça et est loin d'être insurmontable.

### Manipuler un fichier en ligne

Sur Github, vous pouvez l'ouvrir dans Github Desktop, le modifier en live ou le supprimer. Chaque changement apporté sur le fichier vous imposera de faire un commit de celui-ci.

Vous aurez à ce niveau-là plusieurs informations sur le fichier comme son poids, sa dernière modification voire vous avez accès à son histo-

rique complet ce qui peut être fort utile. Pour avoir cet historique sur votre poste en ligne de commande vous pourrez faire :

```
git log monfichier
```

En ajoutant l'option -p, vous pourrez également lister les lignes qui ont été modifiées sur chaque version.

### Créer une branche ou un tag en ligne

Ça parait évident mais l'interface de GitHub permet de créer des branches et des tags sous le nom que vous désirez.

### Créer une organisation

Dans Github, il est possible de créer une organisation pour partager des dépôts avec différentes personnes de la même communauté. Vous pourriez par exemple créer une organisation pour tous les élèves d'une école informatique.
Si vous avez un compte privé, cela ne se répercutera pas sur votre organisation. Chaque organisation imposera de prendre la version payante pour avoir des dépôts privés même si vous êtes vous-même avec un compte payant.
Les dépôts de votre organisation s'appelleront [organisation]/[dépôt] et vous pourrez faire quasiment la même chose avec une organisation qu'avec votre compte Github ; par contre celui-ci peut être partagé avec plusieurs utilisateurs.

### Mine d'or de projets open source

Github propose une page Web classique pour voir de nombreux projets en open source par les utilisateurs de Github : https://github.com/explore. Vous pourrez par exemple y trouver les sources de Docker, Vagrant, Ansible, Nodejs, Symfony, Redis, MongoDb…
Si avec ça, vous ne trouvez pas votre bonheur, je ne sais plus quoi faire pour vous. Nous verrons à la suite de cet article d'ailleurs comment contribuer à ces projets si l'envie vous venait.

### Un peu de marketing dans tout ça ?

Github est une plateforme avec une interface sobre mais qui peut devenir jolie. Je sais qu'un grand nombre de développeurs ne sont pas du tout intéressés par cela, mais il est impératif d'améliorer votre page d'accueil. Mettre une belle photo ou un logo, créer une description complète de qui vous êtes (ou de votre organisation), mettre un lien, voire une localisation, peut être un tremplin pour votre projet.
Plus votre page d'accueil fait pro, plus les gens s'intéresseront à votre projet ; je pense que c'est même aussi important que l'intérêt de votre projet. Je vous conseille fortement de faire également un README.md de qualité car il apparaît sur la page principale de votre projet.
Voici un exemple pour faire un titre (avec les ===), un sous-titre (avec les ----) et du texte :

```
Authorization
=============

Installation
------------

content
```

Il y a d'autres options mais je vous laisserai aller les découvrir par vous-même. Il existe de nombreux outils qui peuvent certifier la qualité de votre code, dont un que nous verrons plus tard dans cet article. Je vous conseille également de mettre les petits badges dans votre README.md qui permettront de montrer aux visiteurs que votre code est de qualité. Voici un exemple concret sur un de mes projets (n'hésitez pas à utiliser le code source de ce README.md pour vos projets) : https://github.com/judicaelpaquet/authorization.
J'ai fait un tour très rapide sur le wiki mais dès qu'un projet se complexifie, le README.md ne suffit pas et je vous conseille de consacrer un peu de temps à l'élaboration d'une documentation claire. Ne pas le faire vous fera perdre beaucoup de contributeurs et d'utilisateurs potentiels.
Vous l'aurez compris, au-delà de votre savoir-faire de génie, n'oubliez pas de marketer un minimum votre espace pour donner un aspect sérieux à vos projets.

## Gestion de mon premier dépôt

### Mon premier fichier envoyé

Maintenant que vous avez compris les possibilités de GitHub, nous allons nous lancer dans la pratique. Le nom d'un projet sous GitHub est sous deux formes : [pseudo]/[dépot]. Par exemple, pour un de mes projets, j'ai judicaelpaquet/authorization où mon pseudo est judicaelpaquet et le dépôt est authorization.
Commencez par vous créer un compte sur Github à cette adresse : https://github.com/. Dès que vous êtes sur votre compte, commencez par créer un nouveau dépôt (new repository) que nous allons appeler test et qu'on mettra en public afin de proposer en open-source notre code qui sera disponible à l'adresse https://github.com/[pseudo]/test.
Juste pour votre information, nous aurions pu créer une nouvelle organisation pour faire des projets sous la forme [organisation]/[projet] au même endroit que pour la création de dépôt. L'organisation permet d'avoir plusieurs administrateurs sur votre projet.
SI vous n'avez pas encore Git, je vous propose de l'installer en choisissant le système d'exploitation correspondant au votre sur le site de l'auteur : https://git-scm.com/downloads.
Pour ce tutoriel, j'ai décidé de vous proposer de tout faire en ligne de commande (shell ou dos) afin de vous familiariser plus facilement avec Git. Cependant, vous pourrez utiliser des logiciels comme Tortoise Git, le plugin Git des IDE, voire le logiciel Github qui permet aussi de faire le job. Git installé, nous allons faire le commit d'un premier fichier README.md où vous allez mettre le contenu désiré. Ce contenu s'affiche sur la page principale du dépôt automatiquement dès que[…] le fichier est sur GitHub.

```
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/[pseudo]/test.git
git push -u origin master
```

**git init** permet de créer un nouveau dépôt en local dans le dossier où vous vous situez.
**git add** permet d'ajouter un fichier qu'on voudra mettre sur notre dépôt local dès qu'on aura envoyé celui-ci. Si vous ne voulez pas envoyer des fichiers utilisés pour des tests personnels, c'est très pratique de bien cibler les fichiers que vous voudrez ajouter à votre dépôt local. Si vous voulez tout envoyer, vous pouvez faire un git add *.

**git commit** permet d'envoyer votre fichier dans votre dépôt local. L'option -m permet d'ajouter un message ; sans celui-ci, il vous faudra en rajouter un sur l'étape suivante sous un éditeur vi (même sur dos).
**git remote** permet de relier votre dépôt local au dépôt de votre Github. Ici on va donner un alias origin et l'URL correspondante.
**git push** permet d'envoyer notre dépôt sur notre Github (celui indiqué par notre alias).
A présent si vous retournez sur votre dépôt Github, vous verrez votre fichier dessus.

## Cloner un dépôt

Cette méthode est différente car le but est de copier un projet existant afin de participer activement à celui-ci. Vous pouvez trouver le lien du git à cloner sur la page principale de votre dépôt https://github.com/[pseudo]/test ou prendre son URL et rajouter tout simplement .git derrière.
git clone https://github.com/[pseudo]/test.git test2
Nous créons notre copie de dépôt dans le dossier test2.
Allez dans ce nouveau dossier test2, créez un fichier ADD.md avec le contenu de votre choix, et envoyez votre modification sur Github :

```
git add *
git commit -m "second commit"
git push -u origin master
```

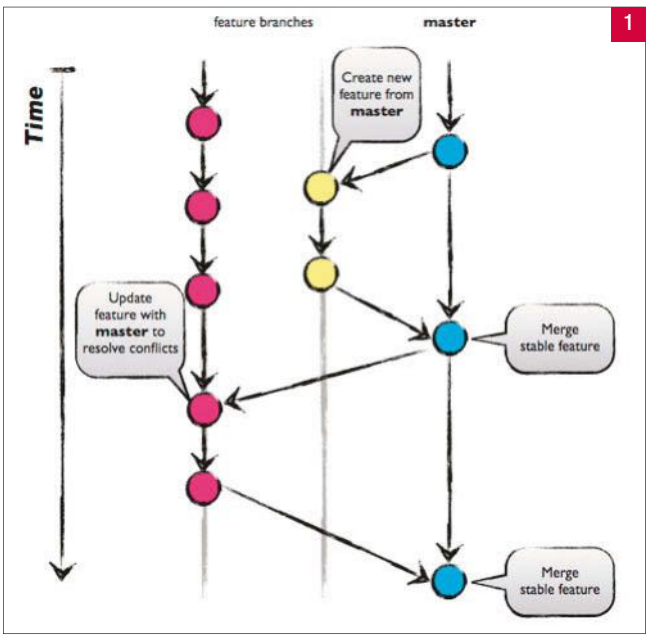
A présent nous avons deux fichiers dans notre Github.

## Récupérer les modifications

Afin de récupérer vos modifications sur l'autre dossier, allez dans celui-ci et faites cette commande :

```
git pull
```

Cette fonction permet de mettre à jour notre branche par rapport aux dernières modifications faites sur le dépôt distant (notre Github).
Pour votre culture, un git pull est en réalité un git fetch suivi d'un git merge (notion que nous verrons plus tard).
Vous connaissez à présent les premières lignes de commande pour travailler en équipe avec Git, bien que si cela peut sembler très simple, nous allons avancer dans la complexité car Git n'est pas si simple à maîtriser.



1

## Astuce avancée

Dans les deux dossiers que vous venez de créer pour Git, vous pouvez voir un nouveau dossier caché .git. Dans celui-ci, vous trouverez un fichier config qui contient la configuration de votre dépôt. Vous pouvez sans soucis le modifier directement si besoin. Lorsque vous gérez plusieurs dépôts et plusieurs comptes, cette astuce vous sera d'un grand secours.

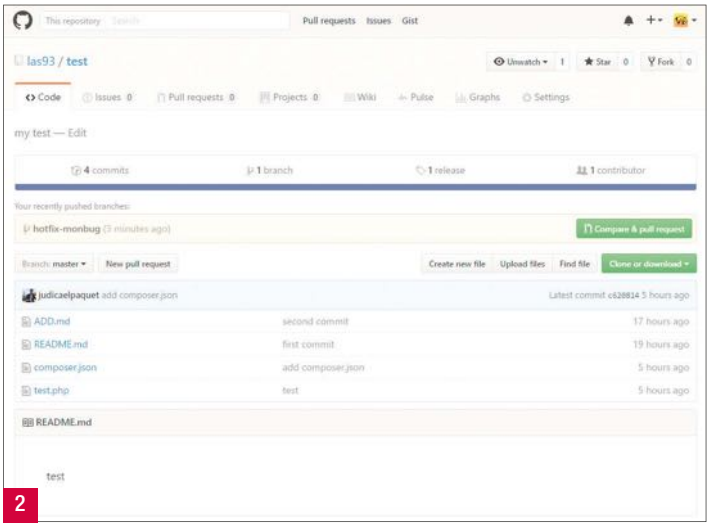
# Travail en collaboration

## Github Flow

Nous avons vu les premières commandes pour travailler sur les dépôts, mais travailler en équipe impose encore plus de rigueur et de connaissances. Il existe plusieurs modèles de travail collaboratif que l'on appelle des Git workflow comme Git flow, Gitlab Flow et Github flow ; nous allons voir ici comment travailler avec le modèle Github Flow ; je vous laisserai aller voir les autres modèles par vous-même qui sont également très intéressants. [Fig.1].
Comme vous pouvez le constater sur ce modèle, travailler sur le modèle GitHub Flow impose une bonne maîtrise des branches, mais il est le plus accessible des 3 workflows les plus populaires.
Tout ce qui est dans master doit pouvoir être déployé en production ; vous en déduirez qu'il ne faudra plus jamais travailler en direct sur la branche master.
Pour tout hotfix ou nouvelle feature, il vous faudra créer une nouvelle branche à partir du master en essayant de garder des noms cohérents à mes branches. Il est généralement recommandé d'écrire le nom sous la forme feature-[titre] ou hotfix-[titre].

```
git checkout -b hotfix-monbug
```

Cette méthode est un raccourci pour faire un git branch hotfix-monbug puis git checkout hotfix-monbug.
Je fais une modification qui consiste à corriger un bug (pour tester ce tutoriel, faites un simple ajout de fichier) et faites un commit/push pour que votre modification soit envoyée sur votre Github.
L'étape suivante est de créer un pull request qui permet de lancer une revue de code et une discussion sur celle-ci. [Fig.2].
Nous allons sur Github et nous proposons notre branche en « pull re-



2

quest » qui est vraiment un des points forts du travail collaboratif. Dès que celui-ci est déclenché, l'ensemble des collaborateurs de ce projet est mis à contribution pour discuter du « pull request ». [Fig.3].
Les autres membres peuvent faire un « add review » et approuver la modification ou lancer une discussion sur le pull request.
Les collaborateurs ayant des droits avancés pourront faire un « merge pull request » qui va merger les modifications sur le master pour indiquer que le code peut partir en production.

### Ajout de collaborateurs

En allant dans l'onglet « settings » de mon dépôt test, puis le menu « Collaborators », je peux rajouter tous les collaborateurs que je désire.

### Interdire les push sur le master

L'intérêt de faire du pull request, c'est de bloquer la branche master pour la sécuriser. Seul le pull request pourra apporter des modifications au master. Pour le faire, il suffit d'aller dans l'onglet « settings » de mon dépôt test puis le menu « branches » ; je définis le master comme « protected branches ». Et voilà, vous maîtrisez maintenant le Github flow très utilisé dans le monde communautaire.

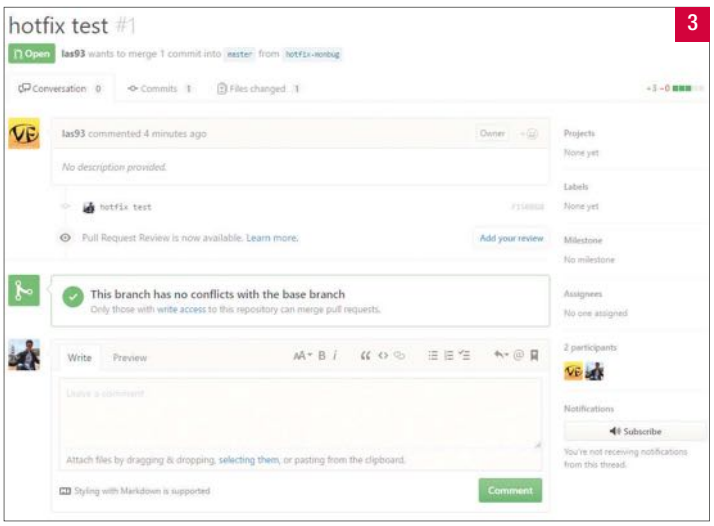
### Quelques options supplémentaires

Dans la partie « settings » et le menu « options », vous pouvez également définir quelques options supplémentaires comme : utilisation du wiki (la restriction de modification au public), utilisation des issues et les droits sur les merge. Vous pourrez également créer à partir de ce menu, un site complet pour accompagner votre projet qui s'appelle Github Page que nous verrons plus en détail à la suite de ce dossier complet.

## Un peu plus de Git

### Gestion d'un conflit

Le grand intérêt des logiciels de gestion de versions est de permettre à plusieurs personnes de travailler en même temps sur les mêmes fichiers.



3

Si deux personnes travaillent sur un même fichier mais sur des lignes différentes, Git sera assez intelligent pour rassembler le travail des deux. Vous pouvez lui faire confiance, il fera ce qu'il faut.
Par contre, si les deux personnes travaillent sur les mêmes lignes, Github vous indiquera des « conflits » qu'il faudra gérer manuellement.
Voici un exemple de fichier où il y a un conflit à gérer :

```
<<<<<<< HEAD
cc
test re
=======
test fd
dd
>>>>>>> ff78cfd213a8fe3ee91ec37441f0117e2ad082e1
```

Vous aurez à choisir le code à garder dans ce cas et de sauvegarder votre fichier. Quand cela est fait, il vous suffira de faire ceci pour corriger le conflit et envoyer le résultat au dépôt GitHub :

```
git add monfichier.md
git commit -i * -m "conflict resolved"
git push
```

Vous connaissez maintenant toutes les ficelles pour travailler en équipe ou avec une communauté GitHub même en cas de conflit sur un fichier.

### Connaître l'état de notre dépôt

Pour connaître l'état de votre dépôt local, vous pourrez utiliser cette méthode Git très utile :

```
git status
```

En cas de modification, voici ce que vous retournera Git :

```
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   src/MonBundle/Controller/MonController.php

no changes added to commit (use "git add" and/or "git commit -a")
```

Du coup, vous pourrez utiliser une ligne de commande qui vous permettra d'avoir plus de précisions sur la différence indiquée par le git status :

```
git diff
```

Vous pourrez également spécifier le fichier en ajoutant le nom du fichier sur lequel vous voulez faire le diff. •

*Suite le mois prochain*

# Virutal Game Jam 2016 : succès et beaux projets

*Du 4 au 6 novembre se tenait la Virtual Game Jam, compétition organisée par la Virtual Association. L'objectif est simple : créer un jeu en réalité virtuelle en 40h. Peu de repos, beaucoup de codes et de bugs de dernière minute. Les équipes étaient plus que motivées. Certaines étaient déjà familières de la VR, mais pas forcément avec le casque Vive, d'autres ne connaissaient pas.*


Il a fallu créer le jeu de bout en bout : design, animation, musique, codage, etc. Unity a été le plus utilisé par les équipes, une équipe a misé sur le moteur Unreal, pas forcément très connu en VR, mais l'outil est plutôt stable et on peut prototyper très rapidement un jeu.

La VR et le casque Vive permettent des usages et des interactions différentes. D'ailleurs, les équipes n'ont pas utilisé de la même manière les manettes. Vous avez des gameplay où le corps est au centre du jeu (évitement, saut, etc.), dans d'autres, vous jouez plutôt avec les manettes ; les bras et le corps ne sont qu'un élément relativement mineur. Le Vive est plutôt stable même si des sauts de vision se produisent de temps en temps ou quand tout se bloque. Mais l'expérience est vraiment immersive, et, avec une bonne audience sonore, c'est réellement top. Il faut tout de même faire attention aux câbles et à l'espace ambiance même si nous sommes censés être dans une zone délimitée visuellement.

## Les vainqueurs sont :

**Prix du public :** No Coffee Yet de Western Pony Nodes, un jeu où les mugs sont importants.

**Prix du jury :** Déjà Simulator 2017 VR de Midipil, un jeu constitué de plusieurs mini-jeux frustrants (assez drôle à tester).

**Prix de l'achievement :** The Room is on Fire de Green Team. Vous êtes seul, ou presque, dans une salle en feu.

D'autres équipes auraient mérité d'être vainqueurs, mais il faut choisir. Nous avons beaucoup aimé le jeu de Saloon Bottles Battle, le manque de temps et les bugs n'ont pas permis à l'équipe de tout développer. No Coffee Yet nous a beaucoup plu pour l'ambiance, les mugs, l'ambiance et le choix d'Unreal comme moteur de VR. Le jeu Jean-Brice rising était assez drôle et bien dans l'ambiance VR. La mission : ne pas laisser tomber des oeufs et éviter les obstacles… Petite mention à Midnightwest dans lequel vous êtes le gameplay en évitant les pièges. Là, il faut un peu d'espace…

## TENDANCES PRIX

- Sony Playstation VR : 399,99 €
- Oculus Rift : 699 €
- HTC Vive : 949 €
- Microsoft Hololens Development Edition : 3 299 €

Ces tarifs peuvent fluctuer d'un magasin à un autre, mais ce sont les tarifs généralement pratiqués. On peut s'attendre à quelques réductions pour les fêtes de fin d'année. Bien entendu, vérifiez bien les capacités matérielles de votre PC.
Tarifs sans accessoires.

## Unreal Engine, l'autre moteur pour la VR

Il faut avouer que nous connaissions surtout le développement sur Unity, l'éditeur étant très actif. Durant la VGJ2016, j'ai pu voir en action Unreal Engine et le support natif du HTC Vive via le SteamVR. L'éditeur Blueprint est vraiment intéressant pour prototyper rapidement. Par contre, plus le projet est complexe, plus le workflow devient complexe. Mais cela évite de tout coder. Vous allez payer des royalties : 5 % des revenus générés au niveau mondial (et pour chaque production). Bien entendu, vous avez des exclusions et des quotas. Par exemple, vous ne payez pas pour les 5 premiers millions générés sur l'Oculus Store. Le modèle Unity est différent. Vous ne payez pas de royalties, mais vous souscrivez à un plan contenant des quotas de revenus et d'utilisateurs. •

## Windows 10 Creators Update : des casques VR à 299 $

Durant sa grande conférence Windows 10 d'octobre dernier, Microsoft a annoncé sa volonté de proposer rapidement des matériels de réalité virtuelle à un prix accessible pour le plus grand nombre. Pour le moment peu de détails précis, mais des annonces devraient être faites en décembre. Ces casques seront compatibles avec Windows Holographic et plusieurs partenaires ont d'ores et déjà été annoncés : HP, Lenovo, Dell, Asus, etc. Ces casques seront les compagnons parfaits à la prochaine version de Windows 10 (= Windows 10 Creators Update) qui misent beaucoup sur la création, la 3D, la réalité virtuelle. Cette version devrait sortir en mars – avril 2017. Ces casques pourraient être vendus 299 $ (tarif encore incertain). Il faudrait aussi voir les caractéristiques matérielles et les capacités réelles. Visiblement, ils ne seront pas autonomes comme le casque Hololens. Des câbles seront connectés au PC. Il s'agit d'une initiative forte, car Microsoft ne veut pas laisser les concurrents s'installer et prospérer. Hololens n'est pas destiné au grand public, dans sa version actuelle, et les casques HTC, Sony, Oculus, Samsung, etc. sont chers, généralement à partir de 400 €. Les Cardboards sont bien moins chers, mais les possibilités technologiques et l'expérience ne sont pas comparables. •

# L'équipe **Orange Agynnar** raconte son Virtual Game Jam

*40h non stop de création, de debug, de développement ! Le Virtual Game Jam met au défi de faire un jeu en réalité virtuelle dans un temps limité. Huit équipes ont relevé le défi, avec brio ! Jérémie Rabusseau, chef de l'équipe Agynnar et développeur, nous raconte cette aventure de l'intérieur.*


*Jérémie – Corentin – Stéphane – Roméo - Photo réalisée par 803Z*

## « Déjà ? »

C'est avec ce simple mot suivi de ce point d'interrogation que l'équipe Orange qui répond au nom d'***Agynnar*** a débattu 4 heures pour définir un concept qui corresponde au thème, réfléchir à un concept original tout en répondant aux limites de la VR (Virtual Reality).

## L'équipe, le concept & le jeu

Agynnar, composée de 5 membres :

- Roméo → Graphiste
- Aurélien → Sound Designer
- Stéphane → Game Designer
- Corentin → Développeur
- Jérémie (moi-même) → Chef d'équipe & Développeur

Le défaut de notre équipe était le fait de ne pas avoir d'animateur 3D, c'est grâce à un outil libre de droits sur Internet que nous avons pu remplacer ce corps de métier : Mixamo. Stéphane est un Game Designer, son expérience et ses folles idées nous ont permis de débattre et détourner sa première idée : un duel de cow boy. Après un long moment de recherche au sein de l'équipe, nous avons fini par trouver notre concept : *« On incarne un barman dans un saloon. A un moment donné, une voix off intervient et annonce la fermeture de l'établissement. Les clients surpris par cette nouvelle, manifestent et deviennent même agressifs puis se mettent à lancer des bouteilles pour éliminer le barman. L'objectif ? Se défendre avec tout ce qui nous tombe sous la main (bouteilles, boîte de métal, poêle ? ) et mettre à terre les casseurs. »*

## Pourquoi « Déjà ? » ?

Nous avons fait le choix de ne pas utiliser ce thème comme un élément temporel car c'était notre toute première idée. Nous voulions une expérience unique et originale donc nous nous sommes détachés instantanément des concepts basés sur le temps de jeu.

Dans notre jeu, le ***« Déjà ? »*** est un élément déclencheur. C'est ce mot qui va lancer l'interaction entre le joueur et l'environnement. Fort heureusement, nous sommes les seuls à avoir utilisé cette technique et c'est grâce à ça que nous nous sommes démarqués des autres groupes.

## Déroulement de l'événement

Par habitude, nous avons décidé de développer notre jeu sous Unity et C#.

La première nuit, nous avons défini les **MUST TO HAVE** & **NICE TO HAVE** puis réparti les tâches pour avoir un jeu fonctionnel et voir par la suite pour intégrer quelques améliorations pour rendre le jeu plus *sexy*.

Toute la journée du Samedi a été consacrée pour ma part à l'intégration des modèles 3D suivi des animations ainsi que la partie consistant à lancer des projectiles des ennemis vers le joueur (barman que l'on incarne). Pendant ce temps, notre graphiste a entièrement créé un saloon de qualité avec le logiciel Maya.


Aurélien, notre sound designer (qui a réalisé la totalité des sons de **5** projets différents en plus du nôtre au cours des 40h) nous a fait deux musiques d'ambiance correspondant au thème du saloon ainsi que tous les bruitages qui meublent totalement le jeu et donnent le côté immersif que l'on attendait. Respect mec o/

Pendant la période de réalisation, le Game Designer est un peu mis de côté car sa seule tâche est de bien vérifier que nous sommes dans les temps et que l'équipe ne fasse pas un hors-contexte. Stéphane a su être malin et optimiser son temps en réalisant astucieusement des modèles de bouteilles à notre effigie, puis à celle

de ses confrères Game Designers également présents lors de cet événement. C'est grâce à ce petit plus que notre jeu a un côté un peu plus élaboré, c'est ce genre de détail qui fait que le jeu semble plus mature et recherché. Alors pour ce magnifique geste et cette créativité : merci Stéphane.

*Photo réalisée par* ***803Z***

***Mais dis donc, que faisait le petit Corentin pendant tout ce temps ?!***


Il s'est occupé des interactions entre les objets, les collisions, les trajectoires selon la prise de l'objet puis la puissance que le joueur veut transmettre au projectile. Pour faire simple, il s'est occupé entièrement du moteur du jeu ainsi que de son fonctionnement. Sans cette partie, le jeu aurait été vide de sens et complètement inutilisable.

En milieu de journée, nous avons commencé les choses sérieuses : les lancers d'objets et la gestion de la collision, que ce soit de l'ennemi vers le joueur ou du joueur vers l'ennemi. C'est au bout de quelques heures et grâce à l'aide d'autres participants (et membres du STAFF : BIGUP Damien et Cécilia) que nous avons pu jouer avec nos bouteilles dans notre saloon.

Durant la 2e nuit, nous avons défini le nom : **Bottles' Battle**. Simple, explicite, efficace, rigolo à énoncer… Ce nom est parfait !

Le dimanche a été consacré uniquement à la beauté du jeu, c'est-à-dire, intégrer toutes les textures aux différents éléments qui constituent notre terrain de jeu afin d'avoir des couleurs et non pas une scène complètement grise.

Pour ma part, j'ai accumulé un taux de 9h de sommeil pour 40h de jam (il faut penser à ajouter le temps de route : 7h aller-retour + la journée de travail le vendredi pour comprendre l'état de fatigue…).

Hormis cela, pour notre premier hackathon, l'organisation que nous avons établie était plutôt satisfaisante, personne n'était en attente en regardant les autres bosser et nous avons réussi à établir notre jeu dans les temps et le rendre à 12h00 précisément (c'était chaud, on a couru pour le rendre à temps).

## « Quel est l'objet de ta motivation pour venir ici ? »

C'est une question qui est revenue plusieurs fois et à laquelle Corentin et moi aimons répondre avec un grand sourire : « on est passionnés de jeux vidéo ainsi que du développement de logiciels. C'est l'occasion d'apprendre, de se challenger, de découvrir de nouvelles expériences, d'apprendre à travailler en équipe et à s'organiser rapidement avec des inconnus rencontrés 1 heure avant le lancement du projet. »

L'échange, le partage, l'amusement et la bonne humeur. Ce sont des motifs pour lesquels nous sommes prêts à arpenter les routes et venir relever le défi.

*A tous les jammers de l'édition 2016 : on se dit à l'année prochaine !*

## Que retiens-tu de cette expérience ?

Corentin et moi sommes venus uniquement pour une chose : nous amuser, faire un jeu sympa et nous améliorer dans les technologies qui nous étaient obscures (Unity / HTC Vive). Peu importe la distance que nous devrons parcourir, nous sommes prêts à nous investir à fond pour arpenter les routes et relever les challenges que nous offrent les hackathons.

Durant cet événement, nous avons appris que tous les corps de métiers sont nécessaires et nous avons appris leur fonction ainsi que leur importance. **L'expérience** et le **partage** sont les maîtres-mots que je retiens de ce weekend. C'est assez beau de voir des étudiants, des personnes qui travaillent dans le domaine de la VR qui viennent vous aider sans vous prendre de haut. La plupart des gens que j'ai pu rencontrer sont des passionnés ou simplement des personnes curieuses prêtes à échanger, vous aider, même si ça doit leur prendre du temps sur leur réalisation alors que le temps est précieux. Pour rappel, 40h pour l'élaboration d'un jeu est très court surtout lorsqu'il faut penser à manger, dormir, faire des pauses, recentrer les équipes, discuter sur le projet, s'organiser…

Être à la tête de l'équipe Agynnar, fut une très bonne expérience. J'ai pour habitude de me mettre en bonnes conditions pour travailler, donc j'ai fait de même pour mon équipe :

- Temps de sommeil imposé 5h minimum ;
- Aller manger avec les potes pour se détendre ;
- Faire des pauses et surtout aller boire dès que la soif se fait ressentir ;
- Grignoter si une petite faim est gênante à la concentration ;
- Isolation du bruit pour la concentration ;
- Imposer un échange pour faire une mise au point sur l'avancement du jeu.

Si vous ne respectez pas les temps de sommeil, voici ce qu'il vous attend.

Ce sont tous ces détails qui nous ont permis de passer un super moment. Nous étions 5, nous avons appris à nous connaître rapidement et avons coopéré instantanément en partageant nos points de vus efficacement. C'était vraiment superbe vu d'un œil extérieur.

Si vous avez l'occasion de faire un hackathon et que vous n'avez pas peur de faire de nouvelles rencontres, allez-y seul ou à deux et faites un mélange de personnes pour harmoniser vos compétences puis améliorer votre expérience.

## Résultat des courses ?

Nous avons réalisé 4 défis sur 10 possibles et l'équipe gagnante du prix de l'audace avait 5 défis. Concernant la performance, notre lancer de bouteille n'était pas assez intuitif, puis quelques bugs lors de la réception de la bouteille au vol ont un peu gêné le jury. Ces deux facteurs ont fait que nous avons obtenu une **mention spéciale** du jury à défaut d'avoir été sélectionnés pour un de ces prix. •

# L'enseignement de la **programmation** arrive dans les programmes scolaires

*Cette année scolaire est un peu particulière car le code fait enfin son entrée officielle à l'école.*

• Nicolas Decoster
*Co-fondateur de* **La Compagnie du Code** *et animateur*
*Informaticien scientifique chez Magellium*
@nnodot

Pourtant, et cela peut paraître surprenant, cela fait très longtemps que l'apprentissage du code a mis le pied à l'école. Par contre il n'y a que le pied qui est entré, pas plus… Et il en est même un peu ressorti par moments. En effet, beaucoup se souviennent des cours de LOGO (vous vous souvenez de la petite tortue à déplacer ?) ou des ateliers MO5 et TO7 de l'option informatique du bac (fin des années 80). Et depuis cette époque il n'y a pas eu grand-chose de plus, ou, du moins, rien de bien conséquent ; jusqu'à cette année, donc, qui voit une entrée franche de la programmation informatique dans les programmes de l'éducation nationale ! Il y a des notions de programmation en primaire et un peu plus au collège avec des concepts plus avancés qui apparaissent dans les programmes de mathématiques et de technologie. Par contre, il n'y a toujours pas de CAPES ou d'AGREG informatique qui garantirait un enseignement de qualité (bien que, autre nouveauté de cette année, il y ait maintenant une option informatique au CAPES de mathématiques), mais au moins cette présence officielle dans les programmes montre qu'il est jugé important que tous les enfants soient confrontés à la pensée informatique.


La question essentielle à se poser maintenant est : « quels moyens se donne l'éducation nationale pour permettre cet enseignement ? » Donc, il n'y a pas de CAPES ni d'AGREG pour l'instant (même si des acteurs reconnus comme la Société Française d'Informatique poussent pour qu'ils soient créés). Il faut donc former le corps enseignant actuel à ces nouvelles disciplines. Des professeurs de mathématiques et de technologies passionnés se sont déjà penchés sur la question et certains expérimentent de belles choses depuis quelques temps. Il y a une dynamique très positive. Mais cela ne fait pas tout. Il y a énormément de monde à former et cela va quand même être compliqué pour beaucoup d'enseignants de s'approprier cette nouvelle discipline si éloignée de leur expertise. En effet, l'informatique et les mathématiques sont deux disciplines complètement différentes, et, même s'il y a des ponts, un informaticien n'est pas forcément un mathématicien, et vice versa. Même chose pour les professeurs de technologie. Sans parler de la formation continue qui en est à ses balbutiements : les plans académiques de formation commencent à proposer quelques stages sur le sujet, des projets comme Class'Code (voir encadré) ou des ressources comme le livre "1, 2, 3… codez !" permettent aux enseignants qui en ont la volonté de se prendre en main pour s'auto-former. Certaines structures comme la Compagnie du Code proposent des accompagnements pour des établissements qui en ont la volonté. Mais tout cela n'est clairement pas suffisant pour un réel démarrage généralisé d'un enseignement de la programmation de qualité dans tous les établissements dès cette année. Espérons juste que les années de tâtonnements qui nous attendent ne dureront pas trop longtemps et que la montée en charge ne sera pas trop douloureuse pour un corps enseignant déjà bien malmené par ailleurs. Surtout, ajoutons qu'il est essentiel que cette tentative enfin un peu sérieuse d'introduction de la programmation à l'école ne retombe pas comme les précédentes. Mais soyons optimistes : beaucoup de voyants sont au vert et, même si ça pourrait être beaucoup mieux, il y a beaucoup de bonnes volontés qui œuvrent pour le succès de cette initiative. •

## CLASS'CODE : Un projet dans lequel les développeurs peuvent aider les professionnel-le-s de l'éducation à initier nos enfants à la pensée informatique

Le projet Class'Code est un projet financé par le plan d'investissement d'avenir pour la mise en place d'une formation gratuite hybride à destination de tout pédagogue (enseignant ou animateur) non informaticien qui veut transmettre la pensée informatique et la programmation à des enfants. Hybride, parce qu'elle est composée d'une formation en ligne, et de temps de rencontre où les participants se retrouvent pour échanger ensemble avec l'aide de facilitateurs informaticiens qui, eux, ne sont pas pédagogues. Toute personne connaissant bien la programmation peut donc devenir facilitateur avec un minimum d'investissement : il faut juste se familiariser avec la formation (quelques heures), et, bien sûr, être présent aux temps de rencontre. D'après le site du projet : "Vous êtes un(e) professionnel(le) de l'informatique et vous désirez aider des débutants, partager votre expérience, votre culture scientifique et technique ? Sans pour autant tout savoir, vous êtes prêts à offrir un peu de votre expertise en informatique ? Bienvenue à Class'Code https://classcode.fr !" La formation est composée de cinq modules qui, au travers d'une pratique concrète de la programmation et d'activités débranchées, permettra de comprendre les concepts clés de l'informatique, et les enjeux de société qui y sont liés ; ceci pour être en capacité d'animer des ateliers sur ces sujets dès le premier module. Voici les cinq modules de la formation :

- 1 Module fondamental : découvrez la programmation créative ;
- 2 Module thématique : manipulez l'information ;
- 3 Module thématique : dirigez les robots ;
- 4 Module thématique : connectez le réseau ;
- 5 Module fondamental : le processus de création de A à Z.

# **Apprendre** à programmer, pour quelle finalité ?

*Cette question peut de prime abord sembler curieuse dans une telle publication. Il est vrai que jusqu'à récemment, hormis quelques périodes relativement limitées dans l'histoire de l'informatique (années 80 par exemple), apprendre à programmer était et reste principalement du fait des "geeks" et futurs développeurs de métier, une intersection large joignant ces deux catégories.*

Youmna Ovazza
*Fondatrice, Teen-Code (Teen-Code : stages et ateliers d'initiation à la programmation pour ados www.teen-code.com)*

Aujourd'hui, alors que le monde entier se découvre un intérêt pour la programmation et pour son enseignement, sont-ce les mêmes motivations qui prévalent ? Celles des amateurs et novices reposent-elles sur les mêmes bases que celles des passionnés et professionnels ? Et sinon, comment les adresser ? Moi-même "non-geek" et initiée à la programmation sur le tard, dans un objectif d'éducation et d'information par rapport à un projet et non dans un but de reconversion professionnelle, je crois pouvoir avancer que les motivations de nous autres personnes ordinaires qui s'intéressent à la programmation ne sont pas tout à fait les mêmes que celles des experts du sujet ; et qu'il y a un grand malentendu sur le sujet, qui peut freiner l'accès à cet apprentissage, pourtant si riche, du plus grand nombre, si l'on ne prend pas davantage en considération les motivations des apprentis potentiels. Si l'on a certainement besoin des développeurs de métier et des passionnés pour transmettre leur savoir et leur enthousiasme, non, tout le monde :

- N'adore pas les robots ;
- Ne tombe pas en extase devant une imprimante 3D ;
- Ne joue pas à Minecraft, Pokémon Go ou World of Warcraft selon l'âge (ou pas :)) ;
- Ne rêve pas de faire le site de son CV en ligne en html/css !

Faut-il forcément apprendre tous les modes de cuisson de la viande pour se mettre à la cuisine ? Ne peut-on pas commencer directement par faire un gâteau, même s'il n'a pas 4 couches de crème et de génoise, un glaçage et s'il n'est pas moulé à la perfection ? Pourtant, au vu des cours et thèmes proposés pour s'initier à la programmation, il semblerait que seules les motivations "geek" prévalent : et on se demande, par exemple, pourquoi il y a toujours aussi peu de filles ? Mais aussi d'artistes, de littéraires, de sportifs, de parents, de personnes au foyer, etc. ? Pour qui en a fait l'expérience - mon propre témoignage - l'apprentissage de la programmation revêt de multiples intérêts, bien au-delà de la joie d'imprimer son porte-clé avec une imprimante 3D :

- Être enfin côté coulisses, et voir, même entrevoir, comment se fabrique une grande partie de notre monde actuel !
- Agir, au lieu de simplement consommer ;
- Comprendre par l'expérience ;
- Développer son autonomie de jugement par rapport aux nombreux enjeux du numérique ;
- Développer une plus grande efficacité dans les projets professionnels numériques ;
- Développer une meilleure collaboration avec ses collègues et partenaires développeurs ;

Ces motivations ont un champ d'application très large, en rapport avec tous les sujets et centres d'intérêt des personnes concernées ! Pourquoi ne pas partir justement de ces usages et centres d'intérêt, pour construire des cursus d'initiation et d'enseignement s'adressant au plus grand nombre, adaptés à cette logique de vulgarisation de qualité, et non une version "dégradée" d'un cursus professionnalisant avec toujours les mêmes langages et mêmes sujets ? Tout le monde a un téléphone dans sa poche, tout le monde peut être concerné un jour ou l'autre par le piratage et la cyber-sécurité, tout le monde utilise les réseaux sociaux... Tout comme la chimie appliquée à la cuisine, répondre à ces motivations requiert la rencontre de deux univers, de deux expertises : les programmeurs d'une part, les utilisateurs ou experts d'autres métiers, d'autre part.

Ce n'est qu'en joignant leurs forces et en mettant en commun leur créativité qu'on pourra véritablement adresser les motivations du plus grand nombre et intéresser de vastes populations. A l'heure où le numérique occupe une place si importante dans nos vies, ce ne serait pas du luxe que de se pencher un peu sur cette question qui touche autant à la citoyenneté qu'aux loisirs ou à l'efficacité économique ! •

## INFORMATIQUE CRÉATIVE : *bien plus que l'apprentissage de la programmation*

• Nicolas Decoster
*Co-fondateur de* **La Compagnie du Code** *et animateur, Informaticien scientifique chez Magellium,* @nnodot

A l'origine de l'informatique créative, il y a l'idée d'utiliser l'ordinateur et la programmation comme prétextes pour créer des choses. L'apprentissage de la programmation n'est plus un objectif mais un moyen. On ne va pas apprendre à quelqu'un comment utiliser les boucles mais on va lui demander de faire une animation visuelle pour se présenter. N'étant plus une finalité, les apprentissages arrivent uniquement quand on en a besoin et l'accent est donné sur l'expérimentation et la découverte avec un maximum d'autonomie. Dans le projet que je suis en train de réaliser j'aimerais que mon personnage saute quand j'appuie sur la touche espace ? J'apprends par moi-même comment agencer des instructions d'événements, de déplacement et de contrôle pour donner l'illusion de ce mouvement. C'est ma motivation à réaliser un effet que j'ai décidé qui va me donner l'envie d'aller à la recherche du moyen pour le faire. Et, comme cela vient de mon désir, je retiendrai bien mieux ce que j'ai appris. La recherche de la manière de faire, va me pousser à chercher des informations en ligne ou auprès d'autres personnes, à me poser pour prendre du recul sur le problème en question et surtout à expérimenter.

La programmation est une activité qui se prête très bien à l'expérimentation. On pourrait presque dire, qu'en fait, l'expérimentation est l'essence même de la programmation. Le cycle de base de l'activité créatrice du développeur est : j'ai une idée, je réalise quelque chose, je teste, ça ne marche pas, j'analyse, je corrige, ça marche. Puis je passe à l'idée suivante. Cela permet aux enfants d'être dans le faire, de se familiariser avec le fait que c'est normal de se tromper et surtout de réussir à construire quelque chose et de trouver cela gratifiant. La programmation permet de très vite être confronté à cela.

Et, de plus c'est une opportunité pour certains enfants en difficulté de renouer avec l'apprentissage, le faire et la réussite. •

# Où et comment **coder** en dehors de l'école ?

Il existe beaucoup de clubs d'informatique et de structures qui contribuent d'une manière ou d'une autre à la diffusion auprès des enfants et des ados, de la programmation, de l'informatique créative et de la pensée informatique. Il serait difficile de les nommer tous. Un premier point de départ est le site jecode.org qui contient une liste non exhaustive d'un grand nombre d'initiatives (http://jecode.org/initiatives). La typologie des structures est très riche : associations, centres de loisirs, entreprises, coopératives, établissements scolaires, éditeurs de logiciels et même des instituts de recherche impliqués dans la médiation scientifique comme l'Inria, qui est très actif. Il peut s'agir de structures dont la vocation est de proposer régulièrement des ateliers (comme Teen-Code ou la Compagnie du Code), ou bien des initiatives qui fournissent des ressources et qui organisent ponctuellement des événements sur ces sujets (comme Devoxx4Kids).

## TEEN-CODE

Teen-Code (http://www.teen-code.com/) initie les adolescents à partir de 13 ans à la programmation, à travers des stages de vacances et des ateliers thématiques, sur Paris. Teen-Code est une toute jeune société fondée par Youmna Ovazza, ex-Chief Digital Officer d'un groupe international de communication et spécialiste de marketing numérique, également formée à la programmation et mère de trois enfants. Teen-Code aspire à faire découvrir aux ados les coulisses de leurs usages quotidiens pour les rendre un peu plus acteurs et créateurs, et un peu moins simples consommateurs du numérique. La création d'applications mobiles sous Android est le contenu phare proposé par Teen-Code aujourd'hui, enrichi récemment de la création de jeux vidéo sous Unity, et bientôt d'autres sujets. Le parti pris est de traiter les ados comme des adultes en devenir ; les programmes sont donc conçus à partir des pratiques des professionnels et adaptés aux ados.

## LA COMPAGNIE DU CODE

La Compagnie du Code (http://www.lacompagniedu-code.org/fr/) est une coopérative (plus précisément, une SCIC, pour société coopérative d'intérêt collectif) qui a pour vocation de transmettre la pensée informatique au plus grand nombre.
Elle est basée à Toulouse et organise des ateliers d'informatique créative pour les enfants sous forme de stages de vacances ou d'ateliers hebdomadaires, en extra et en péri scolaire. Elle est impliquée dans l'accompagnement des enseignants pour la prise en compte des nouveaux programmes de l'éducation nationale sur la programmation informatique. Elle anime également des ateliers et des enseignements pour les adultes pour démystifier le numérique et pour avoir un premier contact avec ce qu'est l'activité de développeur.
La CdC est portée par des informaticiennes et des informaticiens qui ont à cœur de transmettre leur goût de la création par la programmation et de démystifier le fonctionnement du monde numérique. Elle est impliquée dans l'accompagnement de l'introduction de la programmation dans les nouveaux programmes de l'éducation nationale en participant au Plan Académique de Formation, en intervenant dans des établissements et en étant partenaire du projet national Class'Code. Elle bénéficie du support d'étudiants du numérique (en particulier, des apprenants de Simplon.co et de Digital Campus).

## DUCHESS FRANCE

Duchess France (http://www.duchess-france.org/) est une association destinée à valoriser et promouvoir les femmes développeuses, ou exerçant dans un domaine technique de l'informatique. Les actions de l'association visent à faire connaître ces métiers techniques notamment auprès des femmes et jeunes filles pour susciter de nouvelles vocations.

Duchess France publie des portraits de rôles modèles, intervient dans toute la France pour participer lors de forums et événements afin de parler du métier de développeuses à des collégiennes et lycéennes (Science Factor, Forum de la mixité, Science de l'ingénieur au féminin, HeForShe…), et organise plusieurs types d'ateliers : ateliers de prise de parole en public, ateliers mixtes de coaching "Call for Paper", ateliers techniques/de code, conférences, mais également ateliers d'apprentissage du code pour les enfants ainsi que les adultes débutants.

### Autres structures

- Les Voyageurs du Code à Paris et dans certaines villes ;
- Magic Makers dans la région parisienne ;
- Tech Kids Academy à Paris et Saint-Germain-en-Laye ;
- Combustible à Toulouse ;
- Planète Sciences à Toulouse ;
- Alsace Digitale en Alsace ;
- Coding & Bricks dans le Nord Pas-de-Calais ;
- Ch'ti code liste les initiatives dans le Nord Pas-de-Calais ;
- Club Code France à Troyes ;
- Cod Cod Coding à Vandoeuvre lès Nancy ;
- Fantasticode à Gap ;
- Jeunes-Science Bordeaux à Bordeaux ;
- Geek School à Sophia Antipolis, Nice et Monaco ;
- Les Petits Hackers à Brest ;
- Savoirs Numériques Pour Tous à Grenoble ;
- MuseoMixLIM à Limoges ;
- Coding Gouters wiki qui liste les coding goûter de France ;
- Et pour bien finir, Devoxx4Kids France qui organise des ateliers dans toute la France.

# Des **outils ludiques** pour tous les âges

*Lorsque l'on pense à la programmation on se dit que son apprentissage va être très compliqué et semé d'embûches. Grâce aux outils existants, pour les enfants et adolescents, la programmation s'avère être un jeu d'enfant !*

• Nicolas Decoster
*Co-fondateur de* **La Compagnie du Code** *et animateur*
*Informaticien scientifique chez Magellium*
@nnodot

Youmna Ovazza
*Fondatrice,*
**Teen-Code**
*(Teen-Code : stages et ateliers d'initiation à la programmation pour ados*
www.teen-code.com*)*

Aurélie Vache
*Lead Developer chez*
**atchikservices**
*Duchess France Leader*
@aurelievache

## PROGRAMMATION PAR BLOCS

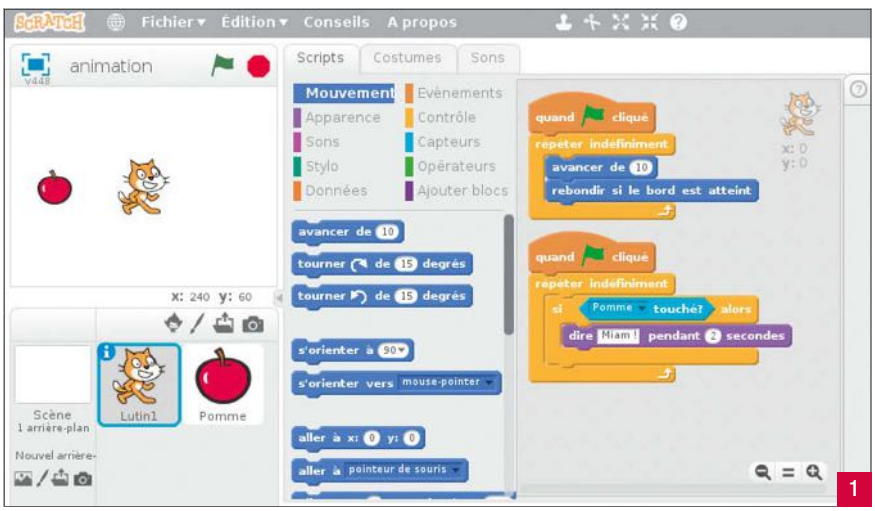
Lorsqu'on parle d'initiation à la programmation pour les enfants il est difficile de ne pas parler de Scratch. Il s'agit d'un outil de programmation visuelle par blocs qui a plus de dix ans. Il est développé par le MIT et remporte un très grand succès, auprès des petits et des grands, surtout depuis la version 2 qui est disponible en ligne sur un site communautaire qui promeut l'échange, le partage et le remix de projets (http://scratch.mit.edu). Sa syntaxe visuelle qui emboîte des blocs de textes comme des lego a inspiré de très nombreux outils similaires dont Snap! (un logiciel libre en ligne développé par Berkeley, une autre université aux USA) et Blockly (développé par Google). D'ailleurs le MIT et Google ont décidé de travailler ensemble pour réécrire Scratch, heuuu… from scratch ;-) entièrement en HTML et en JavaScript (alors que Scratch 2 est écrit en ActionScript et nécessite donc d'avoir un plugin flash pour fonctionner). Leur idée est de partir du moteur de rendu de Blockly et d'y brancher un moteur d'exécution Scratch. Le travail est en cours, affaire à suivre !

## Scratch, la référence

Mais que peut-on faire avec Scratch ? En fait, il s'agit d'un langage tellement expressif que le mieux est d'en parler à l'aide d'un exemple. La figure ci-dessous montre l'interface de Scratch avec un exemple de programme. Si Captain Obvious était là, il dirait que ce programme fait se déplacer le personnage en permanence en le faisant rebondir sur les bords et en lui faisant dire "Miam !" dès qu'il touche une pomme. Voilà, un premier programme simple à faire mais qui produit déjà une première animation. [1]

L'interface de Scratch est basique. Il y a une scène à gauche, avec la liste des lutins en dessous, au milieu il y a les blocs utilisables regroupés par catégories et la partie programmation se trouve à droite. Pour programmer on glisse tout simplement les blocs avec l'aide de la souris et on les assemble comme des Lego. Nous n'allons pas rentrer dans les détails du fonctionnement de Scratch, il y a beaucoup de ressources en ligne pour cela et même de plus en plus de livres. Indiquons juste que Scratch permet d'utiliser beaucoup de concepts de base de la programmation : les séquences d'instructions, les exécutions conditionnelles, les boucles, les variables, les événements, les interactions avec l'utilisateur, l'affichage, l'encapsulation (on peut définir de nouveaux blocs), les messages, la création dynamique d'objets. Et, au final, les notions de base essentielles qu'utilisent les développeurs sont quasiment toutes là.

Cependant lorsque l'on commence à avoir une pratique un peu poussée de Scratch on atteint certaines limites : on ne peut pas définir de blocs qui retournent quelque chose, on ne peut pas attacher des informations aux messages que l'on envoie, on ne peut pas accéder aux informations ou aux actions des autres lutins, etc. Ces limitations qui sont réellement handicapantes lorsque l'on veut réaliser un projet un peu complexe, sont pleinement assumées par l'équipe de développement de Scratch : la volonté est de garder quelque chose de simple pour que cela reste un outil très accessible dédié à la découverte de la programmation, tout ce qui est un peu complexe à appréhender a été volontairement laissé de côté.



1

## Snap! pour aller plus loin

Snap! reprend les principes et les fonctions de Scratch, et les deux outils sont très similaires d'aspect. Par contre Snap! offre plein de nouvelles possibilités qui pallient des manques, qui facilitent la vie du développeur, qui lui permettent d'explorer des concepts plus avancés ou qui lui ouvrent de nouvelles possibilités.

Au final, Scratch et Snap! sont tellement simples et amusants à utiliser que l'on pourrait penser qu'ils sont de simples jouets qui ne font qu'effleurer ce qu'est réellement la programmation. Mais cela est trompeur, au-delà de la simple découverte du code, il y a moyen de construire des choses ambitieuses et on peut aller très loin dans l'exploration de la programmation !

## App Inventor

Curieusement méconnu en France, App Inventor (AI) est certainement une des meilleures

manières de s'initier à la programmation mobile, et à la programmation tout court, pour débutants. Conçu en 2009 par une équipe conjointe de Google et du MIT (Massachussets Institute of Technology) et depuis développé par ce dernier, App Inventor est un logiciel de programmation visuelle sous Android, (comme Scratch par exemple pour les enfants), basé quant à lui sur Blockly, open-source et accessible gratuitement en ligne en plusieurs langues.

Quelques chiffres clés pour en présenter l'impact international avant de détailler ses avantages : en 2015, la communauté d'utilisateurs de AI 3 était de millions d'utilisateurs issus de 195 pays différents. Plus de 100 000 utilisateurs actifs / semaine, ont construit à ce jour plus de 7 millions d'applis Android.

## App Inventor, pour qui et pour quoi faire ?

Grâce à sa double interface de création, d'interface utilisateur d'une part, et de programmation d'autre part, AI permet de s'initier, bien au-delà de la programmation "pure", à l'ensemble de la logique qui préside à la création d'un projet : l'utilisateur est au cœur du projet, le design et l'ergonomie sont en permanence aussi importants et visibles que le code lui-même, et en cela, c'est un outil excellent pour s'initier à la programmation ou sensibiliser des profils non-techniques aux différentes dimensions qui y sont associées. [2]

App Inventor est également un outil qui permet la réalisation complète d'un projet, de zéro à l'installation d'une application fonctionnelle sur son téléphone ou sa tablette, au partage avec d'autres personnes voire à la publication sur le Google Play Store. C'est donc un outil qui n'est pas un simple bac à sable mais un véritable moyen de création d'une réalisation concrète qu'on transporte et manipule avec soi, dans sa poche, ce qui situe bien les enjeux de l'apprentissage en prise directe avec les usages quotidiens des utilisateurs.

Grâce à la grande variété des composants précodés déjà installés, AI se prête à de multiples sujets d'application, du dessin aux jeux vidéo ou à la photo en passant par la géolocalisation, la traduction, la gestion des SMS, les échanges d'information via Internet, etc. ce qui en fait un cadre flexible et ouvert aux centres d'intérêt du plus grand nombre, et notamment de tous les "non-geek" qui n'en utilisent pas moins un téléphone tous les jours !

Du cadre scolaire à la formation pour adultes, AI permet d'intégrer les bases de la logique et des bonnes pratiques de développement, en réalisant de vrais projets, sans y passer des mois : que demander de plus pour s'y mettre ?

## PROGRAMMATION EN MODE TEXTE

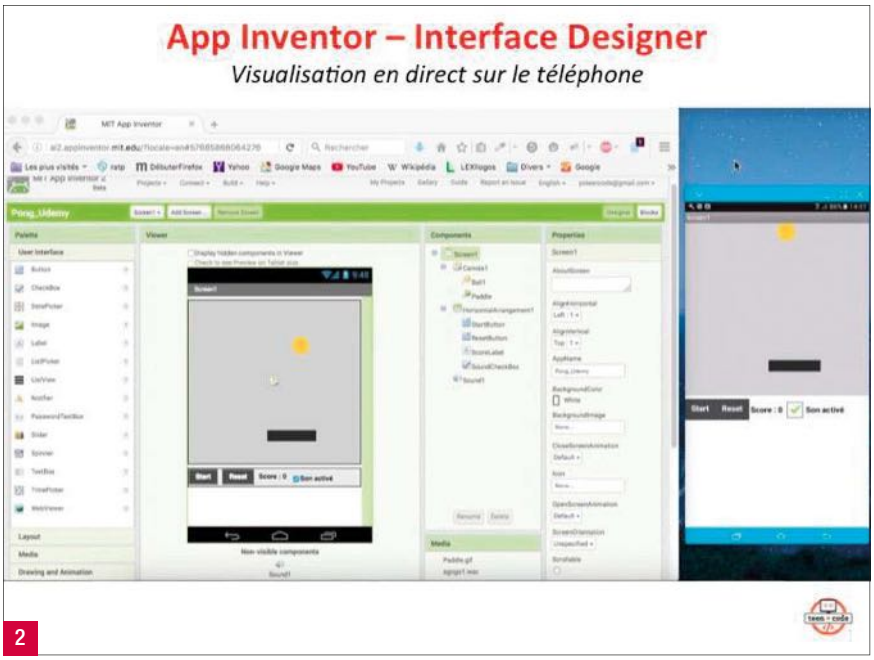
On l'a vu, les outils de programmation par bloc sont des candidats sérieux pour découvrir et approfondir la programmation et ses concepts. Cependant, actuellement le métier de développeur est essentiellement centré sur les langages de programmation en mode texte. De nos jours les logiciels, les applis, les sites Web font appel à des programmes écrits en C, C++, Java, JavaScript, Python, C#, bash, etc. L'attrait pour ces "vrais" langages de programmation par les enfants et les adolescents est donc tout naturel, ils ont envie d'entrer dans la cour des grands et de pratiquer les langages qui ont construit les logiciels qu'ils utilisent. Mais quels sont les langages textuels qui sont adaptés pour eux dans l'optique d'apprendre la programmation ? Et quels sont les avantages, les inconvénients et les spécificités par rapport à la programmation par blocs ?

En fait chaque langage a ses particularités qui fait qu'il sera plus ou moins adapté à l'apprentissage. Certains sont plus complexe, d'autres sont moins accessibles et d'autres sont des langages obligés pour un usage particulier. Passons en revue certain d'entre eux.

Tout d'abord le langage le plus disponible : JavaScript. Ce langage est présent partout, il suffit d'avoir un navigateur Web et vous pouvez faire du JavaScript. Vous ouvrez votre éditeur de texte préféré, vous mettez quelques lignes de code dans un fichier et vous l'exécutez avec votre navigateur. De plus, la plupart des navigateurs embarquent des outils de développement intégrés (console, débugger…). C'est le standard Web reconnu et beaucoup d'efforts sont fait pour disposer de moteurs performants : Google, Mozilla, Microsoft et Apple sont tous en compétition sur ce terrain. On dispose même de moteur JavaScript en dehors des navigateurs Web (le projet node.js est un moteur en ligne de commande). Cela fait que JavaScript est un langage avec lequel il faudra compter encore longtemps et sa communauté est l'une des plus bouillonnantes du moment.

Ensuite, parlons d'un vieux langage qui est le socle de beaucoup de choses : le langage C. Beaucoup de logiciels existants sont écrit en C, comme de grands pans du système Linux par exemple. C'est également un langage de prédilection pour de l'informatique embarquée car il est bas niveau et très performant. En particulier, c'est le langage qui est généralement utilisé pour programmer les cartes Arduino, et il existe tout un outillage pour s'y mettre facilement. Un avantage du langage C, qui peut être aussi un inconvénient, est qu'il est plus bas niveau que beaucoup d'autres, en particulier sur les problématiques de gestion de mémoire : c'est instructif sur le fonctionnement interne d'un programme mais c'est plus délicat à programmer. Ce langage à l'avantage, ou l'inconvénient ;-), d'imposer au développeur de se frotter aux joies de la compilation et de son lot d'erreurs.

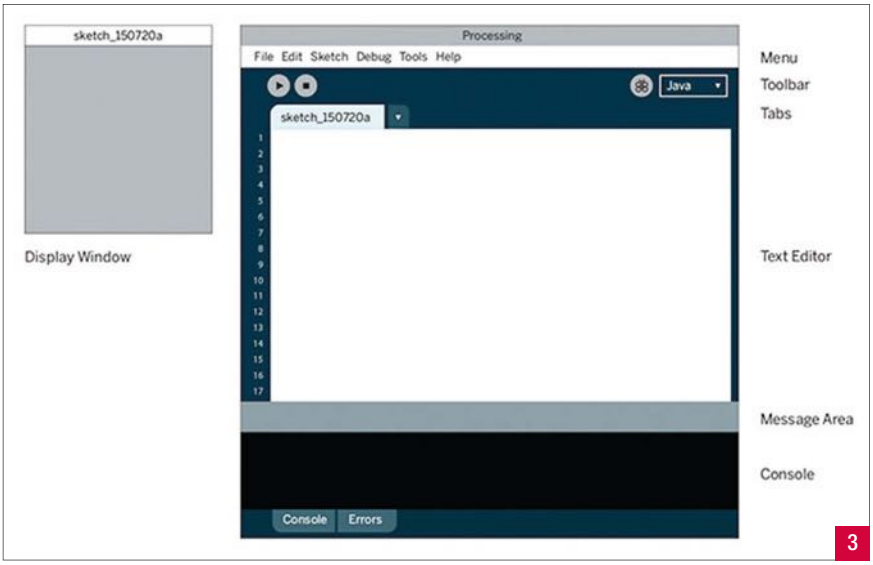
Un langage qui est un peu moins bas niveau est Python. C'est un langage interprété ce qui, en particulier, évite les problèmes de compilation. Il est très populaire et offre un écosystème très




2

riche. Quel que soit le domaine qui vous intéresse vous trouverez des modules python pour vous faciliter la vie : jeux, calcul scientifique, serveur Web, visualisation de données, communication avec du matériel, etc.

Enfin, parlons de Java, langage très utilisé en entreprise. C'est également un langage compilé (un peu comme le C mais avec quelques nuances), il est portable (comme JavaScript et Python) et dispose également d'une très grande communauté (mais peut-être avec un écosystème moins riche que pour Python). C'est également un langage très présent dans des problématiques d'actualité comme la Big Data. Java est le langage de la bibliothèque processing qui est très adapté à l'informatique créative et qui permet de découvrir la programmation de manière ludique (voir les figures ci-contre). [3]




*Interface de développement de processing*

```
void setup() {
  size(480, 120);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

[4]


*Résultat de l'exécution du code*

Rappelons également que le Java est utilisé pour programmer des applications mobiles sous Android.

Citons rapidement le C++ pour dire qu'il s'agit un langage très intéressant et très actif mais qui est plus difficile d'accès (on peut y venir après avoir appris le langage C), et la programmation *shell* (comme *bash*) qui est radicalement différente des autres mais qui présente un certain intérêt en tant que tel et qui est quasiment incontournable dès qu'on veut se plonger dans le fonctionnement des systèmes Unix (comme Linux). Il y a donc deux grandes grandes approches pour programmez et, donc, pour apprendre à programmer. Soit la programmation par bloc, soit la programmation en mode texte. Voyons les avantages et les inconvénients du code texte par rapport au code bloc.

Inconvénients :

- Le programmation textuelle est en général moins accessible, il faut souvent des outils à installer avant de pouvoir commencer à coder. Cependant, il y a de plus en plus de possibilités pour coder en ligne dans ces langages ;
- Souvent le résultat en mode texte est accessible moins rapidement qu'en mode bloc. Par exemple, avec Scratch ou Snap! on clique sur un bloc "avancer de 10" et le personnage avance tout de suite. Dans certains langages textuels il faut écrire le code, sauvegarder le fichier, éventuellement compiler, exécuter pour enfin voir le résultat ;
- Dans le même genre d'idée, en général il n'y a pas d'interactivité ou de *live coding*. Dans Scratch ou Snap!, pendant que le programme tourne on peut changer une valeur ou ajouter un bloc et l'exécution prend en compte la modification instantanément. Cela ouvre des possibilités d'explorations et d'expérimentations plus importantes. La plupart des langages textuels ne permettent pas cela (bien que cela commence à arriver, en particulier avec les développement Web en JavaScript) ;
- Pour développer en mode texte, même si ce n'est pas indispensable, c'est souvent mieux d'avoir un bon outillage et il peut parfois être complexe à mettre en place : il faut trouver le bon éditeur, le bon débogueur, le bon *linter*, etc. Mais on peut trouver quand même des choses bien intégrées, voire en ligne ;
- Avec un langage textuel on est confronté aux erreurs de syntaxe qui par définition ne sont pas présentes avec les blocs. Cependant un bon *linter* permet de limiter les dégâts ;
- Il en est de même avec les erreurs de compilation, c'est un frein supplémentaire, même si un bon outillage permet d'aider le développeur dans la recherche des problèmes.

Avantages :

- Comme nous l'avons dit, actuellement l'essentiel de la programmation se fait en mode texte. S'y mettre c'est se former à l'existant ;
- Un simple éditeur de texte est suffisant pour créer quelque chose, pas besoin d'une interface graphique ;
- Les langages textuels peuvent être plus puissants dans ce qu'ils permettent de faire ;
- Ils peuvent être plus bas niveau et donc offrir un contrôle plus fin de certaines choses (comme la gestion de la mémoire par exemple) ;
- Ils permettent de développer des choses complexes d'une certaine ampleur.

## POUR LES PLUS PETITS

On a vu qu'il y avait des outils pour les adolescents, pour les enfants mais qu'en est-il pour les plus petits, peuvent-ils également être initiés au merveilleux monde de la programmation ?

L'initiation de la programmation n'est pas réservée qu'aux grands, les petits sont également gâtés. Avez-vous déjà entendu parler de Cubetto ou de Thymio, non ? Nous allons vous guider vers ces fabuleux outils.

### *Cubetto (à partir de 3 ans)* [5]

Cubetto, sous ce nom se cache un mignon petit robot en forme de cube. Une campagne Kickstarter a été lancé cette année et a été plus que financée, plus de 1 596 457 $

sur les 100 000 $ initiaux ont été réunis par 6 553 contributeurs ! Lorsque l'on sait que cette campagne de crowd-founding a été soutenue

par Randi Zuckerberg (sœur aînée de Mark Zuckerberg et entrepreneuse américaine) et Massimo Banzi (le co-fondateur d'Arduino), on se dit que ce succès est normal et mérité.
Le but de ce robot est d'apprendre les bases de la programmation à des enfants (filles et garçons) de trois ans et plus.

## Comment ?

Le kit est composé d'un tapis de jeu, d'un livre d'instructions, d'une console de programmation et de 16 blocs d'instructions. Les blocs d'instructions permettront au robot d'avancer, de reculer, de tourner à gauche, de tourner à droite ou d'exécuter une fonction.
Si l'enfant désire que le robot aille d'un point A à un point B, il va devoir concevoir le programme en disposant les blocs d'instructions sur la console de programmation. Lorsque le bouton de démarrage est pressé, le robot se met en marche et suit les instructions sur le plateau de jeu. Un robot que l'on peut piloter et programmer en le touchant, sans être devant un écran, quelle façon ludique et sympathique d'initier les enfants ! :-) Lorsque l'on voit le petit cube en bois, on tombe immédiatement sous son charme, par contre ce coup de cœur aura un coût, il est disponible sur le site https://www.primotoys.com/ au prix de 225$, et 245$ pour la version avec 5 cartes et 5 storybooks.

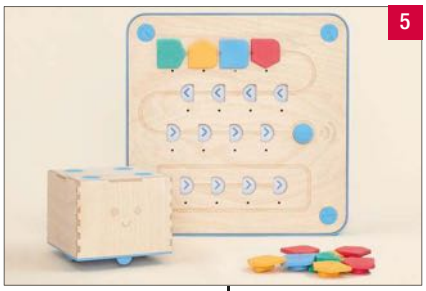
6

### *Thymio (de 4 à 15 ans)* [6]

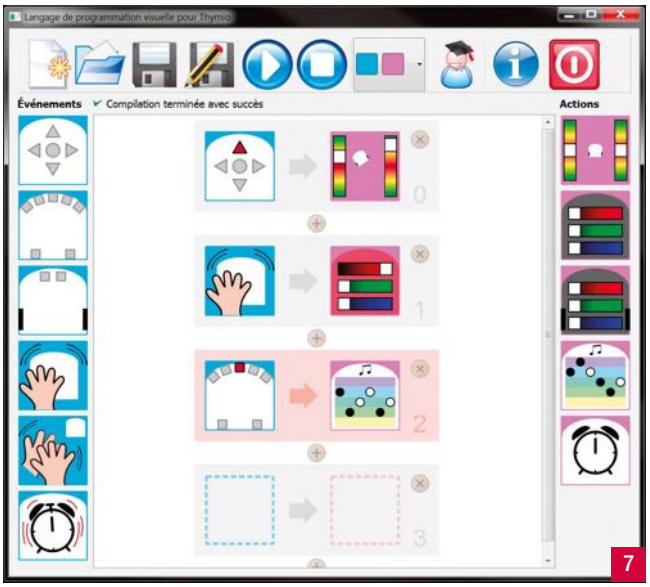
Thymio est un petit robot, d'origine suisse, qui permet de découvrir l'univers de la programmation, de la robotique et d'apprendre le langage des robots. Le robot est programmé avec six comportements : amical, explorateur, craintif, explorateur, obéissant et attentif. Grâce à ces comportements préprogrammés, l'enfant démarre son robot, appuie sur le bouton central permettant de sélectionner une couleur pour démarrer le comportement, il peut ainsi voir son petit robot rouler, bouger, changer de couleur, se comporter comme demandé.
Ensuite l'enfant peut passer à la seconde étape, programmer son robot !
Ce qu'il y a de super sympa avec Thymio c'est que l'apprentissage de la programmation et de la robotique avec ce robot s'adapte à l'âge et au niveau de l'enfant. En effet, plusieurs modes de programmations sont possibles : programmation visuelle, programmation blocky, programmation scratch et programmation texte.

### *Langage de Programmation Visuelle (VPL) pour Thymio* [7]

Quoi de mieux que découvrir la programmation grâce à la programmation visuelle ? L'enfant n'a pas besoin de savoir lire, en glissant les pictos/les blocs d'images au centre de l'écran,


5

*"A playful programming language you can touch. Montessori approved, and LOGO Turtle inspired. Learn programming away from the screen."*



7

l'enfant peut demander au robot d'exécuter des instructions. Il suffit d'appuyer sur Play (bouton avec un triangle) et le tour est joué. Programmer un Thymio avec cette interface c'est vraiment un jeu d'enfant.

### *Blockly* [8]

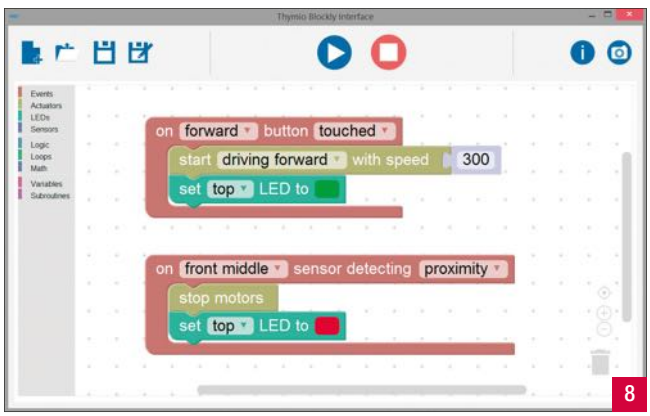
On connaît plus Scratch, au moins de nom, mais un autre langage de programmation par blocks existe. Développé par Google, *Blockly* (https://developers.google.com/blockly/), une bibliothèque logicielle JavaScript permettant de créer des environnements de développement utilisant un langage graphique, est le pont idéal entre la programmation visuelle et textuelle. En déplaçant les blocs d'instructions l'enfant/l'adolescent va pouvoir demander au robot ce qu'il souhaite qu'il fasse, et va ainsi s'initier à la programmation : aux boucles, aux conditions…
Avec Blockly l'enfant pourra utiliser 100% des capacités du robot sans écrire une seule ligne de code tout en apprenant la logique de l'informatique et de la robotique.

### *Scratch*

Si vous ou votre enfant avez plus l'habitude de Scratch, il est également possible de programmer son robot Thymio en Scratch grâce à Asebascratch, une connexion logicielle entre Scratch 2 et le Thymio, qui permet aux programmes Scratch et à ses lutins d'interagir avec le robot.

### *Programmation texte avec Aseba Studio*

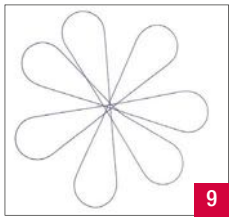
En démarrant l'IDE on entre dans le vif du sujet, dans la programmation à part entière. Grâce à la programmation textuelle l'adolescent peut accéder à tous les capteurs du robot, vérifier l'état du thymio, créer des graphiques en temps réel, contrôler les LEDs, les moteurs, donner des instructions de plus en plus complexes au robot…
Si l'on veut par exemple faire dessiner notre Thymio, il suffit de mettre un stylo ou feutre dans le trou et d'exécuter le code suivant :



8

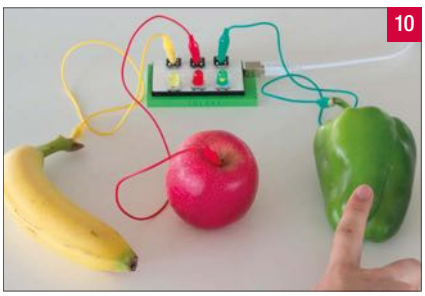
```
var itera = 0
timer.period[0] = 1000 # 1000ms = 1s

onevent timer0
  itera = itera + 1
  if itera==1 then
    motor.left.target = 230
    motor.right.target = -120
  end
  if itera==4 then
    motor.left.target = 80
    motor.right.target = 80
  end
  if itera==7 then
    itera = 0
  end
```

Le dessin qu'il en résultera sera le suivant : [9]
Vous pouvez essayer de modifier les valeurs des variables **motor.left.target** et **motor.right.target** et tester afin de voir quel sera le rendu.


9

Autre cas d'utilisation, on met le robot sur une feuille blanche comprenant un chemin en noir, si l'on souhaite que le Thymio suive le tracé noir, vous pourrez utiliser le code suivant :

```
onevent startup
  mini=1023
  maxi=0
  mean=512
  vari=512
  ireg=0

  speed=100 #or another value

onevent prox
  p1=prox.ground.delta[1]
  callsub statistics
  call math.muldiv(ndev,SIGMA,p1-mean,vari)
  if abs(ndev)<SIGMA then
    preg=60*ndev/100
    ireg=ireg+10*preg/30
    motor.left.target=speed+(preg+ireg)
    motor.right.target=speed-(preg+ireg)
  else
    ireg=0
    motor.left.target=1*ndev/2
    motor.right.target=-1*ndev/2
  end

sub statistics
  call math.max(maxi,maxi,p1)
  call math.min(mini,mini,p1)
  if maxi-mini>400 then
    call math.muldiv(vari,45,maxi-mini,100)
    mean=(mini+maxi)/2
  end
```

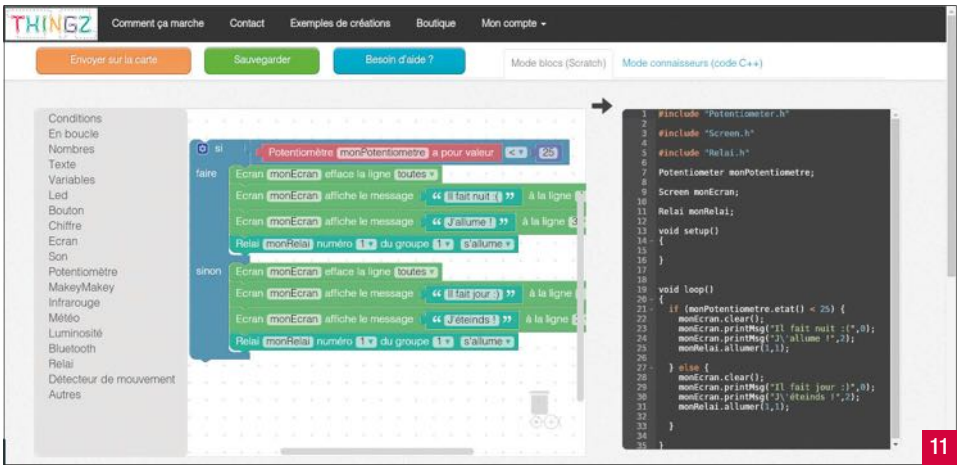

10

*Exemples de montages Thingz et de modules disponibles, dont le module Bluetooth et le module simple connecteur qui permet de s'interfacer avec des… fruits et légumes ! comme avec Makey Makey.*

Côté prix, il vous faudra débourser 179 CHF (soit environ 165€) pour la version sans-fil du Thymio, je vous invite à aller sur le site de Thymio (https://www.thymio.org/) pour découvrir les autres versions du petit robot ainsi que les nombreux packs.
Je vous avoue que le petit Thymio me tente bien car contrairement à Cubetto, même un adolescent pourra s'en amuser car il pourra programmer son robot en programmation blocks puis texte, donc cela peut être un investissement sur le long terme, et même pour nous en tant qu'adulte, développeur ou non, cela peut nous permettre de nous familiariser avec la robotique (et à l'IoT) en douceur grâce aux capteurs (de proximité, de température, de sol, microphone, accéléromètre…) que possède le robot.
Quelques classes ont testé l'initiation de la programmation à l'école grâce à ces robots, je vous invite à lire leur histoire, notamment celle dans l'académie de Strasbourg : http://tice67.site.ac-strasbourg.fr/wordpress/aspects-de-la-programmation-de-thymio-en-classe/. Je vous invite également à découvrir des exemples d'activités à faire avec un Thymio : http://www.netpublic.fr/2016/08/robots-et-code-pour-les-enfants-25-fiches-pour-mettre-en-place-des-activites-pedagogiques/
Si vous désirez des tutoriels, des exemples d'instructions à donner à votre Thymio, je vous invite à aller vers le Github de aseba-community : https://github.com/aseba-community

## THINGZ

Pour apprendre à programmer avec un pied dans le monde réel, il y a Arduino. Avec ce système, on assemble des composants électroniques à une carte et on programme en langage C le comportement du montage. C'est très bien pour se construire des systèmes complets en se frottant aux concepts bas niveau de l'électronique et de la programmation, mais cela demande un investissement non négligeable avant de pouvoir créer un objet fonctionnel. Pour ceux qui veulent accéder à l'essence de la construction et de la programmation de ce type de dispositif mais de façon plus simple et ludique, il y a **Thingz**, qui permet de créer des objets en branchant des briques comme des LEGOs et d'apprendre la programmation pour devenir l'inventeur d'objets électroniques.
Le fonctionnement est vraiment très simple. Il y a une base dans laquelle on branche des modules : led, bouton, buzzer, mini écran, afficheur sept segments, Bluetooth, capteur de température ou de mouvement, simple connecteur (à la Makey Makey), etc. On branche la base sur un prise USB de l'ordinateur et en allant sur le site de Thingz on peut créer un programme soit en mode bloc, soit en mode texte. On peut d'ailleurs commencer avec les blocs, puis voir le code texte généré pour ensuite éventuellement le modifier. Cela peut être une très bonne manière de passer d'un mode de programmation à l'autre. [10 et 11]



11

*Exemple d'un programme avec le code bloc et le code texte correspondant*

# ET LA ROBOTIQUE DANS TOUT ÇA ?

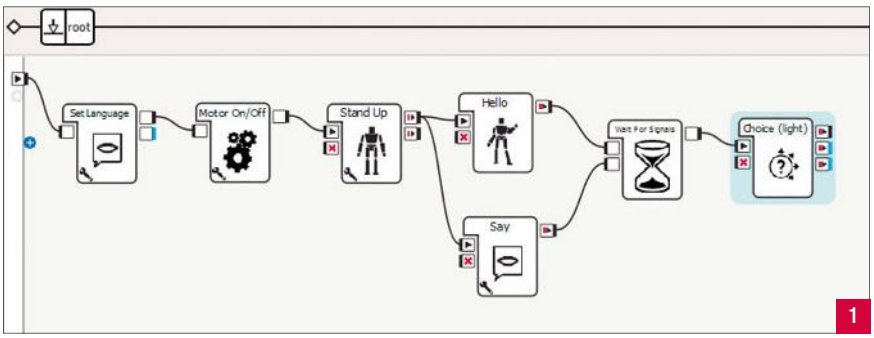
*Lors de Devoxx France cette année j'ai découvert le célèbre robot Nao d'Aldebaran, enfin, de SoftBank Robotics Europe, et j'ai été bluffée. En fin d'année dernière, Blandine et Corinne, de Duchess France, ont animé un atelier Nao lors du RivieraJUG Devoxx4Kids 2015 à Sophia Antipolis, voici un retour de Blandine sur cette expérience :*

Blandine Bourgois
*Développeuse Java chez*
**ENOVACOM**
*Duchess France Leader*
@bbourgois

C'est grâce à Corinne (Krych), une autre Duchess, que je me suis retrouvée dans l'aventure. Elle avait participé à l'édition de l'année précédente de Devoxx4Kids Sophia par le RivieraJUG et m'a proposé de rejoindre l'équipe.
J'ai la chance d'avoir un robot NAO chez moi mais faute de temps et de motivation (ce n'est pas forcément amusant toute seule) il reste souvent au placard. Je savais qu'il existait déjà des ressources pour faire un atelier pour enfant avec NAO (atelier Devoxx4Kids : http://www.devoxx4kids.org/materials/workshops/nao-robot/, atelier de Nicolas Rigaud : http://fr.slideshare.net/NicolasRigaud/nao-robot-workshop-for-kids-2-french-46241830), je me suis dit que c'était l'occasion de me lancer.

Le temps de préparation a été assez court : récupérer le TP, le tester et faire quelques modifications puis expliquer à Corinne le fonctionnement de chorégraphe (le logiciel permettant de programmer NAO). Comme c'était une 2ème édition, l'organisation était déjà rodée (salle, goûter, planning, inscription …).

Le jour J, je suis juste arrivée un peu plus tôt pour préparer les machines et c'était parti pour deux sessions de 1h30 avec 4 binômes d'enfant de plus de 10 ans pour 3 encadrants. Ludovic, le fils de Corinne, était venu nous prêter main forte. Le but étant que les enfants puissent faire plusieurs ateliers dans l'après-midi.



1

L'atelier NAO consiste à programmer le robot pour lui faire raconter une histoire. Le début était détaillé en pas à pas (histoire de prendre ses marques : mettre le robot en français, allumer les moteurs, tester les briques parler et question) puis les enfants pouvaient laisser libre court à leur imagination. Le TP était réalisé sur les PC et testé dans un simulateur puis chaque binôme pouvait lancer son programme sur le robot une fois fini.

Développer sur un robot peut paraître compliqué et rébarbatif. Heureusement, le robot NAO est fourni avec des briques de bases. Il suffit d'appeler la bonne API pour que le robot parle ou se lève. Ce qui est plus déroutant, c'est l'ordonnancement des actions. Le robot peut bouger et parler en même temps. Concrètement, dans Choregraphe, l'IDE, on retrouve toutes les actions de base dans des boîtes (ces boîtes appellent les méthodes des API des briques de base). Chaque boîte a des entrées (start et stop), des sorties et des paramétres. Pour créer un programme, il suffit de relier les boîtes entre elles. Le côté graphique permet une prise en main rapide et les boîtes de logique (if, wait for signals, switch …) une découverte en douceur de l'algorithme.

Dans l'exemple ci-dessous, le robot se lève puis parle pendant l'animation Hello. [1]
C'est un très bon souvenir et mon robot n'a pas souffert. Chaque binôme a fait un programme très différent. Certains n'ont même pas suivi la partie pas à pas pour partir à la découverte des possibilités du robot.

En bilan, c'était une après-midi intense ; on a initié 16 enfants. Mon Nao connaît maintenant quelques histoires farfelues, les enfants étaient très contents : ils nous ont même remerciés, et les parents ont apprécié les démos.

# Play'n'Code : un jeu pour apprendre à coder

• Christelle PLISSONNEAU CEO

*Early Birds Studio, une startup née d'étudiants d'Epitech, développe Play'n'Code : le jeu vidéo pour PC et Mac qui apprend aux enfants de 8 à 12 ans à coder en s'amusant ! Découvrons ensemble les dessous du projet.*

## Comment installer Play'n'Code ?

Depuis votre ordinateur, rendez-vous sur le site Internet de la startup https://early-birds-studio.fr et cliquez sur « Jouer à la démo ». Choisissez PC ou Mac, un installateur va se télécharger. Suivez les instructions pour installer le jeu sur votre ordinateur. Une fois le jeu ouvert, vous arrivez devant les portes de votre vaisseau spatial ! Pour y rentrer, vous devrez créer un compte famille. Cliquez sur « Créer un compte » vous emmènera sur le site Internet. Une fois vos informations remplies, un mail de confirmation vous sera envoyé. Après avoir validé votre email, vous pourrez revenir dans le jeu et vous connecter pour rentrer à bord de votre vaisseau ! A partir de là, vous pourrez laisser vos enfants prendre les commandes, le jeu est en pilote automatique. Il pourra créer une nouvelle partie et choisir son personnage pour commencer le jeu.


## Play'n'Code, c'est quoi ?

Après avoir choisi un garçon ou une fille dans le menu principal, les enfants choisissent un nom de héros, saisissent leur âge, et le jeu démarre ! Malheureusement, pendant le chargement du jeu, le vaisseau s'est écrasé sur une planète inconnue. Les pièces du vaisseau se sont éparpillées aux quatre coins de cette planète, et le but est de les retrouver pour réparer le vaisseau et le faire rentrer chez lui. L'enfant va devoir programmer un petit script à la fin du jeu grâce à tout ce qu'il a appris au cours du jeu.

Pour récupérer les pièces du vaisseau, votre enfant va parcourir la planète sur laquelle il s'est écrasé, et parler aux différents personnages qui se trouvent sur son passage. Avec un peu de chance, ils auront des pièces du vaisseau ! Et surtout, il va découvrir la programmation en faisant des mini-jeux. TTY, le robot copain du joueur, volera à ses côtés tout au long de l'aventure pour lui expliquer les notions de programmation qui sont expliquées.

## Qu'est-ce qu'on apprend dans ce jeu ?

L'équipe a créé un langage de programmation en français, sans ponctuation complexe telle que le point-virgule (pour fermer l'instruction) ou l'accolade (pour délimiter une fonction). Ce langage, proche du C ou du Lua, permet aux enfants de comprendre la logique du code, travailler l'algorithmie, tout en se rapprochant au maximum des langages existants sur le marché et qui sont en anglais. Un programme pédagogique a été créé autour de ce langage, et les notions que l'on retrouve dans tous les langages et tous les programmes, sont expliquées : variables, boucles, conditions et fonctions. Au début du jeu, les enfants apprennent à se servir des touches du clavier de l'ordinateur, la souris, et la notion de « programmation » leur est expliquée au moyen de mini-jeux. Toutes les notions sont expliquées en plusieurs étapes, qui sont elles-mêmes sous découpées selon le degré de difficulté de l'exercice. Les notions de code sont expliquées simplement au début, dans leur concept. Par exemple, une variable est une petite boîte dans laquelle on peut mettre des choses. Mais pas n'importe lesquelles : chaque conteneur ne peut accueillir qu'un type de contenu (ex : de l'eau dans un verre). Les syntaxes des notions sont ensuite expliquées, au moyen de glisser-déposer comme Scratch. Le but ici est d'afficher les premiers bouts de code liés aux notions que l'enfant a vues dans la première partie. Dans la 3e étape, les enfants sont amenés à réutiliser les mots-clés qu'ils ont vus dans les 2 premières étapes, et les écrire eux-mêmes. Des tutoriels sont là pour expliquer, et les enfants peuvent les regarder aussi souvent qu'ils veulent. La dernière étape arrive à la moitié du jeu. Rapidement, les enfants vont devoir écrire leur premier script ! Ils ont vu tout le programme pédagogique, et ils sont de plus en plus autonomes.

## Organisez le jeu en sessions

Avec une quarantaine de mini-jeux, le jeu dure au moins 4 heures. L'équipe conseille d'organiser des sessions de 30 minutes pour ne pas noyer vos enfants.

Une application mobile « Compagnon » est disponible pour suivre la progression des enfants dans le jeu : temps total sur le jeu, temps passé par mini-jeu, explication des notions de programmation qu'ils ont vues… Une alerte est paramétrable depuis l'application mobile pour que le petit robot TTY déclenche un message dans le jeu : « Et si nous revenions plus tard ? Je commence à être fatigué ! ». •

# Quelques outils et matériels

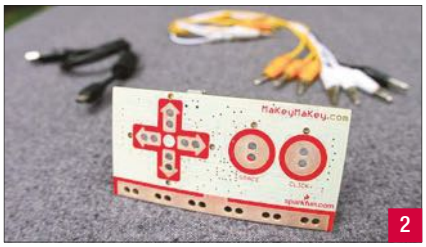
*(liste non exhaustive)*

## MAKEY MAKEY

Même s'il ne s'agit pas directement de programmation, il y a un compagnon incontournable pour piloter vos créations. Nous voulons parler de Makey Makey, un prolongement tentaculaire de votre ordinateur. Makey Makey est un kit basé sur Arduino qui permet d'interfacer des objets conducteurs d'électricité (des fruits par exemple) avec l'ordinateur et de faire des applications amusantes. Le Makey Makey émule les 4 flèches du clavier et les 2 boutons de la souris. Le fonctionnement est simple : on branche le Makey Makey sur un port USB, on branche quelques fils du Makey Makey sur, disons, quelques bananes, et juste en touchant les fruits vous pouvez piloter votre jeu (ou tout autre logiciel en fait) ! Cela fonctionne par exemple aussi avec de la pâte à modeler. En réalité ce dispositif est vu par l'ordinateur comme un clavier et il affecte automatiquement chaque fil (et donc ce à quoi le fil est connecté) à une touche du clavier (les quatre flèches, la touche espace…). Voilà, aucune configuration, on branche et ça marche. On vous invite à aller regarder la vidéo de présentation de Makey Makey (https://www.youtube.com/watch?v=rfQqh7iCcOU) pour voir tout ce qu'on peut faire : un piano avec les marches d'un escalier, une fausse guitare en carton qui joue les sons de l'ordinateur, etc. On vous fait confiance pour trouver d'autres idées bien délirantes. [Fig.1, 2 et 3]

Si vous voulez aller plus loin avec Makey makey : http://magicmakers.fr/blog/comment-faire-un-petit-jeu-fun-avec-scratch-et-makey-makey


1



2

3

## LEGO MINDSTORMS

***(à partir de 10 ans)***

Les Lego Mindstorms sont des Lego bien particuliers : équipés de plusieurs moteurs, de capteurs et d'une brique programmable, ceux-ci peuvent exécuter des programmes écrits à l'aide du logiciel fourni par LEGO. L'éditeur de Lego peut, par certains aspects faire penser à Scratch. On y assemble des briques pour construire notre programme, chaque brique correspondant à une instruction. De la même manière, les briques de même nature sont rassemblées ensemble pour nous aider à nous y retrouver.

Par exemple, toutes les instructions concernant les moteurs seront regroupées ensemble, alors que tout ce qui concerne les boucles sera dans un autre groupe.

Pour nous aider, le kit propose plusieurs modèles de montages (véhicule, robot, araignée), chaque modèle ayant également plusieurs tutoriels intégrés.

Une fois le programme écrit, un câble va nous permettre de le charger dans la brique programmable et de le faire exécuter par celle-ci.

Le kit est fourni avec 3 moteurs : deux qui servent au déplacement de la structure, le troisième servant à faire fonctionner d'autres mécanismes montés par l'enfant comme un lance bille par exemple.

On peut ajouter à ces moteurs un détecteur de couleurs et un capteur d'ultrasons, ce qui peut par exemple nous permettre de faire se déplacer un véhicule sur un circuit et d'adapter son comportement en fonction des couleurs qu'il détecte sur ce circuit, ou encore de stopper lorsque son capteur ultrason rencontre un obstacle.


4

5

Deux versions sont disponibles :
la version "grand public" fonctionne avec des piles, la version "éducation" est un petit peu plus chère mais fonctionne sur batterie rechargeable, ce qui peut être un plus en atelier.

Ces Legossont plutôt destinés aux jeunes de plus de 10 ans, le montage et l'interface pouvant paraître un peu complexes à la plupart des enfants plus jeunes. [Fig.4 et 5]

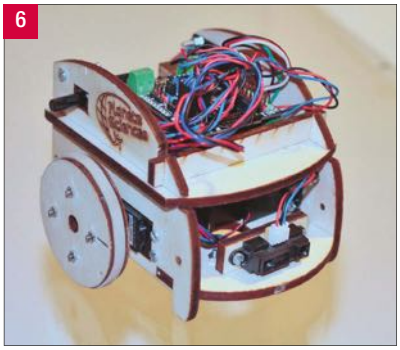
Si vous voulez aller plus loin, l'association Devoxx4Kids France met à disposition une documentation sur les Mindstorms permettant de créer des ateliers sur ce sujet : http://devoxx4kidsfr.github.io/materials/ateliers/mindstorm/index.html

## BOUM'BOT, le robot du bricoleur

L'association Planète Sciences est particulièrement active dans la pédagogie en robotique notamment à travers l'organisation de concours : les Trophées de la Robotique, pour les moins de 18 ans, et la Coupe de Robotique, pour les adultes et en particulier les étudiants. Il y a quelques années, elle s'est lancée dans le développement d'un robot en kit pour initier les jeunes et moins jeunes à la programmation.

Boum'Bot — c'est son nom — est fabriqué uniquement à partir de matériel que l'on peut

acheter dans le commerce : carte Arduino UNO, shield et capteurs DFRobot, servomoteurs standards… Côté structure, le châssis est découpé au laser dans du contreplaqué et peut être réalisé dans n'importe quel FabLab.
Le robot dispose de 2 moteurs pour se déplacer (pilotables en vitesse) et d'un petit haut-parleur pour émettre des sons. Côté capteurs, il dispose d'un capteur de distance infrarouge et de deux capteurs de ligne au sol. L'ensemble des plans et des manuels est téléchargeable sur le site Web de l'association (http://www.planete-sciences.org/robot/boumbot).
La boutique Arobose propose un kit avec tous les éléments inclus au prix de 90€.
[Fig.6]
Côté programmation, on peut au choix programmer directement dans l'Arduino IDE en C++, ou alors en utilisant l'interface UniBot, développée par l'association, qui propose de programmer à l'aide de blocs qui s'emboîtent. Tous les concepts fondamentaux de la programmation peuvent être abordés : séquences d'instructions, tests et sélections, boucles "tant que" et variables. Des notions plus avancées comme l'asservissement et les contrôleurs PID peuvent être mises en pratique lors de concours de suivi de ligne par


6

exemple ou des sorties de labyrinthe.
Le robot est utilisé régulièrement par l'association lors d'évènements comme la fête de la science ou en partenariat avec les écoles, collèges et lycées (voire même parfois à l'université !).
L'utilisation d'un matériel 100 % standard permet d'ajouter facilement des capteurs et autres actionneurs pour étendre les possibilités du robot, ou simplement des petites barres sur les côtés afin qu'il puisse attraper des objets au sol. Son look en bois et ses connexions apparentes et accessibles en font le robot idéal pour les bricoleurs ou les enseignants en technologie souhaitant l'adapter pour un projet scolaire. L'assemblage du kit nécessite seulement un tournevis et un peu de patience.
Présente dans plusieurs régions, l'association propose des formations et une aide technique dans différentes villes : Paris, Toulouse et Lyon.
Âge minimum pour le programmer : 8 ans
Âge recommandé : à partir de 10 ans.

## CONCLUSION

Nous aurions aimé vous parler de bien d'autres choses encore telles que de la création de mods Minecraft, du monde merveilleux des Arduino, de l'initiation à la programmation sans écran avec le jeu de cartes The Bugs de Magik Square et bien plus encore mais pour cette fois-ci, toute bonne chose à une fin, et sera l'objet d'un autre article prochainement, avec en bonus track une revue de livres et des témoignages de programmeur.se.s en herbes. •

# Vacances de Toussaint : un stage de **Kid Coding Robot** remporte un franc succès

• Olivier Pavie

*C'est à Saint Raphaël, dans le Var, que l'espace de Coworking Point Sud a lancé il y a quelques mois une initiative de demi-journées de Kid Coding. Après le succès remporté, l'idée a été de proposer chaque après-midi pendant une semaine, un stage pour apprendre à programmer un robot en le faisant évoluer tous les jours. Des enfants heureux qui sont repartis avec leur robot sous le bras.*

Louis Casa est l'instigateur et le dirigeant de l'espace de Coworking de Saint Raphaël, un homme de communication et d'évènementiel qui aime surfer sur tout ce qui touche aux nouvelles technologies. Julien Devincre est un des coworkers, dirigeant de sa société de services Technolan, informaticien, jeune papa, qui a su répondre à l'appel du pied de Louis pour se pencher sur cette proposition d'activité ouverte à tous les enfants de 8 ans à 15 ans. Certes, chaque enfant doit venir avec un ordinateur personnel pour pouvoir participer, et le stage a aussi un coût : 180 €, les enfants repartant avec leur robot.

## Un important travail de préparation

« Préparer un stage de 5 après midi de Kid Coding est un véritable travail » nous confie Julien. « Mais la satisfaction du résultat est à la hauteur de l'investissement, les enfants ont passé une super semaine et il était étonnant de voir à quel point ils se concentraient ». Comment s'est préparé ce stage ? Julien a parcouru les sites, s'est documenté, le but était aussi de trouver un robot qui ne soit pas trop cher et soit capable d'exploiter des capteurs infrarouge, des capteurs ultrason, qu'il puisse disposer du WiFi pour arriver à être autonome. Mais l'essentiel étant aussi qu'il puisse être évolutif, à la fois dans les options qu'il pouvait embarquer mais aussi dans les modes de programmation qui pouvaient être exploités. Le mBot de Makeblock s'est imposé pour son rapport qualité/prix, sa possibilité d'être programmé en Scratch mais aussi sa carte électronique de type Arduino avec toutes les possibilités de connectiques déjà intégrées sur la carte en natif.

## Se fixer des objectifs et essayer de les atteindre

Dans la préparation du Stage, Julien Devincre s'est aussi et bien évidemment penché sur le contenu éducatif. Comment arriver à motiver huit enfants et ne pas en laisser en route. « Avec

*1er jour, les enfants découvrent le robot et apprennent à le monter*

*Les robots s'affutent à tâcher de suivre la ligne*

*Les enfants ont tous eu un "diplôme de fin de stage"*

## Du côté des enfants

Victor est un des enfants qui ont fortement apprécié le stage. Il est prêt à revenir et à apprendre d'autres choses. Il ne connaissait rien à la programmation des robots et il avoue que maintenant il s'amuse à tester de nouvelles possibilités avec les capteurs, des finesses dans la programmation. « Ce qu'il y avait de réellement passionnant c'est la manière dont on a réfléchi au comportement du robot qui devait suivre cette ligne noire et qui après devait trouver un moyen de contourner un obstacle tout en retrouvant la ligne noire qui lui donne son cheminement. »

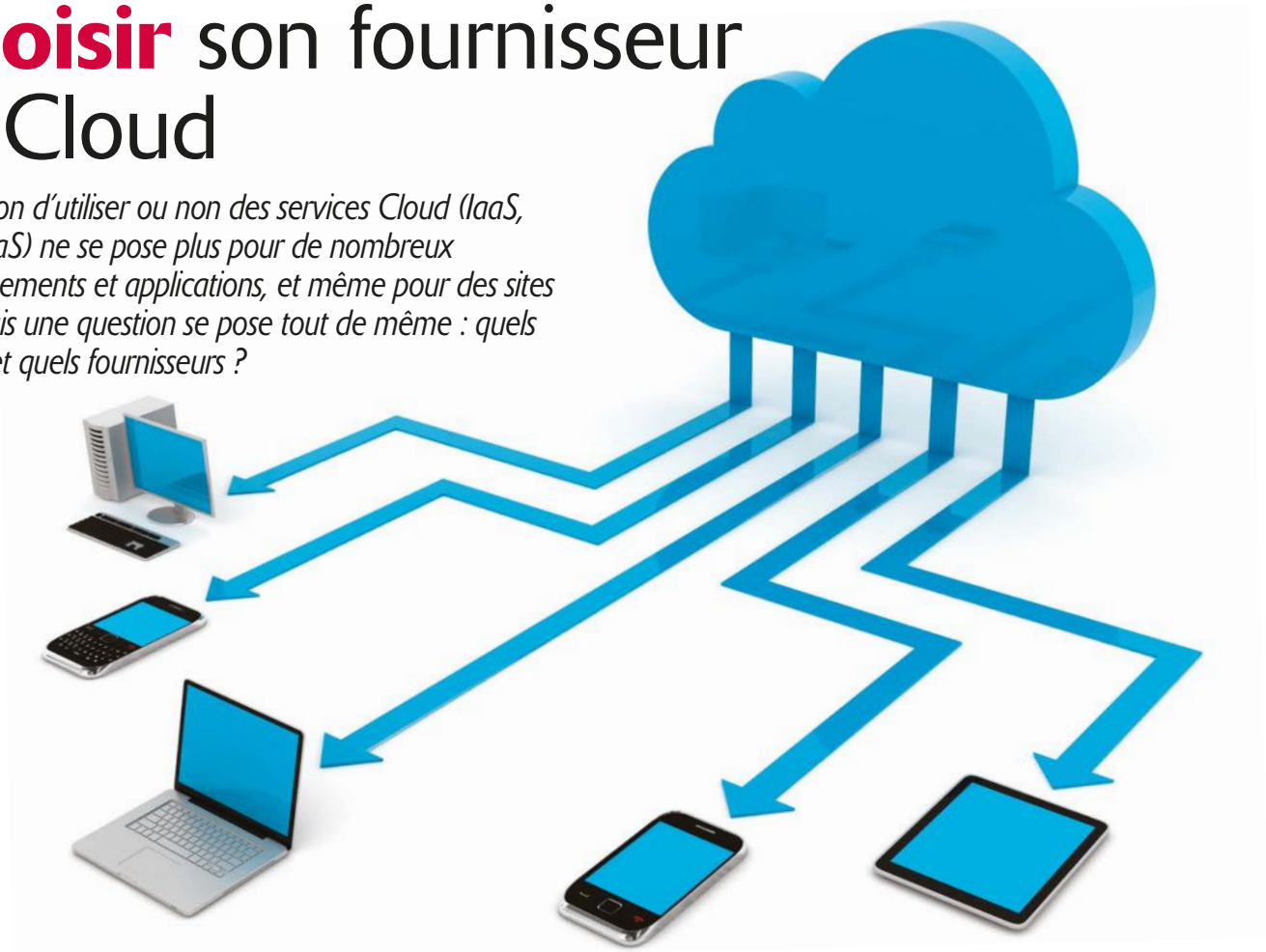
les enfants de 8 à 10 ans, c'était parfois compliqué en matière de concentration, les pré-ados étaient vraiment toujours en réflexion et dans l'action. On a commencé la première demi-journée à détailler les pièces du robot et à les assembler : les enfants avaient hâte de passer à la programmation ! Dès le second jour on s'est attaqués à la programmation : le but, comprendre leur fonctionnement et commencer à initier un comportement en suivant une ligne noire tracée sur le sol ». L'imagination de Julien Devincre et de Louis Casa a permis de préparer un genre de Robot Park dans lequel tous les robots pouvaient se retrouver et s'essayer à mettre en pratique les premiers éléments permettant de déplacer le robot en utilisant le repère de la ligne noire. L'avantage du mBot, c'est qu'il peut travailler dans un mode où il exécute les commandes en même temps qu'elles sont entrées dans la machine, sans besoin de télécharger le programme dans l'Arduino ; « évidemment, c'est lents, ce n'est pas du vrai temps réel, mais pour le débogage et l'analyse du comportement des capteurs, c'est un mode de fonctionnement très pratique et très ludique ; ce n'est qu'une fois que le programme est téléchargé dans l'Arduino que le robot exécute rapidement l'algorithme » précise Julien.

## UNE EXPÉRIENCE À RENOUVELER

Que ce soit pour les organisateurs ou pour les enfants dans leur majorité, l'expérience est réussie. Plusieurs enfants pensent déjà à ce qu'ils vont essayer avec leur robot, voire attendent que les organisateurs proposent des demi-journées de kid coding robot en attendant peut-être un nouveau stage de programmation encore plus avancée. •

# Choisir son fournisseur de Cloud

*La question d'utiliser ou non des services Cloud (IaaS, PaaS, SaaS) ne se pose plus pour de nombreux développements et applications, et même pour des sites Web. Mais une question se pose tout de même : quels services et quels fournisseurs ?*



Pour un même service, le développeur aura souvent plusieurs choix possibles chez différents fournisseurs de Cloud, mais pas toujours au même niveau de fonctionnalités ou reposant sur les mêmes piles techniques. Parfois ce n'est pas gênant, parfois, c'est contraignant.

Si vous utilisez surtout des services IaaS (donc côté infrastructure, serveurs), le choix du fournisseur sera plus agnostique, dans le sens où vous aurez moins de contraintes ou d'adhérences techniques ; sauf si vous utilisez des VM ou des conteneurs Windows par exemple, car là, le nombre de fournisseurs est plus restreint. On trouvera plus facilement le support des machines virtuelles Linux que Windows.

Aujourd'hui, les services se multiplient, les fournisseurs aussi, mais seule une poignée domine réellement le marché mondial dont Amazon, IBM Bluemix, Google, Microsoft, Heroku (Salesforce). À cela se rajoutent des acteurs locaux ou venant de l'hébergement Internet tels OVH, 1&1, Ikoula, Gandi, Treeptik, etc.

De nombreux critères peuvent être utilisés pour choisir : les technologies ou langages supportés, le SLA (la qualité de services / disponibilité), les tarifs, localisation des datacenters, latence réseau, fonctions de réplication, sécurité, fonctionnalités, connectivité hybride, base de données, type de machine virtuelle supportée, outils DevOps, fonctions de préproduction, fonctions de montée en charge automatique, template Cloud de type JSON, outils et portail d'administration, support, open source ou pas, etc. Bref, vous n'aurez que l'embarras du choix pour les questions à se poser.

## Des mondiaux, de plus en plus mondiaux et locaux

Des acteurs français mettaient en avant la disponibilité de datacenters en France, mais avec les annonces faites par Amazon et Microsoft qui ouvriront des datacenters français en 2017, l'argument deviendra moins décisif. Le développeur sera plus sensible aux langages et frameworks supportés, et la possibilité de déployer directement depuis son IDE. On peut aussi dire qu'un développeur .Net sera plus sensible à Azure, même si d'autres Clouds supportent certaines piles Microsoft, même si vous pouvez interfacer un front .Net avec un back non .Net. Dans le cadre d'un projet Java, JavaScript, PHP, DRUPAL, de backend mobile ou de contenus média, le choix sera large.

L'investissement sur les 9 premiers mois de l'année des 3 principaux fournisseurs de Cloud (= Amazon Web Services, Microsoft, Google) est impressionnant : presque 30 milliards $. Ils sont engagés dans l'hyperscale. Cette notion désigne le déploiement mondial avec x régions et x datacenters des services Cloud pour une couverture mondiale de ceux-ci. Quand vous devez servir des utilisateurs en Asie, en Amérique, en Europe, mieux vaut déployer les applications au plus près de ceux-ci pour opti-

## LES CRITÈRES DE NICOLAS B.

1 : Pertinence de la solution pour le besoin et pertinence globale ;
2 : Sécurité (notamment des données en fonction des projets/acteurs) ;
3 : Coût ;
4 : Compatibilité/Ouverture/Support à moyen/long terme ;
5 : Écosystème partenaire et compétences disponibles sur le marché.

À pondérer / adapter selon le contexte projet

miser le réseau et les performances (la latence n'est pas un mythe). On parle de plusieurs dizaines de datacenters, autant de zones et de régions. Une région correspond à une zone géographique plus ou moins grande.

Si nous prenons le marché français, l'offre Cloud n'est pas moins riche qu'ailleurs, même si au début, les entreprises et les développeurs ont eu du mal à s'y mettre. Contraints ou non, ils y vont. Et les grands acteurs mondiaux investissent désormais le pays pour y installer des datacenters. Amazon et Microsoft les ont annoncés pour 2017. Un des derniers arguments pour les acteurs français tombera dans quelques mois. IBM était déjà installé, mais continue à étendre le déploiement de Softlayer et surtout de Blumix, le PaaS d'IBM s'appuyant sur CloudFoundry.

## Éternel débat : pour les nouveaux projets ou les apps existants

Il y a fondamentalement deux types de projets pour les développeurs :

- Les projets existants qui évoluent ou non. Ils font partie du patrimoine applicatif, ou legacy. Ils sont parfois très anciens ou utilisent des technologies aujourd'hui obsolètes ou plus supportées. Souvent difficiles à migrer vers une nouvelle technologie sans une réécriture en profondeur ou le besoin de revoir l'architecture logicielle.
- Les nouveaux projets : ils peuvent offrir une grande souplesse dans les technologies (en principe).

On dit souvent que les nouveaux projets sont les candidats parfaits pour les nouvelles architectures, et encore pour utiliser des services Cloud dans les applications mobiles, des solutions Web, pour étendre les services serveur sur le Web, etc. Mais si vous n'avez pas besoin de services Cloud, ne vous obligez pas à en utiliser.

La refonte d'un projet, d'une application déjà en production peut être l'occasion de revoir l'architecture et d'y injecter des technologies, des fonctions nouvelles, du moins plus récentes que celles utilisées précédemment. Par exemple, remplacement de Flash, utiliser du responsive design, nouveaux types de données supportés, API à la place de connecteurs monolithiques, frameworks pour nettoyer et alléger le code, etc.

Des fonctions peuvent être déportées dans un service : bases de données, analyses / traitements de données, stockage, données non structurées, traitement 2D / 3D, authentification, services médias, services cognitifs, reconnaissance vocale, etc. Nous serons souvent dans du backend et non sur le frontend.

## Au-delà du code, il y a le DevOps

Vous vous demandiez quand nous allions parler du DevOps. Oui, au-delà de la programmation et de votre développement, il y a votre environnement de développement et de déploiement qui devient de plus en plus Cloud, que vous le vouliez ou non. Le mouvement DevOps accélère cette tendance qui est là pour durer.

Depuis plusieurs années, les IDE sont connectés au Cloud notamment pour le déploiement. On pouvait assez facilement déployer son projet sur le Cloud en quelques minutes. Aujourd'hui, de plus en plus d'outils et de services projets sont en mode Cloud. Les outils de Build en sont un exemple, tout comme les outils ALM et de gestion de projets, les bases de données.

Avec le Cloud, vous pouvez monter très rapidement vos environnements de tests fidèles aux environnements de production.

Le Dev & Test était un des premiers scénarios les plus utilisés.

## Un support étendu des frameworks et des langages

Hier, pour déployer un site Web, vous utilisiez la plupart du temps un hébergeur Web en serveur mutualisé ou dédié. Pour créer des workloads de production pour déployer vos applications, il fallait créer des instances ; quand vous étiez dans une entreprise, cette demande pouvait prendre plusieurs jours ou semaines.

Le Cloud Computing a apporté une flexibilité que nous n'avions pas, même si les hébergeurs ont fait beaucoup de progrès ces dernières années pour faciliter l'allocation des ressources et le support des technologies plus nombreuses. Aujourd'hui, les fournisseurs de Cloud généralistes supportent une quantité impressionnante de langages, de runtimes d'exécutions, de frameworks. Vous voulez déployer du Java, PHP, Drupal, ASP.Net, un backend pour application mobile, créer 2 bases de données, faire du stockage objet, utiliser Node.JS ou tout framework JS ? Vous aurez souvent plus fournisseurs possibles et le provisionnement prendra seulement quelques minutes. 5 minutes suffisent pour déployer une solution Drupal, à peine plus pour créer et déployer un cluster Hadoop…

Cette grande souplesse se retrouve sur la partie PaaS, mais aussi sur le IaaS. Le IaaS, côté développeur, permet de déployer un workload technique dédié, idéal pour les anciennes applications par application, avec tout ce qu'il faut à l'intérieur. Que l'on soit en machine virtuelle ou en conteneur. L'approche conteneur permet une plus grande souplesse, mais attention aux contraintes des conteneurs Linux ou Windows. N'oubliez pas d'avoir de bons outils d'administration et d'orchestration. Même si le développeur n'a pas vocation à gérer le déploiement et la production, il peut savoir comment cela fonctionne et les mécanismes liés. Ce n'est jamais inutile !

## Soyez pragmatique et non dogmatique

Le choix du Cloud ne doit pas être pris à la légère. Vous devrez avoir une vision d'ensemble, bien définir vos besoins et les évolutions possibles, calculer, si besoin, le coût du Cloud, les nécessités de connectivités hybrides ou inter-cloud, etc. Si aujourd'hui, il est plus simple de migrer d'un Cloud à un autre Cloud, tout n'est pas simple, et malgré les progrès réalisés sur certains services Cloud, vous ne pourrez pas facilement passer d'un fournisseur à un autre. Car, si vous allez gagner en abstraction et couplage lâche, une adhérence peut apparaître. Qu'on le veuille ou non, le Cloud n'est pas synonyme de liberté totale, d'agnosticisme ou de couplage lâche partout. •

La rédaction.

## LES LEADERS DU MARCHÉ MONDIAL

***IaaS :***

Amazon Web Services ..... +40 %
Microsoft, Google, IBM ..... 18 %

***PaaS :***

Amazon Web Services ..... env. 33 %
Salesforce, Microsoft, IBM ..... env. 35 %

***Cloud privé managé :***

IBM ..... env. 18 %
Amazon Web Services,
Rackspace, NTT ..... 20 %

La forte présence d'Amazon étonne sur la partie PaaS mais la définition est parfois très large. La concurrence y est très forte.

*Source : Synergy Research Group pour le 3e trimestre 2016. Sur les revenus générés par le Cloud.*

# Comment choisir son fournisseur de Cloud ou son hébergeur ?

• Ludovic Piot
*Responsable du pôle Conseil, Architecture et DevOps*
@0xalide

*L'évolution des bonnes pratiques de développement vers l'agile et l'intégration continue a exacerbé la pression sur l'infrastructure : livrer toujours plus vite, de manière toujours plus sûre, sur un éventail de technologies toujours plus large. Dans le même temps, de nouveaux acteurs sont apparus (comme ce spin-off d'un challenger de la Redoute), qui ont totalement révolutionné la manière dont on consomme de l'infrastructure.*

Le Cloud public a transformé les services de l'IT traditionnel en autant de produits rendus directement accessibles aux développeurs, en *self-service*, sans (grande) limite, et sans engagement. La facture correspond à l'usage. Mais avec l'emploi de matériel ordinaire et un modèle de résilience et de performance basé sur des architectures distribuées, le Cloud public a aussi fortement altéré les principes suivant lesquels on conçoit les applications. Là où la résilience se faisait par le choix de matériel sophistiqué, elle est portée aujourd'hui par l'architecture logicielle. Là où la performance se faisait par une sélection minutieuse du matériel le plus adéquat (mais aussi le moins réutilisable à d'autres usages), elle est aujourd'hui portée par la multiplication de nœuds *idempotents* d'un cluster. Là où les ops proposaient des design d'infrastructures sur-mesure, il s'agit maintenant d'offrir vite et efficacement du générique en *brute force*. L'infrastructure devient un consommable.
A l'heure où l'équipe de développement doit faire le choix d'un hébergeur ou d'un fournisseur de Cloud public, ces bouleversements influent sur les critères de sélection de l'offre adéquate, voyons ce qui a changé.

## 1 - Une plateforme ready-to-use

Dès le début du développement d'une nouvelle application, le développeur se confronte à un besoin immédiat : disposer rapidement de la pile de composants techniques nécessaires pour faire fonctionner son application : un langage de programmation, bien sûr, et des frameworks de développement, mais aussi une base de données, un serveur Web, un composant de gestion de cache, un outil d'indexation pour gérer les recherches, etc. Dans une démarche fail fast, moins le développeur passe de temps pour construire sa pile technique et mieux il se porte.

### 1.1 - Les plateformes PaaS généralistes

C'est la promesse des solutions *PaaS* : mettre à disposition du développeur, dans le Cloud public, une plateforme de *run* pré-construite et gérée pour lui, embarquant les composants techniques nécessaires et suffisants pour faire tourner son application. Les 3 plateformes de référence dans le domaine sont Google App Engine, Heroku et Elastic Beanstalk sans oublier Azure et bluemix.
Les 2 premiers présentent une philosophie similaire : le développeur ouvre un compte sur la plateforme, identifie les environnements ( *stagging*, *prod*) qui seront utilisés dans son pipeline d'intégration continue / déploiement continu. Puis il sélectionne les composants de la stack technique qu'il souhaite utiliser (langage de programmation, moteur de persistance, etc.), soit dans la console Web, soit dans un fichier de description de la configuration, puis il provisionne une première plateforme. Le sizing que l'on choisit est forcément théorique, dans la mesure où ces plateformes font une abstraction de la notion de nœuds de calcul vers une notion "*maison*" ( *dynos* chez Heroku, *runtime instances* chez GAE), dont il faudra éprouver la performance à l'usage. Il est alors possible de déployer son application sur cette première plateforme, à travers quelques lignes de commande. Elastic Beanstalk présente une philosophie légèrement différente, dans la mesure où il s'agit plus d'*IaaS* que l'on peut pré-configurer au *provisionning* : il n'y a aucune abstraction de l'implémentation technique de la plateforme, on peut manipuler les instances EC2 et les *middlewares* qui y sont déployés, autorisant ainsi du fine tuning, impossible sur les 2 autres plateformes *PaaS*. Pour ces solutions, il existe quantité de modules pour Maven, npm, ou composer dans le monde PHP. On peut aussi piloter ces plateformes depuis les outils de gestion de configuration comme Ansible. Enfin des plug-ins permettent de piloter directement le *provisionning* et le déploiement dans ces plateformes depuis les principaux *IDE*, Eclipse et IntelliJ. On le voit, la promesse de ces plateformes de fournir rapidement des stacks techniques prêtes à l'emploi, après un rapide apprentissage des quelques commandes nécessaires n'est pas usurpée. Par ailleurs, ces plateformes présentent aujourd'hui une très grande polyvalence, en offrant une compatibilité avec les principaux langages de développement du moment (Java, JS, Python, PHP, Ruby, voire Go) et avec des milliers (littéralement) de composants techniques (bases de données, moteurs de cache, indexeurs, message brokers, injecteurs pour tirs de performance, etc.). Elles savent donc répondre rapidement et simplement à une très grande majorité des cas d'usage qu'un développeur peut rencontrer.

### 1.2 - Les plateformes PaaS dédiées

Pour forcer encore le trait et aller plus loin, des plateformes PaaS tout spécifiquement dédiées à des cas d'usage spécifiques sont disponibles depuis quelques mois. Le développeur PHP qui souhaite développer une application de contenus basée sur Drupal ou Wordpress, ou un site de e-Commerce basé sur Magento, va tout naturellement se diriger vers Platform.sh, qui propose des solutions clé-en-main pour ce genre d'usage. Ainsi, pour créer une application Drupal, le développeur n'a à produire que 2 fichiers de configuration, l'un pour décrire l'application et l'autre pour décrire les services sous-jacents (base de données en premier lieu) …

***Description de l'application***

**.platform.app.yaml**

```
# This file describes an application. You can have multiple applications
# in the same project.
#
# See https://docs.platform.sh/user_guide/reference/platform-app-yaml.html

# The name of this app. Must be unique within a project.
name: 'app'
```

```
# The runtime the application uses.
type: 'php:5.6'

# Configuration of the build of this application.
build:
  flavor: drupal

# The build-time dependencies of the app.
dependencies:
  php:
    "drush/drush": "^8.0"

# The relationships of the application with services or other applications.
#
# The left-hand side is the name of the relationship as it will be exposed
# to the application in the PLATFORM_RELATIONSHIPS variable. The right-hand
# side is in the form `<service name>:<endpoint name>`.
relationships:
  database: 'mysqldb:mysql'
#   solr: 'solrsearch:solr'
#   redis: 'rediscache:redis'

# The configuration of app when it is exposed to the web.
web:
  # Specific parameters for different URL prefixes.
  locations:
    '/':
      # The folder from which to serve static assets, for this location.
      #
      # This is a filesystem path, relative to the application root.
      root: 'public'

      # How long to allow static assets from this location to be cached.
      #
      # Can be a time in seconds, or -1 for no caching. Times can be
      # suffixed with "s" (seconds), "m" (minutes), "h" (hours), "d"
      # (days), "w" (weeks), "M" (months, as 30 days) or "y" (years, as
      # 365 days).
      expires: 5m

      # Whether to forward disallowed and missing resources from this
      # location to the application.
      #
      # Can be true, false or a URI path string.
      passthru: '/index.php'

      # Deny access to static files in this location.
      allow: false

      # Rules for specific URI patterns.
      rules:
        # Allow access to common static files.
        '\.(jpe?g|png|gif|svgz?|css|js|map|ico|bmp|eot|woff2?|otf|ttf)$':
          allow: true
        '^/robots\.txt$':
          allow: true
        '^/sitemap\.xml$':
          allow: true

    '/sites/default/files':
      # Allow access to all files in the public files directory.
      allow: true
      expires: 5m
      passthru: '/index.php'
      root: 'public/sites/default/files'

      # Do not execute PHP scripts.
      scripts: false

      rules:
        # Provide a longer TTL (2 weeks) for aggregated CSS and JS files.
        '^/sites/default/files/(css|js)':
          expires: 2w

# The size of the persistent disk of the application (in MB).
disk: 2048

# The mounts that will be performed when the package is deployed.
mounts:
  '/public/sites/default/files': 'shared:files/files'
  '/tmp': 'shared:files/tmp'
  '/private': 'shared:files/private'
  '/drush-backups': 'shared:files/drush-backups'

# The hooks executed at various points in the lifecycle of the application.
hooks:
  # We run deploy hook after your application has been deployed and started.
  deploy: |
   cd public
   drush -y updatedb

# The configuration of scheduled execution.
crons:
  drupal:
    spec: '*/20 * * * *'
    cmd: 'cd public ; drush core-cron'
```

### *Description des services*

**.platform/sevices.yaml**

```
# The services of the project.
#
# Each service listed will be deployed
# to power your Platform.sh project.
mysqldb:
  type: mysql:10.0
  disk: 2048
#rediscache:
#   type: redis:3.0
#
#solrsearch:
#   type: solr:3.6
#   disk: 1024
```

De manière encore plus extrême, à la limite du *SaaS*, des offres de Cloud public proposent des plateformes orientées cas d'usage métier pré-installées,

où le développeur ne viendra faire que quelques modifications dans un applicatif déjà monté pour lui. Côté e-commerce, des offres à base de Magento existent chez Zoey ou TheOtherStore. Côté contenu et CMS, on a du Drupal chez Acquia, par exemple.

### 1.3 - Avantages et inconvénients

Très rapides à mettre en œuvre et à appréhender, polyvalentes, riches de milliers de composants techniques, managées et résilientes, ces plateformes ont tout pour plaire. Alors, PaaS = monde idéal ? Pas tout à fait.
Tout d'abord, ces plateformes sont coûteuses dès que l'on en fait un usage réel de production. Et il est fréquent que le développeur soit amené à migrer vers des solutions Cloud plus économiques au bout de quelques mois. Ensuite, les mécanismes de déploiement sur ces plateformes restent très liés à la plateforme, et le portage vers une autre solution nécessite un refactoring de la chaîne de CI/CD.
Enfin, ces plateformes sont par essence exclusivement sur le Cloud. Impossible, pour le développeur d'utiliser son écosystème sur son ordinateur portable pour une utilisation dans le TGV, par exemple.

## 2 - Le monde des containers

L'écosystème Docker et son principal orchestrateur *prod ready* Kubernetes ont ouvert la voie d'une nouvelle forme de plateformes Cloud : le *CaaS* pour *Container as a Service*.
La promesse, c'est d'utiliser les mêmes briques de base (les images Docker) depuis le poste de travail du développeur, jusque sur les serveurs de production d'une seule et même manière. Si on regarde du côté de Google Container Engine, on est dans une pure API Kubernetes et donc réutilisable sur des clusters on-premise, ou sur miniKube, l'installation Kubernetes dédiée aux installations sur poste de travail.
Ainsi, on tire parti de l'immense galaxie d'images Docker contribuées par la communauté, pour se faciliter l'installation de composants techniques divers, tout en ayant la souplesse de pouvoir produire les siennes en propre. On conserve aussi une seule et même technique de déploiement sur l'ensemble des environnements, poste de travail compris.
Pour installer son cluster Kubernetes sur son MacBook via miniKube, voici comment faire :

```
brew install docker-machine-driver-xhyve
sudo chown root:wheel $(brew --prefix)/opt/docker-machine-driver-xhyve/bin/docker-machine-driver-xhyve
sudo chmod u+s $(brew --prefix)/opt/docker-machine-driver-xhyve/bin/docker-machine-driver-xhyve
curl -Lo minikube https://storage.googleapis.com/minikube/releases/v0.11.0/minikube-darwin-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/
minikube start --vm-driver=xhyve
minikube dashboard
```

Pour déployer son application et la stack technique sur laquelle elle repose sur Google Container Engine (GKE), voici comment faire :

```
$ cat deployment.yml
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: my-app
spec:
 replicas: 3
 template:
  metadata:
   labels:
    app: my-app
  spec:
   containers:
   - name: my-app
    image: my-org/my-app:latest
    imagePullPolicy: Always
    resources:
     limits:
      cpu: 300m
      memory: 200Mi
     requests:
      cpu: 100m
      memory: 50Mi
$ kubectl apply -f deployment.yml
```

Quelques lignes de configuration et une ligne de déploiement plus tard, votre application tourne dans le Cloud, sur l'infrastructure de **Google**.
AWS propose sa propre implémentation de plateforme *CaaS* avec Elastic Container Service. Mais le service et la syntaxe de configuration et de déploiement restent propriétaire et donc moins intéressante pour l'usage de bout en bout du cycle projet.
De son côté Microsoft adosse sa plateforme *CaaS*, Azure Container Service à 2 orchestrateurs différents, au choix. Docker Swarm d'un côté, et Mesos/Marathon DC/OS de l'autre.
D'un côté Docker Swarm permet de produire très simplement des déploiements de bout en bout, depuis le laptop du développeur (qui n'aura besoin que de Docker for Windows ou Docker for Mac).
De l'autre côté, Mesos/Marathon est très solide et consitue un choix d'évidence pour orchestrer des containers sur une plateforme de production. Néanmoins, il s'avère difficile à installer sur un poste de développeur.

## 3 - Les outils as a service du monde Cloud

On l'a vu, qu'il s'agisse d'une solution PaaS générique, dédiée, orientée business, ou d'une solution agnostique CaaS, les choix sont variés et orientés par des questions de standardisation des procédures de déploiement, de souplesse ou de facilité d'utilisation.
Le développeur ne peut toutefois omettre une réalité importante du développement d'aujourd'hui : pour offrir une expérience utilisateur riche et cohérente, il va interfacer son service applicatif avec des services tiers proposant des fonctionnalités à haute valeur ajoutée directement disponibles dans le Cloud.
On peut citer le moteur de recherche full text de type Algolia, ou les services d'intelligence artificielle proposés par Google, comme la reconnaissance d'images (Vision API), l'interface conversationnelle (Natural Language API), la reconnaissance vocale (Speech API) ou la traduction instantanée (Translation API).
On peut aussi parler de services plus orientés vers l'opérationnel comme les outils d'analyse de logs ou de supervision comme Logmatic, Datadog ou NewRelic.
Dès lors, il faut considérer ces services comme faisant partie de la plateforme applicative choisie par le développeur. L'applicatif ne repose plus sur une seule et unique plateforme, mais sur un ensemble de services dans le Cloud, interconnectés entre eux.

## 4 - Celui qui porte la donnée porte votre infrastructure

Quoi qu'il en soit, dans un monde clairement SOLOMO (applications **SO**ciales, tirant partie de la géo **LO**calisation et disponibles sur **MO**biles), les applications et leurs utilisateurs produisent de plus en plus de données, dont l'application va tirer parti pour produire du contenu personnalisé, de la recommandation ou du *crowd-sourcing*.

Toute application Web grand public est candidate pour produire et consommer du BigData. Dès lors, choisir une plateforme revient essentiellement à choisir qui portera votre donnée. De ce point de vue là, chaque acteur du Cloud public propose ses solutions (essentiellement propriétaires et peu interopérables), aussi bien dans le monde noSQL que dans le monde SQL. Chez AWS, on trouve DynamoDB (base noSQL à la MongoDB), RedShift (base orientée colonnes) ou RDS (bases MySQL ou PostgreSQL). Chez Google, on trouve BigTable ou DataStore.

Bien sûr, dans le monde BigData, il convient de s'interroger sur l'architecture des traitements qui vont consommer cette donnée et sa colocalisation avec la donnée (pour optimiser les performances et réduire les latences et les coûts en bande passante).

## 5 - CONCLUSION

Au final, au-delà du choix de la plateforme portant la partie front-end de votre application, choix sur lequel le tableau ci-dessous vous offrira un éventail de critères de sélection, le choix essentiel revient à savoir où et comment est stockée et traitée la donnée massive produite par vos utilisateurs.

Dès lors, des critères plus spécifiques entrent en jeu comme :

- L'éthique du service face à la protection des données privées ;
- La localisation géographique de l'hébergement de ces données (en Europe, aux USA ou ailleurs) ;
- Le coût des services (en particulier de stockage des données et de bande passante) ;
- La disponibilité des services de stockage et de traitement du big data que le développeur aura choisi pour traiter sa BigData.

Avant de conclure cet article, je voudrais remercier tout particulièrement Théo Chamley et Cyril Bonté, qui ont fait un énorme travail de comparaison des différentes offres PaaS des acteurs du Cloud public. Et vous pointer un ultime outil pour faciliter vos choix sur la plateforme à retenir pour vos développements futurs : https://paasfinder.org/filter

## 6 - Tableau comparatif (liste non exhaustive)

| | AWS Elastic Beanstalk | Google Container Engine - GKE | Google App Engine - GAE | Azure App Service - AAS | Platform.sh |
|---|---|---|---|---|---|
| **Versatilité/ Technos dispo** | • Docker<br>• Python<br>• PHP<br>• NodeJS<br>• .NET<br>• Ruby/Rails<br>• Java | • Docker | Standard :<br>• Python<br>• Java<br>• PHP<br>• Go<br><br>Flexible :<br>• NodeJS<br>• Ruby<br>• Docker | • .NET<br>• NodeJS<br>• Python<br>• Java<br>• PHP | • PHP<br>- Drupal<br>- Symfony<br>- eZ<br>- Magento<br>- WordPress<br>- Typo 3<br>• NodeJS<br>• Ruby (beta)<br>• Python (beta) |
| **Efforts d'intégration CI/CD** | Grâce aux APIs très fournies d'AWS, automatiser la gestion de Beanstalk dans un pipeline d'intégration et de déploiement continus se fait relativement facilement. La CLI Beanstalk s'interface notamment avec Git pour faciliter la gestion de plusieurs environnements en fonction des branches. | GKE étant l'offre Kubernetes-as-a-Service de Google, le produit bénéficie de toutes les fonctionnalités standard de Kubernetes. Notamment, tout le déploiement d'une application est décrit par le biais de fichiers yaml. Il est donc relativement simple de générer de manière programmatique ces fichiers et de les déployer sur GKE, que ce soit pour de la production ou pour des environnements de tests. | Les CLI gcloud et appcfg.py (qui fait partie du SDK App Engine) permettent d'automatiser facilement toutes les actions d'administration, que ce soit le déploiement d'une nouvelle version, un rollback, etc. Comme GKE, les déploiements sont décrits sous la forme de fichiers yaml, facilement créés de manière programmatique. | Azure App Service s'intègre avec un certain nombre d'autres outils tels que Github ou Bitbucket pour faciliter la mise en place des pipelines d'intégration et déploiement continus. Il est également possible d'utiliser Azure comme remote git pour gérer ses déploiements. | Platform.sh fournit une API complète ainsi qu'un CLI pour automatiser le déploiement d'application.<br>La solution s'intègre à github et Bitbucket. Les projets/plateformes se gèrent sous git et utilisent les branches pour gérer différents environnements |
| **Services managés** | Beanstalk profite de la galaxie des services managés AWS : RDS (bases de données), ElastiCache (Memcached/Redis), ElasticSearch, DynamoDB, Cognito, etc. | GKE, comme App Engine, étant un produit de Google Cloud Platform, il profite des différents services managés de GCP : bases de données diverses, Google Cloud Storage, Google Container Registry, Google Natural Language API, etc. | GAE, tout comme GKE, profite de la galaxie des services managés Google Cloud Platform et notamment de ses puissantes APIs orientées Intelligence Artificielle. | AAS s'intègre avec les autres services managés de Microsoft Azure, notamment les bases de données, ou le service de cache Redis. | Elasticsearch<br>redis<br>RabbitMQ<br>MariaDB<br>PostGreSQL<br>Solr<br>mongoDB |
| **Orienté business vs. versatile** | Beanstalk est un produit versatile où l'on peut déployer tout type d'application | GKE est par nature un produit versatile qui est destiné à faire tourner tout type d'application. | GAE est un PaaS agnostique et n'est pas orienté vers un segment métier particulier. | AAS est un PaaS versatile sur lequel n'importe quel type d'application doit pouvoir tourner. | Initialement très orienté PHP, spécialisé sur certaines stacks. Intègre de nouvelles stacks, et ouvre les services à du PHP Custom.<br>Offre étendue à NodeJS, et en beta Ruby et Python. |

| | AWS Elastic Beanstalk | Google Container Engine - GKE | Google App Engine - GAE | Azure App Service - AAS | Platform.sh |
|---|---|---|---|---|---|
| **Simplicité d'emploi** | Au premier abord, Beanstalk est simple d'utilisation. Cependant, dès que l'on veut accéder à des fonctionnalités poussées (options de déploiements, utilisation de services managés, etc.), alors une vraie connaissance AWS est nécessaire pour réussir à gérer la solution proprement. | Kubernetes offre une couche d'abstraction très intéressante dans la gestion d'une application, qui permet de faire des choses très puissantes en terme sd'architecture logicielle et de management du cycle de vie. Cependant cette abstraction peut être compliquée à appréhender dans un premier temps, et il faut bien la comprendre pour faire les choses proprement. Une fois maitrisé, Kubernetes est un outil qui permet de réaliser des opérations de production complexes très facilement. | En tant que PaaS, le premier attrait de GAE est sa simplicité de mise en œuvre. Cependant, comme Beanstalk avec AWS, utiliser les différents services managés par GCP requiert une certaine expérience. | AAS est un PaaS et son principal argument de vente est la simplicité. | Prise en main rapide, création de compte simplifiée en utilisant un compte existant Bitbucket, GitHub ou Google. Création de plateforme guidée, mais la 1ère plateforme a quand même abouti à des erreurs (un template officiel sûrement trop jeune).<br>Les actions sont loggués et visible dans l'interface Platform.sh, le CLI est relativement bien documenté. |
| **Coût** | Beanstalk utilise des ressources AWS classiques (EC2, load-balancers) et ce sont celles-ci qui sont facturées (à l'heure). À titre indicatif, une machine de 2vCPU, 8Go de RAM coûte un peu moins de 100 $ par mois. | Un cluster GKE est composé de nodes et d'un master. Le master est gratuit si le cluster a 5 nodes ou moins et coûte à peu près 110 $ par mois au delà. Les nodes sont des machines virtuelles GCE classiques et sont facturées (à la minute) comme telles. À titre indicatif, une machine de 2vCPU, 8Go de RAM coûte un peu moins de 60 $ par mois. | Le prix de GAE dépend de beaucoup de paramètres différents et il est donc compliqué de donner des exemples pertinents. Pour plus d'information, rendez-vous sur la page pricing du service : https://cloud.google.com/appengine/pricing ou sur le pricing calculator : https://cloud.google.com/products/calculator/ | AAS propose plusieurs niveaux de service. Si les premiers sont gratuits ou presque, il faut compter un peu plus de 110 $ par mois pour une machine avec 2vCPU et 4Go de RAM dans le plan Basic. | 4 offres avec plusieurs déclinaisons:<br>1 - Développement, à partir de 10€/mois<br>2 - Professionnelle, à partir de 200€/mois (multi-environnements, multi-applications, MySql, Redis, Solr, ElasticSearch, Postgres, RabbitMQ)<br>3 - Entreprise, à partir de 760€/mois (SLA 99,99%, performance)<br>4 - Région dédiée, à partir de 6500€/mois (pour hébergement massif, multi-cloud, HA, ...) |
| **Portabilité/ Vendor lock-in** | Si le code de l'application déployée sur Beanstalk pourra être réutilisé, tout l'outillage qui entoure ce produit n'est pas réutilisable sur d'autres plateformes, Beanstalk restant une solution propriétaire AWS. | Kubernetes est un projet open-source et peut être déployé sur un large panel de solutions : GKE, AWS, Azure, OpenStack, bare-metal, etc. Tant que vous disposez d'un cluster Kubernetes, le run de vos applications sera quasiment identique quel que soit le provider d'infrastructure sous-jacent. | GAE est un produit propriétaire Google, et si le code déployé dans GAE pourra être réutilisé (pour peu qu'il ne dépende pas trop des APIs Google), tout l'outillage qui entoure ce produit ne sera pas réutilisable. | Tout comme Beanstalk, Azure App Service s'appuie sur une API de déploiement propriétaire. En cas de portage de l'application vers une autre plateforme, il faudra migrer l'outillage de déploiement associé. | Outillage spécifique Platform.sh mais pas de dépendance forte.<br>Propose la possibilité d'importer des applications existantes (non testé). |
| **Exploitabilité/ Souplesse** | Au premier abord, Beanstalk est une solution NoOps pure. Cependant, la réalité de la plupart des applications fait qu'une expérience opérationnelle AWS est préférable pour gérer proprement des applications importantes. Si un certain nombre de fonctionnalités avancées sont déjà intégrées dans la solution (notamment l'auto-scaling), leur gestion et leurs configurations requiert une certaine expertise. Beanstalk étant très fortement intégré à l'écosystème AWS, il faut tout de même noter que les autres produits AWS permettent assez facilement d'étendre les fonctionnalités de base (notamment en se basant sur le système de notification SNS ou les différentes offres de bases de données). | GKE (et Kubernetes en général) apporte une abstraction forte entre l'infrastructure et l'application. Dans le cas de GKE, il n'y a donc presque pas de gestion d'infrastructure à faire. Par contre, gérer des applications sur Kubernetes demande une certaine maîtrise de cette abstraction. Kubernetes supporte aussi la notion de "Third Party Resources" qui permettent d'étendre les fonctionnalités de base. Cela demande cependant une connaissance profonde du fonctionnement de Kubernetes. | Comme Beanstalk, GAE est pensé comme un produit NoOps. Comme Beanstalk, la dépendance des applications réelles sur un certain nombre de services externes et notamment des services managés Google exigent une certaine expérience.<br>Cependant, la palette complète des services fournis par GCP permet d'étendre de manière puissante les fonctionnalités de base de GAE. | Comme tout PaaS, AAS est un produit dont on est peu censé s'occuper. Les services managés associés peuvent eux nécessiter de l'administration. | Approche NoOps, la notion d'infrastructure est totalement masquée pour l'utilisateur (un service "système" est une dépendance comme une autre dans l'application). |

# Témoignage : choix d'un hébergement pour un petit éditeur de logiciels

Patrice Lamarche
*Leader Technique Société*
**Ars Data**

## Contexte

Je suis actuellement Leader Technique pour un petit éditeur de logiciels. Nous développons un CRM spécialisé à destination des établissements pédagogiques liés à l'art, telles que les écoles de musique, de danse et de théâtre. Nous sommes donc sur un petit secteur de niche, et notre clientèle est essentiellement composée de conservatoires et de collectivités publiques (collectivités d'agglomération, mairies, etc.). Notre petite infrastructure d'hébergement composée de deux frontaux Web, et d'un serveur SQL Server était vieillissante à la fois d'un point de vue logiciel et hardware. Nous avons fait le choix de quitter notre hébergeur local afin de bénéficier d'un meilleur service.

Comparé aux grands hébergeurs et fournisseurs de services Cloud, les hébergeurs locaux devraient en effet mettre en valeur leur proximité mais également des services à valeur ajoutée. Un accompagnement sur l'impact applicatif des différents changements techniques au niveau infrastructure serait le bienvenu pour des développeurs qui n'ont généralement que peu de compétences dans ce domaine. Par exemple, lors de la mise en place de Load Balancing, il serait intéressant d'avoir un accompagnement technique afin d'éviter toute erreur de configuration d'ASP.net, ou de gestion des sessions.

L'absence de ce type de service, et le décalage de coût avec des hébergeurs nationaux nous ont donc poussés à étudier la concurrence.

## Services d'hébergement classique ? Hébergement sur le Cloud ?

Nous souhaitions une bonne évolution technique sur la partie hardware et au niveau de la bande passante proposée, un système d'exploitation à jour, et le tout avec si possible une réduction de coût.

D'un point de vue charge, l'infrastructure actuelle suffit pour répondre aux sollicitations des utilisateurs. Une seule période de pic de charge est à constater lors de l'ouverture des périodes d'inscriptions mais cela ne nécessite pas d'infrastructure de débordement.

D'un point de vue technique, l'utilisation de services Cloud tels que de l'IaaS ou du PaaS n'est donc pas une nécessité. Nous pourrions opter pour une solution IaaS si le prix était inférieur à une offre d'hébergement classique qui est loin d'être le cas.

Les offres IaaS ciblent en effet une offre de serveurs ayant des configurations hardware limitées avec peu de cœurs, et peu de RAM, mais qui peuvent être scalées en nombre et réparties dans les régions du monde souhaitées. L'idée est en effet d'être capable d'héberger les différents services / modules de votre applications au sein de différentes VM déployables et scalables à la demande.

En ce qui concerne la société et les serveurs utilisés, nous souhaitions faire appel à une entreprise pérenne, montrant une bonne compréhension du marché afin que l'on puisse faire évoluer notre hébergement en fonction de nos besoins sans changer de fournisseur. Autre contrainte importante, notre clientèle étant essentiellement composée de collectivités publiques, certaines d'entre elles imposent dans leur appel d'offre d'héberger leurs données sur le territoire national.

Notre choix s'est porté sur l'hébergeur français OVH pour une raison simple : il s'agit de fournisseur qui propose le meilleur rapport qualité/prix au niveau de leur offre. Les configurations techniques proposées, la bande passante sont supérieurs à ce que peuvent proposer les concurrents et pour un prix bien moindre. D'un point de vue pérennité, pas de question particulière à se poser, la petite entreprise familiale est devenue un mastodonte d'envergure mondiale.

L'entreprise permet de sélectionner la localisation des serveurs souhaités, et il est donc tout à fait possible d'utiliser des serveurs localisés en France, notamment dans les datacenters de Roubaix ou Gravelines.

| Serveurs | Disque Dur | RAM |
|---|---|---|
| 2 frontaux Web | 40Go / 2To | 4Go / 32 Go |
| Serveur BDD SQL Server | 130Go / 500 Go SSD | 8Go / 64 Go |

Tableau : Changements hardware

En plus des changements décrits ci-dessus, nous sommes passés d'une bande passante garantie de 20Mb/S à 500Mb/s. Et le tout avec une réduction de 50% de notre facture !

## Evolutions possibles ?

Une étude des conteneurs Windows proposés par Windows Server 2016 est en cours afin de proposer une solution de déploiement unique pour notre offre SaaS et nos clients On Premise. Cette solution technique permettrait une réduction du coût de maintenance des clients On Premise, et une indépendance vis-à-vis de l'hébergeur choisi. •

# Azure Resource Manager
## Le nouveau modèle de déploiement Azure

• Thibaut Ranise
**Infinite Square**
http://blogs.infinitesquare.com/search?q=Thibaut+Ranise


*La plateforme Azure est en pleine évolution. On peut même parler à l'heure actuelle de « transition » car au-delà des fonctionnalités, c'est le fonctionnement interne de la plateforme qui évolue. En fin d'été 2014, Microsoft a annoncé la disponibilité en GA (General Availability) de son nouveau modèle de déploiement « Azure Resource Manager » (ARM). Ce nouveau modèle expose son lot de nouveautés et un ensemble de nouveaux paradigmes par rapport au modèle existant nommé « Azure Service Management » (ASM).*

### Deux modèles de déploiement pour une plateforme de Cloud

Historiquement la plateforme Windows Azure utilisait un seul et unique modèle de déploiement nommé "Azure Service Management" (ASM) pour provisionner des ressources dans le Cloud Microsoft. Lorsque le nouveau portail Azure est sorti en version « preview », nous avons découvert un ensemble de nouveautés qui n'étaient pas présentes sur l'ancien portail, notamment les « groupes de ressources » et la gestion des droits « RBAC » (role-base-access-control). Ces nouveautés ne sont pas de nouvelles fonctionnalités du modèles ASM mais de nouveaux paradigmes natifs au modèle de déploiement ARM. Les services de ces deux modèles de déploiement sont entièrement accessibles par API, l'ancien portail (manage.windowsazure.com) utilise l'API ASM alors que le nouveau portail permet d'utiliser les deux API ASM et ARM. Actuellement, la plateforme Azure est donc utilisable avec les deux modèles de déploiement. Tous les outils permettant d'administrer des ressources Azure sont utilisables dans deux modes ASM et ARM. Pour administrer des ressources Azure, Microsoft met à notre disposition un ensemble d'outils que l'on peut diviser en trois grandes familles :

- Les portails d'administrations ;
- Les outils en ligne de commandes ;
- Les SDK clients.

Ces outils utilisent de manières sous-jacentes les modèles de déploiement ASM et ARM par l'intermédiaire de leurs API respectives. Ils cachent la complexité des appels http vers les APIs par l'intermédiaires d'interfaces Web ou par l'utilisation de lignes de commandes.

Alors que pour les portails Web l'utilisation des modèles est tout à fait transparente, pour les outils en lignes de commandes (PowerShell et Azure Cli), il faut explicitement déclarer le modèle de déploiement à utiliser. Avec Azure Cli (Command Line Interface), l'outil en lignes de commandes nodeJS utilisable sur les systèmes Windows et Linux, l'outil utilise par défaut le modèle de déploiement ASM, il faut donc explicitement spécifier l'utilisable du modèle ARM : [Fig.1].

En PowerShell, il est de même nécessaire d'installer le module de base AzureRM et d'importer l'ensemble des sous-modules ou des modules particuliers : [Fig.2].

### « Infrastructure as Code » et template Azure

Une des nouveautés majeures de ce nouveau modèle de déploiement est la possibilité d'utiliser des templates de déploiement pour créer et mettre à jour un environnent Azure. Un template de déploiement également appelé template Azure ou template ARM est un fichier .json qui contient la définition de l'environnement à déployer, soit l'ensemble des ressources Azure nécessaires, mais pas uniquement ! Un template Azure n'est pas qu'une collection de ressources Azure, un template peut prendre en entrée des paramètres et il contient également des variables. De plus un ensemble de « fonctions » sont disponibles pour insérer la logique au sein du template. Certaines fonctions permettent par exemple de définir des dépendances entre les ressources définies, ainsi le déploiement va être ordonnancé de manière logique selon les besoins de l'infrastructure. Si aucune dépendance n'est définie entre les ressources, elles sont déployées en parallèle.

On peut diviser un template Azure en quatre parties distinctes :

- Les paramètres ;
- Les variables ;
- Les ressources Azure ;
- Les données de sorties. [Fig.3].

Selon l'infrastructure Azure à déployer, la mise en place d'un template ARM peut se révéler plus ou moins complexe, il est donc conseillé

```
C:\Users\ThibautRANISE>azure config mode arm
info:    Executing command config mode
info:    New mode is arm
info:    config mode command OK

C:\Users\ThibautRANISE>
```
1

```
#Installation du module ARM
Install-Module AzureRM

#Import de tous les modules AzureRM.*
Import-AzureRM

#Import d'un seul module ARM : "Compute"
Import AzureRM.Compute
```
2

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {},
    "variables": {},
    "resources": [],
    "outputs": {}
}
```
3

de se baser sur des templates existants. Une « banque » de template nommée « Azure QuickStart Template » est un repository Github qui met à disposition de tout un ensemble de templates prêt à l'emploi pour mettre en place des architectures IaaS et PaaS. Généralement le fichier .json de paramètres est utilisé conjointement au template principal, ce dernier permet de préciser des valeurs aux paramètres définis dans le template principal.

Pour mettre en place un template « From scratch », ou même pour en modifier un existant, un simple éditeur json suffit, puis une seule ligne de PowerShell est nécessaire pour déployer l'infrastructure définie dans un groupe de ressources. [Fig.4]

Visual studio propose cependant un type de projet spécifique pour la mise en place de template Azure. Ce type de projet génère automatiquement le script PowerShell de déploiement, et propose un éditeur json permettant d'ajouter et supprimer facilement des ressources Azure dans le template. [Fig.5].

Cette manière « déclarative » de créer son environnement est une bonne pratique, il permet notamment de stocker la définition de l'environnement Azure dans le gestionnaire de code source, au même titre que le code de l'application. De plus il permet de mettre en place des environnements paramétrables très rapidement !

## Les nouveaux paradigmes ARM

Le nouveau modèle de déploiement ARM apporte de nouveaux paradigmes qui donnent une réelle plus-value, à la plateforme Microsoft Azure.

### Groupes de ressources et « tags »

Organiser ses ressources Cloud est un élément essentiel pour avoir la main et gérer convenablement son infrastructure, avec l'ancien modèle de déploiement ASM. L'abonnement Azure faisait office de conteneur unique dans lequel l'ensemble des ressources (sites Web, base de données, Cloud services…) était regroupé. Cependant, il est souvent nécessaire d'organiser ses ressources Cloud d'une manière plus fine : par environnement (développement, test, production…), par client ou encore par développeur ! Les groupes de ressources sont des conteneurs « logiques » prévus à cet effet. Ces derniers permettent de regrouper des ressources Azure hébergées dans des régions différentes, ces ressources partagent alors un cycle de vie commun, elles sont déployables avec un template Azure et supprimables en une seule requête ! De plus il est possible de gérer les droits d'accès et d'administration sur ces groupes de ressources par l'intermédiaire de droits RBAC (role-base-access-control). Sur la figure 3, on retrouve les ressources déployées par l'intermédiaire du template Azure dans le groupe de ressources nommé « contact-apps-rg ». [Fig.6]. Ces groupes de ressources sont également présents sur les factures Azure, ce qui permet de connaître avec exactitude le coût d'un groupe de ressources par mois. Afin d'affiner la granularité de détails d'une facture Azure, il est de même possible d'attribuer des « tags » sur chaque ressource.

### Autorisations RBAC

Les autorisations RBAC donnent un nouveau souffle à la gestion des droits et autorisations sur les ressources et abonnements Azure. Avec le modèle ASM, seuls deux rôles étaient attribuables au niveau d'un abonnement Azure pour attribuer des droits d'accès : « Administrateur » et « Coadministrateur ». RBAC permet quant à lui d'affecter des rôles à tous les utilisateurs et groupes présents dans le répertoire Azure Active Directory de l'abonnement ainsi qu'a des utilisateurs possédant un compte Microsoft. Un role est par définition un ensemble de droits qui peut être affecté sur trois niveaux : l'abonnement, le groupe de ressources et la ressource Azure. Chaque rôle attribué à un niveau parent est de même valide sur les niveaux enfants. Il existe deux types de rôles, les prédéfinis par la plateforme Azure, et il est possible de créer des rôles personnalisés en PowerShell.

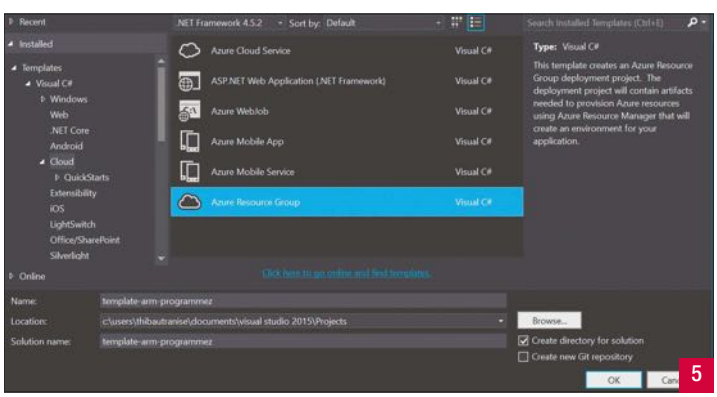
### Ressources Policies

Les rôles RBAC permettent d'attribuer des droits à certains utilisateurs et groupes, cependant il peut être utile de définir des règles globales à tous les utilisateurs d'un abonnement Azure ou d'un groupe de ressources ; c'est le rôle de « ressource policies ». Elles permettent la mise en place de systèmes globaux d'autorisation et de restrictions : « allow and deny system ». Leur mise en place se rapproche fortement de la création d'un template Azure, puisqu'une « ressource Policy » est un fichier .json dans lequel il faut définir un type d'actions à effectuer « Deny », « Append » ou « Audit » selon une condition définie.

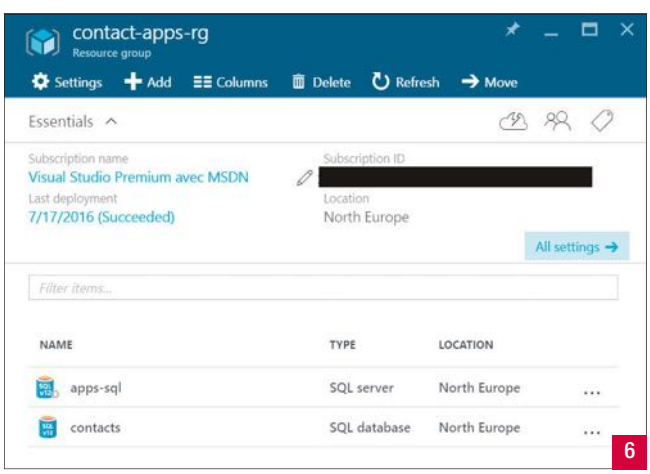
*Le modèle ARM devient petit à petit le nouveau « cœur » de Microsoft Azure. Ses nouveaux paradigmes apportent plus de souplesse dans la gestion des ressources Azure et des droits d'accès à ces dernières. Il est par ailleurs conseillé de migrer ses ressources déployées avec ASM vers ARM, puisque ARM va devenir progressivement l'unique modèle de déploiement de la plateforme Microsoft Azure.* •

```
$TemplateFile = 'C:\programmez\WebSiteSQLDatabase.json'
$TemplateParametersFile = 'C:\programmez\WebSiteSQLDatabase.parameters.json'
$ResourceGroupName = 'programmez-rg'
$ResourceGroupLocation = 'North Europe'

# Connexion à l'abonnement Azure
Login-AzureRmAccount

# Creation d'un groupe de ressource
New-AzureRmResourceGroup -Name $ResourceGroupName -Location $ResourceGroupLocation -Verbose -Force

# Déploiement des ressources définies dans un template de déploiement ARM
New-AzureRmResourceGroupDeployment -Name ('deploy_' + '-' + ((Get-Date).ToUniversalTime()).ToString('MMdd-HHmm'))
                                   -ResourceGroupName $ResourceGroupName `
                                   -TemplateFile $TemplateFile `
                                   -TemplateParameterFile $TemplateParametersFile `
                                   -Force -Verbose
```

4




5




6

# Azure Functions
## Des webjobs as a service

• Thibaut Ranise et Maxime Caroul
**Infinite Square**
http://blogs.infinitesquare.com/


*Une des annonces Azure de la Build 2016 est la mise à disposition d'un nouveau service nommé « Azure Functions ». Au même titre que les webjobs Azure, les fonctions Azure permettent d'exécuter des tâches en arrière-plan. Nous allons découvrir dans cet article les spécificités de ce nouveau service.*

### Exécuter des tâches en arrière-plan dans Azure

Avant l'arrivée des fonctions Azure, il n'existait qu'une seule et unique façon d'exécuter des tâches en arrière-plan dans Azure : les webjobs. Ces deux services sont intimement liés car les fonctions Azure sont basées sur le webjob sdk ; c'est donc le même « runtime » qui est utilisé pour ces deux services. La quasi-totalité des concepts utilisés par les webjobs sont donc présents dans les fonctions Azure.

### Le webjobs, les fondations des fonctions Azure

Il y a maintenant plusieurs années, Microsoft ajoutait une nouvelle extension à la brique « App Service » en mettant à disposition les « webjobs ». Un webjob est par définition une tâche en arrière-plan qui s'exécute dans le contexte d'une application Web Azure. Un webjob s'exécute sur le même plan de service que l'application Web qui l'héberge. Ce peut être un script (PowerShell, bash, python, php) ou un programme (.exe, .jar).

On distingue trois types de webjobs :

- « Schedule » : exécution programmée dans le temps ;
- « On demand » : exécution à la demande ;
- « Continous » : exécution continue.

Le déploiement d'un webjob est une phase importante du cycle de ce dernier, car il faut le déployer de manière explicite au côté d'un site Web Azure, dans un dossier nommé : /site/wwwroot/App_Data/Jobs. Il existe plusieurs façons de déployer son webjob : FTP, Web Deploy et Kudu. Avec un webjob, le développeur est responsable du déploiement et de la mise à l'échelle de ce dernier. Ces deux phases demandent donc du temps, c'est sur ces deux points que les fonctions Azure apportent des nouveautés.

### Azure Functions

#### *Focus sur le code*

Comme son nom l'indique, « Azure Functions » propose une solution pour exécuter du code (une fonction !) dans le Cloud Microsoft. Avec ce service, Microsoft met à notre disposition « des WebJobs as a service » ; c'est une couche d'abstraction sur le webjob sdk qui permet au développeur de produire uniquement la logique métier nécessaire à sa problématique métier et de l'exécuter dans Azure sans se préoccuper de la phase de déploiement du webJob. L'idée principale est de produire du code exécutable dans Azure sans se soucier de l'infrastructure. Tout comme les webjobs, il est possible de rédiger des fonctions Azure dans plusieurs langages : C#, Node.js, Python, F#, PHP, Java, Batch et bash.

Ces fonctions sont facilement utilisables dans des architectures micro-services pour effectuer des tâches longues et répétitives comme le traitement des images, les calculs, les purges de base de données ou encore des tâches de fond comme l'envoi massif d'emails. Ces tâches peuvent être soit programmées dans le temps ou exécutées à la demande.

#### *Scénarios logiques et chaînage*

Avec Azure Function, il devient simple de faire transiter des messages et du contenu entre plusieurs services Azure, sans problématique de déploiement. Il est tout à fait possible de créer plusieurs fonctions (dans des langages différents !) qui vont posséder des responsabilités différentes mais qui vont travailler ensemble en faisant transiter de la donnée entre elles et / ou en provoquant la levée d'événements qui feront s'exécuter les autres fonctions : insertion d'un blob, requête http, ajout d'un message dans un service bus ou dans une Storage Queue… Par exemple, nous pourrions créer deux fonctions :

- MyFunction1.js : cette fonction « écoute » un blob storage, effectue un traitement sur le blob puis envoie un message contenant le nom du blob dans une queue nommée « MyQueue ».
- MyFunction2.csx : cette fonction « écoute » les messages de la queue « MyQueue » et envoie ce message dans une Azure Table nommée « MyTable »…

#### *Un coût à l'exécution*

Le coût d'une fonction Azure s'exécutant dans le contexte d'un site Web Azure va être relatif au plan de service défini pour le site Web. Dans un plan de service « classic », le site Web et la fonction Azure vont s'exécuter sur une machine virtuelle dédiée. L'exécution de la fonction sera donc limitée aux capacités de la machine virtuelle (Mémoire / CPU) et le prix sera relatif au plan de service choisi (Basic, Standard, Prenium). Que la / les fonctions Azure s'exécutent ou non, le prix du plan de service sera fixe par mois. De plus selon la charge requise, il peut être nécessaire de mettre à l'échelle ce plan de service.

Avec les fonctions Azure, un nouveau type de plan de service « dynamique » est apparu : « Pricing tier » dans lequel la fonction va s'exécuter sur une / plusieurs instance(s) « dynamiques » qui ne seront actives que lorsque la fonction sera exécutée. C'est un mode « pay-per-use » où la consommation correspond uniquement au temps d'exécution de la /des fonction(s) multiplié par la mémoire configurée sur le plan de service (Gigabit /s). Avec ce nouveau plan de service, les applications Azure Function sont complètement autonomes et se mettent à l'échelle automatiquement selon le besoin, sans engendrer de coûts lorsque les fonctions ne sont pas utilisées. Cette mise à l'échelle automatique peut lancer jusqu'à 10 instances en parallèle.

### Fonctionnement interne (web app, bindings, triggers…)

#### *Architecture*

Pour créer une fonction Azure, il est nécessaire de créer une « Azure Function App ». Une « Azure Function App » est un conteneur de fonctions. En arrière-plan, une « Azure Function App » crée un compte de stockage ainsi qu'un

site Web Azure qui va héberger les fonctions.
La création d'une Azure Function App « test-dev-functions » va donc créer un site Web Azure : test-dev-functions.azurewebistes.com. Ce site Web est accessible depuis le portail Azure, il est donc possible d'agir sur sa configuration (AppSettings).
Toutes les fonctions présentes dans leur conteneur ont besoin d'une capacité de mémoire pour pouvoir s'exécuter. Avec un service plan classique, la mémoire utilisable sera relative à la capacité en mémoire de la machine virtuelle sous-jacente. Mais si l'on opte pour l'utilisation d'un service plan dynamique, il est nécessaire de configurer au niveau du service plan la mémoire qui sera utilisée par l'ensemble des fonctions du conteneur. Si la mémoire utilisée par les fonctions est plus importante que la mémoire configurée, l'application s'arrête et plus aucune des fonctions ne sera donc exécutée. [Fig.1].

### *« Binding » et « Triggers »*

Chaque fonction utilise un événement provenant d'une ressource externe pour déclencher sa propre exécution, ce sont les « triggers ». Chaque fonction peut prendre en entrée des données provenant de l'exécution du trigger ou de sources externes, ce sont les « input ». Enfin elles sont capables d'écrire des données dans des sources de stockage externes, ce sont les « outputs ». Une fonction va donc être liée à un trigger et des inputs et potentiellement des outputs, ces liaisons sont les "bindings" de la fonction [Fig.2].
Ces "bindings" sont définis dans un fichier de configuration nommé « function.json » éditable directement par interface graphique.

## Comment créer sa première Azure Function ?

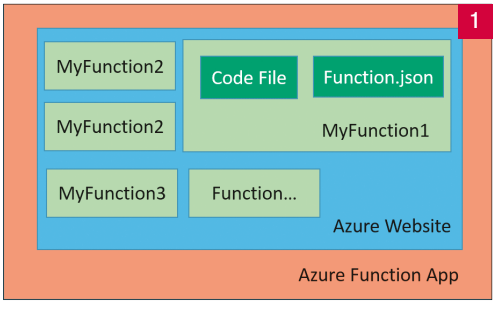
Actuellement, il existe deux points d'entrée pour créer sa « function App » :

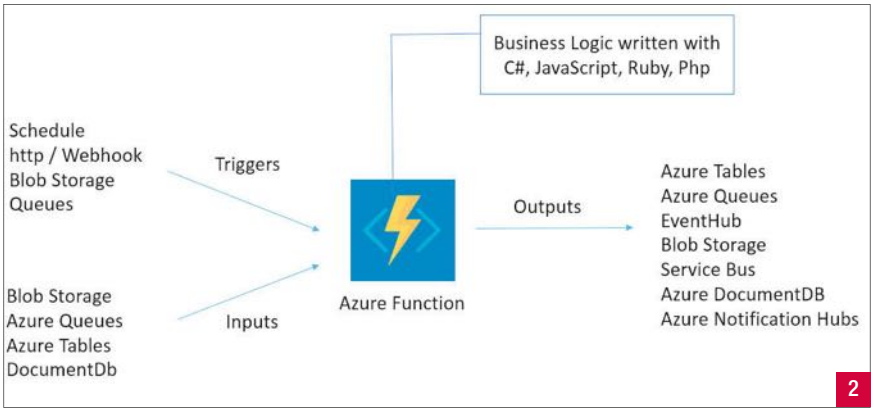
- Le portail Azure : portal.azure.com ;
- Le portail : functions.azure.com.

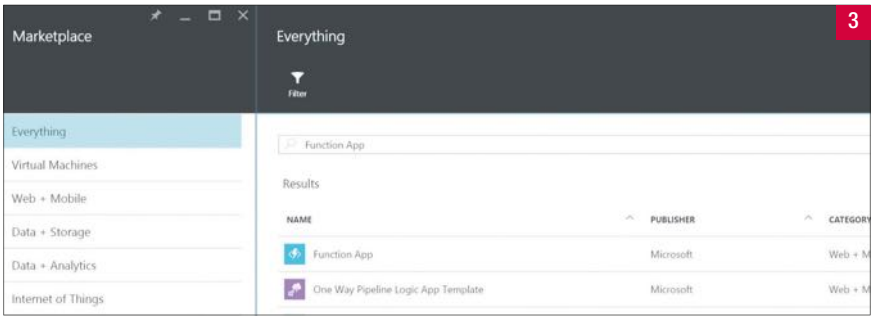
Le portail « Azure Functions » permet de créer en quelques clics son « Azure functions App », il suffit de se connecter avec ses identifiants Azure et de donner un nom à son application ; par défaut un « dynamic service plan » sera utilisé. Depuis le portail Azure, il est nécessaire de passer par le market place, rechercher « function app » comme dans la [Fig.3].
Il est ensuite nécessaire de configurer l'application en choisissant un service plan classique ou dynamique, un compte de stockage et un groupe de ressources.
Une fois l'application créée, un tableau de bord nous permet de manager nos fonctions ainsi que la Web App. [Fig.4].
Pour chaque fonction 4 onglets sont disponibles :

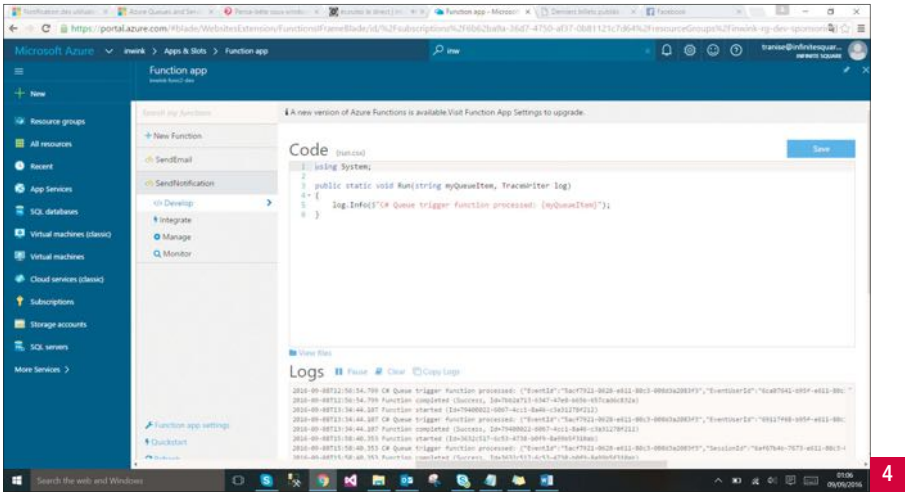
- « Develop » : interface permettant de définir le code de la fonction ;
- « Integrate » : permet de configurer les « bindings » ;
- « Manage » : permet d'activer / désactiver / supprimer la fonction ;
- « Monitor » : interface permettant de vérifier le bon fonctionnement de la fonction.

*Les fonctions Azure apportent une alternative intéressante aux webjobs car elles permettent de gagner du temps en se concentrant sur le code métier, sans se soucier de l'infrastructure sous-jacente ! De plus, le nouveau type de plan de service « Dynamique » permet de faire des économies et de ne plus se soucier des problématiques de mise à l'échelle.* •



MODÉLISATION DES DONNÉES
2e partie

# Générer un classeur Google Sheets à partir d'un modèle

Steve Berberat
*Collaborateur scientifique*
**Haute école de gestion Arc, HES-SO**
steve.berberat@he-arc.ch

Pierre-André Sunier
*Professeur HES*
**Haute école de gestion Arc, HES-SO**
pierre-andre.sunier@he-arc.ch

*Dans toute organisation, les données gérées numériquement sont de plus en plus nombreuses. Les logiciels qui permettent de les gérer peuvent être de simples outils bureautiques comme les tableurs, des progiciels de gestion comme les ERP et les CRM ou les applications développées sur mesure pour des besoins spécifiques.*

Les progiciels et les applications sur mesure ont un coût certain. De fait, le service informatique et les développeurs de l'organisation ne peuvent pas proposer des solutions pour répondre à tous les besoins des collaborateurs. Ces derniers comblent alors leurs manques en s'aidant souvent des outils bureautiques qu'ils connaissent : Microsoft Excel par exemple, le plus souvent installé sur leur propre ordinateur.
La gestion de données par un tableur installé en local pose toutefois un certain nombre de problèmes : le fichier peut être corrompu, plusieurs personnes ne peuvent pas l'éditer au même moment, les données qui s'y trouvent sont souvent redondantes, le manque de contrôles rendent les erreurs de saisie possibles, l'intégrité des données n'est pas garantie.
N'existe-t-il pas une solution intermédiaire, qui permettrait au service informatique et aux développeurs de déployer une solution rapidement et très peu coûteuse, mais en limitant ces différents inconvénients ? Oui, cela existe !
Nous allons vous présenter une méthode qui consiste à automatiser la création de feuilles de calculs de type bureautique auxquelles sont ajoutés des contrôles qui assurent l'intégrité des données. Cela est possible en utilisant de simples modèles de données ainsi qu'un automate. Nous illustrerons cette méthode en utilisant un automate capable de générer un classeur en ligne Google Sheets à partir d'un modèle conceptuel de données.

## Les feuilles de calculs avec macros

Les feuilles de calculs sont très permissives et permettent, par défaut, de saisir tout type d'information, sans contrôle ou validation particulière. Dans un tel contexte, la qualité des données n'est pas assurée. Si nous prenons l'exemple d'une feuille qui permet de saisir une liste d'employés, avec comme colonnes le sigle de l'employé, son prénom et son nom, nous pouvons très bien saisir les informations présentées en Figure 1.
Dans cette liste, deux employés ont le même sigle « STB », alors que ce sigle devrait être unique.
De plus, Léo n'a pas de nom. Ces problèmes sont très typiques d'une utilisation de feuilles de calculs pour gérer des données. La plupart des éditeurs de feuilles de calculs nous permettent de programmer des contrôles. C'est avec leur aide qu'il devient possible d'assurer une certaine qualité des données. Avec Excel, par exemple, des macros peuvent être écrites de façon à alerter l'utilisateur qui ne saisit pas la donnée d'une colonne obligatoire (le nom de l'employé dans notre exemple) ou qui saisit une donnée devant être unique et qui existe déjà (le sigle de l'employé en l'occurrence).
Ces contrôles constituent le premier élément nécessaire à la solution que nous vous présentons ici.

| Sigle | Prénom | Nom |
|---|---|---|
| STB | Steve | Berberat |
| PAS | Pierre-André | Sunier |
| ALD | Albert | Dubois |
| LEM | Léo | |
| ADD | Adam | Durand |
| STB | Steve | Benoit |

1

*Données de mauvaise qualité*

## Les feuilles de calcul dans le Cloud

Les macros d'une feuille de calcul peuvent améliorer la qualité des données, mais d'autres problèmes persistent avec une solution de type Excel : plusieurs utilisateurs ne peuvent pas éditer le fichier en même temps, ce dernier peut être corrompu si une erreur survient durant sa sauvegarde, les données ne sont pas facilement accessibles en dehors du réseau à moins d'en réaliser une copie…
Aujourd'hui, des solutions de stockage et d'édition de feuilles de calcul dans le Cloud existent. Ces solutions résolvent les problèmes cités ! Ils permettent une édition collaborative et simultanée entre plusieurs personnes, un accès depuis n'importe où à l'aide d'un navigateur et une gestion des droits en lecture et écriture pour chaque classeur, voire pour chaque feuille de calcul.
Parmi les plus connus, nous trouvons Google Sheets, Microsoft Excel Online et Zoho Docs. Chacun d'eux prend en charge la programmation de « macros » et permet de mettre en place des contrôles automatiques

sur les données saisies. La Figure 2 illustre une feuille de calcul Google Sheets éditée par deux personnes simultanément.
L'externalisation de données dans le Cloud pose toutefois la question de la sécurité et nécessite bien entendu une réflexion. Si des données sont sensibles, le Cloud privé peut être une solution. Microsoft, par exemple, propose des solutions permettant d'héberger son propre serveur Office Online.

## Modéliser un classeur

La feuille de calcul en ligne, complétée par des macros qui assurent une certaine cohérence des données, est une solution intéressante. Cette solution n'est toutefois pas très productive en soi si l'on doit créer le classeur soi-même et programmer les macros.

### Correspondance entre un MCD et un classeur

La force de la méthode que nous vous proposons est de vous éviter de créer vous-même le classeur et les macros mais, au lieu de ça, de le modéliser pour ensuite le générer automatiquement à partir du modèle. Comment modéliser un classeur ? Cela se fait aisément à l'aide d'un modèle conceptuel de données (MCD). Les règles suivantes expliquent le principe :

- Une entité du MCD correspond à une feuille de calcul ;
- Un attribut de l'entité correspond à une colonne de la feuille ;
- Une association de degré 1-n correspond à une liste déroulante ;
- Une association de degré n-n correspond à une feuille de calcul incluant 2 listes déroulantes.

### MCD d'illustration

La Figure 3 est un exemple de MCD qui illustre la gestion d'un référentiel de compétences. Pour l'entreprise qui ne dispose pas d'un outil de gestion des RH, utiliser un classeur en ligne, agrémenté de contrôles sur les données, est une solution attrayante pour référencer les compétences de ses employés !
Dans ce MCD, réalisé à l'aide d'un diagramme de classes UML, nous retrouvons les employés, les compétences ainsi que l'affectation des compétences aux employés. Une compétence est détenue par un employé avec un éventuel niveau de compétence.

### Contraintes et règles incluses sur le MCD

Le modèle illustré en Figure 3 comprend, en plus des entités, attributs et associations, les contraintes et règles métier décrites en Figure 4.

4

| Contraintes et règles | Signification |
|---|---|
| <<UID-1>> sigle | Le sigle permet d'identifier de façon unique l'employé. |
| <<M>> prenom | Le prénom de l'employé est obligatoire (Mandatory) |
| <<M>> nom | Le nom de l'employé est obligatoire (Mandatory) |
| <<UID-1>> acronyme | L'acronyme permet d'identifier de façon unique la compétence. |
| <<U-1>> libelle | Le libellé de la compétence est unique mais n'est pas obligatoire. |
| <<UID-1>> nom | Le nom donné à un niveau de compétence doit être unique. |
| {ordered} | Les niveaux de compétences sont définis selon un ordre précis. |

*Contraintes et règles incluses sur le MCD*

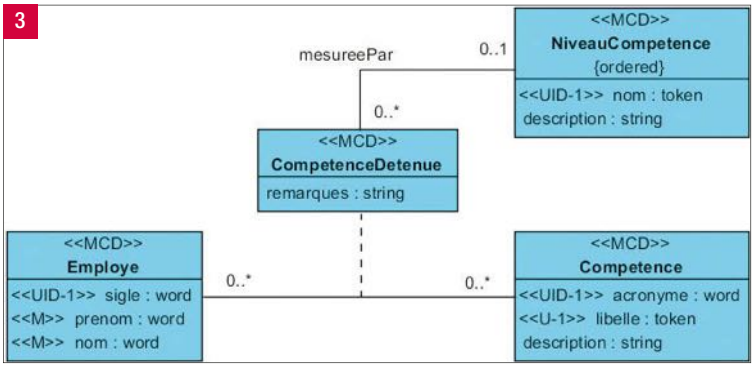
Le MCD est suffisamment précis pour être transformé automatiquement en un classeur de type tableur. Alors que les entités, attributs et associations peuvent être transformés en feuilles de calcul, colonnes et listes déroulantes à l'aide d'un automate. Les contraintes et règles peuvent, elles, être transformées en macros ! Par exemple, le stéréotype <<UID-1>> sur le sigle de l'employé peut donner lieu à une macro qui va vérifier qu'à chaque insertion ou mise à jour d'un employé, aucun sigle portant le même nom n'existe déjà pour un autre employé.

| Sigle | Prenom | Nom |
|---|---|---|
| STB | C4 e | Berberat |
| PSU | Pierre-André | Sunier |
| ALD | Albert | Dubois |
| LEM | Lé Christophe Francillon | |
| ADD | Adam | Durand |
| STB | Steve | Benoit |

2

*Édition simultanée dans Google Sheets*

3




*Modèle conceptuel de données*

5

| | A | B | C |
|---|---|---|---|
| 1 | Acronyme | Libelle | Description |
| 2 | BPM | Business Process Management | |
| 3 | BPMN | Business Process Modeling Notation | |
| 4 | UML | Unified Modeling Language | |
| 5 | MCD | Modélisation Conceptuelle des Données | |
| 6 | | | |

Ajoutez 1000 lignes de plus en bas.

Competence | NiveauCompetence | Employe | CompetenceDetenue

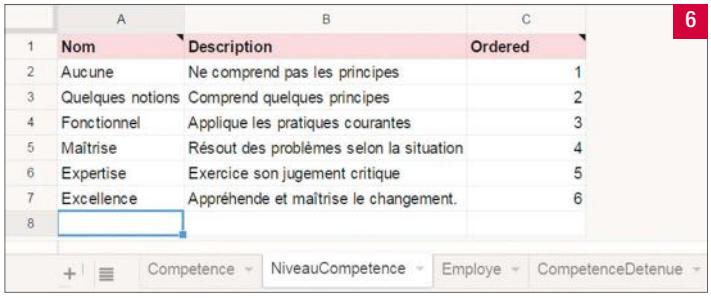
*Feuille de saisie des compétences*

## Générer un classeur Google Sheets

À la Haute école de gestion Arc, nous avons développé un automate capable de transformer un modèle conceptuel de données en classeur Google Sheets. Nous illustrons ici le classeur Google obtenu après exécution de cet automate à partir du modèle de la Figure 3. La partie « L'automate de transformation MVC-CD » vous apportera plus de détails sur cet automate expérimental libre en téléchargement.

### Génération des feuilles de calculs

Nous paramétrons l'automate en lui indiquant le compte Google Drive vers lequel il pourra créer le classeur, puis lançons son exécution. À la fin de la procédure, c'est un classeur comprenant 4 feuilles de calculs qui apparaît : « Competence », « NiveauCompetence », « Employe » et « CompetenceDetenue ». Chaque feuille est accessible à l'aide de son onglet dédié comme le montre la Figure 5. Il nous est dès lors possible

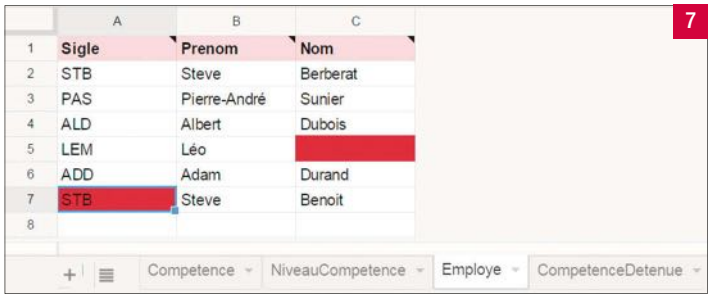


| | A | B | C |
|---|---|---|---|
| 1 | Nom | Description | Ordered |
| 2 | Aucune | Ne comprend pas les principes | 1 |
| 3 | Quelques notions | Comprend quelques principes | 2 |
| 4 | Fonctionnel | Applique les pratiques courantes | 3 |
| 5 | Maîtrise | Résout des problèmes selon la situation | 4 |
| 6 | Expertise | Exercice son jugement critique | 5 |
| 7 | Excellence | Appréhende et maîtrise le changement. | 6 |
| 8 | | | |

Competence | NiveauCompetence | Employe | CompetenceDetenue

6

*Feuille de saisie des niveaux de compétences*



| | A | B | C |
|---|---|---|---|
| 1 | Sigle | Prenom | Nom |
| 2 | STB | Steve | Berberat |
| 3 | PAS | Pierre-André | Sunier |
| 4 | ALD | Albert | Dubois |
| 5 | LEM | Léo | |
| 6 | ADD | Adam | Durand |
| 7 | STB | Steve | Benoit |
| 8 | | | |

Competence | NiveauCompetence | Employe | CompetenceDetenue

7

*Feuille de saisie des employés*

d'utiliser la solution et de saisir nos données. Nous saisissons par exemple les compétences possibles.

La feuille de calcul présentée en Figure 6 nous permet de saisir les niveaux de compétences. Nous remarquons une colonne « Ordered » qui a été créée automatiquement et qui permet de saisir un numéro d'ordre. Elle provient de la contrainte {ordered} spécifiée sur le modèle initial de la Figure 3.

## Contrôles de qualité des données

Les contraintes et règles métier indiquées sur le MCD ont été prises en compte et transformées en méthodes Google Apps Script qui contrôlent automatiquement les données saisies et mises à jour. En Figure 7, nous saisissons volontairement un employé sans nom de famille et un autre portant un sigle déjà existant. La feuille Google nous avertit des erreurs en coloriant les cellules en rouge.

La vérification de l'unicité du sigle a été mise en place par l'automate puisque le sigle a été défini comme identifiant unique à l'aide du stéréotype <<UID-1>> sur l'attribut « sigle » du MCD initial.

Il en est de même pour le prénom et le nom qui sont rendus obligatoires suite à la spécification du stéréotype <<M>> sur les attributs respectifs. Voici un extrait simplifié du code de programmation Google Apps Script que l'automate a généré automatiquement pour contrôler l'unicité du sigle :

```
function testUnique(activeCell){
 var activeSheet = application.getActiveSheet();
 //Get cells of active cell column
 var cellsToTest = activeSheet.getRange(2,cell.getColumn(),activeSheet.getLastRow());
 var valueAlreadyExist = false;
 for (i = 1; i < activeSheet.getLastRow(); i++){
  var cellToTest = cellsToTest.getCell(i,1);
  if (cellToTest.getValue() == activeCell.getValue() && cellToTest.getValue()!= ""){
   valueAlreadyExist = true;
  }
 }
```

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Remarques | Employe | Competence | NiveauCompetence |
| 2 | Formation de 150h à la HES-SO | STB | BPM | Fonctionnel |
| 3 | Formation de 120h à la HEG | STB | BPMN | Maîtrise |
| 4 | Enseigne le cours UML à la HEG | PAS | UML | Maîtrise |
| 5 | Professeur dans le domaine depuis 20 ans | PAS | MCD | Excellence |
| 6 | | | | |

Ajoutez 1000 lignes de plus en bas.

Excellence
Aucune
Quelques notions
Fonctionnel
Maîtrise
Expertise

8 Competence | NiveauCompetence | Employe | CompetenceDetenue

*Feuille de saisie des compétences détenues par les employés*

```
 if(valueAlreadyExist){
  cell.setBackground('White');
 }
 else{
  cell.setBackground('Red');
 }
}
```

Bien qu'il puisse être optimisé, ce code prouve le concept de génération automatique de contrôles sur les données saisies. La fonction testUnique() est appelée dès que la valeur d'une cellule est modifiée. Elle parcourt toutes les cellules de la même colonne et vérifie si l'une d'entre elles contient la même valeur que la cellule modifiée. Si c'est le cas, cette dernière est coloriée en rouge.

## Listes déroulantes

L'entité associative « CompetenceDetenue », qui permet l'affectation des compétences aux employés, a été transformée en une feuille de calcul comme montré à la Figure 8. Sur chaque ligne de la feuille, une première liste déroulante permet de sélectionner l'employé, une seconde permet la sélection de la compétence qu'il détient, et une troisième fait référence au niveau de compétence.

Les listes déroulantes réfèrent les données identifiées de manière unique dans les autres feuilles. Cette mécanique met en œuvre les relations des données entre elles et permet ainsi d'éviter la redondance ! Elle n'est pas aussi contraignante que la Foreign Key connue dans les bases de données relationnelles, mais elle offre une alternative louable et utilisable.

Remarquez également que la liste des niveaux de compétences s'affiche en fonction de l'ordre défini dans la feuille en Figure 6. Cet ordonnancement a pu être automatisé grâce à la contrainte {ordered} placée sur l'entité « NiveauCompetence » du MCD.

Cette illustration amène le constat que, à l'aide d'un outil de transformation du MCD en un classeur Google Sheets, les développeurs peuvent mettre en place une solution simple, qui assure un certain contrôle qualité sur les données, très rapidement et avec peu de coûts.

# L'automate de transformation MVC-CD

L'automate qui a été utilisé pour illustrer la transformation du MCD vers Google Sheets a été développé dans le cadre d'un projet intitulé « MVC-CD ». Le nom du projet a notamment été repris et donné à l'automate. Ce genre d'automate pouvant être développé au sein d'une équipe de développeurs dans le but d'améliorer la productivité, nous présentons ici son principe de fonctionnement et son contexte d'utilisation.

## Principe de fonctionnement

L'automate est capable de générer une base de données relationnelle Oracle embarquant les triggers et procédures qui assurent le respect des contraintes et règles métier décrites sur le MCD. La transformation à partir du MCD se fait en passant par le modèle logique de données relationnel (MLD-R), qui est indépendant du constructeur et qui permet une transformation pour d'autres constructeurs, comme Microsoft SQL Server. Ce cheminement est représenté par les éléments bleus de la Figure 9. Si cet aspect vous intéresse, sachez qu'il a fait l'objet d'un précédent article publié dans Programmez.

Le principe de transformation de modèles est le même pour Google Sheets. Un modèle logique intermédiaire, qui est indépendant du constructeur a également été mis en place. Nous l'avons nommé « Modèle logique de données Tableur » (MLD-T).

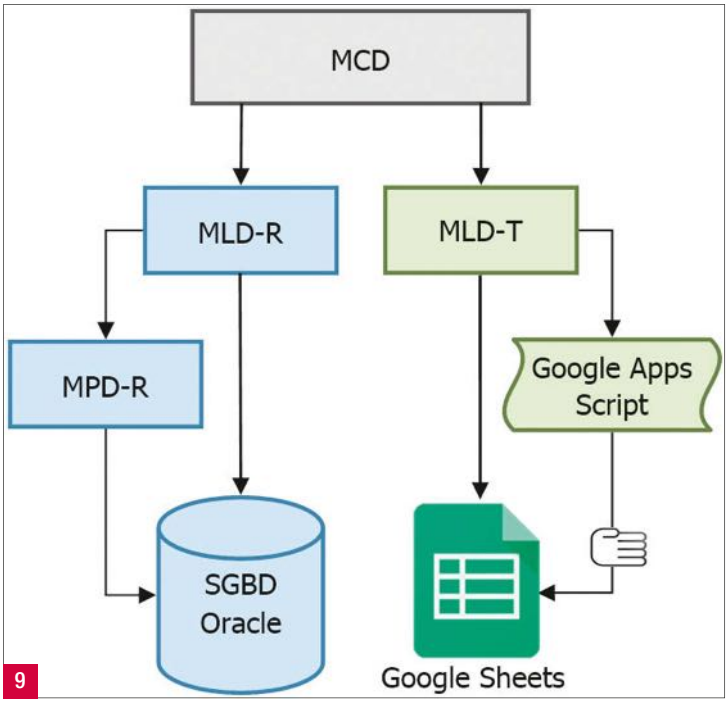
Il permettra, par la suite, de prendre en charge la transformation vers d'autres types de classeurs, par exemple vers Excel Online.

Sur la Figure 9, les éléments de couleur verte présentent les étapes de la transformation vers Google Sheets effectué par l'automate.

La création du classeur et des feuilles de calculs se fait automatiquement par l'automate au travers de l'API Google Drive.

Les scripts Google Apps de contrôle des données sont créés par l'automate mais ne sont pas déployés automatiquement, l'API ne permettant pas cette possibilité actuellement.

Une copie du script doit être faite manuellement dans le classeur Google généré. L'opération reste toutefois rapide et le temps consacré est dérisoire comparé à celui qui serait nécessaire si le script devait être développé à la main !



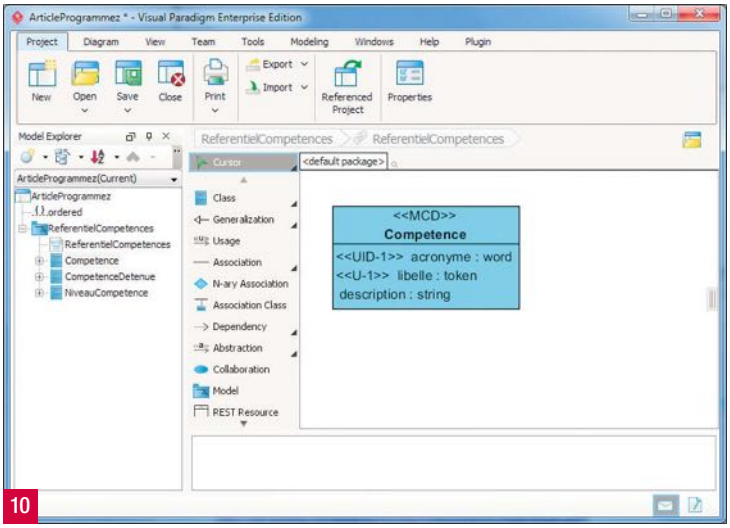

*Fonctionnement de l'automate MVC-CD*

## Visual Paradigm comme outil de modélisation

Le développement de l'automate consistait surtout à écrire les règles de transformation. Nous n'avons ni développé la gestion d'un référentiel ni la gestion de l'interface graphique.

À la place, nous nous sommes appuyés sur un outil de modélisation existant, Visual Paradigm, auquel nous y avons adjoint un plug-in. Grâce à l'API de cet outil, nous avons pu profiter entièrement de son interface graphique et de ses fonctionnalités.

Cette stratégie nous a permis de nous concentrer sur l'essentiel et d'être ainsi plus rapides dans le développement de l'automate. En Figure 10 vous trouverez une capture-écran de l'interface de dessin de Visual Paradigm.




*Visual Paradigm*

## Téléchargement du plug-in

Le plug-in pour Visual Paradigm que nous avons développé est téléchargeable librement. Un manuel d'utilisation, qui vous permet de réaliser vous-même une transformation d'un MCD vers Google Sheets, est également disponible. Vous trouverez toutes les informations nécessaires dans les références de cet article.

# CONCLUSION

Au travers de cet article, vous avez pu voir qu'il existe un compromis intéressant entre les applications de gestion développées sur mesure et l'utilisation de tableurs comme Microsoft Excel pour gérer des données. En utilisant un automate tel que « MVC-CD », vous pouvez générer des classeurs Google Sheets à partir d'un modèle conceptuel.

Ces classeurs utilisent les listes déroulantes pour effectuer le référencement entre données et comportent des scripts vérifiant que les données saisies respectent les contraintes et règles décrites sur votre modèle. Vous évitez ainsi la redondance et améliorez fortement la qualité des données !

En développant son propre automate, une équipe de développement peut accroître sa productivité et ainsi proposer des solutions sur mesure qui sont rapidement mises en place et très peu coûteuses.

Cette solution est probablement une meilleure alternative que celle consistant à laisser les collaborateurs créer eux-mêmes leurs propres feuilles de calcul en local. •

## Références

Projet MVC-CD et téléchargement de l'automate de transformation : http://lgl.isnetne.ch/Sagex35793/index.htm

Guide de génération d'un classeur Google Sheets à partir d'un MCD : http://lgl.isnetne.ch/Sagex35793/Manuels/GenererGoogleSheet.pdf

# Le **management** en couleurs avec le DISC

Partie 1

Thierry Leriche-Dessirier
*Consultant freelance*
http://www.icauda.com

*Votre chef revient d'un séminaire dédié aux outils du manager. Il est enthousiaste. Il ne parle plus que de profils en couleur ; des rouges, des jaunes, des verts ou encore des bleus. Il vous explique que ça va révolutionner la communication entre les membres de l'équipe et la rendre plus efficace. Vous voulez y croire mais cela vous semble bien mystérieux…*

Au quotidien, il est essentiel de savoir bien communiquer. C'est le fondement des relations humaines. Et il est naturel de croire que la communication est avant tout une question de parole et d'écoute mais c'est faux. L'art de la communication repose en premier lieu sur l'observation. C'est là que le DISC, un outil à la fois simple et efficace, entre en jeu.

## LE MODÈLE

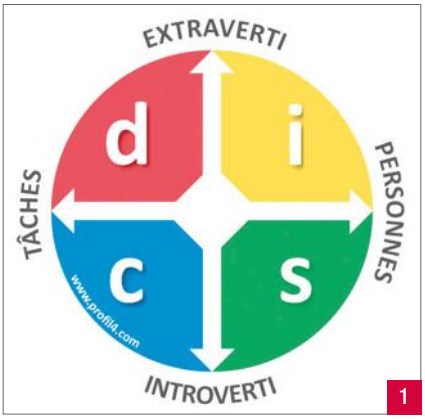
DISC est l'acronyme de Dominant, Influent, Stable et Consciencieux. Le modèle DISC permet d'évaluer le profil des personnes selon ces quatre composantes. Un peu d'histoire : en 1928, William M. Marston, aussi connu pour être le créateur de Wonder Woman(1), publie un ouvrage intitulé « emotions of normal people(2) », dans lequel il introduit les quatre typologies comportementales provenant de la perception de soi dans la relation de la personne à son environnement.

> ***« Are you a normal person ? Probably, for the most, you are.***(3) **»**

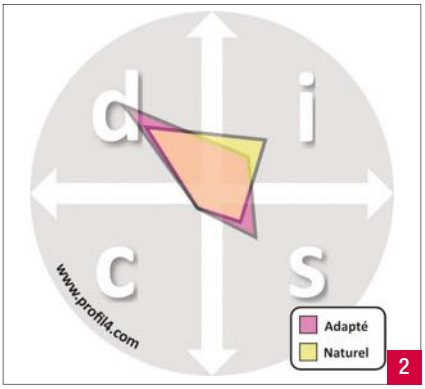
Quelques années plus tard, Walter V. Clarke, un psychologue industriel, est le premier à construire les fondations d'un instrument d'évaluation basé sur les théories de Marston. Utilisé à l'origine pour la sélection du personnel, il met en avant quatre facteurs : agressivité, sociabilité, contrôle des émotions et adaptabilité. Et depuis, l'outil n'a cessé d'évoluer. On en trouve aujourd'hui des déclinaisons(4) intéressantes mais cet article restera concentré sur le modèle classique, qui couvre la quasi-totalité des besoins.

## Roue DISC

On représente généralement les profils DISC sur une roue (disque) dont les quartiers sont les quatre composantes [fig. 1]. Chaque composante possède ses caractéristiques propres et deux quartiers contigus partagent un certain nombre de ces caractéristiques. La roue est construite autour de deux axes principaux. L'axe horizontal sépare les profils extravertis (dominant et influent), qui se sentent à l'aise avec leur environnement, des profils introvertis (stable et consciencieux), qui subissent leur environnement plus qu'ils ne le dominent. L'axe vertical sépare les profils centrés sur les tâches des profils orientés vers les personnes.



1



2

Le profil des personnes est toujours distribué selon les quatre composantes. En général, on observe une ou deux composantes principales. Il est assez rare qu'un profil ne soit distribué que sur une seule composante. Les quartiers opposés sur la roue étant en opposition, il est également rare qu'un profil soit uniformément reparti sur les quatre composantes.

## Adapté Vs Naturel

En outre, les profils de communication et de comportement sont toujours exprimés en naturel et en adapté [fig. 2]. Le style adapté représente le « Moi public ». C'est celui qu'on présente en réponse à l'environnement. Jung appelait cet aspect « le masque ». C'est celui qu'on choisit de montrer aux autres. C'est la façon de paraître. Le style naturel représente le « Moi privé ». Il indique l'aspect du comportement qui est le moins susceptible de varier, c'est celui qu'on a choisi inconsciemment, celui qui a le moins de chances d'être influencé par les attentes de l'entourage. Ce deuxième indicateur montre quelle est la « personne réelle », c'est celui vers lequel on se rabat automatiquement quand on a du mal à maintenir le profil adapté. Il est tout à fait normal que le profil adapté et le profil naturel divergent. C'est le signe que, consciemment ou inconsciemment, à tort ou à raison, on ressent le besoin de s'adapter (d'où le nom) à son environnement, équipe, contraintes, missions, etc. Par exemple, un comptable aura tendance à forcer son côté consciencieux, un commercial essaiera d'être plus avenant, un chef d'équipe se voudra plus autoritaire, etc.

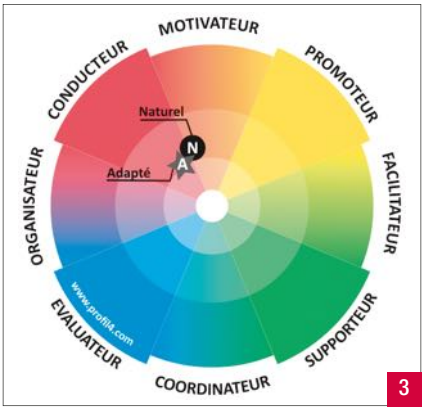
## Roue des tendances

Enfin, selon le niveau de précision souhaité, on pourra compléter la roue DISC (à quatre quartiers) par une roue des tendances (à huit quartiers) : conducteur, motivateur, promoteur, facilitateur, supporteur, coordinateur, évaluateur et organisateur. Chacun de ces huit types correspond à une orientation bien particulière au sein de l'équipe. Il est donc conseillé, quand c'est possible, de choisir les membres de l'équipe en privilégiant une répartition uniforme de

*(1) À noter l'impact de l'héroïne de Comics dans le développement des mouvements féministes outre Atlantique.*
*(2) Les émotions des gens normaux. (3) Introduction de « Emotions of normal people ». (4) Propriétaires.*

leurs tendances. Établir la cartographie [fig. 3] ou la température de l'équipe, et non pas seulement de ses membres individuellement, est le corolaire logique.

Chacun de nous a sa propre façon de communiquer et de se comporter. En connaissant son profil DISC et celui de ses interlocuteurs, ou à défaut une estimation, on peut adapter notre communication pour répondre à leurs besoins. Il ne s'agit pas de manipulation mais de respect. En employant le bon mode de communication, le message passera mieux. La communication sera efficace. L'erreur classique consiste à s'adresser aux autres comme à nous-même, ce qui, en toute logique, est une erreur fondamentale. Par analogie, on pensera au touriste qui emploie la langue du pays qu'il visite et non sa langue maternelle pour demander son chemin.



3

# LES QUATRE PROFILS

## Le dominant

On associe souvent le dominant à la couleur rouge : feu, urgence, pompier, etc. On l'associe volontiers à l'éléphant et au taureau. Le dominant est en haut à gauche du disque. Ça veut dire qu'il est plutôt extraverti et orienté vers les tâches.

Le dominant possède une vision macro. Il aime avoir une vue d'ensemble. Il n'aime pas les détails et ne s'en embarrasse pas. Les détails ont même tendance à l'ennuyer ou à lui faire peur, comme l'éléphant qui a peur de la petite souris. Le dominant est franc. Il ne tourne pas autour du pot pour dire ce qu'il a à dire. Il ne s'encombre pas de fioritures. Il peut d'ailleurs (souvent) être trop franc, ce qui peut mettre mal à l'aise ses interlocuteurs.

Le dominant est motivé par les challenges. Il n'aime pas les tâches répétitives. Ça l'ennuie. Il est capable de les effectuer mais sans entrain. Au contraire, il va se donner à 100% et avec passion dès que le challenge pointe le bout du nez.

Le dominant va droit au but. Il ne s'encombre pas des détails. Il avance. Il fonce dans le tas comme le taureau. Il se donne les moyens d'atteindre ses objectifs. S'il y a des embuches sur son chemin, il va trouver le moyen de les surmonter. Notons que les embuches peuvent être des difficultés techniques mais également des personnes. Le dominant peut écraser ses interlocuteurs pour arriver à ses fins. Il ne fait pas cela méchamment ; il ne s'en rend même pas compte.

Le dominant parle fort. Il est fréquent de voir deux dominants parler et donner l'impression de se crier dessus. Et lorsqu'on leur demande ce qui ne va pas, on est surpris de la réponse car ils indiquent que tout va bien et qu'ils sont simplement en train de discuter. Le dominant parle vite. Il n'aime pas les trous dans les phrases. Il pense qu'une pause est le signe que c'est à son tour de parler.

Le dominant est tourné vers l'action. Il aime que ça avance, que ça bouge. Il n'a pas peur de se tromper et n'a aucun souci pour reconnaître qu'il s'est trompé lorsque ça arrive. Le dominant préfère se tromper que de rester immobile. Il peut prendre des décisions avec très peu de cartes en main, comme dans la variante Texas Hold'em du Poker.

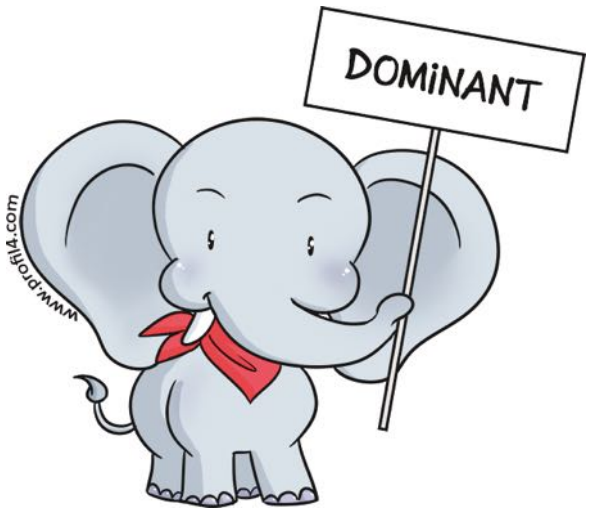
## L'influent

On associe souvent l'influent à la couleur jaune : soleil, joie, etc. On l'associe volontiers au lion et au coq. L'influent est en haut à droite du disque. Comme le dominant, il est donc plutôt extraverti. En revanche, il est orienté vers les personnes.

L'influent montre de l'enthousiasme. Il se passionne très vite pour les nouveautés et possède une réelle capacité pour monter en compétence. L'influent sait transmettre son enthousiasme et sa motivation à son entourage. Il sait comment entrainer ses interlocuteurs à le joindre et le suivre.

L'influent aime travailler avec les autres. Il déteste la solitude. Il sait tout sur les gens avec qui il travaille. On trouvera souvent l'influent à la machine à café, en pleine discussion à propos de tout et de rien avec des collègues. L'influent a le contact facile et parle bien.

L'influent possède un réseau, qu'il entretient. Quand on a un problème, il est capable de nous aiguiller sur la bonne personne qui nous aidera à le résoudre. L'influent délègue beaucoup. C'est le champion de la mise en relation. Il a parfois du mal à comprendre que, lorsqu'on lui confie une tâche, c'est lui qui doit l'effectuer. L'influent vit au travers des yeux des autres. Il aime être le centre d'attention. Il ne supporte pas d'être seul ou ignoré. Il est important de ne pas délaisser un influent trop longtemps sous peine qu'il ne le vive mal et dépérisse. On peut lui faire des compliments, même sur des choses sans importance, car il aime ça et ça le motive.




L'influent a du mal à finir ses tâches, ce qui est certainement le contrecoup de son enthousiasme. L'influent va se passionner très vite pour les nouveautés et devenir compétent sur le sujet. Mais dès qu'une nouveauté arrive, il en oublie littéralement la précédente. Avant de lui confier une tâche, il est donc préférable d'attendre que la précédente soit finie.

L'influent est une fashion victim. Il possède le dernier gadget à la mode. Il sait adapter son apparence à son environnement et est toujours très bien habillé. Il va être soit complétement décalé, avec de nombreuses couleurs notamment, tout en restant très classe. Il sera bien visible. Ou alors il sera très chic, avec un costume de couturier. L'influent ne passe pas inaperçu.

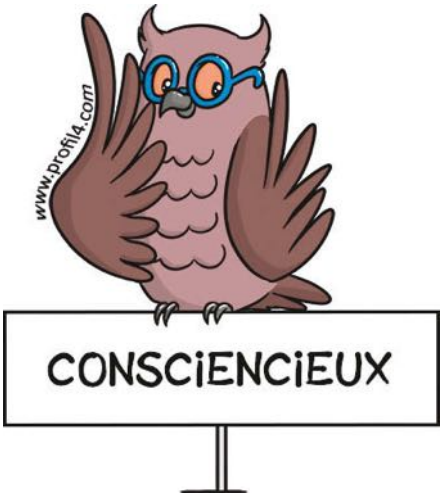
L'influent est le profil typique des commerciaux et VRP, ce qui ne l'empêche pas d'exceller dans d'autres domaines.

## Le stable

On associe souvent le stable à la couleur verte : calme, herbe, nature, etc. On l'associe volontiers au loup ou au chien. Le stable est en bas à droite du disque. Comme l'influent, il est donc orienté vers les personnes. En revanche, il est plutôt introverti. La grosse différence qu'on notera entre l'influent et le stable vis-à-vis de l'orientation vers les personnes réside dans le fait que ce que le stable aime dans cette relation, ce sont les personnes alors c'est le processus de communication qu'aime l'influent.

Le stable parle doucement. Au téléphone, il faut systématiquement lui demander de parler plus fort. Il ne parle pas souvent. Ses interventions sont rares mais pertinentes. Il impose le silence. Quand il prend la parole en réunion, les participants se taisent.

Le stable ne supporte pas d'être stressé, ce qui est une grosse différence avec son profil opposé qu'est le dominant. Ce dernier est dans l'action, il aime que ça aille vite et que ça bouge. Au contraire, le stable a besoin de temps pour digérer l'information et être capable de réagir. Il ne faut pas presser un stable sous peine de le bloquer.


Le stable fait passer les autres avant lui-même. C'est le profil typique de Mère Teresa ou du bon père de famille. Le stable est incapable de résister quand on lui demande son aide. Il va tout lâcher pour aider sa tribu. Le stable agit de façon calme et modérée. Il parle doucement, sans hausser le ton. Il est d'humeur égale. Il n'agite pas les mains lorsqu'il parle. Il les garde sur la table ou dans ses poches. Il ne montre pas spécialement de réaction face aux situations de stress, mais les vit très mal en réalité.

Le stable est humble. Il n'aime pas être mis en avant ou recevoir des compliments, en particulier en public. En revanche, il aime qu'on adresse des compliments à son équipe. Lorsqu'on veut féliciter un stable, il vaut mieux le faire en privé ou alors adresser les félicitations à son équipe si on souhaite le faire en public.

### « de Mère Teresa à démon… »

Le stable peut se transformer en démon. Le stable fait passer sa tribu avant lui-même. Il est calme et serviable. Comme le chien, on peut lui taper dessus et il encaisse les coups. C'est un ange. Mais attention, car cela peut dégénérer. Il y a deux choses à ne jamais faire avec le stable. D'abord il ne faut pas dépasser la limite, faute de quoi il va se transformer en démon et devenir votre pire cauchemar. Ensuite, il ne faut jamais s'en prendre à un membre de sa tribu car, comme un loup, il va le défendre (ou défendre la tribu) avec force. Et on peut être en conflit avec un stable sans même ne lui avoir jamais parlé car il suffit d'avoir agressé (selon ses critères) un membre de sa tribu.

Le stable ne fait pas dans la demi-mesure dans ses réactions. Le stable ne se bat pas pour blesser mais pour tuer. Le stable est patient. Pour lui, la vengeance est un plat qui se mange froid. A moins de vous avoir pardonné vos fautes, le stable vous fera payer très cher vos agressions/attaques.

## Le consciencieux

On associe souvent le consciencieux à la couleur bleue : mer, calme, police, etc. On l'associe volontiers au castor ou au hibou. Le consciencieux est en bas à gauche du disque. Comme le stable, il est plutôt introverti. Et comme le dominant, il est orienté vers les tâches.

Le consciencieux travaille seul. Il n'aime pas particulièrement travailler en équipe. S'il a la chance d'avoir un bureau pour lui tout seul, il en fermera la porte. Il n'apprécie pas de travailler avec les autres, qui le retardent souvent.

Le consciencieux est réfléchi. Il a besoin qu'il y ait une logique pour adhérer à une décision. N'importe quelle logique fera l'affaire. Et si c'est la sienne, c'est encore mieux. Les consciencieux sont les champions pour voir quand quelque chose ne fonctionnera pas. Le consciencieux sait détecter qu'il y a une faille (sans forcément savoir laquelle) dans un système de façon incroyable. Il faut toujours écouter un consciencieux qui vous alerte sur une erreur car il aura raison, et il ne faut surtout pas croire que c'est juste de l'inquiétude.

Le consciencieux adore les détails. Ses emails sont volumineux. Ils contiennent une introduction, une thèse, une antithèse, une conclusion, des démonstrations, etc. Ils contiennent de nombreuses pièces jointes, et le consciencieux s'attend à ce qu'on les lise.

Le consciencieux collectionne les détails. Il en a besoin pour faire son travail d'une façon qu'il juge méticuleuse. Il faut l'abreuver de détails quand on lui confie une tâche, ce qui est une différence avec le dominant qui, lui, ne les supporte pas.

Le consciencieux a peur de se tromper. Cette simple éventualité suffit à le paralyser. Il a du mal à prendre des décisions par peur de l'erreur. Il a besoin d'avoir toutes les cartes en main pour se sentir à l'aise et aura tendance à retarder l'échéance du choix tant qu'il reste des inconnues. D'ailleurs, le consciencieux ne se trompe jamais. Et si cela arrive malgré ses précautions, il sera dévasté, même s'il ne le montre pas.

Le consciencieux est respectueux des règles et des procédures. C'est le profil type des comptables et des administratifs, ce qui ne l'empêche pas d'exceller dans d'autres domaines. •

# Introduction à Rust

Partie 2

Damien Lecan,
*Directeur Technique,*
**SQLI Nantes**

*Nous poursuivons notre découverte du langage Rust. Si la pratique de Rust directement dans le navigateur était adaptée pour débuter, je vous propose désormais de développer directement sur votre poste.*

## Installer Rust

Le site officiel propose des binaires ou des installeurs pour Linux, Mac ou Windows, qui ne se mettent pas à jour automatiquement. Sachant qu'une nouvelle version du langage Rust et des outils est publiée toutes les six semaines et que l'on est régulièrement amené à jongler entre les différents *channels* de Rust (stable, beta, nightly), je vous déconseille cette façon d'installer Rust. Préférez l'usage de Rustup, le nouveau programme officiel d'installation de Rust (la page officielle de téléchargement commence aussi à y faire référence). Vous trouvez la procédure d'installation sur le site **www.rustup.rs**, à savoir une simple commande à taper dans une console : `curl https://sh.rustup.rs -sSf | sh`. Vous installez Rustup qui ensuite installe pour vous le compilateur Rust `rustc`, l'outil de gestion de dépendances et de *build* `cargo`, ainsi que le débuggueur ou le formateur de code de la version *stable* courante de Rust. Vous pourrez très simplement mettre à jour cette version *stable* avec `rustup update` quand vous en aurez besoin (toutes les six semaines !) ou bien installer la *beta* par exemple (`rustup install beta`). Précis et efficace… Quand vous avez `curl` et un interpréteur `sh` à disposition, ce qui n'est pas le cas par défaut sous Windows. Si vous développez avec Rust sous Windows, vous aurez, tôt ou tard, besoin aussi de Git pour versionner vos fichiers sources. Je vous recommande donc d'installer d'abord l'outillage de Git qui vient avec une ligne de commandes assez complète et qui ressemble à ce que vous pourriez obtenir sur un Linux : git-for- windows.github.io. Téléchargez l'installeur 32 ou 64 bits selon votre machine, puis lancez la commande d'installation de Rustup dans le *shell* proposé par *Git for Windows*. A la fin de l'installation, en ligne de commande (sous Windows celle de *Git for Windows*, n'oubliez pas car je ne répéterai plus :-), tapez :

```
$ rustc --version
```

Si vous obtenez quelque chose comme `rustc 1.10.0 (cfcb716cf 2016-07-03)`, le compilateur Rust est opérationnel sur votre poste !

## Editeur de texte malin ou IDE ?

Autant vous le dire d'emblée, les "assistants" de développement Rust sont loin d'être du niveau de ce que l'on peut trouver dans d'autres langages comme Java ou .Net. Il y a encore beaucoup de travail à faire mais on arrive tout même à se créer un environnement acceptable. Personnellement, je travaille avec Sublime Text, complété par quelques plugins qui me permettent d'avoir la coloration syntaxique, le formatage et le *linting*, une validation moins précise de la syntaxe. Je vous invite à consulter areweideyet.com pour choisir l'environnement le plus adapté à votre contexte : Vim, Emacs, Atom, Visual Studio, Eclipse … ou simplement, si vous souhaitez aller vite, un éditeur de texte type Notepad++, Geany ou Sublime Text seront suffisants.

## Calculer division : reboot

Vous souvenez-vous de votre premier programme Rust écrit dans la première partie de ce dossier ? Nous allons le revisiter avec les nouveaux outils dont nous nous sommes dotés, et en particulier `cargo`. C'est un mix de Maven pour la structure standard des projets Rust, de npm pour la gestion de dépendances ou l'installation d'un programme et de commandes permettant de gérer un projet Rust. Créons-en un de type "programme Rust" (grâce au paramètre `--bin`) :

```
$ cargo new --bin division
```

Avec votre éditeur de texte, copiez-collez le contenu de notre dernier programme dans le fichier `src/main.rs` du répertoire `division` créé par Cargo :

```
fn calculer_division(x: i32, y: i32) -> i32 {
    match y {
        0 => panic!("Division par 0"),
        1 => x,
        _ => x / y
    }
}


fn main() {

    let resultat = calculer_division(-4, 2);

    println!("Résultat : {}", resultat);

}
```

Code complet sur ce Gist : **https://git.io/vKcsS**. Sauvegardez, puis lancez en ligne de commande :

```
$ cargo run

  Compiling division v0.1.0 (file://.../division)

    Running `target/debug/division`

Résultat : -2
```

Cargo compile et lance à la suite le programme sans argument. Si vous souhaitez simplement compiler, lancez `cargo build` et si vous souhaitez lancer le programme vous-même, sachez qu'il se trouve dans le répertoire `target/debug` :

```
$ ./target/debug/division

Résultat : -2
```

## Une API sûre

Nous allons variabiliser le numérateur de notre division et le passer en paramètre de la ligne de commande. Explorons l'API de Rust pour ce besoin : lire les arguments en paramètre du programme s'effectue grâce à une fonction du module `std::env` déclarée comme ceci (**https://doc.rust-lang.org/std/env/fn.args.html**) :

```
pub fn args() -> Args
```

La fonction std::env::args() nous renvoie donc une instance de la *struct* Args (elle-même dans le module std::env), une structure qui va contenir des champs et des méthodes permettant de manipuler les arguments du programme, et ce de manière **sûre**. Qu'est-ce que cela signifie ? *Sûr* implique par exemple le bannissement de la "nullité", source de fréquentes erreurs d'exécution (le fameux *NullPointerException* en Java par exemple). En Rust, toutes les API sont conçues pour renvoyer quelque chose - un résultat ou une erreur - et même "rien" est quelque chose en Rust

Option - Some - None

Regardons la déclaration de la fonction nth de Args qui va nous permettre de récupérer le *n*ième argument de notre programme. Elle est déclarée comme ceci (la signature est légèrement adaptée pour une compréhension plus aisée) :

```
fn nth(&mut self, n: usize) -> Option<String>
```

Mettons de côté le &mut self, nous y reviendrons par la suite. nth est une fonction qui renvoie une Option de type String. Option est une énumération Rust à deux variantes possibles :

```
pub enum Option<T> {
   None,
   Some(T),
}
```

Si vous substituez le type générique T par String, vous comprenez alors que la méthode nth peut renvoyer soit None, qui signifie qu'il n'y a pas de valeur à cette position, soit Some(String), qui signifie qu'il existe une valeur à la position demandée et qu'elle peut être extraite de la valeur de l'énumération. Ce qui est génial, c'est que ces variantes peuvent être "matchées" (Cf. partie 1 de ce dossier) :

```
use std::env;

...

let numerateur = match env::args().nth(1) {
   Some(argument) => argument,
   None => panic!("Argument obligatoire manquant : le numérateur")
};
```

Ici, on effectue aussi du *pattern matching* pour extraire la valeur contenue dans le Some. Some(argument) permet de déclarer la variable argument, affectée à la valeur contenue dans le Some, que l'on peut alors renvoyer (avec un *return* implicite). On arrête le programme prématurément avec la macro panic! et un message explicite en cas de None. Enfin, cerise sur le gâteau, comme tout est expression en Rust, on peut affecter directement le retour de notre match à une variable : let numerateur = match env::args() ....

Notez l'utilisation du module env importé par une ligne en début de programme, use std::env;, à ajouter pour chaque module utilisé dans le programme. Avec les Option et l'API de Rust, nous avons pu extraire notre paramètre de ligne de commande et l'avons rendu obligatoire. Ce n'est cependant pas suffisant : en effet, la fonction nth renvoie une Option<String> dans notre cas, ce qui veut dire que la variable numerateur ci-dessus est de type String, alors que nous attendons un i32. Il faut donc convertir notre valeur.

## Result - Ok - Err

Rust propose une API de conversion de String :

```
fn parse<F>(&self) -> Result<F, F::Err> where F: FromStr
```

Prenons quelques instants pour comprendre la signature de cette fonction :

- fn parse : "parse" est le nom de la fonction :-) ;
- F : il s'agit d'un type générique, que vous pouvez substituer mentalement par le type que vous voulez obtenir après le parsing de votre String. La définition en est donnée en fin de ligne, where F: FromStr, ce qui signifie que F doit respecter le contrat décrit dans le *trait* std::str::FromStr (c'est une sorte d'interface) ;
- &self : il s'agit de la syntaxe spécifique à Rust qui indique que cette fonction n'est pas statique et qu'elle s'applique sur des instances de l'objet courant (String ici) ;
- Result<F, F::Err> : le type de retour, encapsulant F et un type d'erreur associée à F.

Result est la façon élégante en Rust de gérer les éventuels retours en erreur d'un traitement. Il est décrit comme ceci dans l'API de Rust :

```
pub enum Result<T, E> {
   Ok(T),
   Err(E),
}
```

Un peu comme Option vue précédemment, Result est une énumération à deux variantes, sur lesquelles on pourra "matcher" :

- Ok est utilisé pour encapsuler le résultat d'un traitement qui s'est bien déroulé ;
- Err permet de propager les erreurs de traitement.

Il n'y a donc pas d'exception en Rust et l'usage des codes retours pour indiquer un résultat de traitement est considéré comme une mauvaise pratique, car non sûre. Le type i32 implémentant bien le *trait* std::str::FromStr, on pourrait écrire la fonction dédiée au parsing d'une String en i32 :

```
fn parse(&self) -> Result<i32, ParseIntError>
```

L'association de ParseIntError à i32 est décrite dans sa documentation (cherchez "impl FromStr for i32" et juste en dessous le type Err = ParseIntError). Gardez en tête que c'est une simple vue de l'esprit, car elle est redondante avec la définition de la fonction décrite avec un type générique, mais elle permet de fixer les idées quand on n'est pas encore à l'aise avec la syntaxe de Rust. Nous pouvons donc convertir notre String numerateur en i32 après un matching :

```
let numerateur = match numerateur.parse::<i32>() {
   Ok(numerateur) => numerateur,
   Err(error) => panic!("Impossible de convertir notre argument. Raison: {}", error)
};
```

Il faut un peu aider le compilateur car il ne peut pas deviner quelle conversion on souhaite appliquer. Pour cela, on utilise la syntaxe *turbofish*: ::<>, qui permet de spécifier le type de destination. Si le parse s'est bien déroulé, le matching sur Ok permet d'extraire le numérateur sous forme de i32 désormais ; sinon avec le matching sur Err, on arrête une nouvelle fois le programme avec un message d'erreur adéquat. Enfin, substituez le premier paramètre de l'appel de la fonction calculer_division par la variable numerateur (code complet : https://git.io/v6ypr), compilez et exécutez en une fois :

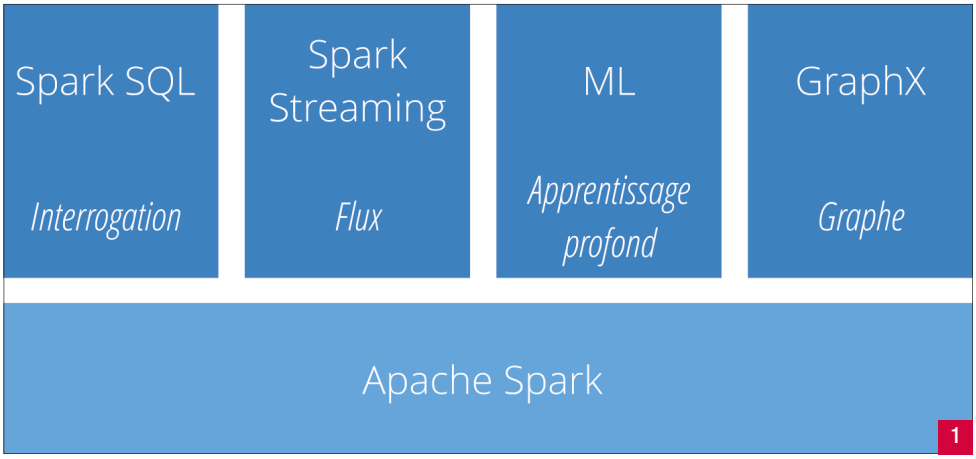
```
$ cargo run 4

  Compiling division v0.1.0 (file://.../division)

   Running `target/debug/division 4`

Résultat : 2
```

Bravo, en quelques lignes de code robustes, vous avez géré la présence et l'absence d'arguments lors de l'exécution de notre programme et ce, de manière plutôt élégante. A bientôt. •

# Spark en quelques heures

• Jean Georges Perrin
(jgp@jgp.net)
*architecte (software architect) pour Zaloni. Zaloni est un éditeur de logiciels, proposant des outils de gestion des lacs de données (Data lakes) construits autour d'Apache Hadoop. Il vit aujourd'hui en Caroline du Nord.*

ZALONI
THE DATA LAKE COMPANY



1

Il était une fois de simples mages qui voulaient traiter énormément de données. Ils pensaient que les bases de données traditionnelles ne conviendraient pas. Non, les systèmes de gestion de bases de données (ces fameux SGBD) conventionnels ne convenaient pas : les Oracles et MySQL ne comprenaient pas bien les fichiers, nécessitaient des données structurées. Même les bases un peu plus évoluées comme Informix et son support des objets et des tables virtuelles, ne correspondaient pas. Les mages du royaume de Yahoo avaient besoin de quelque chose de plus puissant, mais qui fonctionne sur des serveurs conventionnels, pas des monstres octo-processeurs, qui coûtent chers. Les mages avaient besoin de beaucoup de potions, pas de potions ultra puissantes. C'est ce que ces mages aux pouvoirs étonnants mais parlant principalement anglais appelaient « scale out ».

Rapidement, les mages ont pu stocker et traiter des quantités impressionnantes de grimoires, livres, recettes, diagrammes et autres sorcellerie… Les mages nomment cette magie « Hadoop » et, dès 2006, Hadoop trie 1.8 To sur 188 nœuds en moins de 2j.

Mais cela ne suffit pas. Les diableries d'Hadoop sont limitées. Hadoop est essentiellement cantonné à son système de fichiers, HDFS, et une méthode de traitement, MapReduce.

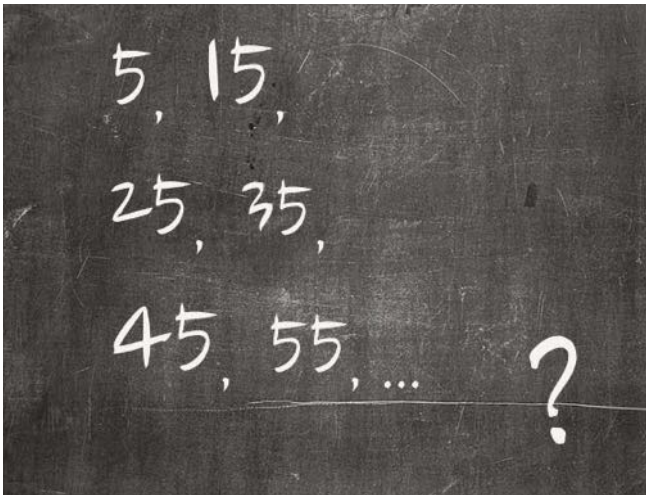
Pour faire des étincelles, de grands érudits de l'Université de Berkeley ont l'idée de mieux utiliser la mémoire et d'ouvrir à d'autres algorithmes. Spark est né.

## Faire travailler les machines

De tous temps, l'homme a essayé de faire travailler soit d'autres hommes, soit des machines. C'est ultimement – avec la rencontre avec les Vulcains - ce qui a permis de créer la Fédération des Planètes Unies qui assurent aujourd'hui la paix dans la galaxie (bon, il y a toujours un ou deux Klingons pour animer tout cela).

Mais, avant d'y arriver, nous avons dû passer par plusieurs étapes d'intelligence artificielle. Les premières étapes sont passées par des langages comme Prolog ou Lisp. Bien évidemment, cela date d'avant la 3ème guerre mondiale de 2026. Au début du 21ème siècle, les capacités de calculs restent en croissance suivant la loi de Moore, mais la croissance des données devient exponentielle avec l'arrivée des premiers téléphones intelligents, (ancêtre de nos communicateurs), le réseau Internet qui permet aujourd'hui la téléportation, etc.

Donc, d'un côté, nous avons une croissance linéaire de la puissance de calcul et de l'autre une croissance exponentielle des données. La solution




– temporaire à ce moment-là – a été de créer des clusters (ou groupement de serveurs) de calcul. Spark a été un noyau fondateur de cette technologie en offrant une unification des APIs qui ont permis d'avancer dans l'**apprentissage profond** (*deep learning*). L'interrogation se fait en SQL (enfin, une version de SQL spécifique à Spark), la gestion des graphes via GraphX et les flux via Spark Streaming. Fin 2016, les couches (*stack*) Spark ressemblent à cela : [Fig.1]

Spark a été écrit en Scala. Ses APIs sont accessibles en Java, Python et, forcément, Scala.

## Machine, travaille !

C'est grâce à l'apprentissage profond que nous avons pu créer l'intelligence des premiers androïdes qui deviendront les T-800 qui aujourd'hui nous menacent.

Avant Skynet, un autre réseau social du nom de Facebook et un autre du nom de LinkedIn (qui ressemblait de plus en plus au premier au grand dam de ses utilisateurs) servaient à amuser les foules. Dès le début de ces réseaux sociaux, de nombreux groupes y lancèrent des bots qui se faisaient passer pour des humains. Sur ces réseaux sociaux, il était d'usage de partager des photos de chats avec des textes amusants (des *memes*) et des petits casses têtes comme celui-ci.

Qu'y a-t-il après 55 dans notre suite ? De façon quasi instantanée, vous allez me dire 65, non ? Mais comment êtes-vous arrivés à 65 ? En ajoutant 10 au dernier nombre ?

Et si mes nombres étaient 5, 13, 27, 39, 41 et 55… Est-ce que vous me diriez toujours 65 ? Vous pouvez toujours ajouter 10 à 55, mais d'où vient ce 10 ?

Ce concept mathématique s'appelle une **régression linéaire**. Il fait partie de l'algèbre linéaire et c'est un des premiers exercices que l'on peut faire quand on s'attaque à l'apprentissage profond ou **apprentissage automatique** (*machine learning*). La librairie Spark ML est au cœur de ce mécanisme.

Pour mieux comprendre le principe, nous pouvons faire un graphe et commencer à placer des points. Sur l'abscisse (*x*), on marque la position dans la suite, en ordonnée (*y*) on marque la valeur.

Dans le cas de notre première suite, cela donnerait : [Fig.2]

On voit bien que le 7ème élément de la série est 65. On peut également imaginer que le 8ème élément est 75…

Comme à chaque fois que l'on travaille avec de nouveaux concepts, la ligne en pointillés (qui est également sous la ligne pleine) est la **droite de régression**. Les éléments sur l'axe des x (1, 2, 3…8) sont des **traits** (*features* en anglais), les valeurs (5, 15…) sont des **traits** (*labels* en anglais).

Pour mieux comprendre le vocabulaire mathématique français et anglais, vous pouvez voir les travaux d'Eric Reyssat (http://bit.ly/2d036Rm) ou de Kai-Wen Lan (http://bit.ly/2cFor33).

Si je fais un graphe à partir de ma 2ème série de données, j'obtiens : [Fig.3]

La droite de régression apparaît beaucoup plus clairement. On comprend mieux que cette droite essaie de passer au plus près de tous les points. Cette droite peut s'exprimer sous la forme d'une équation : $y = \beta_1 x + \beta_0$

Le paramètre $\beta_0$ représente l'**ordonnée à l'origine** (*intercept* en anglais) et $\beta_1$ le **coefficient directeur** de la droite (*regression parameter* en anglais). Ces 2 paramètres peuvent facilement être calculés en algèbre linéaire – ou nous pouvons utiliser des outils pour les calculer (dont Spark). Dans notre premier exemple, notre équation est : $y = 10x - 5$.

Dans notre 2ème exemple, l'équation est $y = 9.8857 \cdot x - 4.6$. Il y a donc clairement une différence. Dans le premier cas, quand x vaut 7, y vaut 65 alors que dans le deuxième cas, y vaut 64.6. Ok, ce n'est pas exact, mais probablement assez dans notre cas, non ?

Au lycée – et après, j'ai adoré les maths, mais je dois admettre que cela n'est plus ma plus grande passion. Codons !

Je vais utiliser Java et Apache Spark ML v2.0.0.

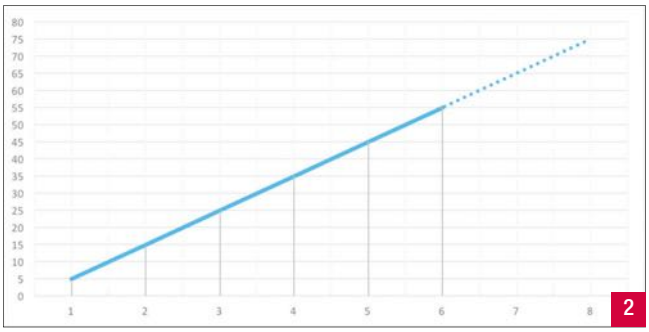
Java est toujours à la base de la programmation des T-800, comme il était un des langages de développement des plus utilisés dans les entreprises en 2016. ML est la librairie dédiée au Machine Learning – oui, l'inspiration n'a pas été leur plus grand fort lorsqu'il a fallu trouver un nom.

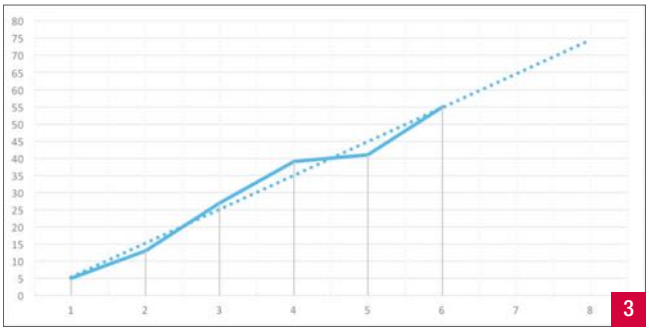
Deux façons d'obtenir le code : cherchez-le sur GitHub ou faites un bon vieux copier/coller (je recommande la version en ligne de Programmez ! pour le copier/coller).

Pour télécharger les exemples à partir de GitHub, allez sur :
https://github.com/jgperrin/net.jgp.labs.spark (et ne soyez pas timide, tant que vous êtes là-bas, *forkez*-le ou faites-en un favori). Il y a quelques dépendances.

Il y a quelques packages à importer. J'ai préféré les laisser parce qu'il peut y avoir des conflits de noms. Vector en est un. Certains packages sont différents avec Spark v1.6.2. Le détail des dépendances est dans pom.xml.

```
package net.jgp.labs.spark;

import static org.apache.spark.sql.functions.callUDF;

import org.apache.spark.ml.linalg.Vector;
```




```
import org.apache.spark.ml.linalg.VectorUDT;
import org.apache.spark.ml.linalg.Vectors;
import org.apache.spark.ml.regression.LinearRegression;
import org.apache.spark.ml.regression.LinearRegressionModel;
import org.apache.spark.ml.regression.LinearRegressionTrainingSummary;
import org.apache.spark.sql.Dataset;
import org.apache.spark.sql.Row;
import org.apache.spark.sql.SparkSession;
import org.apache.spark.sql.types.DataTypes;
import org.apache.spark.sql.types.Metadata;
import org.apache.spark.sql.types.StructField;
import org.apache.spark.sql.types.StructType;

import net.jgp.labs.spark.udf.VectorBuilder;
```

Il y a un main() de base qui va instancier la classe et appeler une méthode de start() qui va lancer le boulot.

```
public class SimplePredictionFromTextFile {

  public static void main(String[] args) {
    System.out.println("Working directory = " + System.getProperty("user.dir"));
    SimplePredictionFromTextFile app = new SimplePredictionFromTextFile();
    app.start();
  }
```

Spark v2.0.0 force l'utilisation d'une session Spark. Dans les versions précédentes, c'était un peu compliqué et il fallait gérer plusieurs sessions et configurations. C'est plus simple maintenant.

```
  private void start() {
    SparkSession spark = SparkSession.builder().appName("Simple prediction from Text
File").master("local").getOrCreate();
```

Nous avons besoin d'une fonction utilisateur (UDF pour User Defined

Function). Elle va nous permettre de transformer notre fichier en entrée dans un format qui pourra être utilisé par Spark.

```
spark.udf().register("vectorBuilder", new VectorBuilder(), new VectorUDT());
```

Nos données sont dans le fichier tuple-data-file.csv. Pour être précis, notre premier set est dans tuple-data-file-set1.csv et le deuxième est dans devine-dans-quel-fichier.csv – nan… ils sont dans tuple-data-file-set2.csv, mais je voulais m'assurer que vous suiviez.

```
String filename = "data/tuple-data-file.csv";
```

Dans notre situation, nous devons forcer la structure de nos données pour que Spark puisse mieux comprendre les metadonnées associées à nos données.

```
new StructField[] { new StructField("_c0", DataTypes.DoubleType, false, Metadata.empty()),
        new StructField("_c1", DataTypes.DoubleType, false, Metadata.empty()),
        new StructField("features", new VectorUDT(), true, Metadata.empty()), });
```

Dans Spark v2.0.0, en utilisant Java, notre bien-aimé dataframe est implémenté sous la forme d'un Dataset<Row> -- d'où l'utilisation de la variable df.

```
Dataset<Row> df = spark.read().format("csv").schema(schema).option("header", "false")
        .load(filename);
df = df.withColumn("valuefeatures", df.col("_c0")).drop("_c0");
df = df.withColumn("label", df.col("_c1")).drop("_c1");
df = df.withColumn("features", callUDF("vectorBuilder", df.col("valuefeatures")));
```

Comme vous le voyez, nous transformons le dataframe pour créer les champs *label* et *features*. Pour être précis, un label associé à un vecteur de traits (*features*). Ces noms sont exigés par Spark ML.

```
df.printSchema();
df.show();
```

Nous sommes désormais prêts pour construire notre régression linéaire. Je vais limiter à 20 itérations, ce qui est parfaitement suffisant vu la complexité (ou non) des données.

```
LinearRegression lr = new LinearRegression().setMaxIter(20);
```

Et on applique la régression linéaire à nos données, stockées dans le dataframe. Nous obtenons notre modèle de régression linéaire.

```
LinearRegressionModel model = lr.fit(df);

// Given a dataset, predict each point's label, and show the results.
model.transform(df).show();
```

Maintenant, je peux jeter le 7ème élément en pâture à notre modèle. Je crée un vecteur et je lance la prédiction…

```
Double feature = 7.0;
Vector features = Vectors.dense(feature);
double prediction = model.predict(features);
    System.out.println("Prediction for feature " + feature + " is " + prediction);
  }
}
```

Dans le code qui est sur GitHub, beaucoup plus d'informations statistiques sont affichées.

Après exécution, nous obtenons :

```
+--------+-------------+------+-------------------------+
|features|valuefeatures| label|               prediction|
+--------+-------------+------+-------------------------+
|   [1.0]|          1.0|   5.0| 4.999999999999992|
|   [2.0]|          2.0|  15.0|14.999999999999993|
|   [3.0]|          3.0|  25.0|24.999999999999996|
|   [4.0]|          4.0|  35.0|                     35.0|
|   [5.0]|          5.0|  45.0|                     45.0|
|   [6.0]|          6.0|  55.0| 55.00000000000001|
+--------+-------------+------+-------------------------+

Prediction for feature 7.0 is 65.0
```

Et pour notre deuxième jeu de données :

```
+--------+-------------+------+-----------------------+
|features|valuefeatures|label|             prediction|
+--------+-------------+------+-----------------------+
|   [1.0]|          1.0|   5.0| 5.285714285714304|
|   [2.0]|          2.0|  13.0|15.171428571428583|
|   [3.0]|          3.0|  27.0|25.057142857142864|
|   [4.0]|          4.0|  39.0| 34.94285714285714|
|   [5.0]|          5.0|  41.0| 44.82857142857142|
|   [6.0]|          6.0|  55.0|  54.7142857142857|
+--------+-------------+------+-----------------------+

Prediction for feature 7.0 is 64.59999999999998
```

C'est un exemple très simple avec un jeu de données très limité. Maintenant vous pouvez cocher la case « connaissance de l'apprentissage automatisé » (ou l'ajouter à votre CV). Si vous êtes sérieux à ce sujet, Zaloni offre les 3 premiers chapitres du livre Fundamentals of Deep Learning chez O'Reilly par Nikhil Buduma (la version finale n'est pas prévu avant Noël 2016), http://bit.ly/2d97LUu.

En conclusion, même si nous ne sommes ni dans Donjons et Dragons, ni dans Star Trek et encore moins dans l'univers de Terminator, le monde des données et de son traitement automatisé (ça s'appelle l'informatique, hein…) est en train de se révolutionner. La loi de Moore nous a accompagnés jusqu'à maintenant, mais elle arrive à une limite, surtout face à l'explosion des données.
Des modèles de développement collaboratifs – qui rappellent étrangement le monde économiquement utopiste de Star Trek – nous permettent d'accéder à des technologies inédites et réellement bouleversantes dont Spark fait partie. •

# A la découverte des **Progressive Web Apps**

• Sylvain Saurel
sylvain.saurel@gmail.com
*Développeur Android*
http://www.ssaurel.com

*Une Progressive Web App tire parti des dernières technologies afin de combiner au mieux le meilleur des applications Web et Mobiles. On peut ainsi résumer une Progressive Web App en disant qu'il s'agit d'un site Web construit avec des technologies Web mais se comportant et ayant l'aspect d'une application Mobile.*

Les dernières avancées au sein des navigateurs avec la mise à disposition des Service Workers ainsi que des APIs Cache et Push ont permis aux développeurs Web d'offrir aux utilisateurs une expérience toujours plus poussée. Ainsi, ces derniers peuvent désormais installer les applications Web directement sur leur écran d'accueil, recevoir des notifications push et même travailler en mode offline.
Les Progressive Web Apps présentent l'insigne avantage de reposer sur l'écosystème Web qui, à bien des égards, est beaucoup plus vaste que celui des applications natives. En outre, les développeurs d'applications Mobiles apprécieront une maintenance simplifiée sans avoir à se soucier des problématiques de compatibilité ascendante puisque tous les utilisateurs accèderont à la même version de code sur le site Web, là où la fragmentation des versions demeure un enjeu majeur des applications natives. De fait, le déploiement et la maintenance d'une Progressive Web App seront plus aisés.

## Pourquoi les Progressive Web Apps ?

Des études récentes ont montré qu'en moyenne une application perd près de 20% de ses utilisateurs potentiels chaque fois que l'on rajoute une étape entre le premier contact visuel avec l'application et son premier lancement effectif. En effet, une application native doit d'abord être recherchée au sein d'un App Store, puis téléchargée, installée pour finalement être enfin lancée. A contrario, lorsqu'un utilisateur trouve votre Progressive Web App, il pourra l'utiliser directement, ce qui élimine les étapes de téléchargement et d'installation. Au second lancement de l'application, il lui sera alors demandé d'installer l'application et de passer à une expérience plein écran.
Une Progressive Web App vise à tirer avantage des caractéristiques des applications Mobiles, avec des performances et une rétention des utilisateurs accrues, tout en se libérant des problématiques induites par la maintenance des applications Mobiles.

## Cas d'utilisations

A la lecture de ces quelques lignes, une question se pose tout naturellement : dans quel cas faut-il basculer vers une Progressive Web App ? L'usage du natif est généralement recommandé pour des applications qui ont vocation à être utilisées fréquemment par les utilisateurs. Néanmoins, cela peut également correspondre à des cas d'usages d'une Progressive Web App. Alors, à quel moment peut-on savoir qu'il est temps de basculer vers ce dernier type d'application plutôt que de rester sur un site Web ou de passer à une application native ? La première chose à faire est d'identifier les utilisateurs ciblés et plus encore quels types d'interactions sont attendus. Par nature, les Progressive Web Apps fonctionnent parfaitement sur tous types de navigateurs et l'expérience utilisateur sera améliorée dès lors que le navigateur de l'utilisateur sera mis à jour avec des nouvelles fonctionnalités et APIs.
Ainsi, il n'y a pas de compromis à faire en termes d'expérience utilisateur entre une Progressive Web App et un site Web traditionnel. Cependant, il faudra décider quelles fonctionnalités seront supportées en mode offline mais également faciliter la navigation puisqu'idéalement en mode standalone, l'utilisateur n'a pas accès au bouton retour. Si votre site Web existe déjà et possède une interface type application, appliquer les concepts associés aux Progressive Web Apps ne pourra que le rendre meilleur et plus confortable pour l'utilisateur.
Enfin, si certaines fonctionnalités nécessaires pour des actions utilisateurs essentielles au bon fonctionnement de l'application ne sont pas disponibles du fait d'un manque de support multi navigateurs, basculer vers une application Mobile sera à ce moment-là la meilleure solution afin de garantir la même expérience d'utilisateur à tous les utilisateurs.

## Caractéristiques d'une Progressive Web App

Le concept général de Progressive Web App présenté, nous pouvons maintenant lister leurs 10 caractéristiques fondamentales telles que définies par Google :

- **Progressive :** comme son nom l'indique, une Progressive Web App doit fonctionner sur tous les types de périphériques et s'améliorer progressivement en tirant partie des fonctionnalités disponibles sur les périphériques et navigateurs des utilisateurs.
- **Discoverable :** une Progressive Web App étant un site Web, elle se doit d'être facilement trouvable au sein des moteurs de recherches. Il s'agit ici d'un avantage majeur par rapport aux applications natives.
- **Linkable :** une autre caractéristique héritée des sites Web qui doit s'appliquer concerne le côté "linkable". En effet, un site Web correctement conçu doit utiliser des URI pour indiquer l'état courant d'une application ce qui permet de retenir ou de recharger un état donné lorsque l'utilisateur met en favori ou partage une URL au sein de l'application.
- **Responsive :** l'interface utilisateur d'une Progressive Web App doit s'adapter à la forme et à la taille d'écran des périphériques sur lesquelles elle est rendue.
- **App-like :** une Progressive Web App doit avoir l'aspect d'une application native et être construite de façon à limiter le nombre de rechargements de pages.
- **Connectivity-independent :** elle doit fonctionner même en conditions de faible connectivité réseau voir en mode offline complet.
- **Re-engageable :** les applications Mobiles ont globalement plus de chance d'être réutilisées par les utilisateurs. En s'appuyant sur les push notifications, les Progressive Web Apps doivent tendre vers le même but.
- **Installable :** une Progressive Web App doit pouvoir être installée sur l'écran d'accueil d'un périphérique afin d'être plus facilement accessible.

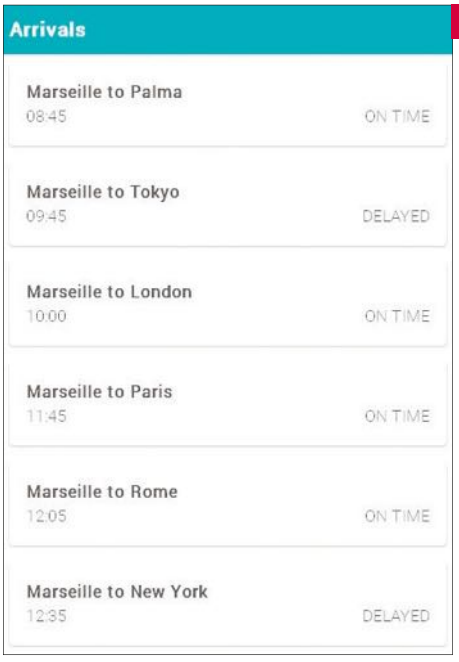
- **Fresh :** un nouveau contenu publié au sein d'une Progressive Web App doit être accessible instantanément pour l'ensemble des utilisateurs connectés à Internet.
- **Safe :** puisqu'elle propose aux utilisateurs une expérience plus approfondie et que toutes les requêtes réseaux qu'elle réalise peuvent être interceptées via les Service Workers, une Progressive Web App se doit impérativement d'être hébergée en HTTPS pour se prémunir des attaques de type Man-in-the-Middle.

## Place au code !

Les fondamentaux concernant les Progressive Web Apps présentés, il est temps de passer à la pratique et au code. Pour ce faire, nous allons développer une Progressive Web App permettant de simuler un tableau des arrivées d'avions au sein d'un aéroport (figure 1). Au premier accès à l'application Web, l'utilisateur verra la liste des vols récupérés depuis une API dédiée. Si l'utilisateur n'a plus de connexion Internet par la suite et recharge l'application Web, nous proposerons à l'utilisateur la liste précédemment obtenue lorsque la connexion Internet était encore active.

La caractéristique première d'une Progressive Web App est de fonctionner sur tous les périphériques disponibles tout en proposant une expérience utilisateur améliorée sur les périphériques et navigateurs offrant plus de fonctionnalités. Nous allons donc construire l'application comme un site Web classique avec du HTML 5 et du Javascript. Afin de faciliter la mise en place d'une approche MVVM (Model-View-ViewModel), nous utiliserons la bibliothèque Knockout qui est un framework Javascript facilitant le binding entre les modèles Javascript et les vues HTML.

Le site Web que nous allons définir va suivre les guidelines Material Design de Google au niveau du design et des interactions. Le respect de ces guidelines nous permettra d'obtenir plus facilement un aspect type application pour notre site. Afin de faciliter la mise en oeuvre de notre exemple, l'API fournissant la liste des avions en cours d'arrivée sera constituée d'un fichier JSON statique. Bien entendu, dans un exemple plus complexe, il sera possible de requêter une API temps réel ou d'utiliser des WebSockets. Coeur du site Web, le fichier index.html aura l'allure suivante :




1 *Aspect de notre Progressive Web App*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Marseille Airport Arrivals</title>
  <link async rel="stylesheet" href="./css/style.css">
  <link href="https://fonts.googleapis.com/css?family=Roboto:300,600,300italic,
600italic" rel="stylesheet" type="text/css">
</head>

<body>
  <header>
    <div class="content">
      <h3>Arrivals</h3>
    </div>
  </header>
  <div class="container">
    <div id="main" class="content">
      <ul class="arrivals-list" data-bind="foreach: arrivals">
        <li class="item">
          <span class="title" data-bind="html: title"></span>
          <span class="status" data-bind="html: status"></span>
          <span class="time" data-bind="html: time"></span>
        </li>
      </ul>
    </div>
  </div>
  <script src="./js/vendor/knockout-3.3.0.js"></script>
  <script src="./js/page.js"></script>
  <script src="./js/arrivals.js"></script>
  <script src="./js/main.js"></script>
</body>
</html>
```

Comme vous pouvez le constater, il s'agit ici d'une page HTML assez standard. Une liste HTML est créée et nous la lions à la propriété arrivals pour l'utiliser ensuite avec Knockout. Ceci se faisant au nouveau de l'attribut data-bind. Le ViewModel arrivals est ensuite défini au sein d'un fichier page.js et exposé au sein d'un module Page. Rappelons qu'au niveau de la page HTML, chaque élément de la liste des arrivées est lié à une propriété. Le fichier page.js a ainsi l'allure suivante :

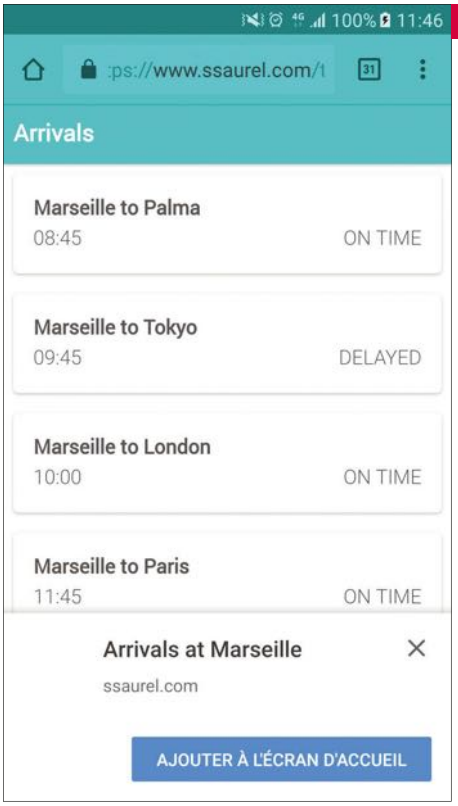
```
(var Page = (function() {
  function ViewModel() {
    var self = this;
    self.arrivals = ko.observableArray([]);
  }

  return {
    vm: new ViewModel(),
    hideOfflineWarning: function() {
      document.querySelector(".arrivals-list").classList.remove('loading')
      document.getElementById("offline").remove();
    },
    showOfflineWarning: function() {
      document.querySelector(".arrivals-list").classList.add('loading')
```

```
        var request = new XMLHttpRequest();
        request.open('GET', './offline.html', true);

        request.onload = function() {
          if (request.status === 200) {
            var offlineMessageElement = document.createElement("div");
            offlineMessageElement.setAttribute("id", "offline");
            offlineMessageElement.innerHTML = request.responseText;
           document.getElementById("main").appendChild(offlineMessageElement);
          } else {
            console.warn('Error retrieving offline.html');
          }
        };

        request.onerror = function() {
          console.error('Connection error');
        };

        request.send();
      }
    }
})();
```

Ici, nous exposons donc le module Page contenant le ViewModel vm et 2 fonctions à savoir hideOfflineWarning et showOfflineWarning. Vous remarquerez également que le ViewModel est un simple littéral Javascript qui sera utilisé tout au long de l'application. La propriété arrivals du ViewModel est un objet observableArray de Knockout, ce qui permet de la binder automatiquement avec la vue HTML. Ainsi, il sera possible d'ajouter ou de retirer des éléments au tableau des arrivées avec une répercussion automatique sur la page HTML. Les fonctions de gestion du mode offline sont assez simples et l'on notera le téléchargement de la page offline. Si le fichier est correctement récupéré, nous l'ajoutons au sein de notre page HTML. Ici, nous n'utilisons pas de Service Workers pour le moment.

La partie métier de l'application va consister à récupérer les données concernant les arrivées d'avion via notre API statique. Pour cela, il est nécessaire de binder les objets ViewModel et les Views au sein d'un fichier arrivals.js comme cela est prévu en standard par la fonctionnalité MVVM de Knockout. Au sein de ce fichier, nous allons initialiser les services et les objets ViewModel pour finalement exposer la fonction Arrivals.loadData() en charge de récupérer les données qui seront ensuite bindées au ViewModel :

```
var Arrivals = (function() {
  function ArrivalViewModel() {
    var self = this;
    self.title = "";
    self.status = "";
    self.time = "";
  }

  function ArrivalApiService() {
    var self = this;

    // récupération des arrivées depuis l'API
    self.getAll = function() {
      return new Promise(function(resolve, reject) {
        var request = new XMLHttpRequest();
        request.open('GET', './api/data.json');

        request.onload = function() {
          if (request.status === 200) {
            resolve(JSON.parse(request.response));
          } else {
            reject(Error(request.statusText));
          }
        };

        request.onerror = function() {
          reject(Error("Network Error"));
        };

        request.send();
      });
    };
  }

  function ArrivalAdapter() {
    var self = this;

    self.toArrivalViewModel = function(data) {
      if (data) {
        var vm = new ArrivalViewModel();
        vm.title = data.title;
        vm.status = data.status;
        vm.time = data.time;
        return vm;
      }
      return null;
    };

    self.toArrivalViewModels = function(data) {
      if (data && data.length > 0) {
        return data.map(function(item) {
          return self.toArrivalViewModel(item);
        });
      }
      return [];
    };
  }

  function ArrivalController(arrivalApiService, arrivalAdapter) {
    var self = this;

    self.getAll = function() {
      // on récupère les arrivées depuis l'API
      return arrivalApiService.getAll().then(function(response) {
        return arrivalAdapter.toArrivalViewModels(response);
      });
    };
  }

// initialisation des services, des adapters et du controller
```




2 *Rendu sur un smartphone*

```
  var arrivalApiService = new ArrivalApiService();
  var arrivalAdapter = new ArrivalAdapter();
  var arrivalController = new ArrivalController(arrivalApiService, arrivalAdapter);

  return {
    loadData: function() {
      document.querySelector(".arrivals-list").classList.add('loading')
      arrivalController.getAll().then(function(response) {
        // binding des arrivées sur l'UI
        Page.vm.arrivals(response);
        document.querySelector(".arrivals-list").classList.remove('loading')
      });
    }
  }

})();
```

Au sein de ce fichier, la fonction ArrivalApiService est en charge de récupérer les données de l'API statique. Pour le parsing des données au format JSON, on utilise ici la méthode parse de l'objet JSON standard. Tout le reste du travail consistant à assembler les différents éléments de notre site Web afin de les faire fonctionner de concert.

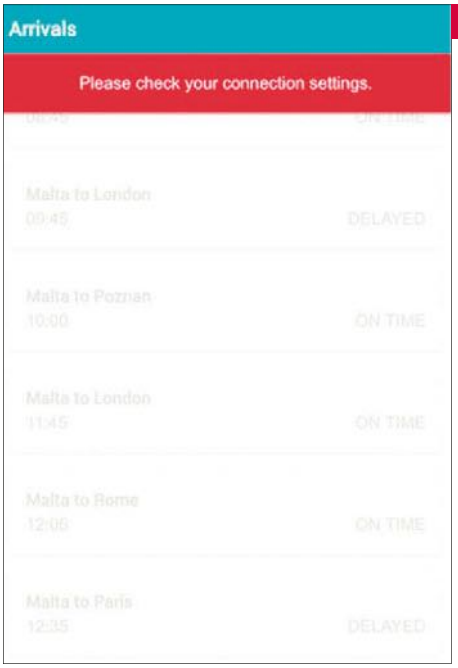
## Web App Manifest

Avant de tester notre application Web, il est important de la rendre plus proche d'une application que ce soit en termes d'aspect ou de comportement avec la possibilité de l'ajouter à l'écran d'accueil d'un smartphone par exemple. Pour ce faire, nous allons définir un Web App Manifest. Il s'agit d'un fichier au format JSON normalisé par le W3C. Une fois ce fichier défini, il sera possible d'exécuter l'application Web en plein-écran, de l'utiliser comme une application standalone, de lui assigner une icône dédiée mais également un thème spécifique. Mieux encore, sur Android, Chrome proposera à l'utilisateur d'installer l'application Web via un message affiché en bas d'écran. A l'exécution sur un smartphone Android au sein de Chrome, on obtient le résultat présenté à la figure 2 :
Nommé manifest.json, notre Web App Manifest est à ajouter au fichier index.html comme suit : <link rel="manifest" href="./manifest.json"> .
Pour notre Progressive Web App, il aura l'allure suivante :

```
{
  "short_name": "Arrivals",
  "name": "Arrivals at Marseille",
  "icons": [
    {
      "src": "launcher-icon.png",
      "sizes": "48x48",
      "type": "image/png"
    },
    {
      "src": "launcher-icon-2x.png",
      "sizes": "96x96",
      "type": "image/png"
    },
    {
      "src": "launcher-icon-4x.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "launcher-icon-8x.png",
      "sizes": "256x256",
      "type": "image/png"
    }
  ],
  "start_url": "./",
  "display": "standalone",
  "orientation": "portrait",
  "theme_color": "#29BDBB",
  "background_color": "#29BDBB"
}
```

## Service Workers

Les Service Workers constituent une des fonctionnalités les plus puissantes des Progressive Web Apps puisqu'ils permettent de proposer facilement une expérience offline poussée à l'utilisateur. Via l'emploi des Service Workers, il devient ainsi possible d'afficher les données récupérées précédemment par l'application (en utilisant IndexedDB) alors même que l'utilisateur n'est pas en ligne. Dans notre application de démonstration, nous avons choisi une approche alternative consistant à signifier à l'utilisateur qu'il n'est pas connecté à Internet et à lui permettre d'utiliser l'application avec les données précédemment obtenues. Une fois reconnecté, nous lui proposons de récupérer les dernières données depuis le serveur.
Nous allons configurer un Service Worker pour notre application. Cela nous permettra de mettre en cache les différentes ressources statiques de l'application afin de limiter l'utilisation de la bande passante tout en améliorant les performances, et ce, dès l'installation. Ensuite, lors de requêtes réseaux, nous allons préciser qu'il faut utiliser les données mises en cache




3 *Support du mode Offline*

lorsque la requête ne peut être réalisée correctement. Tout cela est réalisé au sein d'un fichier sw.js :

```
var cacheName = 'v1:static';

self.addEventListener('install', function(e) {
  e.waitUntil(
    caches.open(cacheName).then(function(cache) {
      return cache.addAll([
        './',
        './css/style.css',
        './js/build/script.min.js',
        './js/build/vendor.min.js',
        './css/fonts/roboto.woff',
        './offline.html'
      ]).then(function() {
        self.skipWaiting();
      });
    })
  );
});

self.addEventListener('fetch', function(event) {
  // utilisation du cache ou requête réseau lorsque cela est possible
  event.respondWith(
    caches.match(event.request).then(function(response) {
      if (response) {
        // cache
        return response;
      }

      // requête normale
      return fetch(event.request);
    })
  );
});
```

La dernière étape étant d'enregistrer le Service Worker au moment du lancement de l'application Web. Cela se fait au sein du fichier main.js, listé précédemment mais non détaillé jusqu'ici :

```
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register('./sw.js').then(function(reg) {
    console.log('Successfully registered service worker', reg);
  }).catch(function(err) {
    console.warn('Error whilst registering service worker', err);
  });
}

window.addEventListener('online', function(e) {
  console.log("Online");
  Page.hideOfflineWarning();
  Arrivals.loadData();
}, false);

window.addEventListener('offline', function(e) {
  console.log("Offline");
  Page.showOfflineWarning();
}, false);

if (navigator.onLine) {
  Arrivals.loadData();
} else {
  Page.showOfflineWarning();
}

ko.applyBindings(Page.vm);
```

Ici, on enregistre notre Service Worker puis on va préciser qu'il faut charger les données de la liste des avions en cours d'arrivée lorsque l'application est en ligne. Lorsqu'elle est offline, on affiche un message à l'utilisateur et on laissera le Service Worker afficher les données mises en cache. Enfin, on applique les différents bindings nécessaires à son bon fonctionnement via l'appel à la méthode applyBindings de Knockout. Pour vérifier si la gestion du mode offline est correcte, il suffit alors de lancer l'application Web en mode avion sur un smartphone par exemple. Le résultat, présenté à la figure 3, montre que le support offline est cohérent avec ce que l'on attend d'une Progressive Web App puisque l'on informe l'utilisateur tout en lui affichant les données précédemment récupérées.

## CONCLUSION

Cet article aura permis de mettre en avant les possibilités offertes par les Progressive Web Apps et la facilité avec laquelle il est possible de se lancer. Bien entendu, notre exemple peut encore être largement amélioré, avec, pourquoi pas, l'ajout de Push Notifications pour améliorer la rétention des utilisateurs, ou le recours à IndexedDB pour proposer une utilisation offline plus poussée. Soutenues par Google, qui en est d'ailleurs à l'initiative, les Progressive Web Apps offrent un compromis plus qu'intéressant entre les applications natives et les sites Web en tentant de garder le meilleur des 2 mondes. L'utilisateur final y gagne en confort d'utilisation et en performances alors que les développeurs devraient y trouver leur compte en termes de maintenance et de déploiement. •

***Ne manquez pas notre dossier complet le mois prochain***

# Comment assurer une **belle vie** à son application mobile ! (Partie 5)

Mathilde Roussel
**R&D Software Engineer chez MEGA International**
Son Blog :
http://www.mathroussel.fr

*Jason De Oliveira*
*CTO | MVP chez*
**MEGA International**
Son Blog :
http://www.jasondeoliveira.com

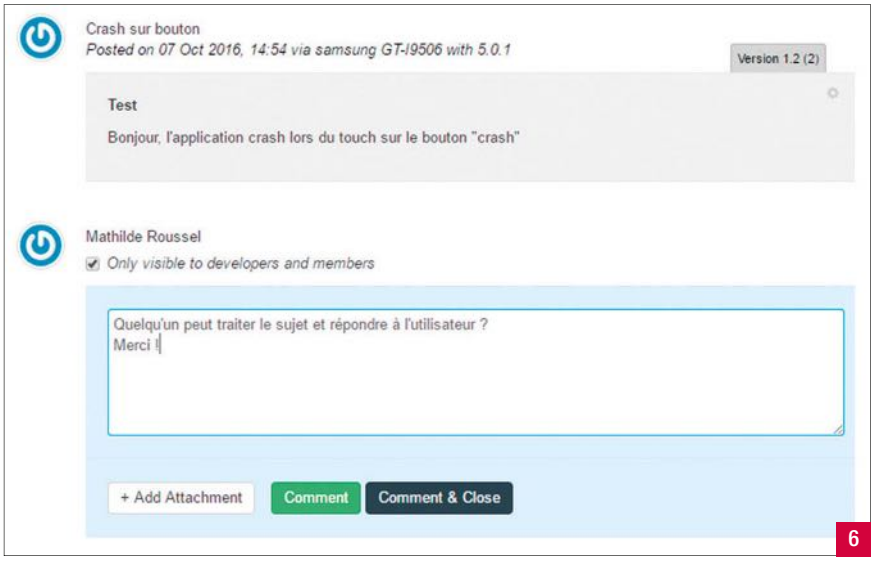
*Dans le domaine du mobile, différents outils gravitent autour d'une application dans le but de faciliter sa gestion pendant sa durée de vie, que l'on espère longue. Nous avons pu voir les outils d'intégration continue ainsi que ceux de tests dans les quatre précédents articles (N°198, 199, 200, 201). Mais si l'intégration continue, les tests unitaires et d'interface impactent en premier lieu les développeurs, comment agir à l'étape suivante, quand les utilisateurs vont devoir être confrontés à l'application ?*

## Les métriques

Comme exposé au début de l'article précédent, HockeyApp propose le rendu de différentes métriques, telles que les téléchargements et sessions. Cela reste néanmoins plutôt basique, et afin d'étoffer cela, HockeyApp a annoncé fin août 2016 la disponibilité d'évènements personnalisés. Ces évènements personnalisés permettent aux développeurs de spécifier dans le code de nouveaux triggers qui seront remontés à HockeyApp. L'ajout d'un évènement personnalisé est très simple puisqu'il prend deux lignes sous iOS et une ligne sous Android. Par exemple, dans l'application MySuperInventory Manager, un évènement va être ajouté pour détecter la tentative d'ajout d'un produit vide ou déjà existant :

```
MetricsManager.TrackEvent("product name empty or already exists");
```

Une fois cet évènement levé, les données se retrouvent dans le nouvel onglet « Events » sur la plateforme Web de HockeyApp. La remontée des informations vers cet onglet peut prendre quelques minutes, il faut donc patienter un peu avant de vérifier si l'évènement personnalisé fonctionne bien. Une fois l'attente terminée, la page « Events » regroupe de façon similaire aux crashs les évènements sous forme de liste. Pour chaque évènement, le nombre de fois où il a été levé et le nombre d'utilisateurs concernés sont affichés. Lors du clic sur un des évènements, les graphiques affichés dessous sont rafraîchis, afin de fournir quelques informations supplémentaires sur les dernières 24h, 7 derniers jours ou 30 derniers jours.




6

## Les Feedbacks

Ce quatrième menu de la plateforme Web de HockeyApp regroupe, comme son nom l'indique, tous les feedbacks donnés pour l'application, et ce pour toutes les versions. Tout comme les crashs, les feedbacks possèdent des statuts : ouvert, en attente, résolu, ignoré ou fermé. Un feedback est visible sous forme d'une discussion, sur laquelle est spécifiée pour le bêta testeur le nom, le type de périphérique, ou encore la version de l'OS. En tant que membre ou développeur de HockeyApp, il est possible de répondre à un feedback. Cette réponse pourra soit être visible uniquement par les autres membres ou développeurs (en cochant la case pourvue à cet effet), ou par tous (dont le bêta-testeur) (Fig. 6). Dans le deuxième cas, la réponse sera aussi disponible pour le bêta-testeur sur la page de feedback de l'application mobile, toujours sous la forme d'une conversation.

## La gestion des utilisateurs

Le dernier volet de l'interface Web de HockeyApp concerne les utilisateurs. Sur cette page, l'ensemble des utilisateurs de l'application est disponible, avec pour chacun son statut (en attente ou accepté). HockeyApp rend possible l'ajout d'utilisateurs un par un, mais aussi l'ajout d'une équipe. Pour ajouter une équipe complète pour la bêta d'une application, il faudra avoir au préalable créé cette équipe depuis le dashboard principal. Chaque utilisateur est invité sur l'application avec un rôle précis, qu'il soit membre, développeur ou testeur. A noter que lors de la création d'une équipe, le rôle de

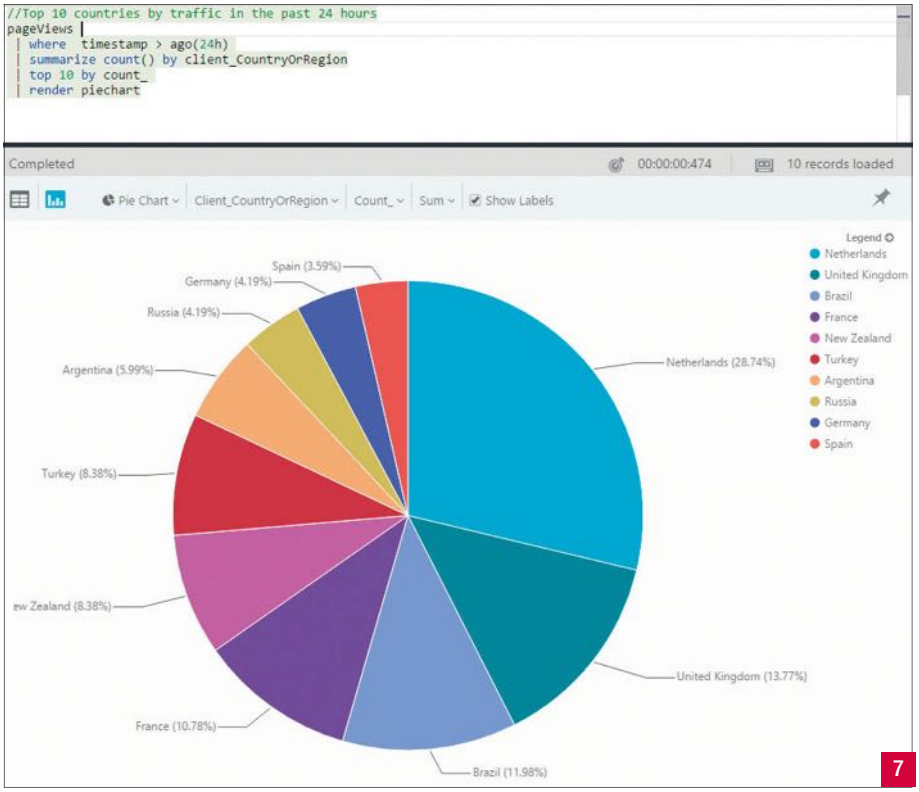
manager s'ajoute à cette liste. Il est possible d'affiner les rôles cités précédemment, à l'aide de tags, qui vont servir pour les restrictions de téléchargement. Lorsqu'un utilisateur valide son inscription et télécharge sur son téléphone l'application HockeyApp (qui sert de plateforme pour télécharger les applications), les informations concernant son téléphone sont transmises à HockeyApp. Cela facilite ainsi le suivi des appareils sur lesquels l'application est testée. Le détail d'un utilisateur fournit aussi la liste des versions qu'il a récemment téléchargées.

## L'intégration de HockeyApp dans MS Application Insights

Si HockeyApp a pour but de remplacer un jour Application Insights pour les applications mobiles, ce dernier reste un outil plus éprouvé et plus plébiscité par les développeurs. Afin de satisfaire les habitudes des développeurs, et, en attendant que la plateforme de HockeyApp acquière la pleine capacité de ses pouvoirs, les équipes de la plateforme ont annoncé fin août 2016, en même temps que les évènements personnalisés, la possibilité de lier HockeyApp et Application Insights. Cette fonctionnalité s'appelle HockeyApp Bridge App, et permet d'accéder aux données de HockeyApp depuis Application Insights. La création d'un bridge se fait via le portail Azure, en créant une nouvelle ressource Application Insights, qui aura pour type « HockeyApp Bridge Application ». Il faut préciser la clé d'API HockeyApp dans les paramètres utilisateurs, dans le menu API Tokens, ainsi que l'application concernée par le bridge.
Application Insights regroupe ainsi non seulement les données en provenance de HockeyApp mais aussi d'autres informations non disponibles à l'affichage, et permet une analyse plus poussée de ces données. Le requêtage des données se fait grâce à un langage proche du SQL, pour un résultat produit sous forme de tableaux ou de graphiques. Il est par exemple possible de voir le nombre de pages vues par pays (Fig. 7).

## Pour aller plus loin…

Nous avons déjà vu dans le premier article de la série (N° 198) que lors du déploiement continu HockeyApp s'intègre facilement avec Visual Studio Team Services. Mais ce n'est pas tout, car il existe un widget pour le dashboard de VSTS. Il est gratuit et configurable facilement puisqu'il requiert uniquement l'appId de l'application HockeyApp. Ce Widget fournit trois liens rapides, vers la page de résumé de l'application, la page de téléchargement, et enfin la page de



7

8

statistiques des crashs (sous forme de graphiques) (Fig. 8).
Lorsqu'une application disponible sur HockeyApp est liée avec Application Insights, toutes les données sont exportées via la fonctionnalité « Continuous Export » vers un stockage Azure. Grâce à ces données, il est possible de tirer parti de briques Azure telles que le Machine Learning par exemple.
Afin de profiter des données fournies par HockeyApp dans d'autres applications, les équipes de HockeyApp proposent des WebHooks. Ces WebHooks fournissent un moyen d'être notifié par des requêtes HTTP à propos de crashs, nouvelles versions ou feedbacks. Il est donc possible grâce à cela d'intégrer HockeyApp dans Slack, un outil de messagerie instantanée de plus en plus utilisé dans les entreprises.
Enfin, HockeyApp permet l'envoi de mails quotidiens pour résumer l'activité de la veille. Cette option est configurable depuis les paramètres de l'utilisateur, avec plusieurs catégories et différentes fréquences possibles (individuel, quotidien, les deux, ou désactivé).

# CONCLUSION

Nous avons vu tout au long de ces articles comment une application mobile pouvait s'armer pour affronter l'épreuve des bêta-tests. Grâce aux outils Microsoft tels que HockeyApp ou Visual Studio Team Services, les développeurs disposent de moyens efficaces pour déployer une application en bêta, récupérer les informations de crashs, des évènements personnalisés, ou encore des feedbacks. Les bêta-testeurs profitent de l'application HockeyApp qui leur permet de mettre à jour facilement une application pour les tests, et profitent aussi d'interactions facilitées avec les développeurs. Cet article clôt ainsi la série consacrée au DevOps mobile, afin d'assurer la réussite sur le long terme d'un projet Xamarin. L'ensemble des outils et bonnes pratiques présentées vous permettront de vous engager sereinement sur le champ de bataille qu'est la mise en production sur les stores d'application ! •

# Une **application Electron** pour suivre un repository GitHub

Partie 3

• Philippe Charrière
Solution Engineer chez **GitHub** (@k33g)
Twitter: @k33g_org

## NOTRE APPLICATION LES COMMITS DES COPAINS

Cette application a un **gros défaut** (au moins un) : si quelqu'un propose une pull request, vous ne serez pas notifié de ses modifications. Pour cela il faut utiliser une autre API de GitHub qui permet de détecter les events sur un réseau de repository. C'est à dire sur un repository et ses **forks**.(https://developer.github.com/v3/activity/events/#list-public-events-for-a-network-of-repositories). Et cette fois-ci, cela reviendra à effectuer une requête de ce type :

```
GET /networks/:owner/:repo/events
```

La modification de notre application pour prendre ceci en compte est extrêmement simple :
Dans main.js, il suffit de changer la ligne :

```
githubCli.getData({path: `/repos/${owner}/${repo}/events`}).then(response => {...})
```

par

```
githubCli.getData({path: `/networks/${owner}/${repo}/events`}).then(response => {...})
```

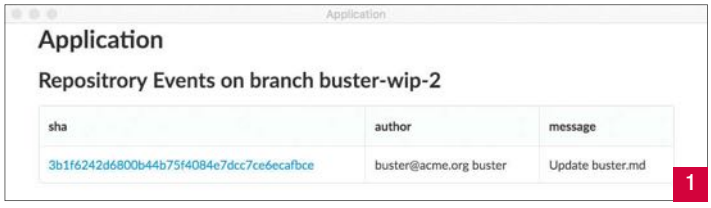
Vous pouvez à nouveau tester.
Si un autre utilisateur fork votre projet et fait une pull request, vous serez ensuite notifié de ses commits.

**une mise à jour par un autre user** : [Fig.1]
Maintenant, je voudrais afficher la liste des fichiers des différents commits (le code qui va suivre peut-être optimisé, j'en suis bien conscient, de même que l'ergonomie de l'IHM).

## Modifions à nouveau main.js

Pour chacun des commits, je vais aller chercher les informations relatives aux fichiers liés au commit grâce à l'API https://developer.github.com/v3/repos/commits/#get-a-single-commit, une requête de type GET (GET /repos/:owner/:repo/commits/:sha) qui permet de récupérer un commit par son sha.

```
// ...
if(lastEvent.type == "PushEvent") {
  let ref = lastEvent.payload.ref.split("refs/heads/")[1];

  let filesList = []; // permettra de stocker tous les fichiers des commits
  let counter = 0;

  /*
    - à chacun synchronisation, je récupère le handle de l'utilisateur
    - puis pour chaque commit j'interroge l'API GitHub GET /repos/:owner/:repo/commits/:sha pour obtenir les infos du commit courant
    - je merge la liste des fichiers du commit avec `filesList`
    - j'incrémente mon compteur
    - une fois que le compteur est égal à la taille tu tableau de commits,
      => j'ai fini mes requêtes
      => je notifie mon IHM avec les informations
  */
  let handleNameOfCommitAuthor = lastEvent.actor.login;

  lastEvent.payload.commits.forEach(commit => {

    githubCli.getData({path: `/repos/${handleNameOfCommitAuthor}/${repo}/commits/${commit.sha}`})
      .then(response => {
        let filesOfCommit = response.data.files;

        filesList = filesList.concat(filesOfCommit)
        counter++
        if(counter==lastEvent.payload.commits.length) {
          // je notifie mon IHM
          sender.send("PushEvent", {ref: ref, commits: lastEvent.payload.commits, filesList: filesList})
        }
      })
  })
}
// ...
```



1

## Modifions à nouveau le tag grid-events.htm

Pour prendre en compte les nouvelles données, je modifie mon tag :

- J'ajoute une nouvelle table pour afficher la liste des fichiers tous commits confondus ;
- Je modifie la portion de code de notification this.filesList = arg.filesList pour prendre en compte les nouvelles données.

```
<grid-events>
  <h2>Repositrory Events on branch {ref}</h2>

  <h3>Commits:</h3>
  <table class="ui celled table">
    <thead>
      <tr>
        <th>sha</th>
        <th>author</th>
        <th>message</th>
      </tr>
```

```
  </thead>
  <tbody>
   <tr each={commits}>
    <td><a href={url} target="_blank">{sha}</a></td>
    <td>{author.email} {author.name}</td>
    <td>{message}</td>
   </tr>
  </tbody>
 </table>

 <h3>Files:</h3>
 <table class="ui celled table">
  <thead>
   <tr>
    <th>sha</th>
    <th>filename</th>
    <th>status</th>
   </tr>
  </thead>
  <tbody>
   <tr each={filesList}>
    <td>{sha}</td>
    <td>{filename}</td>
    <td>status:{status} additions:{additions} deletions:{deletions} changes:{changes}</td>
   </tr>
  </tbody>
 </table>

 <script>
  opts.ipcRenderer.on('PushEvent', (event, arg) => {
   this.ref = arg.ref
   this.commits = arg.commits
   this.filesList = arg.filesList
   this.update();
  });
 </script>
</grid-events>
```

Vous pouvez tester avec votre user ou un autre user :
**un commit par un autre user** : [Fig.2]
**un commit multi fichiers** : [Fig.3]

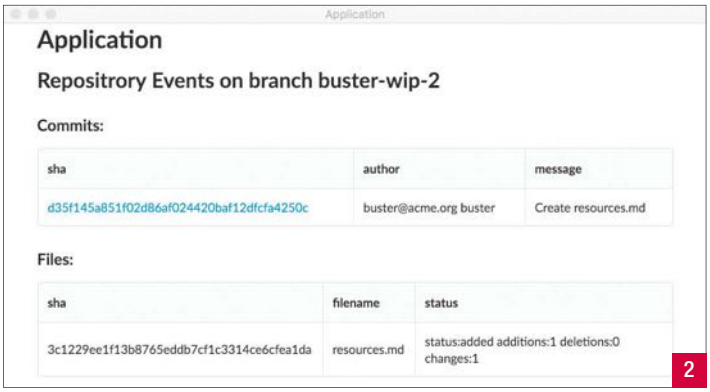
## CHANGER LE STATUT DE LA PULL REQUEST

Pour compléter l'application, je souhaiterais faire des vérifications de contenu de la pull request, et afficher une erreur lorsque l'utilisateur propose autre chose que des fichiers markdown. Pour cela je vais utiliser l'API "Statuses" https://developer.github.com/v3/repos/statuses/ (POST /repos/:owner/:repo/statuses/:sha). Donc pour modifier le statut d'un commit, j'utiliserai ce type de code :

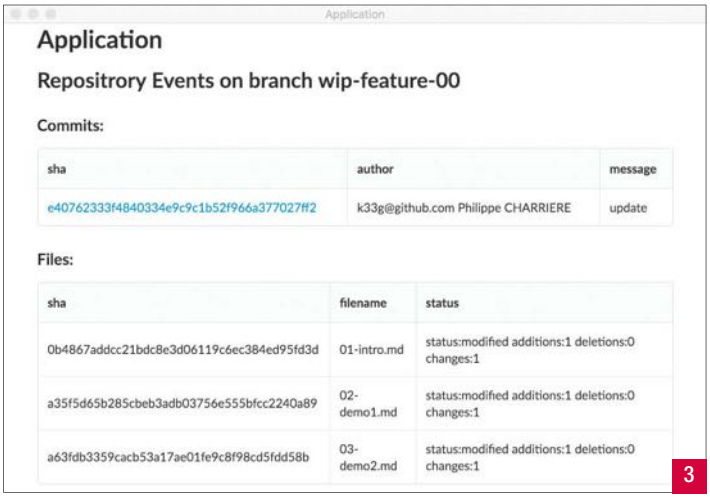
```
githubCli.postData({
 path: `/repos/${owner}/${repo}/statuses/${commit.sha}`,
 data: {
  state: "success",
  description: "you're the best"
 }
})
```

Donc, modifions encore et encore main.js

```
if(lastEvent.type == "PushEvent") {
 let ref = lastEvent.payload.ref.split("refs/heads/")[1];

 let filesList = [];
 let counter = 0;

 let handleNameOfCommitAuthor = lastEvent.actor.login;

 lastEvent.payload.commits.forEach(commit => {
  githubCli.getData({path: `/repos/${handleNameOfCommitAuthor}/${repo}/commits/${commit.sha}`})
   .then(response => {
    let filesOfCommit = response.data.files;

    // je parcours la liste des fichiers
    filesOfCommit.forEach(file => {
     // je teste l'extension du fichier
     if(file.filename.split('.').pop()=="md") {
      // si c'est du markdown, le check est valide
      githubCli.postData({
       path: `/repos/${owner}/${repo}/statuses/${commit.sha}`,
       data: {
        state: "success",
        description: "you're the best"
       }
      })
     } else {
```



2



3

```
      //si ce n'est pas du markdown, le check est invalide
      githubCli.postData({
       path: `/repos/${owner}/${repo}/statuses/${commit.sha}`,
       data: {
        state: "failure",
        description: "ouch!"
       }
      })
     }
    })

    filesList = filesList.concat(filesOfCommit)
    counter++
    if(counter==lastEvent.payload.commits.length) {
     console.log(filesList)
     sender.send("PushEvent", {ref: ref, commits: lastEvent.payload.commits, filesList: filesList})
    }
   })
  })
}
```

Vous pouvez à nouveau tester.

**Tout va bien :)** [Fig.4]

**@buster n'a pas respecté les règles!** [Fig.5]

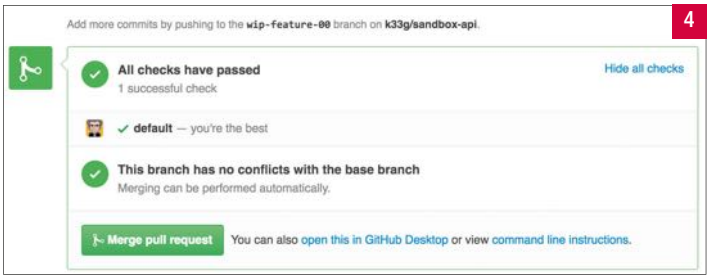
Voilà, c'est tout pour aujourd'hui. Si vous avez des questions, vous pouvez me joindre ici : https://github.com/k33g/q/issues

## QUELQUES IDÉES POUR ALLER PLUS LOIN

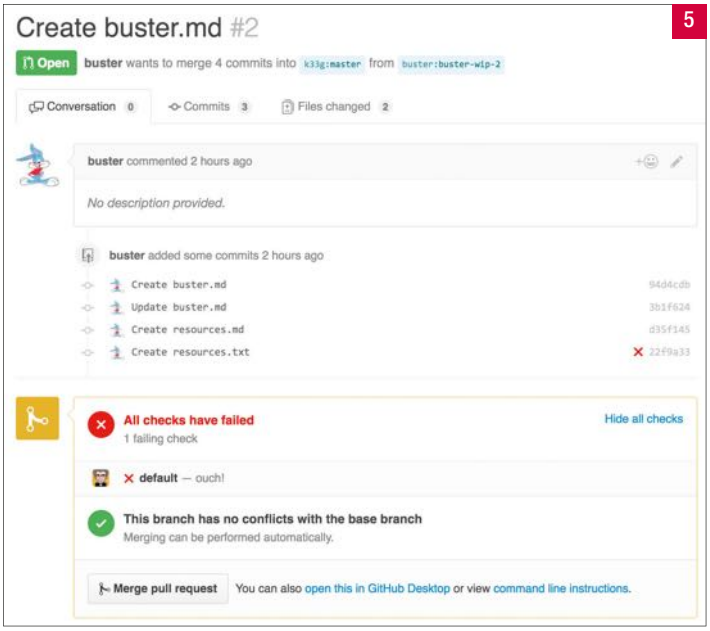
- Utiliser l'API **Tray** d'Electron pour afficher les notifications (https://github.com/electron/electron/blob/master/docs/api/tray.md) ;
- Faire une IHM d'administration ;
- Pouvoir "scruter" plusieurs repositories ;
- Extraire/Lire le contenu des fichiers pour faire des vérifications "plus poussées" (utilisation de l'API contents) ;
- Utiliser les webhooks pour être notifié (votre application devra être "visible" de l'extérieur, mais plus de limite sur le nombre d'appels) ;
- Faire un serveur d'intégration continue (mais si!) ;
- Optimiser le code (notamment sur l'utilisation des promises).

Vous trouverez le code définitif du projet par-là :
https://github.com/k33g/check-my-repo

4




5



# Créer des sites Internet responsive avec **Drupal** et **Zurb Foundation**

Partie 2

• Thomas Lepérou
*Consultant en Ingénieur logiciel et diplômé de l'ESGI, je suis spécialisé dans le développement d'application Web fullstack JS et le site building depuis Drupal 6.*

*La création de sites Internet, pour certains de nos lecteurs, c'est une affaire de plusieurs décennies. Mais aujourd'hui les évolutions engendrées par le responsive - et plus globalement le cross-plateforme - ont radicalement changé la manière de les créer : material design, frontend-development, headless CMS, Sass, Grunt, Bower…*

## FRONT-END RESPONSIVE

Pour constater l'évolution de la mise en place du front-end, il est préférable d'ajouter du contenu. Le module Devel sera très utile pour en générer automatiquement. Créez au préalable quelques terms de taxonomy. Installation de certains modules :

```
drush en -y views panels instantfilter clean_markup
drush en -y views_ui panels_mini page_manager views_load_more views_content
    clean_markup_panels
```

## Pour commencer

Clean Markup permet de définir les markups des panes (les contenus insérés dans les régions du panel layout). Commençons par créer notre page d'accueil :

- Créez sous Page Manager (admin/structure/pages/add/) un custom page "Accueil", au path /accueil, définie comme page d'accueil, de variant type panels, avec Clean Markup comme layout "Clean Markup Reset".
- Exclure l'affichage des blocs de la sidebar (/admin/structure/block) pour le path <front>
- Désactivez le topbar de Zurb Foundation dans admin/appearance/settings/[theme_name]
- Créez une view Articles, avec les fields title et image. Une pager Load more pager, et activez l'ajax.

## La base de Zurb Foundation

Un instant pour se rappeler des Grids et classes utiles à l'intégration qui suit.

### Grids

Pour obtenir ce résultat entre smartphone (small), tablette (medium) et desktop (large), il suffit de poser cette structure :

```
<div class="row">
  <div class="medium-6 large-4 columns"></div>
  <div class="medium-6 large-4 columns"></div>
  <div class="medium-12 large-4 columns"></div>
  <div class="medium-12 columns"></div>
</div>
```

### Utilities

**Les alignements** : .left, .right et .clearfix; pour ancrer votre DOM à gauche ou à droite.
**Les rounded** : .round, .radius.
**Les alignements de text** : .text-center, .text-right, .text-left …
**Les visibilities** : .show-for-small-only, .show-for-medium-up; pour afficher des composants en fonction des media query-ranges.
Les button, inline-list, side-nav, form, dropdown et bien d'autres.

## Off-canvas

Commençons par la navigation principale. Le off-canvas est un espace qui par défaut n'est pas visible. Vous pouvez l'utiliser pour des navigations responsives, car cet espace s'affiche au clic d'un lien, d'un bouton ou directement en Javascript. Le template se présentera ainsi :

Éditons le template page.tpl.php, en copiant collant celui-ci depuis zurb_foundation/templates/page.tpl.php dans templates/page.tpl.php de votre thème.

```
cp ../zurb_foundation/templates/page.tpl.php templates/page.tpl.php
```

Ajoutons ce code HTML pour intégrer le off-canvas de Zurb Foundation.

```
<div class="off-canvas-wrap" data-offcanvas>
 <div class="inner-wrap">

  <a class="left-off-canvas-toggle" href="#" >Menu</a>

  <!-- Contenu du off-canvas -->
  <div class="left-off-canvas-menu">
          <?php print render($page['offcanvas']); ?>
  </div>

  <!--.page -->
  <div role="document" class="document"><!-- [...]--></div>
  <!--/.page -->

  <!-- close the off-canvas menu -->
  <a class="exit-off-canvas"></a>

 </div>
</div>
```

Nous avons ajouté une nouvelle région ($page['offcanvas']), il est nécessaire de la renseigner dans *[theme_name]*.info de votre thème

```
regions[offcanvas]        = Offcanvas
```

Puis de vider les caches.

```
drush cc all
```

**Note**: Ajoutez sur n'importe quel lien la classe left-off-canvas-toggle et un href="#" pour l'associer au off-canvas et ainsi le faire apparaître. Pensez donc à ajouter un lien dans un endroit accessible (ex: position: fixed, au top du viewport).

## Navigation principale

Créons un Mini Panel appelé *Main menu desktop*, dans la catégorie Components, avec Clean Markup Reset comme layout. Insérez-y le main menu, et mettez comme style pour la region et le pane *Clean Markup* à *-- No wrapper --*.
Puis, opérez de la même manière, mais en l'appelant *Main menu mobile*.
Créez dans le dossier scss/components le fichier _main-menu.scss, saisissez l'@import dans *_[theme_name].scss*. Ajoutez dans le *_main-menu.scss* :

```
.menu {
 @include inline-list;
 li a {
  @extend button;
 }
}
```

Vous venez de créer votre propre inline-list en utilisant le mixin de Zurb Foundation, avec des a héritant des styles de button. Consultez la documentation de Zurb sur leur Semantic markup with SASS.
Depuis la page des blocs /admin/structure/block, affectez à la region header le Mini Panel Main menu desktop et le Mini Panel Main menu mobile dans la region Off-canvas.

## Navigation adaptive avec Browscap

Dans votre console, tapez :

```
drush en -y browscap
drush dl browscap_ctools-7.x-1.x-dev
drush en -y browscap_ctools
```

Allez dans /admin/config/system/browscap, et Refresh browscap data.
Ensuite, allez dans le Content du Mini Panel Main menu desktop, ajoutez une règle de visibilité pour le pane Main menu. Sélectionnez Is Mobile, next, cochez Reverse (NOT). Cela aura pour effet de ne rendre (render()) le pane que si la requête est émise depuis un ordinateur (is NOT mobile). Il ne s'agit pas de le cacher, mais bien de générer l'HTML en fonction du support qui le demande.
Opérez de la même manière, mais pour la version mobile.
Vous avez maintenant, une navigation principale qui apparaît soit dans le off-canvas pour un mobile, soit dans le header pour un desktop.
Ajoutez dans un Mini Panel (avec *Clean Markup* à *-- No wrapper --*) un *New custom content* qui contient le lien pour faire apparaître le off-canvas, qui n'apparaîtra que sur la version mobile (avec la *Visibility rule* à *Is mobile*). D'abord,

```
drush en -y fontawesome
```

Ensuite le lien (attention à choisir un text filter qui autorise la base i et a) :

```
<a class="button left-off-canvas-toggle" href="#"><i class="fa fa-bar"></i></a>
```

Ajoutez ce Mini Panel depuis la page d'administration des blocs dans la region header.

## Ajouter un Slideshow responsive avec Slick

Téléchargez la librairie Slick dans le dossier vendor, depuis **http://kenwheeler.github.io/slick/**, ajoutez-la dans l'array jsLibs dans Gruntfile.js. Nous intégrons manuellement la librairie, mais un module Drupal slick existe.
Ajoutez le css de slick au *[theme_name]*.info et videz les caches :

```
stylesheets[all][] = vendor/slick/slick/slick.css
stylesheets[all][] = vendor/slick/slick/slick-theme.css

drush cc all
```

Ajoutez ces lignes à js/_[theme_name].js :

```
$('.slick-it.centered .view-content').addClass('slick centered');
$('.slick.centered, .panel-pane.slick-it.centered').slick({
  arrows: true,
  centerMode: true,
  centerPadding: '200px',
  slidesToShow: 1,
  responsive: [
    {
      breakpoint: 440,
      settings: {
        slidesToShow: 1,
        slidesToScroll: 1,
        centerMode: false
      }
    },
    {
      breakpoint: 1024,
      settings: {
        slidesToShow: 2,
        centerPadding: '75px'
      }
    }
  ]
});
```

Dorénavant, le simple usage de slick-it centered comme classe sur une view transformera son contenu en un slideshow responsive (pour le format unformated list), idem pour les regions des layouts de panels (à l'aide de Clean Markup).
Ajoutez un display Content pane à la views des Articles, qui affiche les 5 derniers articles. Au format Undefined list. Définissez le Class css en bas de l'onglet advanced à slick-it centered.
Tout cela pour ce display seulement. Affichez les fields title et image.
Ajoutez un contenu à votre custom page Accueil : *Views: Articles: 5 last slick* (tel que j'ai nommé dans views) présente maintenant dans la catégorie Views panes. Modifiez le style du pane à Clean panel markup et sélectionnez -- No wrapper --.
Vous avez maintenant un slideshow responsive sensible au touch event.

## Créer une liste d'article en Cards

Créons un component, Cards, qui se présentera ainsi :


Consultez la documentation de Google Material Design pour davantage d'informations sur le Card. Reprenez la view Articles, le premier display créé. Mettez tous les fields en Exclude from display. Ajoutez un field Content: Link, exclude from display aussi, avec "Lire l'article" comme text to display, et button comme class. Ajoutez un field, Global: Custom text, dans lequel vous reprenez tous les champs précédents ainsi (en fonction des champs que vous avez utilisés) :

```
<div class="inner">
 <div class="title"><h3>[title]</h3></div>
 <div class="image">[field_image]</div>
 <p>[body]</p>
 <div class="actions">[view_node]</div>
</div>
```

Ajoutez la class card card-4 au row, dans les settings de Format, et désactivez les classes par défaut dans les settings de Fields. Enfin, ajoutez la class cards dans CSS Class de l'onglet advanced du display (pour ce display). Créez la feuille scss _cards.scss dans components, qui contiendra :

```
.view.cards {
 .view-content {
  @include grid-row();
  .card {
   margin-bottom: $column-gutter;
  }
  .card.card-4 {
   @include grid-column(4);
  }
  .card.card-6 {
   @include grid-column(6);
  }
  .card.card-12 {
   @include grid-column(12);
  }
  // Rendre les images responsive, et mettre une hauteur fixe
  .inner {
   &:hover { box-shadow: $onsoon 0 0 1rem; }
   padding: $column-gutter/2; // divise par 2 la valeur de $column-gutter
   background: $white;

   // Rendre les images responsive, et mettre une hauteur fixe
   .image {
    max-height: rem-calc(250);  // transforme les px (250) en rem
    overflow: hidden;
    img {
     width: 100%;
     height: auto;
    }
   }
  }
  // permet de couper avec "..." le titre s'il est trop long
  .title h3 {
   text-overflow: ellipsis;
   overflow: hidden;
   white-space: nowrap;
  }
 }
}
```

Dorénavant, utilisez cette structure HTML dans vos views, avec cards dans CSS Class et card card-6 (ou card-4, ou card-12) comme Row class pour réaliser un affichage en cards d'une largeur de 6 columns (6 sur un total de 12 pour le grid par défaut).

## Afficher des cards sur l'accueil

Avec Views, nous allons créer des listes comme celle-ci :


Revenez sur votre view Articles, créez un nouveau display Content pane (assurez-vous qu'il hérite des propriétés enregistrées ci-dessous) appelé cards home. Override le settings de Format à card card-4, et dans Allow settings, cochez Items per page.

Utilisez les class card card-4 pour un affichage en trois colonnes, card card-6, en deux et card card-12, en une.

Dans le custom page Accueil, ajoutez ce dernier Content pane, présent dans la catégorie Views panes. Mettez à 3 le nombre d'items. Puis pour son style, Style > Clean Panel Markup > -- No wrapper --.

À ce stade, vous avez avec votre thème :

- Une navigation responsive et adaptive, avec un espace off-canvas ;
- Des contenus slickables à l'infini (avec les class slick-it centered) ;
- Des listes de contenus en Cards pour toutes vos views.

## Créer des layouts responsive pour Panel

Nous avons vu jusque-là comment constituer notre structure avec page.tpl.php et avec views. Réalisons maintenant un template pour Panels. Il se présentera ainsi :

Avec trois cellules, nous avons déjà de nombreuses possibilités. Créez les dossiers layouts/custom_3cel depuis la racine de votre thème, et éditez-y custom_3cel.inc :

```
<?php
/**
* Implements hook_panels_layouts()
```

```
*/
function programmez_custom_3cel_panels_layouts() {
 $items['custom_3cel'] = array(
  'title' => t('Row uncollapsed, 3 cels'),
  'category' => t('CUSTOM LAYOUTS'),
  'icon' => 'custom_3cel.png',
  'theme' => 'custom_3cel',
  //'admin css' => '../foundation_panels_admin.css',
  'regions' => array(
   'cel1' => t('Cellule 1'),
   'cel2' => t('Cellule 2'),
   'cel3' => t('Cellule 3')
  ),
 );
 return $items;
}
```

Puis custom_3cel.tpl.php

```
<div class="row">
   <?php print $content['cel1']; ?>
   <?php print $content['cel2']; ?>
   <?php print $content['cel3']; ?>
</div>
```

Remarquez que l'on retrouve les mêmes keys cel1, cel2 et cel3 de l'array $items['custom_3cel']['regions'] entre le .tpl.php et le inc.php. Le hook _panels_layouts pour le layout _custom_3cel du theme programmez permet d'ajouter ce layout à la liste présentée aux panels de leur section layouts.

**Note**: Videz les caches lorsque vous créez de nouveaux templates pour Panels.

```
drush cc all
```

**Debug** : Si votre template n'apparaît pas dans la liste, il est certain que le hook est mal défini. Contrôlez le nom des fichiers, du dossier ainsi que la valeur de : $items['custom_3cel']['theme'].

## Éditer le template des nodes avec Page Manager

Commencez par activer le template sous Page Manager, node_view. Add new variant, Article, de type Panel, avec a selection rules. Ajoutez la rule Node: type, pour le context Node being viewed au type de node Article. Cette variante là, sera exécutée lorsqu'un node de type article sera parcouru depuis l'adresse /node/[nid].

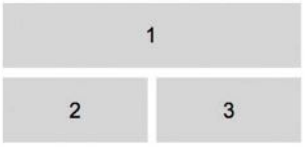
J'ai créé une view pour lister les derniers articles en rapport au term de taxonomy du Node being views (via une relation sur taxonomy term). Dans Allow settings de la view, Override title et Items per page sont cochés.

J'ai également ouvert les commentaires aux visiteurs anonymes (/admin/people/permissions).

Nous définissons la structure du contenu comme nous le ferions dans un fichier .tpl.php, mais directement depuis Page Manager. Pour cela, ajoutez un style Clean panel markup à chaque region (engrenage en haut à gauche de la region) :

- Cellule 1 : tag wrapper DIV, et class columns medium-12
- Cellule 2 : tag wrapper DIV, et class columns medium-6
- Et Cellule 3 : tag wrapper DIV, et class columns medium-6

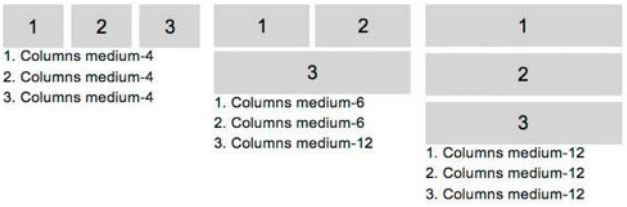
Nous obtiendrons ainsi cette présentation :




- Insérez dans la Cellule 1 les contenus du node (image, body, ...),
- Dans Cellule 2, les commentaires (form & comments)
- Dans Cellule 3, la view des articles en rapport.

**Note** : Pour avoir en admin l'affichage des regions comme celui de la présentation en front-end, il faudra éditer //'admin css' => '../foundation_panels_admin.css', et définir dans la feuille de style les règles nécessaires au bon affichage. Je vous invite à consulter la documentation Panels 3: creating a custom layout in your theme (https://www.drupal.org/node/495654).

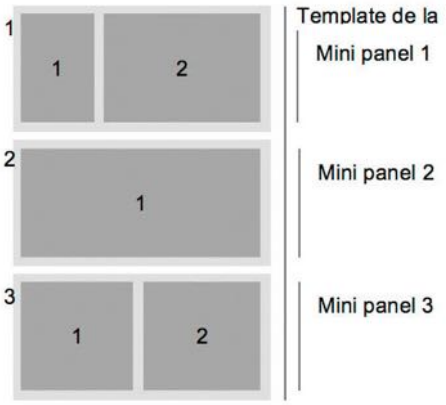
D'autres exemples :




## La flexibilité de Mini Panel

En utilisant Mini Panel et Clean Markup, vous pouvez construire de nombreuses structures de contenu, sans éditer aucune ligne de code. Mini Panel utilise les rules, les layouts et surtout, les contexts !

Imaginons un template plus élaboré :




Template de la page :

- Cellule 1 : columns medium-12
  Mini panel 1
    - Cellule 1 : columns medium-4
    - Cellule 2 : columns medium-8
- Cellule 2 : columns medium-12
  Mini panel 2
    - Cellule 1 : columns medium-12
- Cellule 3 : columns medium-12
  Mini panel 3
    - Cellule 1 : columns medium-6
    - Cellule 2 : columns medium-6

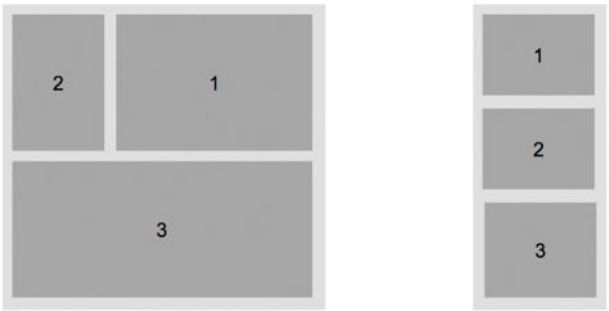
Reste seulement à ajouter le Mini Panel à votre page.

Lorsque vous souhaitez afficher des informations d'un node, ou d'un user

dans un Mini Panel, ajoutez-les au context dans la section Context. Dès lors, vous aurez accès au Content de votre context, notamment les fields.
**Debug**: lorsque vous essayez d'ajouter un Content à une région de votre layout Panel, sachez que seuls les Mini Panels (cela s'applique également au View Content Pane) qui peuvent être utilisés sont ceux qui fonctionnent avec le Context du Panel dans lequel vous vous trouvez. En d'autres termes, si un Mini Panel a un Context "User", qui afficherait son nom et le lien de déconnexion par exemple, il ne sera visible que si au context, où vous souhaitez l'ajouter (dans un custom page, ou un autre Mini Panel), est associé un User.

## Aller plus loin dans la structure de contenu responsive

Zurb Foundation avec les Grids propose les classes push et pull. Ils prennent leur utilité lorsque vous souhaitez faire apparaître une colonne (1), disons des résultats de recherche, après une première colonne (2), les filtres de recherche. Sur un desktop la lecture des informations est cohérente. Sur un mobile par contre, les filtres seront situés en première ligne, avant les résultats donc. Ce qui peut être incohérent. Les résultats de la recherche étant souvent l'information principale de la page.
Nous pourrions obtenir ce résultat :




Pour cela, nous utiliserons les classes push et pull. Les informations les plus importantes doivent apparaître en premier sur le DOM, pas seulement visuellement.

```
<div class="row">
  <div class="medium-8 medium-push-4 columns">
    <!-- résultats de recherche
  </div>
  <div class="medium-4 medium-push-8 columns">
    <!-- filtres de recherche
  </div>
  <div class="medium-12 columns"><!-- autre contenu --></div>
</div>
```

Visuellement, les filtres apparaissent en premier, dans le DOM il est en second.

## Images adaptives et responsives

### Interchange ou picture

Charger une image qui convient au support qui la demande est un facteur considérable à l'optimisation de votre site Internet. Deux possibilités évidentes s'offrent à vous : le module Interchange de Zurb Foundation (https://www.drupal.org/project/zurb_interchange) ou le module Picture (https://www.drupal.org/project/picture).
**Interchange** exige peu d'effort de configuration. La limitation fonctionnelle qui me dérange est qu'il charge l'image par défaut, puis celle correspondant au breakpoint. Nous pourrions mettre un thumbnail comme image par défaut pour limiter l'usage de la bande passante. Cela se présente ainsi, après que le module ait généré l'image :

```
<img data-interchange="[/path/to/small.jpg, (small)], [/path/to/bigger-image.jpg, (large)]">
```

Pour le breakpoint small apparaîtra l'image /path/to/small.jpg et pour le large, /path/to/bigger-image.jpg. Le module renseigne data-interchange pour les images auxquelles vous avez activez Interchange (depuis Manage display d'un content type, ou d'un field de type image dans un Panel).
**Picture** est beaucoup plus complet et nécessitera davantage d'effort à le configurer. Vous définissez vos breakpoints et associez les styles d'image pour certains sets de breakpoints. Utilisez ensuite Picture comme display formatter pour vos images et choisissez l'une des configurations que vous aurez créées.

### Magnific Popup

Il existe de nombreuses librairies qui permettent d'afficher vos images en mode *gallerie* ou plus simplement le détail d'une image dans un popup. Magnific Popup est léger et très simple. Il est évidemment responsive.
https://www.drupal.org/project/magnific_popup

### EasyZoom

EasyZoom permet de créer un effet de zoom au passage de la souris sur une image. Souvent utilisé pour visualiser les produits des sites E-commerce. Cela permet en effet d'afficher davantage l'image, tout en occupant le même espace et de conserver la hiérarchie de l'information de votre page. Il fonctionne également avec le tactile.

## SIMPLES OPTIMISATIONS

Votre thème intègre déjà de nombreuses solutions quant aux problématiques de responsive (côté client) et adaptive (côté serveur).
- Grid ;
- Navigation, off-canvas ;
- Layouts, cards ;
- Visibility rules de Panel pour les mobiles ;
- Images adaptive.

Il reste néanmoins quelques pistes d'optimisation à explorer.

## Lazy pane

https://www.drupal.org/project/lazy_pane

Lazy Pane est un super module. Chargez vos Panes en ajax lorsque ces derniers apparaissent dans le viewport. Vous diminuerez considérablement les temps de chargement en ne téléchargeant que les ressources affichées.

## Lazyloader

https://www.drupal.org/project/lazyloader

Lazyloader charge les images lorsqu'elles apparaissent dans le viewport. J'ai déjà rencontré des conflits avec certains modules, mais globalement, c'est un excellent module qui exige peu de configuration pour fonctionner.

## Cache

Enfin, pensez à activer les caches. Souvent la hantise des développeurs, les caches sont néanmoins essentiels au fonctionnement d'un site Internet en production.
- Activez les caches de Drupal (admin/config/development/performance) ;
- Activez les caches des panes de vos Panels ;
- Activez les caches de vos views.

# Introduction au Web Scraping

Partie 2

Michael Bacci
michael.bacci.software@gmail.com
https://www.linkedin.com/in/michaelbacci
*Senior Software Engineer R&D*
*Expert en HPC, C/C++, J2EE, 3D*
*Michael travaille dans des projets R&D pour l'industrie et instituts de recherche : universités Paris 13, Paris 7 et INRIA*

*Pour avoir les dernières news en temps réel, il suffit de regarder Twitter ou bien une application d'actualités sur son propre smartphone.*
*Certaines applications peuvent être configurées pour chercher certains contenus plutôt que d'autres. Mais comment aller au-delà des limites de ces outils (applications et site Web) ? Comment est-il possible de chercher et extraire n'importe quelle information depuis le net ?*
*Le Web Scraping est la réponse.*

## Simulation d'un cas réel

Examinons maintenant le modus operandi d'un développeur travaillant sur un projet de Web Scraping.

L'objectif est le suivant : créer un script qui extrait les titres des dernières actualités de programmez.com depuis la page www.programmez.com/actualites

Allumez votre ordinateur ! Voici les étapes à franchir :

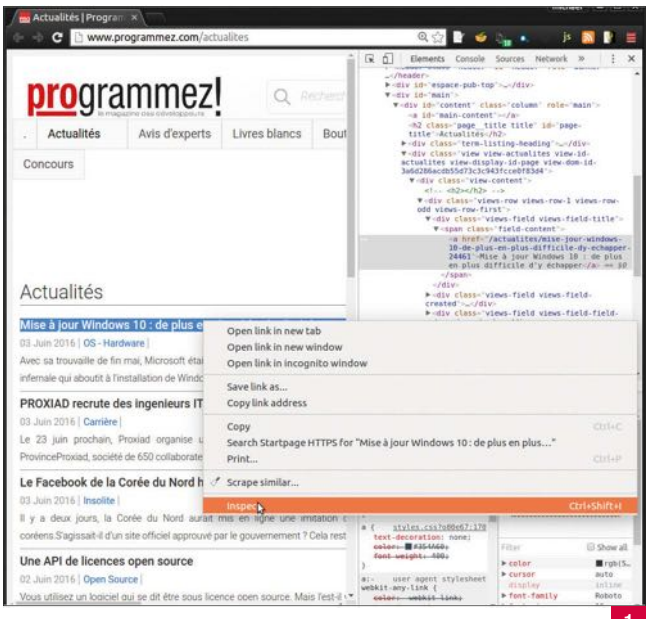
- Ouvrez votre navigateur Web préféré, dans mon cas ce sera Chrome, et allez à l'adresse indiquée ci-dessus ;
- Positionnez le curseur sur le premier titre de l'actualité, click droiteet "inspect element" [Fig. 1] ;
- Maintenant que le but du code est bien visible, il faut étudier la structure du code HTML :
  - Trouver les éléments qui ont la même structure, c'est à dire, les autres titres de l'actualité
  - Chercher les identifiants qui permettront d'isoler les actualités du reste du code HTML en utilisant Xpath.

On a de la chance : dans le code HTML de la page, chaque texte des actualités est situé en cascade dans les tags  et <a> ; il se trouve que ce binôme de tags n'est utilisé que pour les titres des actualités.

En combinant les tags  et <a> avec la simple syntaxe XPath *"//span/a"* il est possible d'extraire tous les titres. Il reste à extraire depuis les titres seulement la partie texte, et non la partie des liens http contenue aussi dans le tag <a>. La [Fig. 2] montre comment en utilisant la console de debug Javascript intégrée dans Chrome, on peut facilement extraire de façon directe les informations recherchées. La syntaxe d'extraction XPath pour cet exercice est donc : *"//span/a/text()"*.

Avec Chrome on peut utiliser la console de debug pour extraire des requêtes XPath en utilisant la syntaxe suivante : *$x("//span/a/text()")*

- Maintenant que la syntaxe Xpath permettant d'extraire les titres a été trouvée, il est possible d'automatiser le processus en créant un script qui exécutera la tâche de façon programmée. Cette opération peut être effectuée avec différents framework et langages. Dans cet article on explorera le framework Scrapy.



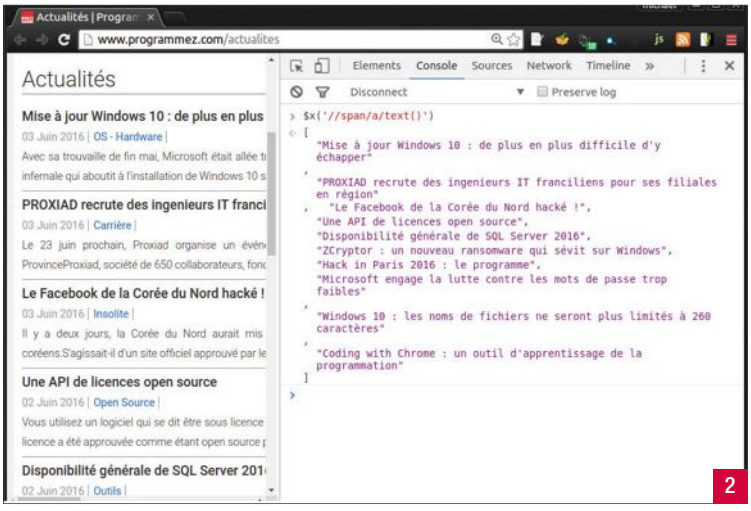

1

## Introduction à Scrapy

En informatique, un framework est un ensemble de librairies, outils et programmes qui aident au développement d'un logiciel. Dans le cas du développement du Web Scraping, des frameworks comme *Pjscrape* (pour le langage Javascript), *Jsoup* (Java) et *Scrapy* (Python) aident le développeur dans ses tâches.

Dans le domaine du Web scraping, un framework doit aider le développeur dans les tâches suivantes :

- Construction des requêtes Web : GET et POST (pour l'envoi de form HTML) ;
- Gestion du protocole http et https : manipulation des headers des requêtes ;
- Extraction des données : Xpath et regular expression ;




2

- Gestion des sessions et cookies ;
- Optionnel: moteur ou proxy Javascript.

Scrapy qui intégre toutes ces fonctionnalités est un framework capable de mettre en place une architecture robuste et une excellente organisation du code. Scrapy est un projet open source accessible à l'adresse http://scrapy.org.

## Premier pas avec Scrapy

L'installation du framework est vraiment simple, un simple 'pip install scrapy' et voilà votre package Python est installé; un utility script est aussi installé pour créer, lancer et administrer les projets créés avec Scrapy. En utilisant l'utility script installé, on ouvre une fenêtre de commande (iTerm, bash, powerShell, etc.) pour lancer la création d'un nouveau projet et la création d'un spider pour la page d'actualité de programmez.com :

```
$scrapy startproject programmez
$cd programmez
$scrapy genspider actualites programmez.com
```

Voici la structure du répertoire créée à la suite des commandes lancées :

| programmez/ | Répertoire du projet |
|---|---|
| scrapy.cfg | Configuration du projet : nom, url, type de déploiement... |
| programmez/ | Nom du package python, ex : import programmez.spiders.actualites |
| __init__.py | (définition package Python) |
| items.py | Représentation des données à extraire |
| pipelines.py | Une fois les données extraites, les pipelines traitent les informations, ex : Filtre des données, Mémorisation sur Database, Ecriture sur fiches... |
| settings.py | Paramètres du projet : cookies, middlewares, timeout, cache... |
| spiders/ | Chaque script de scraping se trouve ici |
| __init__.py | (définition package Python) |
| actualites.py | Spider pour les actualités |

Scrapy permet la création du *squelette* du projet de façon automatisée. Chaque étape du processus de scraping est assignée à une classe désignée pour résoudre un seul problème bien spécifique. La structure des répertoires et des fiches de classes permet une gestion du projet simplifiée. La force de ce framework réside dans la bonne exploitation de la programmation à objet et dans la séparation de chaque classe dans des répertoires bien définis.

L'utility script 'scrapy' est une excellente aide dans les tâches quotidiennes du Web scraper, voici la liste des commandes :

| Commande globale: | |
|---|---|
| startproject <project_name> | Création d'un nouveau projet |
| settings [options] | Montre la configuration du projet |
| runspider <file.py> | Exécute un unique script de spider sans besoin de créer un projet |
| shell [url] | Ouvre une session Scrapy dans un terminal Python |
| fetch <url> | Télécharge l'url spécifique |
| view <url> | Ouvre l'url dans le browser Web principal |

| Commande spécifique à un projet: | |
|---|---|
| crawl <spider> | Lance le processus d'extraction du spider |
| check [-l] <spider> | Test fonctionnel sur le spider |
| list | Montre la liste des spider contenus dans le projet |
| edit <spider> | Ouvre un IDE pour le spider |
| parse <url> [options] | Télécharge l'url et procède au parsing |
| genspider [-t domain] <name> <domain> | Crée un nouveau spider |
| bench | Benchmarks des spiders |

## Scrapy en version live

Dans la section *"Simulation d'un cas réel"*, on a montré comment l'utilisation d'un browser Web peut faciliter la tâche de recherche de l'expression XPath pour l'extraction des données ciblées.

De la même manière Scrapy met à disposition ses différents éléments structurels (classes des connecteur HTTP, classes d'expression régulière, classes d'interrogation XPath…) pour faciliter la construction de la requête Web et l'extraction des données avant de créer chaque composant du pipeline général.

Dans le terminal, lancez la commande suivante :

```
$scrapy shell "www.programmez.com/actualites"
```

Scrapy ouvre une session Python (ou iPython si installé dans le système) dans laquelle les objets (soulignés ici en bleu) sont créés :

```
[scrapy] DEBUG: Crawled (200) <GET http://www.programmez.com/actualites>
[s] Available Scrapy objects:
[s]   crawler    <scrapy.crawler.Crawler object at 0x7f331c17ee90>
[s]   item       {}
[s]   request    <GET http://www.programmez.com/actualites>
[s]   response   <200 http://www.programmez.com/actualites>
[s]   settings   <scrapy.settings.Settings object at 0x7f331c17ee10>
[s]   spider     <DefaultSpider 'default' at 0x7f331bc5eb50>
[s] Useful shortcuts:
[s]   shelp()           Shell help (print this help)
[s]   fetch(req_or_url) Fetch request (or URL) and update local objects
[s]   view(response)    View response in a browser
```

Ayant à disposition ces objets, il ne reste plus qu'à les utiliser ; voici un exemple :

```
In [1]: response.xpath('//title').extract()
Out[1]: [u'<title>Actualit\xe9s | Programmez!</title>']
In [2]: response.xpath('//span/a/text()').extract()
Out[2]:
[u'HTML5 vs Flash : Safari embo\xeete le pas de Chrome',
 u'Paris Container Day',
 u'Dell pourrait c\xe9der sa division logicielle pour 2 milliards de dollars',
 u'BadTunnel : une vuln\xe9rabilit\xe9 qui touchait tous les Windows depuis 20 ans',
 u'Google augmente les primes de son bug bounty Android',
 u"L'entreprise serait-elle Google une voleuse de ballons ?",
 u'Checked C : pour un langage C plus s\xfbr',
 u'DevCon #1 : succ\xe8s sur les IoT et Windows 10 IoT',
 u'La pr\xe9version de Swift 3.0 est arriv\xe9e',
 u'Android N preview 4 : des API stabilis\xe9es']
```

Une remarque sur l'affichage s'impose. Le 'u' devant les éléments des listes extraites signifie que l'encodage est en *unicode*.

Pour visualiser les caractères réels de l'alphabet français, il est possible d'utiliser simplement la fonction *print*, par exemple :

```
In [3]: print response.xpath('//title').extract()[0]
<title>Actualités | Programmez!</title>
```

L'utilisation de Scrapy par sa modalité *en-live* montre deux choses essentielles :

- Les éléments en jeu dans un pipeline de Web scraping representés par des objets ;
- L'ensemble des éléments qu'il faut construire dans l'écosystème Scrapy.

## Construction de la requête

Passons maintenant au codage du fichier pré-rempli *programmez/spiders/actualites.py*.

La méthode "parse" n'extrait rien par défaut. Dans *stats_urls* sont définies les pages qui doivent être interrogées.

Les modifications à apporter sont les suivantes :

- Ajouter l'url de la page à extraire dans la variable de classe *stats_urls ;*
- Extraire les données en utilisant l'objet **response ;**
- Travaillez avec les données extraites ; dans le cas de cet exercice, sauvegarder les données dans un fichier.

Une fois ces modifications appliquées, le code du spider *actualites.py* peut apparaître comme dans la [Fig. 3].

Pour lancer ce spider, il suffit d'ouvrir un terminal, de se déplacer dans la racine du répertoire du projet et de lancer la commande suivante :

```
$scrapy crawl actualites
```

A noter qu'au lieu de sauvegarder ces données sur un simple fichier, les actions auront pu être la mise à jour des données sur database relationnelle, NoSQL ou bien le lancement d'une autre extraction qui utilisera ces données comme input. La seule limite à l'utilisation des données extraites est sa propre imagination. Dans la requête d'exemple, le contenu extrait est de nature statique, c'est à dire, qu'il n'y a ni Javascript ni Ajax qui charge les titres des actualités de façon dynamique, comme par exemple l'une après l'autre. Si c'était le cas, la requête construite avec scrapy n'aurait pas fonctionné, car scrapy aurait téléchargé la page HTML avec le contenu Javascript sans pouvoir l'exécuter.

3

```python
# -*- coding: utf-8 -*-
import scrapy


class ActualitesSpider(scrapy.Spider):
    name = "actualites"
    allowed_domains = ["programmez.com"]
    start_urls = (
        'http://www.programmez.com/actualites',
    )

    def parse(self, response):
        self.logger.info('Parsing de la page %s', response.url)
        news = response.xpath('//span/a/text()').extract()
        with open('news.txt','w') as f:
            [f.write("%s\n" % x.encode("utf-8")) for x in news]
```

## Dans le moteur des Browser

Dans le contexte général du Web scraping, il faut faire face au contenu dynamique (Javascript) des pages et des requêtes asynchrones (Ajax). Une des solutions est d'utiliser le "moteur" d'un browser Web. En utilisant seulement un sous-ensemble de composants qui forme un browser Web (interprète Javascript, parser XML, module communication réseau) il est possible d'émuler le comportement d'un browser Web sans son interface graphique.

Les interactions qui sont normalement réalisées par la souris ou le clavier, sont pilotées grâce à l'utilisation du DOM, c'est à dire, grâce à la structure du document HTML dans laquelle les éléments sont trouvés et manipulés. Il est aussi possible d'utiliser, à l'aide des connecteurs spécifiques (ce qui demande une installation différente pour chaque connecteur) un browser web comme Chrome et Firefox.

L'avantage d'utiliser seulement le 'moteur' d'un browser Web, plutôt qu'un browser 'classique' dans sa totalité, est le facteur temps : dans un browser Web 'classique', la chaîne de traitement d'une requête passe aussi par le moteur du rendering (visualisation) des objets DOM.

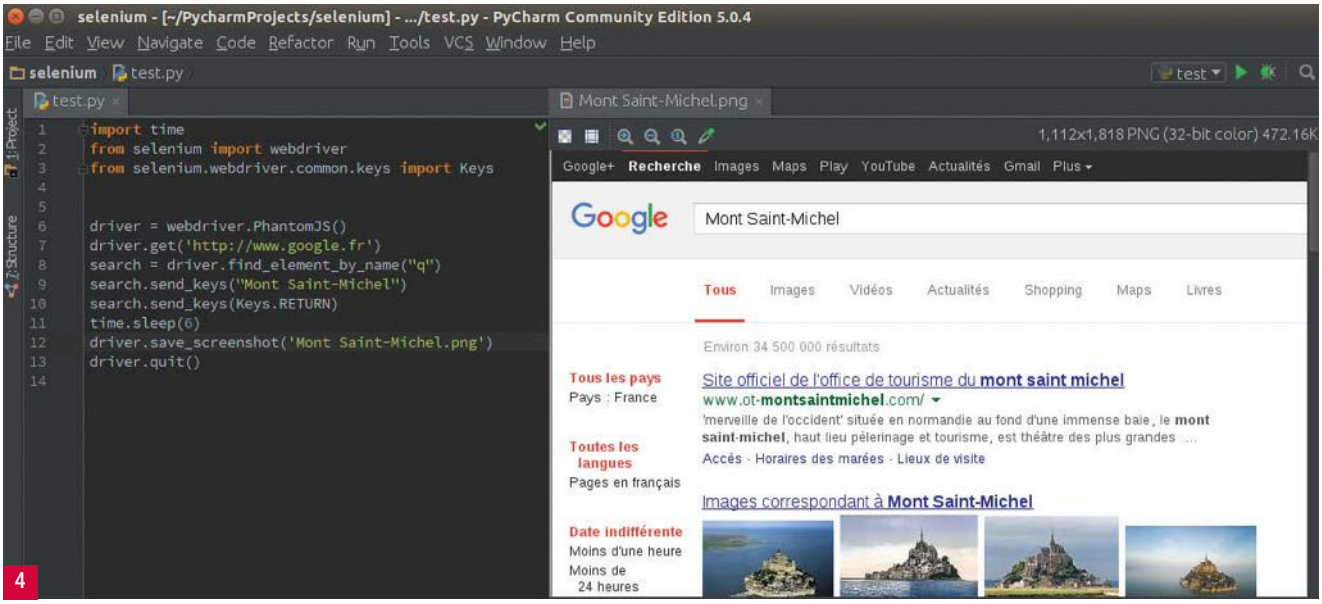
Pour mieux comprendre comment utiliser un moteur Web, on utilisera ici le wrapper PhantomJS. PhantomJS est une interface au moteur WebKit développé par Apple et utilisé dans Safari, Chromium et Google Chrome. L'interaction avec PhantomJS est simplifiée grâce aux connecteurs mis à disposition dans le framework Selenium, qui joue un rôle important dans le test des applications Web. Voici comment installer le framework Selenium et le connecteur/driver PhantomJS sur un système Linux :

```
$sudo pip install selenium
$sudo apt-get install phantomjs
```

La [Fig. 4] montre comment faire une recherche sur Google tout en simulant un browser Web et pouvant ensuite enregistrer la screenshot de la page. Selenium met à disposition un grand nombre de drivers Web avec lesquels simuler une navigation en se faisant passer par un des browsers connus. Pour connaître lesquels sont installés sur votre système, ouvrez une session d'iPython et utilisez la touche <tab> comme montré en [Fig. 5].

## Scrapy sans limites

Maintenant qu'on peut simuler un vrai browser Web, interpréter Javascript et exécuter des requêtes ajax, rien ne peut plus nous arrêter.



4

La [Fig. 6] montre comment transformer le script 'actualités' en une nouvelle version qui utilise PhantomJS. Pour créer la fiche dans la bonne arborescence, on se place dans la racine du projet 'programmez' et on lance la commande :

```
$scrapy genspider actualites-driver programmez.com
```

Une fois le script modifié, il est possible de l'exécuter grâce à la commande :

```
$scrapy crawl actualites-driver
```

## Gestion du projet

Scrapy met à disposition des outils pour déployer et exécuter ses spiders. Ces outils font référence à deux packages Python *scrapyd* et *scrapyd-client* que je recommande d'installer depuis leur repository officiel Github. Une fois ces packages installés, il est nécessaire de lancer le démon *'scrapyd'*. Dans le fichier de notre projet *programmez/scrapy.cfg* decommenté la ligne qui définit l'#url. Le fichier doit donc ressembler au suivant :

```
[settings]
default = programmez.settings

[deploy]
url = http://localhost:6800/
project = programmez
```

Maintenant ouvrez le terminal et placez-vous à la racine du projet 'programmez' pour lancer la commande :

```
$scrapyd-deploy default -p programmez
Packing version 1469455584
Deploying to project "programmez" in http://localhost:6800/addversion.json
Server response (200):
{"status": "ok", "project": "programmez", "version": "1469455584", "spiders": 2}
```

Cette commande a permis de déployer sur le serveur (dans cet exemple *localhost*) le package de notre application.
Pour exécuter le spider déployé, lancez la commande :

```
$curl http://0.0.0.0:6800/schedule.json -d project=programmez -d spider=actualites
{"status": "ok", "jobid": "fb3c34ce527611e6a2ee480fcf445c6d"}
```

La [Fig. 7] montre l'interface graphique, minimaliste mais bien utile, qui permet de garder une trace de chaque exécution d'un spider et aussi du log généré. Dans le cas d'une automatisation complète, une astuce consiste à utiliser un cron pour programmer le jour et l'heure d'exécution des spiders. Voici comme lancer le spider d'actualités tous les jours à midi en utilisant la *crontab* de linux:

```
$(crontab -l ; echo "00 12 * * * /path/to/spiders.sh") | crontab -
```

Le ficher *spiders.sh* doit ressembler à ceci :

```
#!/bin/bash
curl http://0.0.0.0:6800/schedule.json -d project=programmez -d spider=actualites
```

En conclusion, l'avantage de ce système de scheduling des spiders permet de :

- Distribuer la charge de travail des spiders/jobs sur un système à queue ;
- Diviser le travail sur plusieurs serveurs ;
- Avoir une trace des jobs et de relative logs accessible aussi via une interface Web et en temps réel
- Créer des packages de spider versionnés.

# CONCLUSION

Parmi les tâches que l'on répète tous les jours sur le net, il y en a certaines que vous pouvez maintenant automatiser grâce aux techniques et technologies explorées. Tout bon programmeur crée des scripts de "système" (bash, python, NodeJs, etc) qui l'aident dans son travail quotidien. De même le Web Scraping ouvre une nouvelle frontière de scripts qui ont pour but d'optimiser notre temps sur le net et on ne doit pas pouvoir échapper à cette possibilité !
Que ce soit pour gagner de l'argent ou bien pour gagner du temps, ce qui revient bien souvent à la même chose, le Web Scraping est manifestement une pratique à laquelle on doit s'intéresser. On termine avec une suggestion : imaginez que vos *spiders* soient liés à des algorithmes d'intelligence artificielle... •

5

```
In [1]: from selenium import webdriver

In [2]: webdriver.
webdriver.ActionChains          webdriver.Safari
webdriver.Android               webdriver.TouchActions
webdriver.BlackBerry            webdriver.android
webdriver.Chrome                webdriver.blackberry
webdriver.ChromeOptions         webdriver.chrome
webdriver.DesiredCapabilities   webdriver.common
webdriver.Edge                  webdriver.edge
webdriver.Firefox               webdriver.firefox
webdriver.FirefoxProfile        webdriver.ie
webdriver.Ie                    webdriver.opera
webdriver.Opera                 webdriver.phantomjs
webdriver.PhantomJS             webdriver.remote
webdriver.Proxy                 webdriver.safari
webdriver.Remote                webdriver.support
```

7

0.0.0.0:6800/jobs

**Jobs**

Go back

| Project | Spider | Job | PID | Runtime | Log |
|---|---|---|---|---|---|
| **Pending** | | | | | |
| **Running** | | | | | |
| **Finished** | | | | | |
| programmez | actualites-driver | 2895e4de527111e6a2ee480fcf445c6d | | 0:00:17.378416 | Log |
| programmez | actualites | 55e670b6527111e6a2ee480fcf445c6d | | 0:00:00.719363 | Log |

6

actualites_driver.py

```
# -*- coding: utf-8 -*-
import scrapy
from selenium import webdriver
from scrapy.http import HtmlResponse

class ActualitesDriverSpider(scrapy.Spider):
    name = "actualites-driver"
    allowed_domains = ["programmez.com"]
    start_urls = ['http://www.programmez.com/actualites']

    def __init__(self, ip=None, *args, **kwargs):
        self.driver = webdriver.PhantomJS()

    def parse(self, response):
        self.driver.get(self.start_urls[0])
        html = self.driver.page_source
        self.driver.quit()
        response = HtmlResponse(url=self.start_urls[0], body=html)
        news = response.xpath('//span/a/text()').extract()
        with open('news.txt','w') as f:
            [f.write("%s\n" % x.encode("utf-8")) for x in news]
```

# Forensic en Python

Franck Ebel
*Expert R&D et Formations Serval-concept / serval-formation*
*Responsable Licence CDAISI, UVHC*
*Commandant de Gendarmerie réserviste, cellule Cyberdéfense*

*Nous avons vu dans l'article précédent la manière de parcourir un répertoire ou un disque dur entier avec os.walk().*
*Nous avons donc la possibilité maintenant de rechercher les fichiers que l'on désire, que ce soient des pdf, des fichiers odt, des images ou autres.*
*Mais quelles sont les informations que nous allons pouvoir tirer de ces différents types de fichiers ?*
*Cet article a pour but de vous guider dans l'extraction des métadonnées de différents formats de fichiers. Les métadonnées dans les fichiers vont nous permettre de connaître la date de création du fichier, qui l'a créé, et avec quel outil par exemple.*

## 1. Les fichiers « son » de type MP3

La bibliothèque eyeD3 va nous permettre d'examiner les métadonnées des fichiers MP3. Cette bibliothèque possède différentes méthodes qui vont nous être très utiles telles que getArtist(), getAlbum(), getTitle() par exemple. Grâce à getArtist(), getAlbum(), getTitle() par exemple, nous pouvons récupérer les données propres au fichier MP3 obtenues par eyeD3.tag().

```
import eyed3
chemin=raw_input("donnez le chemin du fichier mp3\n")
fichier=eyed3.load(chemin)
print fichier.tag.artist
print fichier.tag.album
print fichier.tag.title
```

Le résultat donnera :

```
bash-3.2# python tag_mp3.py
donnez le chemin du fichier mp3
/Users/franckebel//Documents/personnel/chansons/adam-levine-lost-stars-ost.mp3
Adam Levine
Begin Again
Lost Stars
bash-3.2#
```

Nous voyons qu'il est très simple de récupérer des informations dans les métadonnées de ce type de fichier. De la même manière nous pouvons écrire les métadonnées comme nous le voulons.
Reprenons ce fichier mp3 pour modifier son contenu.

```
import eyed3
chemin=raw_input("donnez le chemin du fichier mp3\n")
fichier=eyed3.load(chemin)
fichier.tag.artist = u"Auteur en Python"
fichier.tag.album = u"Magazine Programmez"
fichier.tag.title = u"Le forensic en Python"
fichier.tag.save()
```

Le résultat donnera :

```
print fichier.tag.artist
print fichier.tag.album
print fichier.tag.title
```

```
bash-3.2# python tag_mp3_write.py
donnez le chemin du fichier mp3
/Users/franckebel//Documents/personnel/chansons/adam-levine-lost-stars-ost.mp3
Auteur en Python
Magazine Programmez
Le forensic en Python
bash-3.2#
```

Nous pouvons avancer dans notre script en lisant le contenu des métadonnées pour ensuite écrire le nom du mp3 en fonction de ces métadonnées. Nous utiliserons donc le module argparse pour aller chercher un nom de fichier mp3 dans mon exemple chanson.mp3, pour renommer ce mp3 avec le vrai nom de l'artiste, de l'album et le titre de la chanson. Voici donc le script :

```
import os
import eyed3
import argparse

parser = argparse.ArgumentParser()
parser.add_argument("filename")

args = parser.parse_args()

audiofile = eyed3.load(args.filename)
new_filename = "tagged/{0}-{1}-{2}.mp3".format(audiofile.tag.artist, audiofile.tag.
album, audiofile.tag.title)
os.rename(args.filename, new_filename)
```

Lançons ce script en lui passant comme argument le nom du fichier et regardons le résultat (ce résultat est placé dans le répertoire tagger que je crée juste avant).

```
bash-3.2# python parse_mp3.py
chanson.mp3      tag_mp3.py
parse_mp3.py     tag_mp3_write.py
bash-3.2# mkdir taggedbash-3.2# python parse_mp3.py "chanson.mp3"
bash-3.2# ls
parse_mp3.py      tag_mp3.py       tag_mp3_write.py  tagged
bash-3.2# cd tagged
bash-3.2# ls
```

```
Linkin Park-Road To Revolution: Live At Milton Keynes-One Step Closer.mp3
bash-3.2#
```

Nous commençons à voir comment travailler sur les métadonnées. Passons maintenant à d'autres types de fichiers.

## 2. Métadonnées dans les images

La bibliothèque PIL est l'outil idéal pour manipuler les images, il faudrait un article complet uniquement sur PIL pour bien comprendre son fonctionnement, nous allons ici voir le principe et les méthodes principales à la recherche d'informations dans ces fichiers images.
Tentons d'ouvrir une image, prenons ma photo (moi.png) pour travailler sur différents éléments.

```
from PIL import Image
img = Image.open('moi.png').convert('L')
img.show()
img.save('moi_1','png')
```

- Nous importons d'abord PIL (vous pouvez l'installer via pip par exemple php install py-pil) ;
- Nous ouvrons ensuite l'image ( Image.open() ) ;
- Nous allons ensuite convertir l'image ( convert ) avec l'option L, qui signifie niveau de gris ;
- Nous sauvons ensuite cette image ;
- Nous obtenons une nouvelle image (moi_1 ) que nous pouvons ouvrir avec n'importe quel éditeur.

Nous voyons que l'image est bien transformée en niveaux de gris. Nous pouvons de la même manière modifier la taille de l'image facilement.

```
from PIL import Image
img = Image.open('moi.png')
new_img = img.resize((256,256))
new_img.save('moi-256x256','png')
```

De la même manière nous allons pouvoir facilement créer des miniatures (thumbnails) :

```
from PIL import Image
import glob, os

size = 128, 128

for infile in glob.glob("*.jpg"):
    file, ext = os.path.splitext(infile)
    im = Image.open(infile)
    im.thumbnail(size)
    im.save(file + ".thumbnail", "JPEG")
```

Nous parcourons ici le chemin courant en y recherchant tous les fichiers en .jpg et nous créons pour chacun de ces fichiers une image miniature.
Il existe beaucoup de fonctionnalités autres et très utiles de la bibliothèque PIL, mais passons à ce qui nous intéresse : l'extraction de données. Une méthode _getexif , expérimentale est incluse dans PIL.
Voici un script qui vous retournera toutes les métadonnées, si elles existent (sinon il retournera une erreur ) :

```
From_PIL import Image
From_PIL.ExifTags import TAGS, GPSTAGS
image = Image.open("einstein.jpeg")
print(image)
info = image._getexif()
for tag, value in info.items():
    key = TAGS.get(tag, tag)
    print(key + " " + str(value))
```

Nous utiliserons la méthode TAGS et GPSTAGS de PIL.ExifTags pour récupérer les données. Il nous suffira ensuite d'ouvrir l'image et de parcourir info.items() , info contenant le retour de images._getexif().
Vous pourrez facilement adapter vos besoins avec des connaissances de bases en Python. Une autre bibliothèque peut être utile pour l'extraction de données : piexif. Essayons d'extraire les données de l'image einstein.jpg que j'ai sur mon PC.

```
import piexif
print( '\n<< Test de Piexif >>' )

data = piexif.load("einstein.jpeg")
for key in [' '0th', '1st', 'GPS', 'Interop']:
        subdata = data[key]
        print( '\n%s:' % key )
        for k, v in subdata.items():
                print( '%s = %s' % (k, v) )
```

Le résultat donnera :

```
bash-3.2# python piexif_exemple1.py

<< Test of Piexif >>

0th:
283 = (720000, 10000)
296 = 2
34665 = 11444
306 = b'2016:11:7 15:39:05'
270 = b''
271 = b'OLYMPUS IMAGING CORP.'
272 = b'E-P1'
305 = b'Adobe Photoshop CS5 Windows'
274 = 1
33432 = b'Franck ebel'
282 = (720000, 10000)
315 = b'Franck Ebel'
```

Nous retrouvons les données inclues dans l'image. Si des coordonnées GPS étaient présentes , nous aurions pu les récupérer aussi facilement.
En parcourant toutes les images d'un smartphone par exemple, nous aurions pu récupérer les coordonnées GPS (si activées) et retracer avec la date de prise de la photo, le chemin parcouru par l'utilisateur.

## 3. Métadonnées des PDF

Il est aussi utile de pouvoir travailler sur les pdf. La bibliothèque pypdf va nous permettre par exemple de rechercher les pdf de plus de 3 pages et ayant comme auteur franck ebel. Voici un script qui va correspondre à nos attentes :

```
import warnings,sys,os,string
From_pyPdf import PdfFileWriter, PdfFileReader
for root, dir, files in os.walk(str(sys.argv[1])):
        for fp in files:
                if ".pdf" in fp:
```

```
            fn = root+"/"+fp
            try:
                pdfFile = PdfFileReader(file(fn,"rb"))
                title = pdfFile.getDocumentInfo().title.upper()
                author = pdfFile.getDocumentInfo().author.upper()
                pages = pdfFile.getNumPages()
                if (pages > 3) and ("ebel" in author):
                    resultStr = "Matched:"+str(fp) +"-"+str(pages)
                    print resultStr
```

Nous utilisons os.walk() vu dans l'article précédent associé à la bibliothèque pyPdf. Nous allons rechercher les métadonnées grâce aux méthodes getDocumentsInfo().tiltle, getDocumentInfo.author et getNumPages(). Nous effectuons ensuite un test (if) afin de n'afficher que les pdf de plus de 3 pages dont l'auteur est ebel.
Avec un peu d'imagination et la lecture de la documentation, nous pourrons aisément aller chercher d'autres informations utiles en forensic avec des mots clés spécifiques pour la recherche de preuves informatiques.

## 4. Contenu des fichiers ZIP

Nous voulons examiner directement un ou plusieurs fichiers contenus dans une archive au format ZIP, sans la décompacter sur le disque.
Une bibliothèque existe, zipfile, qui va nous permettre de travailler directement sur les données contenues dans des fichiers ZIP.

```
#!/usr/bin/env python
import zipfile
z=zipfile.ZipFile("fichier.zip","r")
for nom in z.namelist() :
    print 'le fichier', nom,
    nb_octets=z.read(nom)
    print 'contient ', len(nb_octets),'octets.'
```

Nous pouvons aussi consulter le contenu des fichiers.

```
#!/usr/bin/env python   import zipfile
z = zipfile.ZipFile('test.zip', 'r')
names = z.namelist()
for name in names:
   print 'Attente de %s' % name
   print z.read(name)
for name in names:
    print 'en Attente de %s' % name
   f = z.open(name)
   contents = f.read()
```

## 5. Fichiers Openoffice ou word

### 5.1 Parcourir une arborescence

Nous devons examiner un répertoire ou toute une arborescence de répertoires situés dans un répertoire donné pour itérer sur les fichiers dont les noms correspondent à certains motifs.
Le générateur os.walk suffit à cette tâche.

```
#!/usr/bin/env python   import os, fnmatch
def tous_les_fichiers( racine, motifs='*', un_seul_niveau=False,
repertoires = False):
    motifs=motifs.split(';')
    for chemin, sous_reps, fichiers in os.walk(racine):
        if repertoires:
            fichiers.extend(sous_reps)
        fichiers.sort()
        for nom in fichiers:
            for motif in motifs:
                if fnmatch.fnmatch(nom,motif):
                    yeld os.path.join(chemin,nom)
                    break
        if un_seul_niveau:
            break
for chemin in tous_les_fichiers('/tmp','*.py ;*.htm ;*.html') :
    print chemin
```

fnmatch permet de rechercher les noms de fichiers correspondant à des motifs. Pour préciser plusieurs motifs, nous devons les séparer par des points-virgules, un motif ne devra donc pas posséder de point-virgule.
Dans l'exemple ci-dessus, nous utilisons la définition tous_les_fichiers pour rechercher le chemin de tous les fichiers Python et HTML.

### 5.2 Rechercher dans un document OpenOffice

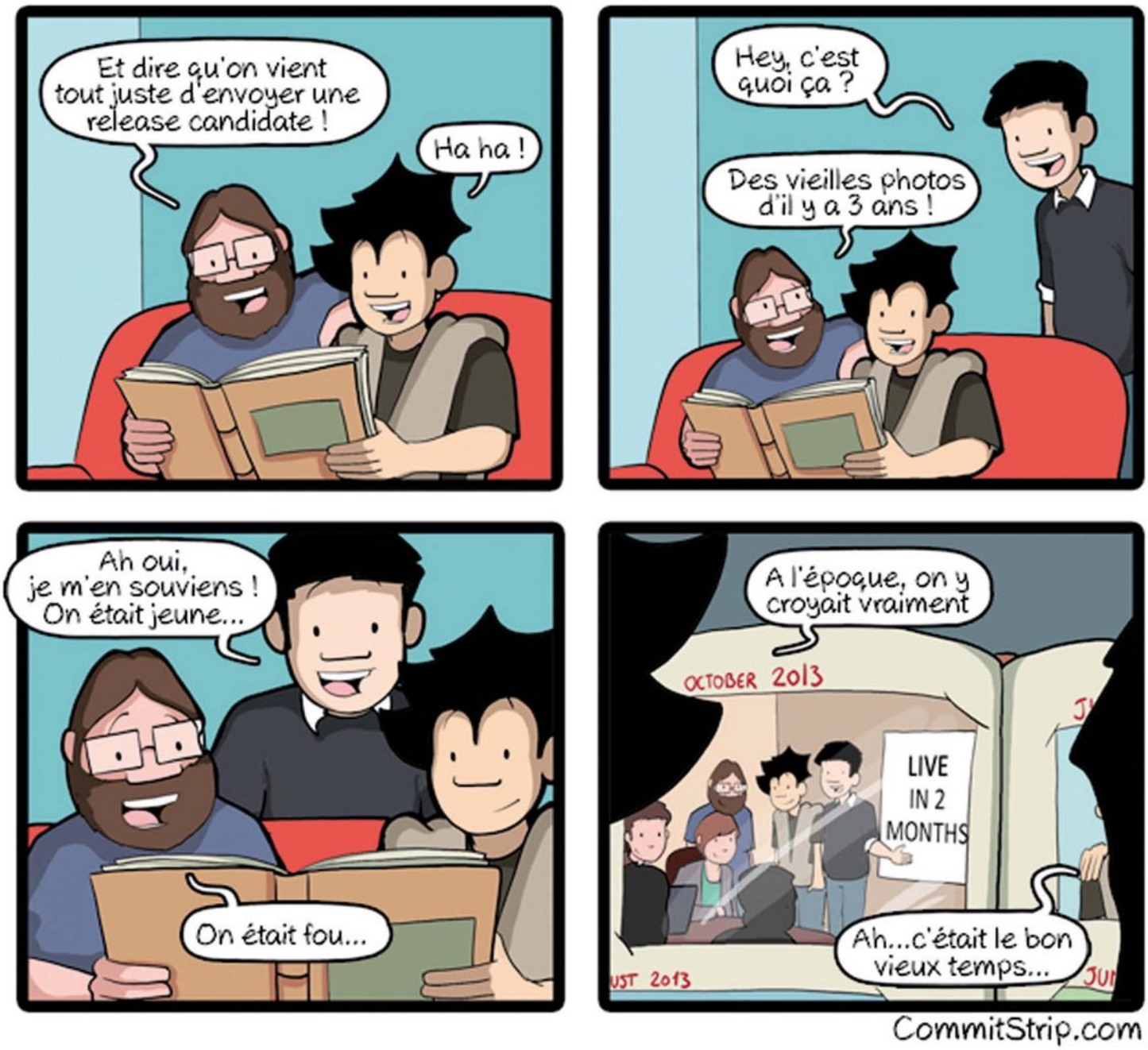
Nous souhaitons extraire des données d'un document OpenOffice ou LibreOffice. Ce type de document est un document en ZIP rassemblant des fichiers XML respectant un standard bien défini. Nous avons déjà étudié le XML et le ZIP. Nous allons donc récupérer le fichier content.xml, que nous allons alléger des balises XML à l'aide d'une expression régulière, puis nous allons découper le résultat selon les espaces et le reconstruire avec un unique espace pour économiser la place occupée.

```
#!/urs/bin/env python   import zipfile, re
re_suppr_xml=re.compile("<[^>]*  ?",re.DOTALL | re.MULTILINE)
def convertir_00(nomFichier,texte_seul=True) :
    fz=zipfile.ZipFile(nomfichier,"r")
    donnees=fz.read("content.xml")
    fz.close()
    if texte_seul :
        donnees= " ".join(re_suppr_xml.sub(" "  ,donnees).split())
    return donnees
if __name__=="__main__" :
    import sys
    if len(sys.argv)>1  :
        for nomdoc in sys.argv[1:]  :
            print 'Texte de ',nomdoc,'  :'
            print convertir_00(nomdoc)
            print 'XML de ',nomdoc, '  :'
            print convertir_00(nomdoc,texte_seul=False)
    else  :
        print 'Donnez des noms de documents OpenOffice pour  les lire au
format texte et XML'
```

## CONCLUSION

Nous avons dans cet article, travaillé sur divers formats de fichiers afin d'y récupérer un contenu ou les métadonnées.
Dans un prochain article nous travaillerons sur un script Python qui permet de rechercher certaines informations inscrites en mémoire.
Ce script, volatility est un outil indispensable pour le forensic.
Vous pouvez retrouver mes vidéos sur Python, Scapy, le forensic en Python sur ma chaîne youtube : http://www.youtube.com/c/franckebel •

# Codeur bohême




**Une publication Nefer-IT,** 7 avenue Roger Chambonnet, 91220 Brétigny sur Orge - redaction@programmez.com
Tél. : 01 60 85 39 96 - Directeur de la publication & Rédacteur en chef : François Tonic
**Secrétaire de rédaction :** Olivier Pavie
**Ont contribué à ce numéro :** S. Saurel, O. Pavie
**Nos contributeurs techniques :** P. Charrière, T. Lepérou, M. Bacci, F. Ebel, A. Galtier, J. Paquet, J. Rabusseau, N. Decoster, A. Vache, Y. Ovazza, B. Bourgois, L. Piot, P. Lamarche, T. Ranise, M. Caroul, S. Berberat, P-A Sunier, T. Leriche-Dessirier, D. Lecan,
**Couverture :** © artisticco - **Maquette :** Pierre Sandré.
**Publicité :** PC Presse, Tél.: 01 74 70 16 30, Fax : 01 40 90 70 81 - pub@programmez.com.
**Imprimeur :** S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles, Belgique.
**Marketing et promotion des ventes :** Agence BOCONSEIL - Analyse Media Etude - Directeur : Otto BORSCHA oborscha@boconseilame.fr
**Responsable titre :** Terry MATTARD Téléphone : 09 67 32 09 34

**Contacts :** Rédacteur en chef : ftonic@programmez.com - Rédaction : redaction@programmez.com - Webmaster : webmaster@programmez.com -
**Publicité :** pub@programmez.com - Evenements / agenda : redaction@programmez.com

**Dépôt légal :** à parution - Commission paritaire : 1220K78366 - ISSN : 1627-0908 - © NEFER-IT / Programmez, décembre 2016

Ce numéro comprend sur l'édition abonné un catalogue et un bon de commande de la société PC SOFT

**Abonnement :** Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex. - Tél. : 01 55 56 70 55 - *abonnements.programmez @groupe-gli.com* - Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30. **Tarifs** abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter.
**PDF :** 35 € (Monde Entier) souscription sur www.programmez.com


**Sur abonnement ou en kiosque**

# Le magazine des pros de l'IT

**Mais aussi sur le web**



**Ou encore sur votre tablette**

Openska

# Formations

## Web & Open Source


www.openska.com