

Flex & Zend AMF

Bonjour,

Je vais vous expliquer dans ce tutoriel comment utiliser le package AMF du framework Zend dans sa dernière version.

L'objectif de ce tutoriel est de vous permettre d'utiliser Zend AMF de la même manière que vous l'auriez fait avec amfphp.

Nous verrons un simple exercice de type « Hello World ».

À propos

Le framework Zend est depuis peu le moyen officiellement supporté par Adobe pour la communication entre Flash, Flex, AIR et les services PHP.

L'intégration d'AMF dans Zend est dirigée par Wade Arnold, celui-là même qui a pris les commandes d'amfphp dans le cadre du développement de la version 2.0.

Son blog : <http://wadearnold.com/blog/>

Vous y trouverez beaucoup de ressources et les avancées du framework.

Ce tutoriel n'abordera pas en détail les fonctionnalités du Zend Framework, ni n'expliquera son fonctionnement. Il suppose que vous ayez déjà une certaine connaissance du framework, l'objectif étant de vous montrer son utilisation dans le cadre du remoting avec Flex.

Pour parfaire votre connaissance de Zend:

- <http://julien-pauli.developpez.com/tutoriels/zend-framework/presentation/>
- <http://g-rossolini.developpez.com/tutoriels/php/zend-framework/debuter/?page=controleur>

« Hello World »

Avant toute chose vous devez vous assurer que vous avez un environnement serveur sur votre ordinateur, pour cela vous aurez besoin de Wamp (si vous êtes sur Windows), de Mamp (si vous êtes sur Mac) ou encore de XAMP (si vous êtes sur Linux).

Ensuite, vous pouvez créer un nouveau projet Flex.

Pour cela, dans Flex, faites : File > New > Flex Project.

Donner lui un nom, choisissez Web Application et sélectionner PHP dans le menu déroulant Application Server Type.

Faites en sorte que votre répertoire de debug se trouve dans le dossier racine de votre serveur local.

A l'intérieur du dossier src, créer un nouveau dossier nommé « services ».

C'est à l'intérieur de ce dossier que nous allons placer tous nos fichiers PHP et le dossier contenant le Framework Zend.

Pour récupérer le Framework, rendez-vous à l'adresse :

<http://www.zend.com/community/downloads>

Sélectionner « Zend Framework 1.7 full », choisissez le package au format zip ou tar.gz et cliquer sur « Download ».

Une fois le Framework extrait, aller dans le répertoire « Library » et copier le dossier Zend dans le dossier « services » de votre projet Flex.

Lançons nous dans la création de votre bootstrap. Pour cela créer un fichier index.php. Nous allons commencer par activer le rapport d'erreur, grâce aux lignes suivantes :

```
error_reporting(E_ALL|E_STRICT);
ini_set("display_errors", "on");
```

Ensuite nous allons redéfinir l'include_path, de manière à inclure le répertoire contenant le framework Zend dans la recherche lors de l'appel de fonctions telles que require(), include(), etc...

```
set_include_path(
    . PATH_SEPARATOR . './Zend'
    . PATH_SEPARATOR . get_include_path());
```

Créer une nouvelle variable \$server et déclarer une nouvelle instance du serveur Zend Amf, d'appeler la méthode setClass() qui permet d'attacher des classes ou fonctions au serveur et ensuite de mettre dans un echo l'appel de la méthode handle().

Voici le code final d'index.php :

```
<?php
```

```
error_reporting(E_ALL|E_STRICT);
ini_set("display_errors", "on");
```

```
set_include_path(
    . PATH_SEPARATOR . './Zend'
    . PATH_SEPARATOR . get_include_path());
```

```
require_once 'Amf/Server.php';
require_once 'HelloWorld.php';
```

```
$server = new Zend_Amf_Server();
$server->setClass("HelloWorld");
```

```
echo($server->handle());
```

```
?>
```

Nous allons maintenant nous pencher sur la création de notre premier service « HelloWorld ».

C'est une simple classe php que nous nommons « HelloWorld » avec une seule méthode, getHelloWorld(), qui retourne une chaîne de caractère.

Voici le code final de HelloWorld.php

```
<?php
```

```

class HelloWorld
{
    public function __construct()
    {

    }

    public function getHelloWorld()
    {
        return "HelloWorld";
    }
}

```

Vous pouvez dès maintenant retourner sur Flex, et plus précisément ouvrir le fichier mxmxml application de votre projet.

Ajouter un écouteur d'événement `creationComplete` sur votre application, qui exécute la fonction `init()` lors du dispatch de cet événement.

```

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute"
creationComplete="init()">

```

Puis, créer un nouveau Remote Object et lui ajouter une méthode, `getHelloWorld`.

Voici le code :

```

<mx:RemoteObject id="myservice"
source="HelloWorld" destination="zend"
endpoint="http://localhost:8888/ZendBase/"
fault="faultHandler(event)">

<mx:method name="getHelloWorld" result="resultHandler(event)"/>

</mx:RemoteObject>

```

Nous allons maintenant créer les fonctions `init()`, `faultHandler(event)` et `resultHandler(event)` dans une balise `Script`.

`init()` appelle la méthode `getHelloWorld` du service distant `HelloWorld`.

`faultHandler()` est chargé d'afficher une alerte contenant le cas échéant un message d'erreur tandis que `resultHandler()` affiche une alerte contenant le message renvoyé par le serveur Zend AMF en cas de succès de la requête.

Voici le code de cette partie :

```

import mx.controls.Alert;
import mx.rpc.events.ResultEvent;
import mx.rpc.events.FaultEvent;

public function init():void
{
    myservice.getHelloWorld();
}

```

```

}

public function faultHandler(fault:FaultEvent):void
{
    Alert.show(fault.fault.faultCode.toString(), "FaultHandler");
}

public function resultHandler(result:ResultEvent):void
{
    Alert.show(result.result.toString(), "ResultHandler");
}

```

Voici le code final du mxml :

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute"
creationComplete="init()">

    <mx:RemoteObject id="myservice"
source="HelloWorld" destination="zend"
endpoint="http://localhost:8888/ZendBase/services/"
fault="faultHandler(event)">

        <mx:method name="getHelloWorld"
result="resultHandler(event)"/>

    </mx:RemoteObject>

    <mx:Script>
    <![CDATA[
        import mx.controls.Alert;
        import mx.rpc.events.ResultEvent;
        import mx.rpc.events.FaultEvent;

        public function init():void
        {
            myservice.getHelloWorld();
        }

        public function faultHandler(fault:FaultEvent):void
        {
            Alert.show(fault.fault.faultCode.toString(),
"FaultHandler");
        }

        public function resultHandler(result:ResultEvent):void

```

```
{
  Alert.show(result.result.toString(), "ResultHandler");
}

]]>
</mx:Script>
```

```
</mx:Application>
```

Et voilà, vous venez de créer votre premier service utilisant Zend Amf Server. C'est un exemple basique mais celui-ci vous permet de vous familiariser avec cette composante du Framework Zend.

Si vous connaissez bien ce Framework vous pourrez mettre à disposition de Flex toute sa puissance, notamment ces adaptateurs pour les bases de données ou encore intégrer votre serveur dans un contrôleur Zend afin de pouvoir gérer deux versions (html et flex par exemple) de votre site.

Liens utiles :

- <http://framework.zend.com/manual/fr/zend.amf.html>
- http://www.adobe.com/devnet/flex/flex_php.html
- <http://diamondtearz.org/blog/zend-amf-collection/>

Si vous avez d'autres questions, contactez moi sur : slyc3isback[at]me.com