EVALUATION

TP2

LabREST_03_Slim

Construire une API REST

Production

- Compressez votre projet au formation zip (sans le répertoire vendor).
- Déposez le fichier dans Teams.
- Nommage du fichier : Nom_Prenom_apiRestPHP_TP2.zip.

Construire une API REST

Objectif

- Développer l'application du TP1 en utilisant un framework PHP.
- Utilisation du micro framework PHP nommé SLIM.
  - https://www.slimframework.com/
- Des tutos
  - https://www.twilio.com/en-us/blog/create-restful-api-slim4-php-mysql
  - https://www.phpflow.com/php/create-simple-rest-api-using-slim-framework/

Construire une API REST

- Pour créer un projet avec le squelette (skeleton)

```
PS C:\xampp\htdocs> composer create-project slim/slim-skeleton [nom_du_projet]
```

- Pour exécuter l'application

```
PS C:\xampp\htdocs\nom_du_projet > php -S localhost:8080 -t public
```

```
PS C:\xampp\htdocs\nom_du_projet > composer start
```

- Accès à l'application : http://localhost:8080/

Construire une API REST

- Pour créer un projet sans le squelette (skeleton)

  https://www.slimframework.com/docs/v4/start/installation.html

```
PS C:\xampp\htdocs> mkdir [nom_du_projet]
```

```
PS C:\xampp\htdocs\nom_du_projet > composer require slim/slim:4.0.0
```

Construire une API REST

- Différentes implémentations de PSR-7

```
PS C:\xampp\htdocs\nom_du_projet > composer require slim/psr7
```

https://github.com/slimphp/Slim-Psr7

```
PS C:\xampp\htdocs\nom_du_projet > composer require nyholm/psr7 nyholm/psr7-server
```

https://github.com/Nyholm/psr7

```
PS C:\xampp\htdocs\nom_du_projet > composer require guzzlehttp/psr7 http-interop/http-factory-guzzle
```

https://github.com/guzzle/psr7

```
PS C:\xampp\htdocs\nom_du_projet > composer require laminas/laminas-diactoros
```

https://docs.laminas.dev/laminas-diactoros/

Construire une API REST

- ## PSR-7 : Http Message Interface

  Spécification définie par le PHP-FIG (PHP Framework Interoperability Group) qui fournit des interfaces standardisées pour manipuler les messages HTTP dans les applications PHP.

  Essentielle pour créer des applications web et des API RESTful car elle permet une interopérabilité entre différentes bibliothèques et frameworks.

  https://www.php-fig.org/psr/psr-7/

Construire une API REST

- PSR-7 : Http Message Interface

  Définit des interfaces liées à la représentation, sous forme d'objets et de messages HTTP :

  - Interface générique MessageInterface : permet de décrire simplement un message (avec un protocole, des en-têtes et un corps).
  - Interface RequestInterface : étend MessageInterface en y ajoutant les concepts de cible, de méthode et d'URL.
  - Interface ServerRequestInterface : étend RequestInterface avec la gestion des paramètres d'url, des cookies, des fichiers uploadés, des données postées et un système d'attributs permettant de passer des informations.
  - Interface ResponseInterface : étend MessageInterface pour représenter une réponse à renvoyer au navigateur (ou reçu suite à un appel à une API) qui ajoute simplement la notion de statut.
  - Interface StreamInterface : permet de représenter un flux de données et est utilisé pour décrire le corps d'un message.
  - Interface UriInterface : permet de représenter un lien (schéma, nom d'hôte et chemin).
  - Interface UploadedFileInterface : représente un fichier uploadé reçu par le serveur et est utilisé par l'interface ServerRequestInterface.

Construire une API REST

- nyholm/psr7

  Bibliothèque PHP pour les messages HTTP, compatible avec la spécification PSR-7. Elle offre une implémentation des interfaces PSR-7 pour gérer les requêtes et les réponses HTTP, avec un focus sur la performance et la simplicité.

- laminas-diactoros

  Bibliothèque PHP conçue pour gérer des objets de requêtes et réponses HTTP en conformité avec les spécifications PSR-7 (HTTP Messages).

- guzzlehttp/psr7

  Bibliothèque PHP qui fournit une implémentation des interfaces PSR-7 pour gérer les messages HTTP. Souvent utilisée avec Guzzle pour simplifier la manipulation des requêtes et réponses HTTP. Elle peut être utilisée de manière indépendante dans n'importe quel projet PHP nécessitant une conformité PSR-7.

Construire une API REST

- Exemple : utilisation de Slim-Psr7
- Créez le répertoire de votre projet

```
PS C:\xampp\htdocs> mkdir [nom_du_projet]
```

- Créez le répertoire nommé public

```
PS C:\xampp\htdocs\nom_du_projet > composer require slim/slim:4.0.0
```

- Installez le framework Slim

```
PS C:\xampp\htdocs\nom_du_projet> mkdir public
```

- Installez l'implémentation Slim-psr7

```
PS C:\xampp\htdocs\nom_du_projet > composer require slim/psr7
```

Construire une API REST

- Exemple : utilisation de Slim-Psr7

- Dans le répertoire public, créez un fichier nommé index.php
- Copiez le code ci-dessous dans le fichier index.php

```php
<?php
use Psr\Http\Message\ResponseInterface as Response;
use Psr\Http\Message\ServerRequestInterface as Request;
use Slim\Factory\AppFactory;

require __DIR__ . '/../vendor/autoload.php';
$app = AppFactory::create();
$app->get('/', function (Request $request, Response $response, $args) {
    $response->getBody()->write("Hello Bob!");
    return $response;
});
$app->run();
```

Construire une API REST

- Pour exécuter l'application

```
PS C:\xampp\htdocs\nom_du_projet\src > php -S localhost:8080 -t public
```

- Accès à l'application : http://localhost:8080/

- Implémentez, via le framework Slim, votre application basée sur le TP1.