

# Présentation

## PHP

- Composer
- PSR

## PHP Toolkit

Proc.DB2  
externes

## ZendServer

- Z-Ray
- SDK
- Apigility

Continuous I&D

## Colloque IBM

19 et 20 mai 2015 – IBM Client Center, Bois-Colombes

## Nouveautés PHP - Zend Server 7 & 8 - Z-RAY

*Mardi 20 mai – 12h30-13h30*

Grégory JARRIGE – [Linkedin](#)



# L'écosystème PHP

- ❑ L'écosystème PHP est en pleine effervescence, avec la montée en puissance de plusieurs projets importants, et l'arrivée prochaine de projets très attendus.
- ❑ Parmi les projets à venir:
  - Zend Framework 3 et PHP7, annoncés pour le second semestre 2015
- ❑ Deux projets majeurs à signaler pour les développeurs PHP, projets qui ont un fort impact sur la manière de concevoir les applications PHP:
  - *Composer (Le gestionnaire de composants qui met tout le monde d'accord)*
  - *Les PSR (PHP Standard Recommendations) du PHP-Fig*



[www.php-fig.org](http://www.php-fig.org)

PHP Framework Interop Group

# Composer



- ❑ Composer est un gestionnaire de composants PHP qui intègre aussi un gestionnaire de dépendances.
- ❑ Composer est un projet open-source développé lui-même en PHP et disponible sous forme d'un fichier `composer.phar`.
- ❑ Composer utilise un fichier `composer.json` dans lequel le développeur définit la liste des composants qu'il souhaite utiliser au sein d'un projet PHP. A partir de ce fichier, Composer va télécharger automatiquement les composants indiqués, ainsi que les dépendances associées.
- ❑ Exemple de fichier `composer.json` :

```
{
    "name" : "test",
    "require" : {
        "monolog/monolog" : "=1.13.1"
    },
    "repositories" : [ {
        "type" : "composer",
        "url" : "https://packagist.org/"
    } ]
}
```

Nom du projet (ici « test »)

Liste des composants à intégrer au projet

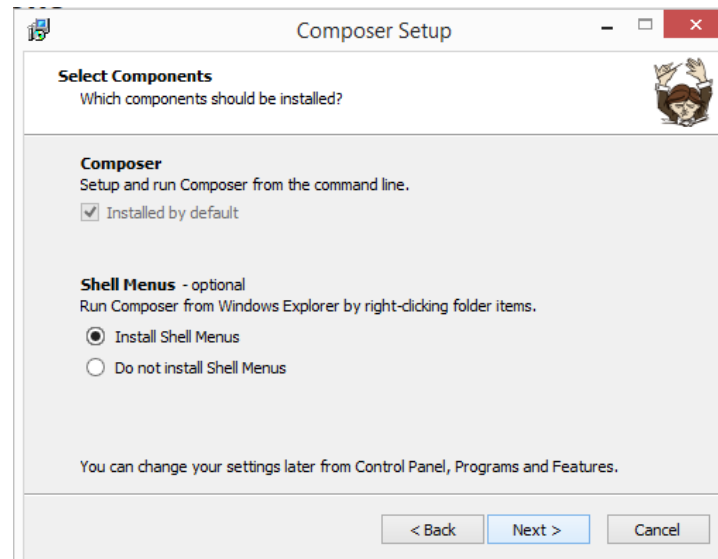
Exemple de composant (ici « monolog/monolog ») avec la version requise (ici 1.13.1)

Définition du dépôt où les composants vont être récupérés par Composer (ici, un dépôt de type Composer, hébergé sur Packagist.org)

# Composer



- ❑ On peut utiliser Composer avec les assistants de Zend Studio, ou en mode « ligne de commande ». Dans ce dernier cas, le télécharger sur le site suivant :
  - <https://getcomposer.org>
- ❑ Petite particularité de la version Windows de Composer: les développeurs sous Windows peuvent – en faisant un clic-droit sur un répertoire – accéder à un menu contextuel regroupant les principales options de Composer (cf. option « install shell menus » du Setup de Composer) :

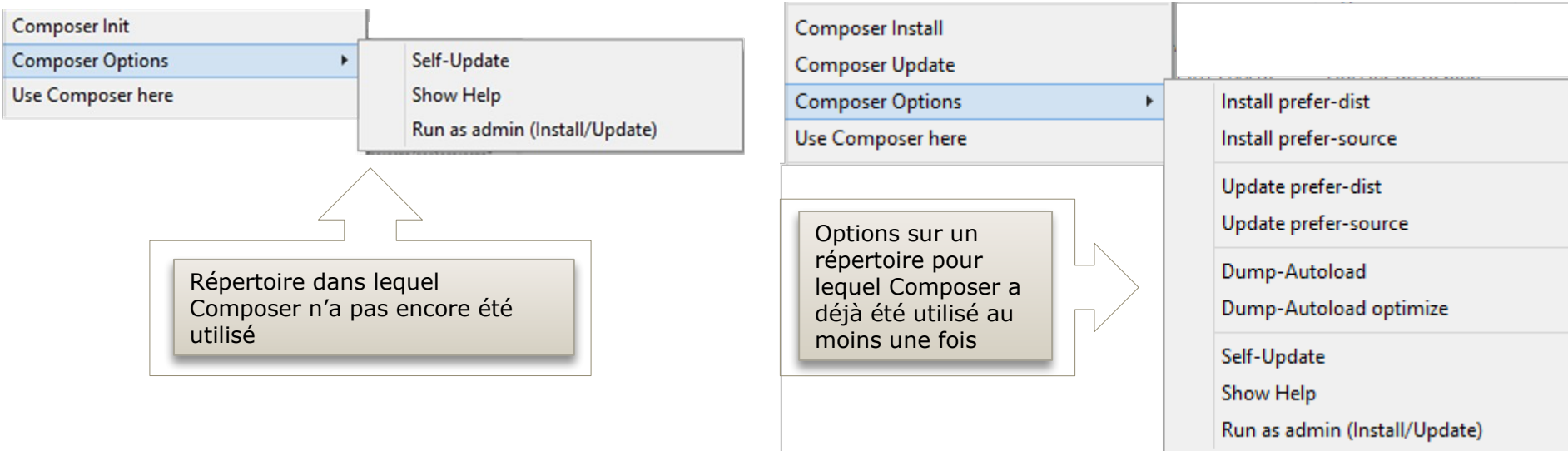


Ce « menu shell » n'existe pas dans les versions Linux et iOS de Composer, elle est donc rarement indiquée dans les tutos, forums et autres blogs.

# Composer



Le menu contextuel diffère selon l'état du répertoire sélectionné :



Essayez l'option « Use Composer here » sur un répertoire de test :

```
C:\> composer require monolog/monolog

Basic usage: composer <command>
For more information just type "composer".

D:\test3>composer require monolog/monolog
```

```
C:\Windows\System32\cmd.exe

D:\test3>dir
Le volume dans le lecteur D s'appelle Data
Le numéro de série du volume est 2AFC-44B7

Répertoire de D:\test3
19/05/2015  21:21    <DIR>          .
19/05/2015  21:21    <DIR>          ..
19/05/2015  21:21                62 composer.json
19/05/2015  21:21             4 809 composer.lock
19/05/2015  21:21    <DIR>          vendor
                2 fichier(s)                4 871 octets
                3 Rép(s)  475 112 050 688 octets libres

D:\test3>
```

# Composer

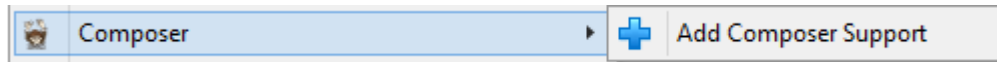


- ❑ Composer crée automatiquement un fichier appelé `composer.lock` qui stocke la version exacte de chaque composant intégré dans le projet. Si vous partagez votre projet avec d'autres développeurs et que le fichier `composer.lock` y est inclus, ils installeront automatiquement les mêmes versions de composants que vous. Pour mettre à jour les composants avec leurs toutes dernières versions, il faut le demander explicitement via la commande
  - `composer update <composant>`
- ❑ Composer génère aussi un « autoloader » permettant d'utiliser simplement les composants définis au sein d'un projet PHP

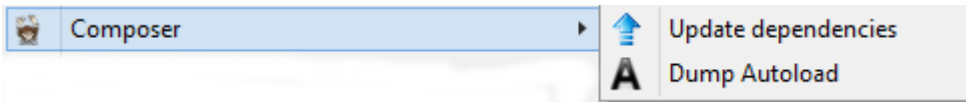
# Composer & Zend Studio



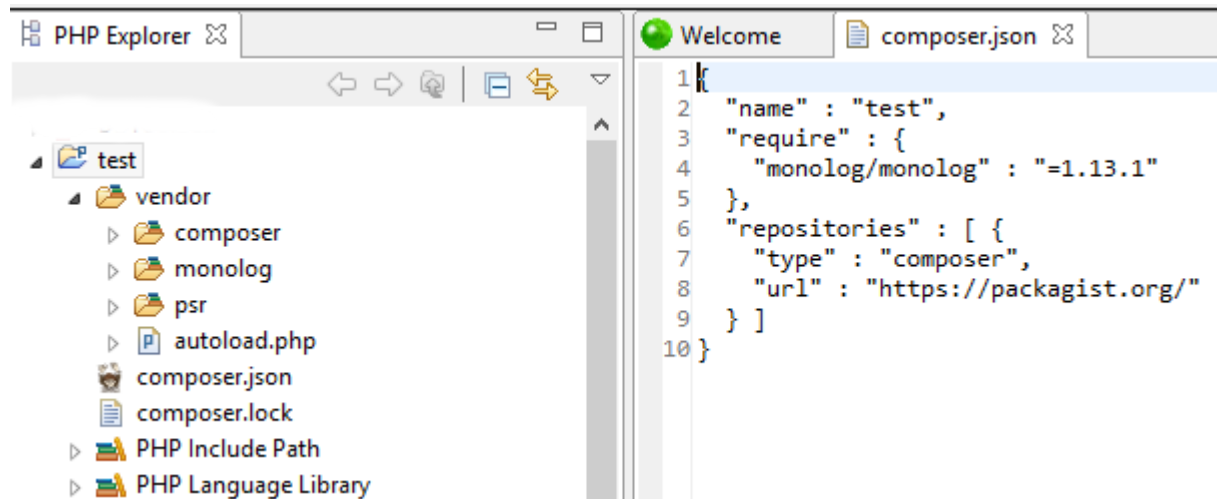
- ❑ Composer est intégré dans Zend Studio
- ❑ Pour l'activer, faites un clic-droit sur votre projet PHP :



- ❑ Si vous avez déjà utilisé Composer sur votre projet, l'option est alors un peu différente :



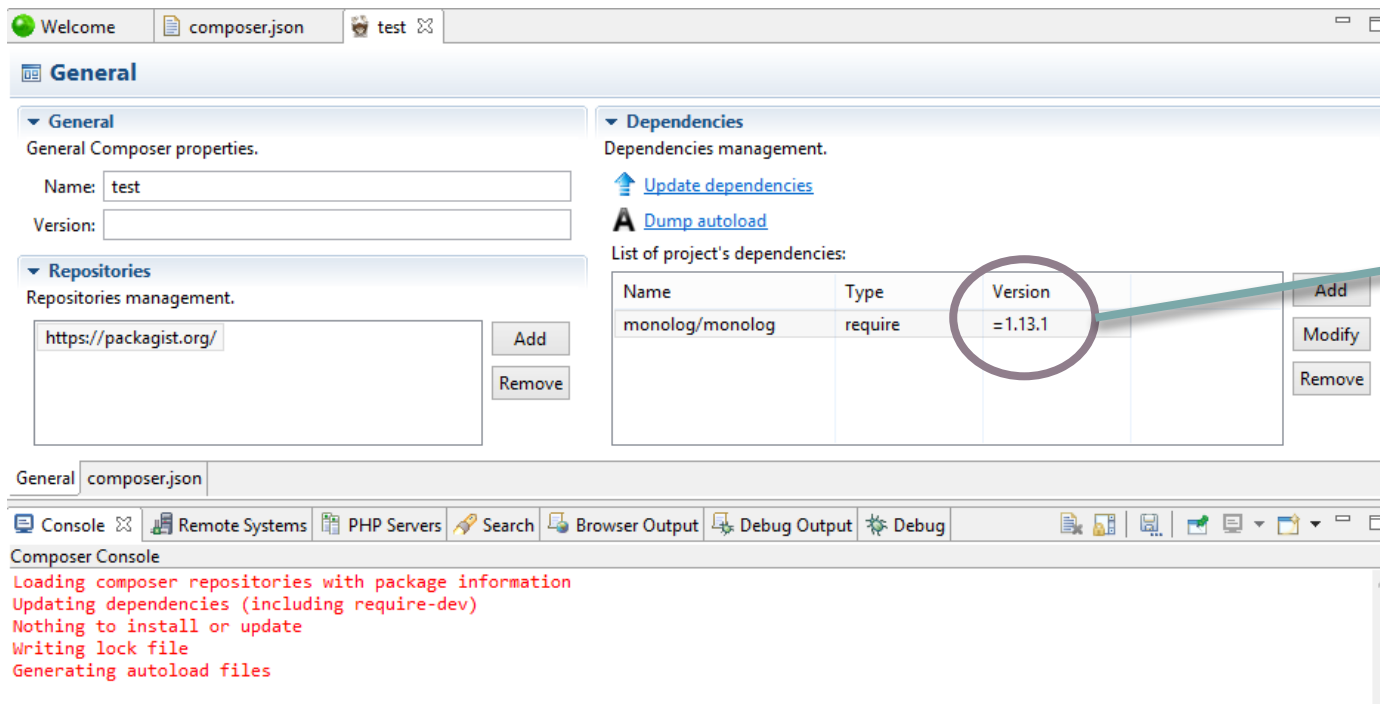
- ❑ Exemple de projet PHP dans lequel le composant « monolog » a été intégré et téléchargé via Composer (et l'assistant de Zend Studio) :



# Composer & Zend Studio

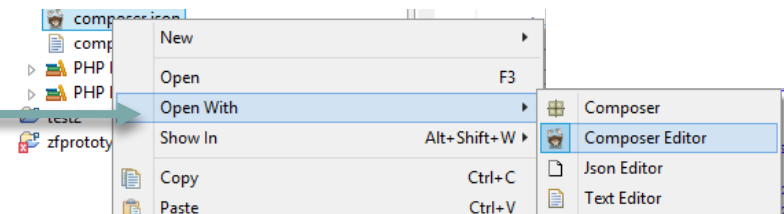


- ❑ L'assistant Composer intégré dans Zend Studio facilite la maintenance des composants et de leurs dépendances :



Pour chaque composant, on peut définir la version que l'on souhaite utiliser, en précisant le critère de sélection (=, >, <, etc..)

NB : l'assistant de Zend Studio ci-dessus s'affiche quand on ouvre le fichier composer.json avec l'option « Composer Editor »






# Composer & Packagist



- ❑ Composer seul n'est pas très utile. Il faut le coupler avec un annuaire de composants, en l'occurrence le projet Packagist :
- ❑ <https://packagist.org>

[Create a new account](#) | [Login](#)



## Packagist

The PHP package archivist.

[Submit Package](#)

Packagist is the main [Composer](#) repository. It aggregates all sorts of PHP packages that are installable with Composer.  
[Browse packages](#) or [submit your own](#).

Veuillez compléter ce champ.

### Getting Started

#### Define Your Dependencies

Put a file named `composer.json` at the root of your project, containing your project dependencies:

```
{
  "require": {
    "vendor/package": "1.3.2",
    "vendor/package": "1.3.2",
    "vendor/package": "1.3.2"
  }
}
```

### Publishing Packages

#### Define Your Package

Put a file named `composer.json` at the root of your package, containing this information:

```
{
  "name": "your-vendor-name/package-name",
  "description": "A short description of your package"
}
```

# Composer & Packagist



- ❑ Attention : Packagist est un annuaire de composants, PAS un hébergeur de composants. La grande majorité des projets qu'il référence sont hébergés chez Github.
- ❑ En conclusion :
  - Composer est en passe de réussir là où PEAR a échoué : mettre en place un annuaire de composants PHP faciles à utiliser et gérant simplement les dépendances. Composer est adopté par la majorité de la communauté PHP (et notamment par les développeurs des projets Zend Framework 2, CakePHP, Laravel, Symfony...).
- ❑ Pour s'initier au fonctionnement de Composer :
  - <http://www.grafikart.fr/tutoriels/php/composer-480>
  - <http://devzone.zend.com/5363/how-use-composer-with-external-tools/>
  - [http://eilgin.github.io/php-the-right-way/#composer\\_et\\_packagist](http://eilgin.github.io/php-the-right-way/#composer_et_packagist)
- ❑ Le livre « Modern PHP », de Josh Lockhart (éd. O'Reilly) contient une excellente introduction à l'utilisation de Composer.

# PHP et les PSR

- ❑ Le PHP-FIG (PHP Framework Interoperability Group), est un groupe de travail qui s'est constitué durant la conférence php|tek 2009, pour réfléchir à certaines problématiques récurrentes rencontrées par les développeurs PHP.
- ❑ Ce travail de réflexion a abouti à la publication des PSR (*PHP Standard Recommendations*).
- ❑ Les PSR sont des recommandations visant à faciliter l'intégration de composants d'origine diverse au sein d'un même projet. Le respect de ces recommandations facilite l'intégration de composants via Composer.
- ❑ Les PSR se déclinent en plusieurs parties
  - PSR-0 : traite du chargement des classes PHP et de l'autoloading
  - PSR-1 : fixe les conventions minimales de codage
  - PSR-2 : définit le style d'écriture et l'organisation du code
  - PSR-3 : définit l'interface des loggers.
  - PSR-4 : proposition de renforcement de PSR-0
- ❑ PSR-0 est aujourd'hui dépréciée au profit de PSR-4
- ❑ PSR-4 est compatible avec l'autoloader de Composer pour charger l'intégralité des classes d'un projet.
- ❑ Le livre « Modern PHP », de Josh Lockhart (éd. O'Reilly) contient une excellente introduction aux PSR.



Autoloading Standard  
PSR-0



Basic Coding Standard  
PSR-1



Coding Style Guide  
PSR-2



Logger Interface  
PSR-3



Improved Autoloading  
PSR-4

[www.php-fig.org](http://www.php-fig.org)

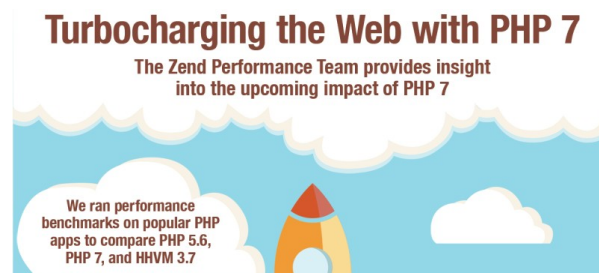
PHP Framework Interop Group

# Spécification PHP

- ❑ Le langage PHP qui a 20 ans, n'avait pas -malgré l'existence d'une abondante documentation- de spécification officielle, contrairement à de nombreux autres langages
- ❑ Facebook a, de sa propre initiative, rédigé une spécification complète de PHP, spécification dont la société avait besoin pour développer son moteur PHP HHVM (concurrent du Zend Engine).
- ❑ Facebook a livré cette spécification en open-source à la communauté PHP, et elle a été adoptée par les développeurs du langage PHP, en partenariat avec Zend, pour le développement du nouveau moteur PHPNG (PHP 7).
- ❑ Dépôt Github hébergeant la spécification :
  - <https://github.com/php/php-langspect/tree/master/spec>

# Des perfs améliorées avec PHP7

- ❑ Rasmus Lerdorf a présenté le résultat de benchmarks mettant en concurrence le moteur de PHP5, le moteur HHVM de Facebook, et le nouveau moteur PHPNG de PHP7. (*conférence Fluent de San Francisco, du 21 avril 2015*)
  - > HHVM et PHPNG y sont souvent au coude à coude, mais dans de nombreux cas PHPNG l'emporte. On constate également de très nets gains de performance de PHPNG par rapport au moteur de PHP5.
  - <http://talks.php.net/fluent15#/>
- ❑ Zend a également publié, à la mi mai 2015, une infographie dans laquelle on retrouve la synthèse de leurs propres benchmarks :
  - (lien plus disponible)



- ❑ NB : si vous êtes intéressés par la conférence Fluent de San Francisco, l'intégralité des conférences est téléchargeable pour bien moins que le prix d'un billet d'avion :
  - <http://shop.oreilly.com/product/0636920041009.do>

# PHP Toolkit

- ❑ Le site internet Young i Professionals est un projet communautaire visant à promouvoir l'intégration de technologies open-source sur la plateforme IBM i, et en particulier de PHP Toolkit (qui s'appuie sur XMLService).
  - <http://www.youngiprofessionals.com>
- ❑ Parmi les nouveautés à signaler :
  - L'évolution du connecteur PDO pour IBM i, qui intègre depuis peu le support des listes de bibliothèques, et le nommage système
    - <http://www.youngiprofessionals.com/wiki/index.php/XMLSERVICE/PHPPDOChangeLog>
  - L'évolution du connecteur ibm\_db2, qui intègre quelques améliorations, notamment pour le support de l'UTF-8
    - <http://www.youngiprofessionals.com/wiki/index.php/XMLSERVICE/PHPDB2ChangeLog>
  - Des améliorations de perfs au niveau du PHP Toolkit
    - <http://yips.idevcloud.com/wiki/index.php/XMLService/XMLSERVICEChangeLog>




YiPs #2

- PHP
- Ruby
- Node.js
- RPG
- GCC
- CCSID
- PASE
- Databases
- Apache
- XMLService
- General



# PHP Toolkit

- ❑ A signaler également, la disponibilité d'un outil développé avec ZF2, et accessible en ligne, destiné à faciliter l'interfaçage du PHP Toolkit avec des programmes RPG, via l'analyse des spécifications D d'un programme RPG :
- ❑ <http://yips.idevcloud.com/Samples/DSpecTool/>

 D Spec Tool Map

Map RPG To PHP Toolkit

PGM (optional)  LIB (optional)

Input: RPG D Spec cut/paste fixed format 'D' specs (example below)

```
D *Zrgebnisdaten:
D $vevsods          ds              occurs (200)
D $vsukz            1              1
D $vpos             2              9
D $vtxt             10             39
D $vkalw            40             174  2 dim(15)
D $vvsww            175            309  2 dim(15)
D $vvsk             310            324  0 dim(15)
D *
D i                 S              10i 0 inz(0)
```

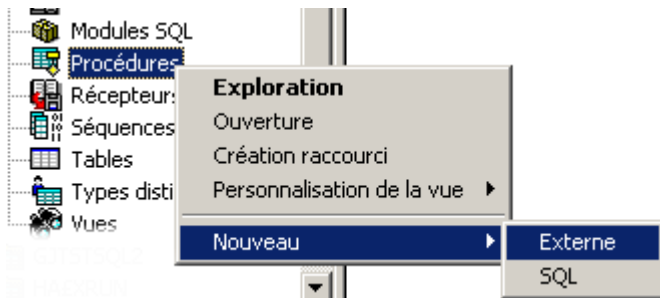
Generate PHP Stub (verbose) Generate PHP Stub (ignore "S") Generate PHP Functions (ignore "S") Clear Form

Output: PHP Toolkit (RPG order of appearance, manually re-order PHP statements)

```
// *****
// Procedure: ProgramA (pref_build)
// *****
$result = null;
$param = null;
$result[] = $tk->AddParameterPackDec("both",5,0,"pref_build_result","pref_build_result",0.0);
$param[] = $tk->AddParameterChar("both",1,"webpgmtype","webpgmtype","");
$output = $tk->PgmCall("PGM", "", $param, $result, array("func"=>"ProgramA"));
```

# Procédures stockées DB2 externes

- ❑ Ce n'est pas une nouveauté, mais un rappel : les procédures stockées DB2 de type externe constituent une solution idéale pour faire dialoguer programmes IBM i (RPG, Cobol, CL, ..) et applications webs (PHP, Java, Ruby, NodeJS, Python..).
- ❑ Le logiciel System i Navigator d'IBM fournit un assistant très pratique pour générer ce type de procédure
- ❑ Officiellement depuis la V7R1 (mais en réalité déjà disponible en V6R1), les programmes RPG sont en mesure de produire des « result sets » DB2, que les procédures stockées externes sont capables de retransmettre aux applications appelantes.
- ❑ Aucun outil complémentaire à télécharger et à installer sur votre IBM i, les procédures stockées externes sont intégrées en standard dans votre base DB2 for i
- ❑ Simple à mettre en œuvre et très performant



```
CREATE PROCEDURE MABIB.PGMREFPRC (  
    IN CODBIB CHAR(10) ,  
    IN CODPGM CHAR(10)  
)  
DYNAMIC RESULT SETS 1  
LANGUAGE RPGLE  
SPECIFIC MABIB.PGMREFPRC  
NOT DETERMINISTIC  
MODIFIES SQL DATA  
CALLED ON NULL INPUT  
EXTERNAL NAME 'MABIB/PGMREFPRG'  
PARAMETER STYLE GENERAL ;
```



# Procédures stockées DB2 externes

- ❑ Exemple de code RPG Free permettant de générer un result set à partir d'une table temporaire créée par le programme RPG

```
sql1 = 'SELECT HORO, RES FROM QTEMP/MODHERATMP ' ;  
EXEC SQL  
    PREPARE REQ1 FROM :sql1 ;  
EXEC SQL  
    DECLARE C1 CURSOR FOR REQ1 ;  
EXEC SQL  
    OPEN C1 ;  
EXEC SQL  
    SET RESULT SETS CURSOR C1 ;
```

- ❑ Exemple de code PHP appelant une procédure stockée externe et récupérant dans la foulée le result set produit par le programme RPG sous-jacent :

```
$stmt = $dbh->prepare("CALL MAPROCD2 (:numadh, :coderr, :horodat)");  
$stmt->bindParam(':numadh', $vnumadh, PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT, 7);  
$stmt->bindParam(':coderr', $vcoderr, PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT, 2);  
$stmt->bindParam(':horodat', $vhorodat, PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT, 27);  
// Appel de la procédure stockée  
if ($stmt->execute()) {  
    do {  
        $row_data = $st->fetchAll(PDO::FETCH_ASSOC);  
        if ($row_data) {  
            $nb lignes = count($row_data) ;  
            for($yIndex = 0; $yIndex < $nb lignes; $yIndex++) {  
                var_dump($row_data[$yIndex]) ;  
            }  
        }  
    } while ($st->nextRowset()); // passage au "result set" suivant  
}
```

# Zend Server – Les nouveautés

- ❑ Présentation des principales nouveautés relatives aux versions 7 et 8 de Zend Server :
  - Pour Zend Server 7 :
    - <http://www.zend.com/en/products/server/whats-new-7>
  - Pour Zend Server 8 :
    - <http://www.zend.com/en/products/server/whats-new>
- ❑ Nouveauté majeure de la version 7 : Z-Ray.
- ❑ La version 8 apporte un certain nombre d'améliorations, notamment autour de Z-Ray, ainsi que de nouveaux SDK.

# Zend Server et Z-Ray

- ❑ Pour présenter certaines fonctionnalités de Z-Ray, nous nous appuierons sur une application de type CRUD développée avec Zend Framework 1

CRUD/ZF1

Accueil

Graphiques

Test JQuery&AJAX

Login

## Exemple de CRUD

Cette application s'appuie sur les composants suivants :

- Zend Framework v1.12.9
- jQuery v1.11.3
- jQuery UI v1.11.4
- Twitter Bootstrap v2.3.2
- Modernizr v2.6.2

CRUD/ZF1

Accueil

Dossiers

Dossiers vJQuery

Graphiques

Test JQuery&AJAX

Logout

» Créer un dossier

Critères de sélection

Code soc.  égal

Recherche

Export XLS

Export DOC

Export PDF

< Précédent

1

2

3

4

5

6

7

8

9

10

Suivant >

CRUD : acronyme anglais pour Create-Retrieve-Update-Delete

# Zend Server et Z-Ray

- ❑ Z-Ray se matérialise, côté navigateur, par la barre d'état suivante:



- ❑ Les problèmes de performance se situant souvent au niveau de la base de données, les développeurs apprécieront l'icône en forme de disque dur :

http://localhost:10090/zfprototype4/public/dossiers

### Database Queries

Showing 1-4/4 20 ▾ Expand all

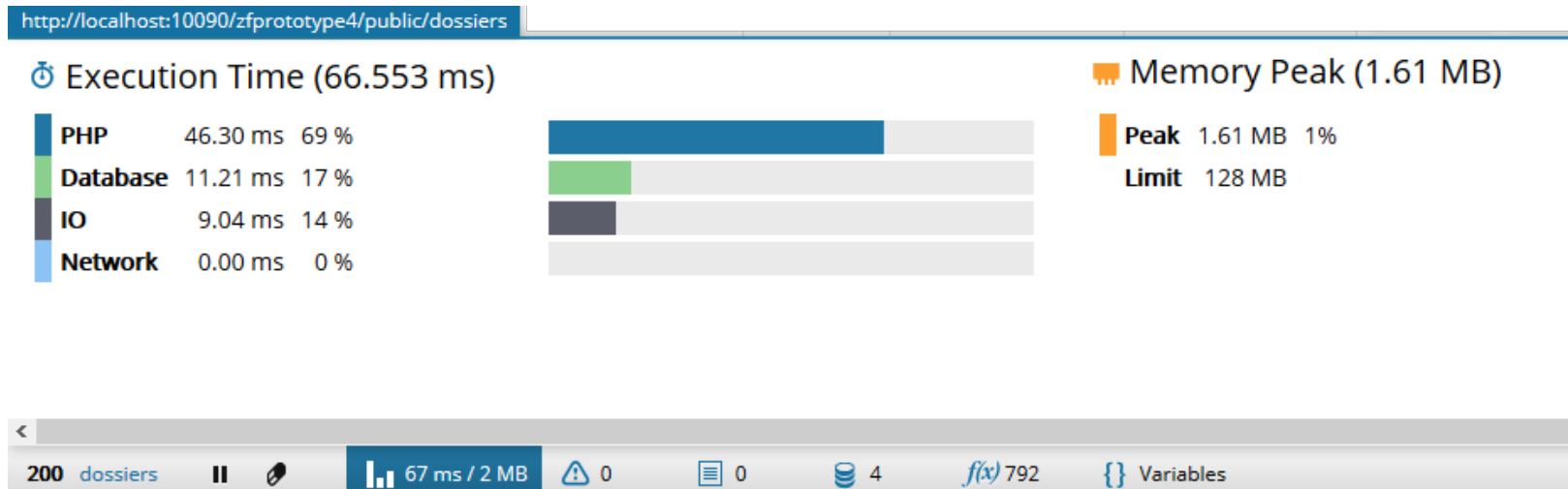
Show	Query	Time (ms)	Rows	Success
<input checked="" type="checkbox"/> All (4) 11.18 ms	1 set names utf8	0.23	0	OK
<input checked="" type="checkbox"/> Describe (1) 9.79 ms	2 DESCRIBE `zfdbproto`.`dossier`	9.79	15	OK
<input checked="" type="checkbox"/> Select (2) 1.17 ms	3 SELECT COUNT(1) AS `zend_paginator_row_count` FROM `zfdbproto`.`dossier` WHERE (statut <> 'D' ) LIMIT 1	0.45	1	OK
<input checked="" type="checkbox"/> Set (1) 0.23 ms	4 SELECT `dossier`.*, (encours_mnt + solde_brut) AS `mnt_total`, (select t b.libelle from prm_cod_soc b where b.id = societe_id) AS `lib_socie`	0.72	10	OK

< >

200 dossiers || 64 ms / 2 MB 0 0 4 f(x) 830 {} Variables

# Zend Server et Z-Ray

- L'option « execution time and memory peak » permet de voir instantanément ce qui consomme le plus de temps et de ressources, dans la page courante :



# Zend Server et Z-Ray

- ❑ L'option d'affichage des variables :

http://localhost:10090/zfprototype4/public/dossiers

## { PHP Variables

Show

			Key	Value
<input checked="" type="checkbox"/>	All	(107)	▶ \$_REQUEST	Array[4]
<input checked="" type="checkbox"/>	New Entries	(3)	▼ \$_POST	Array[4]
<input checked="" type="checkbox"/>	Unchanged	(104)	submit	Recherche
			typ_rech	*eq
			val_rech	3
			zon_rech	societe_id
			\$_GET	Array[0]

200 dossiers || 179 ms / 2 MB 0 0 4 f(x) 792 { Variables

# Zend Server et Z-Ray

- ❑ L'option d'affichage des fonctions permet de distinguer :
  - les appels aux fonctions standard du langage PHP, spécifiques de l'appli (Custom), et spécifiques du framework (ZF)

http://localhost:10090/zfprototype4/public/dossiers

f(x) Functions

Showing 1-20/738 20

Show (Manage Function Groups)

	Function	Count	Duration Inclusive (ms)	Duration Exclusive (ms)	Average (ms)	Defined at
<input type="checkbox"/> All (792) 101.57 ms	54 rows are not displayed					
<input checked="" type="checkbox"/> (Custom) (23) 3.77 ms	{main}	1	175.89	3.03	175.89	
<input type="checkbox"/> (PHP Native) (54) 1.51 ms	Zend_Controller_Front::dispatch()	1	162.20	0.35	162.20	Front.php:835
	Zend_Controller_Dispatcher_Standard::dispatch()	1	157.73	0.10	157.73	Standard.ph...
<input checked="" type="checkbox"/> Zend_ (715) 96.29 ms	Zend_Controller_Action::dispatch()	1	154.29	0.09	154.29	Action.php:4...
	DossiersController::indexAction()	1	133.19	0.37	133.19	DossiersCon...

- ❑ L'option d'affichage des erreurs et avertissements:

http://localhost:10090/zfprototype4/public/dossiers

Errors & Warnings

Showing 1-1/1 20

Show ☐ @Silenced (0) Type

	Message	Filename	Created	Ac...
<input checked="" type="checkbox"/> All 1 Exception	File "Zend\Controller\Router\Rewrite.php" does not exist or class "Zend_Controller_Router_Rewrite" was not found in the file	Loader.php:87	22:02:47	
<input checked="" type="checkbox"/> Exception 1				

200 dossiers 102 ms / 2 MB 0 1 4 f(x) 87 zendServer

# Zend Server et Z-Ray

- ❑ Les nouvelles options (ou « widgets ») de la barre de Z-Ray sur Zend Server version 8:
  - Le widget « Variables » change de nom et devient « Request Info »
  - Z-Ray détecte le type de framework utilisé (ici ZF 1), proposant des widgets supplémentaires spécialisés sur ZF :

The screenshot displays the Z-Ray interface in Zend Server. The top toolbar contains several widgets: **Request Info**, **Plugin (2)**, **Request (1)**, **Request Object (16)**, **View (3)**, and **View Helpers (79)**. The **Request (1)** and **View (3)** widgets are highlighted with green boxes. A curved arrow points from the **Request (1)** widget to the **ZF Request** panel at the bottom. Another arrow points from the **View (3)** widget to the **ZF View** panel on the right.

**ZF Request** panel (bottom left):

URL: <http://localhost:10090/zfprototype4/public/dossiers>

Action	Controller	ModuleClassName
indexAction	DossiersController	DossiersController

**ZF View** panel (bottom right):

URL: <http://localhost:10090/zfprototype4/public/dossiers>

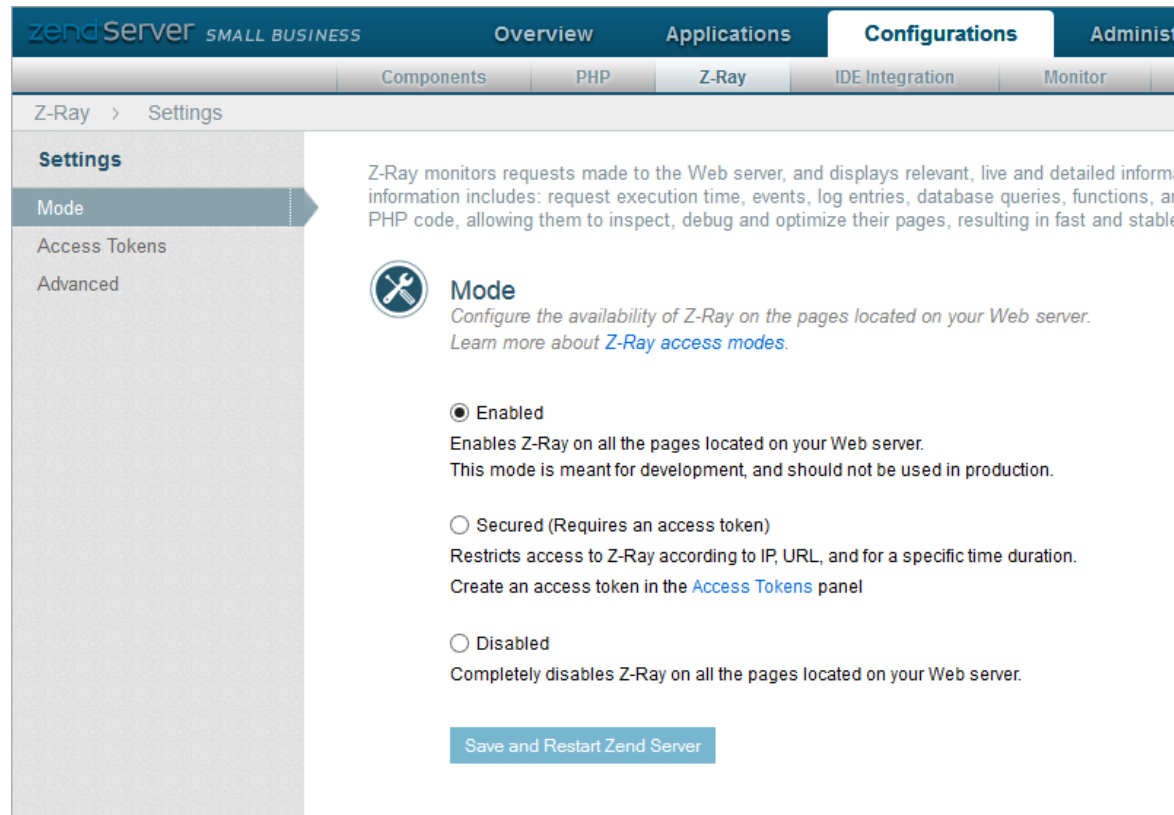
Showing 1-3/3 | 20

- 0
- D:/htdocs/zfprototype4/application/views/scripts/partials/pagination.phtml
- D:/htdocs/zfprototype4/application/views/scripts/dossiers/index.phtml
- ../application/layouts/layout.phtml



# Zend Server et Z-Ray

- ❑ Au niveau de l'interface d'administration, Z-Ray offre la possibilité de filtrer les IP et URL des applications pour lesquelles le tracking de Z-Ray est actif :



The screenshot displays the Zend Server administration interface. The top navigation bar includes 'zendServer SMALL BUSINESS', 'Overview', 'Applications', 'Configurations', and 'Administ'. Below this, a sub-navigation bar shows 'Components', 'PHP', 'Z-Ray', 'IDE Integration', and 'Monitor'. The main content area is titled 'Z-Ray > Settings'. On the left sidebar, 'Settings' is selected, with sub-options 'Mode', 'Access Tokens', and 'Advanced'. The 'Mode' section is active, showing a description: 'Z-Ray monitors requests made to the Web server, and displays relevant, live and detailed information includes: request execution time, events, log entries, database queries, functions, and PHP code, allowing them to inspect, debug and optimize their pages, resulting in fast and stable'. Below this, the 'Mode' configuration is shown with three radio buttons: 'Enabled' (selected), 'Secured (Requires an access token)', and 'Disabled'. Each mode has a brief description. At the bottom, there is a 'Save and Restart Zend Server' button.

zendServer SMALL BUSINESS Overview Applications **Configurations** Administ

Components PHP **Z-Ray** IDE Integration Monitor

Z-Ray > Settings


**Settings**

Mode

Access Tokens

Advanced

Z-Ray monitors requests made to the Web server, and displays relevant, live and detailed information includes: request execution time, events, log entries, database queries, functions, and PHP code, allowing them to inspect, debug and optimize their pages, resulting in fast and stable

 **Mode**  
Configure the availability of Z-Ray on the pages located on your Web server.  
[Learn more about Z-Ray access modes.](#)

☒ **Enabled**  
Enables Z-Ray on all the pages located on your Web server.  
This mode is meant for development, and should not be used in production.

☐ **Secured (Requires an access token)**  
Restricts access to Z-Ray according to IP, URL, and for a specific time duration.  
Create an access token in the [Access Tokens](#) panel

☐ **Disabled**  
Completely disables Z-Ray on all the pages located on your Web server.

Save and Restart Zend Server

# Extensions à Z-Ray




















- ❑ Des extensions pour Z-Ray, prépackagées et téléchargeables librement :
  - <https://github.com/zend-server-extensions>
- ❑ Tutorial pour créer vos propres extensions:
  - <https://github.com/zend-server-extensions/Z-Ray-Documentation>
- ❑ Exemples de code pour vous aider à démarrer :
  - <https://github.com/zend-server-extensions/Z-Ray-Samples>
- ❑ Une vidéo d'introduction :
  - <http://devzone.zend.com/4273/developing-z-ray-extension/>

# ZS, SDK et WebAPI

- ❑ Zend Server embarque un certain nombre d'API que vous pouvez utiliser dans vos propres projets, au travers du protocole REST :
  - [http://files.zend.com/help/Zend-Server/zend-server.htm#web\\_api\\_reference\\_guide.htm](http://files.zend.com/help/Zend-Server/zend-server.htm#web_api_reference_guide.htm)
- ❑ Vous pouvez utiliser ces API pour interroger Zend Server sur divers sujets :

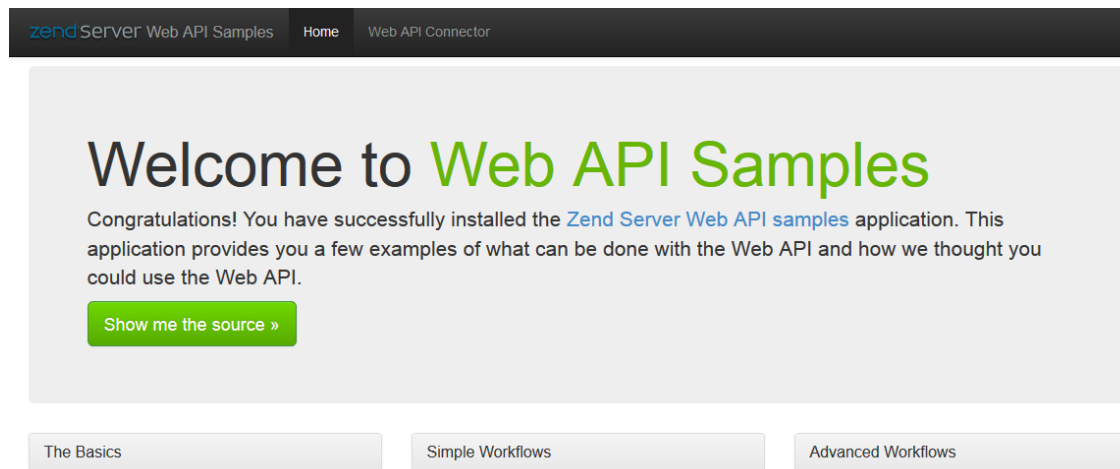


## Available API Methods

- |  |   |
|--|---|
|  Administration Methods           |  Monitor Methods                       |
|  Audit Methods                    |  Monitoring Rules Methods              |
|  Cache Methods                    |  Notification Center Methods           |
|  Codetracing Methods              |  Server and Cluster Management Methods |
|  Configuration Management Methods |  Statistics Methods                    |
|  Deployment Methods              |  Studio Integration Methods            |
|  Filter Methods                 |  Page Cache Methods                  |
|  Job Queue Methods              |  URL Insight Methods                 |
|  Library Methods                |  Virtual Hosts Methods               |
|  |  Z-Ray Methods                       |

# ZS, SDK et WebAPI

- ❑ Un SDK permettant d'utiliser les WebAPI du Zend Server :
  - <https://github.com/zend-patterns/ZendServerSDK>
- ❑ Pour vous familiariser en douceur avec l'utilisation de ce SDK, démarrez avec le projet « WebAPI Samples » :
  - <https://github.com/zend-patterns/SamplesWebAPI>
- ❑ Téléchargez et dézippez le projet, installez le sur votre Zend Server et lancez le projet dans votre navigateur :
  - <http://localhost:10090/SamplesWebAPI-master/public/>



# ZS, SDK et WebAPI

- ❑ Le projet « WebAPI Samples » contient un répertoire « exemples »
- ❑ Pour tester chacun des scripts PHP de ce répertoire, ajouter 2 paramètres dans l'URL d'appel des scripts : name et key.

Exemple :

- <http://localhost:10090/SamplesWebAPI-master/public/examples/applicationGetStatus.php?name=ZendStudio&key=9c1926...>

## Example output

### Synopsis:

```
http://localhost:10090/SamplesWebAPI-master/public/examples/applicationGetStatus.php?key=<key>&name=<key-name>[&version=<version-number>]
[&output=<json|xml>]
```

### WebAPI call parameters:

```
Array
(
    [name] => ZendStudio
    [key] => 9c1926...780e9c
)
```

### URI called:

```
/ZendServer/Api/applicationGetStatus
```

### Headers sent:

```
Array
(
    [Host] => localhost:10081
    [Connection] => close
    [Accept-Encoding] => gzip, deflate
    [User-Agent] => Zend\Http\Client
)
```

Documentation relative à l'utilisation des WebAPI Keys :

[http://files.zend.com/help/Zend-Server/zend-server.htm#working\\_with\\_api\\_keys.htm](http://files.zend.com/help/Zend-Server/zend-server.htm#working_with_api_keys.htm)

# ZS, SDK, WebAPI et API Keys

- ❑ Pour faire appel aux WebAPI de Zend Server au travers du SDK, fournir aux méthodes d'appel des API, une clé composée d'un nom et d'une valeur cryptée (hash).
- ❑ Zend Server fournit au moins une clé de ce type, qui se trouve dans l'interface d'administration :

The screenshot shows the 'Administration' tab of the Zend Server PRO interface. The 'Web API' sub-tab is selected. A table lists the available WebAPI keys. One key is visible with ID 2, Name 'admin', and a partially visible hash 'b6a0...'. A green arrow points from the 'Add Key' button to the 'Name' column header.

ID	Name	Hash	Username	Creation Time
2	admin	b6a0...	...	07/Jan/2015 10:55:50

The screenshot shows the 'Administration' tab of the Zend Server SMALL BUSINESS interface. The 'Web API' sub-tab is selected. The 'Key Management' section is visible, showing a single key named 'ZendStudio' with a partially visible hash '9c192...0e9c'.

Key Name	Hash
ZendStudio	9c192...0e9c

Avec ZS PRO, on peut ajouter de nouvelles WebAPI Keys, contrairement à ZS Small Business

# Les SDK de Zend Server

- ❑ Attention, des projets de SDK pour Zend Server sont hébergés chez Google:
  - Projet apparemment tombé en désuétude :
    - <https://code.google.com/p/zend-sdk/>
  - SDK pour utilisation des WebAPI par les développeurs Java :
    - <https://code.google.com/p/zend-webapi/>
  - Or, pour les développeurs Java, le SDK officiel est le suivant :
    - <https://github.com/zendtech/zendserver-sdk-java>
- ❑ Je vous recommande donc d'utiliser les projets hébergés sur Github, indiqués ci-dessus et dans les précédentes diapos (voir aussi la récap sur la diapo suivante).

# Les SDK de Zend Server

- ❑ Récapitulatif des principaux projets Zend (et dépôts Github) :
  - Site officiel pour la documentation du Zend Server
    - <http://files.zend.com/help/Zend-Server/zend-server.htm>
  - Framework pour gérer plusieurs instances Zend Server au travers d'applications PHP
    - <https://github.com/zend-patterns/zend-server-framework>
  - CLI ZF2 pour la création de packages (zpk) et l'utilisation de WebAPI
    - <https://github.com/zend-patterns/ZendServerSDK>
  - Zend Server Web API samples application
    - <https://github.com/zend-patterns/SamplesWebAPI>
  - Jeu de classes ZF2 pour la création de "hook" sur les fonctions de déploiement de Zend Server
    - <https://github.com/zend-patterns/ZendServerDeploymentHelper>
  - Module ZF2 pour faciliter l'utilisation des API Zend Server
    - <https://github.com/zend-patterns/ZendServerWebApiModule>
  - Récapitulatif des SDK dédiés au Continuous Delivery :
    - <http://www.zend.com/en/solutions/continuous-delivery/sdks>
  - Vidéos :
    - <https://www.youtube.com/user/DocumentationZend/videos>
    - [https://www.youtube.com/watch?v=s3WI\\_XRKdJA](https://www.youtube.com/watch?v=s3WI_XRKdJA)



- ❑ Apigility est un projet d'aide à la construction d'API.
- ❑ Apigility est open-source, soutenu par Zend, et développé avec ZF2, AngularJS et Bootstrap.
- ❑ Lead Developer : Matthew Weier O'Phinney (également lead dev. de ZF)
- ❑ Site officiel :  
<https://www.apigility.org>
- ❑ Pour l'installer, on peut télécharger le zip, ou mieux encore, on peut laisser Composer faire le travail de téléchargement et d'installation:

Positionnement dans le répertoire du serveur web local :

```
cd d:\htdocs
```

Création du projet apigilityskl :

```
composer create-project -sdev zfcampus/zf-apigility-skeleton apigilityskl
```

Positionnement dans le répertoire du projet nouvellement créé:

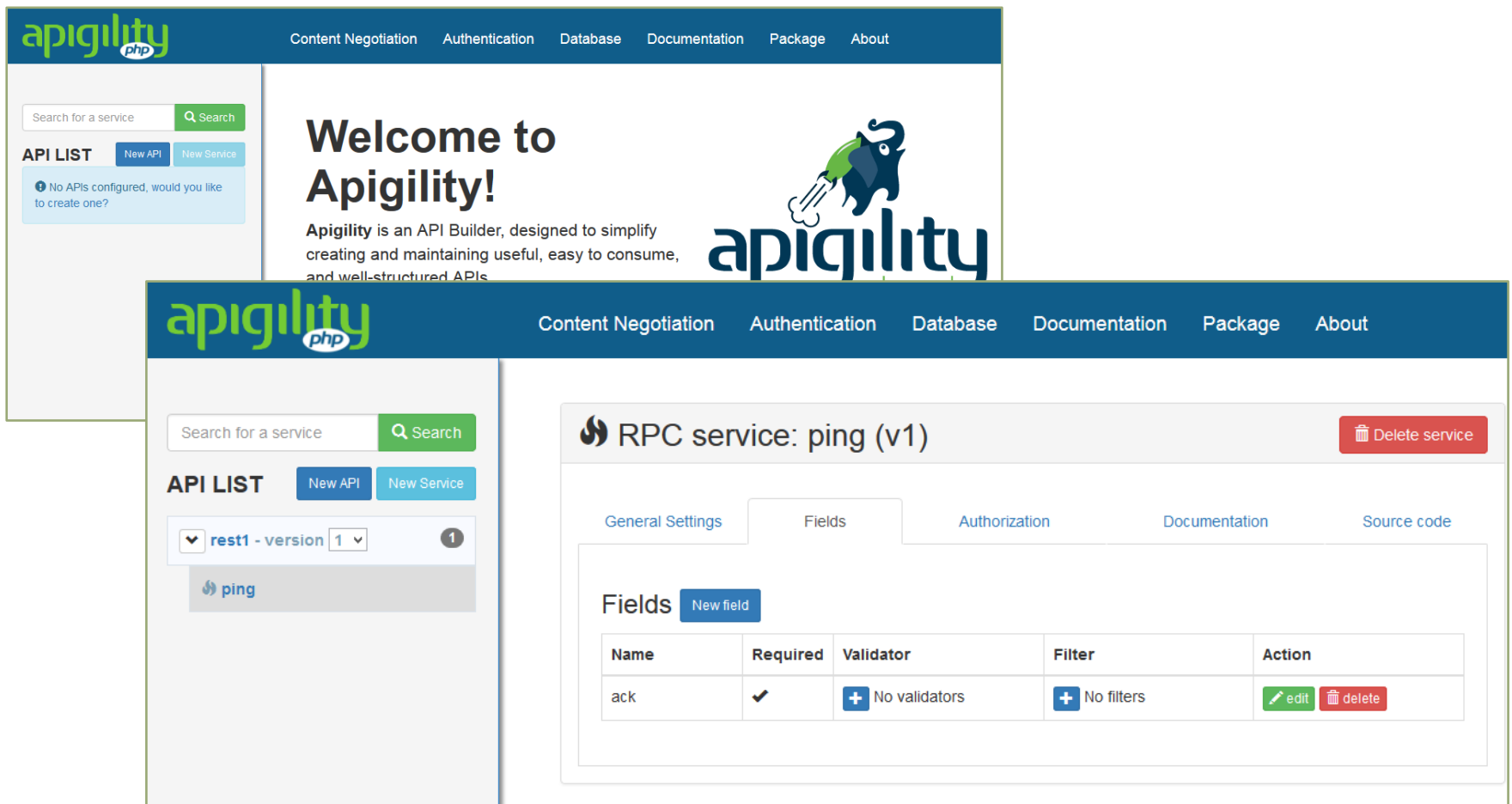
```
cd apigilityskl
```

Passage du projet en mode développement :

```
php public/index.php development enable
```

Vous pouvez maintenant lancer l'application dans votre navigateur :

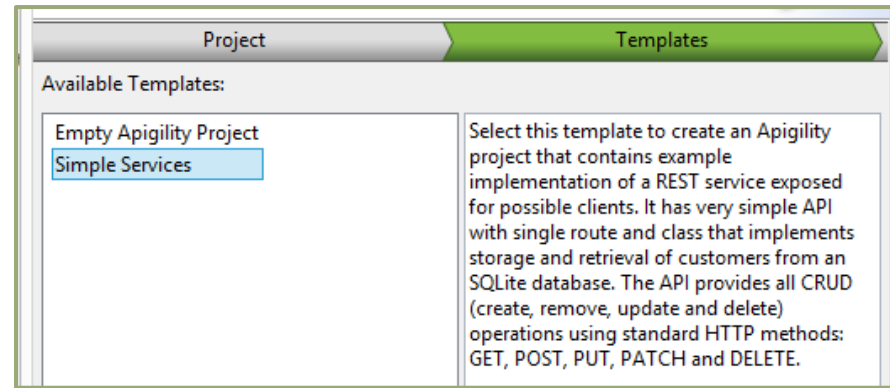
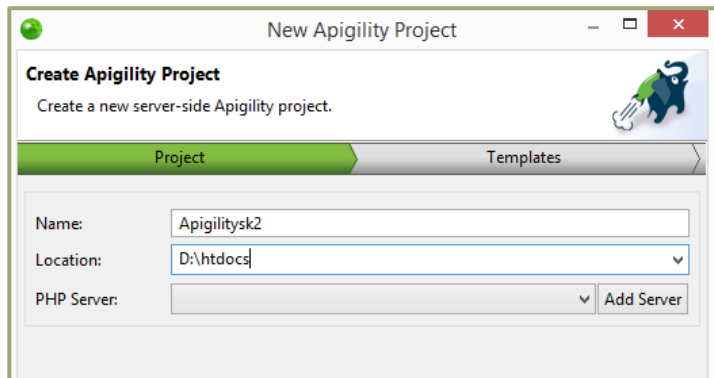
<http://localhost:10090/apigilityskl/public/>



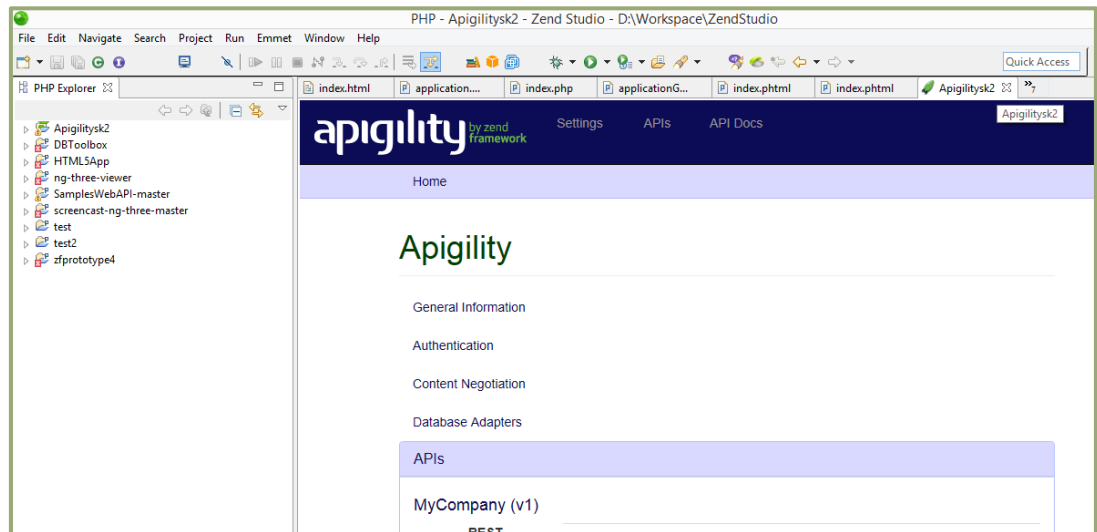
The screenshot displays the Apigility web interface. The top navigation bar includes links for Content Negotiation, Authentication, Database, Documentation, Package, and About. The main content area shows a welcome message and a description of Apigility as an API Builder. On the left, there is a sidebar with a search bar and a list of APIs. The 'API LIST' section shows a dropdown menu with 'rest1 - version 1' selected, and a 'ping' service listed below it. The 'Fields' tab is active, showing a table of fields for the 'ping' service.

Name	Required	Validator	Filter	Action
ack	✓	+ No validators	+ No filters	<a href="#">edit</a> <a href="#">delete</a>

- On peut aussi créer un projet Apigility en passant par l'assistant dédié de Zend Studio



On obtient un projet Apigility directement intégré dans Zend Studio, avec un look un peu différent de celui obtenu au moyen de Composer (cf. diapo précédente), mais avec les mêmes fonctionnalités.



# Apigility

- ❑ Vidéos d'introduction :
  - Apigility Integration in Zend Studio
    - <https://www.youtube.com/watch?v=9L9wf12t370>
  - Developing Mobile Applications in Zend Studio
    - <https://www.youtube.com/watch?v=IJVh6myWRRM>
  - Autres :
    - [https://www.youtube.com/results?search\\_query=apigility](https://www.youtube.com/results?search_query=apigility)



# Continuous I & D ?

Terme anglais	Terme francisé	Abréviation
Continuous Integration	Intégration Continue	CI
Continuous Delivery	Livraison Continue	CD?
Continuous Deployment	Déploiement continu	CD?



Kézako ?

# Continuous Integration (CI)

- ❑ *L'intégration continue est un ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée. /.../ Le principal but de cette pratique est de détecter les problèmes d'intégration au plus tôt lors du développement. De plus, elle permet d'automatiser l'exécution des suites de tests et de voir l'évolution du développement du logiciel. ([https://fr.wikipedia.org/wiki/Int%C3%A9gration\\_continue](https://fr.wikipedia.org/wiki/Int%C3%A9gration_continue) )*

Pour clarifier :

- ❑ L'idée est de tester votre code aussi souvent que possible pour rattraper rapidement les problèmes (on est ici dans un contexte de développement agile).
- ❑ CI implique que les développeurs travaillent sous le contrôle d'un gestionnaire de versions (Subversion ou Git).
- ❑ La plupart du travail de contrôle de la non régression est effectué par des tests automatisés, et cette technique nécessite donc la mise en place de tests unitaires.
- ❑ A la fin du travail de contrôle, le code est déployé sur un serveur d'intégration.
- ❑ Pour la petite histoire, CI aurait été utilisée par IBM dès les années 60, pour le développement de l'OS/360

(Source : <http://www.commentcamarche.net/faq/24318-concept-de-l-integration-continue> )

# Continuous Delivery & Deployment

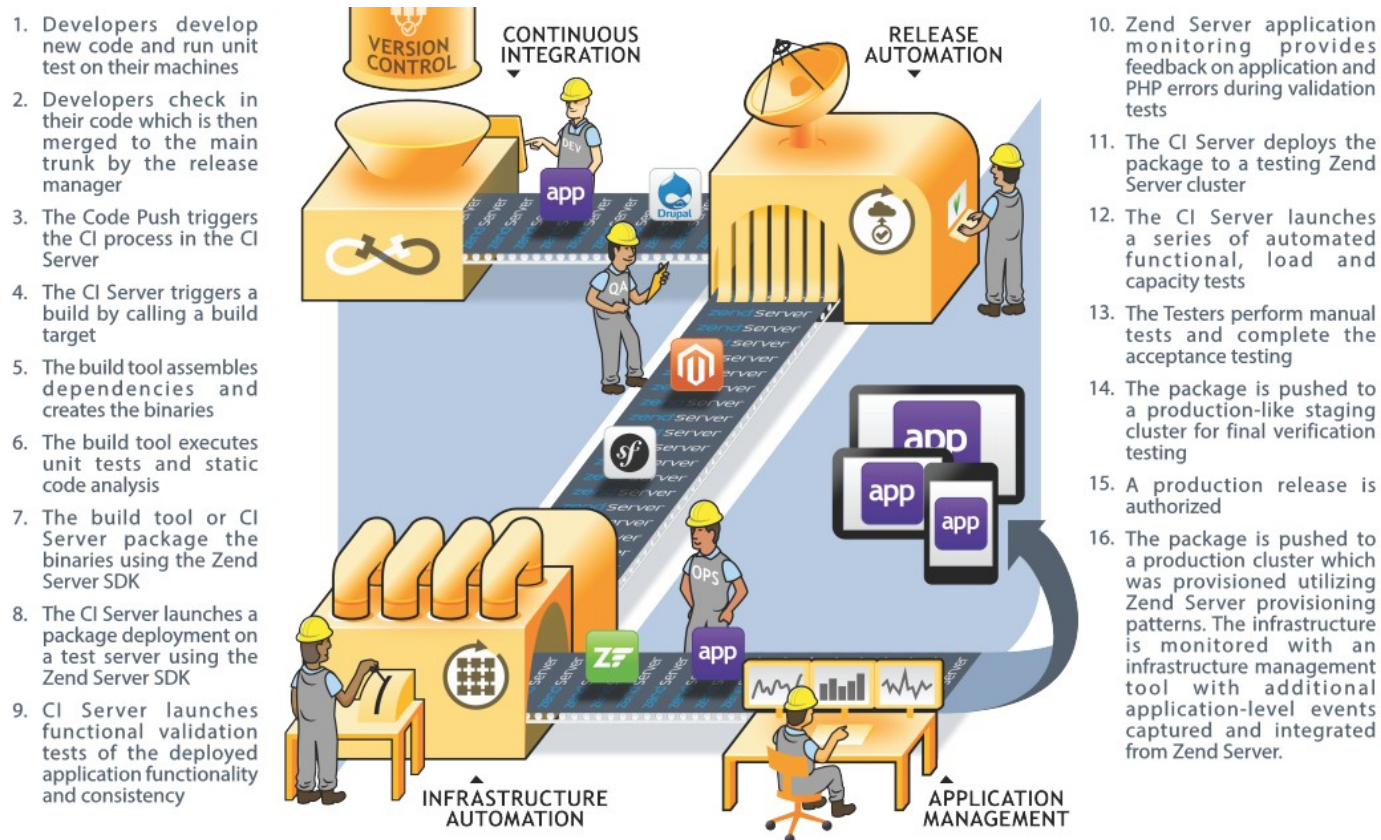
- ❑ Le « Continuous Delivery » (en français : « Livraison continue ») est une approche consistant à **automatiser** et à **fiabiliser** le processus de livraison des logiciels.
- ❑ L'idée est de livrer le **plus souvent possible**, de **petits changements**, plus faciles à circonscrire et à corriger en cas d'anomalie. Eviter autant que possible les grosses livraisons, moins fréquentes et plus difficiles à gérer en cas de régression et de retour-arrières.

Pour clarifier :

- ❑ Continuous Delivery (livraison continue) et Continuous Deployment (déploiement continu) sont souvent confondus, alors que ce sont deux étapes distinctes
- ❑ Mais dans les faits, ces 2 étapes sont quelquefois réellement confondues, car très rapprochées dans le temps
- ❑ En très résumé, on pourrait dire que :
  - Le Continuous Delivery est le CI complété d'une étape supplémentaire: la livraison du code à l'équipe (ou à l'outil) en charge du déploiement
  - Le Continuous Deployment est l'étape finale, consistant à déployer l'application sur un ou plusieurs serveurs de production
- ❑ Une vidéo sur la problématique du Continuous Deployment, de Benoit Lafontaine à l'USI:
  - <https://www.youtube.com/watch?v=UcDtH5s406M>

# Zend et l'intégration continue

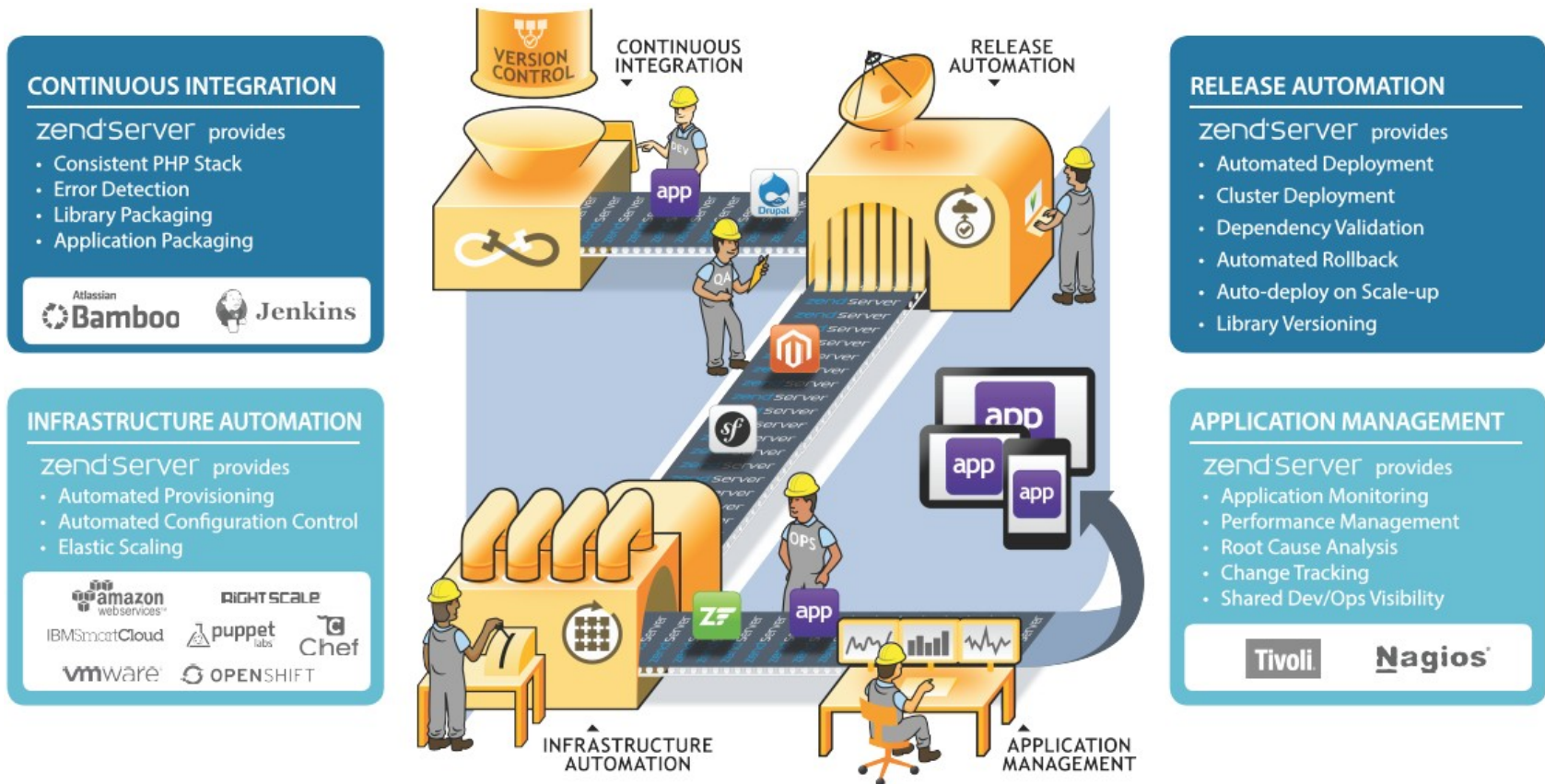
- ❑ Dossier proposé par Zend sur l'intégration continue avec Zend Server: <http://www.zend.com/en/services/continuous-delivery>
- ❑ Ci-dessous un des schémas proposés dans le « blueprint » de Zend :





# Zend et l'intégration continue

- Un panel des solutions qu'il peut être nécessaire de mettre en œuvre dans le cadre d'un processus d'intégration continue. (tiré du « blueprint » de Zend)



# Continuous I&D

- ❑ Qui pratique l'intégration, le déploiement et la livraison continue, aujourd'hui ?

En premier lieu les géants du web (cf. vidéo de Ludovic Cinquin à l'USI)

<http://www.usievents.com/en/talks/42-les-secrets-des-geants-du-web-7-pratiques-pour-mieux-travailler-ludovic-cinquin>



A lire également : <http://blog.octo.com/lean-startup>

# Conclusion

L'année 2015 sera une année passionnante pour les développeurs.

- ❑ Année de transition pour la communauté PHP avec la découverte et/ou le renforcement de certains usages (Composer, Packagist, HTML5, Bootstrap, Git et Github, Apigility, Z-Ray, Continuous Integration..).
- ❑ Des attentes très fortes sur de nouveaux venus, annoncés pour le second semestre 2015:
  - Côté serveur: les très attendus **PHP7** et **ZF3**
  - Côté navigateur: une nouvelle version de **Javascript** ([norme ECMAScript 6](#)), ainsi qu'**AngularJS 2** (co-développé par Google et Microsoft)
- ❑ On notera qu'Apigility constitue un bel exemple – à suivre - de projet réussissant l'intégration de technos diverses (ZF2, AngularJS, HTML5 et Bootstrap).

# Des Questions ?

Fragen

Issues

Cuestiones

Problemi

Temoj

Isu-Isu

問題

문제

Málefni

teeb meem

Exitibus

Qüestions

# **That's All Folks !!**

---

😊 Merci beaucoup 😊