Introduction à Silex

Silex, autoload, routage

Dans ce TD, vous allez vous familiariser avec les composants d'un MicroFramework : Silex qui contient les composants de base de Symfony.

Une introduction générale à ce sujet se trouve ici :
`http://silex.sensiolabs.org/documentation`

Exercice 1. composer, class-loader et http-foundation

Installons d'abord composer. On va d'abord créer un répertoire `bin` à la racine de notre HOME :

```
cd
mkdir bin
cd bin
curl -s https://getcomposer.org/installer | php
```

Vérifiez la définition de vos variables d'environnement http_proxy et https_proxy dans votre `.bashrc` Ajoutez également la ligne suivante à votre .bashrc :

```
export PATH=$PATH:~/bin
```

de manière à ce que tous les programme installés dans le répertoire bin de votre HOME soient accessibles de n'importe où. Puis créez un répertoire de travail silex dans votre dossier /www avec à l'intérieur un fichier composer.json :

```
{
    "require": {
        "symfony/class-loader": "3.3.*",
        "symfony/http-foundation": "3.3.*"
    }
}
```

Ceci indique que nous n'installons pour l'insant que ces 2 seuls composants. Puis utilisez composer.phar pour installer les composants demandés :

```
composer.phar update
```

L'autoloader et les 2 principaux composants de HttpFoundation à savoir `Request` et `Response` sont prêts à l'emploi. Remarquez l'usage des espaces de nommages en PHP semblables à ceux du C++ ou aux import de packages en java.

```php
<?php
require_once __DIR__.'vendor/autoload.php';

use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;

//$request = Request::createFromGlobals();

$request = Request::create('/essai.php?name=Zozo');

// URI demandee (sans les parametres)
$path=$request->getPathInfo();

// recup GET de variables
$nom=$request->query->get('name','World');
$prenom=$request->query->get('surname','Joe');
echo "Bonjour $surname $name<br/>";
?>
```

On peut aussi récupérer d'autres informations sur le Client et fabriquer une réponse :

```php
<?php
// recup  variables SERVER
$host=$request->server->get('HTTP_HOST');

// recup  COOKIES
$request->cookies->get('PHPSESSID');

// HTTP headers
$headers=$request->headers->get('host');
$content_type=$request->headers->get('content_type');

$method=$request->getMethod();  //GET, POST, PUT, DELETE ou
   HEAD
$langs=$request->getLanguages();
$IP == $request->getClientIp();
$response =
new Response($IP." ".$host." ".$path." ".$headers."
".$content_type." ".$method." ".$nom." ".$langs[0]);
$response->send();
?>
```

Exercice 2. Routage avec Silex

Ajoutons la demande suivante au fichier composer.json :

```
{
    "require": {
        "symfony/class-loader": "3.3.*",
        "symfony/http-foundation": "3.3.*",
        "silex/silex": "~2.4"
    }
}
```

En fait Silex possède les premiers packages en dépendances et va en installer quelques autres. Ajoutez le fichier .htaccess suivant dans votre répertoire www/silex/ :

```
FallbackResource /~login/silex/index.php
```

Si cela ne marche pas ou que vous avez une vieille version d'Apache :

```
Options -MultiViews
RewriteEngine On
RewriteBase /~login/silex
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^ index.php [QSA,L]
```

où vous remplacerez login par votre login. Installez ensuite le fichier index.php dans le dossier silex :

```
<?php

require_once __DIR__.'/vendor/autoload.php';

$app = new Silex\Application();
$app['debug']=true;

$app->get('/', function (){
    return 'Bonjour le Monde !';
});

$app->get('/hello/{name}', function ($name) use ($app) {
    return 'Hello '.$app->escape($name);
});

$app->run();

?>
```

Exercice 3. Routage avec Silex Les capacités de routage de Silex sont importantes et agréables à utiliser :

```php
$amis = array(
    1 => array(
        'NAISSANCE'      => '2000-03-29',
        'NOM'    => 'DOE',
        'PRENOM'     => 'CALVIN',
        'VILLE'      => 'Honolulu',
    ),
    2 => array(
        'NAISSANCE'      => '1992-05-27',
        'NOM'    => 'Yoyo',
        'PRENOM'     => 'Albert',
        'VILLE'      => 'Orleans',
    ),
);

$app->get('/contact', function () use ($amis) {
    $content = '';
    foreach ($amis as $ami) {
        $content .= $ami['PRENOM'].'  ';
        $content .= $ami['NOM'].' ';
        $content .= $ami['VILLE'].' ';
        $content .= '<br />';
    }

    return $content;
});
```

Et pour une route avec paramètre :

```php
$app->get('/contact/{id}',
  function (Silex\Application $app, $id) use ($amis) {
    if (!isset($amis[$id])) {
        $app->abort(404, "Le contact $id n'existe pas !");
    }

    $ami = $amis[$id];

    return  "<h1>{$ami['NOM']}</h1>".
            "<p>{$ami['VILLE']}</p>".
            "<p>{$ami['NAISSANCE']}</p>";
});
```

Et on peut également traiter les requêtes POST, PUT, DELETE avec Silex.

Exercice 4. CARNET avec Silex

Reprenons notre petit MVC pour en assurer le routage avec Silex. Proposez des routes pour afficher les contacts sous forme de liste de liens ou de tableau HTML, puis une route pour ajouter un nouveau contact et une autre pour en effacer un. Nous allons étendre cela en un véritable service REST dans le prochain TD.