

# ORM IgnitedRecord

&lt;

[ORM Doctrine >>](#)

IgnitedRecord est un ORM qui peut être associé à CodeIgniter.

Lien de téléchargement : [IgnitedRecord download](#)

L'aide est disponible à l'intérieur du zip téléchargé.

## Installation

Dézipper l'archive. Copier les fichiers du dossier **application/libraries/** dans le dossier du même nom de votre application web.

**ignitedrecord** doit être chargé :

- soit automatiquement avec autoload.php :

```
$autoload['libraries'] = array('database','ignitedrecord/ignitedrecord');
```

Le chargement de la bibliothèque **database** est indispensable

- Soit dans le code d'un contrôleur par exemple :

```
$this->load->library('ignitedrecord/ignitedrecord');
```

ou

```
$this->load->orm();
```

## Création des classes métier

Une classe métier correspond à la notion de **model** dans codeigniter.

Considérons la base de données suivante :



La base de données sera composée de 2 models : utilisateur et categorie. La relation de type CIF entre utilisateurs et categories peut s'exprimer de la façon suivante :

- Chaque utilisateur appartient à 1 catégorie (belongs\_to)
- Dans chaque categorie, on peut compter de 0 à n utilisateurs (has\_many)

## Le model utilisateur

Dans le dossier **application/models**, créer le fichier utilisateur.php :

[|h application/models/utilisateur.php](#)

```
<?php
class Utilisateur extends IgnitedRecord {
    public $table='utilisateurs';
    public $belongs_to ='categorie';
}
?>
```

- Un **model** est un fichier contenant une classe dont le nom commence par une majuscule.
- Le nom du fichier doit être le même que celui de la classe, mais en minuscule.
- Le fichier doit être enregistré dans le dossier **application/models/**
- La table de la base de données associée au model est nommée modelNames

## Le model categorie

Dans le dossier **application/models**, créer le fichier categorie.php :

[|h application/models/categorie.php](#)

```
<?php
class Categorie extends IgnitedRecord{
    public $table='categories';
    public $has_many='utilisateurs';
}
?>
```

## Gestion des utilisateurs

### Contrôleur utilisateurs

Ajouter un contrôleur utilisateurs dans controllers :

- la méthode **all** charge tous les utilisateurs, et leur catégorie correspondante.
- Elle appelle ensuite la vue **v\_utilisateurs** et lui passe les utilisateurs chargés (**users**)

[|h application/controllers/utilisateurs.php](#)

```
<?php
class Utilisateurs extends CI_Controller{
    public function all(){
        $users=$this->utilisateur->find_all();
        foreach ($users as $user){
            $user->cat=$user->related('categorie')->get();
        }
    }
}
```

```
    }  
    $this->load->view('v_utilisateurs',array('utilisateurs'=>$users));  
  }  
}  
?>
```

## Vues

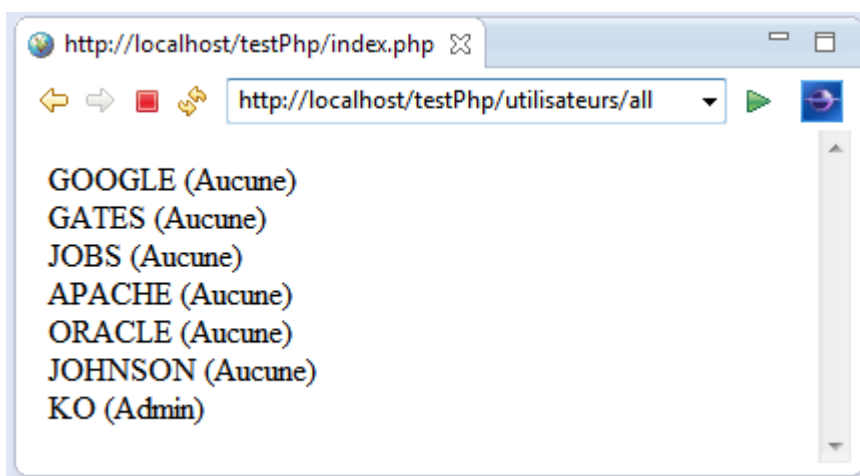
### Liste des utilisateurs

Créer la vue v\_utilisateurs pour afficher la liste des utilisateurs : La variable \$utilisateurs est récupérée par la méthode all du contrôleur utilisateurs

[|h application/controllers/utilisateurs.php](#)

```
<?php  
foreach ($utilisateurs as $user){  
    echo($user->nom." (".$user->cat->nom.")<br>");  
}  
?>
```

Tester en allant à l'adresse <http://localhost/testPhp/utilisateurs/all/>



From:

<http://slamwiki2.kobject.net/> - **Broken SlamWiki 2.0**

Permanent link:

<http://slamwiki2.kobject.net/slam4/php/codeigniter/orm?rev=1355066223>

Last update: **2019/08/31 14:41**

