

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

Attila PHP

[Le nouvel ORM PHP]

Table des matières

1. [Présentation 2](#_Toc409550386)

[Installation](#_Toc409550387)

[Concept du routeur](#_Toc409550388)

1. [Présentation 2](#_Toc409550389)

[Installation](#_Toc409550390)

# Présentation

Attila est une nouvelle alternative d’ORM de qualité de développement open-source dans les environnements LAMP.

## Installation

L’installation de l’ORM est relativement simple. Vous pouvez télécharger la version .zip sur le Github <https://github.com/las93> ou ajouter las93/attila 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’Attila qui pourront beaucoup vous aider dans vos différents projets.

## Les transactions

**Disponibilité : Attila 1.0.0-beta2**

Pour faire une transaction sous Attila, il faut utiliser les fonctions suivantes de l’ORM :

|  |
| --- |
| **// Commencer une transaction $oOrm->begin();  // Valider une transaction qui s’appliquera intégralement en base $oOrm->commit() ;   // Invalider une transaction qui ne s’appliquera pas en base. // Toutes les modifications seront annulées $oOrm->rollback() ;** |

## 4 triggers d’initialisation sur les entités

**Disponibilité : Attila 1.0.0-beta2**

Les entités ont deux triggers qui permettent d’être initialisé :

1/ La fonction initialize permet d’initialiser l’entité la première fois qu’elle est appelée et uniquement cette fois-là pendant tout le déroulement du script :

|  |
| --- |
| **public function** initialize() { ; } |

2/ La fonction onConstruct permet d’initialiser l’entité à chaque fois qu’elle est instanciée :

|  |
| --- |
| **public function** onConstruct() { ; } |

3/ La fonction beforeSave permet d’initialiser l’entité avant chaque save effectué :

|  |
| --- |
| **public function** beforeSave() { ; } |

4/ La fonction afterFetch permet d’initialiser l’entité dès qu’elle a été remplit (par load) :

|  |
| --- |
| **public function** afterFetch() { ; } |