

Javascript et codeIgniter

<

[Introduction à CodeIgniter >>](#)

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

Solutions possibles

Il est possible d'utiliser les frameworks javascript connus :

- JQuery
- Prototype
- Mootools
- Ext Js

ou de mettre en oeuvre un framework Javascript s'intégrant à CodeIgniter : CJAX

CJAX Framework

- [Documentation doc sur sf](#)
- [Téléchargement](#)

Installation

Copier le contenu du zip dans les destinations suivantes :

- **ajax.php** et **ajaxfw.php** et le dossier **cjax** dans /
- le contenu du dossier **application** dans le dossier **application** du site
 - le dossier **response**
 - le contenu de **views** vers **views**
 - le contenu de **controllers** vers **controllers**

Mise en oeuvre

Avec CJAX, Les contrôleurs se situent dans le dossier **response** de application, et sont accessibles par l'url **ajax.php?nomController**.

Message sur click

- Créer un contrôleur dans **response** nommé **TestAjax**

[|h application/response/testAjax.php](#)

```
<?php
class TestAjax extends CI_Controller{
    public function click(){
        $this->load->view('vClick');
    }
}
?>
```

- Créer la vue **vClick.php** dans **applications/views/**
 - `ajax()` permet d'instancier ajax ou de récupérer l'instance active
 - `$ajax->init()` insère le fichier javascript C AJAX dans le header de la page * `$ajax->exec` associe l'appel de **update** au click sur le bouton **btClick** * `$ajax->update` affiche le message "Test de click" dans **divResponse** `<code php|h application/views/vClick.php> <?php require_once "ajax.php"; $ajax = ajax(); $ajax->Exec("btClick",$ajax->update("divResponse","Test de click")); ?> <html> <head> <meta charset="UTF-8"> <title>Message sur click</title> <?php echo $ajax->init();?> </head> <body> <h2>message sur click du bouton</h2> <input type='button' id='btClick' value='Cliquer sur le bouton'> <div id="divResponse">divResponse</div> </body> </html> </code> ===== Requête sur click =====`
- Il s'agit de faire une requête vers une page (/test/affiche/) sur le click d'un bouton, et d'afficher le résultat dans une div * Ajouter une méthode dans le contrôleur **testAjax** : `<code php |h application/response/testAjax.php> <?php class TestAjax extends CI_Controller{ public function click(){ $this->load->view('vClick'); } public function get(){ $this->load->view('vGet'); } } ?> </code>` * Créer la vue **vGet.php** * `$ajax->Exec` exécute l'appel du **call** sur le click du bouton **btGet** * `$ajax->divResponse%%` permet d'affecter le résultat du chargement de la page **affiche** à la div **divResponse**

[application/views/vGet.php](#)

```
<?php

require_once "ajax.php";
$ajax = ajax();

$ajax->Exec('btGet' , $ajax->divResponse=$ajax->call("test/affiche"));

?>
<html>
<head>
<meta charset="UTF-8">
<title>GET</title>
<?php echo $ajax->init();?>
</head>
<body>
<h2>Get</h2>
<input type='button' id='btGet' value='Cliquer sur le bouton'>
<div id="divResponse">divResponse</div>
</body>
</html>
```

POST sur click

- Ajouter 2 méthodes dans le contrôleur testAjax :
 - **post** permet d'afficher la vue permettant de poster une requête
 - **postResult** affiche le résultat du POST

|h application/response/testAjax.php

```
<?php
class TestAjax extends CI_Controller{
    public function click(){
        $this->load->view('vClick');
    }
    public function get(){
        $this->load->view('vGet');
    }
    public function post(){
        $this->load->view('vPost');
    }
    public function postResult(){
        var_dump($_POST);
    }
}
?>
```

From:

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

Permanent link:

<http://slamwiki2.kobject.net/slam4/php/codeigniter/javascript?rev=1355942778>

Last update: 2019/08/31 14:41

