

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec codeigniter.....	2
Remarques.....	2
Versions.....	2
Exemples.....	2
Installation et configuration.....	2
Environnement Windows.....	2
Environnement Mac.....	2
Environnement Linux.....	3
GitHub.....	3
URL de base.....	3
Supprimer index.php de l'URL.....	3
Configuration de la base de données.....	4
Définir le contrôleur par défaut.....	5
AutoLoad Library And Helper.....	5
Exécuter plusieurs applications sur un système CI.....	5
Structure du dossier:.....	5
Configuration du code :.....	6
Augmentez la sécurité en masquant l'emplacement de vos fichiers CodeIgniter.....	6
Chapitre 2: Appel d'une méthode de modèle dans une vue.....	8
Introduction.....	8
Exemples.....	8
Enregistrer un appel de méthode dans une variable.....	8
Chapitre 3: Assistant de tableau.....	9
Introduction.....	9
Exemples.....	9
Chargement de cette aide.....	9
Chapitre 4: Authentification.....	11
Exemples.....	11

Chargement de votre bibliothèque d'authentification pour chaque contrôleur.....	11
Chapitre 5: CAPTCHA Helper.....	12
Exemples.....	12
Chargement de cette aide.....	12
create_captcha (\$ data).....	12
Utilisation de l'aide CAPTCHA.....	12
Exemple complet.....	13
Chapitre 6: Code pagination.....	15
Exemples.....	15
dans cette section, je suppose que vous savez appeler l'assistant, dans le contrôleur.....	15
dans la vue.....	15
Chapitre 7: CodeIgniter - Internationalisation.....	16
Introduction.....	16
Exemples.....	16
Exemple.....	16
Chapitre 8: CodeIgniter Shopping Cart.....	20
Introduction.....	20
Exemples.....	20
Ajout d'articles dans le panier.....	20
et vous pouvez insérer plusieurs éléments comme celui-ci.....	20
Éléments de base du panier d'achat CI.....	21
Afficher les articles du panier.....	21
Mettre à jour l'article du panier.....	22
Supprimer les articles du panier.....	22
Chapitre 9: Commençons par: Hello World.....	23
Exemples.....	23
Une application très simple Hello World.....	23
Utilisons un peu plus le contrôleur.....	24
Choisissons nos salutations: Hello World ou Good Bye World ou ...?.....	25
Chapitre 10: Comment définir un fuseau horaire dans CodeIgniter.....	27
Exemples.....	27
Comment définir le fuseau horaire dans CodeIgniter.....	27

Une autre façon de définir le fuseau horaire dans le codeigniter.....	28
Chapitre 11: Comment utiliser les bibliothèques et l'aide CI.....	30
Exemples.....	30
Utilisation des librairies et des helpers.....	30
Assistant.....	31
Chapitre 12: Comment utiliser les bibliothèques et les aides CI?.....	33
Syntaxe.....	33
Exemples.....	33
Créer et appeler une bibliothèque.....	33
Chapitre 13: Créer cronjob dans codeigniter sur le serveur d'hébergement Linux.....	34
Exemples.....	34
Appel d'un contrôleur CodeIgniter à partir de cron.....	34
Cronjob dans Codeigniter.....	34
Chapitre 14: Créer une API dans Codeigniter.....	36
Introduction.....	36
Exemples.....	36
créer le nouveau contrôleur avec le nom API.....	36
Récupérer des données de l'API Ajouter la fonction suivante dans le contrôleur API.....	36
se connecter à l'API utilisateur pour permettre l'accès à certaines données privées pour u.....	37
l'utilisateur déconnecte api pour détruire la session de l'utilisateur connecté.....	39
créer des api protégés.....	40
Chapitre 15: ensemble de session flashdata.....	42
Exemples.....	42
Comment définir des données de session flash dans le contrôleur.....	42
Comment afficher Flashdata en vue.....	42
Chapitre 16: Envoi d'email.....	43
Remarques.....	43
Exemples.....	43
Charger la bibliothèque de messagerie.....	43
Définissez vos paramètres de configuration du courrier électronique.....	43
Créez votre email.....	44
Envoyer votre email.....	44

Envoyer un email HTML.....	44
Formulaire de contact.....	45
Chapitre 17: Identification de code.....	48
Introduction.....	48
Exemples.....	48
Dépannage.....	48
Chapitre 18: Image / File Uploader dans CodeIgniter.....	49
Remarques.....	49
Exemples.....	49
File Single / Image Uploader.....	49
Chapitre 19: Jouez avec le mot anglais avec l'aide d'INFLECTOR.....	52
Introduction.....	52
Exemples.....	52
Aide à l'inflecteur de charge.....	52
Faites un mot singulier.....	52
Vérifier un mot au pluriel.....	52
Faire un mot au pluriel.....	52
Camélisé la ficelle.....	52
Supprimer / Ajouter un délimiteur entre les mots.....	53
Supprimer le délimiteur.....	53
Ajouter le soulignement.....	53
Chapitre 20: La gestion des erreurs.....	54
Introduction.....	54
Exemples.....	54
show_error ().....	54
Syntaxe.....	54
La source.....	54
show_404 ().....	54
Syntaxe.....	55
La source.....	55
log_message ().....	55

Syntaxe	55
Il existe trois types de messages:	56
Chapitre 21: Sécuriser votre application web	57
Introduction	57
Syntaxe	57
Paramètres	57
Exemples	57
Prévention XSS	57
Prévention d'injection SQL	57
Cacher les erreurs PHP	58
Prévention CSRF	59
Supprimer les données d'abus de la saisie de l'utilisateur	59
Prévention XSS sur entrée utilisateur	59
Chapitre 22: Segment URI CodeIgniter	61
Exemples	61
Segments d'URI:	61
Obtenir le dernier et dernier segment d'URI	61
Chapitre 23: Structure de requête	62
Exemples	62
Sélection de données	62
Sélection de données	62
Sélection de données avec un deuxième paramètre facultatif	63
Joindre des tables à l'aide de Query Builder	63
Chapitre 24: suffixe url	65
Exemples	65
suffixe url	65
Chapitre 25: Supprimer index.php en utilisant WAMP et CodeIgniter	66
Exemples	66
Comment supprimer index.php de l'url avec wamp et codeigniter	66
Chapitre 26: URL de base dans Codeigniter	68
Exemples	68

Définir votre URL de base dans Codeigniter.....	68
Quelque chose de plus à propos de base_url.....	69
Que se passe-t-il si je ne fixe pas base_url ?	69
Si je ne l'ai pas mis ce qui va apparaître?	69
Qu'est-ce que cela signifie http://[::1]/ ??.....	69
Comment définir correctement base_url() ??	69
Comment utiliser base_url() ??	70
Une manière intelligente de configurer la base_url.....	70
Chapitre 27: Utilisation de crochets.....	71
Exemples.....	71
Activation des crochets.....	71
Définir un crochet.....	71
Crochet Points.....	71
pre_system.....	71
pre_controller.....	72
post_controller_constructor.....	72
post_controller.....	72
display_override.....	72
cache_override.....	72
post_system.....	72
Exemple de pré contrôleur Hook utilisant CodeIgniter.....	72
Définir un crochet.....	73
Chapitre 28: Utiliser des sessions.....	74
Remarques.....	74
Exemples.....	74
Créer une session.....	74
Gestion des données de session.....	74
Pour récupérer des données de session.....	74
Pour définir des données de session.....	75
Pour supprimer les données de session et de session.....	75
Chapitre 29: Utiliser un modèle dans un identificateur de code.....	77

Exemples.....	77
Créer un modèle.....	77
Modèle de chargement.....	77
Fonction du modèle d'appel.....	78
Passer des données au modèle.....	78
Recevoir des données du contrôleur.....	78
Renvoyer les données au contrôleur.....	79
Chapitre 30: Validation de formulaire.....	80
Exemples.....	80
Exemple de formulaire de validation.....	80
Crédits.....	81

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [codeigniter](#)

It is an unofficial and free codeigniter ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official codeigniter.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec codeigniter

Remarques

CodeIgniter est un framework MVC écrit dans et pour PHP.

Il est léger comparé aux autres frameworks MVC, au prix de moins de fonctionnalités, par exemple, il n'y a pas de système d'authentification intégré pouvant faire partie d'autres frameworks.

CodeIgniter est un bon choix de frameworks pour ceux qui commencent avec MVC car il ne force aucune norme particulière pour la dénomination et la structure du code; mais il convient également aux projets de grande envergure pour lesquels une large gamme de fonctionnalités contenues dans d'autres frameworks peut ne pas être nécessaire.

Versions

Version	Date de sortie
Version Bêta 1.0	2006-02-28
Version 2.0.0	2011-01-28
Version 2.2.0	2014-06-02
Version 3.0.0	2015-03-30
Version 3.1.3	2017-01-09
La version est 3.1.4	2017-03-20
La version est 3.1.5	2017-06-19

Exemples

Installation et configuration

Environnement Windows

1. Installer [XAMPP](#) ou [WAMP](#)
2. Téléchargez et décompressez le package depuis [Codeigniter.com](#)
3. Extraire tout le document dans l'espace du serveur (répertoire htdocs ou www)

Environnement Mac

1. Installez [MAMP](#)
2. Téléchargez et décompressez le package depuis [Codeigniter.com](#)
3. Extraire tout le document dans l'espace serveur (htdocs)

Environnement Linux

1. Téléchargez et décompressez le package depuis [Codeigniter.com](#)
2. Placez le dossier extrait dans / var / www (dans WAMP) ou xampp / htdocs (XAMPP)

GitHub

```
git clone https://github.com/bcit-ci/CodeIgniter.git
```

Si vous suivez le système correctement, vous obtiendrez l'écran ci-dessous.

URL de base

1. Aller à l' `application/config/config.php`
 2. Définir l'URL de base sous la forme `$config['base_url'] = 'http://localhost/path/to/folder';`
-

Supprimer `index.php` de l'URL

Configuration d'Apache

1. aller à la racine
2. créer un fichier htaccess
3. Ajouter le code ci-dessous à l'intérieur

```
RewriteEngine on
RewriteCond $1 !^(index\.php|assets|image|resources|robots\.txt)
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php/$1 [L,QSA]
```

Remarque: le code .htaccess varie selon le serveur d'hébergement. Dans certains serveurs d'hébergement (par exemple: Godaddy), vous devez utiliser un serveur supplémentaire ? dans la dernière ligne du code ci-dessus. La ligne suivante sera remplacée par la dernière ligne dans le cas applicable:

```
RewriteRule ^(.*)$ index.php?/$1 [L,QSA]
```

Configuration Nginx

1. Ouvrir le fichier de configuration nginx (par défaut: /etc/nginx/sites-available/default)
2. Ajouter le code ci-dessous à l'intérieur

```
server {
    server_name domain.tld;

    root /path-to-codeigniter-folder; //you codeigniter path
    index index.html index.php;

    # set expiration of assets to MAX for caching
    location ~* \.(ico|css|js|gif|jpe?g|png) (\?[0-9]+)?$ {
        expires max;
        log_not_found off;
    }

    location / {
        # Check if a file or directory index file exists, else route it to index.php.
        try_files $uri $uri/ /index.php;
    }

    location ~* \.php$ {
        fastcgi_pass 127.0.0.1:9000;
        include fastcgi.conf;
    }
}
```

Configuration de la base de données

1. Aller à l' application/config/database.php

2. Définissez les variables de configuration suivantes.

- Hôte
- Nom d'utilisateur
- Mot de passe
- Nom de la base de données
- Port

Définir le contrôleur par défaut

1. Aller à l' `application/config/routes.php`
2. définissez la valeur de la variable de configuration suivante avec le nom de votre contrôleur.
 - `default_controller`

AutoLoad Library And Helper

1. Aller à l' `application/config/autoload.php`
2. set Valeur `$autoload['libraries'] = array('database', 'session');` comme
`$autoload['libraries'] = array('database', 'session');`
3. set Helper value comme `$autoload['helper'] = array('url', 'file', 'form', 'html', 'text');`

Exécuter plusieurs applications sur un système CI

Codeigniter peut être configuré pour exécuter plusieurs projets sans dupliquer les fichiers core CI.

C'est possible en divisant le côté de l'application CI. Par exemple , nous allons prendre projet de site Web, qui contient `front-end` et `back-end` système de gestion de contenu des applications (CMS). Dans ce cas, la structure du dossier CI sera comme suit:

Structure du dossier:

```
├── Codeigniter
│   ├── applications
│   │   ├── front-end
│   │   │   ├── views
│   │   │   ├── models
│   │   │   ├── controllers
│   │   │   ├── config
│   │   │   └── ...
│   │   └── back-end
│   │       ├── views
│   │       ├── models
│   │       ├── controllers
│   │       ├── config
│   │       └── ...
│   ├── system
│   └── core
```

```

|   |   |   | database
|   |   |   | helpers
|   |   |   | ...
|   |   |   |
|   |   |   | index.php
|   |   |   | backend.php

```

Dans le dossier des `applications` , nous avons créé deux dossiers: `front-end` et `back-end` et copié tout le contenu par défaut des `applications` sous ces deux dossiers.

Nous avons également dupliqué le fichier `index.php` sous le dossier racine en tant que `backend.php`

Ensuite, configurez `CI` pour qu'il fonctionne avec ces deux instances d'application.

Configuration du code :

Ouvrez les fichiers **`index.php`** et **`backend.php`** et mettez à jour `application_folder` config:

```

//index.php
$application_folder = 'applications/front-end';

//backend.php
$application_folder = 'applications/back-end';

```

Après la configuration ci-dessus, `CI` est prêt à exécuter deux applications sous un système `CI`:

La demande sur `example.com/Codeigniter/index.php` ouvrira `front-end` application `front-end`

Demande sur `example.com/Codeigniter/backend.php` va ouvrir l'application `back-end`

Augmentez la sécurité en masquant l'emplacement de vos fichiers CodeIgniter

Dans le CodeIgniter, il y a deux répertoires principaux à prendre en compte: **système** et **application** . Le dossier système contient les entrailles principales de CodeIgniter. Le dossier de l'application contiendra tout le code spécifique à votre application, y compris les modèles, contrôleurs, vues et autres bibliothèques pertinentes.

Selon les [instructions d'installation de](#) CodeIgniter, dans l'intérêt de la sécurisation de votre application, le dossier du système et celui de l'application doivent être placés au-dessus de la racine Web afin qu'ils ne soient pas directement accessibles via un navigateur. Par défaut, les fichiers `.htaccess` sont inclus dans chaque dossier pour empêcher l'accès direct, mais il est préférable de les supprimer complètement de l'accès public si la configuration du serveur Web change ou ne respecte pas le `.htaccess`.

```

|   |   |   | CodeIgniter
|   |   |   | application

```

```
| |─ system
| |─ wwwroot
| |─ index.php
```

Après avoir déplacé les dossiers système et application, ouvrez le fichier principal `index.php` et définissez les `$system_path` , `$application_folder` , de préférence avec un chemin complet, par exemple `'/www/MyUser/system'` . Cependant, les chemins relatifs doivent fonctionner.

Pour Linux / Apache:

```
$application_folder = './application';
$system_path = './system';
```

Pour Windows / IIS:

```
$application_folder = '../application/';
$system_path = '../system/';
```

Lire Démarrer avec codeigniter en ligne: <https://riptutorial.com/fr/codeigniter/topic/929/demarrer-avec-codeigniter>

Chapitre 2: Appel d'une méthode de modèle dans une vue

Introduction

Parfois, il est plus utile de faire appel à la méthode d'un modèle à notre avis, c'est donc une façon de le faire

Exemples

Enregistrer un appel de méthode dans une variable

Dans le contrôleur:

```
$this->load->model('your_model');  
$data['model'] = $this->your_model;
```

En vue:

```
$model->your_method;
```

Lire Appel d'une méthode de modèle dans une vue en ligne:

<https://riptutorial.com/fr/codeigniter/topic/8163/appele-d-une-methode-de-modele-dans-une-vue>

Chapitre 3: Assistant de tableau

Introduction

Le fichier Array Helper contient des fonctions qui facilitent l'utilisation des tableaux.

Exemples

Chargement de cette aide

Cet assistant est chargé à l'aide du code suivant:

```
$this->load->helper('array');
```

Les fonctions suivantes sont disponibles:

élément()

Vous permet de récupérer un élément d'un tableau. La fonction teste si l'index du tableau est défini et s'il a une valeur. Si une valeur existe, elle est renvoyée. Si une valeur n'existe pas, elle renvoie FALSE ou ce que vous avez spécifié comme valeur par défaut via le troisième paramètre. Exemple:

```
$array = array('color' => 'red', 'shape' => 'round', 'size' => '');  
  
// returns "red"  
echo element('color', $array);  
  
// returns NULL  
echo element('size', $array, NULL);
```

random_element ()

Prend un tableau en entrée et renvoie un élément aléatoire de celui-ci. Exemple d'utilisation:

```
$quotes = array(  
    "I find that the harder I work, the more luck I seem to have. - Thomas Jefferson",  
    "Don't stay in bed, unless you can make money in bed. - George Burns",  
    "We didn't lose the game; we just ran out of time. - Vince Lombardi",  
    "If everything seems under control, you're not going fast enough. - Mario  
Andretti",  
    "Reality is merely an illusion, albeit a very persistent one. - Albert Einstein",  
    "Chance favors the prepared mind - Louis Pasteur"  
);  
  
echo random_element($quotes);
```

éléments()

Vous permet de récupérer un certain nombre d'éléments d'un tableau. La fonction teste si chacun des index du tableau est défini. Si un index n'existe pas, il est défini sur FALSE ou sur tout ce que vous avez spécifié comme valeur par défaut via le troisième paramètre. Exemple:

```
$array = array(
    'color' => 'red',
    'shape' => 'round',
    'radius' => '10',
    'diameter' => '20'
);

$my_shape = elements(array('color', 'shape', 'height'), $array);
```

Ce qui précède retournera le tableau suivant:

```
array(
    'color' => 'red',
    'shape' => 'round',
    'height' => FALSE
);
```

Vous pouvez définir le troisième paramètre sur toute valeur par défaut que vous aimez:

```
$my_shape = elements(array('color', 'shape', 'height'), $array, NULL);
```

Ce qui précède retournera le tableau suivant:

```
array(
    'color' => 'red',
    'shape' => 'round',
    'height' => NULL
);
```

Ceci est utile lors de l'envoi du tableau `$_POST` à l'un de vos modèles. Cela empêche les utilisateurs d'envoyer des données POST supplémentaires à entrer dans vos tables:

```
$this->load->model('post_model');

$this->post_model->update(elements(array('id', 'title', 'content'), $_POST));
```

Cela garantit que seuls les champs id, title et content sont envoyés pour être mis à jour.

Lire Assistant de tableau en ligne: <https://riptutorial.com/fr/codeigniter/topic/8068/assistant-de-tableau>

Chapitre 4: Authentification

Exemples

Chargement de votre bibliothèque d'authentification pour chaque contrôleur

Accédez à codeigniter / application / libraries / créez ou remplacez vos fichiers de bibliothèque ici.

aller à codeigniter / application / core / créer un nouveau fichier php nommé comme MY_Controller.php

à l'intérieur de MY_Controller.php

```
<?php
class MY_Controller extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library('AuthLib'); // AuthLib is your library name
    }
}
```

Et puis, sur chaque fichier de contrôleur, vous devez étendre MY_Controller.

Exemple de contrôleur aller à codeigniter / application / controllers et créer un fichier php

```
<?php
class Profile extends MY_Controller{
    public function __construct(){
        parent::__construct();
        if ($this->AuthLib->logged_in() === FALSE) { //if you wanna make this condition
stament on every controller just write it to inside construct function in MY_Controller.php
        redirect(base_url('/'));
        }
    }
}
```

Lire Authentification en ligne: <https://riptutorial.com/fr/codeigniter/topic/7722/authentification>

Chapitre 5: CAPTCHA Helper

Exemples

Chargement de cette aide

Cet assistant est chargé à l'aide du code suivant:

Dans le contrôleur lui-même (* peut répéter encore et encore *)

```
$this->load->helper('captcha');
```

Dans config/autoload.php (charge une seule fois)

```
$autoload['helper'] = array('captcha');
```

create_captcha (\$ data)

Prend un tableau d'informations pour générer CAPTCHA en entrée et crée l'image selon vos spécifications, renvoyant un tableau de données associatives sur l'image.

```
[array]
(
  'image' => IMAGE TAG
  'time'   => TIMESTAMP (in microtime)
  'word'   => CAPTCHA WORD
)
```

L'image est la balise d'image réelle:

```

```

Le "time" est le micro timestamp utilisé comme nom de l'image sans l'extension de fichier. Ce sera un numéro comme celui-ci: 1139612155.3422

Le "mot" est le mot qui apparaît dans l'image captcha, qui, si elle n'est pas fournie à la fonction, sera une chaîne aléatoire.

Utilisation de l'aide CAPTCHA

Une fois chargé, vous pouvez générer un captcha comme ceci:

```
$vals = array(
  'word'      => 'Random word',
  'img_path'   => './captcha/',
  'img_url'    => 'http://example.com/captcha/',
  'font_path'  => './path/to/fonts/texb.ttf',
```

```

    'img_width'      => '150',
    'img_height'    => 30,
    'expiration'    => 7200
  );

$cap = create_captcha($vals);
echo $cap['image'];

```

- La fonction captcha nécessite la bibliothèque d'images GD.
- Seuls les `img_path` et `img_url` sont requis.
- Si un "mot" n'est pas fourni, la fonction générera une chaîne ASCII aléatoire. Vous pouvez créer votre propre bibliothèque de mots que vous pouvez dessiner au hasard.
- Si vous ne spécifiez pas de chemin vers une police TRUE TYPE, la police GD native sera utilisée. Le dossier "captcha" doit être accessible en écriture (666 ou 777)
- L'expiration (en secondes) indique combien de temps une image restera dans le dossier captcha avant d'être supprimée. La valeur par défaut est deux heures.

Exemple complet

Voici un exemple d'utilisation avec une base de données. Sur la page où le CAPTCHA sera affiché, vous aurez quelque chose comme ceci:

```

$this->load->helper('captcha');
$vals = array(
    'img_path'      => './captcha/',
    'img_url'       => 'http://example.com/captcha/'
);

$cap = create_captcha($vals);

$data = array(
    'captcha_time'  => $cap['time'],
    'ip_address'    => $this->input->ip_address(),
    'word'          => $cap['word']
);

$query = $this->db->insert_string('captcha', $data);
$this->db->query($query);

echo 'Submit the word you see below:';
echo $cap['image'];
echo '<input type="text" name="captcha" value="" />';

```

Ensuite, sur la page qui accepte la soumission, vous aurez quelque chose comme ceci:

```

// First, delete old captchas
$expiration = time()-7200; // Two hour limit
$this->db->query("DELETE FROM captcha WHERE captcha_time < ".$expiration);

// Then see if a captcha exists:
$sql = "SELECT COUNT(*) AS count FROM captcha WHERE word = ? AND ip_address = ? AND captcha_time > ?";
$binds = array($_POST['captcha'], $this->input->ip_address(), $expiration);
$query = $this->db->query($sql, $binds);
$row = $query->row();

```

```
if ($row->count == 0)
{
    echo "You must submit the word that appears in the image";
}
```

Lire CAPTCHA Helper en ligne: <https://riptutorial.com/fr/codeigniter/topic/7902/captcha-helper>

Chapitre 6: Code pagination

Exemples

dans cette section, je suppose que vous savez appeler l'assistant, dans le contrôleur

```
public function house()
{
    $config['base_url']          = site_url().'/user/house/';
    $config['total_rows']       = $this->houses->select_row_house_design();
    $config['per_page']         = 12;
    $config['cur_tag_open']     = '<li><a><b>';
    $config['cur_tag_close']    = '</li></a></b>';
    $config['prev_tag_open']    = '<li>';
    $config['prev_tag_close']   = '</li>';
    $config['next_tag_open']    = '<li>';
    $config['next_tag_close']   = '</li>';
    $config['num_tag_open']     = '<li>';
    $config['num_tag_close']    = '</li>';
    $config['last_tag_open']    = '<li>';
    $config['last_tag_close']   = '</li>';
    $config['first_tag_open']   = '<li>';
    $config['first_tag_close']  = '</li>';
    $this->pagination->initialize($config);
    $from = $this->uri->segment('3');
    $data['design'] = $this->houses->select_all_house_design($config['per_page'],$from);
    $title['menu'] = 'house design';
    $this->load->view('user/template/header',$title);
    $this->load->view('user/house',$data);
    $this->load->view('user/template/footer');
```

dans la vue

`pagination-> create_links (); ?>`

Lire Code pagination en ligne: <https://riptutorial.com/fr/codeigniter/topic/9393/code-pagination>

Chapitre 7: CodeIgniter - Internationalisation

Introduction

La classe de langue dans CodeIgniter fournit un moyen facile de prendre en charge plusieurs langues pour l'internationalisation. Dans une certaine mesure, nous pouvons utiliser différents fichiers de langue pour afficher du texte dans de nombreuses langues différentes.

Exemples

Exemple

Création de fichiers Langue

Pour créer un fichier de langue, vous devez le terminer avec `_lang.php`. Par exemple, vous voulez créer un fichier de langue pour la langue française, vous devez ensuite l'enregistrer avec `french_lang.php`. Dans ce fichier, vous pouvez stocker tous vos textes de langue dans la combinaison clé-valeur dans le tableau `$lang`, comme indiqué ci-dessous.

```
$lang['key'] = 'val';
```

Chargement du fichier de langue

Pour utiliser l'une des langues de votre application, vous devez d'abord charger le fichier de cette langue particulière pour récupérer les différents textes stockés dans ce fichier. Vous pouvez utiliser le code suivant pour charger le fichier de langue.

```
$this->lang->load('filename', 'language');
```

filename : C'est le nom du fichier que vous voulez charger. N'utilisez pas l'extension de fichier ici, mais uniquement le nom du fichier. **Language** : c'est la langue qui la contient.

Récupération du texte de la langue

```
$this->lang->line('language_key');
```

Pour extraire une ligne du fichier de langue, exécutez simplement le code suivant. Où **language_key** est le paramètre clé utilisé pour récupérer la valeur de la clé dans le fichier de langue chargé.

Autoload Languages

Si vous avez besoin d'un langage global, vous pouvez le charger automatiquement dans le fichier `application/config/autoload.php` comme indiqué ci-dessous.


```

| -----
|   Auto-load Language files
| -----
| Prototype:
|   $autoload['language'] = array('lang1', 'lang2');
|
| NOTE: Do not include the "_lang" part of your file. For example
| "codeigniter_lang.php" would be referenced as array('codeigniter');
|
| */
$autoload['language'] = array();

```

Simplement, passez les différentes langues à charger automatiquement par CodeIgniter.

Créez un contrôleur appelé `Lang_controller.php` **et enregistrez-le dans**

`application/controller/Lang_controller.php`

```
<?php
```

la classe `Lang_controller` étend `CI_Controller` {

```

public function index(){
    //Load form helper
    $this->load->helper('form');

    //Get the selected language
    $language = $this->input->post('language');

    //Choose language file according to selected language
    if($language == "french")
        $this->lang->load('french_lang', 'french');
    else if($language == "german")
        $this->lang->load('german_lang', 'german');
    else
        $this->lang->load('english_lang', 'english');

    //Fetch the message from language file.
    $data['msg'] = $this->lang->line('msg');

    $data['language'] = $language;
    //Load the view file
    $this->load->view('lang_view', $data);
}
}
?>

```

Créez un fichier de vue appelé `lang_view.php` **et enregistrez-le à l'** `application/views/lang_view.php`

```

<!DOCTYPE html>
<html lang = "en">
    <head>
        <meta charset = "utf-8">
        <title>CodeIgniter Internationalization Example</title>
    </head>
    <body>
        <?php
            echo form_open('/lang');

```

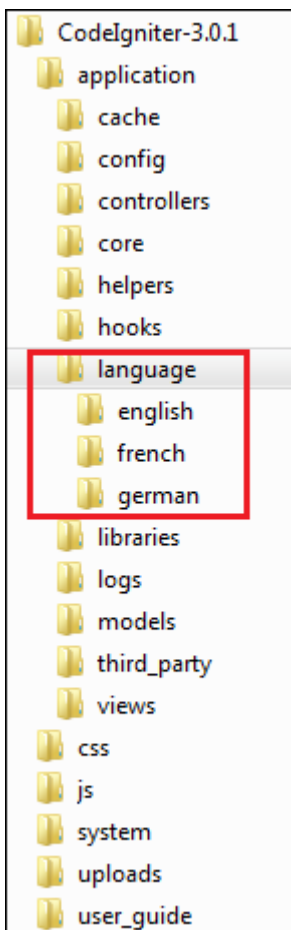
```

?>
<select name = "language" onchange = "javascript:this.form.submit();">
    <?php
        $lang = array('english'=>"English", 'french'=>"French", 'german'=>"German");

        foreach($lang as $key=>$val) {
            if($key == $language)
                echo "<option value = '". $key.'" selected>".$val."</option>";
            else
                echo "<option value = '". $key.'">".$val."</option>";
        }
    ?>
</select>
<?php
    form_close();
    echo $msg;
?>
</body>
</html>

```

Créez trois dossiers appelés **anglais** , **français** et **allemand** dans l' **application / langue**, comme indiqué dans la figure ci-dessous.



Copiez le code ci-dessous et enregistrez-le dans le fichier `english_lang.php` dans le dossier `application/language/english` .

```

<?php
    $lang['msg'] = "CodeIgniter Internationalization example.";

```

```
?>
```

Copiez le code ci-dessous et enregistrez-le dans le fichier `french_lang.php` dans le dossier `application/language/French` .

```
<?php
    $lang['msg'] = "Exemple CodeIgniter internationalisation.";
?>
```

Copiez le code ci-dessous et enregistrez-le dans le fichier `german_lang.php` dans le dossier `application/language/german` .

```
<?php
    $lang['msg'] = "CodeIgniter Internationalisierung Beispiel.";
?>
```

Modifiez le fichier `routes.php` dans `application/config/routes.php` pour ajouter une route pour le contrôleur ci-dessus et ajoutez la ligne suivante à la fin du fichier.

```
$route['lang'] = "Lang_controller";
```

Exécutez l'URL suivante dans le navigateur pour exécuter l'exemple ci-dessus.

```
http://yoursite.com/index.php/lang
```

Ensuite, archivez votre navigateur. Je vous remercie.

Lire CodeIgniter - Internationalisation en ligne:

<https://riptutorial.com/fr/codeigniter/topic/9864/codeigniter---internationalisation>

Chapitre 8: CodeIgniter Shopping Cart

Introduction

Nous pouvons utiliser la bibliothèque de panier d'achat de CI lorsque nous construisons un site de commerce électronique. Nous pouvons configurer ajouter au panier, mettre à jour les éléments du panier, supprimer les articles du panier et même effacer les fonctionnalités du chariot en utilisant cette bibliothèque.

De CodeIgniter Doc: La classe Cart permet d'ajouter des éléments à une session qui reste active pendant qu'un utilisateur navigue sur votre site. Ces éléments peuvent être récupérés et affichés dans un format standard de «panier d'achat», permettant à l'utilisateur de mettre à jour la quantité ou de supprimer des articles du panier.

Exemples

Ajout d'articles dans le panier

Vous devez créer des fonctions dans un contrôleur, telles que l'insertion, la mise à jour, la suppression et la suppression du panier, etc.

```
$cartItem = array(
    'id'      => 'MOTOG5',
    'qty'     => 5,
    'price'   => 100.99,
    'name'    => 'Motorola Moto G5 - 16 GB',
    'options' => array(
        'ram' => '3GB',
        'Color' => 'Fine Gold'
    )
);
```

Et créer des fonctions dans le modèle pour les tâches du chariot telles que l'insertion, la mise à jour, la suppression, la suppression, etc.

ex: pour insérer des articles dans le panier

```
$this->cart->insert($cartItem);
```

La méthode insert () renverra le \$ rowid si vous insérez avec succès un seul élément. Vous pouvez donc vérifier que cet élément a été inséré ou non et afficher le message associé à l'utilisateur.

et vous pouvez insérer plusieurs éléments comme celui-ci

```
$data = array(
    array(
```

```

        'id'      => 'sku_123ABC',
        'qty'     => 1,
        'price'   => 39.95,
        'name'    => 'T-Shirt',
        'options' => array('Size' => 'L', 'Color' => 'Red')
    ),
    array(
        'id'      => 'sku_567ZYX',
        'qty'     => 1,
        'price'   => 9.95,
        'name'    => 'Coffee Mug'
    ),
    array(
        'id'      => 'sku_965QRS',
        'qty'     => 1,
        'price'   => 29.95,
        'name'    => 'Shot Glass'
    )
);

$this->cart->insert($data);

```

Éléments de base du panier d'achat CI

Comme nous pouvons ajouter plusieurs éléments dans le tableau Cart, puis l'ajouter à la session du panier, la classe Cart nécessite 4 éléments de base pour ajouter des données avec succès dans la session du panier.

1. id (chaîne)
2. quantité (nombre)
3. prix (nombre, décimal)
4. nom (chaîne)

Et si vous voulez ajouter plus d'options concernant le produit, vous pouvez utiliser le 5ème élément qui est "options". vous pouvez définir un tableau d'options dans cet élément.

Il ressemblera à ceci :

```

$cartItem = array(
    'id'      => 'MOTOG5',
    'qty'     => 5,
    'price'   => 100.99,
    'name'    => 'Motorola Moto G5 - 16 GB',
    'options' => array(
        'ram' => '3GB',
        'Color' => 'Fine Gold'
    )
);

```

Afficher les articles du panier

Vous pouvez afficher les articles du chariot par boucle dans le panier ou vous pouvez afficher un seul article du panier.

```
$cartContents = $this->cart->contents();
```

Cela renverra un tableau d'éléments de panier afin que vous puissiez parcourir ce tableau en utilisant la boucle foreach.

```
foreach ($cartContents as $items){  
    echo "ID : ". $items["id"] . "<br>";  
    echo "Name : ". $items["name"] . "<br>";  
    echo "Quantity : ". $items["qty"] . "<br>";  
    echo "Price : ". $items["price"] . "<br>";  
}
```

Vous pouvez formater ces données sous forme de cellule de tableau ou de div et ensuite afficher dans la vue.

Mettre à jour l'article du panier

Rowid: l'ID de ligne est un identifiant unique généré par le code du panier lorsqu'un article est ajouté au panier. La raison pour laquelle un identifiant unique est créé est que des produits identiques avec des options différentes peuvent être gérés par le panier.

Chaque élément du panier a un élément rowid et, par rowid, vous pouvez mettre à jour l'article du panier.

```
$updateItem = array(  
    'rowid' => 'b99ccdf16028f015540f341130b6d8ec',  
    'qty'   => 3  
);
```

et ensuite sous le code

```
$this->cart->update($data);
```

Supprimer les articles du panier

En utilisant l'élément rowid, vous pouvez supprimer un élément du panier. il suffit de mettre la quantité de l'article à 0

```
$deleteItem = array(  
    'rowid' => 'b99ccdf16028f015540f341130b6d8ec',  
    'qty'   => 0  
);  
  
$this->cart->update($data);
```

cela va supprimer l'élément avec cette rowid.

Lire CodeIgniter Shopping Cart en ligne:

<https://riptutorial.com/fr/codeigniter/topic/9372/codeigniter-shopping-cart>

Chapitre 9: Commençons par: Hello World

Exemples

Une application très simple Hello World

À partir d'une nouvelle installation de Codeigniter 3, voici un moyen simple de commencer avec une application Hello World, pour briser la glace avec ce solide framework PHP.

Pour ce faire, vous pouvez commencer à créer la vue que nous souhaitons afficher pour notre application Hello World.

Nous allons le mettre dans votre dossier de candidature, ici:

Dans `hello_world.php` (`/application/views/`)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Hello World</title>
</head>
<body>

    <h1>Hello World!</h1>

</body>
</html>
```

C'est juste un simple contenu HTML.

Maintenant, pour faire apparaître cette vue, nous avons besoin d'un **contrôleur** . Le contrôleur est celui qui va rappeler la vue pour que son contenu soit affiché.

Pour que cela fonctionne correctement, le contrôleur doit aller dans le dossier des contrôleurs appropriés.

Voici où nous allons placer notre contrôleur Hello World:

`/application/controllers/Hello_world.php`

(Le nom du contrôleur est généralement *snake_case* avec la première lettre en majuscule)

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Hello_world extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
    }
}
```

```

    }

    public function index(){
        $this->load->view('hello_world');
    }
}

```

La fonction par défaut d'un contrôleur est la fonction d'index.

Maintenant, vous pourrez voir le contenu de votre page Hello World en accédant à l'adresse suivante:

```
http://[your_domain_name]/index.php/hello_world
```

ou, si vous avez appliqué le correctif à l'aide de .htaccess (retournez à la page d'installation du correctif)

```
http://[your_domain_name]/hello_world
```

(Si vous travaillez localement, l'adresse à laquelle vous trouverez votre page est probablement: `http://localhost/hello_world`)

L'URL est en fait formée en appelant votre classe de contrôleur (dans ce cas, `Hello_world`, mais en utilisant toutes les minuscules dans l'URL). Dans ce cas, cela suffit, car nous avons utilisé la fonction `index`. Si nous avions utilisé un nom de fonction différent (disons `greetings`), nous aurions dû utiliser une URL comme celle-ci:

```
http://[your_domain_name]/hello_world/greetings
```

Qui est structuré comme `/[controller_name]/[method_name]`.

Voici! Votre première application Codeigniter fonctionne!

Utilisons un peu plus le contrôleur

Nous allons maintenant essayer un exemple un peu plus complexe, en utilisant les capacités du contrôleur pour remplir la vue.

Voici notre vue: `/application/views/hello_world.php`

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Hello World</title>
</head>
<body>

<h1><?php echo $greetings;?></h1>

```



```
</body>
</html>
```

Nous avons maintenant un espace réservé pour afficher nos salutations.

Voici comment nous changeons le contrôleur pour que cela fonctionne:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Hello_world extends CI_Controller {

    public function __construct() {
        parent::__construct();
    }

    public function greetings(){
        $data = array('greetings'=>'Hello World');
        $this->load->view('hello_world',$data);
    }
}
```

Le tableau `$data` est préparé avec les informations à injecter dans la vue, en utilisant le même libellé (`greetings`) qui a été appelé dans la vue.

Le résultat final est le même qu'avec le premier exemple, mais nous utilisons maintenant plus de potentialité du framework!

Choisissons nos salutations: Hello World ou Good Bye World ou ...?

Disons que nous voulons avoir un message d'accueil alternatif accessible via une URL différente. Nous pourrions créer une nouvelle fonction ou même un nouveau contrôleur pour cela, mais une meilleure pratique consiste à optimiser ce que nous avons déjà, à le faire fonctionner au mieux!

Pour ce faire, nous conserverons la même vue que dans les exemples précédents, mais nous introduirons un paramètre à notre fonction, afin de pouvoir choisir entre deux messages d'accueil différents:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Hello_world extends CI_Controller {

    public function __construct() {
        parent::__construct();
    }

    public function greetings($my_greetings){
        switch($my_greetings)
        {
            case 'goodbye':
                $say = 'Good Bye World';
                break;
            case 'morning':
```

```

        $say = 'Good Morning World';
    break;
    default:
        $say = 'Hello World';
    }
    $data = array('greetings'=>$say);
    $this->load->view('hello_world',$data);
}
}

```

Maintenant, nous avons plusieurs options de salutations! Pour les visualiser, nous allons ajouter le paramètre à l'URL, comme suit:

```
http://[your_domain_name]/hello_world/greetings/goodbye
```

Cela nous montrera le message: "Good Bye World".

La structure de l'URL est la suivante:

```
http://[your_domain_name]/[controller_name]/[function_name]/[parameter_1]
```

Dans ce cas, pour revenir à notre bon vieux "Hello World", il suffit d'appeler l'ancienne URL, sans paramètres:

```
http://localhost/hello_world/greetings
```

Vous pouvez ajouter plusieurs paramètres à votre fonction (par exemple, si vous en avez besoin de trois):

```
public function greetings($param1,$param2,$param3)
```

et ils peuvent être remplis en utilisant l'url comme suit:

```
http://[your_domain_name]/[controller_name]/[function_name]/[param1]/[param2]/[param3]
```

par exemple `http://localhost/hello_world/greetings/goodbye/italian/red`

De cette façon, vous pouvez avoir des paramètres directement transmis depuis l'URL qui affectera le contenu de ce qui sera affiché.

Pour en savoir plus sur la manière de transmettre des paramètres via l'URL, vous pouvez vous intéresser au routage!

Lire **Commençons par: Hello World en ligne:**

<https://riptutorial.com/fr/codeigniter/topic/2411/commencons-par--hello-world>

Chapitre 10: Comment définir un fuseau horaire dans CodeIgniter

Exemples

Comment définir le fuseau horaire dans CodeIgniter

Placer `date_default_timezone_set('Asia/Kolkata');` sur `config.php` ci-dessus l'URL de base fonctionne également.

[Liste PHP des fuseaux horaires pris en charge](#)

application / config.php

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

date_default_timezone_set('Asia/Kolkata');
```

Une autre façon que j'ai trouvée utile est de définir un fuseau horaire pour chaque utilisateur:

- Créez un fichier `MY_Controller.php`.
- Créez une colonne dans votre table `user`, vous pouvez la nommer fuseau horaire ou toute autre chose que vous souhaitez. Ainsi, lorsque l'utilisateur sélectionne son fuseau horaire, il peut être défini sur son fuseau horaire lors de la connexion.

application / core / MY_Controller.php

```
<?php

class MY_Controller extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->set_timezone();
    }

    public function set_timezone() {
        if ($this->session->userdata('user_id')) {
            $this->db->select('timezone');
            $this->db->from($this->db->dbprefix . 'user');
            $this->db->where('user_id', $this->session->userdata('user_id'));
            $query = $this->db->get();
            if ($query->num_rows() > 0) {
                date_default_timezone_set($query->row()->timezone);
            } else {
                return false;
            }
        }
    }
}
```

```
}  
}
```

Aussi, pour obtenir la liste des fuseaux horaires en PHP:

```
$timezones = DateTimeZone::listIdentifiers(DateTimeZone::ALL);  
  
foreach ($timezones as $timezone) {  
    echo $timezone;  
    echo "<br />";  
}
```

Une autre façon de définir le fuseau horaire dans le codeigniter

La définition du fuseau horaire dans Codeigniter en étendant la date de l'assistant est une autre solution. Pour ce faire, vous devez suivre l'activité en deux étapes suivante.

1. Prolonger l'assistant de date avec la fonction suivante:

```
if ( ! function_exists('now'))  
{  
    /**  
     * Get "now" time  
     *  
     * Returns time() based on the timezone parameter or on the  
     * "time_reference" setting  
     *  
     * @param    string  
     * @return    int  
     */  
    function now($timezone = NULL)  
    {  
        if (empty($timezone))  
        {  
            $timezone = config_item('time_reference');  
        }  
        if ($timezone === 'local' OR $timezone === date_default_timezone_get())  
        {  
            return time();  
        }  
        $datetime = new DateTime('now', new DateTimeZone($timezone));  
        sscanf($datetime->format('j-n-Y G:i:s'), '%d-%d-%d %d:%d:%d', $day, $month, $year,  
$hour, $minute, $second);  
        return mktime($hour, $minute, $second, $month, $day, $year);  
    }  
}
```

2. `time_reference` maintenant le fuseau horaire comme une valeur de `time_reference` de `config.php` comme: `$config['time_reference'] = 'Asia/Dhaka';`

Tout est réglé pour utiliser le fuseau horaire.

FYI: la liste de la liste des fuseaux horaires est ajoutée dans le premier exemple.

[Lire Comment définir un fuseau horaire dans CodeIgniter en ligne:](#)

<https://riptutorial.com/fr/codeigniter/topic/3767/comment-definir-un-fuseau-horaire-dans-codeigniter>

Chapitre 11: Comment utiliser les bibliothèques et l'aide CI

Exemples

Utilisation des librairies et des helpers

L'exemple est utilisé à titre d'illustration pour utiliser des bibliothèques et des helpers et non un code valide. Ne pas copier / coller sur vos projets.

Aides **ASSISTANT** / sendEmail_helper.php

```
if ( ! function_exists('sendEmail'))
{
    function sendEmail($email, $subject, $message, $lang, $cc = null, $file = null) {

        $CI =& get_instance();

        $mail_config['protocol'] = 'smtp';
        $mail_config['smtp_host'] = 'host';
        $mail_config['smtp_user'] = 'user';
        $mail_config['smtp_pass'] = 'pass';
        $mail_config['smtp_port'] = '587';
        $mail_config['smtp_timeout'] = 5;
        $mail_config['charset'] = 'utf-8';
        $mail_config['mailtype'] = 'html';
        $mail_config['wrapchars'] = 76;
        $mail_config['wordwrap'] = TRUE;

        $CI->email->initialize($mail_config);
        $CI->email->set_newLine('\r\n');

        if ($lang == "en"){
            $CI->email->from('support.en@domain.com', 'English Support');
        }else{
            $CI->email->from('support.fr@domain.com', 'Support en francais');
        }
        $CI->email->to($email);
        if ($cc != null){
            $CI->email->cc($cc);
        }
        $CI->email->subject($subject);
        $CI->email->message($message);
        if ($file != null){
            $CI->email->attach($file);
        }
        // $CI->email->print_debugger();
        return $CI->email->send();
    }
}
```

Les bibliothèques **Bibliothèque** / Alerter.php

```

class Alerter {

    public function alert_user($user_email, $subject, $message, $lang) {
        //load helper
        $this->load->helper('sendEmail');
        //using helper
        sendEmail($user_email, $subject, $message, $lang);
    }

    public function alert_admin($admin_email, $subject, $message, $lang, $reason){
        //load helper
        $this->load->helper('sendEmail');
        .....
        //using helper
        sendEmail($admin_email, $subject, $message, $lang);
        .....
    }
}

```

MANETTE

```

class Alerts extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    public function send_alert($userid) {

        //load library and model
        $this->load->library('Alerter');
        $this->load->model('alerter_model');

        //get user
        $user = $this->alerter_model->get_one_by_id($userid);

        //using library
        $this->Alerter->alert_user($user->email, $subject, $message, $lang);

    }
}

```

Assistant

Chargez automatiquement votre fonction d'assistance. si vous utilisez beaucoup de temps dans votre projet

```
$autoload['helper'] = array('url', 'form');
```

Utilisez l'aide de formulaire en vue

```

<?php echo form_open('Public/Login/loginAuth'); ?>

<?php
    echo "<div class='row'>";
    echo "<label for='inputEmail' class='col-lg-2 control-label col-lg-offset-2 col-md-2 control-label col-md-offset-2 col-sm-2 control-label col-sm-offset-2'>Enter Email</label>";

```

```

        $email = array(
            "name"=>"email",
            "placeholder"=>"Email",
            "class"=>"form-control"
        );

        echo "<div class='col-lg-6 col-md-6 col-sm-6'>";
        echo form_error('email');
        echo form_input($email). "<br/>";
        echo "</div>";
    echo "</div>";

    echo "<div class='row'>";
        echo "<label for='inputPassword' class='col-lg-2 control-label col-lg-offset-2 col-md-2 control-label col-md-offset-2 col-sm-2 control-label col-sm-offset-2'>Enter Password</label>";
        $password = array(
            "name"=>"password",
            "placeholder"=>"Password",
            "class"=>"form-control"
        );

        echo "<div class='col-lg-6 col-md-6 col-sm-6'>";
        echo form_error('password');
        echo form_password($password). "<br/>";
        echo "</div>";
    echo "</div>";

    echo "<div class='row'>";

        $submit = array(
            "name"=>"submit",
            "value"=>"Submit",
            "class"=>"btn btn-primary col-lg-offset-9 col-md-offset-9 col-sm-offset-9 col-xs-offset-9"
        );
        echo form_submit($submit). "<br/>";

    echo "</div>";

```

?>

Lire Comment utiliser les bibliothèques et l'aide CI en ligne:

<https://riptutorial.com/fr/codeigniter/topic/3776/comment-utiliser-les-bibliotheques-et-l-aide-ci>

Chapitre 12: Comment utiliser les bibliothèques et les aides CI?

Syntaxe

1. `$ this-> load-> library ('library_name');`
2. `$ this-> nom_bibliothèque-> nom_fonction ();`
3. `$ this-> load-> library ('cart'); # pour l'assistant $ this-> load-> helper ('helperName');`
4. `$ this-> cart-> insert ($ Array);`

Exemples

Créer et appeler une bibliothèque

Pour utiliser des bibliothèques dans CodeIgniter, vous devez créer une bibliothèque.

```
class Pro {  
    function show_hello_world()  
    {  
        return 'Hello World';  
    }  
}
```

Dans cette bibliothèque, appelée pro.php, ce fichier doit être ajouté au chemin suivant.

Chemin: \ xampp \ htdocs \ project \ application \ libraries

Maintenant, vous pouvez l'utiliser dans votre contrôleur. Code pour charger cette bibliothèque dans le contrôleur:

```
$this->load->library('pro');
```

Code pour utiliser les fonctions de la bibliothèque:

```
class Admin extends CI_Controller {  
    function index()  
    {  
        $this->load->library('pro');  
        echo $this->pro->show_hello_world();  
    }  
}
```

Lire Comment utiliser les bibliothèques et les aides CI? en ligne:

<https://riptutorial.com/fr/codeigniter/topic/3770/comment-utiliser-les-bibliotheques-et-les-aides-ci->

Chapitre 13: Créer cronjob dans codeigniter sur le serveur d'hébergement Linux

Exemples

Appel d'un contrôleur CodeIgniter à partir de cron

```
// application/controllers/Company_controller.php
<?php
if(!defined('BASEPATH'))
    exit('No direct script access allowed');

class Company_controller extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->model('companies_model');
    }

    // cron entry would be something like this:
    // 1 1 * * * /usr/bin/php [full path to]/index.php company_controller cronCLI AcmeCorp
    >/dev/null 2>&1
    public function cronCLI($firmName) {
        if(PHP_SAPI_NAME == 'cli') {
            $this->companies_model->doSomeDB_Process($firmName);
        } else {
            echo 'CLI only';
        }
    }
}
```

Cronjob dans CodeIgniter

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Cron extends CI_Controller
{
    /**
     * This is default constructor of the class
     */
    public function __construct()
    {
        parent::__construct();
        $this->load->library('input');
        $this->load->model('cron_model');
    }

    /**
     * This function is used to update the age of users automatically
     * This function is called by cron job once in a day at midnight 00:00
     */
    public function updateAge()
    {
        // is_cli_request() is provided by default input library of codeigniter
        if($this->input->is_cli_request())
```

```
        {
            $this->cron_model->updateAge();
        }
        else
        {
            echo "You dont have access";
        }
    }
}
?>
```

Appelez ceci depuis votre gestionnaire cpanel / cron comme suit (j'ai ajouté plusieurs façons de l'appeler):

```
0 0 0 0 0 php-cli /home/your_site_user/public_html/index.php cron updateAge
```

OU

```
0 0 0 0 0 wget http://your_site_url/cron/updateAge
```

OU

```
0 0 0 0 0 /usr/bin/php /home/your_site_user/public_html/index.php cron updateAge
```

Dans mon cas: wget thing fonctionne sur plesk et cpanel (wget créer des fichiers sur le serveur dans votre répertoire racine). php-cli fonctionne sur plesk et cpanel les deux.

Lire Créer cronjob dans codeigniter sur le serveur d'hébergement Linux en ligne:

<https://riptutorial.com/fr/codeigniter/topic/4980/creer-cronjob-dans-codeigniter-sur-le-serveur-d-hebergement-linux>

Chapitre 14: Créer une API dans Codeigniter

Introduction

Codeigniter fournit une classe de sortie auto-initialisée très utile pour créer des API et différents types de documents tels que .pdf, .csv, .image, etc.

REMARQUE: - Le type de document par défaut de Codeigniter est que HTML le transforme en application / json, l'API doit être de type json requis

Exemples

créer le nouveau contrôleur avec le nom API

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Api extends CI_Controller {
    //default value
    private $login_credential;

    function __construct() {
        parent::__construct();
        //for user authentication
        $this->load->library('session');

        //set page header type Json as default
        $this->output->set_content_type('application/json');
        //default credentials for user login
        $this->login_credential = array(
            'username'=>'admin',
            'password'=>'test@123',
            'email'=>'domain@test.com'
        );
    }
}
?>
```

Récupérer des données de l'API Ajouter la fonction suivante dans le contrôleur API

```
/******
@return all events
*****/
public function getallevents(){
    //get data from model
    $events = array(
        array('Event 1', '2015-04-03'),
        array('Event 2', '2015-04-03'),
        array('Event 3', '2015-06-16'),
```

```
array('Event 4', '2015-06-29'),  
array('Event 5', '2015-07-04'),  
array('Event 6', '2015-12-25'),  
array('Event 7', '2016-01-01')  
);  
$this->output->set_output(json_encode(array('status'=>true, 'events'=>$events)));  
}
```

Vue du facteur

se connecter à l'API utilisateur pour permettre l'accès à certaines données privées pour un utilisateur particulier

```
/*****  
login user
```

```

@required : username and password via post method only
@return user data if login successfull otherwise error message
*****/
public function login(){
    $username=$this->input->post('username');
    $password=$this->input->post('password');
    if($username && $password){
        //check username and password
        if($this->login_credential['username']==$username && $this->login_credential['password']==$password){
            //set user data to store in session
            $userdata = array(
                'username' => $this->login_credential['username'],
                'email' => $this->login_credential['email'],
                'logged_in' => true
            );
            //set session
            $this->session->set_userdata($userdata);
            //display log in successfull msg
            $this->output->set_output(json_encode(array('status'=>true,'msg'=>'log in successfully','data'=>$userdata)));
        }else{
            //wrong username or password
            $this->output->set_output(json_encode(array('status'=>false,'msg'=>'invalid Username or password')));
        }
    }else{
        //when username and password not set
        $this->output->set_output(json_encode(array('status'=>false,'msg'=>'provide Username and password')));
    }
}

```

l'utilisateur déconnecte api pour détruire la session de l'utilisateur connecté

```
/******  
log out user  
*****/  
public function logout(){  
    //delete all session  
    session_destroy();  
    $this->output->set_output(json_encode(array('status'=>true,'msg'=>'log Out successfully')));  
}
```

créer des api protégés

Cette API n'est pas accessible à l'utilisateur public, l'authentification est requise

```
/******  
this is protected api this is not accessible if you are not logged in  
*****/  
public function protectedapi(){  
    if($this->session->userdata('logged_in')){  
        //this section only accessible when user logged in  
        $this->output->set_output(json_encode(array('status'=>true,'msg'=>'Access allowed')));  
    }else{  
        $this->output->set_output(json_encode(array('status'=>true,'msg'=>'Access denied')));  
    }  
}
```


Lire Créer une API dans Codeigniter en ligne:

<https://riptutorial.com/fr/codeigniter/topic/10903/creer-une-api-dans-codeigniter>

Chapitre 15: ensemble de sessionflashdata

Exemples

Comment définir des données de session flash dans le contrôleur

Vous pouvez définir des données flash dans le contrôleur simplement en utilisant cette syntaxe

```
$this->session->set_flashdata('message', 'Message you want to set');
```

Ici, "message" est l'identifiant des données d'accès en vue. Vous pouvez définir plusieurs messages en changeant simplement l'identificateur.

pour ex

```
$this->session->set_flashdata('my_alert', 'Message you want to set');  
$this->session->set_flashdata('my_warnig', 'Message you want to set');
```

Comment afficher Flashdata en vue

Vous pouvez simplement accéder aux fashdata en vue comme ceci

```
<?php echo $this->session->flashdata('message'); ?>
```

Pour accéder à plusieurs messages, il suffit de changer l'identifiant

Pour ex.

```
<?php echo $this->session->flashdata('my_alert'); ?>  
<?php echo $this->session->flashdata('my_warnig'); ?>
```

Lire ensemble de sessionflashdata en ligne:

<https://riptutorial.com/fr/codeigniter/topic/9688/ensemble-de-session-flashdata>

Chapitre 16: Envoi d'email

Remarques

Dans CodeIgniter 3, vous devez inclure le paramètre:

```
$config['newline'] = "\r\n";
```

Cela ne marchera tout simplement pas.

Si vous ne vous souciez pas des nouvelles lignes et que vous utilisez CodeIgniter 2, ce paramètre de configuration est facultatif.

Exemples

Charger la bibliothèque de messagerie

Vous devez d'abord charger la bibliothèque de messagerie.

Faites-le soit dans le fichier de contrôleur qui enverra l'e-mail:

```
$this->load->library('email');
```

Ou chargez-le globalement dans le fichier autoload.php du dossier config:

```
$autoload['libraries'] = array('email');
```

Pendant que vous y êtes, vous pouvez charger l'assistant de messagerie si vous souhaitez utiliser certains raccourcis intégrés à CodeIgniter:

```
$autoload['helper'] = array('email');
```

L'assistant de messagerie peut être chargé dans le fichier Controller de la même manière que la bibliothèque de messagerie:

```
$this->load->helper('email');
```

Définissez vos paramètres de configuration du courrier électronique

Créez un nouveau fichier dans le dossier application / config nommé email.php

Définissez les paramètres d'envoi du courrier électronique. Ceux-ci se chargeront lorsque vous envoyez votre email.

```
$config['newline'] = "\r\n"; //You must use double quotes on this one
```

```
$config['protocol'] = 'smtp';
$config['smtp_host'] = 'ssl://smtp.gmail.com'; //Change for your specific needs
$config['smtp_port'] = 465; //Change for your specific needs
$config['smtp_user'] = 'test@test.com'; //Change for your specific needs
$config['smtp_pass'] = 'yourpassword'; //Change for your specific needs
$config['charset'] = 'iso-8859-1';
$config['mailtype'] = 'text'; //This can be set as 'html' too
```

Créez votre email

```
$this->email->from('accounts@yourwebsite.com', 'Tom Webmaster');
$this->email->to('fred@fake.com', 'Freddie Fakeperson');
$this->email->subject('Your Account Is Active');
$this->email->message('Welcome to our new site!');
```

Dans la méthode 'from', le premier paramètre est l'adresse électronique à laquelle vous envoyez, le deuxième paramètre est le nom que vous souhaitez que le destinataire voie.

Dans la méthode 'to', vous définissez à qui l'e-mail est envoyé.

La méthode 'subject' définit le sujet de l'e-mail.

La méthode 'message' définit ce qui sera dans le corps de votre email.

Chacun de ces éléments peut être une donnée envoyée à votre site par un utilisateur. Vous pouvez donc avoir une variable contenant des données postées. Donc, ils peuvent ressembler plus à ceci:

```
$this->email->to($email, $username);
```

Envoyer votre email

```
$sent = $this->email->send();

//This is optional - but good when you're in a testing environment.
if(isset($sent)){
    echo "It sent!";
}else{
    echo "It did not send.";
}
```

Envoyer un email HTML

Mais vous ne voulez pas simplement un message texte en clair. Vous voulez un joli email HTML.

Définissez votre fichier de configuration en html:

```
$config['mailtype'] = 'html';
```

Si vous souhaitez transmettre des données (comme un nom d'utilisateur par exemple) au courrier

électronique HTML, placez-les dans un tableau:

```
$data = array('name' => $name,  
              'email' => $email,  
              'phone' => $phone,  
              'date' => $date);
```

Ensuite, lors de l'envoi, pointez votre message vers une vue. Puis transmettez-lui votre tableau de données:

```
$this->email->message($this->load->view('new_user',$data, true));
```

Dans votre application / vue, créez votre vue.

Dans ce cas, il s'appelle 'new_user.php'.

Vous pouvez le coiffer comme vous le souhaitez. Voici un exemple rapide:

```
<html>  
<head>  
  <style type='text/css'>  
    body {background-color: #CCD9F9;  
          font-family: Verdana, Geneva, sans-serif}  
  
    h3 {color:#4C628D}  
  
    p {font-weight:bold}  
  </style>  
</head>  
<body>  
  
  <h3>Hi <?php echo $name;?>,</h3>  
  <h3>Thanks for contacting us.</h3>  
  
  <p>You've taken your first step into a larger world.</p>  
  <p>We really appreciate your interest.</p>  
  
</body>  
</html>
```

Formulaire de contact

Contrôleur (Pages.php)

```
public function contact()  
{  
  
  $this->load->library('email');  
  $this->load->library('form_validation');  
  
  //Set form validation  
  $this->form_validation->set_rules('name', 'Name',  
  'trim|required|min_length[4]|max_length[16]');  
  $this->form_validation->set_rules('email', 'Email',  
  'trim|required|valid_email|min_length[6]|max_length[60]');
```

```

        $this->form_validation->set_rules('message', 'Message',
'trim|required|min_length[12]|max_length[200]');

//Run form validation
if ($this->form_validation->run() === FALSE)
{
    $this->load->view('contact');
} else {

    //Get the form data
    $name = $this->input->post('name');
    $from_email = $this->input->post('email');
    $subject = $this->input->post('subject');
    $message = $this->input->post('message');

    //Web master email
    $to_email = 'admin@domain.com'; //Webmaster email, who receive mails

    //Mail settings
    $config['protocol'] = 'smtp';
    $config['smtp_host'] = 'ssl://smtp.gmail.com';
    $config['smtp_port'] = '465';
    $config['smtp_user'] = 'mail@domain.com'; // Your email address
    $config['smtp_pass'] = 'mailpassword'; // Your email account password
    $config['mailtype'] = 'html'; // or 'text'
    $config['charset'] = 'iso-8859-1';
    $config['wordwrap'] = TRUE; //No quotes required
    $config['newline'] = "\r\n"; //Double quotes required

    $this->email->initialize($config);

    //Send mail with data
    $this->email->from($from_email, $name);
    $this->email->to($to_email);
    $this->email->subject($subject);
    $this->email->message($message);

    if ($this->email->send())
    {
        $this->session->set_flashdata('msg','<div class="alert alert-success">Mail
sent!</div>');

        redirect('contact');
    } else {
        $this->session->set_flashdata('msg','<div class="alert alert-danger">Problem in
sending</div>');
        $this->load->view('contact');
    }
}

```

Vues (contact.php)

```

<div class="container">
<h2>Contact</h2>
<div class="row">
    <div class="col-lg-6">
        <?php echo $this->session->flashdata('msg'); ?>
        <form action="<?php echo base_url('contact'); ?>" method="post">
        <div class="form-group">

```

```

        <input name="name" placeholder="Your name" type="text" value="<?php echo
set_value('name'); ?>" class="form-control" />
        <?php echo form_error('name', '<span class="text-danger">', '</span>'); ?>
    </div>
    <div class="form-group">
        <input name="email" placeholder="Your e-mail" type="text" value="<?php echo
set_value('email'); ?>" class="form-control" />
        <?php echo form_error('email', '<span class="text-danger">', '</span>'); ?>
    </div>
    <div class="form-group">
        <input name="subject" placeholder="Subject" type="text" value="<?php echo
set_value('subject'); ?>" class="form-control" />
    </div>
    <div class="form-group">
        <textarea name="message" rows="4" class="form-control" placeholder="Your
message"><?php echo set_value('message'); ?></textarea>
        <?php echo form_error('message', '<span class="text-danger">', '</span>'); ?>
    </div>
    <button name="submit" type="submit" class="btn btn-primary" />Send</button>
</form>
</div>
</div>

```

Lire Envoi d'email en ligne: <https://riptutorial.com/fr/codeigniter/topic/5403/envoi-d-email>

Chapitre 17: Identification de code

Introduction

Pour le débogage et le dépannage dans Codeigniter, vous pouvez utiliser **Profiler** , qui fait partie de la bibliothèque de sortie.

Exemples

Dépannage

Si vous trouvez que peu importe ce que vous mettez dans votre URL seulement votre page par défaut chargement, il se pourrait que votre serveur ne supporte pas la `PATH_INFO` variables nécessaires pour servir moteur de recherche convivial `URLs` .

Dans un premier temps, ouvrez votre fichier `application/config/config.php` et recherchez les informations du `URI Protocol` . Il vous recommandera d'essayer plusieurs autres paramètres.

Si cela ne fonctionne toujours pas après avoir essayé, vous devrez forcer Codeigniter à ajouter un point d'interrogation à vos `URLs` . Pour ce faire, ouvrez votre fichier `application/config/config.php` et modifiez ceci:

```
$config['index_page'] = "index.php";
```

Pour ça:

```
$config['index_page'] = "index.php?";
```

Lire Identification de code en ligne: <https://riptutorial.com/fr/codeigniter/topic/7901/identification-de-code>

Chapitre 18: Image / File Uploader dans CodeIgniter

Remarques

Il n'est pas nécessaire que vous deviez utiliser les mêmes noms pour le (contrôleur, fichier, classe, identifiant) ou autre chose. Tout ce que j'ai utilisé sert à comprendre le flux de codage et mes hypothèses. C'est au développeur qui prend le code et édite le code / nom en fonction de son souhait, puis héberge le code et réussit.

Exemples

File Single / Image Uploader

Nous allons maintenant voir comment le code de téléchargement d'images / fichiers fonctionne dans la méthode CI native à l'aide des formulaires proposés par la méthode CI.

Le téléchargement de fichiers en PHP a deux scénarios. Il est mentionné ci-dessous comme suit.

- Single Image / File uploader - Ceci peut être sauvegardé à l'aide de la variable normale dans l'attribut du formulaire. (Par exemple) `<input type="file" name="image" />`
- Multi-image / File Uploader - Il peut être enregistré uniquement à l'aide de la variable tableau pour le nom dans le type de fichier. (Par exemple) `<input type="file" name="image[]" />`.

La variable de tableau à savoir `name="profile[]"` peut également être conservée pour le téléchargeur d' `single image` ainsi que pour le **téléchargeur d'** `multi-image`.

Par conséquent, le code de téléchargement de l'image unique / du fichier au format Native CodeIgnitor est le suivant:

Voir la partie:

```
<?php
echo form_open_multipart('employee/addemployee', array('name' => 'addemployee',
'class'=>'form-horizontal'));
?>
<div class="form-group">
    <label class="control-label col-sm-4" for="pwd">Profile:</label>
    <div class="col-sm-8">
        <input type="file" class="" id="profile" name="userimage">
    </div>
</div>
<div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
        <input type="submit" class="btn btn-primary pull-right" name="save" value="Save
Employee" />
    </div>
</div>
```

```
<?php
echo form_close();
?>
```

Par conséquent, si nous soumettons le formulaire, il sera envoyé au

- Employee - Contrôleur et recherche de la fonction nommée `addemployee`
- Si vous avez besoin de l'attribut requis pour le code de téléchargement de fichier, vous pouvez ajouter l'attribut HTML5 appelé `required` à la balise d'entrée.

Vous trouverez ci-dessous deux exemples d'utilisation de l'attribut requis, mais les deux méthodes sont également identiques.

1. Première méthode: `<input type="file" name="photo" required="required" />`
2. Deuxième méthode: `<input type="file" name="photo" required />`

Voilà donc quelques conseils importants à suivre dans la partie vue du chargeur d'images / fichiers.

Partie contrôleur:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Employee extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->model('employee_model');
        $this->load->helper('url'); //This will load up all the URL parameters from the helper
    }

    public function addemployee()
    {
        if($_FILES["userimage"]['name']=='')
        {
            // Here you can directly redirect to the form page itself with the Error Message
        }
        else
        {
            $new_name = time().$_FILES["userimage"]['name']; //This line will be generating
            random name for images that are uploaded
            $config['upload_path'] = FCPATH ."assets/fileupload/";
            $config['allowed_types'] = 'gif|jpg|png';
            $config['file_name'] = $new_name;
            $this->load->library('upload', $config); //Loads the Uploader Library
            $this->upload->initialize($config);
            if ( ! $this->upload->do_upload('userimage')) {}
            else
            {
                $data = $this->upload->data(); //This will upload the `image/file` using native
                image upload
            }
            $data_value = array(
                'profile'=>$new_name,
```

```

        ); //Passing data to model as the array() parameter
        $this->employee_model->saveemployee($data_value); //save_employee is the function
name in the Model
    }
}
?>

```

Remarque: Par défaut, la routine de téléchargement attend le fichier provenant d'un champ de formulaire appelé `userfile` , et le `form` doit être de type `multipart` .

- Par conséquent, il ira au `$data_value employee_model` avec le tableau `$data_value` - et il enregistrera les données sous la fonction appelée `saveemployee` .
- Si vous souhaitez définir votre propre nom de champ, transmettez simplement sa valeur à la méthode `do_upload()`
- En utilisant la classe `File Uploading`, nous pouvons télécharger des fichiers et nous pouvons également limiter le type et la taille du fichier à télécharger.
- `display_errors()` - Récupère les messages d'erreur si la méthode `do_upload()` retourne `false`. La méthode ne fait pas automatiquement écho, elle renvoie les données afin que vous puissiez les attribuer comme vous le souhaitez

Notations:

Ce sont les notations disponibles dans le CI et nous pouvons le définir dans le `index.php` tant que définition courte et nous pouvons l'utiliser dans le projet entier.

```

EXT: The PHP file extension
FCPATH: Path to the front controller (this file) (root of CI)
SELF: The name of THIS file (index.php)
BASEPATH: Path to the system folder
APPPATH: The path to the "application" folder

```

Pièce modèle:

```

public function saveemployee($data_value)
{
    $this->db->insert('employee',$data_value);
}

```

- Il enregistrera les données sur la table des `employee` avec le nom de l'image téléchargée.
- Et l'image téléchargée sera enregistrée dans le répertoire que nous avons créé dans le dossier racine ou dans tout autre dossier que nous spécifions.

Lire Image / File Uploader dans CodeIgniter en ligne:

<https://riptutorial.com/fr/codeigniter/topic/7450/image---file-uploader-dans-codeigniter>

Chapitre 19: Jouez avec le mot anglais avec l'aide d'INFLECTOR

Introduction

Inflector est une aide très utile pour changer / convertir un mot anglais en singulier, pluriel, casse de chameau, humaniser, etc.

Exemples

Aide à l'inflecteur de charge

Pour utiliser la méthode de helper d'inflector, commencez par charger le helper comme tout autre helper avec le code suivant:

```
$this->load->helper('inflector');
```

Faites un mot singulier

Fonction `singular($string)` , convertit un mot pluriel en singulier. Pour obtenir un résultat parfait, le paramètre `$string` doit être un mot unique. La fonction retournera une `string` .

```
echo singular("books"); //prints 'book'
```

Vérifier un mot au pluriel

`is_countable($string)` est utilisé pour vérifier qu'un mot a une forme plurielle ou non. Le type de retour sera `boolean` signifie que si le mot donné a une forme plurielle, il retournera `true` , sinon retournera `false` .

```
is_countable('book'); // Returns TRUE
```

Faire un mot au pluriel

Pour obtenir la forme plurielle de n'importe quel mot anglais, la fonction `plural($string)` est pratique. Comme `singular($string)` , la fonction `plural($string)` renvoie également le résultat de la `string` .

```
echo plural("book"); //prints 'books'
```

Camélisé la ficelle

Camel Case est la pratique d'écrire des mots composés ou des phrases où chaque mot commence par une majuscule, sans espace entre les mots. La fonction `camelize($string)` permet de créer une chaîne camélisée. Il convertit une chaîne de mots séparés par des espaces ou des traits de soulignement en cas de chameau.

```
echo camelize('Mc donald'); //Prints mcDonald
```

Supprimer / Ajouter un délimiteur entre les mots

Supprimer le délimiteur

La fonction `humanize($words)`, prend plusieurs mots séparés par des traits de soulignement et ajoute des espaces pour les soulignés avec une majuscule pour chaque mot.

```
echo humanize('mac_donald'); // Prints 'Mac Donald'
```

La fonction peut également remplacer tout séparateur / séparateur déclaré. Dans ce cas, le délimiteur sera le deuxième paramètre.

```
echo humanize('mac-donald','-'); // Prints 'Mac Donald'
echo humanize('mac#donald','#'); // Prints 'Mac Donald'
```

Ajouter le soulignement

D'un autre côté, la fonction de `underscore($words)` remplace l'espace entre les mots par un trait de soulignement (`_`).

```
echo underscore('Mac Donald'); // Prints 'mac_donald'
```

Lire Jouez avec le mot anglais avec l'aide d'INFLECTOR en ligne:

<https://riptutorial.com/fr/codeigniter/topic/8057/jouez-avec-le-mot-anglais-avec-l-aide-d-inflector>

Chapitre 20: La gestion des erreurs

Introduction

CodeIgniter vous permet de générer des rapports d'erreur dans vos applications en utilisant les fonctions décrites ci-dessous. En outre, il dispose d'une classe de journalisation des erreurs qui permet d'enregistrer les messages d'erreur et de débogage en tant que fichiers texte.

Exemples

`show_error ()`

Cette fonction affichera le message d'erreur fourni à l'aide du modèle d'erreur suivant:

Chemin d'accès - `application/errors/error_general.php`

Le paramètre facultatif `$ status_code` détermine le code d'état HTTP à envoyer avec l'erreur.

Syntaxe

```
show_error($message, $status_code, $heading = 'An Error Was Encountered')
```

Paramètres:

- `$message` (mixed) - Message d'erreur
- `$status_code` (int) - Code d'état de la réponse HTTP
- `$heading` (string) - En-tête de page d'erreur

Type de retour: vide

La source

1. [show_error](#) dans [codeigniter.com](#)

`show_404 ()`

Cette fonction affichera le message d'erreur 404 fourni à l'aide du modèle d'erreur suivant:

Chemin d'accès - `application/errors/error_404.php`

La fonction attend que la chaîne qui lui est transmise soit le chemin du fichier vers la page introuvable. Notez que CodeIgniter affiche automatiquement 404 messages si les contrôleurs ne

sont pas trouvés.

CodeIgniter enregistre automatiquement tous les `show_404()`. Si vous définissez le deuxième paramètre facultatif sur `FALSE`, la journalisation sera ignorée.

Syntaxe

```
show_404($page = '', $log_error = TRUE)
```

Paramètres:

- `$page` (chaîne) - Chaîne d'URI
- `$log_error` (bool) - Faut-il enregistrer l'erreur?

Type de retour: vide

La source

1. [show_404](#) dans [codeigniter.com](#)

log_message ()

Cette fonction vous permet d'écrire des messages dans vos fichiers journaux. Vous devez fournir l'un des trois "niveaux" dans le premier paramètre, en indiquant le type de message (debug, error, info), avec le message lui-même dans le deuxième paramètre.

Exemple:

```
if ($some_var == "") {  
    log_message('error', 'Some variable did not contain a value.');
```

```
}  
else {  
    log_message('debug', 'Some variable was correctly set');
```

```
}  
  
log_message('info', 'The purpose of some variable is to provide some value.');
```

Syntaxe

```
log_message($level, $message);
```

Paramètres:

- `$level (string)` - Niveau de journalisation: 'error', 'debug' ou 'info'
- `$message (string)` - Message à enregistrer

Type de retour: vide

Il existe trois types de messages:

- **Messages d'erreur** Ce sont des erreurs réelles, telles que des erreurs PHP ou des erreurs utilisateur.
- **Messages de débogage** Ce sont des messages qui aident au débogage. Par exemple, si une classe a été initialisée, vous pouvez vous connecter en tant qu'information de débogage.
- **Messages d'information** Ce sont les messages les moins prioritaires, donnant simplement des informations sur certains processus. CodeIgniter ne génère pas de messages d'information en mode natif, mais vous pouvez le faire dans votre application.

Remarque: Pour que le fichier journal soit réellement écrit, les "journaux" du dossier doivent être accessibles en écriture. De plus, vous devez définir le "seuil" pour vous connecter à l' `application/config/config.php` . Vous pouvez, par exemple, uniquement consigner les messages d'erreur et non les deux autres types. Si vous le définissez sur zéro, la journalisation sera désactivée.

Lire La gestion des erreurs en ligne: <https://riptutorial.com/fr/codeigniter/topic/8699/la-gestion-des-erreurs>

Chapitre 21: Sécuriser votre application web

Introduction

Rappelez-vous que CodeIgniter est un framework de développement. Il ne cherche pas à sécuriser votre application. Cela vous donne simplement les outils pour le faire vous-même. Si vous consultez la page Sécurité de CI, il est évident que le développeur doit comprendre la sécurité des applications et l'intégrer à son application.

Si la sécurité WebApp est relativement nouvelle pour vous, je commencerais par OWASP. Il serait peut-être avantageux d'examiner d'autres frameworks tels que Zend ou Cake qui, je crois, font plus d'avant-garde.

Syntaxe

- `$freshdata = $this->security->xss_clean($user_input_data);`

Paramètres

tableau de saisie utilisateur	blanc
insérer un tableau de saisie utilisateur dans <code>xss_filter(\$array of user input)</code>	Blanc

Exemples

Prévention XSS

XSS signifie un script intersite. CodeIgniter est fourni avec la sécurité de filtrage XSS. Ce filtre empêchera tout code JavaScript malveillant ou tout autre code qui tente de pirater des cookies et de mener des activités malveillantes. Pour filtrer les données via le filtre XSS, utilisez la méthode `xss_clean()` comme indiqué ci-dessous.

```
$data = $this->security->xss_clean($data);
```

Vous ne devez utiliser cette fonction que lorsque vous soumettez des données. Le deuxième paramètre booléen facultatif peut également être utilisé pour vérifier le fichier image pour une attaque XSS. Ceci est utile pour la fonctionnalité de téléchargement de fichiers. Si sa valeur est vraie, cela signifie que l'image est sûre et non différente.

Prévention d'injection SQL

L'injection SQL est une attaque effectuée sur la requête de base de données. En PHP, nous utilisons la fonction `mysql_real_escape_string()` pour empêcher cela avec d'autres techniques,

mais CodeIgniter fournit des fonctions et des bibliothèques intégrées pour empêcher cela.

Nous pouvons empêcher l'injection SQL dans CodeIgniter des trois manières suivantes -

- Échapper à des requêtes
- Query Biding
- Classe d'enregistrement actif

Échapper à des requêtes

```
<?php
$username = $this->input->post('username');
$query = 'SELECT * FROM subscribers_tbl WHERE user_name = ' .
$this->db->escape($email);
$this->db->query($query);
?>
```

`$this->db->escape()` ajoute automatiquement des guillemets simples autour des données et détermine le type de données afin qu'il ne puisse échapper que des données de chaîne.

Query Biding

```
<?php
$sql = "SELECT * FROM some_table WHERE id = ? AND status = ? AND author = ?";
$this->db->query($sql, array(3, 'live', 'Rick'));
?>
```

Dans l'exemple ci-dessus, le point d'interrogation (?) Sera remplacé par le tableau dans le deuxième paramètre de la fonction `query()`. Le principal avantage de la construction de requêtes de cette manière est que les valeurs sont automatiquement échappées, ce qui produit des requêtes sécurisées. Le moteur CodeIgniter le fait automatiquement pour vous, vous n'avez donc pas à vous en souvenir.

Classe d'enregistrement actif

```
<?php
$this->db->get_where('subscribers_tbl',array('status'=> active,'email' =>
'info@arjun.net.in'));
?>
```

À l'aide des enregistrements actifs, la syntaxe de requête est générée par chaque adaptateur de base de données. Il permet également des requêtes plus sûres, car les valeurs s'échappent automatiquement.

Cacher les erreurs PHP

Dans l'environnement de production, nous ne souhaitons souvent pas afficher de message d'erreur aux utilisateurs. C'est bien s'il est activé dans l'environnement de développement à des fins de débogage. Ces messages d'erreur peuvent contenir des informations que nous ne devons pas montrer aux utilisateurs du site pour des raisons de sécurité.

Il existe trois fichiers CodeIgniter liés à des erreurs. Niveau de rapport d'erreur PHP

Un environnement différent nécessite différents niveaux de rapport d'erreurs. Par défaut, le développement affichera des erreurs, mais les tests et le live les cacheront. Il y a un fichier appelé `index.php` dans le répertoire racine de CodeIgniter, qui est utilisé à cette fin. Si nous transmettons zéro comme argument à la fonction `error_reporting()`, cela masquera toutes les erreurs.

Prévention CSRF

CSRF est synonyme de falsification de requêtes intersites. Vous pouvez empêcher cette attaque en activant une option dans le fichier `application / config / config.php`, comme indiqué ci-dessous.

```
$config['csrf_protection'] = TRUE;
```

Lorsque vous créez un formulaire à l'aide de la fonction `form_open()`, il insère automatiquement un jeton CSRF dans un champ masqué. Vous pouvez également ajouter manuellement le jeton CSRF à l'aide de la fonction `get_csrf_token_name()` et `get_csrf_hash()`. Comme leur nom l'indique, la fonction `get_csrf_token_name()` retournera le nom du jeton CSRF, tandis que `get_csrf_hash()` renverra le hachage.

Le jeton CSRF peut être régénéré à chaque fois pour soumission ou vous pouvez également le conserver pendant toute la durée de vie du cookie CSRF. La définition de l'option de configuration `'csrf_regenerate'` force la régénération du jeton comme indiqué ci-dessous.

```
$config['csrf_regenerate'] = TRUE;
```

Vous pouvez inclure les URL de la protection CSRF dans la liste blanche en définissant des correspondances dans le tableau de configuration à l'aide de la clé `'csrf_exclude_uris'`, comme indiqué ci-dessous. Vous pouvez également utiliser des expressions régulières.

```
$config['csrf_exclude_uris'] = array('api/person/add');
```

Supprimer les données d'abus de la saisie de l'utilisateur

```
// XSS Filtering
$data = array(
    'name' => '<script>Abuse Data</script>'
);
$data = $this->security->xss_clean($data); // Clean Data

// Escaping Queries
<?php $username = $this->input->post('username'); $query = 'SELECT * FROM subscribers_tbl
WHERE user_name = ' . $this->db->escape($email); $this->db->query($query); ?>
```

Prévention XSS sur entrée utilisateur

Ne comptez sur aucune entrée utilisateur. entrée utilisateur tout comme la `<script>` ou toute `alert()`; `javascript alert()`; nous devons donc éviter que toutes les données ne soient pas

exécutées dans notre navigateur. Nous devons donc utiliser la méthode de prévention xss pour restreindre nos données sécurisées à celles conservées par les pirates, ainsi que la responsabilité du développeur quant à la validation des entrées de l'utilisateur et à la résolution des erreurs par programmation.

vérifiez donc que ceci est un exemple de prévention xss dans CodeIgniter.

```
$data = array(
    'name' => "<script>alert('abc')</script>",
    'email' => "useremail@gmail.com"
);
var_dump($data);
// Print array without xss cleaning/xss filtering

array(2) { ["name"]=> string(29) "" ["email"]=> string(19) "useremail@gmail.com" } // Result
with alert

// now print data after xss filtering

$data = $this->security->xss_clean($data);
var_dump($data);

//Print array without xss cleaning/xss filtering
array(2) { ["name"]=> string(38) "[removed]alert('abc')[removed]" ["email"]=> string(19)
"useremail@gmail.com" } // Result Without alert
```

Ainsi, après avoir ajouté xss_filtering, nous n'avons aucun problème pour exécuter un code d'abus saisi par l'utilisateur. et CodeIgniter remplace cette balise d'abus par le mot clé [removed] .

Lire Sécuriser votre application web en ligne:

<https://riptutorial.com/fr/codeigniter/topic/9857/securiser-votre-application-web>

Chapitre 22: Segment URI CodeIgniter

Exemples

Segments d'URI:

Par exemple, veuillez considérer l'URI suivant:

```
http://stackoverflow.com/questions/some-number/how-can-i-do-this/others
```

Segment permet de récupérer une chaîne URI de forme de segment spécifique où n est un numéro de segment. Les segments sont numérotés de gauche à droite. Par exemple, le code suivant:

```
$this->uri->segment (n)
```

Est utilisé pour récupérer un segment spécifique de l'URI où n est le numéro de segment.

```
echo $this->uri->segment (0); //it will print stackoverflow.com
echo $this->uri->segment (1); //it will print questions
echo $this->uri->segment (2); //it will print some-number
echo $this->uri->segment (3); //it will print how-can-i-do-this
echo $this->uri->segment (4); //it will print others
```

Obtenir le dernier et dernier segment d'URI

Obtenez le dernier segment

```
echo end($this->uri->segment_array()); //it will print others
```

Obtenez avant le dernier segment

```
echo $this->uri->segment (count ($this->uri->segment_array())-1); //it will print how-can-i-do-
this
```

Plus d'infos: [<http://stackoverflow.com/questions/9221164/code-igniter-get-before-last-uri-segment>][1]

Lire Segment URI CodeIgniter en ligne: <https://riptutorial.com/fr/codeigniter/topic/5402/segment-uri-codeigniter>

Chapitre 23: Structure de requête

Exemples

Sélection de données

Les fonctions suivantes vous permettent de générer des instructions SQL SELECT.

```
$this->db->get()
```

Cela exécute la requête de sélection et renvoie le résultat. Peut être utilisé seul pour récupérer tous les enregistrements d'une table:

```
$query = $this->db->get('tablename'); // Produces: SELECT * FROM tablename
```

Les deuxième et troisième paramètres vous permettent de définir une clause de limite et de décalage:

```
$query = $this->db->get('tablename', 10, 20);

// Executes: SELECT * FROM tablename LIMIT 20, 10
// (in MySQL. Other databases have slightly different syntax)
```

Sélection de données

Sélection de données avec condition

```
$query = $this->db->select('*')
    ->from('table_name')
    ->where('column_name', $value) // Condition
    ->get();
return $query->result();
```

Sélection de données avec plusieurs conditions

```
$conditions = array('column_name_1' => $value_1, 'column_name_2' => $value_2);
$query = $this->db->select('*')
    ->from('table_name')
    ->where($conditions) // Conditions
    ->get();
return $query->result();
```

Sélectionner des données avec condition et limite

```
$query = $this->db->select('*')
    ->from('table_name')
    ->where('column_name', $value) // Condition
    ->limit(10) // Maximum 10 rows
```

```
        ->get();
return $query->result();
```

Sélectionner des données avec condition, lignes maximum et ordre décroissant

```
$query = $this->db->select('*')
        ->from('table_name')
        ->where('column_name', $value) // Condition
        ->limit(10) // Maximum 10 rows
        ->order_by('id', 'DESC') // Order data descending
        ->get();
return $query->result();
```

Sélection de données avec un deuxième paramètre facultatif

Habituellement, nous n'utilisons pas le second paramètre dans `select([$select = '*', $escape = NULL])` dans CodeIgniter. Si vous le définissez sur `FALSE`, CodeIgniter n'essaiera pas de protéger vos noms de champ ou de table.

Dans l'exemple suivant, nous allons sélectionner le champ de type datetime en le formatant à l'aide de `sql query` et le définir sur `FALSE` (ce faisant, nous allons demander à CI de ne pas échapper automatiquement à la requête).

```
public function getUserInfo($id)
{
    $this->db->select('BaseTbl.id, BaseTbl.name, DATE_FORMAT(BaseTbl.createdDtm, "%d-%m-%Y")
AS createdDtm', FALSE); // FALSE is the second optional parameter
    $this->db->from('tbl_users as BaseTbl');
    $this->db->where('isDeleted', 0);
    $this->db->where('BaseTbl.id', $id);
    $query = $this->db->get();

    return $query->result();
}
```

Si nous ne le définissons pas sur `FALSE`, cela échappera automatiquement à la requête.

Joindre des tables à l'aide de Query Builder

Parfois, nous devons joindre plusieurs tables pour obtenir des données agrégées en retour. Voici comment nous pouvons obtenir la même chose en utilisant CodeIgniter Query Builder / Active Records.

```
public function getStudentInfo($studentid){
    $query = $this->db->select("st.id, st.name, st.class, mk.maths, mk.science")
        ->from("students as st")
        ->join("marks as mk", "mk.student_id = st.id", "inner")
        ->where("st.id", $studentid)
        ->get();
    return $query->result();
}
```

Ici, nous utilisons `join ()` pour joindre plusieurs tables et nous pouvons changer le type de jointure en 3ème paramètre comme "inner", "left", "right" etc.

Lire Structure de requête en ligne: <https://riptutorial.com/fr/codeigniter/topic/3769/structure-de-requete>

Chapitre 24: suffixe url

Exemples

suffixe url

```
$ config ['url_suffix'] = 'html';
```

changer tout ce que vous voulez comme html ou asp, cela fonctionnera après votre index.php sur config.php

Lire suffixe url en ligne: <https://riptutorial.com/fr/codeigniter/topic/9379/suffixe-url>

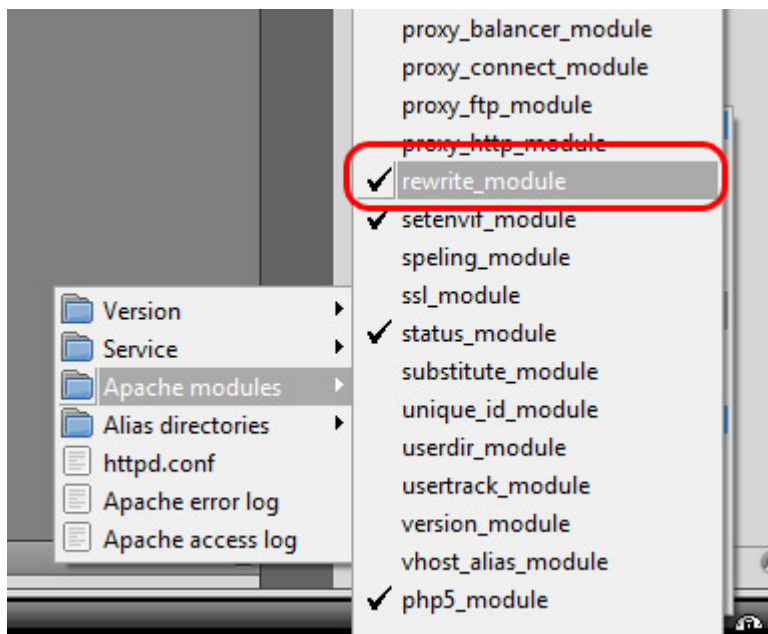
Chapitre 25: Supprimer index.php en utilisant WAMP et CodeIgniter

Exemples

Comment supprimer index.php de l'url avec wamp et codeigniter

La première chose à faire est d'activer la réécriture du mod sur wamp dans les modules Apache et de faire défiler la liste.

Si ce n'est pas le cas, activez-le, puis redémarrez tous les serveurs.



Les utilisateurs de Linux peuvent également utiliser la commande ci-dessous pour activer le module de réécriture

```
sudo a2enmod rewrite
```

Puis redémarrez apache en utilisant:

```
sudo service apache2 restart
```

Ensuite, à l'extérieur de votre dossier d'application, créez un fichier appelé .htaccess

```
project > application
project > system
project > .htaccess
project > index.php
```

Essayez ce code ci-dessous

```
Options +FollowSymLinks
RewriteEngine on
RewriteCond $1 !^(index\.php|images|robots\.txt)
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ ./index.php/$1 [L]
```

Si ce n'est pas le cas, voici quelques [exemples](#) plus complets

Ensuite, allez dans le fichier config.php. Définissez votre base_url et rendez la page index_page vide

```
$config['base_url'] = ((isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] == "on") ? "https" :
"http");
$config['base_url'] .= "://".$_SERVER['HTTP_HOST'];
$config['base_url'] .=
str_replace(dirname($_SERVER['SCRIPT_NAME']), "", $_SERVER['SCRIPT_NAME']);
$config['index_page'] = '';
```

J'espère que cela vous aidera à utiliser les fichiers htaccess à partir d'exemples pour les autres.

[Lire Supprimer index.php en utilisant WAMP et CodeIgniter en ligne:](#)

<https://riptutorial.com/fr/codeigniter/topic/4340/supprimer-index-php-en-utilisant-wamp-et-codeigniter>

Chapitre 26: URL de base dans Codeigniter

Exemples

Définir votre URL de base dans Codeigniter

Vous devrez définir votre URL de base dans `application/config/config.php`

S'il n'est pas défini, alors Codeigniter essaiera de deviner le protocole et le chemin d'accès à votre installation, mais en raison des problèmes de sécurité, le nom d'hôte sera défini sur `$_SERVER['SERVER_ADDR']` s'il est disponible ou `localhost`. Le mécanisme de détection automatique existe uniquement pour des raisons de commodité pendant le développement et NE DOIT PAS être utilisé en production!

```
$config['base_url'] = '';
```

Il devrait être classé comme

```
$config['base_url'] = 'http://localhost/projectname/';  
  
$config['base_url'] = 'http://www.example.com/';
```

Toujours bon à utiliser / à la fin de `base_url`

Lorsque vous ne définissez pas votre URL de base, vous risquez de rencontrer des erreurs lorsque vous ne pouvez pas charger vos éléments CSS, images et autres éléments. De plus, vous pourriez avoir des difficultés à soumettre des formulaires car certains utilisateurs ont été confrontés.

Mettre à jour

Si vous ne souhaitez pas définir votre URL de base, une autre manière est.

Créez un nouveau fichier core dans `application/core/MY_Config.php`

Et collez ce code

```
<?php  
  
class MY_Config extends CI_Config {  
  
    public function __construct() {  
  
        $this->config =& get_config();  
  
        log_message('debug', "Config Class Initialized");  
  
        // Set the base_url automatically if none was provided
```

```

        if ($this->config['base_url'] == '')
        {
            if (isset($_SERVER['HTTP_HOST']))
            {
                $base_url = isset($_SERVER['HTTPS']) && strtolower($_SERVER['HTTPS']) !==
                'off' ? 'https' : 'http';
                $base_url .= '://'. $_SERVER['HTTP_HOST'];
                $base_url .= str_replace(basename($_SERVER['SCRIPT_NAME']), '',
                $_SERVER['SCRIPT_NAME']);
            }

            else
            {
                $base_url = 'http://localhost/';
            }

            $this->set_item('base_url', $base_url);
        }
    }
}

```

Quelque chose de plus à propos de `base_url`

Que se passe-t-il si je ne fixe pas `base_url` ?

Vous n'obtiendrez aucune erreur d'impuissance pour définir cela et continuer. Vous pouvez continuer sans définir, mais vous devez connaître l' [injection d'en-tête HTTP](#)

Si je ne l'ai pas mis ce qui va apparaître?

Vous obtiendrez `http://[::1]/` place de votre URL réelle.

Qu'est-ce que cela signifie `http://[::1]/` ??

Ceci est l'URL temporaire définie par CI par défaut. Cela va pointer la racine de votre document.

`:::1` - Adresse du serveur (localhost) [En savoir plus à ce sujet](#)

Comment définir correctement `base_url()` ??

L'URL de base doit toujours pointer vers la racine du dossier de votre projet. (en dehors du dossier d'application)

```
$config['base_url'] = 'http://localhost/path/to/project'; # If localhost
$config['base_url'] = 'http://stackoverflow.com/'; # If live
$config['base_url'] = 'http://stackoverflow.com/documentation'; # If live & inside subdomain
(assume documentation is subfolder/subdomain)
```

Comment utiliser `base_url()` ??

L'utilisation la plus courante consiste à trouver le bon chemin vers vos fichiers js ou css.

```
<link rel="stylesheet" href="<?php echo base_url('styles/style.css');?>" />
<script src="<?php echo base_url('vendor/jquery/jquery.min.js');?>"></script>
```

L'ajout du code ci-dessus dans votre vue produira HTML comme ci-dessous:

```
<link rel="stylesheet" href="http://localhost/path/to/project/styles/style.css" />
<script src="http://localhost/path/to/project/vendor/jquery/jquery.min.js"></script>
```

Liens

1. [URL Helper](#)

Une manière intelligente de configurer la `base_url`

Les lignes de code suivantes constituent une manière plus intelligente de configurer la `base_url` dans codeigniter:

```
$config['base_url'] = ((isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] == "on") ? "https" :
"http");
$config['base_url'] .= "://".$_SERVER['HTTP_HOST'];
$config['base_url'] .=
str_replace(basename($_SERVER['SCRIPT_NAME']), "", $_SERVER['SCRIPT_NAME']);
```

Recommandé est

```
$config['base_url'] = 'https://stackoverflow.com/';
```

Parce que tout le monde connaît l'espace d'hébergement. Donc, si vous définissez comme ça, vous **pouvez empêcher l'injection sur votre site / hôte**.

Lire URL de base dans Codeigniter en ligne: <https://riptutorial.com/fr/codeigniter/topic/3643/url-de-base-dans-codeigniter>

Chapitre 27: Utilisation de crochets

Exemples

Activation des crochets

La fonctionnalité de hooks peut être globalement activée / désactivée en définissant l'élément suivant dans le fichier `application/config/config.php` :

```
$config['enable_hooks'] = TRUE;
```

Définir un crochet

Les crochets sont définis dans le fichier `application/config/hooks.php` . Chaque hook est spécifié comme un tableau avec ce prototype

```
$hook['pre_controller'] = array(  
    'class'      => 'MyClass',  
    'function'   => 'Myfunction',  
    'filename'   => 'Myclass.php',  
    'filepath'   => 'hooks',  
    'params'     => array('beer', 'wine', 'snacks')  
);
```

L'index de tableau correspond au nom du point d'accès particulier que vous souhaitez utiliser. Dans l'exemple ci-dessus, le point d' `pre_controller` est `pre_controller` . Une liste de points d'accroche se trouve ci-dessous. Les éléments suivants doivent être définis dans votre tableau de hook associatif:

class Le nom de la classe que vous souhaitez appeler. Si vous préférez utiliser une fonction procédurale au lieu d'une classe, laissez cet élément vide.

function Fonction (ou méthode) nom que vous souhaitez appeler.

filename Le nom du fichier contenant votre classe / fonction.

chemin-fichier Nom du répertoire contenant votre script.

paramètres Tous les paramètres que vous souhaitez transmettre à votre script. Cet article est facultatif.

Crochet Points

`pre_system`

Appelé très tôt lors de l'exécution du système. Seules les classes de benchmark et de hook ont été chargées à ce stade. Aucun routage ou autre processus n'a eu lieu.

`pre_controller`

Appelé immédiatement avant l'appel de vos contrôleurs. Toutes les classes de base, routage et contrôles de sécurité ont été effectués.

`post_controller_constructor`

Appelé immédiatement après l'instanciation de votre contrôleur, mais avant tout appel de méthode.

`post_controller`

Appelé immédiatement après l'exécution complète de votre contrôleur.

`display_override`

Remplace la méthode `_display()` , utilisée pour envoyer la page finalisée au navigateur Web à la fin de l'exécution du système. Cela vous permet d'utiliser votre propre méthodologie d'affichage. Notez que vous devrez référencer le super-objet CI avec `$this->CI =& get_instance()` et que les données finalisées seront disponibles en appelant `$this->CI->output->get_output()` .

`cache_override`

Vous permet d'appeler votre propre méthode au lieu de la méthode `_display_cache()` dans la bibliothèque de sortie. Cela vous permet d'utiliser votre propre mécanisme d'affichage du cache.

`post_system`

Appelé après l'envoi de la page finale au navigateur, à la fin de l'exécution du système après l'envoi des données finalisées au navigateur.

Exemple de pré contrôleur Hook utilisant CodeIgniter

Dans le `application/hooks` , créez un fichier nommé `Blocker.php` et collez le code ci-dessous.

```
<?php
class Blocker {

    function Blocker(){
    }

    /**
     * This function used to block the every request except allowed ip address
     */
    function requestBlocker(){

        if($_SERVER["REMOTE_ADDR"] != "49.248.51.230"){
            echo "not allowed";
            die;
        }
    }
}
```


?>

Dans `application/config/hooks.php` , déclarez le hook suivant.

```
$hook['pre_controller'] = array(
    'class'     => 'Blocker',
    'function'  => 'requestBlocker',
    'filename'  => 'Blocker.php',
    'filepath'  => 'hooks',
    'params'    => ""
);
```

Dans `application/config/config.php` , définissez la valeur suivante comme true

Définir un crochet

Les crochets sont définis dans le fichier `application/config/hooks.php` . Chaque crochet est spécifié en tant que tableau avec ce prototype:

```
$hook['pre_controller'] = array(
    'class'     => 'MyClass',
    'function'  => 'Myfunction',
    'filename'  => 'Myclass.php',
    'filepath'  => 'hooks',
    'params'    => array('bread', 'wine', 'butter')
);
```

- **CLASS** - La classe que vous souhaitez appeler s'il s'agit d'un code procédural le laisse vide.
- **FUNCTION** - Le nom de la fonction que vous souhaitez appeler.
- **FILENAME** - Le nom du fichier contenant votre classe / fonction.
- **FILEPATH** - Emplacement du fichier hook.
- **PARAMS** Paramètre `PARAMS` si nécessaire il est facultatif

Lire Utilisation de crochets en ligne: <https://riptutorial.com/fr/codeigniter/topic/3953/utilisation-de-crochets>

Chapitre 28: Utiliser des sessions

Remarques

La classe **Sessions** Codeigniter utilise les cookies du navigateur pour enregistrer les données qui persisteront sur plusieurs chargements de page.

Référence: https://codeigniter.com/user_guide/libraries/sessions.html

Exemples

Créer une session

Pour initialiser une session, vous pouvez simplement la charger dans votre contrôleur, ce qui est généralement placé dans les constructions du contrôleur, mais il peut également être chargé automatiquement dans le tableau de l'application / config / autoload.php:

```
$this->load->library('session');
```

Gestion des données de session

Une session est simplement un tableau constitué des informations utilisateur suivantes:

1. L'identifiant de session unique de l'utilisateur (il s'agit d'une chaîne aléatoire statistique avec une entropie très forte, hachée avec MD5 pour la portabilité et régénérée (par défaut) toutes les cinq minutes)
2. L'adresse IP de l'utilisateur
3. Les données de l'agent utilisateur de l'utilisateur (les 120 premiers caractères de la chaîne de données du navigateur)
4. L'horodatage "dernière activité".

Source ([what-is-session-data](#))

Pour récupérer des données de session

tel que le SessionID:

```
$this->session->userdata('session_id');
```

Note - pour Codeigniter 3.x, vous pouvez utiliser la syntaxe ci-dessus, mais le concept ou les objets magiques ont été introduits, où vous pouvez utiliser `$this->session->session_id`.

Rappelez-vous que `userdata()` renvoie NULL si l'élément de session n'existe pas.

Pour récupérer toutes les données de session

```
$this->session->all_userdata()
```

Pour définir des données de session

La méthode `set_userdata()` vous permet de définir des données dans votre session. L'exemple suivant illustre un exemple de tableau que vous souhaitez insérer:

```
$newdata = array(
    'username' => 'johndoe',
    'email'    => 'johndoe@some-site.com',
    'logged_in' => TRUE
);

$this->session->set_userdata($newdata);
```

Vous pouvez également définir une donnée à la fois, par exemple:

```
$this->session->set_userdata('some_name', 'some_value');
```

ou

```
$some_name = 'some_value';
$this->session->set_userdata($some_name);
```

Pour supprimer les données de session et de session

```
$this->session->unset_userdata('some_name')
```

Cette méthode accepte également un tableau de clés d'élément à désélectionner:

Pour Codeigniter 3.x:

```
$array_items = array('username', 'email');

$this->session->unset_userdata($array_items);
```

Pour Codeigniter 2.x (cette syntaxe héritée ne prend pas en charge 3.x):

```
$array_items = array('key' => 'value');

$this->session->unset_userdata($array_items);
```

Lire Utiliser des sessions en ligne: <https://riptutorial.com/fr/codeigniter/topic/5793/utiliser-des-sessions>

Chapitre 29: Utiliser un modèle dans un identificateur de code

Exemples

Créer un modèle

Aller à l' `application/model`

Nom du fichier - **Home_model.php**

À l'intérieur du fichier

```
class Home_model extends CI_Model {  
  
    public $variable;  
  
    public function __construct()  
    {  
        parent::__construct();  
    }  
  
    public function get_data()  
    {  
        $query = $this->db->get('table_name', 10);  
        return $query->result_array();  
    }  
}
```

Et quand vous avez besoin de charger ce modèle:

```
$this->load->model('home_model');  
$this->home_model->get_data();
```

Ou Si vous souhaitez que votre modèle soit assigné à un nom d'objet différent, vous pouvez le spécifier comme ceci:

```
$this->load->model('home_model', 'home');  
$this->home->get_data();
```

Modèle de chargement

Syntaxe - `$this->load->model('model_name');`

Pratique - `$this->load->model('home_model');`

Si vous souhaitez que votre modèle soit assigné à un autre nom d'objet, vous pouvez le spécifier via le deuxième paramètre de la méthode de chargement:

Syntaxe -

```
$this->load->model('model_name', 'foobar');  
$this->foobar->method();
```

Entraîne toi -

```
$this->load->model('home_model', 'home');  
$this->home->get_data();
```

Fonction du modèle d'appel

Syntaxe

```
$this->load->model('model_name');  
$this->model_name->method_name();
```

Entraîne toi

```
$this->load->model('home_model');  
$this->home_model->get_data();
```

Passer des données au modèle

Syntaxe

```
$array = array(  
    '' => ,  
); # can pass array  
$singelData = ''; # something just a filed value  
$this->load->model('model_name');  
$this->model_name->method_name($singelData, $array);
```

Entraîne toi

```
$array = array(  
    'name' => 'codeigniter',  
    'version' => '3.0',  
    'isArray' => 'yes',  
);  
$singelData = 'using model'; # something just a filed value  
$this->load->model('home_model');  
$this->home_model->get_data($singelData, $array);
```

Recevoir des données du contrôleur

```
public function method_name($single, $array)  
{  
    echo $single;  
    print_r($array);  
}
```

Attention à l'ordre qui passe du contrôleur au modèle.

Renvoyer les données au contrôleur

```
public function get_username($uid)
{
    $query =
    $this->db->select('id')
    ->select('name')
    ->from('user_table')
    ->where('id', $uid)
    ->get();
    return $query->result_array();
}
```

Cela retournera le résultat avec l'identifiant et le nom d'utilisateur correspondants au contrôleur.

Lire Utiliser un modèle dans un identificateur de code en ligne:

<https://riptutorial.com/fr/codeigniter/topic/3777/utiliser-un-modele-dans-un-identificateur-de-code>

Chapitre 30: Validation de formulaire

Exemples

Exemple de formulaire de validation

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

$this->form_validation->set_rules('username', 'Username', 'required|max_length[20]'); // Add
validation rules for require and max
$this->form_validation->set_rules('password', 'Password', 'required|matches[password]'); //
Validation for the input match
$this->form_validation->set_rules('passconf', 'Password Confirmation', 'required');
$this->form_validation->set_rules('email', 'Email',
'required|valid_email|is_unique[userTable.emailColumn]'); // add validation for the email and
check the emailColumn in userTable for unique value
$this->form_validation->set_message('is_unique', 'The %s is already taken, Please use another
%s'); // add message for the is_unique

if ($this->form_validation->run() === FALSE)
{
    // fail
}
else
{
    // success
}
```

[Lien](#)

Lire Validation de formulaire en ligne: <https://riptutorial.com/fr/codeigniter/topic/7398/validation-de-formulaire>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec codeigniter	Abdulla Nilam , Ariful Islam , BIBIN JOHN , Bilal Ahmad , cfnerd , Community , emstawicki , gabe3886 , karel , Mitul , Prakash , Shiva127 , zur4ik
2	Appel d'une méthode de modèle dans une vue	Saul
3	Assistant de tableau	Abdulla Nilam , Rana Ghosh
4	Authentification	Ahmet Can Boyraz
5	CAPTCHA Helper	Abdulla Nilam , Rana Ghosh
6	Code pagination	Muhamad Riyan
7	CodeIgniter - Internationalisation	ImBS
8	CodeIgniter Shopping Cart	Abdulla Nilam , Hemant Sankhla , Muhamad Riyan
9	Commençons par: Hello World	Abdulla Nilam , clami219 , David , KuroKo3 , Shiva127 , Vishal
10	Comment définir un fuseau horaire dans CodeIgniter	Abdulla Nilam , Ariful Islam , Nadim Sheikh , RamenChef , rap-2-h , wolfgang1983
11	Comment utiliser les bibliothèques et l'aide CI	Abdulla Nilam , Adrian P. , karel , Mitul , NAW_AB , shantanu
12	Comment utiliser les bibliothèques et les aides CI?	Abdulla Nilam , karel , Yaseen Ahmad
13	Créer cronjob dans codeigniter sur le serveur d'hébergement Linux	Abdulla Nilam , kishor10d , Tim Duncklee
14	Créer une API dans	Kundan Prasad

	Codeigniter	
15	ensemble de session flashdata	Gopal Bhuva
16	Envoi d'email	gabe3886 , Lucifer MorningStar , Miles , Rahamathullah MK
17	Identification de code	Abdulla Nilam , ankit suthar , Mahdi Majidzadeh , Rana Ghosh
18	Image / File Uploader dans CodeIgniter	Andrey , Hanthony Tagam , Naresh Kumar .P
19	Jouez avec le mot anglais avec l'aide d'INFLECTOR	Ariful Islam
20	La gestion des erreurs	Abdulla Nilam , Rana Ghosh
21	Sécuriser votre application web	ankit suthar , BIBIN JOHN , ImBS , moopet
22	Segment URI CodeIgniter	Adrian P. , Md. Khairul Hasan
23	Structure de requête	abdurRahaman , anju , DogeAmazed , Fabio Widmer , Hemant Sankhla , kishor10d , MAZux , Rahmat
24	suffixe url	Muhamad Riyan
25	Supprimer index.php en utilisant WAMP et CodeIgniter	Ali , Ariful Islam , Gaurav , wolfgang1983
26	URL de base dans Codeigniter	Abdulla Nilam , Adrian P. , Ariful Islam , Blubberguy22 , sudopower , wolfgang1983
27	Utilisation de crochets	Abdulla Nilam , ankit suthar , kishor10d , liamja , Mitul , Rafiqui Islam
28	Utiliser des sessions	Abdulla Nilam , ElmerCat , Prakash
29	Utiliser un modèle dans un identificateur de code	Abdulla Nilam , anju , gabe3886 , Lucifer MorningStar , MAZux , Russ_AB
30	Validation de formulaire	Abdulla Nilam , ankit suthar , Lucifer MorningStar , Mitul , Murilo