

FRAMEWORK DOCUMENTATION USER GUIDE



GUIDE UTILISATEUR



BASE DE DONNEES

Guide utilisateur

Base de données

Important :

Avant d'utiliser la base de données tout d'abord, vous devez créer la connexion de base de données, donc pour créer la connexion de base de données, c'est très facile, aller vers config/config.php fichier et mettre vos informations de base de données dans le fichier config.php.

```
<?php

$database = [

    /**
     * Add your database informations
     */
    'host' => '',
    'username' => '',
    'database' => '',
    'password' => '',

];

$default = [

    'controller' => 'welcome',
    'method' => 'index',
    'param' => [],

];

$setting = [

    'base_url' => 'http://localhost:8888/cmshappyday',

];

?>
```

après avoir créé la connexion à la base de données maintenant vous pouvez communiquer avec la base de données dans le fichier config.php.

Guide utilisateur

Assistant de requête (Query Helper)

Si vous souhaitez insérer/mettre à jour/supprimer/sélectionner des données dans la base de données, vous pouvez utiliser l'assistant Requête.

```
<?php

$this->Query('Your query');
//insert query
$this->Query("INSERT INTO table (column1, column2) VALUES (?,?)", [$value1, $value2]);

//select query
$this->Query("SELECT * FROM table");

//select where query
$this->Query("SELECT * FROM table WHERE column = ?", [$value]);

//delete query
$this->Query("DELETE FROM table WHERE column = ?", [$value]);

//update query
$this->Query("UPDATE table SET column = ?, column2 = ? WHERE column = ?", [$value1, $value2, $value3]);

?>
```

Guide utilisateur

SELECT (Sélectionnez)

La requête Select ne sélectionne que les données de la table de base de données, il faut deux paramètres le premier paramètre est la table qui est importante et le second paramètre est optionnel par exemple si vous voulez passer la condition where puis passer les données where dans le second paramètre comme ['id' => 5] à voir dans second diapo →.

```
1  <?php
2
3      $this->Select('Table name', 'options');
4      // produces
5      SELECT * FROM table_name;
6
7  ?>
```

La requête WHERE est utilisée pour filtrer les enregistrements, par exemple ici nous filtrons depuis la condition where depuis la table où ['id' => 5] et ['email' => example@gmail.com].

```
1  <?php
2
3      $this->Select_Where('Table name', 'Where data');
4      //produces
5      SELECT * FROM table_name WHERE id = 5;
6      SELECT * FROM table_name WHERE email = example@gmail.com
7
8  ?>
```

Guide utilisateur

DELETE (Supprimer)

La requête DELETE permet de supprimer des données d'une table dans une base de données.

```
1  <?php
2
3      $this->Delete('Table name', 'Where Data');
4      //produces
5      DELETE FROM table_name WHERE id = 5;
6      DELETE FROM table_name WHERE email = example@gmail.com
7
8  ?>
```

Guide utilisateur

UPDATE (Mettre à jour)

La requête UPDATE permet de mettre à jour les données d'une table d'une base de données.

```
1  <?php
2
3      $this->Update('Table name', 'Set data & columns', 'options');
4      //produces
5      UPDATE table_name SET name = 'myname', email =
6      'myemail@gmail.com' WHERE id = 5;
7  ?>
```


Guide utilisateur

INSERT (Insérer)

La requête INSERT permet d'insérer des données dans une table d'une base de données.

```
1  <?php
2
3      $this->Insert('Table name', 'Columns & Values');
4      //produces
5      INSERT INTO table_name (column1, column2, column3) VALUES
6      ('value1','value2','value3');
7  ?>
```

Guide utilisateur

JOIN (Jointures)

La requête JOIN permet de récupérer plus ou moins de données dans une table ou dans une autre.

```
1  <?php
2
3      $this->Join('First Table', 'Second Table', 'Matching
4      Condition' 'Join Name');
5      //produces
6      SELECT * FROM First_table INNER JOIN Second_table ON
7      First_table.id = Second_table.id
8
9      SELECT * FROM First_table LEFT JOIN Second_table ON
10     First_table.id = Second_table.id
11
12     SELECT * FROM First_table RIGHT JOIN Second_table ON
13     First_table.id = Second_table.id
14
15 ?>
```

La requête JOIN ci-dessus prend quatre paramètres le premier paramètre est le premier nom de table, le second paramètre est le deuxième nom de table et le troisième paramètre est la condition matching soit les enregistrements qui ont des valeurs correspondantes dans deux tables comme l'id et le quatrième paramètre est optionnel qui est le nom de jointure comme LEFT JOIN, RIGHT JOIN par défaut INNER JOIN

GUIDE UTILISATEUR



ASSISTANT DE FORMULAIRE

Avant d'utiliser l'assistant de formulaire, vous devez d'abord charger l'assistant de formulaire dans le contrôleur et vous pouvez également charger à partir du fichier autoload simple vers le dossier config, puis cliquer sur le fichier autoload.php.

```
<?php
/*
 * autoload helpers
 */
$autoload = [
    'helpers' => ['form'],
];
```

form_open permet d'ouvrir la balise d'ouverture de formulaire html.

```
1  <?php echo form_open(action, method, ["class" = "" "id" = ""]);  
2  
3      //produces  
4      <form action="" method="" class="" id="">  
5  
6  ?>  
7
```

form_close permet de fermer la balise de fermeture de formulaire html.

```
1  <?php echo form_close();  
2  //produces  
3  </form>  
4  ?>
```

Guide utilisateur

form_input (champ de saisie)

form_input permet de définir le champ de saisie html comme le champ de saisie de texte, le champ de saisie d'e-mail, le champ de saisie de mot de passe et le champ de saisie de fichier

```
1  <?php echo form_input(['type' => 'text','name' => 'full_name', 'id' => 'input-id', 'class' => 'form-control',  
2  'placeholder' => 'Enter Full Name', 'value' => '']);  
3  //produces  
4  <input type="text" name="full_name" id="input-id" class="form-control" placeholder="Enter Full Name" value="">  
5  
6  <?php echo form_input(['type' => 'email','name' => 'email', 'id' => 'input-id', 'class' => 'form-control',  
7  'placeholder' => 'Enter Email', 'value' => '']);  
8  //produces  
9  <input type="email" name="email" id="input-id" class="form-control" placeholder="Enter Email" value="">  
10  
11 <?php echo form_input(['type' => 'password','name' => 'password', 'id' => 'input-id', 'class' => 'form-control',  
12 'placeholder' => 'Enter Password', 'value' => '']);  
13 //produces  
14 <input type="password" name="password" id="input-id" class="form-control" placeholder="Enter Password" value="">  
15  
16 <?php echo form_input(['type' => 'file','name' => 'file', 'id' => 'input-id', 'class' => 'form-control']);  
17 //produces  
18 <input type="file" name="file" id="input-id" class="form-control">  
19  
20  
21 ?>
```

form_multipart permet de spécifier comment les données de formulaire doivent être codées lors de leur soumission au serveur.

Remarque : L'attribut enctype ne peut être utilisé que si method = "post".

```
1  <?php echo form_multipart(action, method, ["class" = "" "id" = ""]);  
2  
3  //produce  
4  <form action="" method="" class="" id="" enctype="multipart/form-data">  
5  
6  ?>
```


Guide utilisateur

form_button (Balise de bouton HTML)

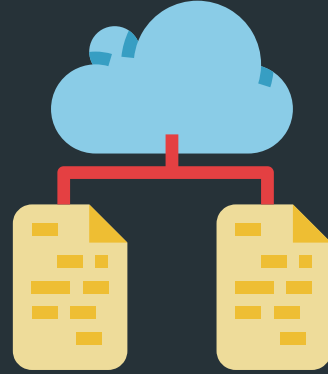
form_button permet de générer un élément de bouton standard. Vous pouvez transmettre de manière minimale le nom et le contenu du bouton dans les premier et deuxième paramètres dans le premier exemple, ou il est aussi possible de passer via un tableau associatif contenant toutes les données que vous souhaitez.

```
1 //First example
2 <?php
3     echo form_button('name','content');
4 ?>
5 //produce
6 <button name="my-btn" type="button">Content</button>
7
8 //Second example
9 <?php
10 echo form_button([
11     "name" => "my-btn",
12     "class" => "btn-default",
13     "id" => "btn-id",
14     "value" => "Content"
15 ]);
16 ?>
17 //produce
18 <button name="my-btn" type="button" class="btn-default" id="btn-id">Content</button>
19
20
21
22
```

form_submit permet de générer un élément de bouton d'envoi standard

```
1  <?php echo form_submit(["name" => "my-btn", "class" => "btn-default", "id" => "btn-id", "value" => "Save"]);
2
3  //produce
4  <input type="submit" name="my-btn" class="btn-default" id="btn-id" value="Save">
5
6
7  ?>
```

GUIDE UTILISATEUR



.HTACCESS CONFIGURATION

Guide utilisateur

Htaccess files (fichier configuration serveur Apache)

Le fichier .htaccess est un fichier de configuration du serveur Apache qui permet de définir les règles dans un dossier ou sous-dossier. Il permet notamment de définir une interdiction d'accès à un dossier ou le protéger par un mot de passe, mais permet aussi de définir des règles de redirections, etc.

Fichier htaccess pour le répertoire principal :

Créer un nouveau fichier .htaccess dans le répertoire principal et copier/coller le code ci-dessous

```
1 <IfModule mod_rewrite.c>
2     RewriteEngine On
3     RewriteRule ^$ public/ [L]
4     RewriteRule (.*?) public/$1 [L]
5 </IfModule>
```

Le fichier .htaccess est un fichier de configuration du serveur Apache qui permet de définir les règles dans un dossier ou sous-dossier. Il permet notamment de définir une interdiction d'accès à un dossier ou le protéger par un mot de passe, mais permet aussi de définir des règles de redirections, etc.

Fichier htaccess pour le répertoire racine dans le dossier public:

Créez un fichier htaccess dans le dossier public et copiez/collez le code ci-dessous dans le fichier . htaccess et ajoutez également votre nom de dossier à la place de Your Folder Name

```
1 <IfModule mod_rewrite.c>
2     Options -Multiviews
3     RewriteEngine On
4     RewriteBase /Your Folder Name/public
5     RewriteCond %{REQUEST_FILENAME} !-d
6     RewriteCond %{REQUEST_FILENAME} !-f
7     RewriteRule ^(.+)$ index.php?url=$1 [QSA,L]
8 </IfModule>
```

GUIDE UTILISATEUR



HTML HELPER

Guide utilisateur

HTML Helpers (fabrique les options HTML plus facilement et rapidement)

Le rôle du Helper HTML est de fabriquer les options du HTML plus facilement, plus rapidement.

L'utilisation de ce Helper permettra à votre application d'être plus légère, plus ancrée et plus flexible de l'endroit où il est placé en relation avec la racine de votre domaine.

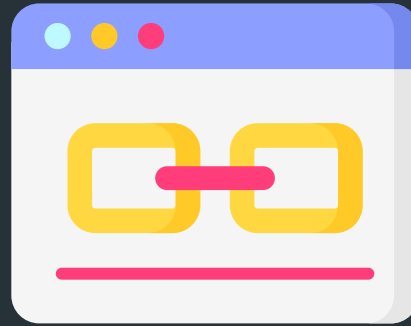
Avant d'utiliser le helper HTML, vous devez d'abord charger le helper HTML dans le contrôleur et vous pouvez également le charger à partir du fichier autoload il suffit d'aller au dossier de configuration, puis cliquez sur le fichier autoload.php.

```
1  <?php
2  ▼ /*
3      * autoload helpers
4  */
5  ▼ $autoload = [
6      'helpers' => ['html']
7  ];
8
9  ?>
```

Le rôle du Anchor Helper dans l'exemple ci-dessous est de générer du html avec du php plus facilement, plus rapidement de sorte que l'application soit plus légère, plus ancrée et plus flexible.

```
1  <?php echo anchor(url, value, ["class"="", "id"=""]);  
2  
3  //produce  
4  <a href="https://example.com/5" class="" id="">value</a>  
5  
6  ?>
```


GUIDE UTILISATEUR



AIDE A LA REDIRECTION

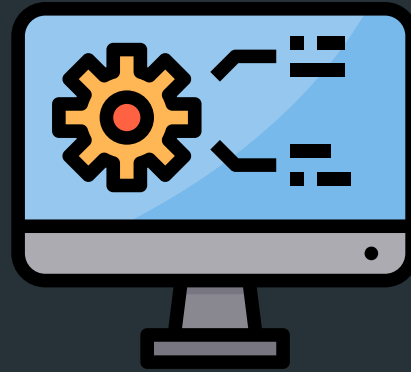
Avant d'utiliser l'assistant de redirection, vous devez d'abord charger l'assistant de redirection dans le contrôleur et vous pouvez également charger à partir du dossier de configuration automatique vers le dossier config puis cliquez sur le fichier autoload.php.

```
1  <?php
2  ▼ /*
3      * autoload helpers
4  */
5  ▼ $autoload = [
6      'helpers' => ['redirect']
7  ];
8
9  ?>
```

La redirection va produire suivant la fonction une redirection d'en-tête php voir ci-dessous :

```
1  <?php  redirect("controller/method");  
2  
3  //produce  
4  header(location="controller/method");  
5
```

GUIDE UTILISATEUR



SESSIONS

Les sessions sont un moyen simple de stocker des données individuelles pour chaque utilisateur en utilisant un identifiant de session unique. Elles peuvent être utilisées pour faire persister des informations entre plusieurs pages. Les identifiants de session sont normalement envoyés au navigateur via des cookies de session, et l'identifiant est utilisé pour récupérer les données existantes de la session. L'absence d'un identifiant ou d'un cookie de session indique à PHP de créer une nouvelle session, et génère ainsi un nouvel identifiant de session.

Il est donc possible de créer une nouvelle session, de paramétrer une session, de la mettre à jour et de la supprimer.

Nous allons voir en détail comment définir les données de session dans le diaporama suivant :

Si vous souhaitez définir les données de session, suivez l'exemple ci-dessous, le premier paramètre est le nom de la session et le second paramètre est la valeur de la session :

```
1  <?php $this->set_session('id','5'); ?>
```

Vous pouvez également transmettre un tableau de noms de session :

```
1  <?php
2
3  $data = ['id' => 5, 'name' => ' Mon Nom ', 'email' => 'example@gmail.com'];
4
5  $this->set_session($data);
6
7  ?>
```

Pour accéder aux données de la session et voir les paramètres ressortants voir ci-dessous :

```
1  <?php
2
3  echo $this->get_session('id');
4
5  ?>
```

message flash signifie par exemple si l'utilisateur crée le compte avec succès puis sur l'écran normalement l'utilisateur affiche le message de réussite, donc ci-dessous le code montre comment créer un message flash le premier paramètre est le nom de la session et le deuxième paramètre est le message flash

```
1  <?php
2
3  $this->set_flash('account_success','Your account is created successfully');
4
5  ?>
```

Le message flash prend deux paramètres, le premier paramètre est le nom de la session et le deuxième paramètre est le nom de la classe CSS, la classe CSS signifie que si vous voulez styliser le message flash, vous pouvez ajouter la classe CSS mais c'est facultatif, il est préférable de le mettre dans le fichier style.css dans le dossier public/css

```
1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5     <meta charset="UTF-8">
6     <title>CMS HAPPY DAY</title>
7     <style>
8         .alert-success {
9             border: 1px solid green;
10            color: green;
11            padding: 10px;
12        }
13
14    </style>
15 </head>
16
17 <body>
18
19     <?php $this->flash("account_success", "alert-success") ?>
20
21 </body>
22
23 </html>
24
```


`session_unset` annule la session, vous pouvez annuler une session spécifique comme les lignes 3 et 4 ou vous pouvez passer un tableau de noms de session dans lequel vous définissez les variables de sessions que vous voulez annuler cette méthode est plus légère.

Les variables de la session courante seront donc détruite, c'est à dire cette fonction retourne **TRUE** en cas de succès ou **FALSE** si une erreur survient.

```
1  <?php
2
3  $this->session_unset('id');
4  $this->session_unset('name');
5
6  //OR
7
8  $data = ['id','name'];
9  $this->session_unset($data);
10
11  ?>
```

`session_destroy()` détruit toutes les données associées à la session courante. Cette fonction ne détruit pas les variables globales associées à la session, de même, elle ne détruit pas le cookie de session. Pour accéder à nouveau aux variables de session, la fonction `session_start()` doit être appelée de nouveau.

Note: Vous n'avez pas besoin d'appeler `session_destroy()` depuis le programme généralement. Nettoyer le tableau `$_SESSION` plutôt que de détruire les données de session.

```
1  <?php
2
3  $this->destroy_session();
4
5  ?>
```

```
1  <?php
2  // Initialization of the session.
3
4  session_start();
5
6  // Destroys all session variables more simply
7
8  $_SESSION = array();
9
10 ?>
11
```

GUIDE UTILISATEUR



BIBLIOTHEQUE DE TELECHARGEMENT DE FICHIERS

Bibliothèque de téléchargement de fichiers, téléchargement de fichiers / images et validation des fichiers / images

Utilisation de la bibliothèque de téléchargement de fichiers

il suffit d'appeler la méthode de fichier et de leur passer un tableau

```
1  <?php
2
3  ▼    $data = [
4
5          'file_name' => 'image',
6          'allowed_extensions' => 'jpg|png|gif',
7          //'allowed_extensions' => 'pdf|docx',
8          'upload_path' => 'images/',
9          'label' => 'Image'
10
11    ];
12
13    $this->file($data);
14
15  ?>
```

Pour afficher l'erreur de téléchargement de fichier, tout d'abord dans la méthode `run_file ()` de l'appel du contrôleur si cela est vrai, cela signifie que vous n'avez aucune erreur si faux, cela signifie que vous avez une erreur

```
1  <?php
2
3  ▼    if($this->file_run()){
4
5        print_r($this->file_data['file_name']);
6
7  ▼    } else {
8
9        //You have an error
10       $this->view('user_view');
11
12       }
13
14  ?>
```

Guide utilisateur

Afficher l'erreur dans la vue

Pour afficher l'erreur dans la vue c'est à dire dans le html voir ci-dessous :

```
1  <!DOCTYPE html>
2  <html lang='fr'>
3
4  <head>
5      <meta charset='UTF-8'>
6      <title>User View</title>
7  </head>
8
9  <body>
10     <h1>User View</h1>
11
12     <?php echo form_multipart("controller/method", "post", ['class' => 'form-class']); ?>
13     <?php echo form_input(['type' => 'file', 'name' => 'image', 'id' => 'input-id', 'class' => 'form-control']); ?>
14     <br>
15     <?php if(!empty($this->file_errors['image'])): ?>
16     <?php echo $this->file_errors['image']; ?>
17     <?php endif; ?><br>
18 </body>
19 </html>
20
```

GUIDE UTILISATEUR



ASSISTANT URL

Guide utilisateur

Chargement de l'assistant URL / link_css / link_js

Avant d'utiliser l'assistant URL, vous devez d'abord charger l'assistant URL dans le contrôleur et vous pouvez également charger à partir du fichier autoload vers le dossier config, puis cliquer sur le fichier autoload.php.

```
1  <?php
2
3  ▾    /*
4        * autoload helpers
5        */
6  ▾    $autoload = [
7
8        'helpers' => ['url']
9
10     ];
11
12  ?>
```

link_css permet de produire la balise de lien externe html / link_js permet de produire celle js

```
1  <?php
2
3  echo link_css("public/css/style.css");
4
5  //produce
6  <link rel="stylesheet" href="public/css/style.css">
7
8  ?>
9
```

```
1  <?php
2
3  echo link_js("public/js/app.js");
4
5  //produce
6  <script src="public/js/app.js"></script>
7
8  ?>
```


GUIDE UTILISATEUR



BIBLIOTHEQUE DE VALIDATION DES FORMULAIRES

Guide utilisateur

Utilisation de la bibliothèque de validation de formulaire / Règles de validation

En utilisant une bibliothèque de framework, vous n'avez pas besoin de charger la bibliothèque en premier, vous pouvez utiliser la bibliothèque de framework sans la charger

```
1  <?php
2
3  $this->validation('input_field_name','Label','Rules');
4
5  ?>
```

Règles de validation :

- 1- obligatoire
- 2- not_int
- 3- int
- 4- min_len
- 5- max_len
- 6- confirmer
- 7- unique

```
1  Règle requise
2
3  <?php
4
5  $this->validation('input_field_name','Label','required');
6
7  ?>
8
9  règle not_int
10
11  la règle not_int signifie que la valeur doit être un caractère alphabétique et non un entier
12
13  <?php
14
15  $this->validation('input_field_name','Label','not_int');
16
17  ?>
18
19  règle int
20
21  la règle int signifie que la valeur doit être un entier
22
23  <?php
24
25  $this->validation('input_field_name','Label','int');
26
27  ?>
```

Guide utilisateur

Utilisation de la bibliothèque de validation de formulaire / Règles de validation

```
1  règle min_len
2
3  après la règle min_len, ajoutez le signe | et ajoutez immédiatement la valeur de longueur minimale
4
5  <?php
6
7  $this->validation('input_field_name','Label','min_len|3');
8
9  ?>
10
11 règle max_len
12
13 après la règle max_len, ajoutez le signe | et ajoutez immédiatement la valeur de longueur maximale
14
15 <?php
16
17 $this->validation('input_field_name','Label','max_len|8');
18
19 ?>
```

```
1  confirmer la règle
2
3  Confirmer la règle que vous pouvez utiliser uniquement pour la confirmation du mot de passe pour confirmer les deux
   champs de mot de passe comme le champ de mot de passe et le champ de confirmation de mot de passe. Après la règle de
   confirmation, ajoutez le signe | et ajoutez immédiatement le nom du champ de mot de passe
4
5  <?php
6
7  $this->validation('input_field_name','Label','confirm|password');
8
9  ?>
10
11 règle unique
12
13 une règle unique signifie vérifier la disponibilité des e-mails dans la table de base de données, ils sont
   disponibles ou non dans la table, alors ajoutez d'abord le nom de la règle après le signe | et enfin ajoutez le nom
   de la table de base de données
14
15 <?php
16
17 $this->validation('input_field_name','Label','unique|users');
18
19 ?>
```

Guide utilisateur

Rassembler toutes les règles

Voici ci-dessous l'exemple de la façon dont vous pouvez utiliser des règles pour rassembler les différentes règles :

```
1  <?php
2
3  $this->validation('input_field_name','Label','required|min_len|4|');
4
5  ?>
6
7  <?php
8
9  $this->validation('input_field_name','Label','required|max_len|8|not-int');
10
11  ?>
12
13  <?
14
15  php $this->validation('input_field_name','Label','required|confirm|password');
16
17  ?>
18
19  <?php
20
21  $this->validation('input_field_name','Label','required|unique|users');
22
23  ?>
24
```

Guide utilisateur

Erreurs d'affichage dans la vue

Ici dans le contrôleur, appelez `run ()`; cette méthode vérifie si c'est vrai, cela signifie que vous n'avez aucune erreur aussi non

il affiche l'erreur.

```
1  <?php
2
3  ▼    if($this->run()){
4
5          //Success no error
6
7  ▼    } else {
8
9          //You have an errors
10         $this->view('user_view');
11
12     }
13
14     ?>
15
```

```
1  <!DOCTYPE html>
2  ▼ <html lang='fr'>
3
4  ▼ <head>
5      <meta charset='UTF-8'>
6      <title>User View</title>
7  </head>
8
9  ▼ <body>
10     <h1>User View</h1>
11
12     <?php echo form_open("controller/method", "post", ['class' => 'form-class']); ?>
13
14
15     <?php echo form_input(['type' => 'text', 'name' => 'full_name', 'id' => 'input-id', 'class' => 'form-control',
16         'placeholder' => 'Enter Full Name', 'value' => $this->set_value('full_name')]); ?> <br>
17     <?php if(!empty($this->errors['full_name'])): ?>
18     <?php echo $this->errors['full_name']; ?>
19     <?php endif; ?><br>
20 </body>
21 </html>
22
```