

Développement d'un framework PHP

Fonctionnalités

- **Générales :**
 - Autoloader (PSR-4)
 - Conteneur / dépendance d'injection (PSR-11, Container-Interop/Service-Provider)
 - Abstraction de BDD : PDO
 - Cache (PSR-16)
 - Config : chargement et registre
 - Debug
 - Http : abstraction de la couche HTTP (PSR-7, PSR-15, PSR-17)
 - Log (PSR-3)
 - Utilitaires : classes outils.
 - Routage : routage de requête http.
 - Session : gestionnaire de session (abstraction de \$_SESSION)
- **Ligne de commande :**
 - Cli + Script
- **Web services :**
 - Serveur REST
 - Service REST
 - Client REST
- **Web socket / Long polling :**
 - Event loop
 - Serveur de websocket / Long polling (?)
- **Application web :**
 - HMVC : Contrôleur + Modèle + Vue + Dispatcher
 - Templating : Php
 - URI : via PSR7 HTTP/URI
 - I18N
- **** Autres :****
 - Auth : via Json Web Token.
 - Acls : gestion des droits.

Dépendances

- PSR-3 : Psr/Log v1.0
- PSR-4 Autoloader :
- PSR-7 : Psr/Http/Message v1.0
- PSR-11 : Psr/Container v1.0
- PSR-15 : HTTP Middlewares
- PSR-16 : Simple Cache
- PSR-17 : HTTP Factories
- Container-Interop/Service-Provider : <https://github.com/container-interop/service-provider>

Sources

- Core : [PSR-4](#) (Autoloader)
- Cache : [PSR-16](#), [PhpFastCache](#)
- Config :
- DependencyInjection : [Simplex](#)
- HTTP : [Slim 3](#) & [Guzzle PSR7](#)
- I18N : [Symfony Translation Component](#)
- Rest : [Slim3](#) (serveur) & [Guzzle PSR7](#) (client)
- Session : [Symfony Http Foundation Component](#)

Ressources

Software As a Service

- <https://12factor.net/>

ACLs

Docs :

- [Symfony 3 Guard](#)

Dependency Injection/Service Location

Codes sources de Conteneurs :

- [Pimple](#)
- [Unbox](#)
- [Container Interop](#) : standard d'interopérabilité entre Conteneur (DI)

Docs :

- [Do you need a Dependency Injection Container ?](#)
- [Aura DI](#)
- [PHP-DI](#)
- [Laravel Container doc](#)
- [Phalcon DI doc](#)

Notes :

- Voir le pattern Service Locator et la différence avec un DI Container.
- Exemple de DIC performants : Dice, Aura and Orno
- Problème éventuel et solution : Using abstract factories with Slim 3
- Aura gère les paramètres dans les appels aux classes du conteneur. C'est la solution retenue pour le Framework.

Config

- **INI :**
 - [IniParser](#) : Parser de fichier ini ajoutant des améliorations
 - [Configula](#) : Parser multi formats : PHP (\$config), JSON (json_decode()), INI (parse_ini_file()), YAML (symfony/yaml package)
- **** YAML ** :**
 - [Symfony YAML](#) : Yaml 1.2
 - [Spyc](#) : Yaml 1.1

Cache

- [phpfastcache](#)
- [Simple-PHP-Cache](#)

Console (CLI)

Codes sources :

- [Symfony Console code source](#)
- [Laravel Console code source](#)

Docs :

- [Console Wars - PHP CLI Libraries](#)
- [Symfony Console doc](#)
- [Laravel Artisan doc](#)

Database

Principes intéressants :

- <https://github.com/auraphp/Aura.Sql>
- <https://github.com/dg/dibi>
- https://docs.phalconphp.com/en/latest/api/Phalcon_Db_Adapter_Pdo.html

Autres possibilités :

- <http://culttt.com/2012/10/01/roll-your-own-pdo-php-class/>
- https://github.com/blag001/class_easy_pdo/blob/master/toolSql/Bdd.class.php
- <https://github.com/ttsuruoka/php-simple-dbi/blob/master/src/SimpleDBI.php>

I18N

- [Symfony translation doc](#)
- [Symfony translation code source](#)
- [PHP gettext](#)

Logger

- [Chrome Logger](#) : permet d'afficher les messages de log côté Console du navigateur.

MVC

- Voir : <https://docs.phalconphp.com/en/latest/reference/mvc.html>

MVC : dispatcher

- Dispatching is the process of taking the request object, extracting the module name, controller name, action name, and optional parameters contained in it, and then instantiating a controller and calling an action of that controller.
- Voir : https://docs.phalconphp.com/en/latest/api/Phalcon_Mvc_Dispatcher.html

MVC : hiérarchique (HMVC)

- [Introduction au HMVC](#)
- [Phalcon : vues hiérarchiques](#)
- [Kohana : vues hiérarchiques](#)

REST : Client

- [Guzzle](#) : PSR-7

REST : Serveur

- [Slim 3](#) : PSR-7

Router

- Routing is the process of taking a URI endpoint (that part of the URI which comes after the base URL) and decomposing it into parameters to determine which module, controller, and action of that controller should receive the request
- Source :
 - FastRoute : plusieurs solutions de routage, très performant... trop de fichiers...
 - AltoRouter : très simple 1 fichier ! Modifier le Router actuel selon ce principe mais avec l'analyse des Regexp de Fastroute
- Voir :
 - https://docs.phalconphp.com/en/latest/api/Phalcon_Mvc_Router.html
 - <http://laravel.com/docs/5.1/routing>

Websocket

Code source :

- [React PHP - Event loop](#) : fourni une boucle d'évènement (*event loop*), "Event-driven, non-blocking I/O with PHP".

Docs :

- [Ratchet](#)

From:

<https://memos.clapas.org/> - **Memos**

Permanent link:

<https://memos.clapas.org/informatique/developpement/projets/mircro-framework-php>

Last update: **2020/05/22 20:08**

