

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Apollina Saint Ouen |  | PAQUET Judicaël |  | Judicael.paquet@gmail.com  https://github.com/las93 |

Apollina 3.0 PHP

[Le nouveau système de templating PHP]

Table des matières

1. [Présentation 1](#_Toc409472878)

[Installation](#_Toc409472879)

[Concept du routeur](#_Toc409472880)

# Présentation

Apollina est une nouvelle alternative de système de templating de qualité de développement open-source dans les environnements LAMP.

## Installation

L’installation du système de templating est relativement simple. Vous pouvez télécharger la version .zip sur le Github <https://github.com/las93> ou ajouter las93/apollina dans votre composer.json.

## Concept du routeur

Xxx

# Utilisation basique

Pour bien utiliser Attila, voici les premières étapes à respecter afin de proposer le meilleur des ORM à vos différents projets.

## XXX

XXX l’ORM

# Utilisation avancée

Voici la liste des utilisations avancées d’Apollina qui pourront beaucoup vous aider dans vos différents projets.

## Indiquer le niveau de rendu

**Disponibilité : Apollina 3.0.0-beta2**

Cette nouvelle version d’Apollina (par rapport à la version intégrée à Venus 1) vous permet d’indiquer un niveau de rendu. Cela permet par exemple de ne pas afficher le layout si vous voulez par exemple juste charger le contenu central en Ajax.

Voici comment le redéfinir dans votre contrôleur :

|  |
| --- |
| $this->view->setRenderLevel(\Apollina\Template::LEVEL\_NO\_RENDER); |

Nous avons aujourd’hui les trois niveaux présentés ci-dessous :

|  |
| --- |
| /\*\*  \* constant to define the render level  \* **@var** int  \*/  **const** LEVEL\_NO\_RENDER = 0;  **const** LEVEL\_VIEW = 1;  **const** LEVEL\_LAYOUT = 2; |

Voici comment l’utiliser par exemple dans Venus 2 ($this->layout représente une instance d’Apollina où l’on a indiqué qu’on était en mode layout) où on ne désire pas afficher le layout mais seulement les vues :

|  |
| --- |
| $this->layout  ->setRenderLevel(\Apollina\Template::LEVEL\_VIEW)  ->display(); |

Vous pouvez également désactiver les couches que vous désirez comme ceci (vous pouvez potentiellement en désactiver plusieurs dans le tableau) :

|  |
| --- |
| $this->layout  ->disableLevel(array(\Apollina\Template::LEVEL\_VIEW => 1))  ->display(); |

## Assign accepte aussi les tableaux

**Disponibilité : Apollina 3.0.0-beta2**

Afin de pouvoir entrer plusieurs variables en même temps, vous pouvez mettre un tableau dans le premier élément de l’assign dans votre contrôleur.

|  |
| --- |
| $this->view->assign($aVariables); |

## Ajouter un deuxième dossier de fonctions

**Disponibilité : Apollina 3.0.0-beta2**

Si vous désirez intégrer Apollina au sein d’un framework, il vous sera bien plus utile d’étendre les fonctions à un deuxième répertoire qui sera séparé du dépôt de ce système de templating. Cela vous permettra de n’avoir jamais besoin de toucher au dépôt Apollina et de pouvoir le mettre à jour facilement sans le moindre conflit possible.

Voici un exemple de l’intégration d’Apollina dans Venus 2 où l’on rajoute un dossier qui stockera de nouvelles fonctions hors du dépôt :

|  |
| --- |
| $oApollina = **new** $sVendorName($mParam, str\_replace('lib', '', \_\_DIR\_\_),  str\_replace('bundles'.DIRECTORY\_SEPARATOR.'lib', CACHE\_DIR, \_\_DIR\_\_), $mParam2);    **return** $oApollina->addFunctionPath(\_\_DIR\_\_.DIRECTORY\_SEPARATOR.'Functions', '\Venus\lib\Functions\\'); |