

Astuces

<

[Bibliothèques >>](#)

1. [Introduction à CodeIgniter](#)
2. [Helpers](#)
3. [Bibliothèques](#)
4. [Sessions CodeIgniter](#)
5. [Validation des formulaires](#)
6. [ORM IgnitedRecord](#)
7. [Doctrine](#)
8. [Javascript et codeIgniter](#)

Création d'un contrôleur de base personnalisé par surdéfinition de CI_Controller

Le contrôleur de base de codeIgniter correspond à la classe **CI_Controller** localisée dans system/core.

Classe dérivée

```
class BaseCtrl extends \CI_Controller{
    public function __construct(){
        // TODO Auto-generated method stub
        parent::__construct();
    }
}
```

Toutes les classes (ne commençant pas par CI_) et présentes dans application/controller ou application/core seront automatiquement chargées.

Modification de config.php

Il est nécessaire de modifier le fichier **config.php** pour que la classe **BaseCtrl** soit automatiquement chargée au lancement de l'application :

Ajouter le code suivant à la fin de config.php, en utilisant la fonction php **_autoload()**

```
//Fin de config.php
function __autoload($class){
    if(strpos($class, 'CI_') !== 0){
        $paths = array(APPPATH . 'core/', APPPATH . 'controllers/');
        foreach($paths as $path){
            if (file_exists($path . $class . EXT)){
                @include_once( $path . $class . EXT );
                break;
            }
        }
    }
}
```

```
    }  
  }  
}
```

Contrôleur personnalisé et complétion de code

L'inconvénient majeur de php est l'absence de typage des variables, combiné dans le cas présent au fonctionnement même de codeigniter, les 2 réunis font qu'aucune complétion n'est présente par défaut sur les principaux objets accessibles depuis un contrôleur :

- les librairies chargées (doctrine, jsutils...)
- le loader load, l'objet ci...

Pour remédier à ce manque et avoir l'auto-complétion Eclipse sur ces objets utilisés fréquemment, il suffit de surcharger CI_Controller, de définir explicitement les membres de données existants, et de les commenter en style javadoc.

Eclipse utilisera les commentaires de type @var pour proposer l'auto-complétion sur les membres concernés.

```
<?php  
  
class BaseCtrl extends \CI_Controller{  
  
    /**  
     * @var CI_JsUtils  
     */  
    private $jsutils;  
    /**  
     * @var CI_Base  
     */  
    private $CI;  
    /**  
     * @var Doctrine  
     */  
    private $doctrine;  
    /**  
     * @var CI_Loader  
     */  
    private $load;  
    public function __construct(){  
        // TODO Auto-generated method stub  
        parent::__construct();  
    }  
}
```

From:
<http://slamwiki2.kobject.net/> - SlamWiki 2.1

Permanent link:
<http://slamwiki2.kobject.net/slam4/php/codeigniter/controller?rev=1417565302>

Last update: 2019/08/31 14:41



