

PROgrammez

Mensuel n°165 - juillet-Août 2013

www.programmez.com

Soyez le geek de l'été !



Découvrez
PHP 5.5

*Picaxe, Leap Motion,
Gadgeteer, Raspberry Pi :*
la programmation en s'amusant

Enquête écoles

Diplômes :
quel prix
pour quel salaire ?



Chrome, Firefox, IE, Safari

Mon code est-il optimisé pour les navigateurs web ?

Utiliser les Pointer events,
des CSS performants sous IE10.

- Maîtriser toute la puissance des Websocket
- Facebook : utiliser le SDK .Net
- Et aussi : Powershell, Android, serveur web Apache, Symfony, Dropbox, iOS

Printed in EU - Imprimé en UE - BELGIQUE 6,45 €
SUISSE 12 FS - LUXEMBOURG 6,45 € DOM Surf 6,90 €
Canada 8,95 \$ CAN - TOM 940 XPF - MAROC 50 DH

M 04319 - 165 - F - 5,95 € - RD

KJERSHILD



WINDEV®

JUSQU'AU

19 JUILLET

OPERATION DE FOLIE!

2 POUR 1 EURO DE PLUS

2x Samsung GALAXY S4

Le tout dernier smartphone de SAMSUNG • Ecran 5" 1920 x 1080 • Android™ 4.2 • APN 13M • 130 grammes • Compatible 4G • Quadri Bande • GPS • Processeur quadri coeur • Webcam HD...



Aucun abonnement à souscrire pour bénéficier de cette offre.

Vos applications sont compatibles : Windows, Linux, Mac, .Net, Java, PHP, J&EE, XML, Internet, Cloud, Windows CE & Phone, Android, iPhone, iPad

ou WINDEV Mobile 18 ou WEBDEV 18

COMMANDEZ WINDEV® ET RECEVEZ NON PAS 1 MAIS 2 MATÉRIELS POUR «1 EURO DE PLUS»

Choisissez 2 matériels :



Nouveau smartphone Samsung Galaxy S4
Ecran 5 pouces 1920 x 1080 Android 4.2 APN 16 Mégas
Ou choisissez la version «S4 mini», bientôt disponible



Nouvelle tablette Samsung Galaxy Tab 3
Ecran géant 10,1 pouces Android 4.2 - 16 Go Wifi. Lecteur vidéo full HD
Ou choisissez la version 8 pouces



Smartphone Pico projecteur Samsung Beam
Ecran 4 pouces Android - Pico projecteur DLP capable de projeter des images et des vidéos
Configuration sur www.pcsoft.fr



PC Portable
2 modèles au choix
• **Samsung 3530EC**
Ecran 15,6 pouces LED 2,3 kg - Webcam - Disque 500 Go Windows 8 Pro 64 bits
ou choisissez
• **Samsung 532U3C**
Ultra-portable - Ecran 13,3 pouces LED - 1,5 kg - ...

Offre réservée aux sociétés, administrations, mairies, GIE et professions libérales, en France métropolitaine.

Aucun abonnement n'est à souscrire pour bénéficier de cette offre.

Le développement pour Android et iOS s'effectue avec WINDEV Mobile ou WEBDEV. Le développement pour Windows s'effectue avec WINDEV ou WEBDEV. Voir tous les détails et des vidéos sur : www.pcsoft.fr

Le Logiciel et le matériel peuvent être acquis séparément; logiciel au prix catalogue de 1.650 Euros HT (1.973,40 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.

Descriptif technique complet des matériels sur www.pcsoft.fr

Tél: 04.67.032.032



Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr

iOS 7 : personne n'est indifférent !

Durant la grande conférence développeur Apple, la WWDC, la Pomme a dévoilé de nombreuses nouveautés logicielles et matérielles. Tout le monde attendait iOS 7, première version sur laquelle le chef designer pommé, Jonathan Ive, a travaillé intensivement. Résultat : tout a été revu !

IOS 7 : REFONTE TOTALE, 1500 NOUVELLES API

Cette fois, on ne pourra pas dire qu'Apple ne prend pas de risque sur iOS. Le système repose sur les mêmes fondations depuis la v1. L'interface avait très peu changé depuis. La v7 garde la même base technique : Unix / OS X. Par contre, toute l'interface a été transformée en profondeur. Le design devient très plat : fini les effets de reflets, les couleurs partout. Apple rompt radicalement avec l'actuel iOS. L'interface exploite mieux la totalité de la surface de l'écran. La police s'affine considérablement. Ive a voulu faire sobre, trop sobre pour certains. iOS 7 introduit le vrai multitâche pour toutes les applications. Aujourd'hui, il s'agit d'une vraie-fausse fonction, limitée à certaines applications. En tout, Apple a modifié et rajouté plus de 1500 API ! Une des nouveautés est le framework Inter-App audio permettant d'utiliser des fichiers audio entre applications.

Depuis sa présentation, la communauté est assez divisée : copie pure et simple sur Android et Windows Phone 8, remise à plat salulaire. Nous pourrions juger à sa sortie officielle, avec sans date, le nouvel iPhone !

iOS 7 fonctionnera au minimum sur un iPhone 4, iPad 2. Sortie : cet automne.

Pour aider les développeurs, Apple a mis en ligne (compte développeur obligatoire) plusieurs guides : iOS UI Transition, iOS Human Interface et UIKit User Interface Catalog. A lire d'urgence !

Site : <https://developer.apple.com/library/prerelease/ios/design/index.html>

Pour lire la note de version : <http://goo.gl/LBzqt>

OS X 10.9 ALIAS MAVERICKS

Sur la partie OS X, Apple a montré la prochaine version du système, nom de code Mavericks. Cette version est bien moins radicale que iOS 7 : interface qui évolue plus en douceur. Tous les détails n'ont pas été donnés mais cette version change beaucoup

Cancel
Add Event
Done

Room 17

All-day ☒

Starts Mon, Jun 17, 2013

March	14	2010
April	15	2011
May	16	2012
June	17	2013
July	18	2014
August	19	2015
September	20	2016

Ends Mon, Jun 17, 2013

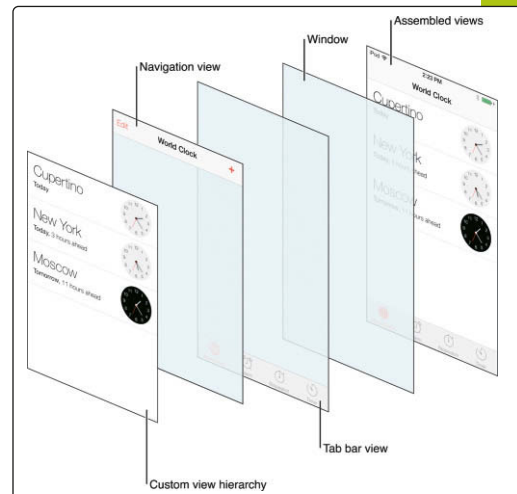
de choses dans les couches basses. Apple promet une optimisation assez poussée de l'utilisation de la mémoire vive et de sa gestion, jusqu'à 72 % d'économie de mémoire utilisée par les applications. Surtout, les applications en arrière fond consommeront beaucoup moins. Les développeurs pourront intégrer, utiliser des fonctions de l'application Plans, qui arrive sur le Mac. Apple a enfin mis de l'ordre dans la gestion du multi-écran : mode plein écran portant, possibilité de dédier une application à un écran (en plein-écran).

XCODE 5.0

Avec les nouveaux systèmes, Apple a dévoilé le prochain XCode, l'environnement intégré de développement Apple : XCode 5.0. L'interface a été elle aussi retouchée, mais très légèrement. Cette v2 se concentre sur les tests pour pouvoir tester son code sur différentes configurations, très rapidement. La gestion de code a été renforcée notamment entre Git et OS X Server.

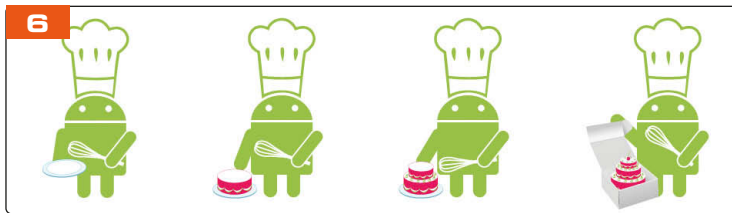
UN MAC PRO... CYLINDRIQUE

La plus belle surprise matérielle a été la présentation du futur Mac Pro, qui sortira d'ici décembre 2013. Bien plus petit que le modèle actuel, ce Mac Pro présente un design totalement nouveau pour Apple. Il ne ressemble à rien d'autre. En réalité, l'extensibilité se fera par l'autre boîtier d'extensions



Thunderbolt 2. Il embarquera les dernières générations de processeurs Intel Xeon (jusqu'à 12 cœurs), 2 processeurs graphiques (AMD FirePro). Apple promet d'ailleurs jusqu'à 7 téraflops de calculs ! Il permettra de supporter les écrans et les vidéos 4K, ainsi que les moniteurs. Une première pour Apple. Le stockage est assuré par du PCIe, une interface bien plus rapide que le SATA standard. Pour démontrer la performance de cette machine, Apple sortira une nouvelle version de son outil de montage : Final Cut. Apple a mis en avant le design de la machine (qui est d'une pureté incroyable) et le système thermique. Car l'ensemble chauffera beaucoup. On pouvait déjà le constater avec l'actuel modèle. Les connecteurs seront assez nombreux : USB 3, HDMI 1.4, Thunderbolt 2, Ethernet...

Deux questions : la date de sortie et surtout le tarif (qui promet d'être très cher)...



WEBMASTER

IIS : après le développement, le déploiement !48
Déployer un serveur Apache, en mode serveur et Cloud50

BUZZWORD

iOS 7 : personne n'est indifférent !3

ACTUS

Actu en bref6

RESSOURCES

Livres9

Événements, agenda10

OUTILS

PHP 5.5 : les nouveautés12

DEVELOPPEUR DU MOIS

Stéphane Sibué :
Le développement, c'est magique !16

Alex Hui :
12 ans chez Microsoft, développeur et créateur
du premier clavier Bluetooth18

CARRIERE

Des écoles d'informatique pour tous les goûts52

PRATIQUE

Le SDK Facebook pour.NET56

JE DEBUTE AVEC...

Flume NG : exploiter les logs !61

Initialiser un nouveau projet Symfony263



52

DOSSIER

Développeurs
n'ayez pas peur
des navigateurs !



Comprendre l'architecture des navigateurs35

Moteurs : chacun chez soi36

Pointer Events : l'avenir du tactile sur le web37

Les navigateurs et HTML 541

Les outils de développement intégrés42

Les nouvelles possibilités CSS de mise en page dans IE1044

A lire dans le prochain numéro

N°166-septembre 2013, à paraître le 2 septembre 2013

iPhone /iPad

Débuter avec Objective-C
et les outils XCode

Utilisez-vous du code illégal ?

Webmaster

l'incontournable JSON

100% Geek en vacances !

Faire tourner .NET avec l'accès aux GPIO du Raspberry Pi, c'est possible20

Premières lignes C# avec Leap Motion23

SLightBox : un système « Ambilight » avec C# et Gadgeteer27

PICAXE : débiter sans peur dans le monde de l'embarqué31

CAS D'USAGE

Prendre en main les capteurs sous Android en réalisant une boussole65

iOS

Objective-C expert : "Associated Object", "Method Swizzling" et "Grand Central Dispatch"69

Développer un explorateur de fichiers Dropbox sous iOS72

WEBMASTER

Devenez le meilleur ami de votre admin, écrivez-lui des modules Powershell77

Mettre en œuvre le protocole WebSocket79



Incroyable, le développeur sert (aussi) à innover !



La phrase culte du mois revient à Phil Schiller qui lors de la présentation du nouveau Mac Pro (que certains soupçonnent d'avoir pactisé avec Dark Vador) a dit : « On (=Apple) ne peut plus innover ? Mon cul ! ». Affirmation claire, non ? L'innovation est au cœur de l'informatique. Cependant, cette innovation prend différentes formes : il y a la vraie nouveauté, la rupture, la proposition technologique et la fausse, qui n'est qu'un lifting d'une technologie ancienne. En informatique, on confond facilement : évolution, innovation et révolution. Le marketing aide beaucoup à cette confusion.

Vous vous dites, bon ok, c'est un fanboy Apple, un partisan de l'Empire, et après ? Imaginez que les sociétés, les employeurs innoveraient réellement envers les développeurs : leur laissent du temps libre pour faire de la veille technologique, les autorisent à travailler sur des projets personnels même au bureau, les incitent à proposer, à réfléchir. Bref, laissent le développeur s'exprimer dans ce qu'il connaît le mieux : le code, la technique, la technologie. Aujourd'hui encore, en France, le développeur est (trop souvent) vu comme un simple exécutant, le bon vieux « pisseur de code » qui coûte et ne rapporte rien. Celui qui met des bugs partout et est parfois asocial (ah notre ami le nerd !).

« Libérez le développeur ». Ce n'est pas toujours possible mais proposez aux développeurs un projet neuf, excitant, avec des défis techniques. Plus le développeur sera expert, plus il sera fasciné par le projet. Prenez le projet GenyMotion de Genymobile, jeune SSII Android française. Ils ont conçu un émulateur Android totalement nouveau. Le projet a duré presque 2 ans, de l'idée à sa présentation officielle. Et quand vous discutez avec ceux qui ont conçu l'outil, on sent l'enthousiasme derrière. C'est cet esprit que j'aime dans le développement, la technologie. Magma Mobile, une belle réussite dans le jeu mobile, est l'illustration de la passion du code et de la technologie dans un marché ultra compétitif.

Et si durant l'été, on s'amuse un peu... A chaque numéro d'été, je me dis, quels gadgets va-t-on pouvoir trouver ? Et il faut avouer que la récolte a été très bonne : Leap Motion, Gadgeteer, Raspberry Pi, Picaxe... De quoi bien s'amuser tout l'été !

Ce mois-ci, nous avons mené l'enquête sur les écoles informatiques, les diplômes et les diplômes qui plaisent aux employeurs. Tous les diplômes se valent-ils ? Sont-ils tous reconnus ? Comment choisir ? Ces questions sont importantes pour l'avenir des futurs développeurs.

Bon été et rendez-vous le 2 septembre.

<embed src= « François Tonic »

/>

mailto:ftonic@programmez.com

Abonnement : Programmez, 17, Route des Boulangers, 78926 Yvelines Cedex 9 - 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 : 30 € (Monde Entier) souscription exclusivement sur www.programmez.com

Editeur : Go-02 sarl, 21 rue de Fécamp 75012 Paris - diff@programmez.com. **Rédaction :** redaction@programmez.com - **Directeur de la Rédaction :** Jean Kaminsky. **Rédacteur en Chef :** François Tonic - ftonic@programmez.com. **Ont collaboré à ce numéro :** F. Mazué, S. Saurel, C. Calais. **Experts :** C. Villeneuve, J. Dollon, L. Ellerbach, P. Legrand, S. Warin, A. Canut, D. Rousset, A. Talavera, G. Lelasseur, S. Pojer, S. Sibué, N. Jozwiak, S. Khalil, V. Saluzzo, M. Perfetti. **Illustration couverture :** image du geek ©StockPhoto/PeskyMonkey - **Publicité :** Régie publicitaire, K-Now sarl.

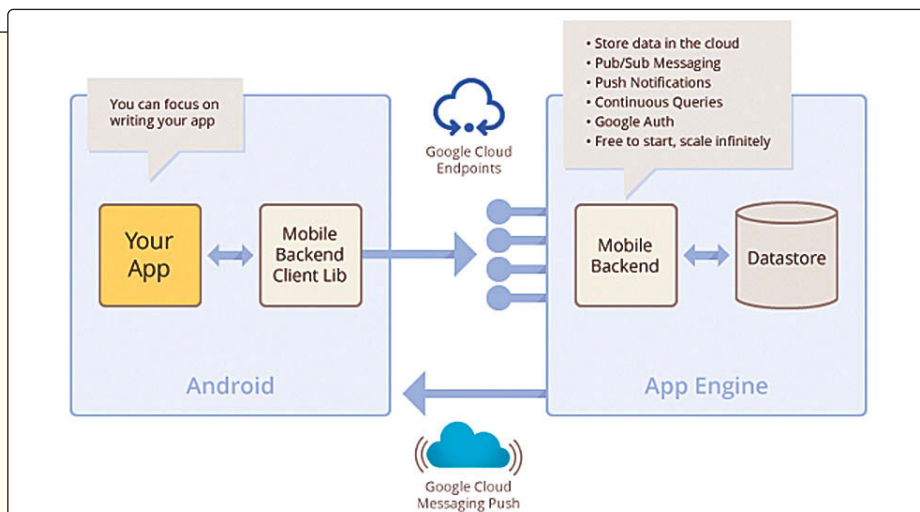
Pour la publicité uniquement : Tél. : 01 41 77 16 03 - diff@programmez.com. Dépôt légal : à parution - Commission paritaire : 0717K78366 ISSN : 1627-0908. Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles Belgique.

Directeur de la publication : J-C Vaudecrane

» Google propose Mobile Backend Starter.

Ce nouveau service va servir dans les applications Android à accéder à des couches serveurs, au stockage en ligne. Il est disponible sur la plateforme Cloud de Google et particulièrement sur App Engine, la plateforme PaaS de l'éditeur. En fait, si on regarde le schéma de fonctionnement, il est très « simple » : une application communique avec le client, Mobile Backend Client Lib, qui lui-même communique avec Mobile Backend (sur App Engine), via Google Cloud Endpoints. Ce dernier assure la connexion entre les deux « mondes ».

On retrouve ici un peu l'idée des services mobiles de Windows Azure. Actuellement, Mobile Backend supporte l'authentification, les requêtes, les notifications, l'échange de messages



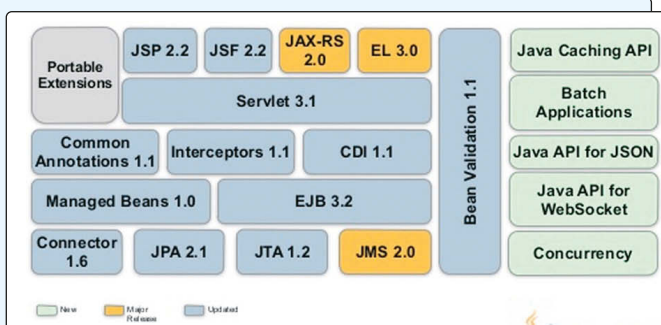
(notamment entre terminaux) via Google Cloud Messaging Push et le stockage de données. Ce service se veut simple d'implémentation et d'utilisation. Il suffit d'activer le service dans le portail Cloud Platform puis d'accéder à sa console

pour l'administrer, la déployer et la démonter.

Présentation complète : <http://goo.gl/oNmQg>
Documentation technique : <http://goo.gl/nWQdy>

» Java EE 7 est sorti des laboratoires Oracle !

Sa sortie est passée presque inaperçue mais Oracle a mis à disposition la version 7 de Java EE le 12 juin dernier. Cette version est la première grosse évolution de la plateforme depuis le rachat de Sun. Cette édition propose de nombreuses améliorations et nouveautés : JSON, JCache, WebSocket et surtout HTML 5. Par contre, Oracle a coupé dans les fonctionnalités pour tenir les délais, ainsi, la partie PaaS Java est désormais attendue dans la version 8. L'arrivée des websocket dans Java EE (JSR 356) permet désormais de créer des applications bidirectionnelles et utilisables avec des clients HTML 5. WebSocket est basé sur TCP, est bidirectionnel, full-duplex. A l'origine, c'était un des modules de HTML 5. Il possède une API et un protocole. La JSR 356 apporte une API standard pour créer des applications websocket. WebSocket est supporté par



Apache Tomcat, Glassfish, Jetty, JBoss, etc. D'ailleurs Oracle semble beaucoup miser sur le support de cette JSR dans Glassfish. Un des enjeux est comment fournir un « HTML 5 côté serveur », capable de monter en charge selon les besoins. Toujours au sujet de HTML 5, nous trouvons JSF 2.2. En effet, JSF supporte tous les attributs HTML 5. D'autre part, il sera possible de changer le thème d'interface grâce à la fonction multi-templating. Chaque template inclut les ressources nécessaires (CSS, JS, images, etc.). Cette fonction est structurée sur le concept déjà connu des Facelets et ressemble à la notion de « site template » de Joomla.

Parmi les autres nouveautés, nous trouvons Concurrency Utilities for Java EE (JSR 236). Il s'agit de fournir les capacités d'exécutions concurrentes pour les applications Java.

Objectif : simplifier l'implémentation de la concurrence sans alourdir le code et l'architecture. La librairie repose sur 4 interfaces :

- ManagedExecutorService
- ManagedScheduledExecutorService
- ContextService
- ManagedThreadFactory

Concurrence :

<http://goo.gl/t4wGP>

JSF 2.2 :

<http://jdevelopment.nl/jsf-22/>

JSR 356 : <http://goo.gl/ER5FO>

Java EE 7 SDK :

<http://goo.gl/Gs4hv>

» Zend annonce le support OpenShift.

OpenShift est la plateforme PaaS de Red Hat, qui a été lancée en version de production « Online » courant juin. Zend a annoncé dans la foulée la disponibilité de Zend Server pour l'édition Online, et prochainement sur l'édition Enterprise (version Cloud privé du PaaS). Cette extension de la collaboration entre les deux sociétés signifie que les clients qui développent des applications PHP seront à même de tirer parti des capacités industrielles de Zend Server, avec les capacités de Cloud ouvert hybride d'OpenShift de Red Hat. Une application peut s'exécuter sur la plateforme applicative Zend Server sur OpenShift Enterprise derrière le firewall, et dans le même environnement que l'offre de Cloud public OpenShift Online. Le service OpenShift Online Silver de Red Hat est disponible depuis le 11 juin 2013 en Amérique du Nord et à partir de 20,00 \$ par mois. Online, sur le principe d'autres PaaS / IaaS, donne accès à un catalogue serveur / applicatif pour créer très rapidement des workloads spécifiques (CMS, PHP, etc.). Par contre, le portail d'administration est encore très limité et Red Hat devra conforter sa stratégie sur OpenShift pour attirer entreprises et développeurs.

DANS LA JUNGLE DU CLOUD, NOUS AVONS PRIS UN PEU D'AVANCE...



b i l z & u o v © Arup Stan / Jürgen Rutkatch / iD Premium / Image Source / Getty Images / Graeme Purdy / Stephen Wallford Photography / Mike Copeland

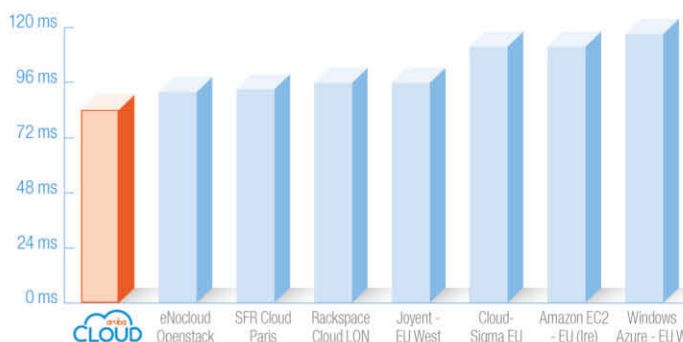
Aruba Cloud, la meilleure offre de Cloud Computing du marché

Le plus performant et le moins cher des Clouds Publics

Fournisseur	Coût mensuel estimé	Performances	Sécurité	Flexibilité
Meilleure proposition			Rackspace	Amazon Web Services
	69 €	★★★★★	★★★★★	★★★★★
Windows Azure	87 €	★★★★★	★★★★★	★★★★★
Amazon Web Services	96 €	★★★★★	★★★★★	★★★★★
CloudSigma	135 €	★★★★★	★★★★★	★★★★★
Rackspace	157 €	★★★★★	★★★★★	★★★★★

Source www.cloudscreeener.com
sur les paramètres standards d'évaluation des offres de Cloud Computing.

Classé n°1 en temps de réponse, temps de connexion et disponibilité



Source : Analyse Cedexis réalisée du 5 mai au 5 juin 2013 depuis la France.
www.cedexis.com

TESTEZ GRATUITEMENT
NOTRE OFFRE

Aruba, le bon partenaire pour bénéficier de la puissance d'un acteur majeur qui considère que chaque client, dans chaque pays, est unique. **MY COUNTRY. MY CLOUD.**

arubacloud.fr | TÉL : 0810 710 300
(COÛT D'UN APPEL LOCAL)

» Comme chaque mois, vous attendez avec impatience l'**index TIOBE** ! Pour le mois de juin, les vainqueurs sont : C, Java, Objective-C, C++ et PHP ! C et Java varient peu, en revanche, Objective-C arrive 3e avec une forte progression et dépasse les 10 %.

Preuve que OS X / iOS sont toujours aussi demandés. A noter, que PHP remonte de deux places. C# et VB subissent une forte baisse.



» Red Hat délaisse MySQL pour MariaDB.

Ce n'est pas la première distribution Linux à utiliser la base de données MariaDB, fork de MySQL. Ce changement était déjà acté dans Fedora et suit la décision de SuSe. Cependant, cette nouvelle arrive mal pour Oracle qui peine à convaincre sur MySQL : quelle stratégie ? Quel futur agenda ? MySQL souffre de ce flou et cela commence à inquiéter les utilisateurs. Ce n'est pas un hasard si MariaDB se pose aujourd'hui comme une alternative au projet mère.

» Oracle a publié un gros patch de sécurité pour Java.

Le 19 juin dernier, un nouveau patch de sécurité pour Java SE était disponible. Oracle annonce plus de 40 fixations de failles. La mise à jour concerne les JDK 7, 6 et 5. Bien entendu, il est fortement conseillé de le déployer.

Site : <http://goo.gl/pxosd>

» Twitter fait le ménage dans ses API.

Twitter a retiré définitivement les versions 1.0 de son API. L'API de référence est désormais la 1.1. Il s'agit d'une révision importante : support de JSON, authentification généralisée, nouvelles guidelines pour l'affichage. Pour en savoir plus :

<https://dev.twitter.com/docs/api/1.1/overview>

» DroidCon Paris 2013

Durant le grand rassemblement de la communauté Android à Paris mi-mai dernier, qui a connu un très beau succès, Genymobile, société de développement française, a montré et démontré son émulateur Android : Genymotion. Le point de départ est le fait que l'émulateur officiel est lent, pas toujours exploitable. Pour pallier ce problème, Genymobile a travaillé durant presque deux ans à son propre émulateur. Le but est de proposer un émulateur Android le plus complet possible, supportant l'ensemble des fonctions (GPS, réseau, OpenGL, écrans multiples, différents profils matériels) disponible sur

OS X, Linux et Windows. L'émulateur est un environnement indépendant mais un plug-in Eclipse est en cours de développement. Genymotion repose sur une couche de virtualisation pour être flexible (couche virtualbox). D'ailleurs, initialement, le projet se dénommait AndroVM. Cette virtualisation permet de monter et démonter très rapidement des environnements de tests.

L'outil est disponible gratuitement. L'éditeur veut récupérer le feedback des développeurs pour le corriger et compléter. A terme, une version commerciale sera proposée.

Site : <http://www.genymotion.com>



» Zend a dévoilé son enquête Pulse 2013.

L'éditeur a interrogé 5 000 développeurs PHP. Quelle est donc la situation du développement mobile ? Un premier constat : 91 % des développeurs interrogés travaillent sur des applications mobiles. Et cela concerne toutes les entreprises. En revanche, on constate une fragmentation des fonctionnalités. Ainsi, 49 % pensent faire de la géolocalisation, 45 % des données client / serveur, 44 % du push notification natif. Par contre, les capacités backend sensibilisent les développeurs. Sur le langage, Pulse 2013 confirme ce que d'autres enquêtes développeurs martelaient : l'usage des applications web

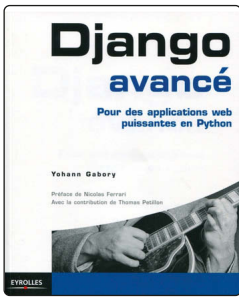
(79 %), les applications hybrides (html / natif) arrivent à 33 % et le pur natif à 26 %.

Une question aborde les raisons des échecs et des retards dans le développement : 41 % évoquent un problème d'environnement de programmation / tests / productions mal adapté, 27 % pointent du doigt des processus de développement non intégrés et automatisés. « Les développeurs perdent également beaucoup trop de temps à corriger leurs applications après les avoir fait passer en production. L'enquête a révélé que 90% des développeurs ont travaillé

week-ends, vacances et jours fériés en raison de situations d'urgence en production. Plus inquiétant encore, lorsqu'on leur demande quel pourcentage de temps ils consacrent à la création de nouvelles fonctionnalités, 80% des développeurs identifient qu'au moins un quart de leur temps de travail est dévolu à la résolution de problèmes. 43% déclarent même que la correction de bugs consomme plus de la moitié de leur temps disponible », précise Zend. Sur la partie Cloud, 68 % disent utiliser du Cloud public (en 2013). Amazon Web Services arrive largement premier.



Notre sélection estivale



YOHANN GABORY Django avancé

Eyrolles

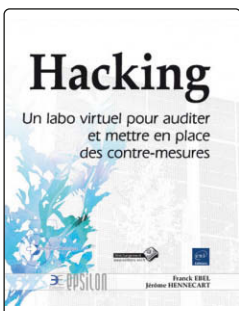
Django est un framework Python MVC, reconnu pour sa puissance. Objectif : développer des applications web rapidement. Avant de vouloir rentrer dans le détail, l'auteur explique tout d'abord comment installer l'environnement (OS X, Linux, Windows). Pour mieux comprendre le fonctionnement et les possibilités du framework, deux cas concrets sont expliqués : un agenda partagé et un tracker. A partir de ces exemples, l'auteur décortique plusieurs fonctions importantes de Django : événement, templates d'affichage, authentification, factorisation de code, la gestion des données, les contribs, etc.



CHRIS APERS iPhone et iPad

Pearson

Même si le nouveau iOS est attendu pour l'automne, vous pouvez dès à présent développer la version 6 pour iPhone et iPad. L'auteur explique comment créer des applications web pour les terminaux Apple. Vous découvrirez l'environnement de travail, les outils de développement et de débogage, avant de passer à la conception proprement dite en HTML et CSS3, puis aux interactions utilisateur avec JavaScript et Ajax. Vous apprendrez à interagir avec les API natives Cocoa et les nombreuses fonctions offertes pour iOS !



COLLECTIF Hacking

édition Eni

Le hacking est un ensemble de techniques que tout développeur ou RSSI doit maîtriser pour en comprendre les mé-

canismes et donc savoir les contrer. Comment vérifier, avant la mise en production, qu'il n'existe pas de faille de sécurité ? Comment tester si les protections mises en place sont efficaces ? Ce sont quelques-unes des questions que les auteurs proposent de résoudre avec l'idée d'un labo virtuel. Dedans, il sera possible d'auditer et de mettre en place les contre-mesures. Pour vous entraîner à cette phase d'attaque, les auteurs vous proposent de mettre en place dans le laboratoire un ensemble d'éléments faillibles. Ceux-ci sont présentés sous forme d'un challenge de sécurité dédié à l'entraînement. Très intéressant à lire !



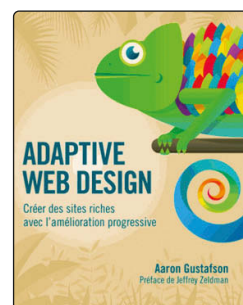
LOÏC THOBOIS IIS 8

édition Eni

Sorti il y a quelques mois, IIS 8 est le serveur web par défaut de Windows Server 2012. Cet ouvrage présente les principales fonctions du serveur web, comment l'installer et déployer des sites, comment configurer la sécurité, optimiser le serveur pour les langages dynamiques. Sans oublier l'administration et la sauvegarde / restauration.

AARON GUSTAFSON Adaptive Web Design

Pearson



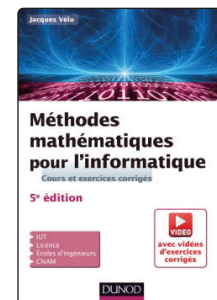
Le Web est en perpétuelle mutation : ses supports et ses usages, sa portée et ses applications ne cessent de changer et d'évoluer. Si vous travaillez dans le Web, vous avez

certainement entendu parler de l'amélioration progressive (progressive enhancement). Mais que recouvre réellement cette notion ? Comment la rendre opérante dans son travail de développement quotidien et dans le design de ses sites ? L'adaptive Web Design est une réponse à ces problèmes du web actuel avec la multiplication des terminaux. L'auteur propose une synthèse claire et pré-

cise, tout en y apportant ses propres idées autour de HTML, CSS, JavaScript. Grâce à l'amélioration progressive, vous remettrez les contenus au cœur de vos développements, de façon à proposer des expériences qui servent les utilisateurs plutôt que les navigateurs. Vous donnerez accès à vos contenus sans limite technologique, pour les supports d'hier, d'aujourd'hui et de demain. Une bonne référence à consulter !

JACQUES VÉLU Méthodes mathématiques pour l'informatique 5e édition

Dunod

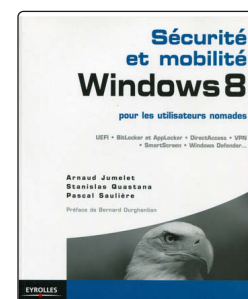


Déjà la 5e édition de cet ouvrage de référence ! L'idée de base est simple : vous donner les connaissances de base en mathématiques pouvant servir à tout informaticien. Bon, dit comme cela, ce n'est pas

forcément très emballant. Et pourtant, cette cinquième édition s'enrichit de cinq nouveaux exercices et problèmes qui font tous l'objet d'une vidéo avec tableau animé et explication audio détaillée de la solution (accessible sur le site dunod.com). Quelques domaines abordés : calcul ensembliste, prédicats, récursivité, loi de composition, etc.

COLLECTIF Sécurité et mobilité Windows 8 pour les utilisateurs nomades

Eyrolles



Si Windows 8 hérite fortement de Windows 7, ce nouveau Windows introduit de nombreuses innovations sur les mécanismes de sécurité pour les postes de travail, les terminaux

mobiles. Cet ouvrage s'adresse à tout utilisateur de Windows 8.

Au programme, de nombreuses technologies comme UEFI SecureBoot, BitLocker, AppLocker, Windows To Go, les cartes à puces virtuelles, SmartScreen, DirectAccess, les bureaux à distance...



Communauté

Le groupe développeurs Google de Nantes

Le GDG Nantes est un groupe passionné par les technologies et services Google : GWT, Android, développement web, App Engine, etc.

Le groupe organise régulièrement des conférences, des apéros, des rencontres et des concours de développement. Ainsi, en juin dernier, deux jours dédiés aux jeux ont été organisés, les Breizh Camp ! Site : <http://www.gdgnantes.com>



Juillet

Groupe utilisateur Unity (Montpellier)



Le MUUG organise deux soirées Unity3D les 11 et 25 juillet à partir de 19h. Ces sessions évoquent le moteur Unity3D, le travail des développeurs du groupe ou des personnes invitées. N'hésitez pas à y aller si vous êtes intéressés. Site officiel : <http://www.unity3d-france.com/unity/2013/montpellier-unity-user-group/>

Août

Afterwork du Guss ? (Paris)

Le groupe utilisateur de SQL Server prévoit un afterwork le 7 août à partir de 20h30. A l'heure où nous bouclons, le thème et le lieu ne sont pas définis. N'hésitez pas à consulter la page : <http://guss.fr/2013/08/07/afterwork-aout-2013/>

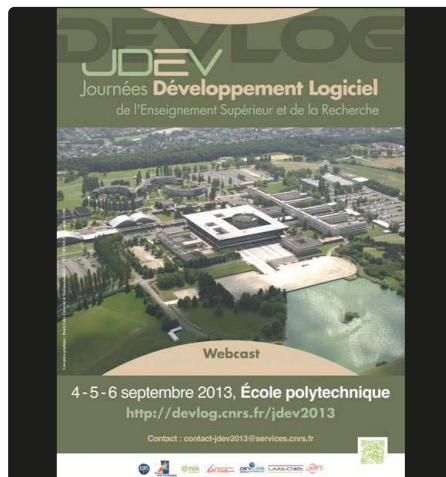
Septembre

JDEV 2013

(près de Paris)

Les Journées développement logiciel 2013 se dérouleront du 4 au 6 septembre prochain à l'école polytechnique (Palaiseau). « Cette manifestation se donne pour objectif d'améliorer la qualité et la pérennité de nos productions logicielles, ainsi que d'ouvrir de nouvelles perspectives avec la présentation des derniers outils et méthodes existants. Les pratiques adaptées au contexte scientifique seront privilégiées », précisent les organisateurs. Huit thèmes seront abordés durant les journées : ingénierie des besoins

logiciels, Open Source, systèmes embarqués, outils de construction du logiciel, méthodes agiles, les tests, développement web, développement pour calculer. Chaque thème aura plusieurs sessions techniques d'une heure. Des ateliers seront proposés. Ces derniers peuvent durer plus de 3 heures ! De nombreux outils et langages seront évoqués et démontrés durant ces trois journées. Inscription et informations : <http://devlog.cnrs.fr/jdev2013>



JUG Summer Camp 2013 (La Rochelle)

En manque de Java durant l'été ? La communauté Java de Poitou Charentes vous propose le 20 septembre une journée entière dédiée à votre langage favori. Le programme n'était pas encore disponible à l'écriture de cette page. Informations : <http://jugsummercamp.com>

JavaOne 2013 (San Francisco)

Du 22 au 26 septembre, Oracle organise la traditionnelle conférence JavaOne. Comme



chaque année, il s'agira de faire le point sur l'univers Java, les technologies, les dernières versions, les futures évolutions. De

nombreuses sessions techniques permettront d'approfondir des fonctions et API spécifiques. JavaOne est aussi organisé en Inde, Chine, Russie, Amérique du Sud et au Japon. Site officiel :

<http://www.oracle.com/javaone/index.html>

Scrum Gathering Paris 2013 (Paris)

Attention ! Voici l'événement Scrum de la rentrée en Europe : la conférence Scrum Gathering ! Elle se déroulera du 23 au 25 septembre à Paris. Elle rassemble les meilleurs spécialistes de la communauté. Preuve de l'importance de l'événement : plus de 260 propositions de sessions ont été déposées à l'organisation ! C'est un lieu unique pour rencontrer, échanger, discuter Scrum ! Informations sur le ScrumAlliance : <http://www.scrumalliance.org>

Rencontres Django (Belfort)

Les 28 et 29 septembre, la communauté Django se rassemblera à Belfort pour parler Django. L'organisation est très simple : « Première demi-journée consacrée aux retours d'expériences sous la forme d'interventions courtes et rythmées : conférences de 15 minutes maximum (plus 5 minutes pour répondre aux questions de l'assistance) et lightning talks de 5 minutes. Vous avez fait quelque chose de génial avec ou autour de Django ? C'est le moment de vous exprimer pour partager votre expérience ! ».

Site : <http://rencontres.django-fr.org/2013/>

Octobre

Qt Developer Days 2013 (Berlin)



Vous utilisez le framework Qt pour vos interfaces ? Ne ratez pas les Qt Developer Days qui se dérouleront du 7 au 9 octobre à Berlin. Parallèlement se tiendront les Qt Developer Days Starter Edition pour tous les développeurs nouveaux sur la plateforme ! Site officiel : <https://devdays.kdab.com>

agenda

ETRANGER

Du 12 au 15 août 2013, USA, Hôtel Hilton San Diego Bayfront de San Diego, en Californie

FileMaker Developer Conference 2013
Au cours de cet événement qui se déroulera sur quatre jours, les développeurs pourront accroître leur expertise technique, mais aussi rencontrer plus de 1 200 passionnés de FileMaker venant de plus de 36 pays, avec lesquels échanger des idées et des astuces.

<http://www.filemaker.com/developers/devcon/>

Surveillance des environnements virtuels

Gestion optimale des ressources avec PRTG Network Monitor

La virtualisation de serveur permet d'optimiser la gestion des ressources existantes. Les administrateurs peuvent par exemple intercepter les pics de consommation de chaque application en attribuant dynamiquement la puissance de calcul et l'espace mémoire. La tendance de la virtualisation dans les SI et les entreprises se poursuit sans relâche. Pour assurer l'efficacité et la fiabilité d'un réseau qui relie les serveurs physiques et le bon fonctionnement des environnements virtuels, une surveillance réseau professionnelle est primordiale et peut même fournir des informations cruciales dès la planification.

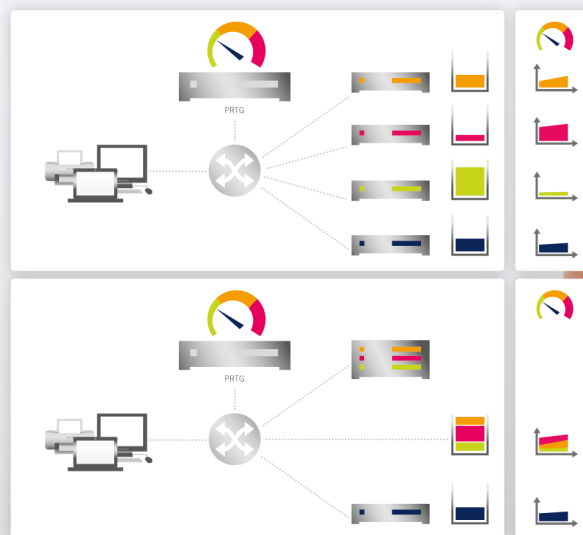
Dans les environnements non-virtualisés, chaque application doit disposer en permanence de ces pleines capacités, pour répondre à un niveau d'exigence maximale à tout moment, car il est souvent difficile de prévoir les fortes fluctuations cycliques résultant de leur utilisation. Cela risque aisément d'induire d'énormes investissements superflus, en puissance de calcul ou en espace mémoire.

Comment assigner dynamiquement des ressources

La virtualisation résout ce problème en assignant dynamiquement les ressources à chaque application, qui peut alors bénéficier des ressources d'une autre application, pour répondre à un besoin ponctuel. Avec la planification des ressources, tous les serveurs et périphériques de stockage d'un environnement virtualisé, peuvent être gérés tel un ordinateur unique. Mais si la consolidation des ressources permise par la virtualisation peut être synonyme de gains de temps et d'économies financières, elle crée également un point unique de défaillance dès lors que tous les serveurs (virtuels) sont consolidés dans un même datacenter ou dans un même rack. Une surveillance drastique de l'environnement s'impose alors, puisque la moindre panne risque de se propager à de nombreux systèmes.

L'utilisation d'un logiciel de surveillance réseau fiable est donc indispensable. Ce logiciel recueille et analyse les données actuelles du réseau et les prépare dans des graphiques clairs et détaillés. Le responsable peut ainsi visualiser en permanence l'activité de son réseau, détecter les faiblesses et améliorer les

« Les environnements virtuels exigent une surveillance réseau performante »



**PRTG
NETWORK
MONITOR**

Stefan, Développeur logiciel chez Paessler



performances réseau avant même qu'un problème n'affecte l'environnement matériel ou virtuel.

PRTG Network Monitor permet une planification précoce des ressources

La solution de surveillance réseau PRTG donne des informations sur la demande présumée de performance de chaque application, dès la planification de virtualisation. Sur la base d'analyses à long terme, ce logiciel permet de définir les tendances et les montées en charge de manière à ce que les administrateurs planifient à un stade précoce les ressources nécessaires et évitent les goulots d'étranglement. PRTG dispose de 130 différents types de capteurs pour surveiller divers aspects d'un réseau informatique. Prenant en charge toutes les technologies de virtualisation actuelles, PRTG propose notamment 10 types de capteurs spécialisés dans la surveillance des systèmes virtuels, comme VMWare, Hyper-V, Virtuozzo, Amazon CloudWatch et Xen.

PRTG garantit ainsi pendant le fonctionnement, la surveillance permanente et sans

faillite des environnements virtualisés comme un aspect de la surveillance continue de l'ensemble de l'infrastructure IT.

Lors d'un événement inhabituel, ou si un serveur est endommagé, PRTG alerte immédiatement le responsable. Ce dernier peut donc gérer les priorités en toute sérénité et assurer à la fois efficacité opérationnelle et continuité de service.

Dans la plupart des cas, grâce aux rapports d'activité générés, l'administrateur peut anticiper les pics des demandes de performances des applications, de même que les failles potentielles, qu'il est en mesure de corriger avant la survenue d'un problème.



Livre blanc : Virtualisation de serveur
www.fr.paessler.com/virtualisation-de-serveur

Paessler AG

T : +49 (911) 9 37 75 - 0
F : +49 (911) 9 37 75 - 409
info@paessler.com
www.paessler.fr
Contact :
Corinne Portenschlager



PHP 5.5 : les nouveautés

L'équipe PHP sort des versions mineures avec des cycles beaucoup plus courts qu'avant, et c'est ainsi que la version de PHP 5.5 vient de sortir. Ces évolutions sont nécessaires pour répondre au mieux au marché et aux attentes des développeurs.

La plupart des améliorations apportées par PHP 5.5 ne posent pas de problème majeur car elles restent compatibles avec les fonctionnalités des versions précédentes. Cependant, il existe tout de même quelques incompatibilités, planifiées depuis quelques versions. L'article effectuera un tour d'horizon de ce que vous devez connaître pour éviter de rester en retard

LES NOUVEAUTÉS

> Nouvelle méthode du mot de passe

Les techniques de cryptage du mot de passe, ont démontré que les fonctions MD5 et SHA ne doivent plus être utilisées pour les champs très sensibles (par exemple identification). Il existe d'autres fonctionnalités de hachage d'un mot pour améliorer la sécurité des comptes utilisateurs, comme Hash() et crypt(). La version d'aujourd'hui propose une amélioration au niveau de l'API, pour rendre plus performant le niveau de hachage et de chiffrement des mots de passe.

Cela se traduit, par la création d'une table de hachage pour un mot de passe. Cette méthode de hachage s'appuie sur la fonction crypt(), et simplifie son utilisation lors de la création du mot de passe, de manière sécurisée.

```
<?php
echo 'Résultat 1 : '.password_hash('rasmuslerdorf', PASSWORD_DEFAULT)."\n";

$options = ['cost' => 7, 'salt' => 'BCryptRequires22Chrcts'];

echo 'Résultat 2 : '.password_hash('rasmuslerdorf', PASSWORD_BCRYPT, $options);
?>
```

Pour obtenir :

Résultat 1 :

```
$2y$10$.vGA109wmRjrwAVXD98HNOgsNpDczlqm3Jq7KnEd1rVAGv3Fykk1a
```

Résultat 2 :

```
$2y$07$BCryptRequires22Chrcte/VlQH0piJtjXl.0t1XkA8pw9dMXTPoQ
```

Comme le montre le résultat, suivant la méthode de hachage définie et sa longueur, le résultat peut-être différent, tout en restant complexe

> Les générateurs

Les générateurs étaient très attendus pour ce langage car ils offrent la possibilité de créer des fonctions personnalisées, ce qui permet de conserver l'état entre les exécutions et d'implémenter des itérateurs sans le coût de l'implémentation d'une classe implémentant l'interface Iterator. Ils ont pour but de faciliter le travail et de rendre notre code plus concis. Par ailleurs, un nouveau mot clef appelé « yield » a été ajouté. Exemples :

```
<?php
function xrange($start, $limit, $step = 1) {
```

```
    for ($i = $start; $i <= $limit; $i += $step) {
        yield $i;
    }
}

echo 'Nombre premier impair : ';

foreach (xrange(1, 9, 2) as $number) {
    echo "$number ";
}
?>
```

Résultat :

```
Nombre premier impair : 1 3 5 7 9
```

Ce générateur permet d'effectuer un chiffrement avec des limites aux extrémités et un saut de pas. Il était possible d'effectuer la même opération avant, mais avec beaucoup plus de lignes de code.

> Finally

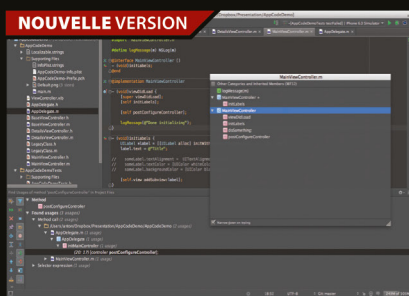
Il s'agit d'une évolution des exceptions étendues (Try/Catch) déjà existantes. L'objectif de celle-ci est de permettre de garantir une portion de code ou de stopper correctement une étape ou une connexion avec la base de données. Nous pouvons résumer comme ceci :

```
<?php
echo «debut\n»;
try
{
    // code Try
}
catch (Exception $e)
{
    // partie catch
}
finally
{
    // partie finally
}
echo «fin \n»;
?>
```

pour obtenir le résultat suivant :

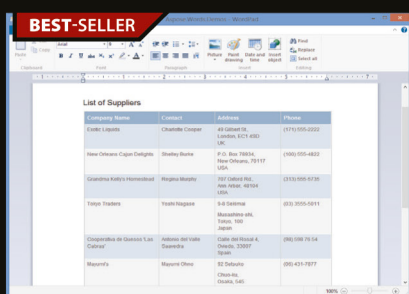
```
debut
try
finally
fin
```

Si une erreur s'était produite, on aurait obtenu le résultat suivant :

**AppCode** à partir de € 87

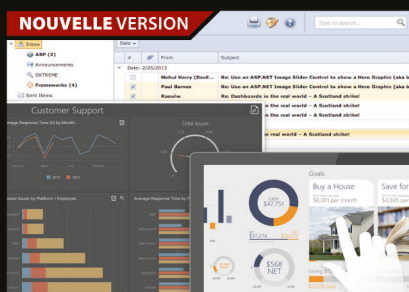
Objective-C IDE intelligent pour accélérer le développement sous iOS/OS X.

- Assistance de codage experte, plusieurs options de génération automatique et navigation rapide
- Analyse approfondie de qualité du code et nombreuses corrections immédiates
- Refactorings sûrs et fiables
- Débogueur puissant et test d'unité avec support des frameworks de test OCUnit et Kiwi
- Support intégré pour Objective-C, C, C++, XML, XPath, JavaScript, HTML et CSS

**Aspose.Words for .NET** à partir de € 763

Lisez, modifiez et écrivez des documents Word sans Microsoft Word.

- Création de documents, manipulation du contenu/formatage, puissante capacité de fusion de courrier et exportation en DOC/HTML
- Accès détaillé à tous les éléments d'un document par programmation
- Support les formats de fichiers: DOC, DOCX, WordprocessingML, RTF, HTML, OOXML, OpenDocument, PDF, XPS, EMF et EPUB

**DevExpress Universal Suite** à partir de € 1 679

400+ outils et contrôles WinForms, ASP.NET, WPF, Silverlight et Windows 8.

- Exploitez votre base de codes pour développer des applications tactiles multiplateformes
- Inclut un tableau de bord de visualisation interplateformes et un serveur de rapports
- Nouveaux contrôles DevExpress incluant Tableur et Carte
- Codage, débogage et refactorisation avec CodeRush pour Visual Studio
- Inclut la nouvelle galerie unifiée de modèles applicatifs DevExpress

**TX Text Control .NET for Windows Forms/WPF** à partir de € 911

Composants complets de traitement de texte pour Windows Forms et WPF.

- Le traitement de texte professionnel pour vos applications
- Chargez, modifiez et enregistrez au format MS Word (DOCX, DOC, RTF)
- WYSIWYG, tableaux imbriqués, cadres, en-têtes, pieds de pages, images, puces, listes numérotées, zoom, sauts de section, etc.
- Exploitez la liaison de données XAML pour lier les barres d'outils et de ruban

```
debut
try
catch
finally
fin
```

Cette évolution évite d'oublier de laisser des connexions ouvertes ou des objets dans la nature.

> Foreach

La fonction `Foreach()` permet de parcourir un tableau à travers une boucle. Maintenant, cette fonction supporte la fonction `list()`, donnant ainsi la possibilité de découper un tableau imbriqué vers des variables distinctes, par exemple :

```
<?php
$array = [
    [1, 2],
    [3, 4],
];

foreach ($array as list($a, $b)) {
    echo «A: $a; B: $b\n»;
}
?>
```

Le résultat obtenu est :

```
A: 1; B: 2
A: 3; B: 4
```

> Array / String

Jusqu'à présent, les `[]` ne pouvaient être utilisés que sur une variable et pas directement lors de la déclaration d'une chaîne ou d'un ARRAY. Maintenant, vous pouvez directement déréférencer pour accéder aux éléments et caractères individuels.

```
<?php
echo ('PHP'[1]) ; // h
?>
```

Le caractère retourné est `h` et il s'agit d'un string.

```
<?php
$tableau = array(1,2,3);
echo $tableau[0];
echo «\n»;
?>
```

La valeur retournée sera `1`

> GD

La librairie GD bénéficie aussi de quelques évolutions pour manipuler facilement les images dynamiquement. Les fonctions disponibles sont :

- La possibilité de retourner une image horizontalement ou verticalement ou les 2 avec la fonction `imageflip()`
- Le découpage d'une zone dans une image avec `imagecrop()` ou `imagecropauto()`
- Un nouveau format de lecture et de sauvegarde est disponible avec `imagecreatefromwebp()`

> Fonctions modifiées

Les fonctionnalités modifiées vont impacter 2 points : le noyau de PHP Core et intl

PHP Core

Tout d'abord, la fonction `set_error_handler()`, permet d'être appelée avec `NULL`, mais aussi retourne le gestionnaire des erreurs ou d'exceptions. Tout comme `set_exception_handler()`. Ensuite, les fonctions `pack` et `unpack()` seront décrites un peu plus loin.

Intl

Il s'agit de fonctions associées avec la possibilité d'avoir des arguments supplémentaires comme le format, le Timezone... Par ailleurs, le formatage de la date bénéficie aussi de ces évolutions. Ainsi, la gestion des dates d'un calendrier est plus précise.

> Les optimisations

L'optimisation passe, entre autres, par la gestion du cache et la version de PHP 5.5 en bénéficie avec la mise à disposition d'une nouvelle extension `ext/opcache`, incluse dans cette version, qui actuellement sera en mode `OFF` (non active). Si le résultat était concluant, elle passerait à `ON` dans une version future. Cette extension fournit une exécution plus rapide à travers la mise en cache d'opcode et d'optimisation.

Les moins

Avec la version de PHP, présentée aujourd'hui, il est important de noter les points négatifs qui suivent une logique planifiée depuis quelques versions antérieures.

> Les incompatibilités ascendantes

Les incompatibilités ascendantes correspondent à l'incompatibilité de systèmes d'exploitation ou de fonctionnalités obsolètes, par exemple :

- L'abandon du support de Windows XP et 2003, mais cette version reste compatible avec les versions Vista et supérieure.
- La modification des fonctions `pack()` et `unpack()` permet la compatibilité avec le langage PERL, en compressant et décompressant les données dans une chaîne binaire par l'intermédiaire de la Lettre Z, qui équivaut à un 'a' dans l'autre langage.
- La sensibilité à la casse ne dépend plus de la locale utilisée pour les noms des fonctions, de classes, et de constantes, pour mieux respecter les règles ASCII.

> Fonctions obsolètes

MySQL

Annoncée depuis très longtemps, l'extension MySQL originale est maintenant obsolète et génère l'erreur suivante : `E_DEPRECATED` lors de la connexion à une base de données. Il est indispensable d'utiliser soit `MySQLi` ou l'extension `PDO_MYSQL`.

> Conclusion

A l'heure où nous rédigeons cet article, la version de PHP 5.5RC3 venait de sortir. Cependant, quand vous le lirez, la version stable de PHP 5.5 sera disponible et vous pourrez l'utiliser sans problème.

Christophe Villeneuve

Consultant pour Neuros, auteur du livre « PHP & MySQL-MySQLi-PDO, construisez votre application », aux Éditions ENI. Rédacteur pour WebRIVER, membre des Teams DrupalFR, AFUP, LeMug.fr, Drupagora, PHPTV.

NOUVEAU !

6h de formation

Vidéo

VBA Excel 2010

Apprendre à
programmer sous Excel

Plus d'1h d'extraits gratuits sur
www.editions-eni.fr/videovbaexcel2010

Découvrez aussi



SAP BusinessObjects
Web Intelligence

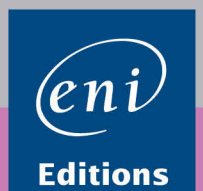


C# 5 sous Visual
Studio 2012



SQL Server 2012

www.editions-eni.fr



Le développement c'est magique !

Stéphane Sibué est tombé très tôt dans la programmation. Après s'être vite spécialisé dans le développement mobile, Il a créé le site CodePPC. Aujourd'hui, Stéphane développe sur les trois principales plateformes mobiles : Android, iOS et Windows Phone 8. Qui est-il ? Comment vit-il la programmation au quotidien ? Voici ses réponses.



Comment es-tu tombé dans l'informatique et plus spécialement dans le développement ?

A 12 ans mon père m'a offert une console Videopac. Parmi les cartouches disponibles, la n°9 m'intriguait beaucoup. Elle permettait de créer de petits programmes en assembleur. Les débuts ont été difficiles pour comprendre ce qu'était un registre ou un accumulateur sans autre aide que le manuel en anglais. La curiosité s'est vite transformée en passion et 30 ans après, elle est toujours aussi forte.

Tout était bon pour programmer, les calculatrices, les ordinateurs de poche, mais c'est avec mon Amstrad CPC464 que j'ai le plus appris (Basic, Assembleur Z80, Turbo Pascal, Logo). J'ai donc voulu faire de ma passion mon métier.

Après quelques années d'études et un BTS en poche, j'ai créé en 91 ma société de développement de logiciels et nous avons sorti en 92 le premier logiciel de gestion de cabinets dentaires sous Windows. Puis les développements se sont enchaînés, plus variés les uns que les autres, et c'est aussi cela que j'aime.

En 1994 je me suis acheté un Psion Series 3A et une nouvelle passion m'a envoûté, complémentaire à la première, le développement mobile. J'ai alors créé en 2001 le site CodePPC pour aider les débutants à se lancer dans le développement mobile. Il s'est ainsi créé une communauté composée de

plus de 6000 développeurs. De 2003 à 2012 j'ai été Microsoft MVP, toujours dans la mobilité.

Pour toi, qu'est ce qui fait que l'on aime toujours et encore le développement, la technique ?

Pour moi le développement est magique. C'est un acte de création pure. On part d'un besoin ou d'une idée, et on crée quelque chose de nouveau. A leur manière, les développeurs sont des artistes et je suis très fier de faire partie de cette grande famille composée de gens formidables.

Tu as gardé un regard très geek : gadget, veille techno. C'est important pour ton job et ta passion ?

Je pense être un geek car mon cœur s'accélère toujours autant quand j'ouvre la boîte d'un nouveau smartphone, d'un nouveau gadget acheté sur le Net, ou même quand je découvre un nouveau jeu sur ma console.

La veille techno est un ingrédient qui rend ce métier palpitant. Essayer de nouvelles techs, comprendre où va notre métier, savoir le plus tôt possible quelles voies privilégier pour faire en sorte que mes clients aient toujours envie de me confier leurs projets. C'est vraiment excitant !

Etre développeur n'est pas toujours facile : pression, évolution constante, frustration des projets et des « chefs », c'est quoi

pour toi être développeur aujourd'hui ? Le job a-t-il changé depuis tes débuts ?

Pour moi, l'évolution constante n'est pas un problème, bien au contraire. C'est la garantie que je ne vais pas m'ennuyer dans mon job.

Pendant ma carrière j'ai travaillé dans le médical, la photo, le tourisme, le jeu, le commerce, etc. J'ai aussi bossé sur des projets scientifiques, et militaires. A chaque fois c'est un nouveau challenge, à chaque fois j'apprends énormément de choses. J'adore cette facette de mon job !

Le métier évolue car les outils utilisés quotidiennement sont de plus en plus puissants et nous permettent d'être concentrés sur le besoin. A une certaine époque, il fallait mettre autant d'énergie dans la résolution du problème que dans la création de la tuyauterie nécessaire au fonctionnement profond des applications. Maintenant, nous pouvons utiliser les fonctionnalités fournies directement par les plateformes pour créer la meilleure solution plus rapidement.

Du coup, il est impératif de connaître les points forts et les points faibles d'une plateforme, ce qui est disponible et ce qui ne l'est pas, ce qui est performant ou non. Quand vous faites un développement sur iOS, Android ou Windows Phone (qui sont les principales plateformes sur lesquelles je travaille), il est impératif de savoir ce qui est différent, ce qui est identique et là où vous allez devoir bosser plus pour réaliser une fonc-

tionnalité. Et comme ces plateformes évoluent très vite, ce n'est pas toujours évident de tout suivre.

Ce qui a changé aussi c'est l'accès aux informations. Avec Internet, vous avez accès à une quantité incroyable d'informations. Il est quasiment impossible de rester bloqué sur un problème de développement, car il y aura toujours quelqu'un qui l'aura eu avant vous et qui en aura parlé dans un blog ou un forum. Internet est un outil fantastique dans notre métier.

Et en dehors du boulot, qu'est-ce que tu aimes faire ? Comment trouves-tu l'équilibre entre travail, vie privée, passion, famille ?

Depuis quelques années mon bureau se trouve chez moi et je travaille seul. Ça n'a pas toujours été le cas. Bosser chez soi a de gros avantages mais ce n'est pas toujours simple pour autant. Il est parfois difficile de faire le break entre le pro et le perso. En fait vous êtes toujours un peu au travail, et dès que vous avez un moment vous vous y remettez. Et le fait d'être passionné n'arrange rien à l'affaire.

Alors pour me détendre j'écoute de la musique, je vais au cinéma. Je suis un fan de science-fiction, de fantastique et j'adore

regarder certaines séries TV. J'aime aussi la natation et les balades et comme je vis à la campagne, près de Chambéry en Savoie, c'est super simple d'aller dans de superbes coins. J'adore passer du temps avec ma femme Corinne et mes fils Kévin et Quentin. Ils sont mon autre passion, et c'est souvent grâce à eux que j'arrive à décrocher.

Etrangement, je me détends aussi en continuant à développer. J'adore créer des applications pour le store Windows Phone, des applications gratuites, juste pour le plaisir de voir qu'elles plaisent au plus grand nombre.

Peux-tu nous présenter ton quotidien en quelques mots ?

Je débute vers 8h. Comme je ne suis pas du matin je commence par les tâches administratives. Vers 9h j'attaque les projets en cours. J'essaie de me focaliser sur un projet précis pendant au moins 2 heures (j'utilise souvent la méthode de la tomate*). Déjeuner vers 12h30, puis reprise vers 14h jusqu'à 18h. Si le planning le permet j'essaie de faire tout ce qui est formation, veille techno, essais de trucs nouveaux en fin de journée, à partir de 18h car c'est là que je suis le plus performant pour ça. Après, vers 20h, dîner en famille puis, suivant la charge de travail du moment, j'en remets une

couche de 22h à minuit, sinon je regarde un bon film avec ma petite femme. Il y a certains jours où je suis en déplacement chez mes clients. Ces jours-là sont assez agréables car ils me permettent de voir du monde, d'échanger, de sortir du bureau, ce sont des jours importants humainement parlant.

Comment vois-tu ton job évoluer ?

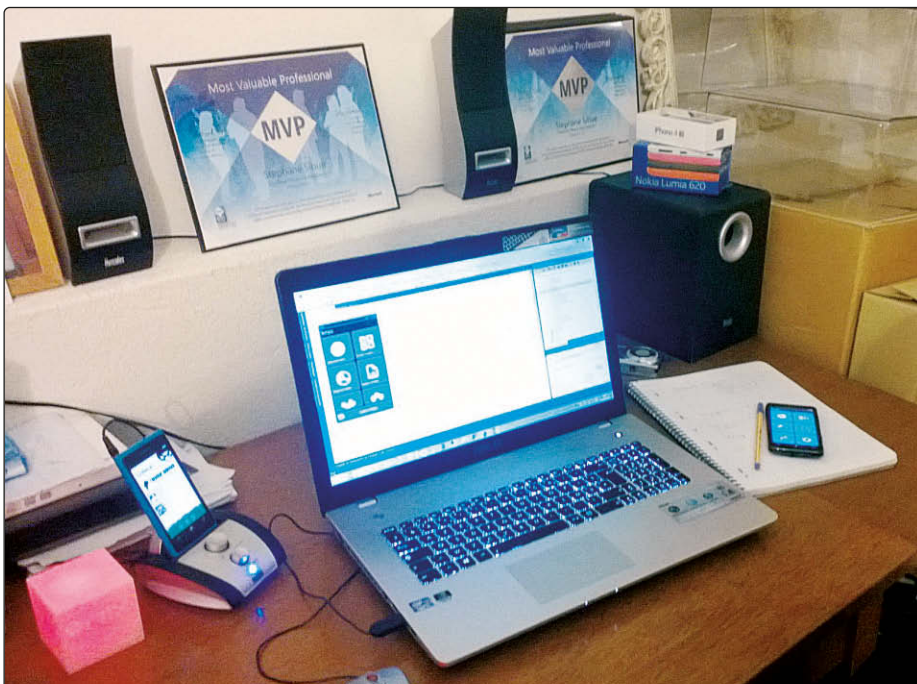
Plus le temps passe et plus nos outils permettent de réaliser presque automatiquement les tâches répétitives, laborieuses et sans véritable intérêt intellectuel. Je pense que plus ça ira, plus nos outils seront performants pour cela. Mais ce que j'aimerais par-dessus tout c'est que les gens voient le côté noble de notre métier. Qu'ils comprennent que nous sommes des « concepteurs » et non des « pisseurs de lignes ». Je souhaite que ce métier retrouve ses lettres de noblesse car sans concepteur pas de logiciel ! Et je suis membre de l'association « fier d'être développeur ».

Des conseils aux étudiants et dévs qui nous lisent ?

Les étudiants ne doivent pas hésiter à se mettre très vite dans le bain. Il existe maintenant des tonnes d'outils gratuits pour se faire les dents très tôt. Rejoignez les communautés de développeurs, participez à des projets ou réalisez tout simplement les vôtres. Ouvrez votre blog et montrez ce que vous savez faire. Partagez, aidez, conseillez à votre tour dès que vous avez la possibilité de le faire. Mon fils Kévin a commencé lui aussi le développement à 12 ans et aujourd'hui, à presque 18 ans, il a un blog, il publie ses applications sur le store Windows Phone, et depuis le début de l'année il est Microsoft MSP. Tout ça, ce sont des plus qu'il pourra mettre sur son CV, soit pour entrer dans certaines écoles, soit plus tard pour trouver un job.

Et pour ceux qui sont déjà dans le métier, le mot d'ordre est « adaptabilité ». A une époque j'ai dû me mettre au dev iOS et Android pour continuer à bosser. Cela n'a pas été simple mais je me suis formé, seul, en quelques mois sur ces deux plateformes. C'est encore une fois une capacité indispensable dans ce job, celle de s'adapter vite, car les technos et les modes changent très vite.

#



Mon bureau

Mon bureau se trouve chez moi. J'ai un PC portable pour les développements .NET et la partie bureautique, et un MacBook pour tous les développements iOS et Android. J'ai aussi un écran que je branche où je veux pour avoir un maximum de confort pendant la partie débogage. Je travaille en musique, avec en fond Radio Meuh, une web radio associative dont je m'occupe et pour laquelle j'ai développé tous les players mobiles.

(*) Cette méthode pour gérer son temps se base sur l'usage d'un minuteur permettant de respecter des périodes de 25 minutes.

12 ans chez Microsoft, développeur et créateur du premier clavier Bluetooth

“ Depuis maintenant un an, je travaille chez Microsoft Redmond dans l'équipe Windows. Cela faisait un bout de temps que je voulais présenter à mes compatriotes, les personnes que j'ai la chance de côtoyer chaque jour, et qui ont un impact énorme sur notre monde (informatique). ”

Il est humble et j'ai parfois l'impression qu'ils ne vivent pas sur la même planète que moi. Alex, comme la totalité de mon équipe, n'a jamais tweeté, il n'utilise pas Facebook, son CV linkedin n'est pas à jour et il n'a pas de blog. Il n'a pas écrit de livre non plus, n'a jamais fait de conférence, mais est toujours disponible pour aider les autres.

« C'est énorme ce que tu as fait ! »

Il travaille chez Microsoft depuis 12 ans, avec un impact phénoménal, avec une passion saine et pure pour ce qu'il fait. On est bien loin de mon cliché de programmeur qui souhaite à tout prix marquer l'histoire. Quand vous lui dites « Non mais attends, c'est énorme ce que tu as fait » il vous répondra « tu penses ? ». Alors voici, la première interview d'Alex Hui, originaire de Hong Kong, Canadien et développeur chez Microsoft.

Entretien réalisé en anglais, mais Alex est trilingue, français/anglais et chinois, mes commentaires sont entre parenthèses.

Quand as-tu commencé à travailler chez Microsoft ? Raconte-moi ton parcours.

Je suis arrivé en 2001, c'est-à-dire il y a 12 ans, j'avais 24 ans à l'époque. Dès le départ, j'ai intégré le groupe Microsoft Hardware. C'est la division qui est en charge de construire 100% du hardware chez Microsoft, des souris, en passant par les claviers



et plus récemment la Surface et Pixel Sense.

J'ai intégré Microsoft après 5 ans d'études à Toronto (Canada). Ces 5 ans m'ont permis d'obtenir un bachelor en Sciences Appliquées. Sciences Appliquées est un mixte entre informatique (computer science) et électronique (electrical engineering). Mes études se concentraient essentiellement sur l'algorithmique, les langages, l'informatique en général (mais rien de très orienté technologie pure) ainsi que l'électronique.

Durant ma quatrième année de cours, j'ai eu l'occasion de faire 1 an de stage. J'ai eu la chance de pouvoir travailler chez Symantec, où J'ai participé notamment au développement de WinFax. J'ai pu financer mes études en travaillant chaque été sur des projets de développement.

Pourquoi avoir choisi Microsoft ?

C'était une sorte d'accident [rire], je pensais rester vivre à Toronto toute ma vie, mais un jour Microsoft est venu dans notre université. Ils ont fait des entretiens directement avec les étudiants.

J'étais curieux et je me suis lancé. L'entretien était plutôt court, et a donné lieu ensuite à un entretien téléphonique de plusieurs heures puis à un entretien chez Microsoft.

C'est lors de cet entretien sur place que je me suis dit « pourquoi pas Microsoft ! », j'ai trouvé le campus magnifique, énorme, ce n'était même plus un campus mais carrément toute une ville.

L'entretien fut long et fatigant, avec de nombreux problèmes à résoudre, des questions de créativité, de project management, des questions sur ma façon d'interagir avec les autres.

Quel job fais-tu chez Microsoft ?

Depuis le premier jour je suis un SDE, c'est-à-dire Software Development Engineer (un codeur !), j'ai commencé en tant que junior et je suis maintenant SDE II. J'ai travaillé sur énormément de projets, je suis plutôt un développeur bas niveau (driver / C++) même si je suis un grand passionné de XAML et de Windows Phone. Le projet auquel j'ai participé qui a eu le plus d'impact fut le premier clavier Bluetooth au monde ! [Note, il a fallu que je lui arrache le mot « au monde » et « premier » il n'osait pas le dire].

Bluetooth était pour la première fois disponible dans Windows XP SP1. Mon travail a consisté à faire la petite couche software qui permet d'initialiser le clavier avec l'API Bluetooth. J'ai travaillé comme un fou sur ce projet car c'était une première, il a fallu faire

des choix importants et accompagner les développeurs de l'API Bluetooth pour avoir quelque chose de fonctionnel.

J'ai aussi codé le logiciel/driver pour Touch Mouse avec la reconnaissance de gestes ainsi que l'arc touch [A noter qu'il était à chaque fois le seul développeur].

"I chose to be an SDE, because as a dev we produce things"

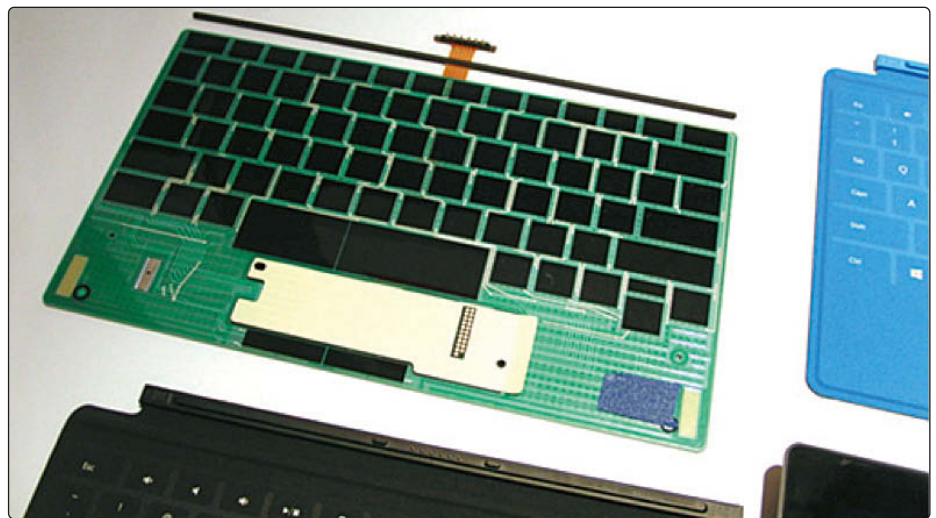
J'ai eu la chance d'être le lead du projet sur le produit IntelliPoint et IntelliType et j'ai vraiment aimé cela, j'ai pu faire partie de la triade [c'est-à-dire 1 représentant des dev, des tests et des PM qui décident des fonctionnalités]. L'un des derniers projets auquel j'ai participé fut le Touch Cover pour Surface.

[Souvent, on imagine des centaines de développeurs derrière certain produits alors que parfois ce n'est pas le cas. Parfois nous ne sommes qu'une équipe de seulement 1 développeur, 1 PM et 1 testeur et on arrive à faire des choses vraiment super.]

Quels sont les « forces » qu'un développeur junior doit avoir pour intégrer ton équipe ?

Microsoft n'attache pas d'importance aux « skills » à un instant T mais plutôt au potentiel. Quand nous avons quelqu'un en entretien, on essaie toujours d'entrevoir toute la puissance qui se cache en lui.

On apporte un grand intérêt à sa passion également. Et plus que tout, à sa capacité à résoudre des problèmes. Je préfère quelqu'un qui n'est pas forcément un pro du C# mais qui aura passé son temps à faire des activités extrascolaires qui lui ouvriront l'esprit, qui lui permettront d'améliorer sa créativité, sa façon d'être avec les autres, qui démontreront sa motivation et sa passion. Par exemple, un des « plus » qui m'a aidé à être embauché fut le fait d'avoir créé avec des copains une voiture avec des panneaux solaires pour faire le tour des USA sur la route 66.



Souhaites-tu rester dans le développement ou penses-tu évoluer vers un autre métier ?

J'ai été attiré pendant un moment par le job de manager car j'aime apporter quelque chose aux autres. J'ai donc tenté l'aventure mais je me suis vite rendu compte que je n'avais plus de temps pour développer. Et pourtant c'est ce que j'aime faire... ! Du coup, je suis repassé développeur assez vite. Mon objectif est désormais de devenir senior SDE. Ce statut est difficile d'accès car il nécessite de nombreuses capacités. Etre expert en développement te permet de monter les échelons jusqu'à SDE II, au-delà on parle plutôt d'impact et leadership.

En gros, si je compare aux arts martiaux, un senior, c'est celui qui est ceinture noire et capable de « défoncer » tout le monde ?

Non, il n'est pas forcément techniquement plus fort que les autres, mais il est capable de savoir tout ce qui se passe dans la pièce, il a conscience de tout, et quand il se bat, il ne se bat pas pour lui mais pour le bien de toute l'équipe.

Si vous avez des questions à lui poser vous pouvez le contacter à travers mon blog : julien.dollon.net

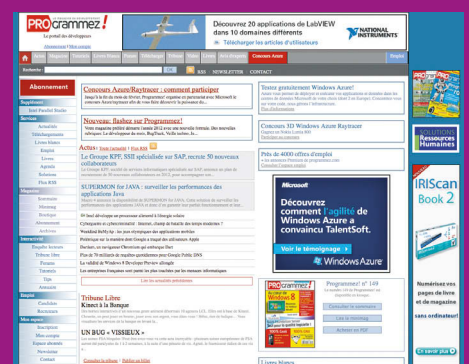
Julien Dollon

L'information permanente

- L'actu de Programmez.com : le fil d'info quotidien
- La newsletter hebdo : la synthèse des informations indispensables.

Abonnez-vous, c'est gratuit !

www.programmez.com



100% Geek en vacances !

Ce n'est pas parce que c'est l'été qu'il ne faut rien faire. Que serait un vrai geek ou un nerd, et bien entendu un développeur, sans un peu de code, de technologies, à la plage, à la montagne, ou franchement installé sur un transat ? Programmez ! vous a déniché quelques devoirs d'été !

Tout d'abord, plongez dans les entrailles de Leap Motion, ce petit boîtier capable de piloter applications et systèmes avec les mains ! Puis nous allons nous amuser avec les effets lumineux Ambilight de Philips, le tout avec un peu de C# et beaucoup de Gadgeteer ! Notre 3e devoir d'été sera (notre chouchou geek à la rédaction) : Raspberry Pi ! Et comme nous ne faisons pas les choses à moitié, nous allons installer du .Net microframework sur ce matériel ! Et en plus, cela fonctionne très bien !

La Rédaction



©StackPhoto/PaskyMonkey

Faire tourner .NET avec l'accès aux GPIO du Raspberry Pi, c'est possible !

Autant le dire tout de suite, je ne suis pas un grand fan de Linux et ne l'ai jamais été. En revanche, je ne dirais pas la même chose de tout ce qui touche à l'embarqué, donc impossible de faire l'impasse sur le Raspberry Pi !

L'intérêt est pour moi d'avoir une plateforme complétant mes Netduino (<http://www.netduino.com>) qui font tourner .NET Microframework directement sur le chip, sans OS. Ce qui m'intéresse particulièrement c'est la partie interface graphique. En voulant installer Mono et avoir accès aux entrées et sorties, je me suis rendu compte que rien n'était vraiment documenté et les forums remplis de questions autour du portage de .NET Microframework sur la plateforme sans vraie réponse. Voici donc un document étape par étape.

> Sur Raspberry Pi

Chargez l'image wheezy, mettez-la sur une SD, bootez le Raspberry relié à un écran, configurez-le lors du premier boot. Pensez à étendre l'image, qu'elle prenne l'ensemble de la SD, pensez aussi à changer le clavier pour un clavier français, et n'oubliez pas la touche Alt Gr. Vous pouvez aussi installer le pack français, changer le fuseau horaire, sélectionner le boot sur le desktop automatiquement. Ne changez pas le mot de passe à ce moment-là, vous seriez en clavier qwerty ! Ma première étape a été l'installation de RDP. Cela me per-

met de laisser branché le board à une simple prise réseau (ou en wifi) et un chargeur 5V micro USB. Ouvrir le terminal LXTerminal, tapez :

```
sudo apt-get update
sudo apt-get install xrdp
```

Cela va mettre à jour la liste des applications et packages, puis installez xrdp qui fournit le service RDP. Attention, il y a quand même des limitations, vous ne pourrez pas l'utiliser derrière une passerelle RDP, la sécurité n'est pas assez forte. Mais vous pourrez y accéder sans problème depuis votre intranet. Ce qui me permet aussi d'y accéder depuis internet en me connectant sur mon serveur.

La commande "ifconfig" vous permet de récupérer l'adresse IP et l'adresse MAC. Je vous recommande de configurer votre serveur DHCP/box pour lui fournir tout le temps la même IP. Bien plus facile pour l'accès à distance. Idem pour un accès en http. Vous pouvez débrancher clavier, souris, écran et vous y connecter à distance en connexion bureau à distance. L'étape suivante est l'installation

Mono. Nous n'aurons besoin que du minimum puisque nous ferons tous les développements et la compilation sur notre machine Windows. Tapez la commande :

```
sudo apt-get install mono-runtime
```

Nous aurons ensuite besoin d'installer Git pour récupérer des sources depuis Github si besoin.

```
sudo apt-get install git
```

Maintenant, on arrive à l'installation de la librairie bcm2835 qui permet l'accès aux entrées sorties. Toujours à partir du terminal, tapez :

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.25.tar.gz
tar -zxvf bcm2835-1.25.tar.gz
cd bcm2835-1.25
./configure
make
sudo make install
cd src
cc -shared bcm2835.o -o libbcm2835.so
cd
rm bcm2835-1.25.tar.gz
```

Ces lignes récupèrent les sources de la librairie d'accès aux GPIO, extraient les fichiers, compilent la librairie, créent une copie partagée de la librairie puis suppriment l'archive.

Copiez maintenant le fichier libbcm2835.so soit sur une clé USB, soit à travers le réseau sur votre machine Windows de développement. Il sera nécessaire que cette librairie se trouve avec le programme que vous ferez en .NET.

> Sur Windows

Le développement se fera directement depuis Visual Studio. Je recommande évidemment la version 2012. Il existe des versions Express gratuites. Des classes de wrapper de cette librairie bcm2835 ont été développées. Le code est disponible sur <https://github.com/cypherkey/RaspberryPi.Net/>. Chargez les sources, décompressez-les et ouvrez-les avec Visual Studio. Changez la cible dans les propriétés du projet en .NET Framework 4.0. En effet, mono ne propose pas l'ensemble des classes de .NET, c'est un sous ensemble qui est majoritairement compatible avec .NET 4.0. Il manque certaines classes et certaines fonctions. Je vous renvoie à la documentation de mono pour en savoir plus. Dans mon cas, pour faire des choses simples, cela fonctionne très bien.

Ajoutez au projet la librairie que vous avez compilée sur le Raspberry. Et mettez là dans vos déploiements avec la dll qui sera générée. Je vous encourage à être curieux et regarder le code du wrapper. Il utilise du code unsafe et l'import de dll. A la place d'être une dll, c'est directement la librairie so qui est importée [Fig.1]. Il ne reste plus qu'à créer un nouveau projet pour tester tout cela. Un code de base est disponible sur le site où se trouvent les

sources du wrapper. Pensez à référencer la dll générée. Changez également la cible de ce projet pour .NET Framework 4.0. J'ai utilisé le code exemple pour faire clignoter une LED sur la sortie GPIO 7, soit le pin 11, la masse est le pin 9. Le montage est très simple, une LED en série avec une résistance en série. Attention, toutes les entrées et sorties du Raspberry sont en 3,3V et ne sont pas tolérantes au 5V ! Protégez bien vos montages [Fig.2].

Compilez le projet, copiez l'exécutable, la dll générée et la librairie so dans le même répertoire sur le Raspberry. Allez dans le répertoire où vous l'avez copiée. Le programme nécessite des droits d'administrateurs pour se lancer, tapez :

```
Sudo mono votrepourprogramme.exe
```

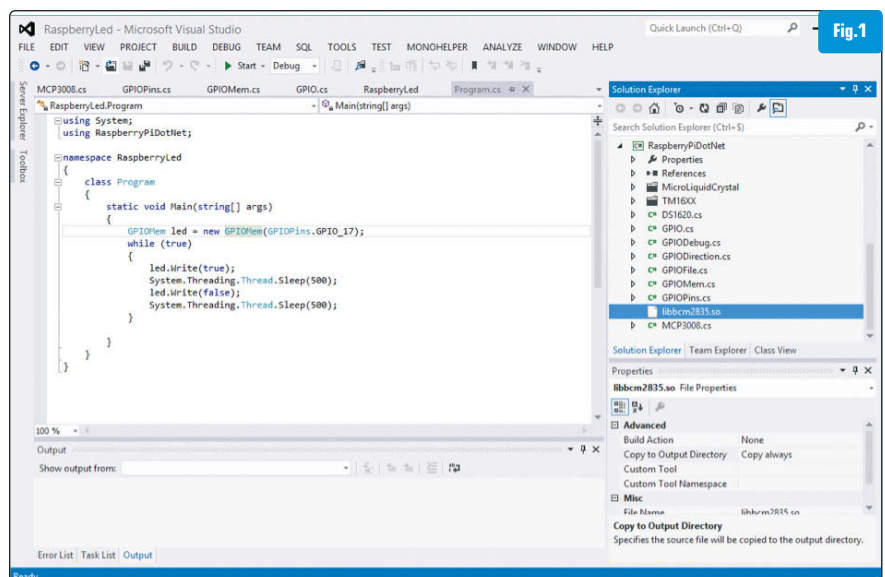
Pour aller plus loin dans les GPIO disponibles, faire les bons branchement, le détail des entrées sorties se trouve sur http://elinux.org/Rpi_Low-level_peripherals. Encore une fois, mono n'est qu'une implémentation partielle de .NET. Restez simples dans votre code.

Pour aller plus loin : pilotage d'un écran Nokia 5110 en SPI

J'ai décidé d'aller beaucoup plus loin car j'ai commencé à implémenter les classes de type SPI, OutputPort, PWM que l'on trouve dans .NET Microframework. Et histoire de commencer par quelque chose de plutôt sympa, je me suis mis à porter l'ensemble des classes SPI. Pour ce faire, j'ai extrait les définitions de classes. Ensuite, j'ai utilisé le même principe que l'exemple précédent en important les fonctions de la librairie à l'aide de dllimport. L'ensemble des fonctions présentes dans la librairie est disponible sur <http://www.airspayce.com/mikem/bcm2835/index.html>.

Attention lorsque vous importez des fonctions implémentant des tableaux, il vous faut impérativement utiliser le marshalling. Dans notre cas, plusieurs fonctions doivent être manipulées avec précaution comme par exemple :

```
[DllImport("libbcm2835.so", EntryPoint = "bcm2835_spi_transfernb")]
static extern void bcm2835_spi_transfernb([MarshalAs(UnmanagedType.LPArray)]byte[] tbuf, [MarshalAs(UnmanagedType.LPArray)]byte[] rbuf, UInt32 len);
```



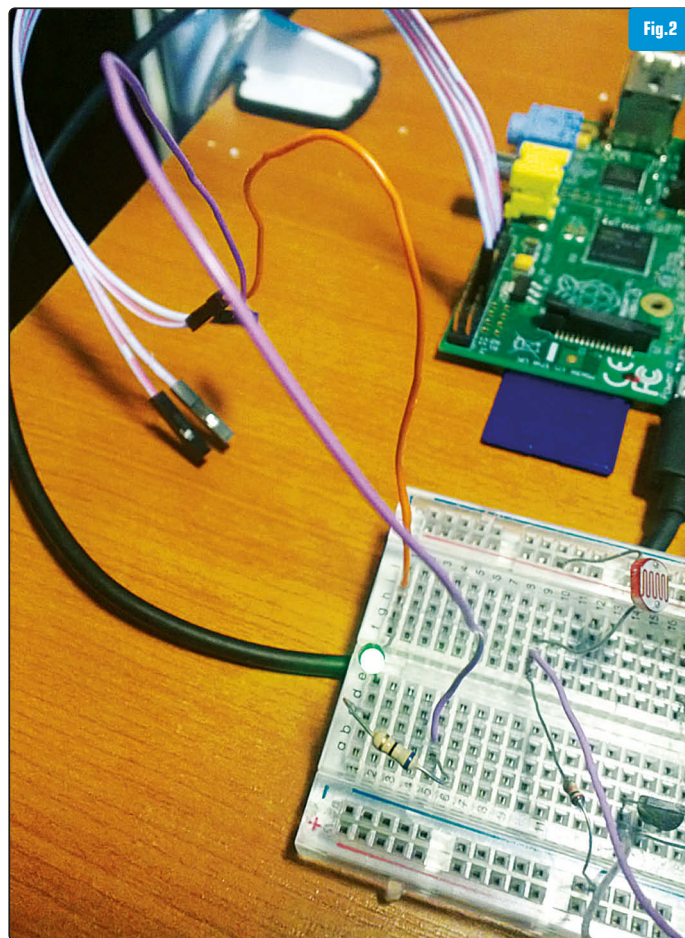
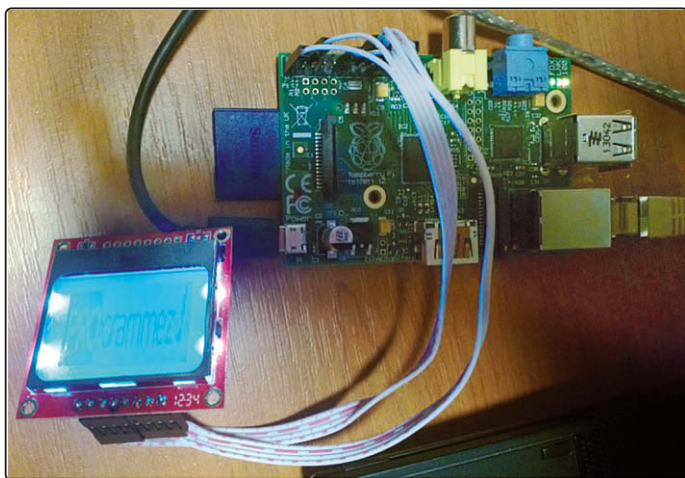


Fig.2

Cette fonction permet d'écrire et de lire des données en même temps. Elle propose un tableau de byte en entrée et en sortie. Les MarshalAs permettent de définir le type de donnée qui doit être transmise. Les tableaux.NET sont des objets et sont loin d'être des données simples. Il est donc nécessaire de les « transformer » avant de les donner à la fonction. En l'occurrence, il s'agit de fournir un pointeur sur les données du tableau en tant que tel. Le protocole SPI (Serial Peripheral Interface) permet d'envoyer en série ou lire à l'aide d'une horloge des informations à très grande vitesse.

Elles sont bien souvent récupérées par des périphériques nécessitant de nombreuses données telles que les écrans LCD ou en lecture les Webcam.

Une fois les principales fonctions implémentées est venu le moment de tester. Et là, problème. En même temps, je ne m'attendais pas à ce que cela fonctionne du premier coup :-). En creusant un peu et sans débogueur [merci les Console.WriteLine], je m'aperçois que la fonction d'écriture pose problème lorsqu'il y a un nombre important de bytes à envoyer. Elle bloque et fait passer le processeur du Raspberry à 100%. La fonction originale que j'ai écrite est :

```
public void Write(byte[] writeBuffer)
{
    bcm2835_spi_writenb(writeBuffer, (UInt32)writeBuffer.Length);
}
```

Je l'ai donc remplacé par une fonction équivalente qui au lieu de tout écrire en une seule fois va écrire byte par byte :

```
public void Write(byte[] writeBuffer)
{
    for (int i = 0; i < writeBuffer.Length; i++)
        bcm2835_spi_transfer(writeBuffer[i]);
}
```

Et là surprise, cela fonctionne ! J'essaie également les fonctions équivalentes, écrivant puis lisant des tableaux de données. Cela fonctionne également. Du coup, pas de doute, le problème doit venir de la librairie Linux. J'ai donc ouvert les sources et recherché la fonction. En modifiant les sources (en C++), pour ajouter la lecture d'une donnée lors de l'écriture d'une donnée, le tout fonctionne parfaitement. J'ai donc fait la modification des sources et recompiler la librairie pour que le tout fonctionne. Je n'ai pas suffisamment creusé

pour vérifier si c'est un problème de paramétrage du chipset ou un bug. J'ai juste « patché » le code pour qu'il fonctionne. L'autre possibilité sans toucher au code est d'utiliser la fonction « bcm2835_spi_transfernb » qui écrit le buffer et lit un autre buffer. Il suffit d'ignorer ce qui a été lu.

Au final, j'ai réussi en une grosse demi-journée à écrire la classe SPI et à la faire fonctionner avec un écran LCD de téléphone Nokia 5110. J'ai récupéré un code pour le piloter sur mon Netduino (<http://wiki.netduino.com/Nokia-5110-LCD.ashx?HL=nokia>). Ces écrans coûtent moins de 4€ et permettent d'apporter dans les solutions embarquées un affichage à très faible coût ! Ne vous en privez pas. Sans changer la moindre ligne dans le code, à l'aide des classes SPI, OutputPort, PWM et Cpu.Pin que j'ai implémentées, j'ai réussi à faire fonctionner exactement le même code sur mon Netduino en .NET Microframework et sur le Raspberry avec mono.

La seule chose qui change c'est l'initialisation de la classe Nokia et la référence using pour utiliser le namespace contenant les fonctions du Raspberry plutôt que celui du Netduino. Le code de ces classes sera disponible sur mon blog (<http://blogs.msdn.com/laurelle>).

Voilà donc une bonne façon de tirer parti du Raspberry et de son prix attractif pour réaliser des développements en .NET et se servir des GPIO. Même si les GPIO disponibles sont assez limités et uniquement numériques, la solution reste attractive pour être interfacée avec des modules i2c ou SPI. Il existe des convertisseurs analogiques au besoin. Et le nombre d'entrées et sorties numériques en fait un outil intéressant pour un développeur .NET.

Laurent **Ellerbach** - Audience Marketing
Microsoft EMEA - DPE - Central and Eastern Europe

Premières lignes C# avec Leap Motion

Le 21 mai 2012, la startup californienne Leap Motion créait le buzz parmi les technophiles en annonçant, vidéo à l'appui, l'arrivée imminente de son capteur de mouvement, le Leap Motion Controller, et ce, à un prix très abordable ! (actuellement disponible en précommande à 80 dollars). Ce capteur, à peine plus grand qu'un briquet, est d'une précision (détection au centième de millimètre près) et d'une fluidité remarquables. Il ouvre la porte à un monde où l'informatique ambiante à la "Minority Report" devient une réalité.

Initialement prévue pour décembre 2012, la sortie du Leap Motion Controller a été reportée à plusieurs reprises pour finalement être arrêtée au 22 juillet prochain. La société Leap Motion souhaite en effet permettre à ses équipes d'approfondir leurs tests, d'améliorer les performances du produit ainsi que son intégration à Windows, Mac et Linux, afin d'offrir une expérience utilisateur unique en son genre. Nous l'aurons compris : l'ambition de cette start-up est bien de marquer les esprits en présentant une avancée technologique aboutie et de faire participer les développeurs à ce projet novateur.

> Architecture matérielle

Depuis ses débuts, le Leap Motion Controller a beaucoup évolué, en se miniaturisant par exemple au fil des versions. Cependant, depuis son annonce officielle, le form factor, ainsi que le hardware, se sont stabilisés [Fig.1]. Le capteur est constitué de deux caméras et de trois LED infrarouges : rien de très excitant côté matériel. A priori, la force du capteur réside dans sa partie logicielle. Matériel et logiciel permettent d'obtenir une détection jusqu'à 200 fois plus précise que celle de l'autre grand nom des capteurs, j'ai nommé le Microsoft Kinect. Beaucoup les comparent mais selon moi, ils sont davantage complémentaires que concurrents. Par exemple, le Leap Motion Controller dispose d'une très grande précision à courte distance, alors

que le Microsoft Kinect nécessite un minimum de recul. De même, le Leap Motion Controller ne permet pas d'accéder aux images prises : on manipule uniquement les trames remontées par le périphérique.

> Les outils Comment développer ?

Avant de se lancer dans le développement d'une application utilisant la technologie Leap Motion, voici certaines notions incontournables à intégrer :

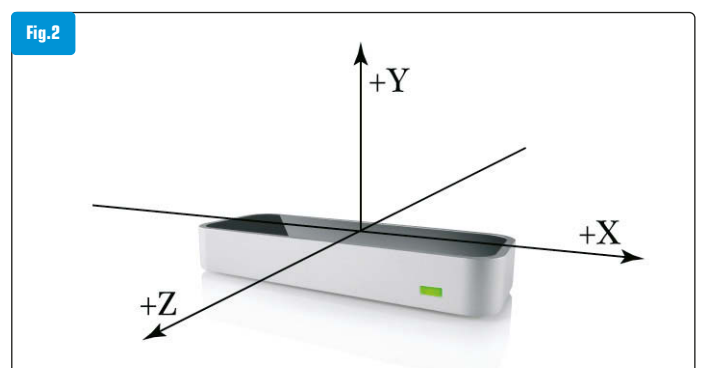
- Un repère cartésien en millimètres, dont la zone de détection est une pyramide inversée de 25 millimètres à 60 centimètres [Fig.2].
- Des éléments traçables :
 - Des objets pointeurs (doigts, stylos, etc.)
 - Des objets mains (jusqu'à quatre)
 - Des gestuelles (attention : seuls les éléments visibles depuis le capteur sont remontés et modélisés)
- Le SDK Leap Motion cible plusieurs plateformes et langages. Ainsi, il est disponible sous Windows, Mac et Linux, pour les langages C++, C#, Objective-C, JAVA, Python et Javascript. Cependant, dans tous les cas, le Leap Motion Controller repose sur les mêmes notions et objets.
- Pour utiliser le capteur, il faut :
 - Démarrer le capteur en lançant l'application Leap Motion fournie lors de l'installation du SDK.



- Instancier un objet "Controller", une sorte de repository ou business manager qui permettra de récupérer les trames du capteur.
- Instancier un objet "Listener", un liseur de trames du capteur.

Note : il est théoriquement possible de brancher plusieurs capteurs à la même machine. Pour les férus de Design Patterns, il s'agit d'un "Observer".

- Associer le "Listener" au "Controller" via la méthode "AddListener()".
- Appeler la méthode "Frame()" du "Controller" afin de récupérer une trame contenant les différentes modélisations des objets pointeurs, mains et gestuelles reconnus par le capteur.



- Les objets Pointeurs contiennent notamment les informations suivantes :

- Longueur et Largeur
- Vecteur Direction
- Position
- Vitesse en millimètre par seconde [Fig.3 et 4].

La fonction "IsTool()" permet de savoir si on a affaire à un outil ou à un doigt. Les différentes fonctions sont communes à tous les langages et sur tous les environnements, moyennant du sucre syntaxique : que l'on soit en C#, C++, JAVA ou Javascript, cette fonction existe et permet de déterminer la nature de l'objet pointeur. Pratique !

- Les objets Main, quant à eux, contiennent les informations suivantes :

- Position de la paume
- Vitesse
- Vecteur Direction
- Vecteur Normal

- Centre de la sphère
- Rayon de la sphère [Fig.5 et 6].

Les gestuelles sont actuellement au nombre de quatre et à activer explicitement dans votre code : [Fig.7].

1. Cercle
2. Swipe
3. Key Tap
4. Screen Tap

- La gestuelle "Cercle" correspond au mouvement de rotation de deux doigts.
- Le "Swipe" correspond à un mouvement de balayement de la gauche vers la droite ou de la droite vers la gauche.
- Le "Key Tap" correspond au mouvement effectué par un utilisateur qui appuierait sur une touche de clavier virtuel.
- Enfin, le "Screen Tap" correspond au mouvement effectué par un utilisateur qui appuierait sur un écran tactile virtuel.

> Ma première application : piloter un AR Drone avec le Leap Motion Controller

En C#, il nous faut lancer Visual Studio et créer un programme Console ou WPF. J'ai pour ma part, choisi une application WPF 4.5, avec deux boutons :

- Le premier bouton me permettra d'initialiser le Leap Motion Controller
- Le second, de récupérer une trame.

J'édite les propriétés de mon projet WPF afin d'indiquer à Visual Studio qu'il s'agit finalement d'une application Console. Ceci me permettra de lancer l'application WPF et de voir simplement mes données. Ensuite, je

référence les DLL en provenance du SDK. Via un clic droit sur "Références", j'ajoute une référence et je récupère la DLL "LeapCharp.Net4.0.dll" présente sous "LeapSDKlib". Si votre projet cible la version 3.5 du framework, alors référencez l'assembly "LeapCharp.Net3.5.dll". Dans le cas d'un projet en .Net 4.5, référencez l'assembly "LeapChar.Net4.0.dll" [Fig.8].

Attention : cela n'est pas suffisant. En effet, LeapCharp.Net4.0 n'est qu'un wrapper sur des DLL natives. Vous ne pouvez donc pas les référencer via "Ajouter une référence". Pour cela, il faut "Ajouter un élément existant", sélectionner le filtre "Tous les fichiers" puis pointer sur les DLL "Leap.dll" et "LeapC-Sharp.dll". En fonction de la plateforme cible (x64 ou x86), ces DLL se trouvent sous "LeapSDKlib\x64" et "LeapSDKlib\x86". Sélectionnez ensuite "Toujours copier" dans la fenêtre "Propriétés" et "Aucun" dans "Action de génération".

Toutefois, si vous lancez dès à présent votre petit programme, il y a de fortes chances pour qu'une erreur "PInvoke" survienne. En effet, sur un PC 64 bits, l'option "Préférer 32 bits" est cochée au sein de l'éditeur de propriétés de génération de build de Visual Studio. [Fig.9 et 10]

Dans mon fichier "MainWindow.xaml", je déclare mes deux boutons ainsi que leur "click handlers" associés (on pourrait également faire du MVVM) [Fig.11].

Dans le code behind de ma MainWindow, je déclare dans le handler "OnInit" un Controller et un Listener que j'associe ensuite.

Dans la méthode "OnTrack()", j'invoque la méthode "Frame()" de mon Controller [Fig.12]. Je n'ai plus qu'à lancer mon application, puis à cliquer sur le bouton "Init" et enfin sur le bouton "Track" pour voir apparaître ma trame. Mais plutôt que d'utiliser un Leap Listener, il est plus pertinent d'être soi-même un Leap Listener. Cela nous permettra d'accéder aux méthodes protected du Listener, de gérer de manière plus souple l'initialisa-

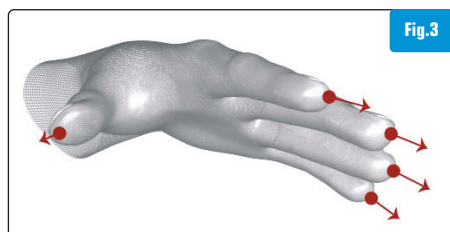


Fig.3

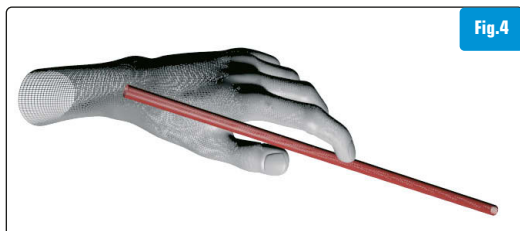


Fig.4

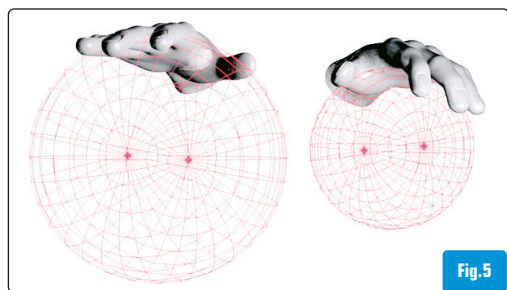


Fig.5

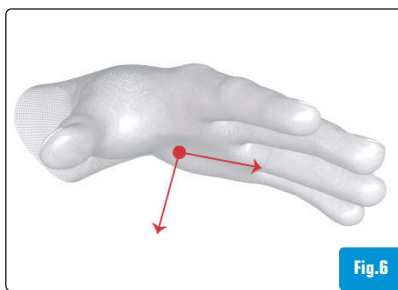
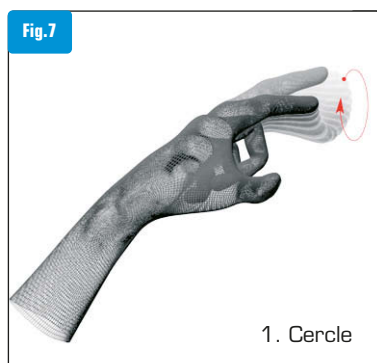
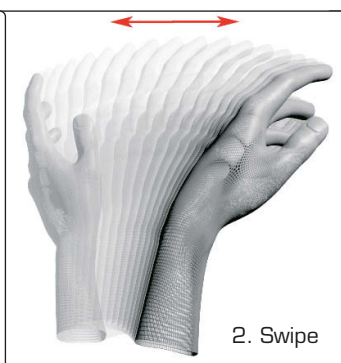


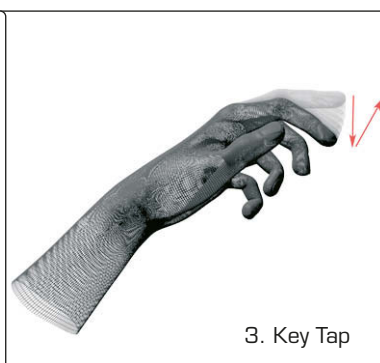
Fig.6



1. Cercle



2. Swipe



3. Key Tap



4. Screen Tap

tion et d'avoir automatiquement accès aux remontées de frames sans avoir à invoquer la méthode "OnFrame()" du Controller.

Parmi les surcharges possibles de méthode, nous trouvons :

- OnFrame
- OnInit()
- OnConnect()
- OnExit().

NB : il est conseillé d'activer les gestuelles au sein de la surcharge de "OnConnect()" [Fig.13]..

Mais la méthode qui nous intéresse le plus est "OnTrack()" : [Fig.14].

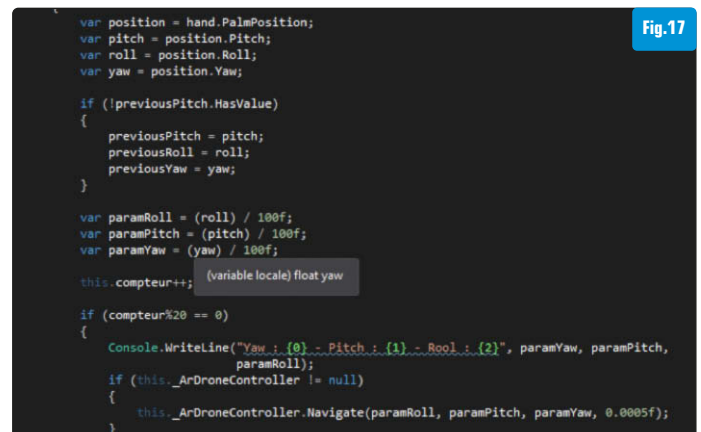
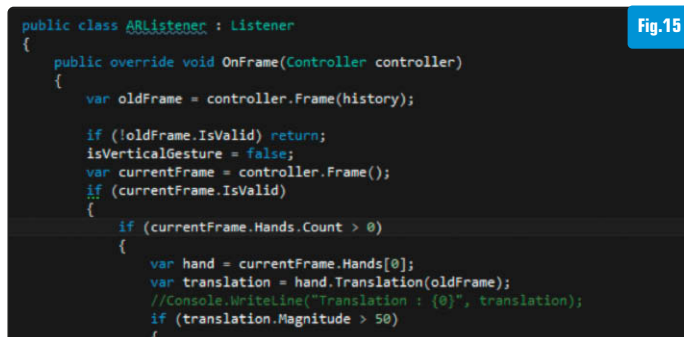
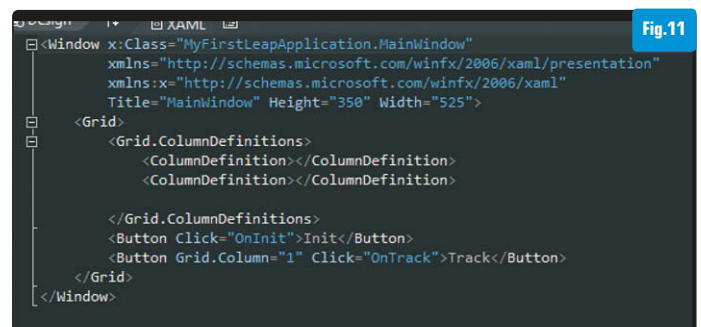
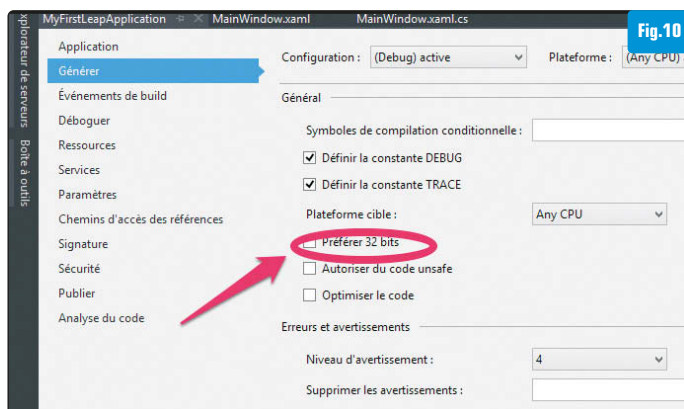
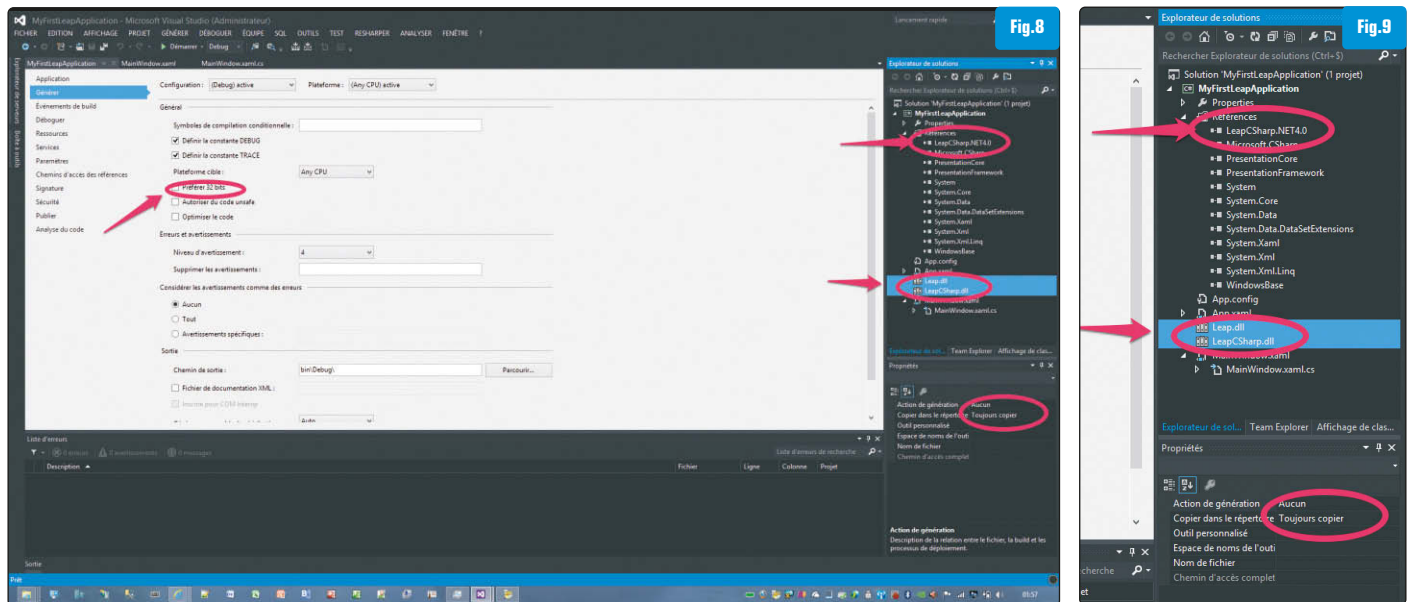
Par ailleurs, la méthode "Frame()" du Controller dispose d'une surcharge prenant un paramètre de type entier nous permettant d'accéder à l'historique. Ainsi, il est possible de récupérer la n-ième frame précédente. Attention : cela a un fort impact sur les performances, et au-delà de 30, le capteur nous retournera une frame vide [Fig.15].

Difficultés rencontrées

Je vous ai donné jusqu'à présent un aperçu des possibilités offertes par Leap Motion Controller. La grande difficulté consiste

désormais à analyser les données remontées. Cette tâche est loin d'être simple, puisque :

- Il y a toujours du "bruit" dans les trames : parfois, 4 doigts sont remontés au sein d'une trame, puis 3 doigts seulement dans la trame suivante.
 - Analyser les mouvements dans l'espace n'est pas simple, et 4 gestuelles prédéfinies ne suffisent pas à détecter toutes les intentions des utilisateurs. Il faudra donc s'appuyer sur les modèles "bas niveaux" remontés par le capteur.
- Ainsi, deux grands cas de figure se posent :



• Les applications en pseudo 2D

Elles permettent d'utiliser une surface non tactile et d'avoir des interactions similaires. A ce niveau, on peut citer les pointeurs de souris, les carrousels (etc.) qui se basent sur des objets Pointeurs (une simple projection des vecteurs sur une surface virtuelle) ou bien sur des gestuelles comme le "Screen Tap" ou le "Key Tap". Ces applications sont assez simples à développer.

```
public class TestListener : Listener
{
    public override void OnConnect(Controller controller)
    {
        controller.EnableGesture(Gesture.GestureType.TYPECIRCLE);
        controller.EnableGesture(Gesture.GestureType.TYPEKEYTAP);
        controller.EnableGesture(Gesture.GestureType.TYPESCREENENTAP);
        controller.EnableGesture(Gesture.GestureType.TYPESWIPE);
    }
}
```

Fig.13

```
using System;
using System.Windows;
using Leap;

namespace MyFirstLeapApplication
{
    /// <summary>
    /// Logique d'interaction pour MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        private Controller controller;
        private Listener listener;

        public MainWindow()
        {
            InitializeComponent();
        }

        private void OnInit(object sender, RoutedEventArgs e)
        {
            this.controller = new Controller();
            this.listener = new Listener();
            this.controller.AddListener(this.listener);
        }

        private void OnTrack(object sender, RoutedEventArgs e)
        {
            var frame = this.controller.Frame();
            Console.WriteLine("Id : {0} - Ts : {1} - Valid : {2} - Hands : {3} - Fingers : {4} - Pointables : {5} - Tools : {6}",
                frame.Id, frame.Timestamp, frame.IsValid, frame.Hands.Count, frame.Fingers.Count, frame.Pointables.Count, frame.Tools.Count);
        }
    }
}
```

Fig.12



Fig.16

```
using System;
using System.Windows;
using Leap;

namespace MyFirstLeapApplication
{
    /// <summary>
    /// Logique d'interaction pour MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        private Controller controller;
        private Listener listener;

        public MainWindow()
        {
            InitializeComponent();
        }

        private void OnInit(object sender, RoutedEventArgs e)
        {
            this.controller = new Controller();
            this.listener = new TestListener();
            this.controller.AddListener(this.listener);
        }

        private void OnTrack(object sender, RoutedEventArgs e)
        {
            var frame = this.controller.Frame();
            Console.WriteLine("Id : {0} - Ts : {1} - Valid : {2} - Hands : {3} - Fingers : {4} - Pointables : {5} - Tools : {6}",
                frame.Id, frame.Timestamp, frame.IsValid, frame.Hands.Count, frame.Fingers.Count, frame.Pointables.Count, frame.Tools.Count);
        }
    }
}
```

Fig.14

• Les applications en 3D

Elles sont beaucoup plus complexes à traiter. Il vous faudra si possible vous rapprocher d'un modèle existant. Ainsi, à l'instar de Google Earth, et ce, afin de piloter l'AR Drone (Parrot), je me suis basé sur la modélisation de la main. En effet, ce modèle comporte les paramètres d'inclinaison "Pitch", "Yaw" et "Roll", tout comme l'AR Drone. La difficulté principale réside dans le calibrage et la mise à l'échelle des valeurs reçues,

puisque le "Pitch" de ma main ne correspond pas au "Pitch" attendu par l'AR Drone ! [Fig.16 et 17].

Enfin, il est capital d'être conscient qu'un mouvement A et son contraire sont souvent indissociables. De ce fait, lorsque je fais un mouvement de la main de bas en haut pour faire décoller l'AR Drone, tôt ou tard, je devrai faire le mouvement inverse, et par conséquent, il est important de bien déterminer les mouvements à effectuer pour chaque action.

> Les usages possibles

Les débouchés du Leap Motion Controller sont nombreux, aussi bien dans des domaines ludiques, domotiques, scientifiques et même au sein d'entreprises. Seule notre imagination en limite les usages ! On peut imaginer utiliser le capteur pour nous permettre de naviguer rapidement

dans des flux de données, mais aussi s'en servir dans des machines-outils pour prévenir les accidents en déclenchant par exemple l'arrêt d'urgence si un doigt s'approche de trop près. Il y a également les applications évidentes comme les vitrines interactives des centres commerciaux ou d'agences immobilières. Un écran tactile

coûtant plus cher qu'un écran non tactile et accueillant malheureusement bien souvent des traces de doigts, la technologie Leap Motion sera certainement préférée !

Enfin, dans des métiers où une approche sensorielle est plus pertinente que l'utilisation d'un clavier et d'une souris (comme par exemple dans les domaines de la recherche) et là où des caméras ne sont pas forcément les bienvenues (respect de la vie privée, sécurité industrielle, etc.), des pistes restent également à explorer.



Patrick Legrand, Architecte

Logiciel .NET chez Palo IT
Passionné d'informatique, d'innovation et de nouvelles technologies, Patrick aime relever les défis et mettre son expertise d'Architecte, son enthousiasme et son conseil au service de clients exigeants.

SLightBox : un système « Ambilight » avec C# et Gadgeteer

Inventée en 2002 par Philips, la technologie Ambilight consiste à générer des effets lumineux autour de l'écran en fonction du contenu vidéo. Nous allons découvrir dans cet article comment développer notre propre système « Ambilight » en utilisant des modules « .NET Gadgeteer » et un peu de C#. Préparez Visual Studio, une perceuse et un fer à souder !

Note : Avertissement : tout montage électronique comporte des risques. Respectez les règles de sécurité et des guides de montage.

> Principe de fonctionnement

Il n'est pas très complexe de développer un système « Ambilight » performant. Dans cette première version, le prototype « SLightBox » est conçu pour fonctionner avec Windows. Cela vous permettra de profiter de ce système sur vos ordinateurs ou media-centers quel que soit le contenu vidéo (photo, vidéo, jeu ou toute autre application ou média) [Fig.1].



Fig.1

Slight

Pour réaliser un prototype rapide du concept, les modules .NET Gadgeteer basés sur le .NET Microframework offrent une plateforme de développement très facile à utiliser pour les développeurs. Comme cette première solution se base sur l'utilisation d'un PC, l'analyse du contenu vidéo sera relativement simple à entreprendre : capturer l'écran afin de

récupérer l'ensemble des pixels affichés ! Pour obtenir une expérience visuelle immersive nous avons besoin de dispositifs lumineux derrière l'écran longeant chacun des côtés. Pour que l'effet soit optimal, l'écran devra être placé à plus ou moins 30 cm d'un mur, si possible de couleur claire (idéalement blanc). De plus ces dispositifs lumineux, comme des LED par exemple, devront être contrôlables individuellement.

Une fois la capture de l'écran réalisée, le logiciel calculera la moyenne des couleurs pour chaque zone afin d'envoyer cette couleur à sa LED associée, située derrière l'écran (on établit qu'une zone est associée à une LED uniquement). Cette opération doit avoir lieu plus de 10 fois par seconde pour rester fluide et réactive par rapport au contenu vidéo.

> Le matériel

Les LED

La solution la plus adaptée au besoin sera d'utiliser des bandes de LED multicolores. Ces bandes de LED offrent la possibilité de les sectionner à certains endroits de la bande (en général par pas de 3 à 10 cm) permettant ainsi une plus grande souplesse d'utilisation. Il existe cependant deux types de bandes LED multicolores : les analogiques et les digitales.

Les bandes analogiques sont les plus répandues dans le commerce et les moins coûteuses car l'ensemble des LED qui les constitue (généralement des SMD5050 qui intègrent 3 LED : une rouge, une bleue et une verte) sont connectées en parallèle. On contrôle ainsi la couleur de toutes les LED en modulant le signal avec une sortie ana-

logique PWM (et quelques MOSFET pour amplifier le signal).

Pour permettre un contrôle individuel de la couleur de chaque LED, les bandes dites « digitales » intègrent des microcontrôleurs pour pouvoir adresser chaque LED individuellement.

On trouve dans le commerce une multitude de microcontrôleurs pour cet usage (HL1606, WS2801, WS2811, LPD8603, TM1812, UCS1903, etc...), les plus répandus étant le WS2801 et le LPD8806. Ce dernier permet de contrôler 6 canaux séparés de 7 bits. Il y a donc un microcontrôleur pour deux LED où chaque couleur est codée sur 21 bits, soit 2 097 152 combinaisons de couleur/brillance possibles. En général il y a 32 LED (donc 16 microcontrôleurs LPD8806) par mètre mais certains fabricants proposent jusqu'à 52 LED/mètre [Fig.2].

Pour l'installation, il suffit de découper votre bande de LED suivant les dimensions de votre écran. Chacune de vos bandes doit être connectée en série en les reliant ensemble. Il y a deux connecteurs pour l'alimentation (5V) et deux connecteurs pour l'interface SPI



Fig.2

Digital LED Strip LDP8806



Fig.3

Installation derrière l'écran

(données et horloge). Sur l'une des extrémités, vous soudez un connecteur JST 4 pôles pour la connexion à la « SLightBox » [Fig.3].

La « SLight Box »

Pour piloter le bus SPI depuis une application Windows, nous utiliserons des modules .NET Gadgeteer. Il faut donc une carte mère (mainboard) Gadgeteer comme la FEZ Spider, FEZ Hydra ou moins cher, la FEZ Cerberus (ainsi que son module d'alimentation comme le module « SP Client »). Pour la communication avec l'ordinateur nous utiliserons le module « USB-Serial » qui expose un port COM virtuel pour dialoguer depuis notre PC. Pour le contrôle des LED, les cartes

.NET Gadgeteer disposent d'une interface SPI sur les sockets de type « S » où la pin 7 est utilisée pour écrire les données (le MOSI) et la pin 9 pour l'horloge (SCK). Nous utiliserons le module Extender en soudant une petite barrette mâle (pas standard 2,54 mm) pour faciliter les connexions [Fig.4].

L'ensemble de ces modules sera installé dans une petite boîte en plastique avec une connectique pour :

- l'alimentation 5V,
- le connecteur USB du module « USB-Serial »,
- le connecteur JST vers les bandes de LED.

En façade :

- un interrupteur,

- une LED témoin rouge d'alimentation [Fig.5 et 6].

L'entrée 5V est coupée par l'interrupteur puis alimente le module SP (donc la mainboard Gadgeteer), la sortie JST (pour les LED) et la LED témoin en façade [Fig.7 et 8].

> Le logiciel Côté Gadgeteer

La « SLightBox Gadgeteer » est ni plus ni moins qu'une interface USB/SPI vers les bandes de LED. Son rôle est simple : attendre les données du PC via le port USB et les recopier sur le port SPI [Fig.9]. Au démarrage de la carte Gadgeteer, nous initialiserons la sortie SPI sur la socket du module Extender en spécifiant l'horloge à 16mhz, ainsi que le module USB en spécifiant le débit du port COM au maximum (115200 bauds) :

```
// Configure SPI Port
GTI.SPI.Configuration spiConfig = new GTI.SPI.Configuration(
    false, // chip select active state
    0,     // chip select setup time
    0,     // chip select hold time
    false, // clock idle state
    true,  // clock edge (true = rising)
    16000); // 16mhz

mLedStrip = new GTI.SPI(Socket.GetSocket(extendor.ExtenderSocket
Number, true, extendor, null), spiConfig, GTI.SPI.Sharing.
Exclusive, null);

// Configure USB Serial port
usbSerial.Configure(115200, Gadgeteer.Interfaces.Serial.Serial
Parity.None, Gadgeteer.Interfaces.Serial.SerialStopBits.One, 8);
// Open USB Serial port
usbSerial.SerialLine.Open();
// Ready !
Debug.Print("SLightBox Ready !");
```

Pour la suite, il suffit d'attendre les données du port USB pour les recopier sur l'interface SPI :

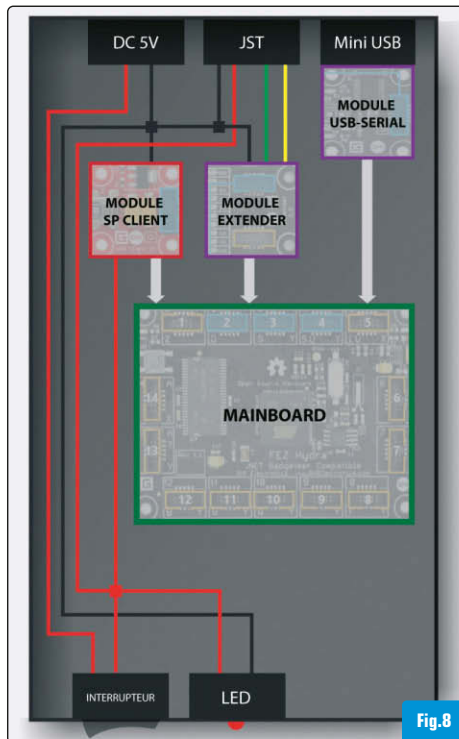
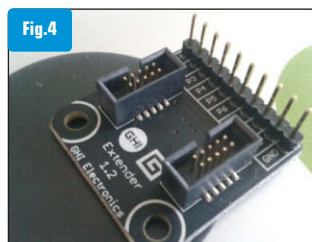


Schéma SlightBox



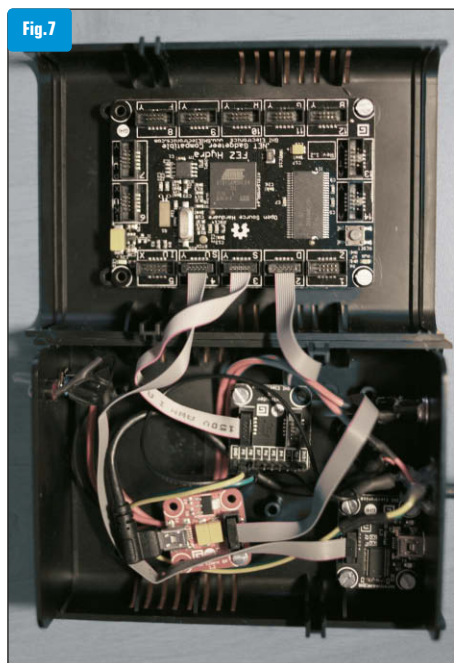
Module Extender avec barrette



SlightBox Recto



SlightBox Verso



SlightBox vue intérieure

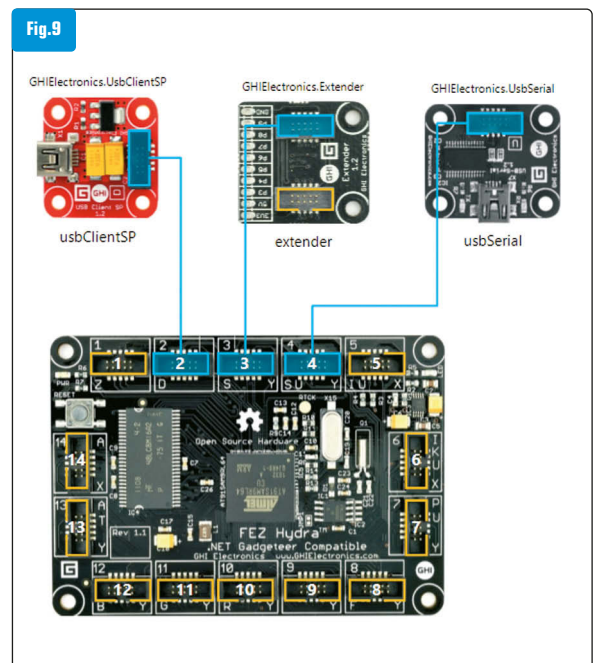


Schéma Gadgeteer


```
while (true)
{
    if (usbSerial.SerialLine.BytesToRead > 0)
    {
        int bytesRead;
        if ((bytesRead = usbSerial.SerialLine.Read(buffer, 0, usbSerial.
SerialLine.BytesToRead)) > 0)
        {
            byte[] dataToWrite = new byte[bytesRead];
            Array.Copy(buffer, dataToWrite, bytesRead);
            mLedStrip.Write(dataToWrite);
            Thread.Sleep(10);
        }
    }
    else
    {
        Thread.Sleep(1);
    }
}
```

La « SlightBox » est prête, il suffit de la connecter aux LED par le connecteur JST, au PC par un câble USB et à un transformateur 5V. En enclenchant l'interrupteur, la « SlightBox » sera détectée comme un « port COM virtuel » et attendra les données à envoyer aux LED ! Pour finir, n'oubliez pas de bien dimensionner la puissance de l'alimentation 5V : il faut prévoir 0,5 ampères pour les modules Gadgeteer, puis 2 ampères par mètre de LED (une LED consomme 60mA au maximum lorsqu'on affiche du blanc. Pour 32 LED/mètre, on peut théoriquement consommer près de 2A).

Côté Windows

Une fois la partie « hardware » terminée, nous pouvons nous attaquer au « cœur » de cette solution : le programme Windows : Slight. Ce programme doit être capable de capturer l'image de votre écran afin d'analyser les couleurs dominantes et les envoyer sur le port COM virtuel, elles seront ensuite transmises par SPI aux LED via le programme Gadgeteer de la « SlightBox ». Pour la capture de l'écran, nous nous servirons de DirectX pour des questions de performance à l'aide de SlimDX, un framework .NET open source pour DirectX. Par exemple pour capturer l'écran principal :

```
// Initialization
this.dxDevice = new Device(
    new Direct3D(), 0, DeviceType.Hardware, IntPtr.Zero, Create
Flags.SoftwareVertexProcessing,
    new PresentParameters() { Windowed = true, SwapEffect = Swap
Effect.Discard });

// To capture, get the screen surface
Surface surface = Surface.CreateOffscreenPlain(
    this.dxDevice,
    Screen.PrimaryScreen.Bounds.Width,
    Screen.PrimaryScreen.Bounds.Height,
    Format.A8R8G8B8, Pool.Scratch);
// Capture surface screen buffer data
Surface screenSurface = this.dxDevice.GetFrontBufferData(0, surface);
DataRectangle screenRectangle = screenSurface.LockRectangle(Lock
```

```
Flags.None);
DataStream screenDataStream = screenRectangle.Data;

// - Do work -

// Dispose screen surface
screenSurface.UnlockRectangle();
screenSurface.Dispose();
```

Le « DataStream » est un flux où chaque pixel est représenté sur 4 bits (B, G, R, A). Au démarrage, le programme Slight calcule les coordonnées de chaque zone en fonction de la disposition des bandes de LED et de la résolution de l'écran. On sélectionne ensuite les coordonnées d'un pixel sur vingt dans cette zone comme échantillon pour calculer la couleur moyenne de la zone. Comme chaque zone est associée à une LED uniquement, nous ajouterons des Panels dans notre fenêtre pour visualiser les couleurs calculées pour chaque LED. A la fin de l'initialisation nous disposons donc de la position dans le DataStream de chaque « pixel échantillon » pour chaque zone. A l'aide d'un « Timer » déclenché toutes les 50ms, nous capturerons le DataStream de l'écran afin de calculer la couleur moyenne de chaque zone par la méthode *GetColorAverage* :

```
private Color GetColorAverage(DataStream screenDataStream, Collection
<long> positions)
{
    byte[] buffer = new byte[BITS_PER_PIXEL];
    int r = 0, g = 0, b = 0, count = 0;
    // For each pixels
    foreach (long pos in positions)
    {
        screenDataStream.Position = pos;
        screenDataStream.Read(buffer, 0, BITS_PER_PIXEL);
        r += buffer[2];
        g += buffer[1];
        b += buffer[0];
        count++;
    }
    // Return the average color for the list of pixels
    return Color.FromArgb(r / count, g / count, b / count);
}
```

Pour adresser la couleur à la bonne LED nous devons écrire la liste des couleurs des zones les unes derrière les autres.

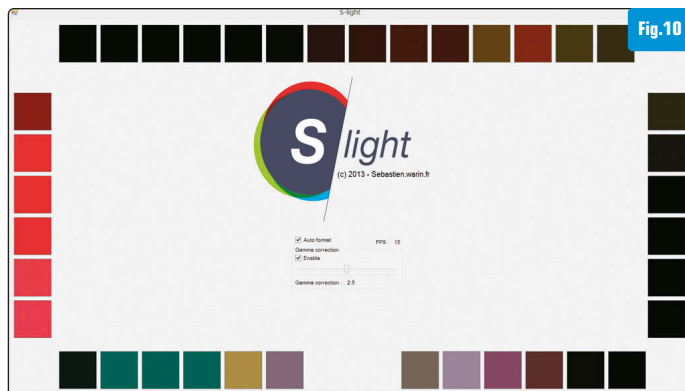
Pour rappel, avec le LPD8806 les couleurs sont définies en GRB (Vert, Rouge, Bleu) sur 7 bits. On pourrait donc diviser la valeur capturée du R, G et B par deux. Néanmoins pour rendre les couleurs plus riches, nous appliquerons une petite correction gamma : $\{127 * [La_Couleur_sur_8_bits / 255] ^ Correction_Gamma\} + 0.5$ (où « Correction_Gamma » est par défaut à 2,5 mais peut varier entre 1 et 5 en fonction de vos préférences). Ainsi pour chaque capture :

```
byte[] colorData = new byte[ledBytesCount];
// For each zones (= for each leds)
for (int i = 0; i < numberOfLeds; i++)
{
    Color pixelColor = Color.Black;
    if (pixelsList.Count > i)
```

```
{
    // Get color average
    pixelColor = this.GetColorAverage(screenDataStream, pixels
List[i]);
    // Update UI
    this.updatePanelColorActions[i](pixelColor);
}
// Build colorData
int index = i * BITS_PER_LED;
if (this.GammaCorrectionEnabled)
{
    colorData[index + 0] = (byte)(LPD8806_MASK | (byte)(Math.
Pow((double)pixelColor.G / 255.0, this.gammaCorrectionValue) *
127.0 + 0.5));
    colorData[index + 1] = (byte)(LPD8806_MASK | (byte)(Math.
Pow((double)pixelColor.R / 255.0, this.gammaCorrectionValue) *
127.0 + 0.5));
    colorData[index + 2] = (byte)(LPD8806_MASK | (byte)(Math.
Pow((double)pixelColor.B / 255.0, this.gammaCorrectionValue) *
127.0 + 0.5));
}
else
{
    colorData[index + 0] = (byte)(LPD8806_MASK | (pixelColor.G / 2));
    colorData[index + 1] = (byte)(LPD8806_MASK | (pixelColor.R / 2));
    colorData[index + 2] = (byte)(LPD8806_MASK | (pixelColor.B / 2));
}
}
// Send datas to the SerialPort
byte[] datas = new byte[ledBytesCount + LPD8806_LATCH_BITS];
Array.Copy(colorData, 0, datas, LPD8806_LATCH_BITS, ledBytesCount);
this.port.Write(datas, 0, datas.Length);

// Avec ...
SerialPort port = new SerialPort("COMx", 115200, Parity.None,
8, StopBits.One);
const int BITS_PER_PIXEL = 4;
const int BITS_PER_LED = 3;
const int LPD8806_LATCH_BITS = 21;
const byte LPD8806_MASK = 0x80;
int ledBytesCount = numberOfLeds * BITS_PER_LED;
```

Le tableau d'actions « updatePanelColorActions » permet de mettre à jour la couleur du Panel associée dans la fenêtre : [Fig.10]. Voilà l'application terminée, il ne reste plus qu'à tester son bon fonctionne-



Programme Slight

ment et ses performances grâce à des vidéos de test spécialement conçues pour Ambilight que vous trouverez sur YouTube [Fig.11].

Nous pouvons encore enrichir son fonctionnement en ajoutant la capacité de s'adapter dynamiquement au format de la vidéo ! En effet, lorsque vous regardez une vidéo en 4:3 ou 21:9 sur un écran 16:9 vous n'échapperez pas aux bandes noires horizontales ou verticales. Les zones calculées par Slight risqueraient de ne « rien voir » d'autre que du noir sur certains bords. Pour combler cette lacune, il faut recalculer les coordonnées des zones en tenant compte de la taille des bandes noires.

Pour cela, au démarrage de l'application, on calcule les coordonnées des pixels d'une ligne fictive partant de chaque bord de l'écran vers le centre. Plusieurs fois par seconde, on remonte cette ligne tant que le pixel est noir ! Dès lors qu'il ne l'est plus, cela signifie alors que nous sommes au début du contenu vidéo ! Pour éviter un comportement instable, chaque côté utilise trois lignes différentes (au ?, ? et ?) où chaque pixel de couleur trouvé doit être à la même distance du bord sur les deux autres lignes.

De plus, le redimensionnement n'a lieu que si les dix dernières mesures ont donné le même résultat. Ainsi peu importe le contenu vidéo, Slight s'adaptera en toute circonstance !

> Conclusion

Comme vous l'avez découvert dans cet article, la réalisation d'un système Ambilight pour PC est relativement simple avec quelques modules Gadgeteer, un Visual Studio et quelques composants, sans oublier les bandes de LED digitales.

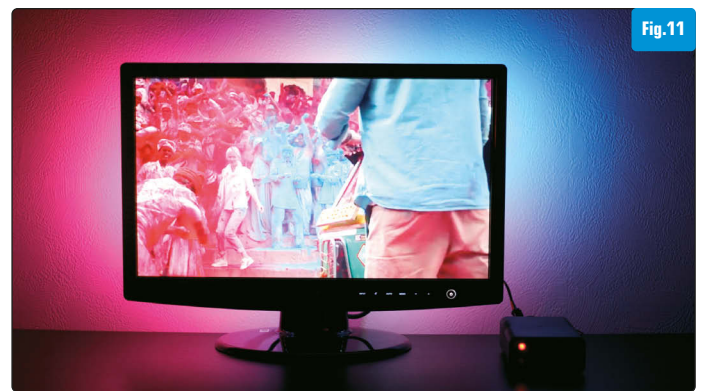
Les modules Gadgeteer sont très pratiques pour confectionner des prototypes de tous genres, bien que dans ce projet ils soient largement sous-utilisés. Pour rendre le système moins cher, la « Slight-Box v2 » est basée sur une puce MCP2210 USB/SPI dédiée et remplace ainsi les 4 modules Gadgeteer. L'application Windows reste la même dans sa logique mais communique avec la SlightBox non pas via un port COM virtuel du module Gadgeteer mais par le driver HID du fabricant de cette puce.

Pour aller encore plus loin, la « SlightBox v3 » sort du cadre du PC : branchez-la directement sur vos périphériques HDMI (box internet, câble/satellite, DVD/Blu-ray, consoles de jeux, etc..) et profitez des avantages de Slight sur votre TV sans aucun PC ! Toutes les informations sur les SlightBox et guides de montage complets sur www.slight-box.net !

Sébastien Warin

Software Architect @ ETO

<http://sebastien.warin.fr>



Demo Slight

PICAXE : débiter sans peur dans le monde de l'embarqué

Débiter dans le monde de l'embarqué et de l'électronique fait (souvent) peur. Peur d'avoir des processeurs trop compliqués, trop chers, ne pas réussir à les monter, peur de ne pas savoir faire de montages électroniques, peur d'apprendre un nouveau langage. Avec juste un peu de courage, PICAXE (<http://www.picaxe.com>) est vraiment fait pour vous. C'est aussi une excellente plateforme pour des enfants, à partir du collège. De simples notions d'électronique et d'informatique suffisent largement.

D'abord le prix : entre 2€ et 6€ pour les plus chers. Avec le câble USB pour pouvoir télécharger les programmes et l'adaptateur, il faut compter 20€ au total. Et encore, le montage est très simple à réaliser et documenter sur le site. L'idéal est d'ajouter une platine d'essai enfichable pour prototypage rapide, comptez environ 5 à 10€ de plus. Et disons qu'il vous faudra quelques résistances, diodes et transistors, pour 5 à 10€ au maximum, vous arriverez à vous en sortir. Pour les moteurs et autres accessoires plus avancés, un peu de récup fera largement l'affaire pour commencer. Ensuite le langage : le bon vieux BASIC (il résiste, il résiste) ou alors du développement visuel. Franchement, difficile de faire plus simple, parole de geek ! Un peu plus rassuré ? Bon, passons à la pratique.

> Qu'y a-t-il dans la bête ?

Ce sont de petits processeurs cadencés de 32 à 64 MHz, qui possèdent une mémoire de stockage de 2048 à 4096 octets et d'exécution de 128 à 1280 octets. Oui, on parle bien d'octets et pas de kilo ni de méga. Donc là, vous l'aurez compris, il n'y a bien sûr pas d'OS qui tourne en tant que tel. Juste un firmware léger qui contient l'environnement d'exécution.

Le plus intéressant c'est que ces petits processeurs sont très bien équipés en entrées et sorties de tout type. Il y a bien sûr des entrées et sorties numériques mais toute une panoplie bien plus avancée : SPI, i2c, Série, Touch, PWM, entrées et sorties analogiques. Jugez plutôt sur le plus petit des processeurs [Fig.1].

> Un peu d'électronique

L'électronique, c'est souvent la partie qui effraie le plus. Lorsque je m'y suis remis il y a

quelques années, je ne savais pas trop par quel bout commencer. J'étais motivé car je voulais pouvoir accéder à mon arrosage automatique à distance depuis mon téléphone ou n'importe quel PC. Alors j'ai commencé par les bases : $U = RI$. Avec ça, on peut tout refaire. Et pour faire plus efficace et plus simple, sans avoir vraiment à trop vouloir comprendre, vous pouvez vous lancer avec l'excellente documentation de PICAXE (http://www.picaxe.com/docs/picaxe_manual3.pdf). Les principaux montages sont faits avec les explications qui vont avec, sans formule, avec l'exemple de code en plus. Ici un moteur de jouet très classique [Fig.2].

Le guide vous emmène jusqu'aux moteurs pas à pas et les afficheurs LCD. Le tout en passant par les capteurs. Les éléments électroniques sont indiqués et il suffit donc de les commander.

Pour l'alimentation, récupérez de bons vieux transformateurs, ils feront l'affaire. Attention, le processeur s'alimente entre 4,5 et 5V. Idem, tout est très bien expliqué dans les 3 chapitres des manuels. De quoi donner plein de courage !

> Le code

Comme indiqué, pas de panique, le code est simple, il s'agit de BASIC pour les instructions. Ensuite, il y a un nommage intuitif pour l'accès aux fonctions de lecture des ports

et des protocoles avancés. Par exemple pour lire le port C.2 en analogique, voici le code :

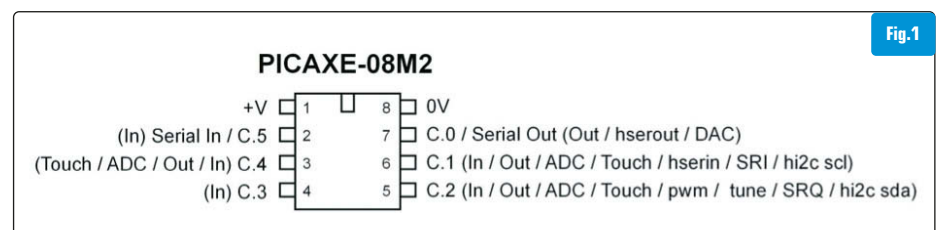
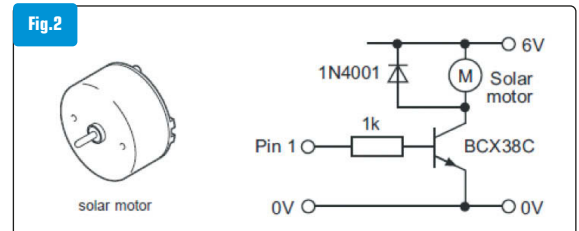
```
readadc C.2,b1
```

La valeur est mise dans le registre b1. Le processeur possède des registres d'un octet pour y stocker les valeurs de lecture ou d'écriture. Il est possible d'avoir des bits, des octets et des mots. Le nombre de ces registres est limité. En plus il y a des variables système permettant d'avoir un timer, de lire la dernière valeur du port série ou protocole i2c. Et pour de la programmation avancée, il y a même un pointeur sur la mémoire.

Voici un autre exemple un peu plus avancé, il s'agit d'un bouton :

```
button C.3,1,200,100,b2,1,pushed
```

Le bouton possède plusieurs paramètres dont le dernier qui est le label où le code sera dirigé quand le bouton sera dans l'état 1 (avant dernière variable).



Et puis, comme il y a un port série intégré, on peut aussi envoyer des commandes, comme :

```
SerTxd ("Bonjour Programmez ! »)
```

Il y a bien plus de commandes spécifiques, par exemple piloter un servomoteur, un afficheur LCD, se mettre en mode hibernation, etc. A noter que la documentation est vraiment très bien faite, comptez 270 pages en anglais très simple pour l'ensemble des commandes, les éléments du langage pour l'ensemble de la famille de processeur.

A noter également qu'il est possible d'avoir plusieurs « threads » qui s'exécutent en parallèle. Cela fonctionne correctement tant qu'il n'y a pas une charge bloquante ou longue sur une des tâches. Ce n'est pas exactement une vraie gestion de thread, donc il ne faut pas compter sur une véritable parallélisation. En revanche, c'est très pratique et cela permet d'avoir des boucles d'attente de valeurs de capteurs pendant que des actions se réalisent en parallèle comme le pilotage d'un moteur.

Et si vous ne voulez pas faire de code, il existe un environnement de programmation graphique offrant les mêmes possibilités. Ici un exemple simple mettant au niveau haut le pin O, attendant 1 seconde, puis mettant le niveau à bas et attendant 1 seconde et rebouclant ainsi de suite.

Cela permet par exemple de faire clignoter une LED branchée sur le pin O. Il correspond au pin C.O du processeur [Fig.3].

> Un exemple concret

Nous allons réaliser un montage simple qui va émettre un son quand la lumière devient importante. Pour cela, nous allons avoir besoin d'un capteur résistif de lumière. Sa résistance diminue quand la luminosité aug-

mente. Il faudra aussi une résistance de 10KΩ et un haut parleur. J'ai récupéré un vieux haut parleur de PC. Il fera parfaitement l'affaire [Fig.4 et 5].

Les schémas sont ceux de la documentation. Nous nous passerons du condensateur de 10µF qui n'a pas d'intérêt fonctionnel.

Le pin analogique que nous utiliserons sera le C.4 et la sortie pour le son, le pin C.2. Le code est vraiment très simple : [Fig.6].

Pratique également, l'environnement de développement qui propose un émulateur et permet de vérifier le comportement du système. Il est possible de modifier les valeurs d'entrées analogiques, ici j'ai changé la valeur du pin C.4 à 205, j'ai ajouté une ligne de sortie sur le port série pour envoyer cette chaîne. L'émulateur propose également la vue de la mémoire. Parfait quand on veut jouer avec les pointeurs [Fig.7].

Il est également possible d'utiliser des fonctions de PWM (Pulse Width Modulation) qui permettent de moduler à une fréquence

donnée le signal envoyé. Cela a pour effet de faire varier la tension moyenne. A une fréquence donnée, quand vous envoyez un signal modulé à 50%, vous obtiendrez 50% de la tension nominale, 25% à 25%, etc. Cela permet donc d'avoir une pseudo sortie analogique avec une sortie numérique. Et cela ne fonctionne qu'avec certains éléments tels les ventilateurs, moteurs et autres périphériques de ce type. C'est également utilisé pour piloter les servomoteurs. Les commandes de pilotage de servo sont également présentes dans les PICAXE. Son utilisation est simplifiée et il y a un assistant qui permet de créer le code nécessaire [Fig.8].

Mon fils de 9 ans s'amuse à faire varier la vitesse de rotation d'un ventilateur en fonction de la lumière. Et il utilise la sortie PWM pour cela avec le montage type du moteur. L'initialisation se fait simplement en paramétrant la sortie PWM et en demandant une sortie modulée.

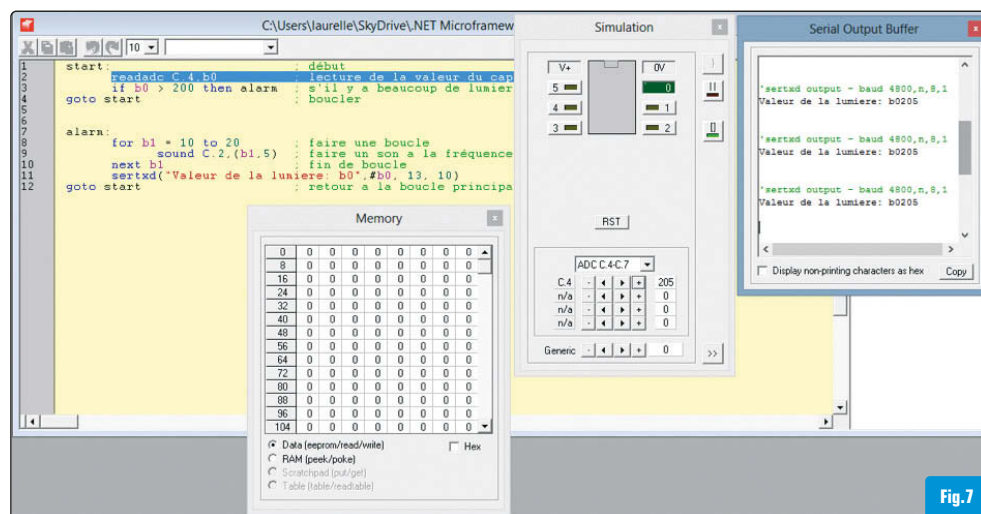


Fig.7

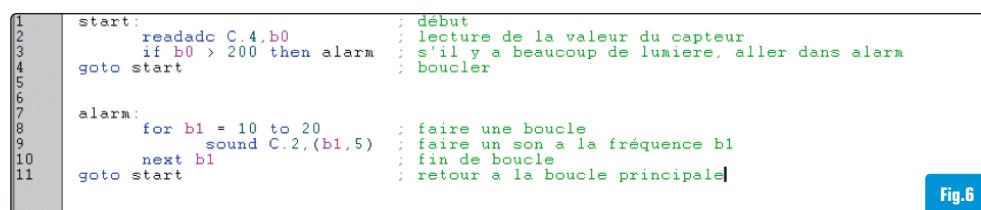


Fig.6

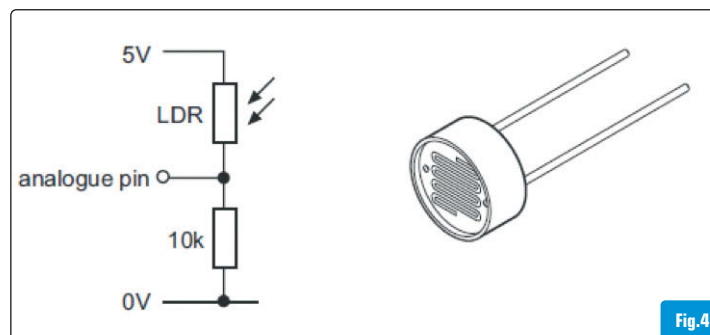


Fig.4

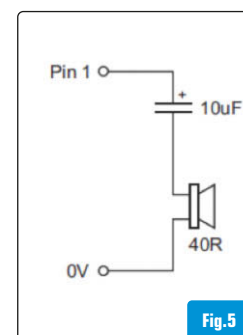


Fig.5

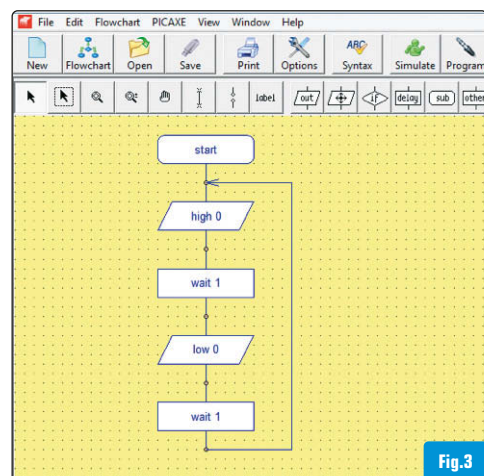


Fig.3

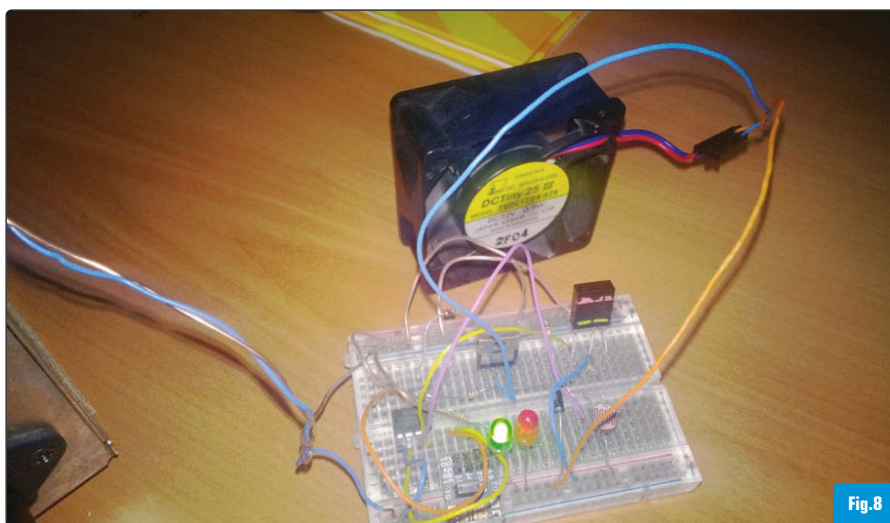


Fig.8

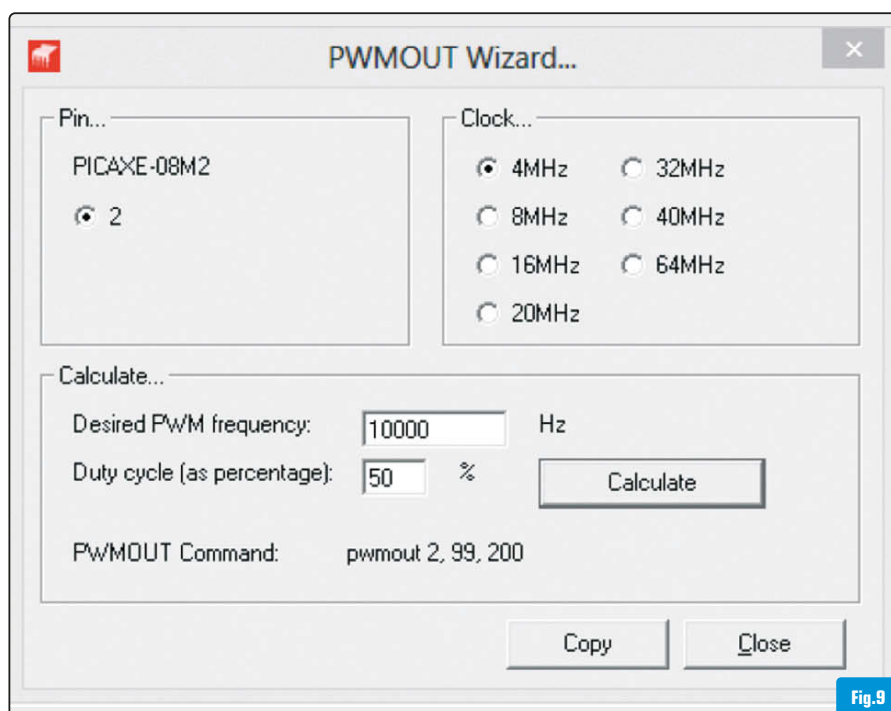


Fig.9

```
init:
pwmout 2, 50,500
pwmduy 2,0
```

La commande pwmduty est celle qui permet de faire varier la période entre 0 et 1024. Et donc entre 0% de la fréquence nominale et 100% avec un pas de 1/1024.

> Pour aller plus loin

Ca y est, je vous sens rassuré pour vous lancer. Ce petit processeur est vraiment parfait pour débiter et faire des robots simples ou des objets communicants. En tout cas, il fait le bonheur de mes enfants qui s'amusent à réaliser ce type de montage simple et faire le code qui va avec. Ils

réalisent des montages allumant, éteignant des LED, faisant fonctionner des moteurs avec des potentiomètres, des interrupteurs ou des capteurs de température ou de lumière en entrée.

De mon côté, je les interface avec des modules Bluetooth ou Xbee pour apporter l'aspect communicant et sans fil. Ils sont parfaits également pour ajouter un peu d'intelligence à des capteurs. Et bien évidemment, le champ des possibilités est très large, il y a de nombreux projets partagés sur le site.



Laurent Ellerbach
Audience Marketing
Microsoft EMEA
DPE - Central and Eastern
Europe

Trucs et astuces pour commencer en électronique

Pour commencer, je recommande vivement l'achat d'une platine d'essai enfichable pour prototypage. Même si le coût initial pour une petite platine est de 5 à 10€ et il est essentiel dans la phase de réalisation. Vous en trouverez assez facilement sur les sites d'électronique grand public.

Pour le câblage de la platine d'essai, récupérez de vieux câbles réseaux ou téléphone, dénudez le plastique autour et récupérez les fils torsadés à l'intérieur. Ils feront des straps parfaits.

Les packs de résistances, condensateurs chimiques et céramiques, LED sont parfaits pour faire le stock de départ. Idem, il est facile d'en trouver pour 5 à 10€ par type de composant. L'avantage c'est que vous serez à peu près sûr d'avoir ce qu'il vous faut pour réaliser tout type de montage. Et l'inconvénient, c'est que certains composants ne vous serviront jamais. L'alternative est de ne commander que ce dont vous avez besoin.

La plupart des composants ne se vendent pas seuls, en général par 5 à 10 pièces. C'est le cas pour les composants discrets mais aussi les actifs tels les circuits intégrés.

L'achat d'un fer à souder est également à faire quand il s'agira de passer de la phase de prototypage à un vrai montage. Pour les plaques, prenez des platines d'essais à pastilles, cela permet de souder et de relier facilement les composants. Utilisez les fils des paires torsadées pour relier les points éloignés entre eux.

Pensez à récupérer ! Il y a les moteurs, certains capteurs, des afficheurs LCD, les claviers, les connecteurs et plus ! Pour l'alimentation, récupérez les vieux chargeurs, l'alimentation des PICAXE se fait en 5V. Mais si vous utilisez des moteurs, ils ont parfois une tension d'alimentation plus importante. Utilisez un régulateur de tension pour l'alimentation du PICAXE et la tension nominale du chargeur pour le reste.

Utilisez PowerPoint pour faire vos schémas électroniques. Cela va vraiment très vite. Idem pour les schémas de montage. Numérisez votre platine d'essai et posez des carrés, rectangles, ronds reliez les points de soudure entre eux pour vous faire votre schéma de soudure.

Développeurs, n'ayez pas peur des **navigateurs** !



Tout devait être plus simple, plus facile. HTML 5 devait être le Sésame pour simplifier le développement et tout rendre multiplateforme, sans aucun problème. Mais en informatique, nous savons qu'entre les belles paroles et la réalité, il y a parfois un océan. Et le navigateur ressemble de plus en plus à un océan où il est difficile de trouver le cap et d'y naviguer sans trop de tempêtes.

Aujourd'hui, on parle de Responsive Design, de navigateur mobile, de desktop, de HTML 5 / CSS, etc. Les uns maudissent les préfixes webkit, mais oublient que d'autres ont produit un navigateur qui a longtemps pourri la vie des développeurs : IE 6 !

Concevoir des sites et applications web demande des efforts importants de tests et d'ajustements. Car il faut tester sur plusieurs navigateurs (souvent sous différentes versions), sur différents systèmes. Cela est valable sur desktop et mobile. Difficile de ne pas tester et

éprouver son développement sur Chrome, Firefox, IE, Safari. Et malheureusement, chaque navigateur propose son architecture, ses moteurs de rendu et d'exécution. Avec parfois des différences d'implémentations.

Plusieurs problèmes surviennent quand le développeur web veut bien faire son travail :

- comment optimiser sans pour autant défavoriser tel ou tel navigateur
- que proposent les navigateurs pour m'aider ?
- faut-il supporter les derniers navigateurs ou coller à la réalité « marché » ?
- la problématique du Responsive Design
- suivre les bugtracks des navigateurs

Cette liste n'est pas exhaustive, loin de là. Le sujet est vaste mais c'est la réalité aujourd'hui du web. Et elle est tout sauf simple.

Ce mois-ci, nous aurons un focus sur IE10 mais dans les prochains numéros, nous aborderons l'ensemble des navigateurs.

François Tonic

Comprendre l'architecture des navigateurs

Chaque navigateur Internet est différent, techniquement d'une part pour ses priorités lors de son développement et d'autre part pour sa manière d'implémenter les standards du Web, définis par le W3C (dernier exemple en date : HTML5) et l'ECMA. Pour afficher le contenu des sites Internet, les navigateurs utilisent un moteur de rendu qui peut être Trident pour Internet Explorer, WebKit pour Safari, Blink pour Google Chrome, etc.

De manière générale

Tous les navigateurs se connectent à Internet et proposent des affichages en fonction de leurs implémentations des standards. Cela dit, on va regarder de plus près ce que nous réservent les navigateurs d'aujourd'hui :

1. Télécharger les données sur le serveur distant est sans aucun doute la première opération qu'effectue le navigateur client, après avoir reçu la requête de l'utilisateur à travers l'URL.
2. Une couche appelée « réseau » est en charge de toutes les communications entre le navigateur et Internet, en passant par la gestion du cache et le contenu du logiciel.
3. Le HTML est analysé pour lancer le téléchargement des médias (images, sons, feuilles de style, ...) à travers le composant du réseau. Ce travail lancé, le code HTML lui-même et les documents associés comme du XML, CSV, SVG, etc. sont mis en mémoire.
4. C'est au composant CSS de prendre la main : une analyse des styles trouvés est faite et une représentation est stockée en mémoire.
5. Les documents HTML contiennent très souvent des métadonnées pour les décrire. Une composante est dédiée à les stocker et à les rendre accessibles.
6. Certains navigateurs n'ont pas intégré nativement un interpréteur JavaScript au sein de leur navigateur, c'est pourquoi les scripts passent par une couche, appelée Marshalling, pour communiquer entre le navigateur et le moteur de script.
7. Le code JavaScript va ensuite interagir à travers les API du DOM.
8. Ensuite, le document construit et mis en mémoire au préalable, le navigateur doit appliquer les styles CSS eux aussi en

mémoire, bloc par bloc. Tous les styles appliqués, la mise en page est effectuée.

9. Le rendu final du document à afficher fait appel à la dernière composante du navigateur. Ce processus peut être traité par le processeur de la machine (CPU) et/ou par le processeur graphique (GPU).

Petit focus sur IE

Dans le cas d'Internet Explorer et de beaucoup d'autres navigateurs, on parle d'architecture. Du moins dans sa conception, elle est indispensable pour définir sa structure. Celle d'Internet Explorer est bien spécifique et repose sur le COM (Component Object Model). Ainsi cela donne une hiérarchie intégrée de la façon suivante :

- En premier lieu, notre navigateur qui, rappelez-le, est un programme qui se lance, nous donnant le IExplore.exe. C'est cette partie qui met en forme tout ce que les autres composants d'Internet Explorer lui envoient.
- Ensuite, au coude à coude, on retrouve deux composants particulièrement importants : le « Browsui.dll » qui fournit l'interface (le chrome) qui regroupe la barre d'adresse, les menus, la barre d'état, etc. Mais également « Shdocvw.dll », derrière ce nom barbare se cache une DLL capable de s'occuper de l'interface de contrôle ActiveX. Cela permet généralement, lorsque vous développez vous-même une application avec un contrôle Web Browser, d'avoir toutes les caractéristiques d'Internet Explorer mais sans le Chrome.
- À ce niveau, on dévoile le moteur de rendu de notre navigateur. Internet Explorer utilise « Trident », mais son véritable nom est le suivant : « Mshtml.dll ». Il s'agit du cœur du navigateur, il va mettre en forme le CSS et le HTML mais ce n'est pas tout. En

effet, d'autres moteurs de script sont également utilisés pour interpréter du JavaScript, du VB Script, les contrôles ActiveX, les données XML, etc.

- Arrivé ici, que reste-t-il ? La réponse est la sécurité. C'est avec « URLMon.dll » que la protection de l'utilisateur est mise en place lors du téléchargement de données. Une fonction simple et pourtant indispensable. Cette DLL offre par ailleurs des gestions MIME qui permettent au navigateur de gérer les actions sur certains des fichiers reçus.
- Enfin, « Wininet.dll » agit en tant que gestionnaire de protocole Internet Windows : HTTP et FTP et gère la gestion du cache. C'est sur cette base que se développent les mises à jour du navigateur. On notera le terme de compatibilité HTML5 et CSS3, l'ajout de nouveaux moteurs... Ceux-ci sont impactés sur Trident. D'autres extensions pourront être ajoutées au navigateur : des raccourcis de menus, des barres d'outils personnalisés ou de nouveaux contrôles ActiveX. Ce sont ces extensions qui sont une véritable force pour le navigateur qui devient alors personnalisable et unique pour chaque utilisateur. Ceci est possible à l'aide de la classe d'extensibilité qui se lie à l'interface utilisateur en tant qu'add-on pour altérer l'expérience de navigation.

Andrés Talavera

Développeur passionné par les technologies Microsoft
Microsoft Student Partner
cresus.net

Antony Canut

Développeur C# .NET / Windows Phone
Microsoft Student Partner

Moteurs : chacun chez soi

Une des problématiques du développement web est que chaque navigateur propose sa propre architecture, ses propres moteurs et des implémentations des protocoles et technologies pas toujours identiques, sans compter parfois, les limitations liées à chaque navigateur.

Il suffit de regarder les tableaux ci-dessous pour comprendre l'ampleur du problème.

Navigateurs	éditeur	moteurs	systèmes desktop
Chrome	Google	Blink (fork WebKit) V8	Linux, OS X, Windows
Firefox	Mozilla	gecko spidermonkey	Linux, OS X, Windows
Internet Explorer	Microsoft	Trident Chackra	Windows
Safari	Apple	WebKit	OS X

Navigateurs mobiles	systèmes mobiles
Chrome	Android, iOS
Firefox	Firefox OS, Android
Internet Explorer	Windows Phone
Safari	iOS

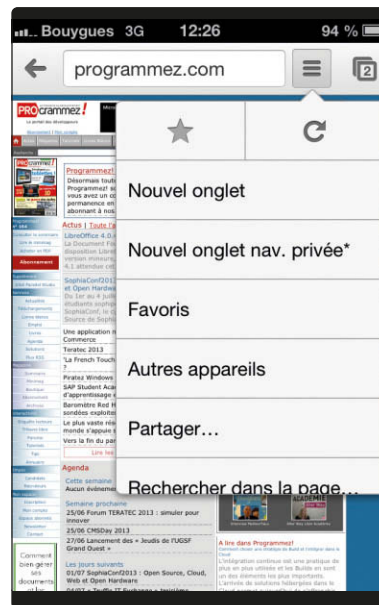
> Une cadence infernale

Il y a quelques années, nous étions habitués à la lente, très lente, évolution des navigateurs. IE était mis à jour tous les 2/3 ans. Souvenez-vous de l'attente de IE7 (5 ans, une éternité dans l'informatique). Même Firefox avait un cycle de développement assez long, tout comme Safari, côté Apple. Mais depuis 2 ans, les cycles de sortie des nouvelles versions se sont considérablement accélérés.

Le responsable en est Chrome / Chromium. Google a décidé dès

l'été 2010 de sortir une nouvelle version du navigateur tous les 6 semaines. Pour pouvoir implémenter les nouveautés, corriger les bugs, faire évoluer les moteurs internes. Deux navigateurs étaient visés par Chrome : IE et Firefox. Ce dernier a été le plus touché par la déferlante Chrome auprès des utilisateurs. En 2011, Mozilla avait tout d'abord prévu un cycle de 16/18 semaines pour chaque nouvelle version. Mais rapidement, le cycle s'est calqué sur celui de Chrome : 6 semaines.

version de Firefox	Date de disponibilité prévue
21	14 mai
22	25 juin
23	6 août
24	17 septembre



Apple n'a pas de cycle de développement officiel pour les développeurs et utilisateurs. Les membres des programmes développeurs peuvent accéder aux versions les plus récentes du navigateur (pré-version). La v6.0.5 est disponible avec OS X 10.8.4. Apple met le plus souvent à jour son navigateur lors des mises à jour systèmes. Microsoft inclura avec Windows 8.1, IE 11. L'éditeur va lui aussi raccourcir son cycle de développement, même si pour le moment, il n'est pas question d'imiter Google et Mozilla.

Ces cycles courts posent divers problèmes pour les développeurs :

- tester à chaque nouvelle version les sites et applications
- déployer le navigateur
- modifier si besoin le code et le redéployer

Parmi les 4 principaux navigateurs, trois proposent des « nightly build » : Chrome (Chromium), Firefox et WebKit (moteur de Safari). Bien entendu, ces versions, instables et en développement, ne doivent jamais être prises comme plateformes de développement ou de tests. Ces versions permettent de juger des prochaines nouveautés, de certaines fonctions ou de corrections de bug. Il est d'ailleurs dommage qu'Internet Explorer ne propose aucune version de ce genre ou que Microsoft n'accélère pas le rythme.

FAQ Blink Google I/O : <http://www.chromium.org/blink/developer-faq>

François Tonic

Google «forke» WebCore : Blink

Oui, Google quitte le socle WebKit. Non, WebKit n'est pas réellement abandonné. Google a forké WebCore, le moteur de rendu de WebKit, pour le purger des éléments que l'éditeur trouvait inutile. Blink sera donc une version modifiée de WebCore. Mi-mai, presque 9 millions de lignes de codes avaient été purgées ! Difficile à l'heure actuelle de savoir comment Blink et WebKit vont collaborer. Car Blink va modifier beaucoup de codes et des modifications qui ne seront pas disponibles dans WebKit, a priori. Il faudra un minimum de collaboration entre les deux projets pour éviter de fragmenter le marché. Des développeurs WebKit ont rapidement évoqué le retrait des codes liés à Chromium et à V8 (le moteur JavaScript de Google) qui servaient essentiellement à Chrome. Par contre, des développeurs Chromium / Blink ont annoncé leur volonté de proposer des patches à WebKit... Cela va permettre à WebKit de se remobiliser et de proposer quelque chose de plus optimisé en étant plus à l'écoute des différents éditeurs implémentant le moteur. En définitive, cela change relativement peu les choses dans l'immédiat, car chaque éditeur utilisant WebKit peut le modifier, et l'adapter à ses besoins.

Pointer Events : l'avenir du tactile sur le web

Aujourd'hui, la gestion du tactile (ou « touch » en anglais) est devenue obligatoire sur vos sites web. C'est un passage obligé si vous voulez que votre site soit bien adapté aux nombreux périphériques tactiles du marché : tablettes, smartphones et la panoplie de nouveaux PC tactiles. Ce n'est pas pour autant quelque chose de forcément simple à mettre à œuvre.

Entre les différentes implémentations disponibles et l'obligation de continuer à supporter la souris en parallèle, il faut parfois écrire du code un peu rébarbatif à réaliser. J'aimerais ainsi vous présenter dans cet article l'approche de Microsoft appelée **MSPointers**, une technologie multi-touch émergente permettant de mieux gérer ce scénario. Elle est aujourd'hui uniquement supportée par IE10 mais est reconnue par les autres acteurs (Google, Mozilla) comme la meilleure des solutions pour le futur. En attendant son adoption partout, nous allons voir ensemble qu'il existe déjà une solution de support multi-navigateurs grâce à une librairie de polyfill nommée **Hand.JS** que nous avons écrite en France pour rendre le support de cette technologie moins complexe. Avec tout cela, vous devriez être en mesure de commencer à expérimenter des solutions multi-touch universelles sur vos sites web.

Avant toute chose, il faut savoir qu'il y a plusieurs modèles aujourd'hui disponibles sur le web. Ainsi, pour supporter tous les navigateurs, en plus de nos **MSPointers**, il vous faut jeter un œil au modèle de touch iOS en plus du bien connu modèle W3C pour gérer la souris. Cela fait donc 3 modèles différents à gérer pour l'instant dont un seul de véritablement standardisé : la souris. Le modèle d'Apple a malheureusement longtemps été bloqué au niveau du W3C pour des raisons de brevets. Mais heureusement, il y a une volonté grandissante de standardisation sur le tactile puisque ce dernier est devenu de plus en plus critique. En septembre dernier, Microsoft a ainsi soumis son modèle **MSPointers** auprès du W3C pour une standardisation et nous sommes passés depuis le 9 mai 2013 au statut de Candidate Recommendation : <http://www.w3.org/TR/pointerevents>, ce qui signifie que la spécification est presque terminée et définitivement prête à la production.

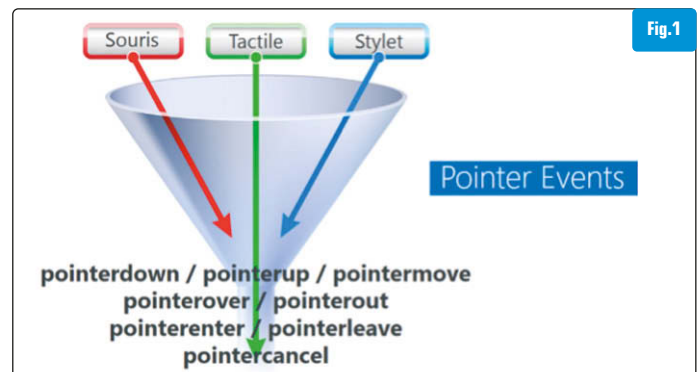
Le modèle **Pointer Events** a donc toutes les chances de son côté pour devenir le standard tactile du web. L'approche de Microsoft sur la gestion des différents types d'interactions est vraiment différente de ce qu'il se passe sur le web actuellement et mérite que l'on s'y attarde. La principale différence avec les autres modèles est que le développeur va écrire son code de manière unique en visant une abstraction de toutes les formes d'entrées via ce que nous appelons un pointeur (« Pointer »). Un pointeur peut représenter n'importe quelle forme d'interaction avec l'écran : le pointeur d'une souris, un stylet, un ou plusieurs doigts. Ainsi, vous n'aurez plus besoin de perdre votre temps à coder pour chacune des formes de manière séparée. Nous allons donc commencer par analyser le comportement d'applications fonctionnant au sein d'Internet Explorer 10 et exposant les événements **MSPointers**. Nous verrons ensuite les solutions permettant de continuer à supporter tous les navigateurs tout en visant

le modèle des **Pointer Events**. C'est quand même là bien le but du web : écrire du code pour tous les navigateurs. Dans un second temps, nous verrons alors comment bénéficier du service de reconnaissance de gestes propriétaires à IE10. Ce service vous permettra de coder les fameuses « gestures » en JavaScript de manière extrêmement simple. Sachez que comme Windows 8 et Windows Phone 8 partagent le même moteur IE10, le code et les concepts présentés ici seront identiques pour chacune des 2 plateformes. De la même manière, tout ce que vous apprendrez dans cet article vous permettra de réaliser exactement les mêmes choses au sein des Windows Store apps écrites en HTML5 puisque, à nouveau, c'est toujours strictement le même moteur utilisé derrière : IE10.

> Les concepts

L'idée derrière l'objet **MSPointer** est donc de vous permettre d'adresser la souris, le stylet et les périphériques tactiles en utilisant une seule base de code via un pattern identique à celui que vous connaissez déjà pour gérer la souris. En effet, la souris, un stylet et un point de contact tactile ont des propriétés en commun : vous pouvez déplacer un pointeur avec eux et vous pouvez aussi cliquer sur un élément avec. Il suffit alors d'adresser les scénarios communs avec exactement le même code. Le modèle de **Pointers** s'occupera alors d'agrégier ces propriétés communes et de les exposer de manière similaire à ce que vous connaissez déjà aujourd'hui avec les événements souris.

Les événements communs les plus évidents sont bien sûr : **MSPointerDown**, **MSPointerMove** & **MSPointerUp** qui se calquent directement sur les événements souris équivalents. Vous aurez ainsi des coordonnées en X et Y sur l'écran, disponibles au sein de ces événements [Fig.1]. Mais vous avez aussi des événements plus spécifiques comme : **MSPointerOver**, **MSPointerOut**, **MSPointerHover** ou **MSPointerCancel**.





Il y a également des cas où vous allez souhaiter gérer le tactile de manière différente de la souris afin de fournir une ergonomie adaptée à votre utilisateur en fonction de sa façon d'interagir avec votre site. Par ailleurs, grâce aux écrans multipoints, un utilisateur peut facilement zoomer ou faire tourner un élément à l'aide de 2 doigts ou plus. De son côté, un stylet peut fournir des informations de pression sur l'écran qu'une souris ne peut vous renvoyer. Le modèle de Pointer Events s'occupera à nouveau d'agréger ses différences de manière à vous permettre facilement d'écrire du code spécifique à chaque usage.

GÉRER LES ÉVÈNEMENTS TACTILES SIMPLES

> Etape 1 : ne rien faire d'autre que d'ajouter une ligne de CSS

Commençons par les bases. Vous pouvez très facilement reprendre n'importe lequel de votre code JavaScript exploitant les événements souris et il fonctionnera tel quel avec les stylets et périphériques tactiles sous Internet Explorer 10. En effet, IE10 lance les événements souris en dernier ressort si vous ne gérez pas directement les Pointer Events dans votre code. C'est pour cela que vous pouvez cliquer sur un bouton ou n'importe lequel des éléments d'une page web avec vos doigts, même si le développeur n'avait jamais imaginé que quelqu'un le fasse un jour ainsi. Bref, n'importe quel code s'abonnant aux événements *mousedown* et/ou *mouseup* fonctionnera sans aucune modification du tout. Mais qu'en est-il de *mousemove* ?

Pour le savoir, passons en revue le comportement par défaut en analysant le résultat de cet exemple de code :

```
<!DOCTYPE html>
<html>
<body>
  <canvas id="drawSurface" width="400px" height="400px" style="border: 1px dashed black;">
</canvas>
<script>
  var canvas = document.getElementById("drawSurface");
  var context = canvas.getContext("2d");
  context.fillStyle = "rgba(0, 0, 255, 0.5)";

  canvas.addEventListener("mousemove", paint, false);

  function paint(event) {
    context.fillRect(event.clientX, event.clientY, 10, 10);
  }
</script>
</body>
</html>
```

Ce code s'occupe simplement de dessiner des carrés bleus de 10 pixels par 10 pixels en suivant les mouvements de la souris au sein d'un canvas HTML5. En utilisant ce comportement par défaut, seuls *MouseDown/Up/Click* fonctionnent au touch. Si vous avez un écran tactile, vous verrez ainsi que lorsque vous bougez la souris au sein du canvas, cela dessine des séries de petits carrés bleus. Mais si vous utilisez l'écran tactile à la place, vous ne pourrez afficher un seul carré bleu qu'à l'endroit précis où vous taperez le canvas avec l'un de vos doigts. Dès que vous allez essayer de bouger votre doigt

à l'intérieur du canvas, le navigateur va naturellement faire défiler le contenu de la page puisque c'est le comportement qu'on lui demande d'avoir par défaut.

Il nous faut donc indiquer au navigateur de changer de comportement et de rediriger les événements tactiles vers votre code JavaScript plutôt que d'essayer de les interpréter lui-même. Pour cela, il vous suffit de cibler les éléments de votre page qui ne doivent plus réagir au comportement par défaut en appliquant cette règle CSS :

```
-ms-touch-action: auto | none | manipulation | double-tap-zoom | inherit;
```

Vous avez différentes valeurs possibles en fonction de ce que vous souhaitez filtrer ou non. Pour ceux qui connaissent le modèle Touch Events de WebKit, c'est l'équivalent d'un *preventDefault()* mais par CSS.

Le cas d'usage typique est le contrôle de cartographie que vous insérez dans votre page. Vous voulez alors en général laisser l'utilisateur se déplacer dans la carte avec les gestuelles classiques du touch (pour zoomer et se déplacer) mais garder le comportement par défaut sur le reste de la page. Dans ce cas, il faut simplement appliquer la règle ci-dessus avec la valeur *none* uniquement sur le conteneur HTML embarquant la cartographie.

Dans notre cas, il faut ajouter ce bloc de CSS :

```
<style>
  #drawSurface
  {
    -ms-touch-action: none; /* Désactive tous les comportements tactiles par défaut, comme le défilement et le zoom */
  }
</style>
```

Maintenant, si vous bougez l'un de vos doigts à l'intérieur du canvas, il se comportera comme un pointeur de souris. Mais vous devriez rapidement vous poser alors la question suivante : pourquoi cela ne suit-il qu'un seul de mes doigts ? Tout simplement parce que nous tombons dans le cas de dernier ressort justement. IE10 se contente de mapper l'un de vos doigts au pointeur de la souris. Et autant que je sache, on n'utilise qu'une seule souris à la fois. Donc 1 souris == 1 point de contact tactile maximum en utilisant cette approche. Bon, alors, comment vraiment gérer le multipoint alors ?

> Etape 2 : utilisez les événements MSPointer en lieu et place de ceux de la souris

Prenez n'importe lequel de vos codes existants et remplacez vos abonnements aux événements « *mousedown/up/move* » par « *MSPointerDown/Up/Move* » et votre code deviendra alors directement compatible avec le support multi-touch d'IE10 !

Dans l'exemple précédent, il suffit ainsi de changer cette ligne de code :

```
canvas.addEventListener("mousemove", paint, false);
```

par celle-ci :

```
canvas.addEventListener("MSPointerMove", paint, false);
```

Vous pourrez désormais dessiner autant de séries de carrés que

vosre écran supporte de points de contacts et/ou que vous avez de doigts ! Encore mieux, le même code marche pour gérer le tactile, la souris et le stylet. Cela veut dire par exemple que vous pouvez dessiner une série de carrés à la souris tout en en dessinant d'autres en même temps avec vos doigts.

Mais si vous souhaitez aller plus loin et changer le comportement de votre code en fonction du type d'entrée, il faut alors tester la valeur de la propriété `pointerType`. Si vous faites partie des chanceux qui disposent d'une machine supportant les 3 types d'entrées (comme la *Microsoft Surface Pro* par exemple), vous serez en mesure de voir les différences visuelles entre les 3.

Tout cela est bien sympathique mais nous avons désormais un problème de taille avec ce code. Il gère effectivement bien tous les types d'entrées au sein d'IE10 mais ne marche pas du tout dans les navigateurs ne supportant pas encore le modèle de Pointer Events comme IE9, Chrome, Firefox, Opera & Safari !

> Etape 3 : détection de fonctionnalité pour fournir une expérience alternative

Comme tout bon développeur web qui se respecte, vous savez déjà sûrement que la meilleure approche pour supporter l'ensemble des navigateurs est de faire de la détection de fonctionnalités. Dans notre cas, cela consiste à tester cette fonctionnalité :

```
window.navigator.msPointerEnabled
```

Faites attention cependant à ce test. Il vous indiquera uniquement si le navigateur actuel supporte les MSPointers. Il ne vous dira pas si un écran tactile est disponible ou pas. Pour tester la fonctionnalité tactile, il vous faut en plus vérifier `msMaxTouchPoints`.

En conclusion, pour avoir un code fonctionnel supportant à la fois le modèle MSPointer d'IE10 et proposant une expérience alternative dégradée vers le pointeur de souris pour les autres navigateurs, vous avez besoin du code suivant :

```
var canvas = document.getElementById("drawSurface");
var context = canvas.getContext("2d");
context.fillStyle = "rgba(0, 0, 255, 0.5)";
if (window.navigator.msPointerEnabled) {
    // Si les Pointer events sont supportés.
    canvas.addEventListener("MSPointerMove", paint, false);
}
else {
    canvas.addEventListener("mousemove", paint, false);
}

function paint(event) {
    context.fillRect(event.clientX, event.clientY, 10, 10);
}
```

Cet exemple fonctionnera alors en multipoints, à la souris et au stylet dans IE10 et uniquement à la souris dans IE9, Chrome, Firefox, Opera, Safari.

> Etape 4 : supporter tous les modèles de touch

J'imagine que vous aimeriez bien aller plus loin qu'une simple expérience souris dégradée pour les navigateurs autres qu'IE10. Pour

cela, il vous faut alors supporter les 3 modèles dont je parlais au début : Touch Events d'Apple, événements souris W3C et MSPointer en attendant sa standardisation finale. Il y a 2 manières de mettre cela en place :

- 1 – Ecrire du code pour adresser tous les types d'événements en parallèle (`mousemove`, `touchmove`, `MSPointerMove`) et rapidement avoir potentiellement un code spaghetti
- 2 – Ajouter simplement une référence à Hand.JS, une librairie de polyfill en JavaScript conçue par mon ami David Catuhe, disponible en open source ici : <http://handjs.codeplex.com>

Cette superbe librairie vous permettant de viser la spécification Pointer Events du W3C. En effet, bien que déjà au statut de « CR », la spécification Pointer Events n'est pas encore un standard adopté par tous. Mais grâce à cette librairie, vous pouvez déjà implémenter ce futur standard et être prêt lorsqu'il deviendra adopté par tous les navigateurs modernes. En attendant, la librairie s'occupera de propager les Pointer Events vers la forme actuellement propriétaire MSPointer Events dans IE10, vers les Touch Events pour les navigateurs implémentant le modèle d'Apple, voire même vers les événements souris en dernier ressort pour les anciens navigateurs ! Je vous invite donc à lire l'article qu'il a écrit sur son blog pour savoir comment cela fonctionne : <http://aka.ms/handjs>

Pour avoir une idée de comment utiliser cette librairie, je me suis amusé à m'en servir dans un article sur mon blog (<http://aka.ms/handjsdemo>) qui montre comment écrire un joystick tactile virtuel utilisant les Pointer Events. Grâce à Hand.JS, il fonctionne dans IE10 sur Windows 8/RT, Windows Phone 8, iPad/iPhone et sur les périphériques Android/Firefox OS avec la même base de code ! [Fig.2] Ce code a déjà été utilisé dans certains jeux vidéo HTML5 que j'ai croisés.

RECONNAÎTRE DES GESTUELLES

Maintenant que nous savons comment fonctionnent les bases du multi-touch, regardons comment reconnaître des gestuelles simples comme taper un élément ou le maintenir puis ensuite des gestuelles plus avancées comme déplacer ou agrandir/rétrécir un élément. IE10 fournit un objet nommé `MSGesture` qui va bien nous aider pour ce travail. Notez d'ailleurs que cet objet est spécifique à IE10 et ne fait pas partie de notre soumission au W3C. Combiné avec l'élément `MSCSSMatrix` (équivalent à `WebKitCSSMatrix`), vous serez en mesure de construire des expériences multi-touch très intéressantes, le tout de manière fort simple. `MSCSSMatrix` représente en effet une matrice 4x4 homogène permettant un accès au DOM par script aux fonctionnalités CSS 2-D et 3-D.



Fig.2



Ce qu'il faut retenir en premier lieu est qu'il est d'abord nécessaire de s'abonner à l'évènement **MSPointerDown**. Ensuite, à l'intérieur du handler prenant en charge **MSPointerDown**, vous devrez choisir quels sont les pointeurs que vous souhaitez envoyer à l'objet **MSGesture** pour qu'il s'occupe de détecter certaines gestuelles spécifiques. Ce dernier renverra alors l'un de ces événements : **MSGestureTap**, **MSGestureHold**, **MSGestureStart**, **MSGestureChange**, **MSGestureEnd**, **MSInertiaStart**. En effet, l'objet **MSGesture** travaille sur un jeu de pointeurs en entrée et s'occupe ensuite d'appliquer un algorithme de reconnaissance de gestuelles par-dessus. La seule chose que vous devez faire est donc de choisir/filtrer les pointeurs devant faire partie de la gestuelle (basé sur leurs identifiants, coordonnées à l'écran, etc... à vous d'imaginer la logique qui vous correspond). L'objet **MSGesture** s'occupera alors magiquement du reste.

> Exemple : gestion des déplacements, rotation et mise à l'échelle d'un élément

Maintenant que vous savez presque tout, vous allez voir que mettre à l'échelle, déplacer ou tourner un élément est l'enfance de l'art grâce à quelques lignes de code. Vous devez tout d'abord vous abonner à l'évènement **MSGestureChange**. Cet évènement va vous envoyer les différents attributs comme définis dans la documentation de l'objet **MSGestureEvent** comme les valeurs actuellement positionnées sur votre élément HTML. Encore mieux, par défaut, l'objet **MSGesture** fournit un algorithme inertiel automatique. Cela signifie que vous pouvez prendre n'importe quel élément HTML et le jeter à travers l'écran en utilisant vos petits doigts et l'animation naturelle sera gérée pour vous. Pour finir, afin de refléter ces changements envoyés par **MSGesture**, vous devez bien entendu bouger l'élément vous-même. La manière la plus simple consiste à appliquer une transformation CSS parfaitement calquée sur les valeurs de rotation, mise à l'échelle et déplacement fournis par la gestuelle reconnue. Pour cela, utilisez l'élément **MSCSSMatrix**.

En conclusion, si vous souhaitez mettre en place toutes ces gestuelles sympathiques et modernes, regardez le peu de lignes de code nécessaire :

```
var myGreatPic = document.getElementById("myGreatPicture");
var myGreatPicAssociatedGesture = new MSGesture();
myGreatPicAssociatedGesture.target = myGreatPic;

// Vous devez d'abord vous enregistrer auprès de MSPointerDown
// avant d'accéder au moteur de reconnaissance gestuelles
myGreatPic.addEventListener("MSPointerDown", pointerdown, false);
myGreatPic.addEventListener("MSGestureChange", manipulateElement, false);

function manipulateElement(e) {
    // Décommenter ce code si vous souhaitez désactiver l'algo
    // d'inertie fournit par défaut
    // if (e.detail == e.MSGESTURE_FLAG_INERTIA)
    // return;
}
```

```
// On récupère la dernière matrice de transformations CSS appliquée
// et on en calcule une nouvelle en fonction des dernières
// interactions
var m = new MSCSSMatrix(e.target.currentStyle.transform);
e.target.style.transform = m
    .translate(e.offsetX, e.offsetY)
    .rotate(e.rotation * 180 / Math.PI)
    .scale(e.scale)
    .translate(e.translationX, e.translationY)
    .translate(-e.offsetX, -e.offsetY);
}

// Dès que l'on touche l'écran, on envoie tous les pointeurs
// à l'objet MSGesture
function pointerdown(event) {
    myGreatPicAssociatedGesture.addPointer(event.pointerId);
}
```

Ce qui donne le résultat final suivant : [\[Fig.3\]](#).

Avec cet exemple, on peut déplacer et lancer l'image dans la zone à fond noir avec 1 ou plusieurs doigts. On peut également zoomer ou faire tourner l'image avec 2 ou plusieurs doigts. Sur cette copie d'écran, on voit par exemple que j'utilise 5 doigts et qu'IE10 calcule automatiquement le barycentre et la rotation/mise à l'échelle pour nous. Le résultat est vraiment agréable et le code pour le réaliser extrêmement simple puisque toute la complexité a été prise en charge nativement par le navigateur.

Logiquement, avec tous les détails que je vous ai partagés dans cet article ainsi que les ressources associées, vous devriez être en mesure de commencer à implémenter le modèle de Pointer Events dans vos sites web & vos applications pour le Windows Store. Vous avez désormais l'opportunité de facilement augmenter l'expérience utilisateur de vos sites sous Internet Explorer 10.

David Rousset - @davrous

Technical Evangelist – Microsoft France



Fig.3

L'info continue sur www.programmez.com

CODE

Les sources
des articles

NOUVEAU

Livres blancs :
langages, outils...

TÉLÉCHARGEMENT

Les dernières versions de vos
outils préférés + les mises à jour

QUOTIDIEN

Actualité, Forum
Tutoriels, etc.

Les navigateurs et HTML 5

HTML 5 introduit de nombreuses fonctionnalités : pour améliorer le stockage, le traitement des fichiers et tout ce qu'il s'ensuit, d'autres pour adapter l'expérience utilisateur au contexte où la mobilité est plus que jamais d'actualité.

> Implémentation de HTML5

Données

Pour améliorer le stockage des données au sein du navigateur, les développeurs ne sont désormais plus obligés de se limiter à l'usage des cookies. La mise en place des objets `localStorage` (et `sessionStorage`) et des bases de données locales ouvre désormais la possibilité de stocker et de gérer davantage de données en mode "hors ligne" plus facilement. L'application de streaming musical Deezer a d'ailleurs opté pour ces différentes nouvelles fonctionnalités pour offrir un mode hors ligne à ses abonnés premium, il est ainsi possible d'utiliser le site et d'écouter sa musique sans être connecté à Internet.

```
<label id="label1"></label>
<script>
  localStorage.MagazineNom = "Programmez!";
  document.getElementById("label1").innerHTML = "Magazine : " +
  localStorage.MagazineNom;
</script>
```

Géolocalisation et mobilité

Dans un contexte où la mobilité est plus que jamais d'actualité avec nos smartphones capables de presque tout faire, utiliser le navigateur de son téléphone ou de sa tablette ouvre de nouveaux scénarios : Géolocalisation, vibreur, ...

```
<!-- Exemple d'obtention de coordonnées -->
<p id="demo"></p>
<button onclick="getLocation()">Obtenir coordonnées</button>
<script>
  var x = document.getElementById("demo");
  function getLocation() {
    navigator.geolocation.getCurrentPosition(showPosition);
  }
  function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
    "<br>Longitude: " + position.coords.longitude;
  }
</script>

<!-- Exemple de vibreur -->
<button onclick="Vibrer()">Vibrer pendant 10 secondes</button>
<script>
  function Vibrer() {
    navigator.vibrate(1000);
  }
</script>
```

Communication

Le besoin de communication d'informations en temps réel est de plus en plus répandu (bourse en ligne, jeux multijoueurs, actualité live, passage du prochain bus/train en temps réel, ...). Les clients lourds sont bien équipés pour répondre à cette problématique, grâce à leur accès direct aux couches réseau du système d'exploitation.

Pourquoi ces limitations ?

En fait, l'une des sources principales du problème est due au protocole HTTP qui est à la fois half-duplex, stateless et orienté requêtes (le client doit demander explicitement les données au serveur). Les données ne peuvent donc circuler que dans un seul sens à la fois (half-duplex) et chaque échange doit utiliser une connexion différente (stateless). Il existe à l'heure actuelle différentes techniques pour contourner ces restrictions et récupérer des données toutes fraîches, le polling, le long-polling, le streaming.

> Les préfixes CSS et optimisations pour les navigateurs ?

Étant donné que certaines implémentations de HTML 5 sont encore en phase dite de brouillon, chaque navigateur intègre ces nouvelles normes comme bon lui semble, au point, de rajouter des préfixes aux styles CSS pour les intégrer de manière « officieuse » :

- -moz- pour Gecko dans Mozilla Firefox
- -ms- pour Trident dans Internet Explorer
- -webkit- pour le moteur de rendu du même nom
- -o- pour Opéra

Ainsi, ce n'est plus étonnant de retrouver par exemple ce qui suit dans certaines feuilles de styles :

```
-moz-box-shadow : 10px 10px 10px blue;
-o-box-shadow : 10px 10px 10px blue;
box-shadow : 10px 10px 10px blue;
```

Note de la rédaction : ces préfixes ont vocation à disparaître.

> Tester le rendu sur plusieurs navigateurs

Vu que chaque éditeur intègre les recommandations du W3C comme il le souhaite, les sites internet HTML5 n'auront peut-être pas le même rendu. Pour cela il existe plusieurs possibilités pour tester son site. Le premier consiste à installer les différents navigateurs Internet sur différents systèmes d'exploitation... Microsoft a mis à la disposition des développeurs un site Internet (www.modern.ie) qui regroupe des solutions diverses pour permettre notamment de tester son site sur plusieurs navigateurs, systèmes d'exploitation, etc.

Des outils d'analyse sont là pour vérifier les problèmes courants associés à l'utilisation d'anciennes versions d'Internet Explorer, pour maximiser l'expérience utilisateur sur différents navigateurs et pour Windows 8. Pour commencer, rendez-vous dans la section « Analyse d'une page Web » et entrez l'URL de la page à tester. Un rapport est généré et est téléchargeable. Microsoft s'est associé à BrowserStack qui propose une solution pour tester sur plusieurs navigateurs, de plusieurs systèmes d'exploitation : Internet Explorer, Chrome, Firefox sur Windows, OS X, iOS, Android, ...

Andrés **Talavera** - Développeur passionné Microsoft
Microsoft Student Partner- www.cresus.net

Guillaume **Lelasseur** - Développeur HTML5 / PHP
Étudiant à la Web School Factory - www.lelasseur.com

Les outils de développement intégrés

Les navigateurs disposent d'outils pour les développeurs qui leur permettent de déboguer rapidement leurs scripts, d'examiner le comportement sur les différents navigateurs ou d'effectuer une courte itération pour établir de nouveaux prototypes, ou encore simplement essayer de résoudre un problème.

INTERNET EXPLORER

Les outils (F12 pour les ouvrir), se présentent sous la forme d'une fenêtre ancrée pouvant être détachée avec des menus, 6 onglets et deux volets.

> Les différents modes d'affichage

L'une des principales préoccupations des développeurs Web est de s'assurer du bon fonctionnement d'un site dans tous les navigateurs qui pourraient demander à afficher votre site Internet, quelle que soit sa version. Internet Explorer fournit des outils de développement pour simuler un autre navigateur et/ou une autre version. D'ailleurs, sur la partie droite du menu, on peut voir apparaître "Mode navigateur : " et "Mode de document : "

Le premier simule la version choisie d'Internet Explorer (en changeant l'User Agent), le second change le mode d'affichage du document avec l'interpréteur choisi. Pour aller plus loin, il est possible de changer radicalement l'User Agent et de passer par un autre navigateur Internet (Outils, Modifier la chaîne de l'agent utilisateur).

Pour aller encore plus loin, sans installer les autres navigateurs, et voir quand même comment chacun d'eux réagit, Microsoft a sorti un outil pour les développeurs Web : Expression Web SuperPreview.

> Travailler avec HTML et CSS

L'une des principales fonctionnalités des outils de développement est la manipulation de la structure et du design des pages. Des

outils de sélection (par clic ou arborescence du code) et de recherche sont là pour nous aider à trouver facilement une partie de code que l'on souhaite mettre en avant.

Astuces :

- La zone de texte dédiée à la recherche supporte les sélecteurs qui sont utilisés par les feuilles de style CSS et par jQuery, à la condition de commencer votre recherche par @. Exemple : pour chercher une div avec comme id "maincontent", il faudra taper "@#maincontent".
- Vous pouvez ajouter des contours aux éléments pour les voir ressortir dans l'affichage de votre document à l'aide de la commande "Appliquer un contour aux éléments" du menu Outils.

Lorsque vous avez trouvé le contenu à modifier, l'inspecteur HTML vous permet d'ajouter des attributs à vos balises sélectionnées à l'aide d'un clic droit dans l'arborescence des balises à gauche et de modifier les styles appliqués à l'aide du panneau de droite. Ce dernier est parfait pour diagnostiquer les problèmes du genre "Pourquoi j'ai une bordure rouge autour de mon élément ? ". Cette vue affiche tous les styles qui s'appliquent hiérarchiquement à l'élément sélectionné.

Pour trouver rapidement un moyen de découvrir pourquoi un style particulier est utilisé, Microsoft a créé le "Suivre des styles" qui propose une approche différente : c'est un affichage par attribut plutôt que par style. Le module "disposition" est une représentation graphique de la taille des offset, des marges, des bordures et des marges internes de l'élément sélectionné. Les valeurs de ces dernières peuvent être modifiées. L'onglet CSS affiche les styles CSS employés, vous permettant de visualiser

le(s) fichier(s) CSS utilisé(s) dans la page, de faire des changements (activer, désactiver des styles, modifier les valeurs) et d'effectuer des recherches.

> JavaScript Débogage

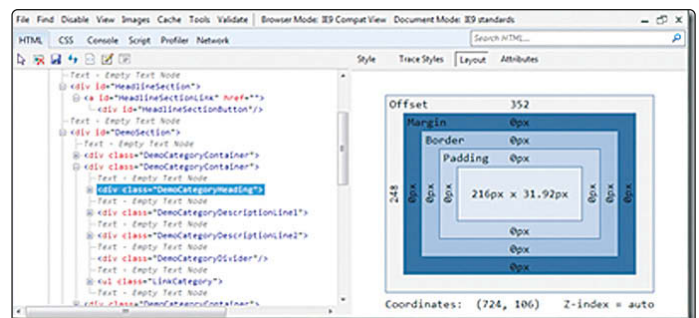
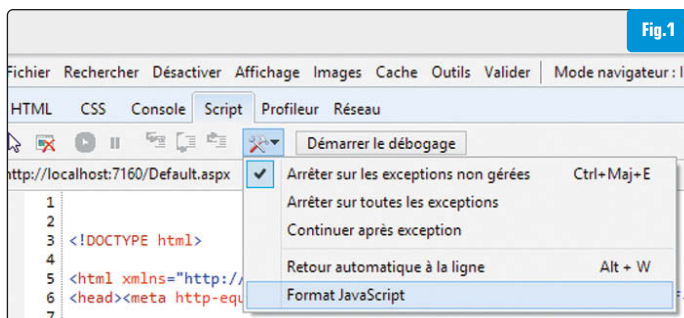
L'onglet Script vous permet de travailler sur le JavaScript, de rajouter des points d'arrêt et de déboguer pour détecter les éventuelles erreurs dans votre code.

Astuces :

- Pour améliorer la lisibilité de votre JavaScript, Internet Explorer vous propose dans la configuration de formater le code JavaScript [Fig.1].
- Les raccourcis clavier sont les mêmes que sur Visual Studio : F5 pour continuer, F11 pour le pas à pas détaillé, F10 pour le pas à pas principal, ...
- Un clic droit sur un bout de code et un menu contextuel vous invite à insérer un point d'arrêt ou d'exécuter jusqu'au curseur sélectionné et, en cas d'exécution, d'ajouter un espion de la variable sélectionnée

La console va vous afficher les erreurs, les informations de diagnostic rencontrées pendant l'exécution de votre script, grâce à l'objet "window.console". Vous pouvez cliquer sur les hyperliens pour naviguer directement vers la ligne incriminée quand elle est disponible. Dans la console, vous pouvez utiliser les commandes suivantes pour détecter rapidement les problèmes :

- console. log – Message générique
- console. info – Message d'information
- console. warn – Message d'avertissement
- console. error – Message d'erreur
- console. assert – Émet un message uniquement si une condition échoue
- console. clear – Efface la fenêtre de console
- console. dir – Affiche les détails d'un objet



- console. profile/profileEnd – Démarre / arrête une session de profilage script

La zone de texte en dessous de la console vous permet d'exécuter des scripts ou des commandes en live - l'affichage sur votre page sera alors mis à jour instantanément. C'est une excellente façon d'apprendre et de tester les commandes de bibliothèques telles que jQuery.

Points d'arrêt

Rien de plus important que les points d'arrêt lors d'un débogage, pour les ajouter, cela ne varie pas sur les logiciels équipés de cette fonctionnalité. Notez qu'un clic droit sur la pastille rouge indiquant le point d'arrêt suffit pour dérouler un menu contextuel vous permettant, entre autres, de rajouter une condition de passage.

Suivi des variables

Les variables locales et espions vous aident à connaître les valeurs des variables et des objets. Les premières concernent toutes les valeurs à la portée de l'exécution en cours. Les deuxièmes sont celles que vous avez choisies de suivre.

Si vous vous demandez quelle série d'appels de fonctions vous a conduit là où vous vous trouvez, utilisez le volet pile d'appels. Sur chaque ligne, un double clic vous conduit à son origine dans le code.

Profilage

Analyser l'exécution de votre JavaScript peut vous amener à l'améliorer pour le plus grand confort des utilisateurs finaux. Vous pouvez

l'utiliser à partir de l'onglet "Profilage".

Lorsque le profilage est terminé, Internet Explorer vous génère un rapport détaillé par fonction ou par arborescence des appels sur son temps et le nombre de fois où il a été exécuté, ainsi que son emplacement dans le code, par exemple. Des colonnes pouvant être ajoutées et/ou supprimées.

MOZILLA FIREFOX

Les outils de développement de Mozilla sont également puissants. Comme sur Internet Explorer, ces derniers sont installés nativement et sont accessibles via le menu "Web Developer". Pas de différents modes d'affichage pour simuler tel ou tel navigateur, mais d'autres outils se montrent bien plus puissants: Design Responsive, rendu 3D, ...

> Travailler avec HTML et CSS

Ce qui saute aux yeux quand on ouvre les outils de développement, c'est l'inspecteur de code HTML/CSS qui nous permet de naviguer dans l'arborescence des balises HTML et parmi les blocs de styles CSS qui s'appliquent à l'élément sélectionné.

L'outil de recherche supporte les sélecteurs, ainsi, si l'on tape ".style1", l'outil révèle tous les éléments où est appliquée cette classe [Fig.2].

Pour sélectionner aisément vos éléments, Mozilla a développé une Vue 3D. Cette dernière permet de mettre en relief la totalité des éléments d'une page Web. Sinon, le survol sur les éléments de votre affichage vous

permet de sélectionner une partie de la page et d'y voir le code qui se cache derrière. L'élément à modifier trouvé, vous pouvez ajouter, modifier et supprimer des attributs par des doubles clics et clics droits. Avec cette dernière opération, un menu contextuel nous permet de simuler les éléments, lorsque la souris passe par-dessus.

L'outil vue adaptative, permet de tester directement le rendu du site avec différentes résolutions [Fig.3].

JavaScript

L'onglet Debugger vous permet de travailler sur vos scripts tels que le JavaScript, comme son nom l'indique, l'onglet va surtout vous permettre de les déboguer, de rajouter des points d'arrêt, de suivre vos variables, etc.

Astuces :

Un clic sur le bouton « Développer les panneaux » : permet d'accéder directement aux grandes fonctionnalités de l'outil.

Il est possible d'ajouter des points d'arrêt, des points d'arrêt conditionnels et des expressions pour espionner directement à l'aide d'un clic droit au sein du code.

Andrés Talavera

Développeur passionné par les technologies Microsoft

Microsoft Student Partner

www.cresus.net

Guillaume Lelasseur

Développeur passionné HTML5 / PHP

Étudiant à la Web School Factory

www.lelasseur.com

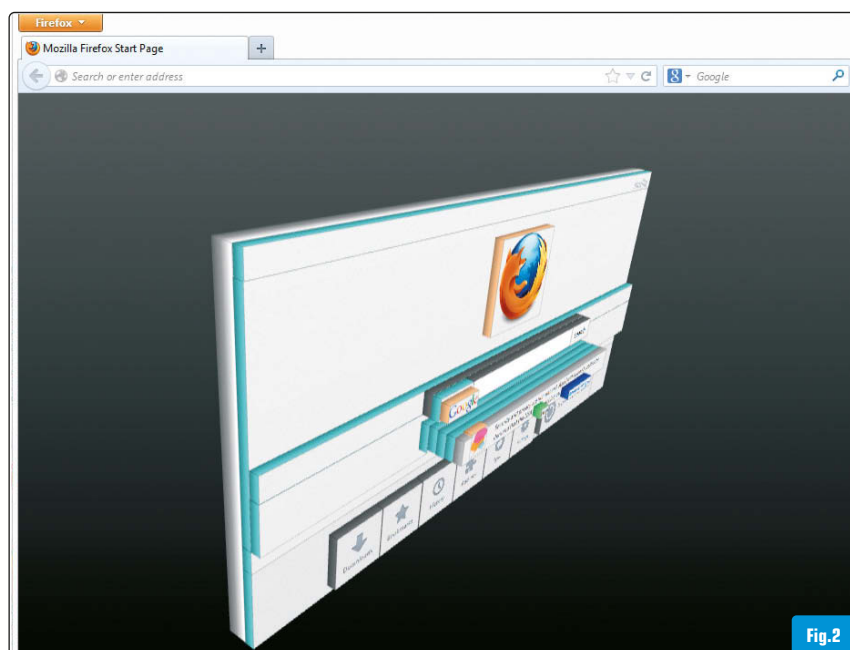


Fig.2

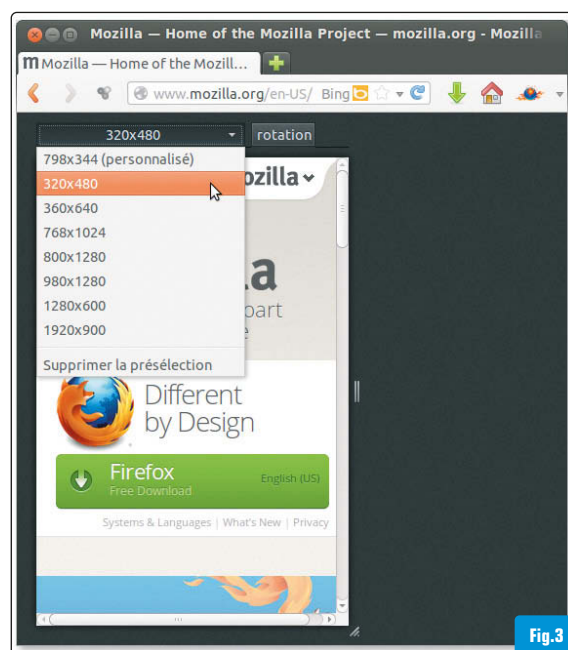


Fig.3

Les nouvelles possibilités CSS de mise en page dans IE10

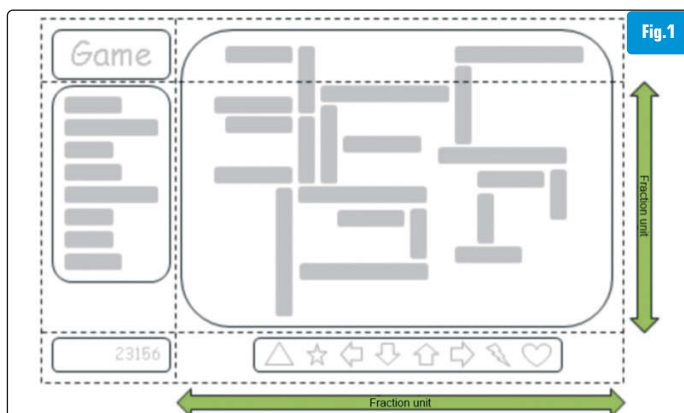
Internet Explorer 10 intègre beaucoup de nouvelles API HTML5 comme IndexedDB, File API, la gestion de l'offline et bien d'autres encore. Mais comme il existe déjà beaucoup de tutoriels et d'articles sur le sujet, j'ai préféré ne pas le ré-aborder avec vous. Considérez juste que cela fonctionne de façon standard dans IE10, et même sans préfixes ! Intéressons-nous plutôt à CSS3.

Il y a beaucoup de nouveautés intéressantes comme le support de CSS3 3D Transform ou de CSS3 Transitions/Animations par exemple. IE10 a d'ailleurs été le premier à supporter toutes ces spécifications modernes sans préfixes. Vous bénéficierez aussi du même niveau de support dans IE10 sur Windows Phone 8.

Pour ce dossier, j'ai plutôt décidé de vous présenter des spécifications que vous ne connaissez peut-être pas, car encore peu répandues sur le web et quasiment uniquement supportées par IE10. L'idée est donc de vous montrer le potentiel futur de ces spécifications standards du W3C en espérant qu'elles arrivent vite dans les autres navigateurs. Elles sont d'autant plus intéressantes qu'elles vont grandement aider à la création de sites web dits « Responsive » et adaptés aux mobiles et tablettes. Par ailleurs, ces spécifications sont déjà disponibles en production (mais préfixées) dans les applications Windows Store Apps se reposant sur le moteur de rendu IE10. Donc, vous pouvez déjà vous faire plaisir en écrivant des applications HTML5 pour le Store de Windows 8 avec ces nouvelles spécifications en cours de standardisation. Cela vous donnera un avant-goût du web de demain.

> CSS Grid Layout

Cette spécification est vraiment simple à l'usage mais se révèle extrêmement puissante pour la mise en page. Elle fut créée par Microsoft et soumise au W3C pendant le développement de Windows 8 et pour servir les applications écrites en HTML5 pour le Windows Store. En effet, c'est tout simplement cette spécification qui est à la base de notre nouveau design Modern UI (ex-Metro). Mais vous allez voir que cela va également vous ouvrir des possibilités très intéressantes pour vos propres idées de design. Pour illustrer son usage, imaginons que nous partons du layout suivant, censé représenter une planche de jeu : [Fig.1]. On a donc une structure en grille. Cette grille dispose de 3 lignes et 2 colonnes. Certaines parties doivent



également être flexibles et donc s'adapter automatiquement aux différentes résolutions. Il y a une époque, on aurait été tenté de réaliser cette mise en page avec l'élément HTML `<table/>`. Mais nous savons tous aujourd'hui que c'est une très mauvaise idée. Il faut absolument séparer le contenu de la mise en forme. CSS Grid Layout va donc nous permettre de faire cela. Pour construire cette grille, il nous faut tout d'abord définir la structure HTML toute simple :

```
<div id="grid"> _
  <div id="title">Game Title</div>
  <div id="score">Score</div>
  <div id="stats">Stats</div>
  <div id="board">Board</div>
  <div id="controls">Controls</div>
</div>
```

Vous pouvez bien entendu utiliser n'importe quel élément sémantique HTML5 à la place si vous le souhaitez (`<header />`, `<article />`, etc.). Une fois le contenu de votre document HTML défini, regardons la facilité de mise en forme par CSS. Commençons par viser le conteneur principal « grid » :

```
#grid {
  display: -ms-grid;
  -ms-grid-columns: auto 1fr;
  -ms-grid-rows: auto 1fr auto;
}
```

On bascule son display en « `-ms-grid` » puis on définit le nombre de colonnes et lignes qu'il pourra contenir. La définition ci-dessus correspond exactement au schéma de planche de jeu présenté précédemment. « `auto` » signifie que la cellule devra adapter sa taille automatiquement en fonction de la taille de l'élément qu'elle contiendra. Ensuite, on découpe l'espace de manière proportionnelle avec une unité de fraction (fr). Dans l'exemple ci-dessus (le plus simple), on demande simplement de prendre tout l'espace disponible quelle que soit la résolution (1fr). Mais si on avait voulu diviser l'écran en 2 colonnes prenant respectivement 1/3 et 2/3 de l'écran, on aurait défini cela avec le code suivant :

```
-ms-grid-columns: 1fr 2fr;
```

On calcule la somme de la totalité des fractions définies ($1\text{fr} + 2\text{fr} = 3\text{fr}$) et cela détermine ensuite la proportion pour chaque colonne ($1\text{fr}/3\text{fr}$ et $2\text{fr}/3\text{fr}$). Si nous revenons à l'exemple de notre `#grid`, imaginons ainsi que l'écran du navigant fait 1920 de largeur et que vous mettez un élément (une image) de 920 de large dans la 1^{re} colonne (définie en « `auto` »). La 2^e colonne (1fr) occupera donc l'es-

pace restant soit $1920 - 920 = 1000\text{px}$. C'est donc extrêmement pratique à l'heure du design web adaptatif. Une fois la grille définie, il faut indiquer comment ventiler chacun des contrôles HTML dans les cases. Pour réaliser notre planche de jeu, voici une solution possible :

```
#title { -ms-grid-column: 1; -ms-grid-row: 1; }
#score { -ms-grid-column: 1; -ms-grid-row: 3; }
#stats { -ms-grid-column: 1; -ms-grid-row: 2; -ms-grid-row-align: start; }
#board { -ms-grid-column: 2; -ms-grid-row: 1; -ms-grid-row-span: 2; }
#controls { -ms-grid-column: 2; -ms-grid-row: 2; -ms-grid-column-align: center; }
```

Si vous avez déjà goûté à d'autres technologies d'affichage (comme XAML), cela ne devrait pas être bien dur à comprendre ! Sinon, dans tous les cas, le concept est simple. Vous visez l'élément qui vous intéresse via un sélecteur CSS (ici ciblage par ID, le plus simple) et vous indiquez où il doit habiter dans la grille, les indices commençant à 1. Vous voyez aussi un autre côté puissant du moteur de layout avec des attributs permettant d'indiquer que l'on souhaite centrer l'élément verticalement et/ou horizontalement dans la cellule.

Pour aller encore plus loin, on peut bien évidemment coupler cette nouvelle spécification avec une autre que vous devez sûrement déjà connaître : CSS Media Query (supportée depuis IE9). Imaginons en effet vouloir définir 2 grilles différentes basées sur le même contenu HTML en fonction d'un affichage portrait ou paysage (cas classique sur téléphone ou tablette). Voici la tête de la grille en affichage paysage : [Fig.2]. Et le CSS associé :

```
@media (orientation: landscape) {
    #title { -ms-grid-column: 1; -ms-grid-row: 1 }
    #score { -ms-grid-column: 1; -ms-grid-row: 3 }
    #stats { -ms-grid-column: 1; -ms-grid-row: 2; -ms-grid-row-align: start }
    #board { -ms-grid-column: 2; -ms-grid-row: 1; -ms-grid-row-span: 2 }
    #controls { -ms-grid-column: 2; -ms-grid-row: 2; -ms-grid-column-align: center }
}
```

Et pour afficher cette grille en mode portrait, il suffit d'appliquer ce bloc de CSS à la place :

```
@media (orientation: portrait) {
    #title { -ms-grid-column: 1; -ms-grid-row: 1 }
    #score { -ms-grid-column: 1; -ms-grid-row: 2 }
    #stats { -ms-grid-column: 2; -ms-grid-row: 1; -ms-grid-row-span: 2 }
}
```

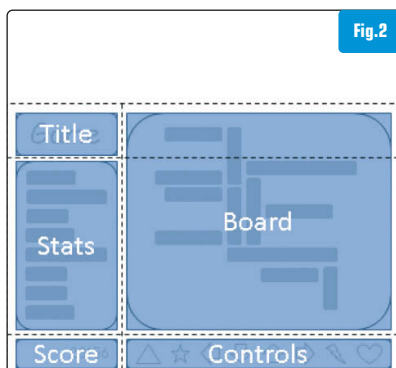


Fig.2

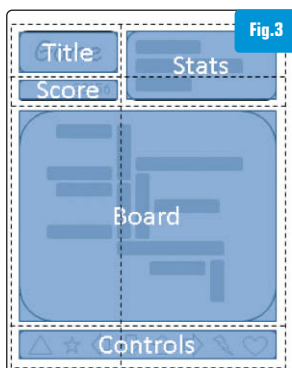


Fig.3

```
#board { -ms-grid-column: 1; -ms-grid-row: 3; -ms-grid-column-span: 2 }
#controls { -ms-grid-column: 1; -ms-grid-row: 4; -ms-grid-column-span: 2; -ms-grid-column-align: center }
```

Ce qui donne ce résultat-là : [Fig.3].

Comme vous pouvez le voir, on peut même redéfinir la grille elle-même lorsque l'on change de résolution ou d'orientation. Elle passe ici d'une définition à 3 lignes/2 colonnes en mode paysage à 4 lignes/2 colonnes en mode portrait. On voit également apparaître la possibilité de s'étendre sur plusieurs cellules avec le concept de « *span* ». Vous avez déjà un bon aperçu de la spécification. Sachez également que vous pouvez bien entendu définir des grilles dans des grilles et ainsi de suite. Vous pouvez par exemple vous rendre avec IE10 sur le site web suivant : <http://snapyx.azurewebsites.net>. Comme vous pouvez le voir, la page principale utilise une structure en grille pour présenter l'information via CSS Grid Layout [Fig.4].

> CSS Multi-columns

Cette spécification est également supportée par les autres navigateurs comme Chrome, Firefox, Safari ou Opera. Elle est par contre non préfixée dans IE10. Internet Explorer, Firefox et Safari supportent également en plus quelques parties supplémentaires comme la césure automatique du texte via la propriété « *hyphens* », préfixée quant à elle. L'usage de « *Multi-columns* » est à nouveau particulièrement simple et permet la réalisation de mises en page très populaires dans le monde de l'impression comme dans le magazine que vous êtes en train de lire par exemple. Le multicolonnage est en effet bien adapté à la physiologie humaine. Il est plus agréable pour l'œil humain de parcourir des petites distances sur plusieurs colonnes plutôt que de devoir lire une ligne faisant toute la largeur de votre superbe écran de plus de 2000 pixels de largeur. Il y a 2 grandes façons d'utiliser le multicolonnage. Soit vous décidez à l'avance du nombre de colonnes à créer quelle que soit la résolution disponible, soit vous décidez d'une taille précise de colonne et le moteur de rendu créera autant de colonnes que de largeur disponible. Dans le 1^{er} cas, la taille des colonnes sera donc plus ou moins large en fonction de la largeur de l'écran et dans le 2^e cas, vous aurez plus ou moins de colonnes visibles. Exemple avec une demande de 3 colonnes fixes grâce à cette déclaration CSS :

```
#content {
    columns-count: 3;
}
```

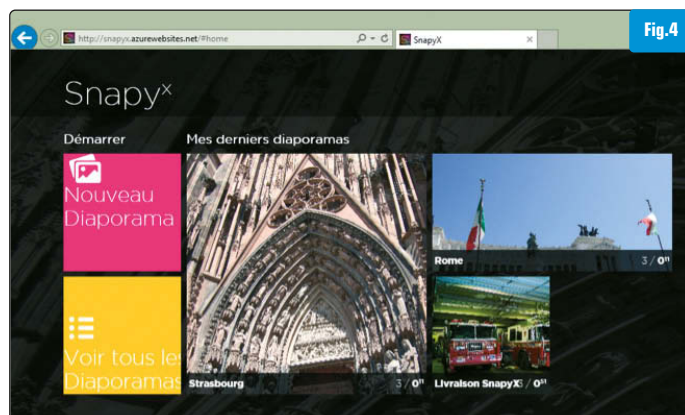


Fig.4

En variant la largeur de fenêtre du navigateur, voici le résultat : [Fig.5]
Observons le même résultat avec la 2^e approche : [Fig.6]

```
#content {
  column-width: 150px;
}
```

Cette fois-ci, plus la largeur de l'écran augmente, plus le nombre de colonnes augmente. Ensuite, on peut bien évidemment gérer l'alignement du texte et l'espacement minimum (*gap*) entre chaque colonne. Grâce à la césure, on peut même optimiser l'utilisation de l'espace disponible en coupant les mots de manière appropriée. Il y a 3 valeurs disponibles : *none*, *manual* ou *auto*. En *auto*, le navigateur va décider lui-même de la manière de découper les mots. Voici un exemple sans, puis un exemple avec cette propriété activée :

```
-ms-hyphens: auto;
```

Avant :

In the traditional CSS box model, the content of an element is flowed into the content box of the corresponding element. Multi-column layout introduces a new type of container between the content box and

on
be
ele
pai
na

Après :

In the traditional CSS box model, the content of an element is flowed into the content box of the corresponding element. Multi-column layout introduces a new type of container between the content box and the con-

on
be
ele
na

Cette spécification peut commencer à être mise en place en production dès aujourd'hui vu son bon niveau de support. La dégradation harmonieuse fonctionnera pleinement pour les anciens navigateurs qui n'auront tout simplement pas le texte sur plusieurs colonnes. L'utilisateur fera plus travailler ses yeux !

> CSS Exclusions

Cette spécification, précédemment appelée *Positioned Floats*, fut créée conjointement par Microsoft et Adobe. L'idée à nouveau est de s'inspirer du monde de l'impression et d'intégrer ses bonnes pratiques dans les futures mises en page du Web.

Le principe de CSS Exclusions est de pouvoir potentiellement définir des formes complexes dans lesquelles ou autour desquelles le texte va s'écouler. Pour l'instant, IE10 supporte la propriété *flow-wrap*. Cette dernière permet d'avoir l'équivalent d'un élément flottant au-dessus du texte qui aura quand même un impact sur son écoulement. Vous pouvez définir si le texte doit couler autour de chacun des 2 bords ou soit au début, ou soit à la fin. Voici un exemple où on

peut bouger le logo de Programmez dans un texte sur lequel on a appliqué un multicolonnage de 3 : [Fig.7]. Que l'on bouge l'image à gauche, au centre ou à droite, le moteur de rendu d'IE10 s'occupe automatiquement de faire écouler le texte correctement. Tout cela avec, accrochez-vous bien, cet énorme bloc de CSS :

```
#content {
  column-count: 3;
  column-gap: 2em;
  text-align: justify;
}

.flow-both {
  position: absolute;
  -ms-wrap-flow: both;
  left: 300px;
  top: 50px;
}
```

Les 3 colonnes de texte présent dans le DIV « *content* » sont justifiées et séparées par un espace de 2em. La classe « *flow-both* » est mise sur l'élément de type image contenant le logo. Elle indique que l'image est en position absolue mais que le texte sous-jacent doit quand même s'écouler entre chacun des côtés (« *both* »).

Pour bien comprendre l'effet de la propriété CSS « *wrap-flow* », regardez ce qu'il se passe si on l'enlève de l'exemple précédent : [Fig.8]. Le logo est désormais bien en absolu mais au-dessus du texte (il le cache) sans aucune forme d'interaction entre les 2. Si l'on combine toutes ces nouvelles spécifications CSS entre elles, voici le type de résultat que l'on peut obtenir : [Fig.9]. Ces mises en page sont issues de l'application Marmiton écrite en HTML5 pour le Windows Store et pourraient donc parfaitement être conçues pour IE10 en mode web. L'écran d'accueil fait un usage massif de l'affichage avec CSS Grid Layout tandis que la fiche descriptive utilise CSS Multi-columns dans une grille CSS plus simple. Ces nouveaux layouts modernes sont désormais bien plus simples à réaliser grâce au travail des différents éditeurs et membres du W3C. Il ne reste plus maintenant qu'à ce que les navigateurs les implémentent et que ces derniers prennent de plus en plus de parts de marché.

David Roussel - @davrous

Technical Evangelist – Microsoft France

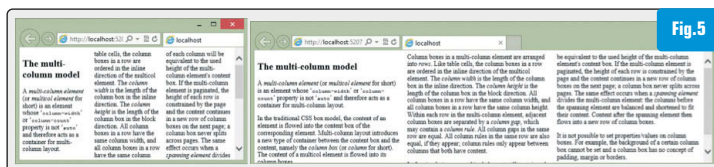


Fig.5

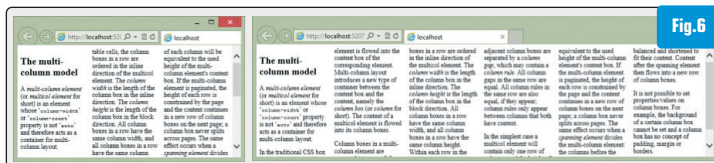


Fig.6

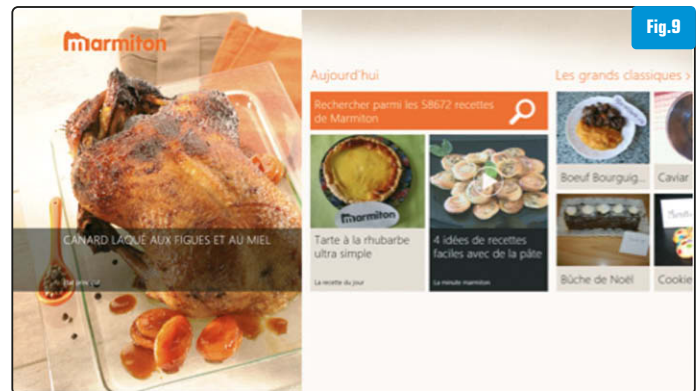


Fig.9

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

PROgrammez !
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

PROgrammez !
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

PROgrammez !
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Fig.8

Fig.7

VITE, ABONNEZ-VOUS jusqu'à -50%

Profitez de cette offre :
bénéficiez pour 1 euro de plus,
d'un des «livres numériques» au choix,
sélectionnés par la rédaction
de Programmez.

Pour 1€ de + Un «livre numérique» des Editions Eni*

Offre limitée et
réservée à la France
Métropolitaine



HTML5, CSS3 et JavaScript
Prix Version numérique : 29,26 €



Objective-C
Prix Version numérique : 40,5 €



Android
Prix Version numérique : 29,26 €

*Accès html au
contenu intégral de
l'ouvrage sur la
Librairie numérique
des Editions ENI,
d'une valeur de
29,26 € à 40,50 €
selon les titres

Valeur des 11 numéros du magazine : 65,50€ + Valeur du livre : 29,26 € à 40,50 € = Valeur totale : 94,76 € à 106 €
Offre : 50€ seulement (ou 60€ avec accès illimité aux archives)**

Toutes les offres en ligne : www.programmez.com

Abonnez-vous à partir de 3,80 € seulement par mois

(*) Tarifs France métropolitaine

Oui, je m'abonne

à retourner avec votre règlement à
Programmez, 17 route des Boulangers 78926 Yvelines cedex 9

Attention : l'offre avec livre Eni est limitée et réservée à la France métropolitaine

- ☐ **Abonnement 1 an au magazine + 1 « Livre Numérique ENI »** : 50 € (au lieu de 65,45 €, prix au numéro + valeur du « livre numérique ») ☐ Abonnement seul : 49 €
☐ **Abonnement Intégral : 1 an au magazine + Archives** Internet et PDF + 1 « Livre Numérique ENI » : 60 € ☐ Abonnement Intégral seul : 59 €
☐ **Abonnement 2 ans au magazine + 1 « Livre Numérique ENI »** : 79 € (au lieu de 130,9 €, prix au numéro + valeur du « livre numérique ») ☐ Abonnement 2 ans seul : 78 €
☐ **Abonnement intégral 2 ans au magazine + Archives** Internet et PDF + 1 « Livre Numérique ENI » : 89 € ☐ Abonnement intégral 2 ans seul : 88 €

Livre à Choisir : ☐ Html 5 CSS 3 et JavaScript ☐ Objective-C ☐ Android

Détails sur www.programmez.com/abonnement.php

Tarifs France métropolitaine

☐ M. ☐ Mme ☐ Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____

E-mail : _____ @ _____

☐ Je joins mon règlement par chèque à l'ordre de Programmez ! ☐ Je souhaite régler à réception de facture

Serveur IIS : après le développement, le déploiement !

La dernière étape lors du développement d'un projet Web est le déploiement de l'application pour la rendre accessible aux utilisateurs. Pour publier un site Internet, il nous faut donc un serveur Web. Il en existe plusieurs mais on fera le tour rapidement d'IIS pour l'environnement Windows, et d'Apache pour l'environnement Linux. Pour ces deux serveurs Web, nous allons considérer qu'il s'agit d'un premier déploiement.

INTERNET INFORMATION SERVICE (IIS)

Ces dernières années Microsoft a fait des efforts considérables en ce qui concerne le développement d'applications mais également le déploiement. IIS s'est vu en effet ajouter des extensions, améliorer la gestion et l'hébergement des applications, bref de quoi chouchouter ceux qui veulent publier leurs sites Internet.

> Installation

...Sur Windows

Sur Windows, il suffit de se rendre dans "activer ou désactiver des fonctionnalités Windows" et de cocher la case "Internet Information Services".

...Sur Windows Server

Sur Windows Server 2012, il faut lancer le gestionnaire de serveur, ajouter des rôles et fonctionnalités et installer le rôle "Serveur Web (IIS)" [Fig.1].

Des services de rôle sont proposés pour compléter davantage IIS. En y ajoutant "Publication WebDev" et "Serveur FTP", ceux cochés par défaut vont nous être utiles pour cet article [Fig.2].

Attention ! Certaines opérations sont évitables et pourtant, on n'y pense pas à chaque fois, ce qui peut poser problème lors de la publication du projet Web. Entre autres :

- Lors de la création d'un nouveau site, il faudra faire attention par exemple au mode d'authentification, les droits sur les répertoires, le paramétrage de l'application pool
- Les certificats de sécurité
- Les bases de données ainsi que leurs chaînes de connexion et les droits pour celles-ci

> Différentes méthodes de publication

Pour déployer une solution Web, Visual Studio vous offre la possibilité de le faire avec la publication en un clic ou via un package de déploiement :

- Le package de déploiement Web est un fichier .zip qui contient non seulement tous les fichiers mais également des informations nécessaires au déploiement. Il suffira de l'installer sur le serveur distant à l'aide du Gestionnaire des services IIS.
- La publication en un clic fait référence à une fonctionnalité dans Visual Studio qui vous permet de déployer directement via

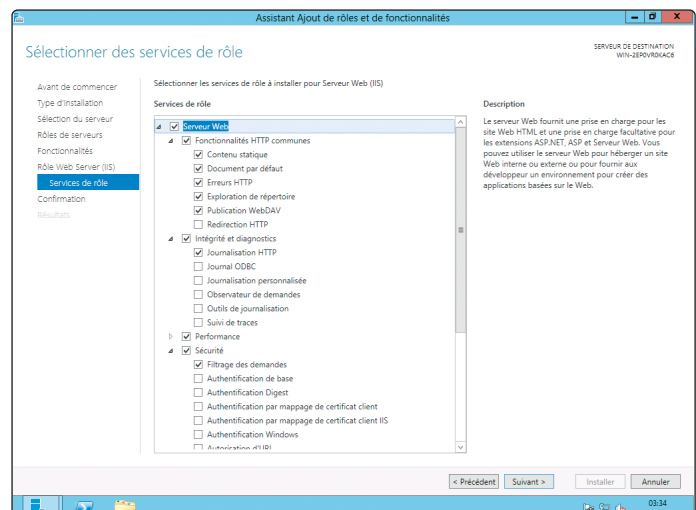
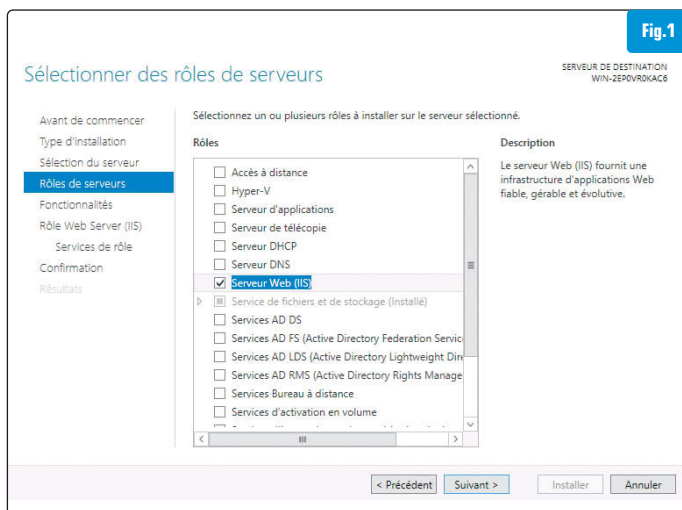
Astuce : utiliser IIS à la place du moteur Cassini

Jusqu'à la version 2010 de Visual Studio, c'était un serveur Web simple et entièrement géré qui exécutait vos sites et applications Web. Depuis, une option (cochée par défaut dans la section "Projet Web") permet d'utiliser d'office IIS. Pour chaque projet, il est cependant possible de choisir entre Cassini ou IIS, à l'aide des Propriétés du projet, dans la section "Web". Il suffit de sélectionner "Utiliser le serveur Web IIS local" et de personnaliser les paramètres donnés par défaut.

un seul bouton. L'IDE se connecte au serveur distant, copie les fichiers nécessaires et effectue d'autres tâches de déploiement.

Web Deployment Tool (ou MSDeploy, ou WebDeploy)

Proposé par Microsoft, l'outil aux trois noms (Web Deployment Tool, MsDeploy ou Web Deploy) a pour but de simplifier la migration, la gestion du déploiement sur les serveurs



Web IIS ; c'est tout naturellement que l'outil a été intégré dans Microsoft Visual Studio.

Lors de la création d'un package par l'outil de déploiement, différents fichiers sont générés, dont la solution zippée et deux manifests. Ces derniers contiennent toutes les informations de l'environnement de développement et de déploiement/publication. Cela va permettre d'obtenir l'intégralité du contenu "local" et "distant" (que nous aurons configuré), et ce, à l'aide des fournisseurs inclus dans MsDeploy, à savoir : DB Provider pour SQL Server, un provider pour gérer les objets COM, un provider pour les paramètres IIS, etc.

Exemple : Si l'on a une base de données SQL avec des données, le fournisseur MsDeploy va répliquer cette base de données, à l'aide de scripts et des fichiers nécessaires.

Générer un package

Au préalable, il faut vérifier les propriétés de notre projet Web, notamment au niveau de "Package/Publication Web" et "Package/Publication SQL" pour la configuration release. Faites donc attention aux éléments à déployer et à la configuration IIS à appliquer en ce qui concerne la partie Web d'une part, et d'autre part aux chaînes de connexion de chaque base de données en entrée (source) comme en sortie (destination). Dans l'onglet dédié à SQL, il vous est possible de générer des scripts pour les schémas relationnels et/ou pour les données.

Pour que Visual Studio nous génère un package, il suffit maintenant de "modifier les profils de publication" et grâce à l'assistant, si ce n'est pas déjà fait, créer un nouveau profil avec comme méthode de publication "Package de Web Deploy".

À l'étape suivante, renseignez le profil à publier (debug ou release) que vous avez configuré et remplissez les champs (le nouveau chemin d'accès physique, nouvelle chaîne de connexion, ...). S'ensuit un récapitulatif de la configuration.

Astuce :

Le .zip peut vous être utile pour déployer votre solution sur un autre poste de développement et continuer vos tâches pour l'améliorer par exemple ; cela peut être utile lors d'un changement de machine.

Installer un package

Maintenant que vous avez généré votre package, il est prêt à être déployé. Envoyez-le sur le serveur par n'importe quel moyen

(réseau, mail, ftp, ...).

Sur votre serveur Web IIS, sélectionnez ou créez un nouveau site. Le volet de gauche contient une partie dédiée au déploiement. Parmi les tâches disponibles : "Importer une application...".

Un nouvel assistant vous demande où se trouve votre solution zippée. Analysé par IIS, l'assistant va vous demander quels éléments, présents dans le WebPackage, vous voulez importer. Dans les paramètres avancés, le paramètre "WhatIf" vous permet de simuler l'installation du package.

Confirmez l'emplacement des fichiers par rapport au site sélectionné et la chaîne de connexion à la base de données. Ceci étant fait, ajoutez les fichiers et cliquez sur suivant. Un résumé vous indique les tâches qui ont été faites.

Publication One-Click

Pour déployer de manière automatique sur un serveur distant IIS un site Internet ou une Application Web, Visual Studio intègre désormais la « publication One-Click ». Cette 'nouvelle' fonctionnalité ressemble fortement à ce qu'on pouvait trouver auparavant, cependant elle prépare une véritable migration et inclut toutes les dépendances nécessaires pour tous les serveurs de destination (jusqu'à 50 profils par projet!).

Avant de commencer...

Si votre serveur IIS n'est pas équipé de Web-Deploy, vous pouvez le télécharger à l'adresse

suivante : <http://www.iis.net/download/webdeploy>.

Le package installé, vous devriez voir « Délégation du service de gestion » dans la page d'accueil de votre serveur Web IIS.

Cette fonctionnalité du panneau est là pour définir les règles de déploiement : que faire pour les chaînes de connexion par exemple. Vous pouvez ajouter et/ou supprimer des règles à l'aide de la tâche « Ajouter une règle » [Fig.3].

Pour déployer votre production, sachez que Visual Studio utilise le Web Deployment Handler qui est stocké dans votre serveur IIS. Pour que le déploiement fonctionne, il faut penser à activer les connexions distantes dans le gestionnaire.

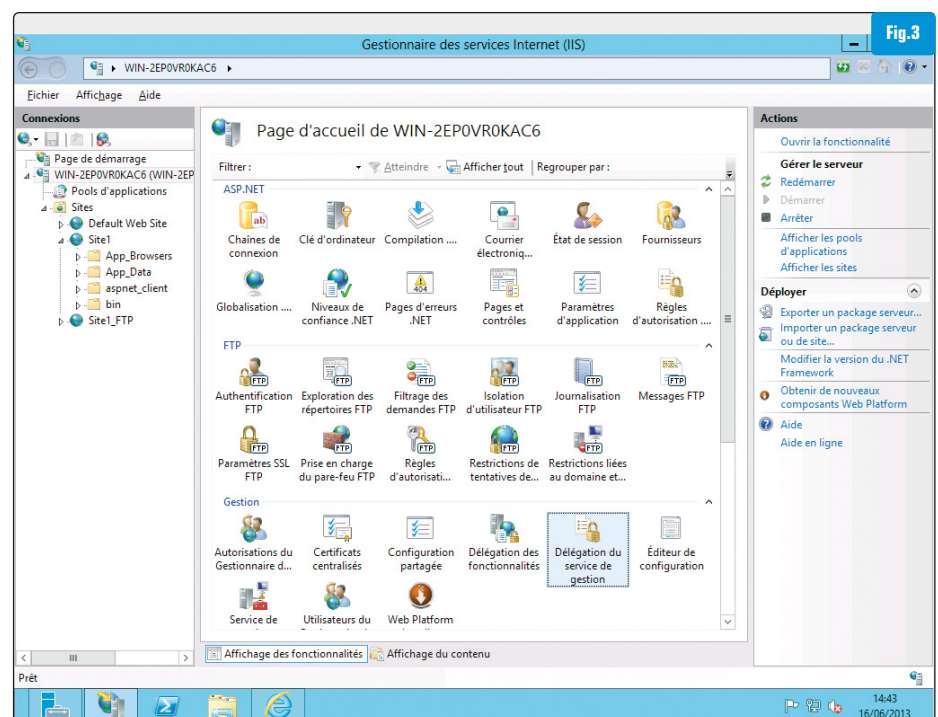
Pour ce faire, à l'aide de la tâche « service de gestion » et d'un arrêt momentané du service de gestion WMSVC, il suffit d'activer l'option « Activer les connexions distantes ».

> Conclusion

L'environnement de développement Visual Studio améliore davantage son système de déploiement, les problèmes pouvant être rencontrés lors des publications, que ce soit au niveau des fichiers web.config ou au niveau de la configuration du serveur IIS.



Andrés Talavera
Microsoft Student
Partner
Communautés
Microsoft



Déployer un serveur Apache, en mode serveur et cloud



L'application web terminée, il faut maintenant lui préparer un endroit pour exister. Il y a diverses possibilités, nous allons voir l'une des principales : Apache.

APACHE SERVER

Apache est le serveur web le plus utilisé. On trouve facilement de la documentation ou des tutoriaux sur le net. De plus, divers outils sont fournis allant de tests de configuration (man apachectl) au benchmark d'un serveur web (man ab).

> Configuration

Dans un premier temps, intéressons-nous à l'organisation de la configuration : il faut que ce soit lisible et ordonné ! Un bon exemple serait l'organisation d'Apache sous Debian : tous les modules et les vhosts sont dans un dossier non inclus dans la configuration (*-available) où l'on sélectionne seulement ce dont on a besoin (a2ensite ou a2enmode pour activer les configurations, c'est-à-dire un lien symbolique dans le dossier *-enabled). Un fichier vhosts par domaine serait une bonne pratique, surtout dans le cas où le vhost 80 ne fait qu'une redirection vers le 443. Exemple d'organisation sous Debian :

```
/etc/apache2
├── apache2.conf
├── conf.d
│   ├── charset
│   ├── localized-error-pages
│   ├── other-vhosts-access-log
│   └── security
├── envvars
├── magic
├── mods-available
│   ├── *.conf
│   └── *.load
├── mods-enabled
│   ├── *.conf -> ../mods-available/*.conf
│   └── *.load -> ../mods-available/*.load
├── ports.conf
├── sites-available
│   ├── default
│   ├── default-ssl
│   └── fqdn
├── sites-enabled
└── fqdn -> ../sites-available/fqdn
```

Exemple d'organisation sous OSX :

```
/etc/apache2/
├── extra
│   ├── httpd-autoindex.conf
│   ├── httpd-dav.conf
│   ├── httpd-default.conf
│   ├── httpd-info.conf
│   ├── httpd-languages.conf
│   ├── httpd-manual.conf
│   ├── httpd-mpm.conf
│   ├── httpd-multilang-errordoc.conf
│   ├── httpd-ssl.conf
│   ├── httpd-userdir.conf
│   └── httpd-vhosts.conf
├── httpd.conf
├── httpd.conf.pre-update
├── magic
├── mime.types
├── other
│   └── php5.conf
├── users
└── pojer_s.conf
```

On peut voir que la configuration est prévue pour être agglutinée dans des fichiers...

Le fichier .htaccess, comme l'indique la documentation elle-même, est à utiliser seulement si vous n'avez pas la main sur la configuration principale.

La configuration peut devenir très désordonnée si vous ne faites pas attention et, en cas de problème, la compréhension et la résolution peuvent prendre beaucoup de temps si l'on doit chercher chaque bout de configuration partout sur le serveur. Et même pire, vous pouvez très bien ajouter des règles saines et casser votre serveur car cela peut annuler d'autres règles ou créer des conflits.

> Outil apachectl

apachectl, ce n'est pas un simple outil, c'est Apache lui-même. Je conseille à ceux qui ne connaissent pas, de lire le man et de regarder le script apache2 dans /etc/init.d/. Au moins trois commandes à retenir :

- Apachectl configtest : teste votre configuration
- Apachectl -S : liste vos vhost par port
- Apachectl graceful : relance votre serveur sans couper les connexions en cours.

> Mode serveur

Un Apache basique est simple à mettre en place. Souvent, seule l'installation des paquets suffit. Il faut définir quelles technologies sont à utiliser : php5.3, php5.4, MySQL, etc. Debian, dotdeb (<http://www.dotdeb.org/>) vous permet d'avoir des paquets plus à jour que les dépôts de base (5.4.4 pour wheezy et 5.4.15 pour dotdeb). Au niveau des logs, ne pas hésiter à les personnaliser (http://httpd.apache.org/docs/2.0/mod/mod_log_config.html), surtout si vous utilisez des applications telles que Subversion et bien d'autres.

Exemple CustomLog pour svn :

```
CustomLog ${APACHE_LOG_DIR}/svn.log "%h %l %u %t %>s %b %(SVN-ACTION)e" env=SVN-ACTION
```

Vous pouvez aussi faire passer les logs dans Syslog via la commande logger.

Exemple Log avec logger :

```
ErrorLog "| logger -t apache-error"
LogLevel warn
CustomLog "| logger -t apache" "%a %v %m \"%U%Q\" %s %B %D \"%{User-agent}i\" \"%{X-Forwarded-For}i\""
```

Nous reviendrons plus tard à la gestion des logs.

> Mode Cloud

Par Cloud, on peut entendre :

- Faire héberger son site
- Rendre son site extensible

Je vais principalement vous parler de la deuxième possibilité. Pour

ceux qui ont lu mon article du numéro 160 de Programmez, j'ai pris comme exemple l'automatisation de la mise en place des serveurs web à l'ETNA (création et configuration de la VM, front web qui s'auto-configuré afin de faire la redirection vers le bon serveur). Nous allons donc voir les différentes directives de la configuration Apache en tant que front web.

Tout d'abord, pensez à activer les modules pour permettre la mise en place du proxy :

- proxy : le module de base
- proxy_http : le module pour le proxy web
- proxy_balancer : ce module est utilisé si vous avez plusieurs serveurs pour un site

Un serveur peut avoir plusieurs applications. La directive permet de garder le nom du site que l'on veut atteindre.

Configuration Proxy Preserve Host :

```
ProxyPreserveHost On
```

Il faut ensuite définir le nombre de serveurs que vous souhaitez avoir par application. S'il n'y en a qu'un, la configuration est simple.

Exemple configuration Proxy simple :

```
ProxyPass / http://IP_DU_SERVEUR/
ProxyPassReverse / http://IP_DU_SERVEUR/
```

Mais s'il y en a plusieurs, la configuration sera sans surprise différente. Exemple configuration Proxy Balancer :

```
<Proxy balancer://NOM_DU_SITE>
  BalancerMember http://IP_1_DU_SERVEUR
  BalancerMember http://IP_2_DU_SERVEUR
</Proxy>

ProxyPass / balancer://NOM_DU_SITE
ProxyPassReverse / balancer://NOM_DU_SITE
```

On voit donc ici que plusieurs serveurs sont pris en compte par le proxy, ce qui revient à peu près au load balancing DNS. L'utilisation du proxy pose un problème avec les logs et les IP des clients. Étant donné que toutes les requêtes passent par le proxy, les serveurs qui sont derrière ne voient que l'IP du proxy. Mais une solution existe : rpaf (paquet libapache2-mod-rpaf pour debian). Exemple configuration rpaf :

```
<IfModule mod_rpaf.c>
  RPAFenable On
  RPAFsethostname On
  RPAFproxy_ips 127.0.0.1 IP_DU_PROXY_1 IP_DU_PROXY_2
</IfModule>
```

Maintenant que votre configuration est prête, vous pouvez lancer la commande suivante pour la prendre en compte :
apachectl configtest && apachectl graceful

Attention : l'utilisation d'un seul proxy est une mauvaise solution, car au premier problème, comme un kernel panic, tous vos sites seront inaccessibles. Pensez à faire de la haute disponibilité avec heartbeats par exemple, avec un DRBD afin de bien répliquer la configuration.

Pour la partie hébergement, je vais être assez bref, car chaque hébergeur est différent et a ses spécificités.

Avant de choisir votre hébergeur, vous devez donc vous renseigner sur la configuration du serveur web. Pour certains, vous pourrez avoir la chance d'éditer vous-même la configuration principale d'Apache et un vaste choix de modules. Pour d'autres, vous n'aurez que le .htaccess pour vos règles ou encore un fichier de configuration exotique qui générera la configuration principale.

> Gestion des logs

Maintenant que l'infrastructure web est en place, automatisée ou non, il faut vous attaquer à la gestion des logs. J'ai mentionné un peu plus haut syslog avec logger. Si vous avez un serveur centralisé, vous pouvez mettre en place la redirection des logs vers votre serveur. Si vous n'en avez pas, je vous invite à regarder du côté de kibana et leur exemple d'infrastructure de log (<http://kibana.org/infrastructure.html>). À l'ETNA, nous utilisons cette solution (l'installation et la configuration sont bien sûr automatisées via Chef) car la gestion des logs est très personnalisable :

- Réception des logs de toutes les applications via rsyslog
- Filtre personnalisable via logstash (vous pouvez écrire vos propres filtres en Ruby)
- Avec rabbitMQ, le traitement des données peut se faire en parallèle.
- Aperçu des logs de façon pratique via kibana
- Possibilité d'exploiter les logs dans d'autres systèmes comme graphite, statsd (module implémenté par logstash)

Steven Pojer, Administrateur Système et Réseau @ ETNA

Le mois des cadeaux (offre limitée au 31 août)

Abonnez-vous jusqu'à -50%

et pour 1€ de +

recevez un "livre numérique" des Éditions ENI

Coupon d'abonnement page 47 et sur www.programmez.com/abonnement.php



Des écoles d'informatique pour tous les goûts

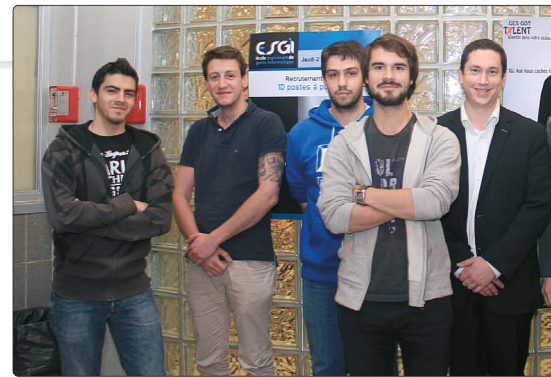
Concurrentes, les écoles d'informatique sont également complémentaires, en acceptant des profils différents et en proposant des pédagogies et des cursus variés. Les bac+5 en informatique sont courtisés par les entreprises avant même leur sortie de l'école ; partisans du droit à la paresse, passez votre chemin ! Tous les autres, hommes et femmes, encore plus recherchés car rares, sont conviés à étudier et travailler dans la filière.

Les écoles d'informatique, c'est comme le chocolat : on part d'une même matière première pour concevoir des cursus et des pédagogies divers. Il y en a pour tous les goûts !

Certes, elles ont quelques caractéristiques communes. La première est de vouloir gommer l'image de l'étudiant geek à lunettes, asocial, qui leur colle à la peau. La seconde est liée : cette étiquette fait fuir les jeunes femmes. En effet, si les directeurs d'écoles d'informatique ne sont pas forcément des fans de Patrick Juvet, ils s'exclament tous en chœur : où sont les femmes ? Selon les écoles, elles représentent 10% ou moins des effectifs en général, sauf à l'Ensimag où le chiffre a atteint de façon mirobolante 20% après des actions en lycée depuis plusieurs années. Malgré les efforts réalisés pour les attirer, la proportion évolue peu. « C'est le

drame absolu, déplore Joël Courtois, directeur général de l'Epita. *Pourtant, rares, elles obtiennent des salaires supérieurs en première embauche* ». C'est souvent en évolution de carrière que la question salariale se complique... Cette belle unanimité est partagée par les employeurs. Albert Cohen, secrétaire général du Groupement Professionnel National de l'Informatique (GPNi) s'écrit : « *C'est le gros problème du secteur. S'il y avait autant de filles que dans d'autres secteurs, cela réglerait une bonne partie de la pénurie de personnel. La pénurie vient également du problème d'image de marque de l'informatique, et de la concurrence féroce du secteur de la finance qui pompe de bons éléments* ».

Autre point commun, en partie pour évacuer l'image du geek mais aussi pour répondre aux attentes des entreprises, les cours déve-



Campus Ionis Group (Epita, Epitech, Sup'Internet).

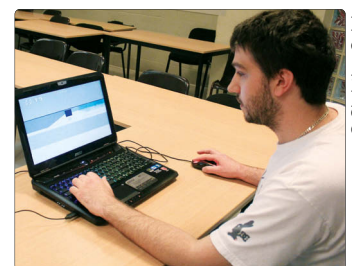
loppant l'ouverture et les qualités humaines prennent une place grandissante : gestion de projet, communication, anglais, management, connaissance de l'entreprise... « *Nous développons les cours de droit avec le cabi-*

À l'ESGI, des jeunes déjà créatifs

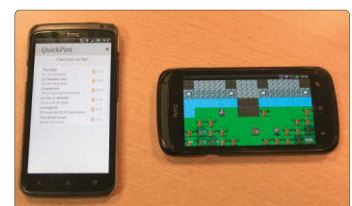
Quentin Cheval et Florian Garnier, après des BTS ou DUT, ont choisi d'intégrer l'Ecole Supérieure de Génie Informatique (ESGI), car elle a ouvert une spécialisation en 3e année d'ingénierie des applications mobiles. Leur projet annuel est le développement de l'application pour Android QuickPint, qui géolocalise les bars parisiens en fonction du prix de la pinte. Reliée avec Facebook, elle permet de créer un événement Facebook pour inviter ses amis à se joindre. Michaël Fournier et Alexandre Dubois, en spécialisation architecture des logiciels, ont développé un jeu de

plateformes. Jonathan Bihet, en spécialisation développement de jeux vidéo, a développé Cubeways, un jeu de labyrinthe en 3D où le héros s'aide de cubes. Ruslan Kassimov, en spécialisation management et conseil en SI, a l'esprit d'entreprise : il a créé en auto-entrepreneur une activité de formation continue en programmation Java et PHP et une SAS de développement Web avec deux associés. Après avoir créé le labo Microsoft de l'école en 2e année, il s'occupe cette année du club des entrepreneurs de l'ESGI, qui compte une quinzaine de membres. Avec Ludovic

Nitoumbi et Jonathan de Galvez, nous avons ainsi rencontré huit jeunes de 3e année à l'ESGI qui se félicitent d'être suivis et encadrés par des professeurs, des offres de stage et d'alternance, des speed meeting avec les entreprises qui permettent au besoin de faciliter l'insertion professionnelle. Ils ont paru bien dans leur peau et pas trop geek... quoique l'un d'entre eux a avoué être en réseau un jour à 6h du matin pour ne pas rater la sortie d'un jeu vidéo. Ils sont ravis d'appartenir au réseau Ges de six écoles dans des domaines variés, qui organisent des soirées communes.



Jonathan Bihet, étudiant à l'ESGI joue à Cubeways.



A gauche, l'application QuickPint de Quentin Cheval et Florian Garnier, à droite le jeu de plateformes de Michaël Fournier et Alexandre Dubois, étudiants à l'ESGI.



© Christine Calais

Emmanuel Peter, directeur de l'ESGI, entouré de huit étudiants de 3e année

net Bensoussan, spécialiste du droit informatique », indique Arnaud Berthier, directeur du département académique de SupInfo, diplômé 2008 de l'école. « *Au-delà des cours, nous les poussons à travers Open ESGI à développer la fibre du management, insiste Emmanuel Peter, directeur de l'Ecole Supérieure de Génie Informatique (ESGI). C'est un programme d'initiatives large : l'étudiant s'implique dans une association, un laboratoire de recherche, dans le club des entrepreneurs, organise des événements ou participe à des challenges...* » Frédéric Meunier, directeur de l'EFREI, l'école d'ingénieur informatique et technologies du numérique, ajoute : « *Les entreprises repèrent les jeunes actifs dans la vie associative* ».

> Partenariats entreprise

Une autre clef pour répondre aux besoins des entreprises, ce sont les partenariats avec les SSII et éditeurs de logiciels, tant au niveau pédagogique que des stages ou alternance. Des professionnels viennent donner des cours ou des conférences. Les écoles sont friandes de journées spéciales organisées avec une grande entreprise qui donne plusieurs conférences sur un thème. Les bourses d'emploi et les forums rassemblant des entreprises à la recherche de stagiaires ou d'alternants restent de grands classiques. Séances de jobdating et tables rondes sur le recrutement viennent compléter le panel. Les entreprises améliorent ainsi leur visibilité auprès des étudiants. Pour répondre à l'évolution rapide des technologies, beaucoup, par exemple l'Epita, s'appuient sur des conseils réunissant professeurs et professionnels qui ont lieu une à deux fois par an pour faire évoluer les programmes. A l'ESGI ont ainsi été récemment mises en place les spécialisations sécurité informatique et ingénierie des applications mobiles.

> International et alternance

Les écoles s'ouvrent progressivement à l'international. Certaines, comme l'Epita,

demandent un semestre ou une année obligatoire à l'étranger, en stage ou à l'université, voire proposent des doubles diplômes. Pour d'autres, cela reste facultatif, comme à l'Exia.Cesi. SupInfo qui dispose de 36 campus, met en avant ceux à l'étranger (Canada, Etats-Unis, Royaume-Uni, Italie, Chine, Maroc) et le partenariat avec des universités chinoises.

Un certain nombre d'écoles propose de suivre le cursus en alternance, un plus pour l'employabilité. A l'ESGI, à partir de la 3e année, la grande majorité des élèves est en alternance deux semaines sur trois dans l'entreprise, qui paye les frais de scolarité, un aspect non négligeable.

Les deux dernières années à l'école In'tech Info, groupe Esiea, se font obligatoirement en alternance.

Cela rentre aussi dans les mœurs des écoles d'ingénieur, l'Ensimag proposant ainsi un cursus en alternance aux diplômés de DUT ou BTS, et l'Efrei un cycle ingénieur par apprentissage.

> Ecoles d'ingénieurs vs écoles d'ingénierie

Les écoles d'ingénieurs avec des spécialités informatiques sont accessibles après deux ans de classe préparatoire, parfois intégrée, ou un BTS ou un DUT. Elles se distinguent par un bagage théorique important, avec des cours fondamentaux en mathématiques et physique.

L'ingénieur dispose ainsi de capacités d'abstraction et de modélisation développées. Les écoles d'ingénieurs privilégient les bases scientifiques, la méthode et le management, ainsi que la recherche. « *Grenoble est en pointe dans la recherche numérique, au niveau universitaire et professionnel, se félicite Yves Denneulin, directeur d'Ensimag, l'école du numérique du groupe d'écoles d'ingénieurs Grenoble INP. Notre filière ingénierie pour la finance travaille en avance de phase par rapport aux besoins des banques en matière de calcul parallèle fondée sur les GPU (Graphic Processing Unit) permettant d'accélérer le temps de calcul. Nous allons monter des formations sur le Big Data avec Grenoble Ecole de Management, de « data scientist » avec l'université Joseph Fourier* ». Les écoles d'ingénieurs sont réputées pour leur solidarité entre anciens élèves, même si les autres écoles développent aussi ces réseaux.

« *Le réseau de 6000 anciens de l'Epita dans plus de 2000 entreprises met à l'abri de la*

Accenture mise sur la diversité des profils

Accenture recrute en France 500 personnes par an dans les métiers du numérique ; un tiers de stagiaires et d'alternants (dont les deux tiers seront ensuite embauchés), un tiers de jeunes diplômés, et un tiers ayant une expérience professionnelle. Le groupe de conseil recrute de bac+2 pour les développeurs en Java EE et les testeurs, à bac+5 pour les experts SAP, Oracle, les consultants fonctionnels. Pour développer des partenariats avec les écoles et universités, Accenture mise sur ses collaborateurs qui y ont étudié et qui aujourd'hui y interviennent. A SupInfo, le groupe organise une journée de simulation d'entretiens. Il propose une journée portes-ouvertes dans l'entreprise. « *Nous sommes en concurrence avec d'autres sur certaines compétences comme les profils Java EE, souligne*

Anissa Deal, directrice du recrutement Accenture France. *Notre politique d'évolution, notamment l'appui d'un conseiller carrière au niveau n+1 du salarié, fait qu'un bac+2 peut se trouver au même niveau de salaire et*



© Accenture

Anissa Deal, directrice du recrutement Accenture France

de responsabilité qu'un bac+5 au bout de trois à quatre ans. » Et d'ajouter : « *sur des compétences techniques pointues, c'est forcément à nous de les former. Nous apprécions l'esprit entrepreneur et innovant et la diversité des profils ; ainsi, les diplômés d'université ayant eu l'habitude de se débrouiller seuls, ont de bonnes capacités d'adaptation.* » Pour promouvoir les carrières féminines, Accenture mise sur son programme et réseau Accent sur Elles.

crise », souligne Joël Courtois, directeur général de l'Epita. Pour les allergiques aux mathématiques et à la physique, les écoles privées d'ingénierie informatique délivrant un titre enregistré dans le Répertoire National des Certifications Professionnelles RNCP ouvrent grand leurs portes.

> Pédagogie pratique ou mixte

C'est dans la pédagogie que nous avons trouvé le plus de différences. Certains offrent une pédagogie par la pratique. Ainsi, Epitech (cf. *Programmez !* du mois dernier) offre une méthode projet à 100%, sans cours. In'Tech Info, école du groupe ESIEA créée en 2002, propose très peu de cours en salle, mais des supports en ligne et du e-learning vidéo. Elle demande surtout deux projets longs par semestre aux étudiants, pour développer les aspects techniques mais aussi humains et organisationnels. « Toutes les écoles offrent de bonnes bases en informatique », explique Eric Lalitte, directeur d'In'Tech Info. La différence se fait sur la compréhension des enjeux de l'entreprise, et la communication. Suivis dans leurs projets, nos étudiants les présentent au forum des projets où viennent une trentaine d'entreprises parmi la centaine partenaire ».

D'autres ont des pédagogies mixtes alliant cours et projets. C'est le cas de l'Epita, de l'ESGI ou de SupInfo. **Morgan Saveuse**,



Morgan Saveuse,
directeur des études
exia.cesi

directeur des études de l'Exia.Cesi, une école privée d'informatique en cinq ans créée en 2004, met en exergue la pédagogie d'apprentissage par problèmes (APP), issue de l'Université du Québec à Montréal : « les étudiants confrontés à une « situation-problème » font émerger des hypothèses puis des solutions. Ils ont deux problèmes à résoudre chaque semaine plus des travaux pratiques et une conférence technologique donnée par un expert. Ils ont un projet d'une semaine chaque mois. Ils doivent avoir acquis au final un référentiel de compétences ».

Les écoles d'ingénieurs peuvent aussi avoir une pédagogie mixte, comme l'Efrei. L'institut de Poly-informatique, groupe IGS, repose sur une pédagogie modulaire, par briques, qui permet de démarrer la formation tous les mois de l'année.

Elle est ainsi ouverte aux demandeurs d'emploi et salariés en formation continue, en plus de la formation initiale ou en alternance qui représente les deux tiers des effectifs. Elle délivre des diplômes du niveau bac à niveau bac+5 : « le diplôme de technicien micro-informatique niveau bac permet à des jeunes sortis du système scolaire de

Le bac+5, un atout pour évoluer

La France est un pays qui accorde traditionnellement beaucoup d'importance aux diplômes. Les diplômes d'études supérieures peuvent être reconnus par l'Etat. La certification de niveau I correspond à un diplôme bac+5, celle de niveau II à un diplôme de niveau licence. Les certifications enregistrées dans le Répertoire National des Certifications Professionnelles (RNCP) sont reconnues sur l'ensemble du territoire national. La Commission nationale de la certification professionnelle met à jour ce répertoire. La Commission des Titres d'Ingénieur, organisme indépendant, habilite les formations qui délivrent le titre d'ingénieur diplômé.

Mais toutes les écoles privées ne voient pas leurs formations homologuées, ou parfois à des titres différents. Dans les grandes entreprises, les grilles de salaire sont fonction des diplômes, et bien sûr, le salaire à l'embauche est fortement dépendant du niveau de diplôme.

Pour un informaticien, la question se pose alors dès son orientation : à bac+5, diplôme d'ingénieur ou diplôme d'expert en informatique de niveau I ?

Ce qui est flagrant aujourd'hui, c'est qu'un informaticien à bac+5 est assuré d'un salaire bien supérieur à celui d'un diplômé de niveau bac+2 ou +3. Par exemple, un diplômé bac+2 de l'Institut de Poly-Informatique (IPI) sera embauché à 21, 27 K€, contre plus de 30 K€ pour un bac+5. Et le chômage touche moins le bac+5. Albert Cohen, secrétaire général du Groupement Professionnel National de l'Informatique (GPNI), constate : « la grosse majorité des compétences recherchées est à niveau bac+5. En termes d'employabilité à moyen terme, l'évolution du salarié dépendra de son niveau d'études, et la priorité est donnée à ceux qui ont le niveau d'études le plus élevé ».

Ensuite, ingénieur ou pas, à chacun de voir ! L'essentiel est de choisir en fonction de ses aspirations et du programme d'études de l'école d'informatique ciblée. Toutefois, dans l'imaginaire collectif, et surtout dans l'esprit, il faut le reconnaître, parfois figé dans un moule des chasseurs de tête, le diplôme d'ingénieur est très valorisé.

< Ici, on recrute !

SSII et éditeurs de logiciels sont face à un marché très tendu, si ce n'est pénurie, sur certains métiers. « Les écoles d'informatique ne suffisent pas à fournir assez de jeunes dont les entreprises du Syntec ont besoin », souligne Guy Mamou-Mani, président du Syntec numérique, alors même que le secteur recrute plus de neuf fois sur dix en CDI et à des salaires bien supérieurs à la moyenne. Il y a un problème d'attractivité et de méconnaissance du secteur : certaines écoles n'arrivent pas à remplir leurs classes. L'objectif de notre campagne de communication 2013 auprès des jeunes de 15 à 18 ans, les S'NNums est que 40 000 à 50 000 jeunes s'inscrivent pour découvrir les métiers du numérique ». Il poursuit : « Il y a des métiers pour tous les niveaux, de l'ingénieur, dont nous apprécions quand il s'oriente vers les secteurs industriel ou numérique, au non-diplômé, une priorité nationale. Une initiative comme celle de l'école 42 de Xavier Niel est plutôt sympathique et volontariste, mais c'est aussi la reconnaissance d'un certain échec de notre filière de formation qu'un entrepreneur en arrive à créer une telle école ». Et M. Mamou-Mani de pousser un cri du cœur vers les jeunes : « Orientez-vous vers le numérique ! »

Albert Cohen, secrétaire général du Groupement Professionnel National de l'Informatique (GPNI), représente les TPE et PME de l'informatique, un vivier de recrutement assez mal connu des jeunes attirés par les grandes entreprises dont ils ont déjà entendu parler, insiste sur les compétences recherchées : « les entreprises ont besoin de gens capables de développer, d'analyser, de concevoir et d'évoluer. Aujourd'hui, un jeune informaticien au cours de son évolution de carrière, va utiliser l'anglais, gérer des projets, manager des équipes, développer des relations avec les clients ».

repandre confiance en eux et de révéler leurs talents. Certains poursuivent les études ensuite, indique Sylvie Holic, responsable du pôle informatique du groupe IGS, Polyinformatique. Nous disons à tous nos étudiants : allez vers le haut à votre rythme ! Les échanges entre personnes actives et jeunes sont un plus ». Ainsi y a-t-il de la place

pour tout type d'école, au vu du déficit abyssal de compétences par rapport aux besoins des entreprises. Car le taux d'emploi six mois après la sortie frise les 100%, avec par exemple une pré-embauche supérieure à 90% avant même la fin du cursus à l'Efrei.

Christine Calais

LES ÉCOLES D'INFORMATIQUE ET LEURS DIPLÔMES NIVEAU BAC+5

Ecole	Diplôme	Niveau	Spécialisations	Possibilité d'alternance	Frais de scolarité annuels (€)	Salaire moyen jeune diplômé M€ (3)	Site Internet
Ecoles d'ingénieurs							
Efrei	Ingénieur	bac+5 (CTI)	5: Systèmes d'information, ingénierie logicielle, informatique et finance de marché, Imagerie et réalité virtuelle, Réseaux d'entreprise	oui	7950	42	www.efrei.fr
Ensimag	Ingénieur	bac+5 (CTI)	7: Ingénierie pour la finance, Ingénierie des systèmes d'information, Modélisation mathématique, images et simulation, Systèmes et logiciels embarqués (Ensimag & Phelma), Télécommunications (Ensimag & Phelma), Communication Systems Engineering (ouverture internationale)	oui	380	40	http://ensimag.grenoble-inp.fr
EPITA	Ingénieur	bac+5 (CTI)	8: Multimédia et technologies de l'information, Sciences cognitives et informatique avancée, Recherche calcul scientifique et image, Systèmes, réseaux et sécurité, Génie Informatique des systèmes temps réel et embarqués, Télécommunications, Systèmes d'Information et génie logiciel, Global IT management.	oui (2)	cycle préparatoire : 5 750, cycle ingénieur : 7 635	37	www.epita.fr
INSA Toulouse	Ingénieur	bac+5 (CTI)	1: Informatique et réseau	non	505	33,9	www.insa-toulouse.fr
Polytechnique	Master	bac+5	5: Master parisien de recherche en informatique, Master parisien de recherche opérationnelle, Master Bioinformatique et bio-statistiques, Master Conception et management des systèmes informatiques complexes, Master Réseaux de communication avancée	non	6600 (M1), 4400 (M2)	NC	www.polytechnique.edu
Télécom Bretagne (1)	Ingénieur	bac+5 (CTI)	2: Informatique et réseaux, Réseaux et télécoms	oui	NC	NC	www.telecom-bretagne.eu
Ecoles d'ingénierie informatique							
Epitech	Titre d'Expert en Technologies de l'Information	bac+5 (niveau I RNCP)	1: Tronc commun et modules complémentaires	non	7720 € (1e et 2e année), 8530 € (3e, 4e et 5e année)	37,5	www.epitech.eu
Ecole Supérieure de Génie Informatique (ESGI)	Titre d'Expert en ingénierie de l'informatique et des systèmes d'information	bac+5 (niveau I RNCP)	7: Systèmes Réseaux Cloud Computing, Architecture des Logiciels, Ingénierie du Web et e-business, Ingénierie de l'Image et des jeux vidéo, Ingénierie des applications mobiles, Management et Conseil en Systèmes d'Information, Sécurité Informatique	oui	5600 (1e et 2e année), 6400 (3e année), 6600 (4e et 5e année)	35,2	www.esgi.fr
ETNA	Titre d'Expert en Informatique	bac+5 (niveau I RNCP)	2: Ingénieur logiciel développeur d'applications, Ingénieur système réseaux & sécurité	oui	gratuit (1e, 2e année), puis 1600 hors frais pris en charge par l'entreprise en alternance	37,8	www.etna-alternance.net
Exia.Cesi	Titre de Manager des Systèmes d'Information	bac+5 (niveau I RNCP)	2: Systèmes réseaux et télécoms, Génie logiciel	non	5000 (1e et 2e année), 6200 (3e, 4e et 5e année)	33	www.exia.cesi.fr
In'Tech Info	Master	bac+5 (niveau I RNCP)	2: Ingénierie logicielle, Systèmes et réseaux	oui	8360	36,8	www.intechinfo.fr
Ionis School of Technology and Management	MBA	bac+5 (niveau I RNCP)	2: Ingénierie Informatique & Management, Technologies de l'information et Management	non	8440	35	www.ionis-stm.com
IPI (institut de Poly-informatique)	Titre de Manager en Systèmes d'Information	bac+5 (niveau I)	2: Conception de solutions applicatives, Administration des systèmes et réseaux	oui	3900 (1e année), 6500 (2e année), 7000 (3e année), 7500 (4e année), 8000 (5e année)	fourchette supérieure à 30	www.inspi.fr
SUPINFO	Master	bac+5 (niveau I RNCP)	1: Expert en Informatique et Système d'Information	oui	5490	36	www.supinfo.com

Données déclarées par les écoles (NC: non communiqué)

(1) Télécom Bretagne en partenariat avec l'ITII Bretagne (Institut des techniques d'ingénieur de l'industrie de Bretagne)

(2) Titre d'Expert en Ingénierie Informatique bac+5 RNCP

(3) Salaire moyen d'embauche du jeune diplômé bac+5 en milliers d'€

Le SDK Facebook pour .NET

Aujourd'hui, tout le monde utilise les réseaux sociaux. Ils sont devenus incontournables et pour les développeurs d'applications c'est un constat qu'on ne peut que confirmer. Il ne se passe pas un jour sans qu'il faille inclure dans une application, nouvelle ou non, le support des réseaux sociaux. Et comme le réseau social de référence est Facebook, c'est souvent lui qu'on nous demande d'implémenter en premier.

Seulement, il n'est pas toujours aisé de le faire. L'API de Facebook est assez complexe et nous devons rester avant tout productifs et rentables. La bonne nouvelle c'est qu'un SDK Facebook vient d'être mis à disposition des développeurs .NET. Il est disponible pour les développements Web, Silverlight, desktop, Windows 8 RT et Windows Phone.

Etant développeur d'applications mobiles, et fan de Windows Phone 8, j'ai tout naturellement choisi de faire mes premiers pas avec ce nouveau SDK sur cette plateforme. Mais rassurez-vous, les différents points qui seront abordés dans cet article seront très facilement adaptables sur les autres plateformes .NET que l'on peut cibler.

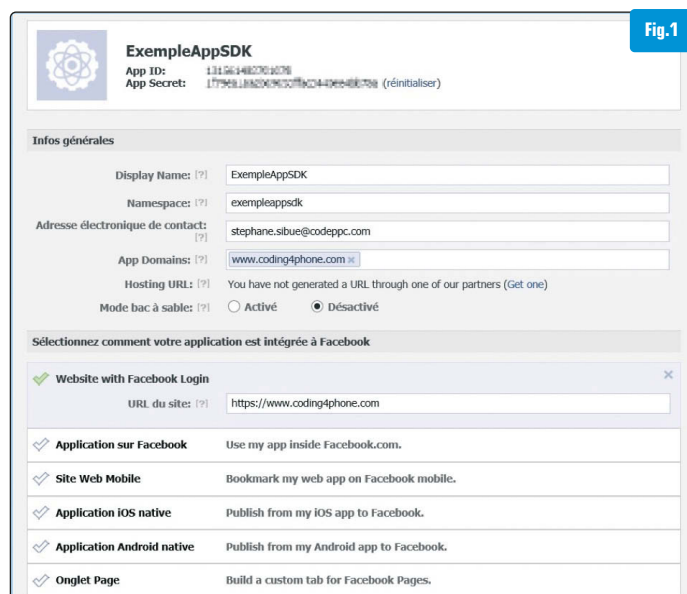
APPLICATION FACEBOOK

Pour être en mesure d'utiliser le SDK il est impératif de créer une application Facebook. Elle va vous donner les identifiants nécessaires pour la connexion à Facebook et la possibilité d'utiliser ses fonctionnalités. Tout se passe à l'adresse <https://developers.facebook.com>

Par défaut, Facebook vous propose de créer une application pour iOS ou Android, rien pour Windows Phone (ça va venir, du moins je l'espère). Pas d'inquiétude, il suffit de choisir une application Web, elle fera très bien l'affaire.

Une fois que vous aurez créé votre application, vous devrez passer en modification pour obtenir un écran de ce genre (attention au savant mélange entre français et anglais dans la page) : [Fig.1].

Le point important est la zone « App Domains ». Vous devez y placer un domaine sans le préfixe http://. Il faut bien choisir « Website with Facebook Login » dans la partie « Sélectionnez comment votre application est intégrée à Facebook »



ExempleAppSDK
 App ID: 131456482761026
 App Secret: 17796818A20X9C37BAC0430E54E708 (réinitialiser)

Infos générales

Display Name: [?] ExempleAppSDK
 Namespaces: [?] exempleappsdk
 Adresse électronique de contact: [?] stephane.sibue@codeppc.com
 App Domains: [?] www.coding4phone.com
 Hosting URL: [?] You have not generated a URL through one of our partners (Get one)
 Mode bac à sable: [?] ☐ Activé ☒ Désactivé

Sélectionnez comment votre application est intégrée à Facebook

☒ Website with Facebook Login
 URL du site: [?] https://www.coding4phone.com

☒ Application sur Facebook Use my app inside Facebook.com.
☒ Site Web Mobile Bookmark my web app on Facebook mobile.
☒ Application iOS native Publish from my iOS app to Facebook.
☒ Application Android native Publish from my Android app to Facebook.
☒ Onglet Page Build a custom tab for Facebook Pages.

Fig.1

application est intégrée à Facebook » avec une url de site commençant par https://

Même si votre site web n'est pas configuré avec un véritable certificat SSL (pour le https://) ce n'est pas grave, on saisit cette information juste pour faire plaisir à Facebook, mais elle ne servira pas. Ce qui est extrêmement important au final, c'est l'App ID, qui est le sésame pour accéder à Facebook.

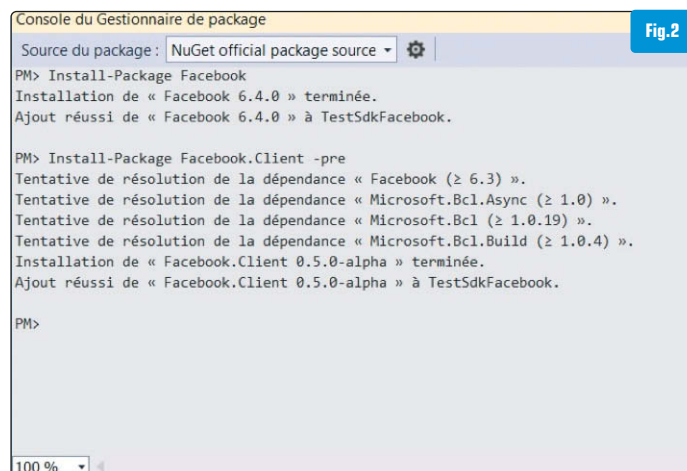
INSTALLATION

Maintenant que vous avez du côté de Facebook une application pour vous y accrocher, vous devez installer le SDK pour être en mesure de l'inclure à votre projet. L'installation se fait en utilisant NuGet. Cela simplifie beaucoup les choses. Il vous suffit de créer un nouveau projet Windows Phone 8 avec Visual Studio 2012. Une fois le projet créé, il faut ouvrir la console de gestion NuGet (Menu Outils > Gestionnaire de package de bibliothèque > Console du Gestionnaire de packages). La première commande à taper est « Install-Package Facebook ». Une fois la commande correctement exécutée (elle installe le package Facebook en version 6.4.0, du moins au moment où j'écris ces lignes), vous devez taper une seconde commande « Install-Package Facebook.Client -pre » qui installe le client Facebook en version 0.5.0-alpha. Au final, vous devriez obtenir ce qui suit : [Fig.2]. Dans mon exemple, le projet s'appelle « TestSdkFacebook » (très original me direz-vous), on peut voir que de nouvelles références ont fait leur apparition (Facebook et Facebook.Client) : [Fig.3].

AUTHENTIFICATION

L'étape incontournable est de s'authentifier sur Facebook. Sans cela rien ne sera possible. Cette opération permet de récupérer un token qui sera la clé pour utiliser toutes les fonctions.

Pour simplifier les choses, nous allons regrouper tout ce qui est



Console du Gestionnaire de package

Source du package : NuGet official package source

```
PM> Install-Package Facebook
Installation de « Facebook 6.4.0 » terminée.
Ajout réussi de « Facebook 6.4.0 » à TestSdkFacebook.

PM> Install-Package Facebook.Client -pre
Tentative de résolution de la dépendance « Facebook (≥ 6.3) ».
Tentative de résolution de la dépendance « Microsoft.Bcl.Async (≥ 1.0) ».
Tentative de résolution de la dépendance « Microsoft.Bcl (≥ 1.0.19) ».
Tentative de résolution de la dépendance « Microsoft.Bcl.Build (≥ 1.0.4) ».
Installation de « Facebook.Client 0.5.0-alpha » terminée.
Ajout réussi de « Facebook.Client 0.5.0-alpha » à TestSdkFacebook.

PM>
```

Fig.2

nécessaire à l'utilisation de Facebook dans une classe « FacebookManager » composée de différents membres statiques :

```
public class FacebookManager
{
    private static readonly String FacebookAppId = «VOTRE APPID ICI»;

    internal static String AccessToken;
    internal static String FacebookId;

    public static bool IsAuthenticated;
    public static FacebookSessionClient SessionClient = new FacebookSessionClient(FacebookAppId);
}
```

Cette classe gardera en mémoire (de manière privée) le token d'accès et l'ID Facebook de l'utilisateur identifié. Le booléen « IsAuthenticated » permettra de savoir rapidement si l'authentification a déjà été réalisée ou non. Dans notre exemple, l'authentification est lancée à la demande via un simple bouton : [Fig.4].

Et dont voici le code exécuté lors d'un click :

```
async private void BouConnexion_Click(object sender, RoutedEventArgs e)
{
    if (!FacebookManager.IsAuthenticated)
    {
        await AuthAsync();
    }
    else
    {
        this.NavigationService.Navigate(new Uri("/Pages/UserInfoPage.xaml", UriKind.Relative));
    }
}
```

Si l'authentification n'a jamais été réalisée, on l'effectue (par l'appel à la fonction asynchrone « AuthAsync »), sinon on navigue vers la page « UserInfoPage.xaml » qui sera chargée d'afficher les informations

concernant l'utilisateur fraîchement authentifié. Le code d'authentification est très simple. On utilise la méthode asynchrone « LoginAsync » de la session Facebook fournie par notre classe « FacebookManager » avec en paramètre les droits d'accès souhaités. Les droits demandés seront affichés en clair à l'utilisateur qui sera ainsi en mesure de refuser d'aller plus loin s'il le souhaite. Dans cet exemple nous demandons les droits « user_about_me » et « read_stream » (je vous laisse voir la documentation développeur de Facebook pour plus d'informations sur les droits d'accès). Si l'authentification va jusqu'au bout, on récupère le token d'accès et l'ID Facebook de l'utilisateur.

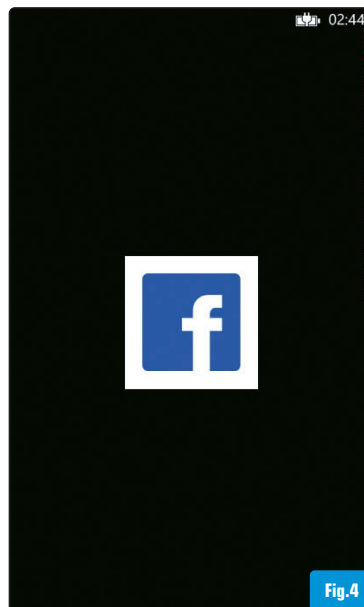
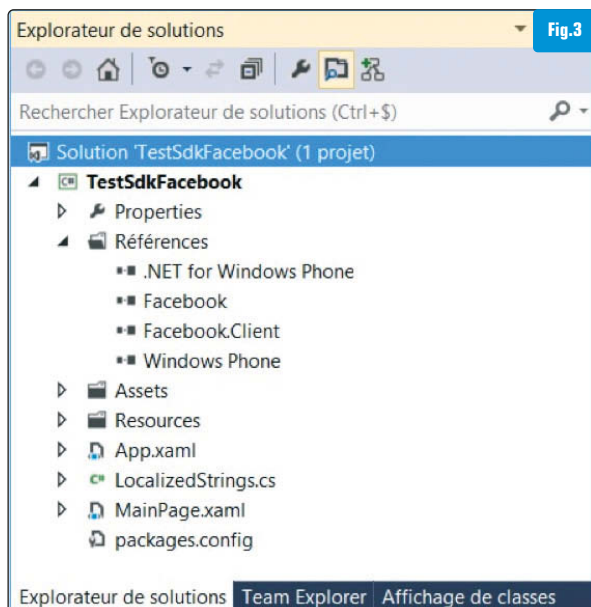
```
private async Task AuthAsync()
{
    try
    {
        var wFacebookSession = await FacebookManager.SessionClient.LoginAsync(«user_about_me,read_stream»);
        FacebookManager.AccessToken = wFacebookSession.AccessToken;
        FacebookManager.FacebookId = wFacebookSession.FacebookId;
        FacebookManager.IsAuthenticated = true;

        this.NavigationService.Navigate(new Uri("/Pages/UserInfoPage.xaml", UriKind.Relative));
    }
    catch (Exception ex)
    {
        String wMessage = «Login impossible !\n\n» + ex.Message;
        MessageBox.Show(wMessage);
    }
}
```

Lors de l'authentification, un écran 100% Facebook est affiché dans lequel il faut saisir le login et le mot de passe Facebook de l'utilisateur : [Fig.5].

RÉCUPÉRATION DES INFORMATIONS UTILISATEUR

Grâce à l'ID Facebook récupéré pendant l'authentification il est possible de récupérer les informations de l'utilisateur connecté. Dans



notre exemple c'est la page « UserInfoPage.xaml » qui s'en charge. Elle est appelée automatiquement dès que l'authentification est réussie. Elle stocke les informations utilisateur dans un objet maison de type « UserProfile » contenant le nom et la photo. Cette classe hérite de « BaseViewModel » qui fournit les outils pour utiliser au mieux le binding Silverlight (voir les sources complètes de cet article pour plus d'infos) :

```
public class UserProfile : BaseViewModel
{
    private String pName;
    private BitmapImage pPicture;

    public String Name
    {
        get { return pName; }

        set
        {
            pName = value;
            NotifyPropertyChanged(«Name»);
        }
    }

    public BitmapImage Picture
    {
        get { return pPicture; }

        set
        {
            pPicture = value;
            NotifyPropertyChanged(«Picture»);
        }
    }
}
```

Pour simplifier les choses, le profil est stocké en statique dans la classe « FacebookManager ». Si le profil n'existe pas encore à l'ouverture de la page, il est créé pour y être affiché. Cette opération est réalisée dans la méthode « OnNavigatedTo » de la page. Le profil ainsi créé devient le DataContext de la page, ce qui permet d'en afficher les informations en utilisant les fonctions de binding Silverlight :

```
async protected override void OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedTo(e);

    if (FacebookManager.UserProfile == null)
    {
        await GetUserProfileAsync();
    }
    else
    {
        this.DataContext = FacebookManager.UserProfile;
    }
}
```

Le code de récupération des informations utilise la classe « FacebookClient » fournie par le SDK et initialisé avec le token d'accès. Cette classe s'occupe de la communication avec Facebook et de la récupération de la réponse. Tout cela de manière asynchrone :

```
async private Task GetUserProfileAsync()
{
    var wFacebook = new FacebookClient(FacebookManager.AccessToken);
    var wResult = await wFacebook.GetTaskAsync(«me») as IDictionary<String, Object>;

    if (wResult != null)
    {
        // Formats de l'image
        // square (50x50)
        // small (50x?)
        // normal (100x?)
        // large (200x?)

        String wPrenom = wResult[«first_name»] as String;
        String wNom = wResult[«last_name»] as String;
        String wPictureUrl = String.Format(«https://graph.facebook.com/{0}/picture?type={1}&access_token={2}», FacebookManager.FacebookId, «large», FacebookManager.AccessToken);

        var wUserProfile = new UserProfile();
        wUserProfile.Name = String.Format(«{0} {1}», wPrenom, wNom);
        wUserProfile.Picture = new BitmapImage(new Uri(wPictureUrl));

        FacebookManager.UserProfile = wUserProfile;

        this.DataContext = FacebookManager.UserProfile;
    }
}
```

Les informations sont retournées sous la forme d'un dictionnaire dont les clés sont les noms des champs Facebook en face de chaque valeur.

Pour connaître le format du dictionnaire de retour, il est possible d'utiliser l'outil « Graph API Explorer » fourni par Facebook pour les développeurs, permettant d'explorer le format des réponses JSON, et donc d'en déduire le format des données retournées : [Fig.6].

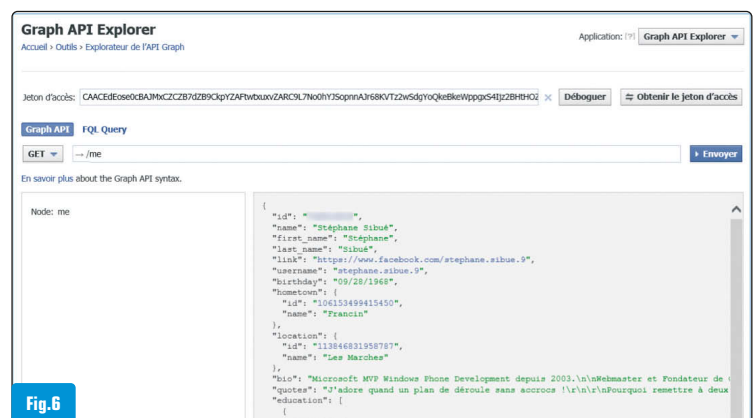


Fig.6

Une fois les informations utilisateur récupérées on peut les afficher. Dans notre exemple la page sert aussi de menu pour tester quelques autres fonctions : [Fig.7].

LISTER LES AMIS

Nous allons maintenant lister les amis de l'utilisateur connecté. Le principe est exactement le même que pour récupérer les informations utilisateur, le SDK va s'occuper de toute la partie communication avec Facebook. Les informations sur chaque ami seront stockées dans un objet maison de type « Friend » dont voici la définition :

```
public class Friend
{
    public String Id { get; set; }
    public String Name { get; set; }
    public BitmapImage Picture { get; set; }
}
```

On va donc stocker l'ID, le nom et l'image de tous les amis de l'utilisateur connecté. La collection des amis sera, elle aussi, ajoutée comme membre statique dans la classe « FacebookManager ». Dans notre exemple c'est la page « FriendsPage.xaml » qui va s'en occuper et cette collection d'amis sera la source de données de la liste incluse dans cette page spécialisée. Au chargement de la page, si la liste des amis n'existe pas encore, elle est créée :

```
async protected override void OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedTo(e);

    if (FacebookManager.UserFriends == null)
    {
        await GetUserFriendsAsync();
    }
    else
    {
        LstFriends.ItemsSource = FacebookManager.UserFriends;
    }
}
```

La fonction asynchrone « GetUserFriendsAsync » fonctionne sur le même principe que celle qui a permis de récupérer les informations utilisateur, sauf que là on demande « /me/friends » :

```
async private Task GetUserFriendsAsync()
{
    var wFacebook = new FacebookClient(FacebookManager.AccessToken);

    var wResult = await wFacebook.GetTaskAsync("/me/friends") as IDictionary<String, Object>;

    if (wResult != null)
    {
        var wData = wResult["data"] as IEnumerable<Object>;
        var wFriends = new ObservableCollection<Friend>();

        foreach (var wItem in wData)
        {
            var wDicoFriend = wItem as IDictionary<String, Object>;

            String wId = wDicoFriend["id"] as String;
            String wName = wDicoFriend["name"] as String;
            String wPictureUrl = String.Format("https://graph.facebook.com/{0}/picture?type={1}&access_token={2}", wId, "square", FacebookManager.AccessToken);

            var wFriend = new Friend();
            wFriend.Id = wId;
            wFriend.Name = wName;
            wFriend.Picture = new BitmapImage(new Uri(wPictureUrl));

            wFriends.Add(wFriend);
        }

        FacebookManager.UserFriends = wFriends;
        LstFriends.ItemsSource = FacebookManager.UserFriends;
    }
}
```

Là aussi il faut connaître le format du dictionnaire de retour. Le dictionnaire retourné est composé d'un seul élément portant le nom

« data ». Cet élément est un tableau JSON, donc une collection d'éléments. Chaque élément est un ami qui prend la forme d'un dictionnaire contenant les valeurs de ce champs « Id » et « Name ». Ce n'est pas facile au début mais avec un peu d'entraînement et avec l'aide de l'outil « Graph API Explorer » on y arrive assez bien.

Au final, on obtient la liste des amis (j'en profite pour saluer mes amis au passage) : [Fig.8].

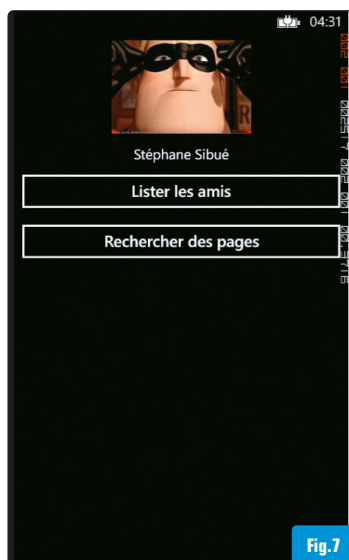


Fig.7

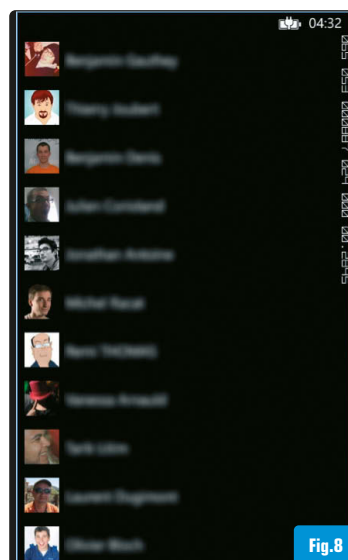


Fig.8



Fig.9

RECHERCHER DES PAGES

Autre chose que l'on peut faire très facilement c'est la recherche. Dans notre exemple nous allons créer une page permettant de rechercher des pages Facebook à partir de leur nom. Les informations collectées seront stockées dans un objet maison de type « FacebookPage » dont voici la définition :

```
public class FacebookPage
{
    public string Id { get; set; }
    public string Name { get; set; }
    public string Category { get; set; }
    public BitmapImage Picture { get; set; }
}
```

Comme pour les amis, les pages trouvées seront placées dans une collection dans la classe « FacebookManager ». Cette nouvelle collection sera source de données pour la liste qui affichera le résultat de la recherche.

Au final on aura un écran avec une zone pour saisir la page recherchée, un bouton permettant de lancer la recherche et la liste résultat. On utilisera aussi un « Progress Indicator » pour montrer qu'une recherche est en cours [Fig.9].

Voici le code appelé sur le click du bouton de recherche :

```
async private void BouRechercher_Click(object sender, RoutedEventArgs e)
{
    if (!string.IsNullOrEmpty(TxtRecherche.Text))
    {
        IsEnabled = false;

        pProgressIndicator.Text = «Recherche en cours...»;
        pProgressIndicator.IsVisible = true;

        await GetPagesAsync(TxtRecherche.Text);
    }
}
```

Pendant la recherche, la page est désactivée et un indicateur de progression est affiché avec le message « Recherche en cours... ». La recherche proprement dite ne diffère pas dans sa structure des autres appels que nous avons déjà vus ensemble. La structure est toujours la même :

```
async private Task GetPagesAsync(string wPageName)
{
    var wQuery = string.Format("/search?q={0}&type=page", wPageName);
    var wFacebook = new FacebookClient(FacebookManager.AccessToken);
    var wPages = new ObservableCollection<FacebookPage>();

    try
    {
        var wResult = await wFacebook.GetTaskAsync(wQuery) as IDictionary<string, Object>;

        if (wResult != null)
```

```
{
    var wData = wResult[«data»] as IEnumerable<Object>;

    foreach (var wItem in wData)
    {
        var wDicoPages = wItem as IDictionary<string, Object>;

        string wId = wDicoPages[«id»] as string;
        string wName = wDicoPages[«name»] as string;
        string wCategorie = wDicoPages[«category»] as string;
        string wPictureUrl = string.Format(«https://graph.facebook.com/{0}/picture?type={1}&access_token={2}», wId, «normal», FacebookManager.AccessToken);

        var wPage = new FacebookPage();
        wPage.Id = wId;
        wPage.Name = wName;
        wPage.Category = wCategorie;
        wPage.Picture = new BitmapImage(new Uri(wPictureUrl));

        wPages.Add(wPage);
    }
}
catch (Exception ex)
{
    TxtRecherche.Text = «»;
    MessageBox.Show(«Recherche impossible !»);
}

FacebookManager.SearchResult = wPages;
LstPages.ItemsSource = wPages;

pProgressIndicator.IsVisible = false;
IsEnabled = true;
}
```

Là encore, l'outil « Graph API Explorer » de Facebook s'avère indispensable pour connaître la structure des données retournées. Les recherches possibles sont très vastes, vous pouvez par exemple rechercher tous les restaurants autour de vous dans un rayon de 300 mètres.

CONCLUSION

Le SDK .NET Facebook simplifie énormément l'authentification des utilisateurs et la communication avec le service lui-même. Il est en revanche toujours aussi important de connaître l'API Facebook pour être en mesure d'en tirer le maximum et de savoir quelles informations vont être retournées. Les fonctionnalités asynchrones (async / await) sont un véritable délice et rendent l'utilisation de ce SDK très sympathique. Les sources du projet d'exemple de cet article sont disponibles en téléchargement. N'hésitez pas à les étudier et profitez-en pour ajouter d'autres fonctions afin de vous entraîner.

Stéphane Sibué
CTO chez SOFTELITE
stephane.sibue@softelite.fr
Développeur iOS, Android et Windows Phone
Blog : www.coding4phone.com

Flume NG : exploiter les logs !

Dans tous les projets informatiques, la problématique de logs est récurrente : stratégie, format, volume, etc... En cas de problèmes, ces derniers représentent un des premiers points d'entrée dans une application : actions réalisées par un utilisateur, erreurs affectant un utilisateur.

Cependant, soumises à de fortes volumétries, la collecte et l'exploitation des logs peut vite devenir difficile, voire impossible. Comment alors exploiter cette mine d'informations qui dort dans tous ces fichiers ? C'est pour répondre à cette problématique que Flume NG a été créé.

> Qu'est-ce que Flume NG ?

Flume NG (Next Generation) est un projet Apache et une solution de collecte et d'agrégation de gros volumes d'événements. Ce dernier permet notamment de pousser ses logs dans un référentiel centralisé (Filesystem, HDFS, HBase, etc.). Il a été également pensé pour être scalable. Flume NG est une refonte complète de Flume OG (Old Generation) offrant des fonctionnalités plus avancées et une utilisation simplifiée. Flume est composé d'agents possédant les caractéristiques suivantes : [Fig.1]

> Comment fonctionne un agent ?

- Une *source* consomme des événements provenant de sources externes (syslog, autre agent Flume, etc.) et place ces derniers dans un ou plusieurs channel(s).
- Un *channel* (base de données, en mémoire, etc.) permet de stocker temporairement les événements attendant d'être consommés.
- Un *sink* va ensuite pousser ces événements vers la destination souhaitée (HDFS, Flume agent source, etc.)

Voici une liste non exhaustive de possibilités de configuration des agents :

- Sources :
 - Avro : écouter sur un port TCP. Utilisé également pour communiquer entre différents agents Flume
 - Spooling Directory Source : lire des fichiers arrivant dans un répertoire
 - Syslog (TCP ou UDP): capter les événements d'un server Syslog. Il est par exemple possible de configurer Tomcat pour envoyer les logs vers Syslog et Flume les prendra en charge.
 - HTTP: un handler se charge de traduire les requêtes POST et GET (encore en phase d'expérimentation).
- Channels :
 - Memory : garde les événements en mémoire
 - JDBC : les événements sont stockés en base de données

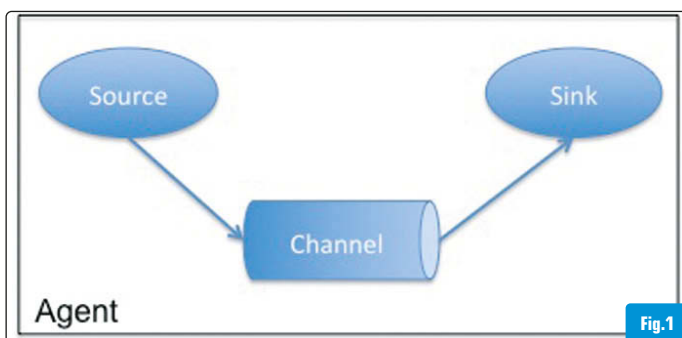


Fig.1

- File : les événements sont stockés sur un filesystem.
- Sinks :
 - HDFS : pousse les événements dans Hadoop
 - Avro : permet de communiquer avec un autre agent
 - IRC : pousse les événements sur un channel IRC
 - HBase : pousse les événements dans une base HBase
 - ElasticSearch : pousse les événements dans une base ElasticSearch
 - File : enregistre les événements sur un fichier local.

Il est également possible de chaîner plusieurs agents Flume.

> Installation et configuration

L'installation de Flume est simple :

- Avoir un JDK installé
- Dézipper Flume et le mettre dans un répertoire de votre choix (ex : /usr/local/flume)
- Avoir installé Hadoop si vous souhaitez pousser les logs dans un HDFS (ex : /usr/local/hadoop), ou, a minima, posséder en local le jar hadoop-core.

Voici un exemple de configuration avec un agent Flume écoutant un Syslog pour mettre les logs dans un HDFS (PATH_FLUME/conf/flume.conf) : [Fig.3 et 4].

Ensuite pour démarrer votre agent vous devez saisir la ligne de commande suivante au niveau de votre répertoire d'installation (PATH_FLUME) :

```
flume-ng agent -conf ./conf -f conf/flume.conf -n agent
```

Votre agent Flume écoute les événements sur le port 5140 et les pousse dans HDFS.

> Customiser Flume

Flume offre la possibilité d'implémenter nos propres sources, channels et sinks à partir d'interfaces Java. Il suffit ensuite de mettre les implémentations dans le classpath de Flume et de l'utiliser dans le fichier de configuration : [Fig.5].

> Autres composants

Flume permet d'ajouter de l'intelligence aux agents grâce aux composants suivants :

- *Interceptors* : ils peuvent modifier, filtrer les événements en ajoutant des en-têtes (timestamp, hostname, etc) : [Fig.6].

Dans cet exemple, notre intercepteur va ajouter dans l'en-tête une clé hostname et la valeur qui est l'adresse IP de la machine courante. Dans une configuration complexe de Flume (plusieurs machines), cet intercepteur est très pratique pour savoir par quelles machines les logs ont été traités.



Fig.3

- *Channel selectors* : permet d'aiguiller les événements vers des channels spécifiques : [Fig.7].

Notre source analyse le header et si la clé est hostname et que sa valeur est host1, les événements seront alors envoyés vers les channels 2 et 4. Lorsque le sélecteur n'arrive pas à trouver de règle correspondante, les événements sont envoyés vers le channel par défaut (channel1).

- *Sink processors* : permet de définir un groupe de sinks et de définir des stratégies de load balancing ou failover : [Fig.8].

Dans cet exemple, les deux sinks déclarés recevront des événements chacun leur tour (round_robin).

> Reporting

Il est possible de pousser des métriques Flume vers Ganglia. Il suffit d'ajouter les options suivantes au démarrage de vos agents :

```
-Dflume.monitoring.type=ganglia -Dflume.monitoring.hosts=host
name1:port1, hostname2:port2
```

Les métriques sont également accessibles au format JSON :

```
{
  "typeName1.componentName1" : {"metric1" : "metricValue1",
  "metric2" : "metricValue2"}
  "typeName2.componentName2" : {"metric3" : "metricValue3",
  "metric4" : "metricValue4"}
}
```

Au démarrage il suffit d'ajouter les paramètres suivants :

```
-Dflume.monitoring.type=http -Dflume.monitoring.port=port
```

```
agent.sources = syslog
agent.channels = memoryChannel
agent.channels.memoryChannel.type = memory
agent.channels.memoryChannel.capacity = 10000

agent.sources.syslog.type = syslogtcp
agent.sources.syslog.port = 5140
agent.sources.syslog.host = 127.0.0.1
agent.sources.syslog.channels = memoryChannel

agent.sinks = HDFSEventSink
agent.sinks.HDFSEventSink.channel = memoryChannel
agent.sinks.HDFSEventSink.type = hdfs
agent.sinks.HDFSEventSink.hdfs.path = hdfs://PATH_TO_YOUR_HDFS
```

Fig.4

```
agent.sources = mySource
agent.channels = memoryChannel
agent.sources.mySource.type = com.xebia.MyCustomSource
agent.sources.mySource.channels = memoryChannel
```

Fig.5

```
agent.sources = mySource
agent.channels = memoryChannel
agent.sources.mySource.interceptors = myInterceptor
agent.sources.mySource.interceptors.myInterceptor.type = host
agent.sources.mySource.interceptors.myInterceptor.hostHeader = hostname
```

Fig.6

```
agent.sources = mySource
agent.channels = channel1, channel2, channel3, channel4
agent.sources.mySource.selector.type = multiplexing
agent.sources.mySource.selector.header = hostname
agent.sources.mySource.selector.mapping.host1 = channel2, channel4
agent.sources.mySource.selector.mapping.host2 = channel3
agent.sources.mySource.selector.mapping.default = channel1
```

Fig.7

```
agent.sinkgroups = myGroup
agent.sinkgroups.myGroup.sinks = sink1, sink2
agent.sinkgroups.myGroup.processor.type = load_balance
agent.sinkgroups.myGroup.processor.selector = round_robin
```

Fig.8

> Exemple de configuration plus complexe

Les host 1 et 2 reçoivent des événements de Syslog et doivent les transmettre à des machines distantes (host 3 et 4) qui se chargent de les pousser dans un HDFS.

Afin d'ajouter de la scalabilité, nous souhaitons mettre en place un load balancing entre les host 3 et 4.

Voici les configurations nécessaires :

Host 1 et Host 2 : [Fig.10].

Host 3 et Host 4 : [Fig.11].

Avec cette configuration, nous voyons que la mise en place de ce type d'architecture devient triviale.

Pour aller plus loin, il est possible d'ajouter des intercepteurs ou bien d'aiguiller nos logs avec des Channels selectors vers les host 3 ou 4 selon des règles définies (IP de la machine par exemple).

> Conclusion

Flume NG propose une solution de collecte de logs performante. De par la richesse des composants proposés (*sources, channels et sinks*), Flume permet des interactions avec une multitude de technologies (HDFS, HBase, ElasticSearch, etc). Ajouté à sa simplicité de mise en place et ses possibilités de scalabilité, Flume se place comme un outil incontournable.

<http://flume.apache.org/>

<http://flume.apache.org/FlumeUserGuide.html>

<http://ganglia.sourceforge.net>

Nicolas Jozwiak, Développeur et agiliste chez Xebia.

```
agent.sources = syslog
agent.channels = memoryChannel
agent.channels.memoryChannel.type = memory
agent.channels.memoryChannel.capacity = 10000
```

Fig.10

```
agent.sources.syslog.type = syslogtcp
agent.sources.syslog.port = 5140
agent.sources.syslog.host = 127.0.0.1
agent.sources.syslog.channels = memoryChannel
```

```
agent.sinkgroups = myGroup
agent.sinkgroups.myGroup.sinks = avroSink1, avroSink2
agent.sinkgroups.myGroup.processor.type = load_balance
agent.sinkgroups.myGroup.processor.selector = round_robin
```

```
agent.sinks = avroSink1, avroSink2
agent.sinks.avroSink1.channel = memoryChannel
agent.sinks.avroSink1.type = avro
agent.sinks.avroSink1.hostname = host3
agent.sinks.avroSink1.port = 5141
```

```
agent.sinks.avroSink2.channel = memoryChannel
agent.sinks.avroSink2.type = avro
agent.sinks.avroSink2.hostname = host4
agent.sinks.avroSink2.port = 5142
```

```
agent.sources = avroCollection1, avroCollection2
agent.channels = memoryChannel
agent.channels.memoryChannel.type = memory
agent.channels.memoryChannel.capacity = 10000
```

Fig.11

```
agent.sources.avroCollection1.channels = memoryChannel
agent.sources.avroCollection1.type = avro
agent.sources.avroCollection1.bind = 127.0.0.1
agent.sources.avroCollection1.port = 5141
```

```
agent.sources.avroCollection2.channels = memoryChannel
agent.sources.avroCollection2.type = avro
agent.sources.avroCollection2.bind = 127.0.0.1
agent.sources.avroCollection2.port = 5142
```

```
agent.sinks = HDFSEventSink
agent.sinks.HDFSEventSink.channel = memoryChannel
agent.sinks.HDFSEventSink.type = hdfs
agent.sinks.HDFSEventSink.hdfs.path = hdfs://PATH_TO_YOUR_HDFS
```

Initialiser un nouveau projet Symfony2

Cet article, d'une série de trois, est une première introduction à Symfony2. Ce dernier est un framework PHP « full-stack » construit autour d'une vingtaine de composants autonomes. Sous licence MIT, il est entièrement Open Source. Il vous faut disposer, au minimum, de PHP 5.3 pour l'utiliser. Dans cet article, nous verrons qu'il est très simple de faire sa première page web, allons-y !

> Prérequis

Veillez à avoir un environnement de développement avec PHP (5.3+) : pour les utilisateurs de Windows, je vous invite à installer l'outil WAMP (<http://www.wampserver.com/>) et suivre les instructions pour avoir un environnement de développement web simple à utiliser.

NB : Pensez à activer le « `rewrite_module` » avec WAMP.

Par ailleurs, pour avoir accès à PHP dans l'invite de commande, il vous faut ajouter dans la variable système « PATH », le chemin du PHP de WAMP (consultez le tutoriel suivant : <http://bit.ly/17nHYxa>).

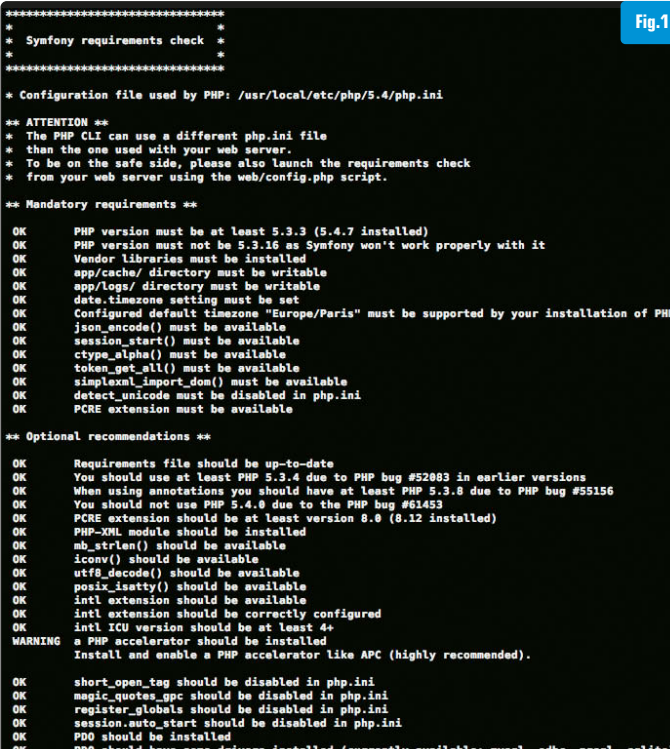
> Téléchargement

Commençons par le téléchargement et l'installation du framework. Deux possibilités :

- téléchargez l'archive sur le site officiel www.symfony.com/download, avec vendors, et décompressez-le dans votre dossier web.
- ou utilisez la ligne de commande avec **Composer** (`getcomposer.org/download/`)

```
php composer.phar create-project symfony/framework-standard-edition MyProject/ 2.3
```

Il vous faut déposer le dossier Symfony2 téléchargé dans votre répertoire web. Symfony2 est maintenant « installé ». Dans votre console (ou invite de commande), déplacez-vous dans votre projet : vérifiez que les prérequis pour Symfony2 sont bien respectés en exécutant le



```
*****
*                               *
*   Symfony requirements check   *
*                               *
*****

* Configuration file used by PHP: /usr/local/etc/php/5.4/php.ini

** ATTENTION **
* The PHP CLI can use a different php.ini file
* than the one used by your web server.
* To be on the safe side, please also launch the requirements check
* from your web server using the web/config.php script.

** Mandatory requirements **
OK      PHP version must be at least 5.3.3 (5.4.7 installed)
OK      PHP version must not be 5.3.16 as Symfony won't work properly with it
OK      Vendor libraries must be installed
OK      app/cache/ directory must be writable
OK      app/logs/ directory must be writable
OK      date.timezone setting must be set
OK      Configured default timezone "Europe/Paris" must be supported by your installation of PHP
OK      json_encode() must be available
OK      session_start() must be available
OK      ctype_alpha() must be available
OK      token_get_all() must be available
OK      simplexml_import_dom() must be available
OK      detect_unicode must be disabled in php.ini
OK      PCRE extension must be available

** Optional recommendations **
OK      Requirements file should be up-to-date
OK      You should use at least PHP 5.3.4 due to PHP bug #52883 in earlier versions
OK      When using annotations you should have at least PHP 5.3.8 due to PHP bug #55156
OK      You should not use PHP 5.4.0 due to the PHP bug #61483
OK      PCRE extension should be at least version 8.0 (8.12 installed)
OK      PHP-XML module should be installed
OK      mb_strlen() should be available
OK      iconv() should be available
OK      utf8_decode() should be available
OK      posix_getpwuid() should be available
OK      intl extension should be available
OK      intl extension should be correctly configured
OK      intl ICU version should be at least 4+
WARNING A PHP accelerator should be installed
        Install and enable a PHP accelerator like APC (highly recommended).

OK      short_open_tag should be disabled in php.ini
OK      magic_quotes_gpc should be disabled in php.ini
OK      register_globals should be disabled in php.ini
OK      session.auto_start should be disabled in php.ini
OK      PDO should be installed
OK      PDO should have some drivers installed (currently available: mysql, odbc, pgsql, sqlite)
```

Fig.1

script `php app/check.php` [Fig.1]. Pour lister les commandes offertes par le framework, tapez dans votre console la commande `php app/console`. Votre projet est amorcé avec les dossiers suivants :

- **app** - configuration de l'application
- **src** - le code de votre application
- **vendor** - les librairies tierces, dont Symfony
- **web** - ce qui est accessible par le navigateur

> Premier affichage dans le navigateur

Un bundle de démonstration est livré avec la distribution *standard* de Symfony2. Si vous disposez de PHP 5.4, il est possible de lancer un serveur web avec la commande `php app/console server:run` [Fig.2].

Vérifiez que votre projet est bien amorcé en affichant la page **localhost:8000/app_dev.php**. Pour les utilisateurs de WAMP, votre première page est accessible à l'adresse **localhost/nom_de_votre_dossier/web/app_dev.php** [Fig.3].

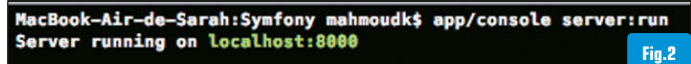
DÉVELOPPER SA PROPRE APPLICATION SYMFONY2

> Premier bundle

Tout le code de votre application est organisé en bundle(s). Un bundle est un ensemble de fichiers structurés correspondant à une fonctionnalité. Le but est de pouvoir utiliser un bundle d'une application à une autre simplement. Nous allons utiliser le bundle **AcmeDemoBundle** livré avec le framework. Pour générer un bundle, tapez la commande `php app/console generate:bundle`. La convention de nommage est la suivante : nom de votre organisation suivi du nom de votre bundle (séparés par un *slash*). Tapez « entrée » à toutes les questions posées.

> Le bundle Acme

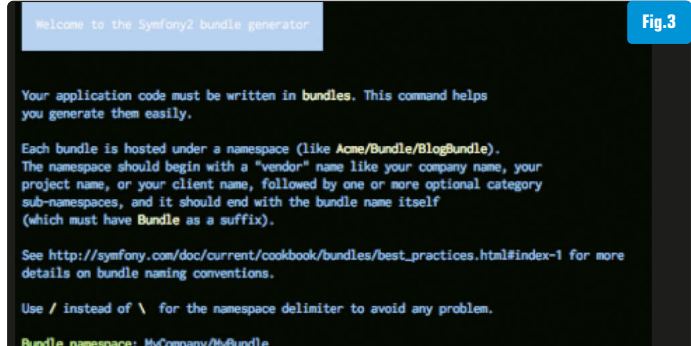
Le bundle livré avec le framework contient une application assez complète. Nous allons créer notre première page très facilement. Il faut d'abord créer une classe de contrôleur dans le dossier



```
MacBook-Air-de-Sarah:Symfony mahmoudk$ app/console server:run
Server running on localhost:8000
```

Fig.2

Serveur web interne



Welcome to the Symfony2 bundle generator

Your application code must be written in bundles. This command helps you generate them easily.

Each bundle is hosted under a namespace (like `Acme/Bundle/BlogBundle`). The namespace should begin with a "vendor" name like your company name, your project name, or your client name, followed by one or more optional category sub-namespaces, and it should end with the bundle name itself (which must have `Bundle` as a suffix).

See http://symfony.com/doc/current/cookbook/bundles/best_practices.html#index-1 for more details on bundle naming conventions.

Use `/` instead of `\` for the namespace delimiter to avoid any problem.

Bundle namespace: `MyCompany/MyBundle`

Fig.3

Génération d'un bundle

src/Acme/DemoBundle/Controller : créez le fichier **HelloWorldController.php** avec le code suivant :

```
<?php
namespace Acme\DemoBundle\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;

class HelloWorldController
{
    /**
     * @Route("/helloworld")
     * @Template()
     */
    public function indexAction()
    {
        return array();
    }
}
```

Créez un nouveau dossier du nom de la classe de contrôleur **HelloWorld** dans le dossier **src/Acme/DemoBundle/Resources/views/**, ainsi que votre template Twig du nom de la méthode de contrôleur (`indexAction`), **index.html.twig**.

Dans le template **index.html.twig**, saisissez le code suivant :

```
{% extends «AcmeDemoBundle::layout.html.twig» %}

{% block content %}
    Ma première page !
{% endblock %}
```

Et ajoutez les lignes suivantes dans le fichier **app/config/routing.yml** :

```
helloworld:
    resource: «@AcmeDemoBundle/Controller/HelloWorldController.php»
    type:      annotation
```

Grâce à ces lignes, le framework prend connaissance des routes, saisies en annotations, présentes dans votre contrôleur **HelloWorldController.php**. Enfin, affichez votre première page dans votre navigateur préféré. Voyons le code un peu plus dans le détail, depuis la requête à la réponse HTTP.

Le contrôleur frontal

Lorsqu'une url est appelée (**localhost:8000/app_dev.php/helloworld**) le contrôleur frontal **app_dev.php** (dans le dossier **web**) est l'unique point d'entrée sur l'application. Assurez-vous de configurer votre serveur afin d'accéder à votre contrôleur frontal.

NB : Pour votre environnement de production, il vous faut utiliser le contrôleur frontal **app.php** (fourni également dans le dossier **web**).

Le routage. Symfony2 offre un système de routage permettant d'associer un motif d'URL à une méthode de classe de contrôleur (ou action). Le format de configuration choisi, ici, est l'annotation : ainsi, au dessus de chaque contrôleur créé, il vous est possible d'ajouter des annotations de ce type pour personnaliser vos URL :

```
/**
 * @Route("/url/de/votre/choix")
 */
```

Pour plus d'informations sur le sujet, je vous invite à rentrer plus dans le détail avec le book disponible en ligne **symfony.com/doc/master/book/routing.html**.

Le contrôleur

Votre contrôleur utilise les annotations :

- **@Route** permettant d'associer le motif d'URL souhaité ;
- et **@Template** permettant de lier l'action à une vue.

Note : Si aucun chemin vers un template n'est indiqué, le framework choisit automatiquement le template en fonction du nom de classe de contrôleur et du nom de la méthode de contrôleur (action). Ainsi, le framework ira donc chercher un template nommé « **index.twig.html** » dans le dossier « **HelloWorld** ».

Ces deux annotations correspondent aux classes PHP définies au-dessus de la classe. Gardez à l'esprit que toute action doit retourner un objet réponse. Ici, l'annotation **@Template** permet de ne renvoyer qu'un tableau. La méthode **indexAction** est, comme vous pouvez vous en douter, celle qui est appelée par notre route : toutes les méthodes de contrôleur doivent être suffixées par « **Action** », c'est d'ailleurs pour cela que nous parlerons « d'action » les prochaines fois que nous évoquerons les méthodes de contrôleur. L'argument passé en paramètre de la méthode n'est autre que ce qui est reçu depuis l'url. Une méthode de contrôleur doit retourner une réponse. Ici, il s'agit du template twig appelé avec la méthode « **render** ». Cette méthode peut prendre un nom logique de template, et un tableau contenant les variables qui seront utilisées dedans. Il est donc indiqué que le template (la vue) twig rendu est situé dans le dossier **HelloWorld** et qu'il s'agit du fichier **index.html.twig**.

Pour plus d'informations au sujet des contrôleurs, je vous invite à rentrer plus dans le détail avec le book disponible en ligne **symfony.com/doc/master/book/controller.html**.

> La vue

Le code du template est simple : nous étendons le template **layout.html.twig**, situé dans le dossier **Resources/views** du **Acme-DemoBundle**, afin de redéfinir le bloc nommé « **content** » (qui y est déclaré). Dans notre exemple, nous l'avons rempli avec le contenu « **Ma première phrase** ». L'instruction **{% extends «AcmeDemoBundle::layout.html.twig» %}** suivante permet d'étendre le template de base « **layout.html.twig** » : en effet, si plusieurs de vos pages ont des éléments qui se répètent (Titre, menu, haut de page, bas de page...), il est possible de les centraliser dans un seul et même fichier que vous étendrez pour chaque nouvelle page. Dans notre cas, le fichier **index.twig.html** étant le fichier twig **layout.html.twig** : dans ce dernier, le « **block** » **content** est présent et son contenu peut-être remplacé (ce que nous faisons avec le texte « **Ma première page !** » dans le fichier **index.twig.html**).

Trois instructions de base sont à retenir pour l'instant :

- « **{ { ... } }** » permet l'affichage avec échappement automatique des variables (l'équivalent du **htmlspecialchars** en PHP) ;
- « **{% ... %}** » permet d'exécuter des instructions (par exemple les structures de contrôle **if**, **for**...) ;
- « **{# ... #}** » permet de commenter le code.

Je vous invite à consulter le site officiel <http://twig.sensiolabs.org/> pour plus de détails concernant le moteur de rendu Twig.

Rendez-vous le mois prochain pour aller plus loin avec Symfony2.

Sarah Khalil - Chef de projet technique
SensioLabs - sarah.khalil@sensiolabs.com

Prendre en main les capteurs sous Android en réalisant une boussole

La plupart des terminaux Android sont équipés de capteurs permettant de mesurer leur mouvement, leur orientation ou bien les conditions de leur environnement. Ainsi, ces capteurs embarqués permettent d'obtenir des données brutes combinant précision et fiabilité, offrant aux développeurs de la plateforme un grand nombre de cas d'utilisation pour leurs applications.

Les exemples d'utilisation sont légion et on peut citer un jeu tirant partie du capteur de gravité afin de réagir aux gestes et mouvements de l'utilisateur, ou bien encore une application Météo s'appuyant sur les données des capteurs de température et d'humidité. Enfin, on peut imaginer des applications désirant donner la direction d'un point particulier sur la Terre (applications de voyage ou bien religieuses) mettre à profit l'accéléromètre et le capteur de champ magnétique embarqués.

La plateforme Android regroupe les capteurs qu'elle supporte en 3 grandes catégories. Représentant la première catégorie, les détecteurs de mouvement permettent de mesurer les forces d'accélération et de rotation suivant 3 axes (x, y et z). Leur liste est la suivante :

- **TYPE_ACCELEROMETER** évaluant les mouvements du terminal autour des 3 axes incluant la force de gravité.
- **TYPE_GRAVITY** mesurant la force de gravité appliquée à un terminal sur les 3 axes.
- **TYPE_GYROSCOPE** mesurant la position angulaire du terminal sur chacun des 3 axes.
- **TYPE_LINEAR_ACCELERATION** donnant la force d'accélération suivant les 3 axes en excluant la force de gravité.
- **TYPE_ROTATION_VECTOR** évaluant l'orientation de l'appareil via les 3 composantes de son vecteur de rotation.

La seconde catégorie correspond aux capteurs des conditions environnementales au sein desquels on retrouve :

- **TYPE_AMBIENT_TEMPERATURE** indiquant la température de l'air à proximité du capteur.
- **TYPE_LIGHT** mesurant le niveau d'illumination ambiant.
- **TYPE_PRESSURE** retournant la pression de l'air ambiant.
- **TYPE_RELATIVE_HUMIDITY** mesurant l'humidité relative de l'air ambiant.
- **TYPE_TEMPERATURE** renvoyant la température du périphérique.

Enfin, la dernière catégorie est celle des capteurs de position donnant accès à la position physique d'un périphérique. Ils sont au nombre de 3 :

- **TYPE_MAGNETIC_FIELD** détectant les modifications des champs magnétiques environnants.
- **TYPE_ORIENTATION** mesurant la position du terminal en fonction des 3 axes, mais déprécié depuis Android 2.2.
- **TYPE_PROXIMITY** mesurant la distance du terminal par rapport à un objet.

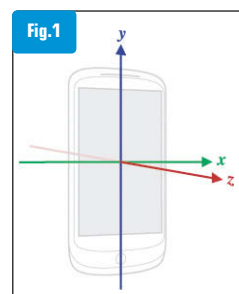
> Sensor Framework

Bien que le SDK propose un support pour un grand nombre de capteurs, il est bon de garder à l'esprit que leur utilisation dépendra

avant tout du terminal puisque certains terminaux ne possèdent pas quelques uns des capteurs listés précédemment. De fait, avant toute utilisation d'un capteur il sera nécessaire de vérifier si le terminal d'exécution en possède bien un.

Fort heureusement, le SDK Android met à disposition une API facilitant la manipulation des capteurs. Située au sein du package **android.hardware**, le Sensor Framework s'utilise au travers de 4 objets au centre desquels on retrouve la classe **SensorManager**. Celle-ci modélise le service système donnant accès aux capteurs embarqués d'un appareil et permet de fait de lister et d'accéder à ces derniers, tout en s'enregistrant aux événements qu'ils vont envoyer lorsque leurs données changent. En outre, elle propose la méthode **getOrientation()** qui renvoie la position du terminal en fonction des 3 axes et qui remplace l'utilisation du capteur **TYPE_ORIENTATION**. Les capteurs sont quant à eux modélisés à l'aide de la classe **Sensor** qui regroupe les constantes représentant l'ensemble des capteurs supportés par la plateforme. Elle permet d'accéder via diverses méthodes aux données techniques propres au capteur révélant leurs capacités. Cela va du vendeur au délai minimum de renouvellement entre deux événements envoyés, en passant par la consommation de la batterie induite par son utilisation. Les événements envoyés par un capteur contenant ses données brutes sont accessibles via la classe **SensorEvent** dont le système de coordonnées considéré est détaillé à la [Fig.1]. Il est bon de signaler qu'il reste identique lorsque le périphérique est en mode paysage.

Typiquement, un **SensorEvent** permet de connaître le capteur source via son objet **Sensor** ainsi que le moment auquel l'événement s'est produit, les données brutes mesurées et la précision de ces dernières. Enfin, on signalera l'interface **SensorEventListener** qui sera à implémenter par un objet souhaitant s'abonner aux mises à jour des capteurs d'un appareil. Pour ce faire, elle expose les méthodes **onAccuracyChanged** et **onSensorChanged** permettant de récupérer respectivement la précision et les données d'un capteur.



Système de coordonnées utilisé par SensorEvent

> Identification des capteurs

La première étape pour utiliser les capteurs consiste à identifier ceux qui sont disponibles sur le terminal ainsi que leurs capacités. En effet, pour un même type de capteur, plusieurs implémentations peuvent exister sur un même terminal, auquel cas il faudra choisir celui que l'on souhaite utiliser selon un certain nombre de critères pou-

vant être la précision ou bien la consommation d'énergie par exemple. Le code suivant permet de lister l'ensemble des capteurs disponibles sur un terminal :

```
SensorManager sensorManager = (SensorManager) getSystemService(
    Context.SENSOR_SERVICE);
List<Sensor> deviceSensors = sensorManager.getSensorList(Sensor.
    TYPE_ALL);
```

Il est également possible de vérifier si le terminal est équipé d'un capteur d'un type précis, gravité par exemple, puis de lister l'ensemble des capteurs de ce type afin de choisir celui qui répond à certaines caractéristiques. Ici, on récupère le capteur gravité fourni par Google en version 3 :

```
if (sensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY) != null) {
    List<Sensor> gravSensors = sensorManager.getSensorList(Sensor.
        TYPE_GRAVITY);
    Sensor gravSensor = null;

    for (int i = 0; i < gravSensors.size() && gravSensor == null;
        i++) {
        Sensor current = gravSensors.get(i);

        if (current.getVendor().contains("Google Inc.") && current.
            getVersion() == 3) {
            gravSensor = current;
        }
    }
} else {
    // ... échec pas de capteur gravité
}
```

> Monitoring des évènements

Les capteurs identifiés, il est maintenant nécessaire de s'abonner pour être informé de leurs changements et accéder ainsi aux nouvelles données les concernant. Une classe souhaitant écouter les évènements d'un capteur doit implémenter l'interface `SensorEventListener` et ses méthodes `onAccuracyChanged` et `onSensorChanged`. L'abonnement aux évènements d'un capteur se fait via la méthode `registerListener` du `SensorManager` en spécifiant le délai de rafraîchissement souhaité. Enfin, il ne faut surtout pas oublier de se désabonner auprès de ce dernier lorsque l'on ne souhaite plus être alerté des modifications d'un capteur afin de soulager la batterie. Pour mettre en pratique ceci, on récupère le capteur de champ magnétique par défaut puis on s'abonne pour être informé de ses variations :

```
public class MyCompassActivity extends Activity implements Sensor
    EventListener {
    private float geomagnetic[] = new float[3];
    private SensorManager sensorManager;
    private Sensor sensorMagnetic;

    @Override
    public void onStart() {
        super.onStart();
        sensorManager = (SensorManager) getSystemService(Context.
```

```
SENSOR_SERVICE);
        sensorMagnetic = sensorManager.getDefaultSensor(Sensor.TYPE_
            MAGNETIC_FIELD);
        sensorManager.registerListener(this, sensorMagnetic, Sensor
            Manager.SENSOR_DELAY_NORMAL);
    }

    @Override
    protected void onStop() {
        super.onStop();
        sensorManager.unregisterListener(this, sensorMagnetic);
    }

    public void onAccuracyChanged(Sensor sensor, int accuracy) {
        if (sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD && accuracy
            == SensorManager.SENSOR_STATUS_UNRELIABLE) {
            // ... Données non fiables
        }
    }

    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
            geomagnetic[0] = event.values[0];
            geomagnetic[1] = event.values[1];
            geomagnetic[2] = event.values[2];
        }
    }
}
```

> Réalisation d'une boussole

Les fondamentaux concernant les capteurs mis en place, il est temps de passer à un exemple plus poussé avec le développement d'une boussole analogique. Notre application devra également afficher de manière textuelle la direction vers laquelle le terminal pointe. Une solution élégante va consister à afficher l'image d'une boussole [Fig.2] au sein d'une vue spécifique sur laquelle nous appliquerons une rotation pour obtenir la direction du Nord souhaitée.

> Création d'une vue boussole

La création d'une vue boussole spécifique se fait de manière assez facile avec Android. Il suffit pour cela d'hériter de la classe `View` et de redéfinir la méthode `onDraw` au sein de laquelle l'image de la bous-

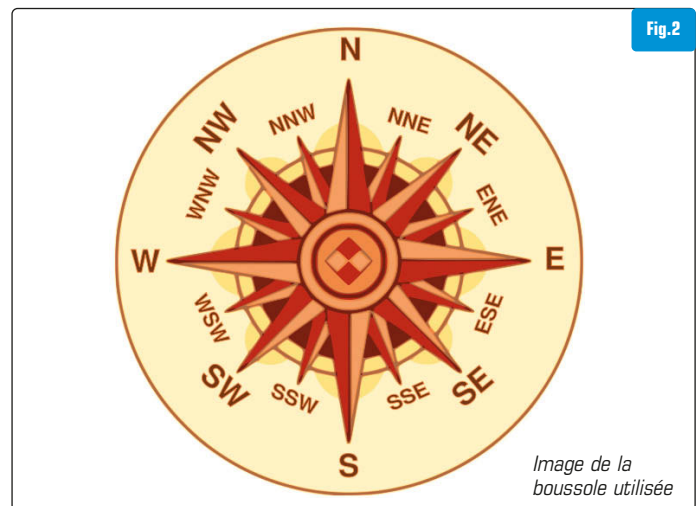


Fig.2

sole sera dessinée au centre, avec une rotation, pour laisser la boussole pointer toujours vers le Nord :

```
public class CompassView extends View {
    // ...
    public CompassView(Context context) {
        super(context);
        matrix = new Matrix();
        bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.compass_icon);
    }

    public void setBearing(float bearing) {
        this.bearing = bearing;
    }

    @Override
    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
        super.onMeasure(widthMeasureSpec, heightMeasureSpec);
        width = MeasureSpec.getSize(widthMeasureSpec);
        height = MeasureSpec.getSize(heightMeasureSpec);
        setMeasuredDimension(width, height);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        int bitmapWidth = bitmap.getWidth();
        int bitmapHeight = bitmap.getHeight();
        int canvasWidth = canvas.getWidth();
        int canvasHeight = canvas.getHeight();

        if (bitmap.getWidth() > canvasWidth || bitmap.getHeight() > canvasHeight) {
            bitmap = Bitmap.createScaledBitmap(bitmap, (int) (bitmapWidth * .9), (int) (bitmapHeight * .9), true);
        }

        // Calcul du centre
        int bitmapX = bitmap.getWidth() / 2;
        int bitmapY = bitmap.getHeight() / 2;
        int parentX = width / 2;
        int parentY = height / 2;
        int centerX = parentX - bitmapX;
        int centerY = parentY - bitmapY;
        // Calcul de l'angle de rotation
        int rotation = (int) (360 - bearing);

        matrix.reset();
        // Rotation du bitmap pour qu'il pointe vers le Nord
        matrix.setRotate(rotation, bitmapX, bitmapY);
        // Translation du bitmap vers le centre du canvas
        matrix.postTranslate(centerX, centerY);
        canvas.drawBitmap(bitmap, matrix, paint);
    }
}
```

Tout d'abord, on note l'utilisation de la méthode statique `decodeResource` de la classe `BitmapFactory` permettant de charger l'image de

la boussole depuis les ressources de l'application. Ensuite, dans la méthode `onDraw`, l'appel à la méthode statique `createScaledBitmap` de la classe `Bitmap` permet d'adapter la taille de l'image à celle du canvas où elle sera dessinée. Enfin, la classe `Matrix` sert de matrice de transformations et permet la rotation de l'image vers le Nord ainsi qu'un déplacement de l'image vers le centre du canvas lorsqu'elle est appliquée sur ce dernier via un appel à `drawBitmap`.

> Récupération des données

La vue `CompassView` réalisée, il reste à récupérer les données permettant son bon fonctionnement. Pour connaître le vrai Nord, on s'appuie sur l'accéléromètre et le capteur de champs magnétiques. Une fois les données de ces capteurs compilées, le calcul de la matrice de rotation correspondante sera fait grâce à la méthode statique `getRotationMatrix` de la classe `SensorManager`. Ensuite, c'est à partir de cette matrice de rotation que l'orientation du vrai Nord sera récupérée. L'emploi du capteur `TYPE_ORIENTATION` est remplacé par un appel à la méthode statique `getOrientation` de la classe `SensorManager` depuis la version 4 d'Android. Il est bon de noter que la valeur du vrai Nord correspondra à une valeur en degrés Est, comprise entre -180 et 180.

Enfin, pour améliorer la précision de la boussole, la différence existant entre le vrai Nord et le Nord magnétique est compensée. La méthode employée pour cette opération consiste à utiliser le terminal pour obtenir sa position GPS ainsi que son altitude. Appliquées en entrée de la classe `GeomagneticField`, ces données permettent d'obtenir l'angle d'écart en degrés entre le vrai Nord et le Nord magnétique. Une fois appliqué à la direction du Nord, il ne restera plus qu'à remettre la valeur de la direction sur la fenêtre [0,360] pour setter la direction sur la `CompassView` et la redessiner.

> Affichage de la boussole

Le code correspondant est implémenté au sein d'une activité dont voici les éléments essentiels :

```
public class MyCompassActivity extends Activity implements SensorEventListener, LocationListener {
    // ...
    @Override
    public void onStart() {
        super.onStart();
        sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        sensorGravity = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        sensorMagnetic = sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
        sensorManager.registerListener(this, sensorGravity, SensorManager.SENSOR_DELAY_NORMAL);
        sensorManager.registerListener(this, sensorMagnetic, SensorManager.SENSOR_DELAY_NORMAL);
        locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, MIN_TIME, MIN_DISTANCE, this);
        // Récupération de la dernière position connue
        Location gpsLocation = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
    }
}
```

```

if (gpsLocation != null) {
    currentLocation = gpsLocation;
} else {
    Location networkLocation = locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);

    if (networkLocation != null) {
        currentLocation = networkLocation;
    } else {
        // ... pas de position
    }
}

onLocationChanged(currentLocation);
}

@Override
protected void onStop() {
    super.onStop();
    sensorManager.unregisterListener(this, sensorGravity);
    sensorManager.unregisterListener(this, sensorMagnetic);
    locationManager.removeUpdates(this);
}

public void onLocationChanged(Location location) {
    currentLocation = location;
    geomagneticField = new GeomagneticField((float) currentLocation.getLatitude(), (float) currentLocation.getLongitude(), (float) currentLocation.getAltitude(), System.currentTimeMillis());
}

public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        smoothed = LowPassFilter.filter(event.values, gravity);
        gravity[0] = smoothed[0];
        gravity[1] = smoothed[1];
        gravity[2] = smoothed[2];
    } else if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
        smoothed = LowPassFilter.filter(event.values, geomagnetic);
        geomagnetic[0] = smoothed[0];
        geomagnetic[1] = smoothed[1];
        geomagnetic[2] = smoothed[2];
    }

    // Récupération de la matrice de rotation
    SensorManager.getRotationMatrix(rotation, null, gravity, geomagnetic);
    // Récupération du cap à cibler
    SensorManager.getOrientation(rotation, orientation);
    // Degrés Est du vrai Nord [-180,180]
    bearing = orientation[0];
    // Conversion radians vers degrés
    bearing = Math.toDegrees(bearing);

    // on compense la différence entre le vrai Nord et le Nord magnétique

```

```

if (geomagneticField != null) {
    bearing += geomagneticField.getDeclination();
}

if (bearing < 0) {
    bearing += 360;
}

compassView.setBearing((float) bearing);

if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER || event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
    compassView.postInvalidate();
}

updateTextDirection(bearing);
}
// ...
}

```

La seule inconnue dans ce code concerne la classe `LowPassFilter` et sa méthode statique `filter` appelée avant de setter les données récupérées des capteurs. Elle implémente un filtre passe-bas lissant les données renvoyées par le capteur entre 2 événements afin d'éviter de trop grandes variations. Ceci permettant un comportement plus proche de la réalité pour la boussole. L'exécution du programme

donne le résultat attendu, à savoir une boussole analogique ainsi qu'un affichage textuel de la direction [Fig.3].



Screenshot de la Boussole

Conclusion

A travers la réalisation d'une boussole, cet article aura mis en avant la simplicité et la puissance du `Sensor Framework` qui permet d'interagir au plus près avec les capteurs embarqués sur les terminaux Android. La démocratisation de ces capteurs permet au développeur d'imaginer des cas d'utilisation toujours plus nombreux aussi bien pour des jeux que pour des applications. L'emploi de ces capteurs amène une meilleure interactivité

avec l'utilisateur et ne peut rendre que plus populaires les applications les utilisant. Néanmoins, il est bon de garder à l'esprit que tous les terminaux ne possèdent pas l'ensemble des capteurs présentés ici pour des raisons hardware ou software. Dans ce cas-là, et pour éviter la frustration de potentiels utilisateurs, il est bon de préciser au sein du Manifest d'une application qu'elle ne cible que les terminaux possédant les capteurs nécessaires à son bon fonctionnement. Ainsi, l'application ne sera pas proposée aux terminaux non compatibles sur le Google Play Store.

Sylvain Saurel – Ingénieur d'Etudes Java / JEE
sylvain.saurel@gmail.com

Objective-C expert : «Associated Object», «Method Swizzling» et «Grand Central Dispatch»

Objective-C, langage propulsé par Apple et héritant du C, propose un ensemble de fonctionnalités avancées. Celles-ci ne sont que rarement utilisées dans des projets basiques, mais lorsque l'on est sur un projet complexe, il peut être intéressant d'utiliser certains de ces concepts.

Parmi ceux-ci, nous allons en étudier trois en particulier :

- Les «Associated Objects»
- Le principe de «Method Swizzling»
- Le système de distribution «Grand Central Dispatch»

La plupart de ces concepts utilisent massivement l'API « Runtime » d'Objective-C, principalement écrite en C.

LES «ASSOCIATED OBJECTS»

La plupart des langages objets actuels proposent la modification du modèle objet depuis l'environnement d'exécution, ou «Runtime». L'Objective-C, par rapport à C++, apporte de nombreuses fonctionnalités Runtime. Pour comprendre les «Associated Objects», il est nécessaire de comprendre l'intérêt du principe de la modification du modèle objet. Prenons par exemple une application connectée à un service en ligne de type «Cloud». Dans la plupart des cas, ce service permettra le stockage des données et l'accès à ces données se fera via des services web de type REST.

Cependant, supposons que nous voulions gérer et mettre à jour l'ensemble de notre modèle de données depuis un service distant. Si les développeurs rajoutent une propriété à un objet dans le modèle distant, il faudrait donc que l'objet de l'application soit mis à jour pour être cohérent avec le modèle distant.

Dans ce genre de situation, nous pourrions très bien envisager que tous les objets de notre application soient des dictionnaires de données (NSDictionary en Objective-C), où les clés représenteraient le nom de la propriété. Cependant, cela ne facilite pas l'utilisation de ceux-ci dans l'application.

La solution idéale est donc de pouvoir modifier notre modèle objet directement depuis l'exécution de l'application, par une méthode d'initialisation par exemple.

Nous allons donc utiliser les méthodes proposées par le «Runtime» du langage pour faire cela.

Les «Associated Objects» permettent d'ajouter une propriété à un objet depuis l'exécution de l'application. Ils sont aussi très utiles lorsque nous souhaitons modifier une classe existante, sans utiliser l'héritage. En effet, le concept d'héritage, bien que très simple à comprendre et à utiliser, oblige le développeur à modifier l'ensemble de notre code afin d'utiliser la classe fille plutôt que la classe mère. Objective-C propose une solution à ce problème : les catégories.

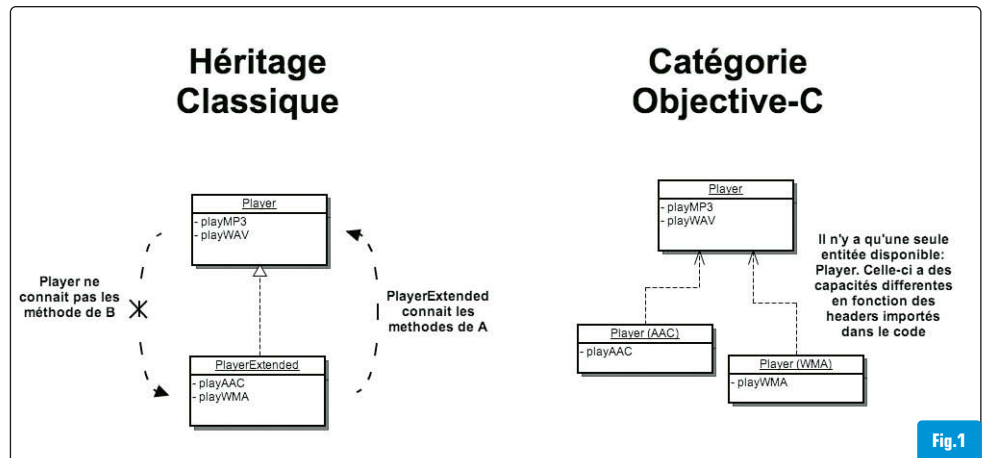


Fig.1

Une catégorie permet d'ajouter des méthodes à un objet existant, sans avoir à utiliser l'héritage [Fig.1].

Le souci avec les catégories, c'est qu'il n'est pas possible d'ajouter des propriétés dans celles-ci.

Pour cela, nous pouvons utiliser les «Associated Objects». Ceux-ci vont nous permettre d'ajouter des propriétés aux classes sans créer de classe fille, tout en utilisant les catégories.

Prenons comme exemple l'objet UIViewController de l'API CocoaTouch et créons une catégorie à celle-ci :

```
UIViewController+Extensions.h
@interface UIViewController (Extensions)
@end

UIViewController+Extensions.m
#import «UIViewController+Extensions.h»

@implementation UIViewController (Extensions)
@end
```

Pour le moment, notre catégorie n'ajoute rien. Commençons donc par ajouter une propriété à celle-ci :

```
@interface UIViewController (Extensions)

@property (nonatomic, retain) NSString* GUID

@end
```

Nous avons rajouté une propriété GUID de type NSString (pour mieux identifier chaque objet et permettre leurs synchronisations

sur le «cloud»). Si vous essayez de compiler, vous obtiendrez une erreur vous informant qu'il est impossible d'ajouter une propriété à une catégorie.

Nous allons donc utiliser les «Associated Objects» pour corriger le tir. Allez dans le fichier UIViewController+Extensions.m et rajoutez les deux méthodes suivantes :

```
-(void) setGUID:(NSString*)guid {
    objc_setAssociatedObject(self, &GUIDKey, guid, OBJC_ASSOCIATION_RETAIN_NONATOMIC);
}

-(NSString*) GUID {
    return objc_getAssociatedObject(self, &GUIDKey);
}
```

Pour que ces deux méthodes puissent être compilées, il est nécessaire de rajouter deux choses avant l'implémentation de votre classe : le header nécessaire à l'utilisation de l'API «Runtime» et le GUIDKey utilisé dans les méthodes précédentes (voir l'explication plus-bas).

```
#import <objc/runtime.h>
static char GUIDKey;

@implementation UIViewController (Ext...
```

Normalement, si vous compilez votre catégorie, vous ne devriez plus obtenir de message d'erreur de la part du compilateur.

Le 4^e paramètre de la méthode objc_setAssociatedObject permet de définir le type d'association entre votre objet et la propriété. Celui-ci doit correspondre à l'association définie dans la définition de la catégorie @property.

Notez ici que nous avons défini un static char GUIDKey, qui est utilisé dans les méthodes d'association. Cette variable n'est pas initialisée, car Objective-C va uniquement se servir de la valeur d'adresse mémoire comme clé, d'où la nécessité que celui-ci soit static et non défini dans la classe, afin de s'assurer que l'adresse mémoire soit la même durant toute l'exécution de l'application.

Il suffit d'importer le header de la catégorie dans le header pré-compilé de votre projet (*.pch) pour que cette nouvelle propriété soit disponible dans l'ensemble de votre application.

LE PRINCIPE DE «METHOD SWIZZLING»

Dans tout langage objet, la notion d'héritage est importante. Nécessairement, lorsque l'on hérite d'une classe, on récupère ses propriétés et ses fonctions, c'est la base de l'héritage.

Il arrive (très souvent) que l'on redéfinisse une méthode héritée pour ajouter un comportement particulier à celle-ci depuis la classe fille. Cependant, il y a des situations où l'héritage d'une classe n'est pas envisageable. Prenons par exemple le cas où vous souhaitez logger tous les appels à des méthodes systèmes, par exemple retain.

Vous pouvez, dans votre application, créer une catégorie à la classe NSObject et créer une méthode secondaire qui va logger l'appel de celle-ci, puis appeler retain. Cependant, vous ne pourrez modifier le code source d'appel à retain que dans votre code source (et pas dans celui des bibliothèques d'iOS par exemple).

Pour cela, un concept existe: le «Method Swizzling». Comme son

nom l'indique, celui-ci permet d'échanger une méthode avec une autre. (Voir [Fig.2]).

Ainsi, lorsque l'on appellera la méthode retain après son «swizzling», ce n'est pas l'implémentation de retain qui sera appelée, mais l'implémentation de la méthode alternative.

Créons une catégorie à la classe NSObject, puis ajoutons une méthode alternative à la méthode retain.

```
NSObject+MoreLog.h
@interface NSObject (MoreLog)

-(void) loggedRetain;

@end

NSObject+MoreLog.m
#import «NSObject+MoreLog.h»

@implementation NSObject (MoreLog)

-(void) loggedRetain {
    NSLog(@"Class %@ (%@) is retain (retain counter: %d)",
        NSStringFromClass([self class]),
        [self description],
        [self retainCount]+1);
    [self loggedRetain];
}

@end
```

Vous remarquerez que nous n'appelons pas la méthode retain dans notre méthode, c'est volontaire (voir l'explication plus bas).

Si vous tentez d'appeler cette méthode, votre programme tombera dans une boucle infinie, c'est normal, ceci est nécessaire pour que le «Method Swizzling» fonctionne une fois qu'il sera mis en place.

Une fois votre méthode correctement implémentée, nous allons demander le «swizzling» de notre méthode, pour cela quelques lignes de code suffisent :

```
Method original, swizzled;
origin = class_getInstanceMethod(self, @selector(retain));
swizzled = class_getInstanceMethod(self, @selector(loggedRetain));
method_exchangeImplementations(original, swizzled);
```

Vous noterez ici que nous demandons l'échange des implémentations, ce qui correspond bien au comportement voulu (et détaillé en [Fig.2]).

Assurez-vous que le swizzling de méthode soit bien exécuté au lancement de votre application. Puis exécutez votre application, vous devriez avoir de nombreux logs dans la console de XCode.

Je suppose que vous avez compris pourquoi dans notre nouvelle

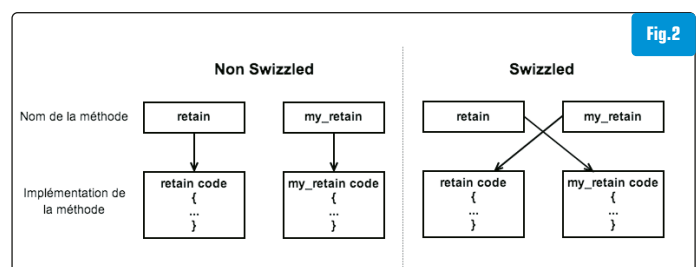


Fig.2

méthode nous faisons un appel récursif, si ce n'est pas le cas, voici une brève explication :

Une fois que les méthodes ont été «swizzlé» ou échangées, l'appel de `loggedRetain` appellera l'implémentation de `retain` et vice-versa. Si dans notre méthode `loggedRetain` nous avons appelé la méthode `retain`, celle-ci pointant sur l'implémentation de `loggedRetain`, nous aurions eu une boucle infinie.

Ce concept comme vous le remarquerez est un peu « tordu » et nécessite une grande précaution d'utilisation. Je vous recommande de n'utiliser l'échange de méthode que dans de rares cas, notamment à des fins de debug, où lorsqu'il n'y a pas d'autres choix.

LE SYSTÈME DE DISTRIBUTION «GRAND CENTRAL DISPATCH»



Grand Central Dispatch (abrégé GCD) est apparu sur iOS depuis sa version 4. Cette technologie permet d'optimiser la prise en charge des processeurs multicœurs en distribuant les processus sur

chaque cœur disponible.

Celui-ci fonctionne de manière asynchrone et est totalement écrit en **C**. Pour information, Apple a ouvert le code de GCD le 11 Septembre 2009 sous le nom de *libdispatch* (voir sur **MacOS Forge**: <http://libdispatch.macosforge.org/>)

Comme il s'agit d'une technologie récente d'Apple, elle utilise massivement les **Blocks** Objective-C, simplifiant ainsi la lecture du code source, à condition que celui-ci soit bien écrit (correctement indenté). Parmi les méthodes importantes à connaître, en voici quelques-unes :

- `dispatch_queue_t dispatch_queue_create(const char *label, dispatch_queue_attr_t attr);`
Permet de créer une file d'exécution dans GCD,
- `void dispatch_async(dispatch_queue_t queue, dispatch_block_t block);`
Permet d'ajouter un **block** Objective-C à exécuter dans une file d'exécution.
- `dispatch_queue_t dispatch_get_current_queue(void);`
Permet de récupérer la file d'exécution en cours d'utilisation là où la méthode est appelée,
- `dispatch_queue_t dispatch_get_main_queue(void);`
Permet de récupérer la file d'exécution principale, celle utilisée par l'UI dans votre application.

GCD est très simple d'utilisation, le meilleur moyen d'expliquer son fonctionnement est de montrer un exemple de code :

```
dispatch_queue_t myQueue = dispatch_queue_create("My new Queue", NULL);
dispatch_async(myQueue, ^{
    // Do something very long

    dispatch_async(dispatch_get_main_queue(), ^{
        //Then, update the UI with the result of your long code..
    });
});
```

Ces quelques lignes de code sont très claires :

- On crée une nouvelle file d'exécution identifiée par son nom (e.g. *My new Queue*),
- On exécute un block Objective-C dans cette file d'exécution,
- On exécute un block Objective-C dans la file d'exécution de l'UI à la fin de notre traitement asynchrone.

La plupart du temps l'utilisation que l'on fait de **Grand Central Dispatch** sous iOS se limite à ce genre d'exécution. Cependant, GCD ne se borne pas à ces quelques méthodes.

GCD fournit un ensemble de méthodes pour la gestion de processus décentralisés tels que les sémaphores, les priorités ou les groupes d'exécutions de **block**, etc.

Prenons par exemple l'exécution concurrente de plusieurs **blocks** et la gestion des priorités. Cette gestion s'effectue lors de la récupération d'un pointeur sur la file d'exécution.

Quatre priorités sont disponibles et sont définies à travers des clés :

- `DISPATCH_QUEUE_PRIORITY_HIGH` - Haute priorité
- `DISPATCH_QUEUE_PRIORITY_DEFAULT` - Priorité normale
- `DISPATCH_QUEUE_PRIORITY_LOW` - Faible priorité
- `DISPATCH_QUEUE_PRIORITY_BACKGROUND` - Uniquement exécutée lorsque le reste des tâches a été fait

Pour récupérer une file d'exécution avec la priorité voulue, il suffit (par exemple avec la file d'exécution globale), d'utiliser la méthode comme telle :

```
dispatch_queue_t globalConcurrentQueue = dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_BACKGROUND, 0);
// use the queue to call asynchronously your Objective-C block...
```



Vincent Saluzzo

Ingénieur étude et développement
Expert en développement iOS



A lire dans le prochain numéro

N°166-septembre 2013, à paraître le 2 septembre 2013

- **iPhone /iPad**
Débuter avec Objective-C et les outils XCode
- **Utilisez-vous du code illégal ?**
- **Webmaster**
l'incontournable JSON

Développer un explorateur de fichiers Dropbox sous iOS



Dropbox est une plateforme de stockage de fichiers en ligne. Celle-ci propose une API pour les développeurs souhaitant avoir accès à leurs services. Nous allons l'utiliser pour créer un explorateur de fichiers...

> Enregistrement d'une application sur Dropbox

Pour pouvoir utiliser l'API de Dropbox, il est nécessaire d'enregistrer votre application auprès de Dropbox. Si vous n'avez pas de compte, il vous faudra en créer un. C'est un des prérequis. Une fois inscrit (ou si vous l'êtes déjà), rendez-vous sur <https://www.dropbox.com/developers> et connectez-vous avec votre compte.

Cliquez ensuite sur le lien «My apps», vous accéderez à une page contenant la liste des applications liées à votre compte Dropbox en tant que développeur.

Cliquez ensuite sur «Create an app», une boîte de dialogue contenant plusieurs informations à saisir s'ouvrira :

- **App type** : Permet de choisir le type d'application. Ce choix détermine le type d'API que vous souhaitez utiliser dans votre application
- **App name** : Le nom de votre application
- **Description** : Une description de votre application
- **Access** : Le type d'accès que nécessite votre application (accès complet ou limité)

Dans notre cas, soyez bien sûr de choisir « API » dans « App type » et « Full Dropbox » dans « Access » [Fig.1].

Ensuite, cliquez sur « Create ». Si le nom de l'application n'a pas déjà été pris, vous serez dirigé sur la page principale de votre application. Celle-ci vous permet de définir certaines informations supplémentaires comme le nom, le site internet, le nom du développeur, etc. Elle permet avant tout d'avoir accès aux informations d'authentification nécessaires pour utiliser l'API Dropbox : l'« App Key » et l'« App Token » [Fig.2].

Ces deux valeurs sont nécessaires pour que l'API puisse déterminer quelle application effectue des requêtes et vérifie si celle-ci est autorisée à le faire.

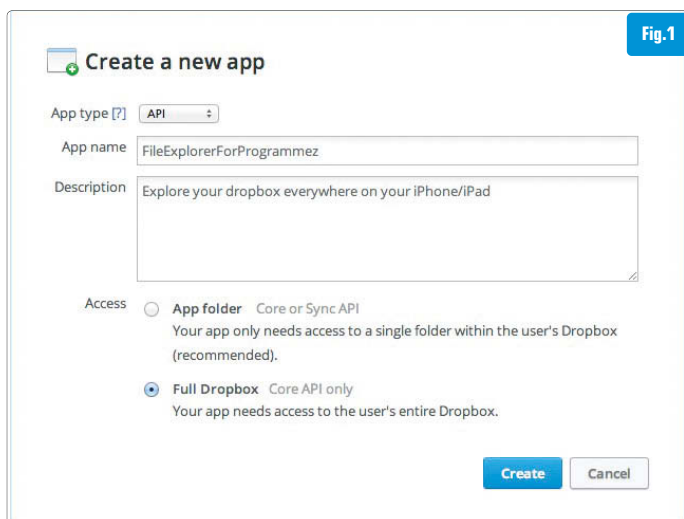


Fig.1

Si vous souhaitez plus de détails sur l'API Dropbox (aussi appelée « Core API » depuis fin 2012), allez sur la page <https://www.dropbox.com/developers/core/api>. Celle-ci permet d'obtenir le détail de chaque méthode disponible et d'autres informations telles que le format des requêtes, le chiffrement SSL, etc.

> Création d'une application utilisant Dropbox

Dropbox nous facilite la tâche sous iOS, un framework est disponible pour simplifier l'utilisation de l'API. Allez sur la page dédiée au SDK <https://www.dropbox.com/developers/core/sdk>, vous verrez que plusieurs SDK sont disponibles pour différentes plateformes et différents langages : iOS, Android, Python, Ruby, Java, etc.

Cliquez sur le SDK iOS. Celui-ci devrait être téléchargé sur votre ordinateur au format Zip. Extrayez l'archive et allez dans le dossier principal de celle-ci. Celui-ci est composé de trois dossiers :

- **DropboxSDK** - Projet XCode permettant la compilation du framework Dropbox
- **DropboxSDK.framework** - Framework Dropbox précompilé
- **examples** - Contient un exemple nommé DBRoulette permettant de voir comment utiliser Dropbox dans Xcode

Pour les plus curieux, nous vous invitons à consulter le projet XCode du framework. Vous pourrez ainsi voir comment sont faites les requêtes vers l'API Dropbox depuis iOS. Dans la suite de l'article, nous utiliserons uniquement le framework précompilé présent dans l'archive.

Commençons par créer un projet XCode. Un projet de type Single View Application sera amplement suffisant pour développer notre explorateur de fichiers.

Ajoutons le framework Dropbox au projet Xcode. Pour cela, faites un clic droit sur « Framework » dans l'explorateur du projet, puis cliquez sur « Add Files to «votre projet»... »

Dans la fenêtre qui vient de s'ouvrir, nous allons indiquer à Xcode qu'il doit importer le framework Dropbox en le copiant dans le répertoire du projet. Pour cela, sélectionnez le fichier `DropboxSDK.framework` précédemment téléchargé, soyez bien sûr d'avoir coché la case `Copy items into destination group's folder (if needed)`, puis cliquez sur `Add`.

Dropbox nécessite l'import de certaines bibliothèques fournies par Apple pour fonctionner : `Security.framework` et `QuartzCore.framework`. Sélectionnez votre projet dans l'explorateur du projet, puis allez sur l'onglet `Build Phases` et déroulez le menu `Link Binary with Libraries`. Le framework `DropboxSDK.framework` doit normalement être présent dans la liste. Cliquez sur l'icône «+» en bas de la liste, puis sélectionnez les deux frameworks nécessaires.

Ceux-ci seront ajoutés dans l'arborescence de votre projet au plus haut niveau. Afin de rendre celui-ci plus propre, Déplacez ces deux

frameworks dans le groupe dédié : Frameworks [Fig.3].

Votre projet doit normalement contenir le framework DropboxSDK.framework ainsi que les frameworks QuartzCore.framework et Security.framework.

Vérifions que le framework fonctionne bien, sélectionnez le fichier AppDelegate.h puis ajoutez une directive d'import du SDK #import <DropboxSDK/DropboxSDK.h> en haut du fichier. Compilez votre projet (Cmd+B), si le framework est correctement lié au projet et qu'aucun fichier nécessaire à la compilation n'est manquant, celle-ci devrait s'effectuer avec succès.

> Authentification de l'utilisateur à l'application

La première étape lorsqu'on utilise le framework iOS de Dropbox consiste à lier le compte de l'utilisateur à l'application. Sinon, aucune Dropbox ne sera accessible. Dropbox fournit toutes les classes nécessaires dans son framework, que demander de plus ?

Allez dans votre fichier AppDelegate.m puis ajoutez ce morceau de code dans la méthode application:didFinishLaunchingWithOptions: en prenant bien soin de remplacer APP_KEY et APP_SECRET par votre App Key et votre App Secret :

```
DBSession* dbSession = [[DBSession alloc]
initWithAppKey:@»APP_KEY»
appSecret:@»APP_SECRET»
root:kDBRootDropbox];
[DBSession setSharedSession:dbSession];
```

(Attention, veuillez faire attention à bien placer ce morceau de code après avoir initialisé le contexte graphique de l'application, après la méthode [self.window makeKeyAndVisible])

Cela va permettre de configurer le framework avec vos clés d'authentification. Ensuite, il est nécessaire de vérifier si l'application n'est pas déjà liée à un compte Dropbox et le cas échéant, proposer à l'utilisateur de le faire. Dans notre cas, nous allons faire cette vérification directement après avoir configuré les clés d'authentification. Rajoutons à la suite de [DBSession setSharedSession:dbSession] ce morceau de code :

```
if (![DBSession sharedSession] isLinked) {
[DBSession sharedSession] linkFromController:self.view
Controller];
}
```

Celui-ci va faire la vérification et envoyer l'utilisateur vers l'application Dropbox de votre appareil pour faire l'authentification (ou vers le site de Dropbox si l'application n'est pas présente sur votre appareil). Cependant, il est nécessaire que l'application (ou le site) sache retourner sur notre application une fois l'utilisateur authentifié. Pour

cela, le framework a besoin d'une URL de callback. Si vous êtes familier avec l'authentification via OAuth, le mécanisme est assez similaire.

Pour cela, cliquez-droit sur le fichier plist de votre projet (e.g FileExplorer-Info.plist) puis cliquez sur Open As > Source code. Ajoutez ceci juste après la première balise <dict>, en prenant soin de remplacer APP_KEY par votre App Key (disponible sur la page de votre application sur le site web de Dropbox) :

```
<key>CFBundleURLTypes</key>
<array>
<dict>
<key>CFBundleURLSchemes</key>
<array>
<string>db-APP_KEY</string>
</array>
</dict>
</array>
```

Cela permettra à iOS de savoir qu'un appel d'URL commençant par db-APP_KEY:// doit être intercepté par notre application. Une fois intercepté, iOS appellera une méthode de notre AppDelegate :

- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url.

Nous devons donc implémenter cette méthode pour vérifier que l'authentification s'est bien passée et que la liaison a bien été acceptée par l'utilisateur :

```
- (BOOL)application:(UIApplication *)application handleOpen
URL:(NSURL *)url {
if ([DBSession sharedSession] handleOpenURL:url) {
if ([DBSession sharedSession] isLinked) {
NSLog(@"Votre application a bien été liée.");
}
return YES;
}
// Add whatever other url handling code your app requires here
return NO;
}
```

Lancez votre application (Cmd+R), celle-ci proposera de vous authentifier avec votre compte Dropbox pour le lier à l'application [Fig.4].

Une fois authentifié, l'application devrait afficher dans la console *Votre application a bien été liée*. L'application est maintenant correc-

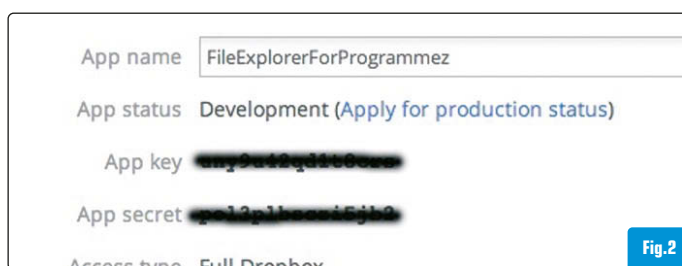


Fig.2

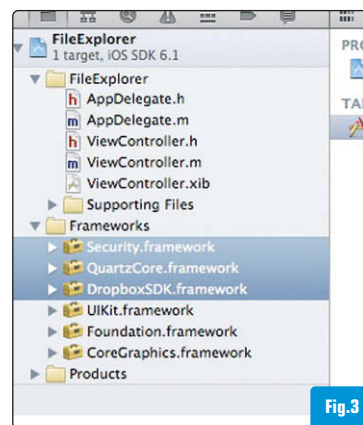


Fig.3

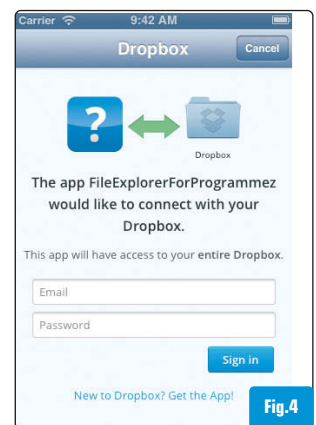


Fig.4

tement authentifiée auprès de Dropbox et peut dès à présent utiliser ses services.

> Réalisation de l'explorateur de fichiers

La mise en place d'un explorateur de fichiers nécessite le parcours d'une arborescence. Apple ne fournit pas de composants graphiques permettant le parcours d'arbre. Pour imiter cela, nous allons utiliser deux classes fournies par le SDK : `UITableViewController` et `UINavigationController`.

La classe `UINavigationController` va nous permettre de naviguer dans chaque répertoire et la classe `UITableViewController` permettra d'afficher la liste des fichiers contenus dans chaque répertoire.

Commençons par créer une nouvelle classe que l'on nommera `DropboxExplorerViewController`. Faites un clic droit sur le dossier `File Explorer` dans l'explorateur de projet puis cliquez sur `New File...`

Dans la boîte de dialogue qui vient de s'ouvrir, sélectionnez «Objective-C class», puis nommez-la comme précisé plus haut. Veillez à bien vous assurer que la case «With XIB for user interface» est bien décochée, nous n'aurons pas besoin d'Interface Builder pour prototyper notre explorateur. Notre nouvelle classe héritera de la classe `UINavigationController`, c'est elle qui sera responsable de la navigation entre les différents répertoires, donc pensez à bien renseigner le champ «Subclass of» en conséquence.

Une fois la classe créée, cliquez sur le fichier `DropboxExplorerViewController.h` puis commencez par ajouter une direction import pour ajouter le framework Dropbox. Modifions ensuite la classe principale de la sorte :

```
@interface DropboxExplorerViewController : UINavigationController
    <DBRestClientDelegate, UITableViewDataSource, UITableViewDelegate> {

        DBRestClient* client;
        NSString* currentPath;
        NSMutableDictionary* historyFileList;
    }

    -(id) init;
    -(void) getFileListInCurrentPath;

@end
```

Notre classe implémente trois protocoles :

- `DBRestClientDelegate` - Permet de gérer les résultats retournés par le framework Dropbox (fonctionnement asynchrone)
- `UITableViewDataSource` - Permet de définir le contenu des cellules représentant la liste des fichiers/répertoires
- `UITableViewDelegate` - Permet de définir les actions à effectuer par le contrôleur en cas d'interaction sur la liste (sélection, suppression, édition, etc.)

Trois propriétés et deux méthodes sont nécessaires au fonctionnement de notre explorateur :

- `DBRestClient* client` - Classe du framework Dropbox, elle permet d'encapsuler les requêtes vers l'API Dropbox.
- `NSString* currentPath` - Chaîne de caractères permettant de savoir dans quel répertoire l'utilisateur se trouve
- `NSMutableDictionary* historyFileList` - Dictionnaire de données per-

mettant de sauvegarder la liste des fichiers

- `-(id) init`; - Constructeur de la classe
- `-(void) getFileListInCurrentPath` - Méthode qui encapsulera l'utilisation du framework Dropbox.

Allez ensuite dans le fichier `DropboxExplorerViewController.m` pour donner du sens à notre classe. Commençons par créer nos deux méthodes de classe :

```
- (id)init {
    self = [super init];
    if (self) {
        client = [[DBRestClient alloc] initWithSession:
            [DBSession sharedSession]];
        client.delegate = self;
        currentPath = @"/";
        historyFileList = [NSMutableDictionary dictionary];
    }
    return self;
}
```

Ici rien de spécial, la méthode d'initialisation va instancier les propriétés de notre classe.

```
-(void) getFileListInCurrentPath {
    [client loadMetadata:currentPath];
}
```

Cette méthode simplifie l'utilisation du framework Dropbox (permet de faire abstraction de la méthode à appeler du framework Dropbox dans l'ensemble de notre classe). La méthode `-(void) loadMetadata:(NSString*)path` de la classe `DBRestClient` télécharge de manière asynchrone les informations d'une Dropbox. Si le path indiqué est un répertoire, les informations récupérées contiendront la liste des fichiers présents dans ce répertoire.

Nous allons ensuite enrichir notre classe en surchargeant les méthodes des protocoles implémentés qui nous intéressent :

• `DBRestClientDelegate` :

`-(void) restClient:(DBRestClient*)client loadedMetadata:(DBMetadata*)metadata`; Pour pouvoir récupérer la liste des fichiers/répertoires du répertoire courant

• `UITableViewDelegate` :

`-(void) tableView:(UITableView*)tableView didSelectRowAtIndexPath:(NSIndexPath*)indexPath`; Pour déclencher la navigation vers un nouveau répertoire

• `UITableViewDataSource` :

`-(NSInteger) tableView:(UITableView*)tableView numberOfRowsInSection:(NSInteger)section`; Pour indiquer le nombre de fichiers présents dans le répertoire courant

`-(UITableViewCell*) tableView:(UITableView*)tableView cellForRowAtIndexPath:(NSIndexPath*)indexPath`; Pour définir les cellules de la liste avec le nom des fichiers présents dans le répertoire courant.

Commençons par définir la méthode la plus concrète, celle qui va nous permettre de recevoir les informations provenant de Dropbox :

```
-(void) restClient:(DBRestClient *)client loadedMetadata:
    (DBMetadata *)metadata {
    [historyFileList setObject:metadata.contents forKey:
```

```
metadata.path];
    UITableViewController* tbv = [[UITableViewController
alloc] initWithStyle:UITableViewStylePlain];
    tbv.title = metadata.filename;
    tbv.tableView.dataSource = self;
    tbv.tableView.delegate = self;
    [self pushViewController:tbv animated:YES];
}
```

La méthode récupère et sauvegarde la liste des fichiers dans notre dictionnaire de données puis crée une instance de la classe UITableViewController en lui spécifiant son titre (le nom du répertoire désiré) ainsi que le delegate et le dataSource de l'objet tableView. Elle pousse ensuite le nouveau vue-contrôleur dans le contrôleur de navigation. Définissons ensuite comment doivent être affichées les informations du répertoire courant :

```
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionIn
Section:(NSInteger)section {
    NSArray* contents = [historyFileList objectForKey:currentPath];
    return [contents count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cell
ForRowAtIndexPath:(NSIndexPath *)indexPath {
    UITableViewCell* cell = [tableView dequeueReusableCellWithIdentifier:
Identifier:@"file cell"];
    if(cell == nil) {
        cell = [[UITableViewCell alloc] initWithStyle:UITableView
CellStyleDefault reuseIdentifier:@"file cell"];
    }
    NSArray* contents = [historyFileList objectForKey:currentPath];
    DBMetadata* fileMetadata = [contents objectAtIndex:index
Path.row];
    cell.textLabel.text = fileMetadata.filename;

    return cell;
}
```

La première méthode indique le nombre de fichiers contenus dans le répertoire courant en allant chercher l'information dans le dictionnaire de données. La seconde méthode récupère la liste des fichiers du répertoire courant et définit le titre de chaque cellule avec le nom de chaque fichier ou répertoire.

Il ne reste plus qu'à gérer la navigation entre les différents répertoires en interceptant la sélection d'une cellule :

```
-(void) tableView:(UITableView *)tableView didSelectRowAtIndex
Path:(NSIndexPath *)indexPath {
    NSArray* contents = [historyFileList objectForKey:currentPath];
    DBMetadata* fileMetadata = [contents objectAtIndex:index
Path.row];
    if(fileMetadata.isDirectory) {
        currentPath = fileMetadata.path;
        [self getFileListInCurrentPath];
    }
}
```

Ici, la méthode récupère le fichier sélectionné dans la liste puis vérifie que celui-ci est bien un répertoire. Si oui, elle modifie le répertoire courant pour correspondre au répertoire désiré et demande son chargement depuis Dropbox.

Il ne reste plus qu'à charger le nouveau vue-contrôleur depuis le fichier AppDelegate.m pour être le rootViewController de votre fenêtre principale.

Exécutez dès à présent le projet XCode (Cmd+R). Celui-ci doit normalement compiler sans erreur et vous affichera la liste des fichiers/répertoires de votre Dropbox [Fig.5].

Cependant, si vous essayez de revenir au dossier précédent et que vous sélectionnez un autre répertoire dans celui-ci, il y a des chances que votre application crashe.

L'explication est plutôt simple : nous modifions la propriété currentPath pour qu'elle reflète le répertoire courant lorsque nous sélectionnons un répertoire dans la liste des fichiers.

Cependant, nous n'avons défini nulle part que lorsque le contrôleur de navigation revient en arrière, celui-ci doit modifier la propriété currentPath en conséquence.

Modifions donc le comportement par défaut du contrôleur de navigation lorsque celui-ci revient en arrière :

```
-(UIViewController*) popViewControllerAnimated:(BOOL)animated {
    currentPath = [currentPath stringByDeletingLastPathComponent];
    return [super popViewControllerAnimated:animated];
}
```

Ré-exécutez le projet XCode et vous devriez normalement pouvoir naviguer dans les répertoires de votre Dropbox comme bon vous semble [Fig.6].

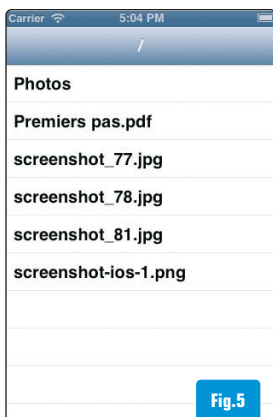


Fig.5

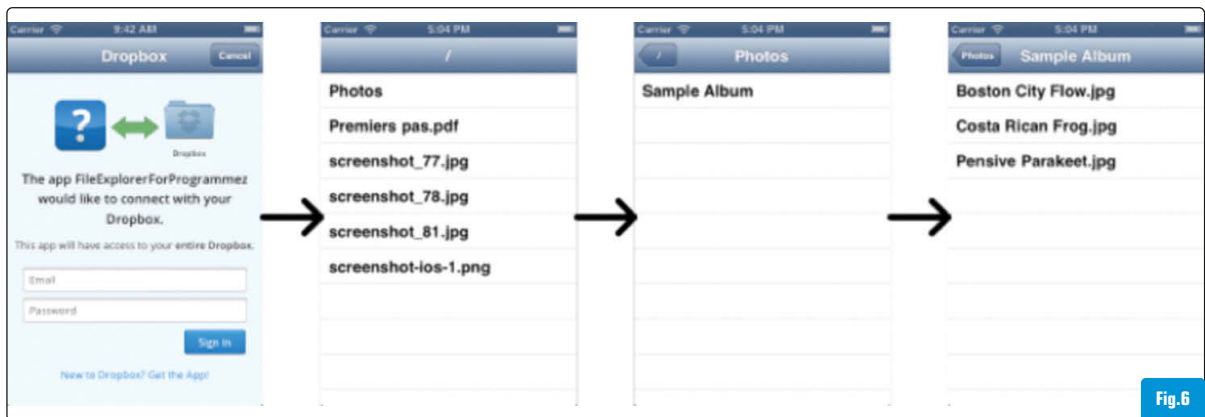


Fig.6

Notez cependant qu'une amélioration est envisageable. Actuellement, le dictionnaire de données n'est utilisé que lorsque l'on effectue un parcours descendant. Il pourrait être intéressant d'utiliser ce dictionnaire de données comme un cache local et proposer à l'utilisateur de rafraîchir la liste via un bouton.

Pour réaliser cela, il est d'abord primordial de modifier le comportement de la méthode `getFileListInCurrentPath` :

```
-(void) getFileListInCurrentPath {
    if([historyFileList objectForKey:currentPath] != nil) {
        [self loadListViewForCurrentPath];
    } else {
        [UIApplication sharedApplication].networkActivityIndicatorVisible = YES;
        [client loadMetadata:currentPath];
    }
}
```

Nous vérifions dans le dictionnaire de données si des informations sont déjà présentes pour le path courant. Si oui, nous appelons la méthode `loadListViewForCurrentPath` pour afficher un `UITableViewController`, sinon nous demandons au framework Dropbox de télécharger les informations.

Notez que pour différencier l'utilisation du cache et le téléchargement depuis Dropbox, nous avons ajouté `[UIApplication sharedApplication].networkActivityIndicatorVisible = YES;`. Cette ligne permet d'afficher un indicateur de chargement dans la barre de statut d'iOS. Nous devons donc définir la méthode `loadListViewForCurrentPath` pour qu'elle charge la liste des fichiers sur l'interface graphique :

```
-(void) loadListViewForCurrentPath {
    UITableViewController* tbv = [[UITableViewController alloc]
initWithStyle:UITableViewStylePlain];
    tbv.title = [currentPath lastPathComponent];
    tbv.tableView.dataSource = self;
    tbv.tableView.delegate = self;
    [self pushViewController:tbv animated:YES];
}
```

Cette méthode fait exactement ce que faisait la méthode déléguée du framework Dropbox. On peut donc remplacer son code par un appel à la nouvelle méthode :

```
-(void) restClient:(DBRestClient *)client loadedMetadata:(DBMetadata *)metadata {
    [historyFileList setObject:metadata.contents forKey:metadata.path];
    [UIApplication sharedApplication].networkActivityIndicatorVisible = NO;
    [self loadListViewForCurrentPath];
}
```

Nous avons légèrement modifié celle-ci pour qu'elle retire l'indicateur de chargement. Exécutez l'application (Cmd+R), vous verrez que l'indicateur de chargement ne s'affiche plus si vous retournez dans un répertoire déjà consulté.

Ajoutez maintenant un bouton permettant de rafraîchir la liste des fichiers du répertoire courant, ajoutez dans la méthode `loadListView-`

`ForCurrentPath`, juste avant de pousser le vue-contrôleur dans le contrôleur de navigation :

```
UIBarButtonItem* refreshButton = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemRefresh target:self action:@selector(refreshList)];
tbv.navigationItem.rightBarButtonItem = refreshButton;
```

Un bouton de rafraîchissement sera automatiquement ajouté dans la barre de navigation. Définissons maintenant la méthode qu'appellera le bouton. Mais avant, il est nécessaire de modifier légèrement la méthode récupérant la liste des fichiers depuis Dropbox :

```
-(void) getFileListInCurrentPathForce:(BOOL)force {
    if([historyFileList objectForKey:currentPath] != nil && force == NO) {
        [self loadListViewForCurrentPath];
    } else {
        [UIApplication sharedApplication].networkActivityIndicatorVisible = YES;
        [client loadMetadata:currentPath];
    }
}
```

Celle-ci prend maintenant un booléen en paramètre afin de déterminer si l'on doit forcer le téléchargement de la liste depuis Dropbox. La méthode `-(void) refreshList` appelée par le bouton ne fera qu'appeler cette méthode en mettant le booléen à vrai. Les autres appels à cette méthode doivent être mis à jour en mettant le booléen à faux. Il ne reste plus qu'à modifier la méthode déléguée du téléchargement depuis Dropbox :

```
-(void) restClient:(DBRestClient *)client loadedMetadata:(DBMetadata *)metadata {
    [UIApplication sharedApplication].networkActivityIndicatorVisible = NO;
    if([historyFileList objectForKey:metadata.path] != nil) {
        [historyFileList setObject:metadata.contents forKey:metadata.path];
        UITableViewController* tbv = [self.viewControllers lastObject];
        [tbv.tableView reloadData];
    } else {
        [historyFileList setObject:metadata.contents forKey:metadata.path];
        [self loadListViewForCurrentPath];
    }
}
```

Dorénavant, celle-ci détermine si le cache local contient déjà des informations sur ce path. Si c'est le cas, il ne charge pas un nouveau vue-contrôleur mais recharge la liste de celui qui est actuellement présenté à l'utilisateur. Nous avons donc un explorateur de fichiers Dropbox doté d'un cache local.



Vincent Saluzzo

Ingénieur étude et développement
«Expert en développement iOS»



Devenez le meilleur ami de votre admin, écrivez-lui des modules Powershell

Lorsque nous devons écrire des utilitaires, nous avons quasiment tous le même réflexe : écrire une application console. C'est là que cela se complique : il y a autant de façons d'écrire une application console qu'il y a de développeurs. Chaque application a son style pour passer des paramètres. En voici une liste non exhaustive : -param valeur ; -param=valeur ; -param:valeur ; /param valeur ; Voire même sans nom de paramètre. Déjà que nous avons du mal à nous mettre d'accord sur une norme de codage entre collègues, je n'imaginais pas une norme d'écriture de paramètres d'application console !

Cela n'est néanmoins pas le problème le plus important, le plus gênant étant que l'application est souvent fournie sans aide. Et, sans aide, impossible de connaître les paramètres.

De plus, tout est au format texte, ce qui ne laisse pas beaucoup de possibilités.

C'est pour toutes ces raisons que les administrateurs n'aiment pas trop ce type d'outils et que depuis quelques années ils se tournent tous vers Powershell.

En Powershell, on ne parle plus d'application console, mais de *Commandlet*. Les Commandlets ont plusieurs avantages par rapport aux lignes de commande classiques. En particulier :

- Elles ont toutes la même tête : « verbe-action » et le verbe doit être choisi dans une liste limitée ;
- Elles peuvent recevoir et renvoyer des objets .net ;
- On peut les découvrir dans le shell et l'écriture du code pour guider la saisie (autocomplétion) ;
- On peut les enchaîner dans un pipeline d'opérations.

Les commandlets sont regroupées en modules, toujours installés de la même façon dans un répertoire de Powershell. Par conséquent, plus besoin de faire de programme d'installation ou de changer une variable d'environnement « PATH » pour utiliser notre console.

Est-ce si difficile de faire un module contenant des commandlets ? Nous allons voir que non, bien au contraire !

Afin d'illustrer cet article nous allons faire un binary module contenant une commandlet qui va récupérer un contenu web. Nous allons

l'appeler : « get-httpcontent ». La première chose à faire lorsque l'on réalise un module est de créer un manifeste qui va décrire le module. Powershell inclut de base une commande pour créer un manifeste : « new-modulemanifest -path.\httpmodule.psd1 ».

Une commandlet étant du code, nous allons créer une « Class Library » qui va contenir ce code. Nous pouvons utiliser le Framework 4 car nous visons Powershell 3 pour cet article. Une fois le composant créé, j'y ajoute le fichier manifest [Fig.2].

La première étape est de préparer le manifeste, de retirer le commentaire sur la partie RootModule et d'y ajouter le nom de notre dll [Fig.3] :

Comme nous n'avons pas d'exécutable, il va falloir utiliser directement Powershell pour le debug. C'est également très simple : dans la partie debug, il faut ajouter comme exécutable : « C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe » et comme ligne de commande : « -NoExit -Command «& { Import-Module '.\HttpModule.psd1'} » ». Cette ligne de commande indique à Powershell de charger le module HttpModule sans fermer le shell. Nous allons donc avoir un shell interactif pour le debug.

Lançons le projet en début et appelons la méthode « Get-Module » comme sur la [Fig.4].

Le module est donc bien chargé. Maintenant nous allons pouvoir rentrer dans le vif du sujet : notre commandlet.

Il faut, dans un premier temps, ajouter les références à Powershell via l'assembly « System.Management.Automation ». Nous créons ensuite une classe GetHttpContentCmdlet qui hérite de PSCmdlet :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Management.Automation;
using System.Text;

namespace HttpModule
{
    public class GetHttpContentCmdlet : Cmdlet
    {
    }
}
```

J'aurais pu hériter de Cmdlet mais PSCmdlet me donne accès au contexte de Powershell, en particulier le chemin courant qui va me permettre de stocker le résultat de la commande.

Commençons à ajouter des paramètres à notre classe :

```
[Parameter(Mandatory = true)]
public string URL { get; set; }

[Parameter(Mandatory = true)]
public string FileName { get; set; }
```

L'attribut « Parameter » nous permet de spécifier comment Powershell va relier la ligne de commande à notre Cmdlet. Pour l'instant, nous allons identifier les arguments de notre cmdlet via le nom des propriétés.

Ecrivons maintenant le code qui va récupérer notre contenu :

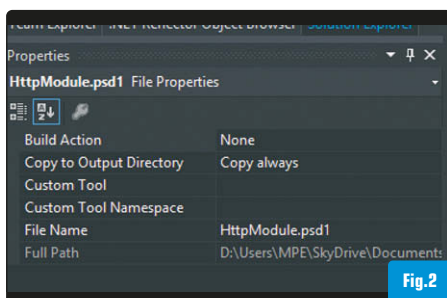


Fig.2

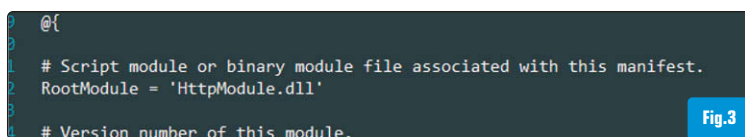


Fig.3

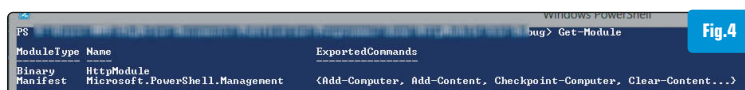


Fig.4

```
protected override void ProcessRecord()
{
    Uri uri;

    try
    {
        uri = new Uri(URL);
    }
    catch (Exception ex)
    {
        this.WriteError(new ErrorRecord(ex, «InvalidURL», ErrorCategory.
InvalidArgument, URL));
        return;
    }

    var currentLocation = this.CurrentProviderLocation(@"Microsoft
.PowerShell.Core\FileSystem");

    string filePath = Path.Combine(currentLocation.Path, File
Name);

    HttpClient client = new HttpClient();

    using (client)
    {
        HttpResponseMessage request;
        try
        {
            var requestTask = client.GetAsync(uri);
            request = requestTask.Result;
            request.EnsureSuccessStatusCode();
        }
        catch (Exception ex)
        {
            this.WriteError(new ErrorRecord(ex, «InvalidURL», ErrorCategory.
InvalidArgument, URL));
            return;
        }

        var streamTask = request.Content.ReadAsStream
Async();

        using (var stream = streamTask.Result)
        {
            using (var outputStream = File.Open
Write(filePath))
            {
                stream.CopyTo(outputStream);
            }
        }
    }
}
```

```
        outputStream.Flush();
    }
}
}
```

Quelques remarques sur ce code source :

- La gestion des erreurs est réalisée via la méthode « TrackError » ;
- J'utilise CurrentProviderLocation pour récupérer le dossier courant : en effet Powershell possède une notion de disque (drive) qui peut être tout autre chose, comme un chemin de la base de registre par exemple.

Lançons maintenant notre nouvelle cmdlet avec comme arguments les noms des propriétés [Fig.5].

Powershell est assez intelligent pour qu'on ne lui donne qu'une partie du nom des paramètres du moment sans qu'il y ait d'ambiguïté.

Un autre grand intérêt de Powershell est de laisser l'utilisateur choisir sa façon de passer des paramètres. N'oublions pas que c'est pour lui que l'on travaille. Je vais ajouter une propriété « position » à nos paramètres :

```
[Parameter(Mandatory = true,Position=1)]
public string URL { get; set; }

[Parameter(Mandatory = true,Position=2)]
public string FileName { get; set; }
```

J'ai maintenant la possibilité de passer directement les valeurs à ma commande [Fig.7]. Une autre fonctionnalité intéressante de Powershell est de renvoyer des objets .Net. Un paramètre très courant en Powershell nommé « passthru » permet de spécifier que l'on veut récupérer un objet de nombre commande. Ici, par exemple, nous voulons récupérer un objet FileInfo sur le fichier généré.

Pour cela, créons un paramètre de type SwitchParameter :

```
[Parameter]
public SwitchParameter PassThru { get; set; }
```

La dernière étape de notre méthode consiste à vérifier si le paramètre a été spécifié dans la ligne de commande et dans ce cas, renvoyer un objet de type FileInfo [Fig.8] :

```
if (PassThru.IsPresent)
{
    WriteObject(new FileInfo(filePath));
}
```

Ceci n'est qu'un petit aperçu de ce que l'on peut faire avec Powershell et les commandes que l'on peut utiliser.

Michel Perfetti

Manager / MVP Visual Studio ALM chez Cellenza

Cellenza - Software Development Done Right

Blog: <http://www.buildmeinfamous.net> - Twitter : @miitch



Fig.5



Fig.6



Fig.7



Fig.8

Mettre en œuvre le protocole WebSocket

Fonctionnalité de HTML 5, le protocole WebSocket est promis à un bel avenir, en raison de ses possibilités intrinsèques et parce qu'il soulage serveurs et réseaux. Il est en outre aisé à mettre en œuvre. Découverte.

Encore en cours de spécification au moment où nous écrivons ces lignes, de même que le standard HTML 5 dont il fait partie, le standard WebSocket est suffisamment défini et implémenté dans les navigateurs majeurs pour que l'on puisse dire de lui qu'il devrait révolutionner le Web et le développement des applications Web. Le standard WebSocket définit à la fois un protocole réseau et une interface de programmation. Ce protocole réseau permet de gérer une communication bi-directionnelle entre un navigateur et un serveur. Ce qui est révolutionnaire dans tout cela, c'est que l'on ne dit (presque) plus que le navigateur est un client. Il l'est encore au moment de l'établissement d'une communication. Mais après cette phase, la communication est réellement bi-directionnelle, tout comme elle l'est entre deux sockets classiques.

1 AVANT WEBSOCKET

La communication entre un navigateur client et le serveur d'un site s'effectue par le protocole HTTP. Ce protocole est sans état, ce qui signifie que chaque fois qu'un navigateur souhaite afficher ou rafraîchir une page, il doit émettre une requête HTTP. Autrement dit, il doit interroger le serveur. Ce dernier ne peut pas informer le navigateur client d'un nouvel état du site, comme, par exemple, la publication d'une actualité tout fraîche. Cela convient parfaitement pour des pages Web statiques, dont le contenu, par définition, ne change que rarement. C'était donc parfait à l'époque de la préhistoire du Web. Cela l'était beaucoup moins dès l'apparition des pages dynamiques, et c'est insuffisant avec le Web d'aujourd'hui et son interactivité. Ainsi le fil d'actualité d'une page Facebook d'un compte ayant un millier d'amis, est en mouvement constant. Pour donner une impression de temps réel, le navigateur doit interroger en continu le serveur. Le trafic réseau explose en conséquence, et la charge des serveurs également. Des techniques sont apparues pour y faire face. Ainsi Ajax et ses frameworks JavaScript qui permettent de modifier de manière asynchrone une partie, et une partie seulement, d'une page Web, au lieu de la reconstruire complètement. On gagne en expérience utilisateur, et en charge serveur. Mais cela devient vite insuffisant dans les cas extrêmes, dont une page Facebook est un exemple représentatif. Il faut alors interroger le serveur en continu, ou en donnant l'impression d'une continuité. Pour limiter la charge serveur et réseaux, des techniques comme le *long polling* ou les *channels comet* sont apparues. Mais elles ne règlent pas le problème qui reste que, fondamentalement, le navigateur client doit questionner le serveur. C'est ici que WebSocket intervient. Avec lui le navigateur peut rester en attente, et le serveur peut lui pousser des informations à bon escient, sans subir d'interrogations inutiles. Nous allons maintenant voir comment travailler avec WebSocket, et montrer comment on peut le mettre en œuvre sur un site avec très peu de code.

2 LES OUTILS

Il est bien évident que nous devons utiliser des navigateurs qui implémentent suffisamment le protocole WebSocket. Les exemples qui

accompagnent cet article, et dont le lecteur trouvera le code sur notre site, ont été essayés avec succès avec Firefox 17.0, Google Chrome 23.0 et Opera 12.12. Il est à noter qu'Internet Explorer 9 **ne supporte pas WebSocket**. En revanche Internet Explorer 10, le navigateur de Windows 8, le supporte, mais nous n'avons pas testé nos exemples avec lui. Enfin, il ne faut pas trop se fier à ce que l'on pourrait trouver sur Internet à propos de WebSocket. En effet, les spécifications ont évolué plusieurs fois au cours du temps et de nombreux articles ne sont plus à jour. Au lieu de cela, il faut se référer, sans modération, à la Bible de WebSocket, à savoir le document RFC 6455, que l'on trouvera à <http://tools.ietf.org/html/rfc6455>.

3 LE PRINCIPE DE WEBSOCKET

Sur le fond, les sockets de WebSocket sont des sockets :) Tout programmeur qui n'a, ne serait-ce que des connaissances de base des sockets classiques, s'en tirera très bien avec WebSocket. En pratique, le travail avec WebSocket comprend deux phases. La première phase dite handshake (en français poignée de main) concerne la négociation de la connexion. La deuxième phase concerne le transfert des données au travers de la connexion établie. Ce transfert de données n'est pas aussi immédiat qu'avec les sockets classiques. Il obéit lui aussi à un protocole que nous verrons plus loin. Le handshake se déroule en deux temps. Dans le premier temps, le client, a priori un navigateur, mais ce pourrait être n'importe quelle application, émet une requête HTTP Upgrade vers le serveur. Par exemple :

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

Que le handshake se fasse à travers une requête permet de n'utiliser qu'un seul port, le port 80. Il faut donc que la requête puisse être comprise par les serveurs HTTP, d'où sa forme. Cependant WebSocket n'oblige pas à utiliser ce port, ce qui permet, par exemple, de se connecter à un petit serveur bricolé maison. Les Requêtes HTTP du handshake peuvent être très fournies. Il convient de bien lire le RFC 6455, afin de bien différencier ce qu'elles doivent contenir de ce qu'elles peuvent contenir. Au niveau des obligations, un client doit donner une clé encodée en base64. Le serveur doit, à partir de cette clé, en construire une autre, la coder en base64 et la retourner au client. Si celui-ci reçoit la valeur de clé à laquelle il s'attend, la connexion s'établit. Dans le cas contraire, la connexion échoue. Voici un exemple de réponse du serveur :

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
```



```
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: chat
```

4 LE TRAVAIL CÔTÉ CLIENT

S'il est parfaitement possible de programmer une application cliente qui implémente le protocole WebSocket, le client naturel sera le navigateur Internet. Pour cette raison, une API Javascript est définie par le W3C dans le cadre du standard HTML 5.

Les spécifications de cette API peuvent être consultées à cette page : <http://dev.w3.org/html5/websockets/>. De ce point de vue, WebSocket se résume à une interface implémentée par les navigateurs.

```
enum BinaryType { «blob», «arraybuffer» };
[Constructor(DOMString url, optional (DOMString or DOMString[])
protocols)]
interface WebSocket : EventTarget {
    readonly attribute DOMString url;

    // ready state
    const unsigned short CONNECTING = 0;
    const unsigned short OPEN = 1;
    const unsigned short CLOSING = 2;
    const unsigned short CLOSED = 3;
    readonly attribute unsigned short readyState;
    readonly attribute unsigned long bufferedAmount;

    // networking
    attribute EventHandler onopen;
    attribute EventHandler onerror;
    attribute EventHandler onclose;
    readonly attribute DOMString extensions;
    readonly attribute DOMString protocol;
    void close([Clamp] optional unsigned short code, optional DOM
String reason);

    // messaging
    attribute EventHandler onmessage;
    attribute BinaryType binaryType;
    void send(DOMString data);
    void send(Blob data);
    void send(ArrayBuffer data);
    void send(ArrayBufferView data);
};
```

Nous comprenons que cette API est de nature asynchrone. En effet, immédiatement après qu'une connexion soit établie, ce qui se passe au sein de celle-ci est rapporté par des événements. Voici un exemple de page HTML cliente (demo-client-echo.html sur notre site), pour nous faire la main avec cette API. Pour cette démonstration, nous nous connectons à un site de démonstration qui propose un service d'écho : il retourne immédiatement dans le socket ce que nous lui envoyons [Fig.1].

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
«http://www.w3.org/TR/html4/loose.dtd»>
<html>
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=
ISO-8859-1">
<title>Demo client WebSocket</title>
</head>
<body>

<script language="javascript" type="text/javascript">

var wsEchoURI = «ws://echo.websocket.org/»;
var websocket;

function onWSOpen(evt)
{
    printConsole(«WebSocket ouvert»)
    if(websocket.readyState == websocket.OPEN)
    {
        printConsole(«WebSocket prêt»);
    }
}

function onWSClose(evt)
{
    printConsole(«WebSocket fermé»);
    printConsole(websocket.readyState);
}

function onWSMessage(evt)
{
    printConsole(evt.data);
}

function onWSError(evt)
{
    printConsole(«Une erreur s'est produite dans le WebSocket : »
+ evt.data);
}

function boutonConnexion()
{
    printConsole(«WebSocket in window»);
    printConsole(«Connexion»);
    websocket = new WebSocket(wsEchoURI);
    websocket.onopen = function(evt) { onWSOpen(evt) };
    websocket.onclose = function(evt) { onWSClose(evt) };
    websocket.onmessage = function(evt) { onWSMessage(evt) };
    websocket.onerror = function(evt) { onWSError(evt) };
}

function boutonDeconnexion()
{
    websocket.close();
}

function boutonEnvoyer()
{
    websocket.send(«Programmez!»);
    websocket.send(«Abonnez-vous ! :-)»);
}
```

```
function printConsole(texte)
{
    var console = document.getElementById("console");
    var elem = document.createElement("p");
    elem.style.wordWrap = "break-word";
    elem.innerHTML = texte;
    console.appendChild(elem);
}

</script>

<h1>D  mo de client WebSocket</h1>

<div id="interface">
<p>Une belle interface</p>
<button id="connexion" onclick="buttonConnexion()">Connexion
</button>
<button id="deconnexion" onclick="buttonDeconnexion()">D  connexion
</button>
<button id="envoyer" onclick="buttonEnvoyer()">Envoyer</button>
</div>

<p>Par ici la sortie :</p>
<div id="console">
</div>
</body>
</html>
```

Ce code est fort simple, ce qui est tout l'int  r  t de l'API JavaScript. On remarquera, au d  but de la fonction `buttonConnexion`, qu'il est possible de tester si WebSocket est support   par le navigateur. Nous affichons un bool  en en cons  quence. Lorsque le socket re  oit quelque chose du serveur, l'  v  nement `onmessage` est d  clench  . Lorsque nous voulons   crire dans le socket, un appel    la m  thode `send` fait le travail. Le code est donc tout simple car l'API encapsule le protocole WebSocket proprement dit, et le rend transparent. Les choses se passent comme si lire ou   crire dans le socket   tait imm  diat. En r  alit  , c'est plus compliqu  , ainsi que l'on s'en rend compte en   tudiant le c  t   serveur de la question.

5 LE TRAVAIL C  T   SERVEUR

D  s qu'il s'agit de mettre en place un serveur de WebSocket, on entend syst  matiquement «node.js» en guise de solution qui va bien. Node.js est incontestablement un bon outil qui encapsule le

protocole WebSocket. Mais bien des webmasters pourraient   prouver le besoin de mettre en place un serveur de WebSocket au sein de leur existant. Nous allons donc voir comment faire. Ce travail est possible a priori avec n'importe quel langage supportant les sockets de base. Nous travaillerons avec Python, parce que de l'humble avis de votre serviteur, il va mieux qu'un autre et fait le travail avec un code tr  s concis. Notre but est de cr  er un serveur (Fig.2) qui pousse des informations    un navigateur d  s que celui-ci a ouvert une connexion WebSocket avec lui. (Fig.3) Notre client (demo-client-echo-1.html sur notre site) est le m  me que le pr  c  dent, hormis l'URL du serveur :

```
var wsEchoURI = "ws://localhost:9999/";
```

Nous travaillons avec un port quelconque (ici 9999) par souci de simplicit  , mais nous aurions pu utiliser le port 80, et passer, par exemple, au travers du SimpleHTTPServer de la librairie de Python.

6 LE HANDSHAKE

Le fonctionnement est le suivant : le client envoie une cl   quelconque, mais encod  e en base64 au serveur. Celui-ci doit concat  ner    cette cl   un GUID, toujours le m  me, et d  fini dans le RFC 6455. Il doit ensuite hacher la cha  ne r  sultante avec l'algorithme SHA-1, puis retourner au client le r  sultat de ce hachage, mais encod   en base64. De son c  t  , le client a fait le m  me travail. Il compare la cl   re  ue    celle qu'il s'attend    recevoir et valide ou non la connexion. Le RFC 6455 donne un exemple de cl     mise par le client et la valeur que doit en cons  quence retourner le serveur. Le script ci-dessous (ws-handshake.py sur notre site, fait ce travail) :

```
#!/usr/bin/env python

import hashlib, base64

SecWebSocketKey = "dGh1IHhxbXBsZSsBub25jZQ=="
magicguid = "258EAF5-E914-47DA-95CA-C5AB0DC85B11"
cle = SecWebSocketKey + magicguid

hashcle = hashlib.sha1()
hashcle.update(cle)
print hashcle.digest()
print hashcle.hexdigest()

cle = base64.b64encode(hashcle.digest())
```

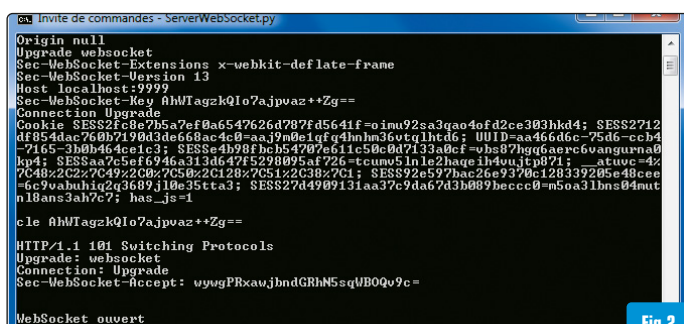


Fig.2

Notre serveur affiche sur sa console ce qu'il re  oit du navigateur client.

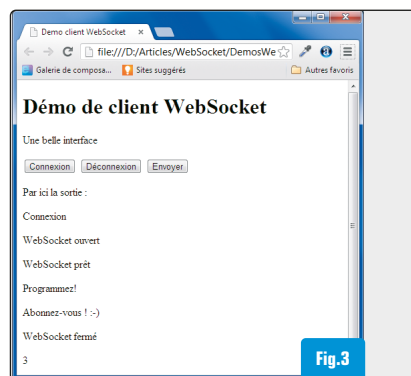


Fig.3

D  s que la connexion WebSocket est   tablie, notre serveur pousse des cha  nes au navigateur client.

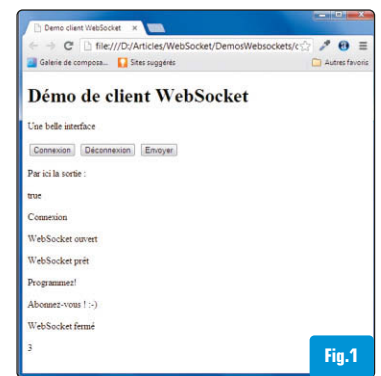


Fig.1

Notre client Javascript se connecte    un service d'  cho.


```
print «Attendu», «s3pLMBiTxQ9kYGzzhZRbK+x0o=»
print «Obtenu », cle
```

7 LA TRANSMISSION DES DONNÉES

Nous arrivons maintenant à la partie un peu difficile, le protocole de transmission des données. C'est difficile car le traitement est différent selon la taille et la teneur des données échangées : du texte, du binaire, une seule trame, plusieurs trames, etc. Le tout est empaqueté dans des octets codés comme indiqué dans le RFC 6455 :

										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
F R R R opcode M Payload len Extended payload length																																							
I S S S (4) A (7) (16/64)																																							
N V V V S (if payload len==126/127)																																							
1 2 3 K																																							
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
Extended payload length continued, if payload len == 127																																							
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
Masking-key, if MASK set to 1																																							
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
Masking-key (continued) Payload Data																																							
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
: Payload Data continued ... :																																							
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
Payload Data continued ...																																							
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+										+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									

Le tout premier bit (FIN) du tout premier octet est armé si les données envoyées sont les dernières d'une transmission. Attention, car c'est parfaitement contre-intuitif, ce bit de numéro 0 est le bit le plus fort de l'octet. Viennent ensuite 3 bits réservés à des extensions de protocole. Ces bits seront à zéro pour nous. Viennent ensuite 4 bits qui décrivent le type de transmission (texte, binaire, continuation, etc.), puis un bit qui, s'il est armé, indique que les données sont cryptées à travers un masque. Ensuite, si la longueur de nos données à transmettre est inférieure ou égale à 125, viennent 7 bits qui codent cette longueur. Mais si ces 7 bits valent 126, ce sont les 16 bits suivants qui codent la longueur des données, et si ces 7 bits valent 127, alors ce sont les 64 bits suivants qui codent la longueur. Pourquoi faire simple quand on peut faire compliqué :-) Viennent ensuite l'éventuel masque, puis, ô miracle, les données :-)

Quand les chaînes sont courtes, les choses sont toutes simples. Ainsi si nous voulons écrire *Programmez!* dans le socket. La longueur de nos données est 11 (0x0B). Nous transmettons les données en une fois, donc le fameux bit FIN du début :-) doit être armé et le code d'opération pour les 4 bits opcode est 1. Nous devons donc écrire cette chaîne dans le socket :

```
'\x81\x0BProgrammez!'
```

Voici maintenant notre programme d'exemple complet. Le principe est tout simple. Nous écoutons le port 9999 et nous ouvrons un socket de bas niveau quand un client tente de se connecter. Nous lisons alors dans ce socket de bas niveau la requête HTTP que le client envoie, nous en extrayons la clé pour réaliser le handshake. Si tout se passe bien, notre socket de bas niveau est devenu un WebSocket. Nous pouvons alors y écrire, via un thread, des données selon le protocole que nous venons d'expliquer. Au passage on appréciera l'efficacité de Python qui en une ligne de code :

```
headers = dict(re.findall(r»(?P<name>.*?): (?P<value>.*?)\r\n», dataheaders))
```

transforme la requête HTTP reçue du client en un dictionnaire de paires clés/valeurs :-)

```
#!/usr/bin/env python
```

```
import socket, threading, time, re, hashlib, base64
```

```
magicguid = «258EAF5-E914-47DA-95CA-C5AB0DC85B11»
```

```
def printHeaders(headers):
    print «Headers reçu du WebSocket client»
    for key in headers.keys():
        print key, headers[key]
    print
```

```
def doHandShake(conn, dataheaders):
    headers = dict(re.findall(r»(?P<name>.*?): (?P<value>.*?)\r\n», dataheaders))
    printHeaders(headers)
    cle = headers['Sec-WebSocket-Key']
    print «cle», cle
    cle += magicguid
    hashcle = hashlib.sha1()
    hashcle.update(cle)
    cle = base64.b64encode(hashcle.digest())
    handshake = «HTTP/1.1 101 Switching Protocols\r\n»
    handshake += «Upgrade: websocket\r\n»
    handshake += «Connection: Upgrade\r\n»
    handshake += «Sec-WebSocket-Accept: « + cle + «\r\n»»
    # ligne vide de fin de header
    handshake += «\r\n»
    print
    print handshake
    conn.send(handshake)
```

```
def handle(conn):
    time.sleep(1)
    conn.send('\x81\x0BProgrammez!')
    time.sleep(1)
    conn.send('\x81\x12Abonnez-vous ! :-)')
    time.sleep(1)
    conn.close()
```

```
s = socket.socket()
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
s.bind(('127.0.0.1', 9999));
s.listen(1);
while 1:
    conn,address = s.accept();
    dataheaders = conn.recv(4096)
    doHandShake(conn, dataheaders)
    print «WebSocket ouvert»
    threading.Thread(target = handle, args = (conn,)).start()
```

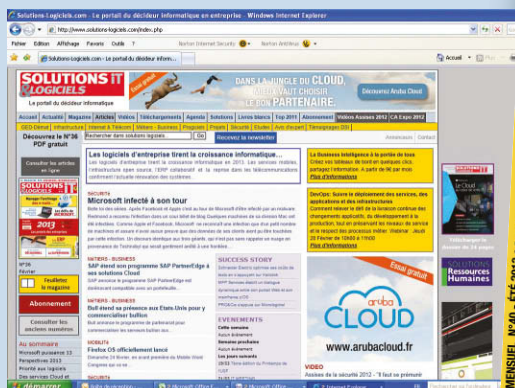
Frédéric Mazué
fmazue@programmez.com

Les outils des Décideurs Informatiques

Vous avez besoin d'info
sur des sujets
d'administration,
de sécurité, de progiciel,
de projets ?
Accédez directement
à l'information ciblée.

Cas clients

Actu triée par secteur | Avis d'Experts



Actus / Evénements | Newsletter | Vidéos

LE MAGAZINE DES DÉCIDEURS INFORMATIQUES

SOLUTIONS & LOGICIELS

ÉTÉ 2013 N°40
www.solutions-logiciels.com

L'OPEN SOURCE
grignote les logiciels d'infrastructure p.10

DOSSIER
L'agilité du datacenter p.20

Stockage NAS p.28
Simplicité d'emploi

CHORUS p.06
Comment l'Etat économise 427M€ an

La percée du COFFRE-FORT NUMERIQUE p.36

Supply-Chain p.42
Solutions écotaxes et mobiles

Cloud Computing
Numergy voit en grand l'usine numérique p.26

Communiquer p.46
pour réussir ses projets

Philippe Tavernier, président de Numergy

"L'automatisation est un élément clé de notre stratégie"

M 09551 - 40 - F: 6,00 € - RD

www.solutions-logiciels.com

☐ **OUI, je m'abonne** (écrire en lettres capitales)

Envoyer par la poste à : Solutions Logiciels, service Diffusion, GLIE - 17 chemin des Boulangers 78926 Yvelines cedex 9 - ou par fax : 01 55 56 70 20
1 an : 50€ au lieu de 60€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 60€ - Algérie, Maroc, Tunisie : 65€ , Canada : 80€ - Dom : 75€ Tom : 100€
10 numéros par an.

☐ M. ☐ Mme ☐ Mlle Société

Titre : Fonction : ☐ Directeur informatique ☐ Responsable informatique ☐ Chef de projet ☐ Admin ☐ Autre

NOM Prénom

N° rue

Complément

Code postal : [] [] [] [] Ville

Adresse mail

☐ Je joins mon règlement par chèque à l'ordre de SOLUTIONS LOGICIELS ☐ Je souhaite régler à réception de facture

HTML5+jQuery

Toute App, Tout Navigateur, Toute Plateforme, Tout Appareil



IGNITEUITM
INFRAGISTICS JQUERY CONTROLS



Téléchargez Votre Version d'Evaluation Gratuite
www.infragistics.com/telechargez



INFRAGISTICSTM
DESIGN / DEVELOP / EXPERIENCE

Infragistics Ventes France  0800 667 307 • Infragistics Ventes Europe +44 (0) 800 298 9055

Copyright 1996-2013 Infragistics, Inc. All rights reserved. Infragistics and NetAdvantage are registered trademarks of Infragistics, Inc. The Infragistics logo is a trademark of Infragistics, Inc.