



EBook Gratis

APRENDIZAJE codeigniter

Free unaffiliated eBook created from
Stack Overflow contributors.

#codeigniter

Tabla de contenido

Acerca de.....	1
Capítulo 1: Primeros pasos con codeigniter	2
Observaciones.....	2
Versiones.....	2
Examples.....	2
Instalación y configuración.....	2
Entorno de Windows	2
Entorno Mac.....	2
Entorno Linux.....	3
GitHub.....	3
URL base.....	3
Eliminar index.php de la URL.....	3
Configuración de la base de datos	4
Establecer controlador predeterminado.....	5
AutoLoad Library and Helper.....	5
Ejecutar múltiples aplicaciones en un sistema CI.....	5
Estructura de la carpeta:.....	5
Configuración de Codeigniter :.....	6
Aumente la seguridad ocultando la ubicación de sus archivos CodeIgniter.....	6
Capítulo 2: ¿Cómo usar las bibliotecas y ayudantes de CI?	8
Sintaxis.....	8
Examples.....	8
Creando y llamando a una biblioteca.....	8
Capítulo 3: Asegurando su aplicación web.....	9
Introducción.....	9
Sintaxis.....	9
Parámetros.....	9
Examples.....	9
Prevención XSS.....	9

Prevencción de inyección SQL.....	9
Ocultar errores de PHP.....	10
Prevencción CSRF.....	11
Eliminar datos de abuso de la entrada del usuario.....	11
Prevencción XSS en la entrada del usuario.....	11
Capítulo 4: Autenticación.....	13
Examples.....	13
Cargando su biblioteca de Autores para cada controlador.....	13
Capítulo 5: Ayudante de matriz.....	14
Introducción.....	14
Examples.....	14
Cargando este Ayudante.....	14
Capítulo 6: CAPTCHA ayudante.....	16
Examples.....	16
Cargando este Ayudante.....	16
create_captcha (\$ data).....	16
Usando el ayudante de CAPTCHA.....	16
Ejemplo completo.....	17
Capítulo 7: CodeIgniter - Internacionalización.....	19
Introducción.....	19
Examples.....	19
Ejemplo.....	19
Capítulo 8: CodeIgniter Carrito de compras.....	23
Introducción.....	23
Examples.....	23
Agregar artículos en el carrito.....	23
y puede insertar varios elementos como este.....	23
Elementos básicos de la cesta de compras de CI.....	24
Mostrar artículos de carrito.....	24
Actualizar artículo de carrito.....	25
Eliminar artículos del carrito.....	25
Capítulo 9: CodeIgniter URI Segment.....	26

Examples.....	26
Segmentos URI:.....	26
Obtener el último y último segmento de URI.....	26
Capítulo 10: Cómo configurar la zona horaria en CodeIgniter.....	27
Examples.....	27
Cómo configurar la zona horaria en CodeIgniter.....	27
Otra forma de configurar la zona horaria en codeigniter.....	28
Capítulo 11: Cómo utilizar las bibliotecas y el ayudante de CI.....	30
Examples.....	30
Usando bibliotecas y ayudantes.....	30
Ayudante.....	31
Capítulo 12: Creando cronjob en codeigniter en un servidor de alojamiento linux.....	33
Examples.....	33
Llamando a un controlador CodeIgniter desde cron.....	33
Cronjob en Codeigniter.....	33
Capítulo 13: Eliminando index.php usando WAMP y CodeIgniter.....	35
Examples.....	35
Cómo eliminar el index.php de url con el uso de wamp y codeigniter.....	35
Capítulo 14: Empecemos: Hola Mundo.....	37
Examples.....	37
Una aplicación muy simple de Hello World.....	37
Vamos a usar el controlador un poco más.....	38
Vamos a elegir nuestros saludos: ¿Hola Mundo o Adiós Mundo o ...?.....	39
Capítulo 15: Enviando email.....	41
Observaciones.....	41
Examples.....	41
Cargar la biblioteca de correo electrónico.....	41
Establezca sus parámetros de configuración de correo electrónico.....	41
Crea tu correo electrónico.....	42
Envía tu e-mail.....	42
Enviar un correo electrónico HTML.....	42
Formulario de contacto.....	43

Capítulo 16: Estructura de consulta	46
Examples	46
Seleccionando datos	46
Seleccionando datos	46
Selección de datos con el segundo parámetro opcional	47
Unir tablas usando el generador de consultas	47
Capítulo 17: Hacer API en Codeigniter	49
Introducción	49
Examples	49
crear el nuevo controlador con nombre API	49
Recupere algunos datos de la API agregue la siguiente función en el controlador API	49
inicie sesión en la API de usuario para permitir el acceso de algunos datos privados para	50
el usuario cierra sesión en la API para destruir la sesión del usuario que ha iniciado ses	52
crear api protegido	53
Capítulo 18: Image / File Uploader en CodeIgniter	55
Observaciones	55
Examples	55
Un solo archivo / cargador de imágenes	55
Capítulo 19: Juega con la palabra inglesa con el ayudante INFLECTOR	58
Introducción	58
Examples	58
Auxiliar de carga del inflector	58
Hacer una palabra singular	58
Comprobar que una palabra tiene plural	58
Hacer una palabra plural	58
Camelizado la cuerda	58
Eliminar / Añadir delimitador entre palabras	59
Eliminar delimitador	59
Añadir subrayado	59
Capítulo 20: Llamar a un método modelo en una vista	60
Introducción	60
Examples	60

Guardar una llamada de método en una variable.....	60
Capítulo 21: Manejo de errores.....	61
Introducción.....	61
Examples.....	61
show_error ().....	61
Sintaxis.....	61
Fuente.....	61
show_404 ().....	61
Sintaxis.....	62
Fuente.....	62
log_message ().....	62
Sintaxis.....	62
Hay tres tipos de mensajes:.....	63
Capítulo 22: Paginación del codificador.....	64
Examples.....	64
En esta sección asumo que usted conoce al ayudante que llama, en el controlador.....	64
en la vista.....	64
Capítulo 23: Resolución de problemas de Codeigniter.....	65
Introducción.....	65
Examples.....	65
Solución de problemas.....	65
Capítulo 24: sesión set flashdata.....	66
Examples.....	66
Cómo configurar los datos flash de sesión en el controlador.....	66
Cómo mostrar Flashdata en la vista.....	66
Capítulo 25: sufijo de url.....	67
Examples.....	67
sufijo de url.....	67
Capítulo 26: URL base en Codeigniter.....	68
Examples.....	68
Configuración de su url base en Codeigniter.....	68

Algo más sobre base_url.....	69
¿Qué pasa si no configuro base_url ?.....	69
Si no lo puse ¿qué aparecerá?.....	69
¿Qué significa esto http://[:1]/ ??.....	69
¿Cómo establecer una base_url() adecuada base_url() ?.....	69
¿Cómo usar base_url() ?.....	70
Forma inteligente de configurar el base_url.....	70
Capítulo 27: Usando el modelo en codeigniter.....	71
Examples.....	71
Creando Modelo.....	71
Cargando modelo.....	71
Función de modelo de llamada.....	72
Pasando datos al modelo.....	72
Recibiendo datos del controlador.....	72
Devolver datos al controlador.....	73
Capítulo 28: Usando Sesiones.....	74
Observaciones.....	74
Examples.....	74
Crear una sesión.....	74
Manejo de datos de sesión.....	74
Para recuperar datos de sesión.....	74
Para establecer datos de sesión.....	75
Para eliminar sesión y datos de sesión.....	75
Capítulo 29: Uso de ganchos.....	76
Examples.....	76
Habilitar ganchos.....	76
Definiendo un gancho.....	76
Puntos de gancho.....	76
pre_system.....	76
pre_controller.....	77
post_controller_constructor.....	77

post_controller.....	77
display_override.....	77
cache_override.....	77
post_system.....	77
Ejemplo de Pre Controller Hook usando CodeIgniter.....	77
Definiendo un gancho.....	78
Capítulo 30: Validación de formularios.....	79
Examples.....	79
Validar ejemplo de formulario.....	79
Creditos.....	80

Acerca de

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

Capítulo 1: Primeros pasos con codeigniter

Observaciones

Codeigniter es un framework MVC escrito en, y para, PHP.

Es liviano en comparación con otros marcos MVC que existen, al costo de tener menos funcionalidad, por ejemplo, no hay un sistema de autenticación integrado que pueda ser parte de otros marcos.

Codeigniter es una buena opción de marcos para aquellos que están comenzando con MVC, ya que no obliga a ningún estándar en particular para nombrar y estructurar el código; pero también es adecuado para proyectos más grandes donde no se necesita una gran variedad de características contenidas en otros marcos.

Versiones

Versión	Fecha de lanzamiento
Versión Beta 1.0	2006-02-28
Versión 2.0.0	2011-01-28
Versión 2.2.0	2014-06-02
Versión 3.0.0	2015-03-30
Versión 3.1.3	2017-01-09
La versión es 3.1.4	2017-03-20
La versión es 3.1.5	2017-06-19

Examples

Instalación y configuración

Entorno de Windows

1. Instalar [XAMPP](#) o [WAMP](#)
2. Descarga y descomprime el paquete de [Codeigniter.com](#)
3. Extraiga todo el documento en el espacio del servidor (directorio htdocs o www)

Entorno Mac

1. Instalar [MAMP](#)
2. Descarga y descomprime el paquete de [Codeigniter.com](#)
3. Extraer todo el documento en el espacio del servidor (htdocs)

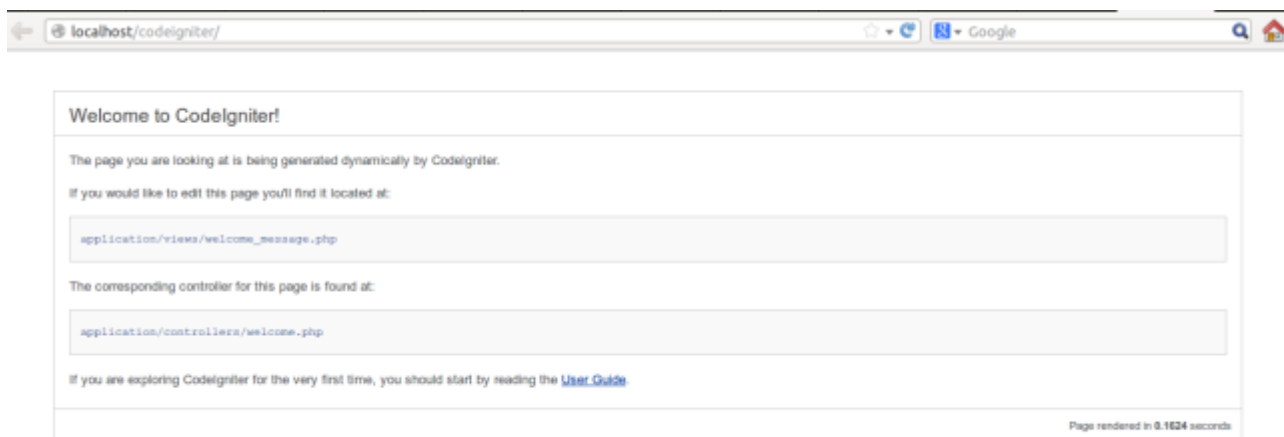
Entorno Linux

1. Descarga y descomprime el paquete de [Codeigniter.com](#)
2. Coloque la carpeta extraída en / var / www (en WAMP) o xampp / htdocs (XAMPP)

GitHub

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

Si sigue el sistema correctamente, aparecerá la siguiente pantalla.



URL base

1. Ir a la `application/config/config.php`
2. Defina la URL base como `$config['base_url'] = 'http://localhost/path/to/folder';`

Eliminar `index.php` de la URL

Configuración de Apache

1. ir a la raíz
2. crear archivo htaccess
3. Agregue el siguiente código dentro de él

```
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]
```

Nota: el código .htaccess varía según el servidor de alojamiento. En algún servidor de alojamiento (por ejemplo: GoDaddy), ¿necesita usar un extra ? En la última línea del código anterior. La siguiente línea será reemplazada con la última línea en caso aplicable:

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

Configuración de Nginx

1. Abra el archivo de configuración nginx (de forma predeterminada: /etc/nginx/sites-available/default)
2. Agregue el siguiente código dentro de él

```
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;
    }
}
```

Configuración de la base de datos

1. Ir a la application/config/database.php

2. Establecer las siguientes variables de configuración.

- Anfitrión
- Nombre de usuario
- Contraseña
- Nombre de la base de datos
- Puerto

Establecer controlador predeterminado

1. Ir a la `application/config/routes.php`
2. establezca el siguiente valor de variable de configuración con el nombre de su controlador.
 - `controlador_predeterminado`

AutoLoad Library and Helper

1. Ir a la `application/config/autoload.php`
2. establezca el valor de carga automática como `$autoload['libraries'] = array('database', 'session');`
3. establezca el valor de Ayuda como `$autoload['helper'] = array('url', 'file', 'form', 'html', 'text');`

Ejecutar múltiples aplicaciones en un sistema CI

Codeigniter puede configurarse para ejecutar más de un proyecto sin duplicar los archivos centrales de CI.

Es posible dividiendo el lado de la aplicación CI. Por ejemplo tomemos proyecto de página web, que contiene `front-end` y `back-end` las aplicaciones del Sistema de Gestión de Contenidos (CMS). En este caso, la estructura de la carpeta CI será como:

Estructura de la carpeta:

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

```

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

```

En la carpeta de `applications` , creamos dos carpetas: `front-end` y `back-end` y copiamos todo el contenido predeterminado de las `applications` en estas dos carpetas.

También `backend.php` archivo `index.php` en la carpeta raíz como `backend.php`

Lo siguiente es configurar el `CI` para trabajar con estas dos instancias de aplicación.

Configuración de Codeigniter :

Abra los archivos `index.php` y `backend.php` y actualice `application_folder` config:

```

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

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

```

Después de la configuración anterior, `CI` está listo para ejecutar dos aplicaciones bajo un sistema `CI`:

Solicitud en `example.com/Codeigniter/index.php` se abrirá la aplicación `front-end`

La solicitud en `example.com/Codeigniter/backend.php` abrirá `back-end` aplicación `back-end`

Aumente la seguridad ocultando la ubicación de sus archivos CodeIgniter

Dentro del CodeIgniter, hay dos directorios principales de los que preocuparse: **sistema** y **aplicación** . La carpeta del sistema contiene las entrañas del núcleo de CodeIgniter. La carpeta de la aplicación contendrá todo el código específico de su aplicación, incluidos los modelos, controladores, vistas y otras bibliotecas relevantes.

De acuerdo con las [instrucciones de instalación de](#) CodeIgniter, en el mejor interés de proteger su aplicación, tanto el sistema como la carpeta de la aplicación deben ubicarse sobre la raíz web para que no sean accesibles directamente a través de un navegador. De forma predeterminada, los archivos `.htaccess` se incluyen en cada carpeta para ayudar a evitar el acceso directo, pero es mejor eliminarlos del acceso público por completo en caso de que la configuración del servidor web cambie o no cumpla con el `.htaccess`.

```

|   |   |   | CodeIgniter
|   |   |   | application
|   |   |   | system

```

```
|   |— wwwroot
|   |   |— index.php
```

Después de mover las carpetas del sistema y de la aplicación, abra el archivo principal `index.php` y configure las `$system_path` , `$application_folder` , preferiblemente con una ruta completa, por ejemplo, `'/www/MyUser/system'` . Sin embargo, los caminos relativos deberían funcionar.

Para Linux / Apache:

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

Para Windows / IIS:

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

Lea Primeros pasos con codeigniter en línea:

<https://riptutorial.com/es/codeigniter/topic/929/primeros-pasos-con-codeigniter>

Capítulo 2: ¿Cómo usar las bibliotecas y ayudantes de CI?

Sintaxis

1. `$ this-> load-> library ('library_name');`
2. `$ this-> library_name-> function_name ();`
3. `$ this-> load-> library ('cart'); # para helper $ this-> load-> helper ('helperName');`
4. `$ this-> cart-> insert ($ Array);`

Examples

Creando y llamando a una biblioteca

Para utilizar las bibliotecas en CodeIgniter, necesita crear una biblioteca.

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

En esta biblioteca, a la que se llama es pro.php, este archivo debe agregarse a la siguiente ruta.

Ruta: \xampp\htdocs\project\application\varies

Ahora puedes usarlo en tu controlador. Código para cargar esta librería en el controlador:

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

Código para usar las funciones de la biblioteca:

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

Lea ¿Cómo usar las bibliotecas y ayudantes de CI? en línea:

<https://riptutorial.com/es/codeigniter/topic/3770/-como-usar-las-bibliotecas-y-ayudantes-de-ci->

Capítulo 3: Asegurando su aplicación web

Introducción

Recordar CodeIgniter es un Framework de desarrollo. No se esfuerza por hacer que su aplicación sea segura. Simplemente te da las herramientas para hacerlo tú mismo. Si miras la página de seguridad de CI, es bastante claro que esperan que el desarrollador entienda la seguridad de la aplicación y la incorpore en su aplicación.

Si la seguridad de la aplicación web es relativamente nueva para usted, comenzaría con OWASP. Podría ser ventajoso ver otros marcos como Zend o Cake que creo que hacen más cosas por adelantado

Sintaxis

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

Parámetros

matriz de entrada del usuario	blanco
inserte una matriz de entrada de usuario en <code>xss_filter(\$array of user input)</code>	Blanco

Examples

Prevención XSS

XSS significa scripts entre sitios. CodeIgniter viene con seguridad de filtrado XSS. Este filtro evitará cualquier código JavaScript malicioso o cualquier otro código que intente secuestrar cookies y realizar actividades maliciosas. Para filtrar datos a través del filtro XSS, use el método `xss_clean()` como se muestra a continuación.

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

Debe utilizar esta función solo cuando está enviando datos. El segundo parámetro booleano opcional también se puede usar para verificar el archivo de imagen para el ataque XSS. Esto es útil para la facilidad de carga de archivos. Si su valor es verdadero, significa que la imagen es segura y no de otra manera.

Prevención de inyección SQL

La inyección SQL es un ataque realizado en la consulta de la base de datos. En PHP, usamos la función `mysql_real_escape_string()` para evitar esto junto con otras técnicas, pero CodeIgniter

proporciona funciones y bibliotecas incorporadas para evitar esto.

Podemos evitar la inyección SQL en CodeIgniter de las siguientes tres formas:

- Consultas de escape
- Consulta de consultas
- Clase de registro activo

Consultas de escape

```
<?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()` agrega automáticamente comillas simples alrededor de los datos y determina el tipo de datos para que solo pueda escapar datos de cadenas.

Consulta de consultas

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

En el ejemplo anterior, el signo de interrogación (?) Será reemplazado por la matriz en el segundo parámetro de la función de `query()`. La principal ventaja de generar consultas de esta manera es que los valores se escapan automáticamente, lo que produce consultas seguras. El motor de CodeIgniter lo hace por usted automáticamente, por lo que no tiene que recordarlo.

Clase de registro activo

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

Usando registros activos, la sintaxis de consulta es generada por cada adaptador de base de datos. También permite consultas más seguras, ya que los valores se escapan automáticamente.

Ocultar errores de PHP

En el entorno de producción, a menudo no queremos mostrar ningún mensaje de error a los usuarios. Es bueno si está habilitado en el entorno de desarrollo para propósitos de depuración. Estos mensajes de error pueden contener cierta información, que no debemos mostrar a los usuarios del sitio por razones de seguridad.

Hay tres archivos CodeIgniter relacionados con errores. Nivel de informe de errores de PHP

Un entorno diferente requiere diferentes niveles de informe de errores. Por defecto, el desarrollo mostrará errores, pero las pruebas y en vivo los ocultarán. Hay un archivo llamado `index.php` en el directorio raíz de CodeIgniter, que se usa para este propósito. Si pasamos cero como argumento a la función `error_reporting()`, eso ocultará todos los errores.

Prevención CSRF

CSRF significa falsificación de solicitud entre sitios. Puede evitar este ataque habilitando una opción en el archivo `application / config / config.php` como se muestra a continuación.

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

Cuando creas un formulario usando la función `form_open()`, insertará automáticamente un token CSRF en un campo oculto. También puede agregar manualmente el token CSRF usando la función `get_csrf_token_name()` y `get_csrf_hash()`. Como sus nombres sugieren, la función `get_csrf_token_name()` devolverá el nombre del token CSRF, mientras que `get_csrf_hash()` devolverá el hash.

El token CSRF se puede regenerar cada vez que se envía o también puede mantenerse igual durante toda la vida de la cookie CSRF. La configuración de la opción de configuración `'csrf_regenerate'` forzará la regeneración del token como se muestra a continuación.

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

Puede incluir en la lista blanca las URL de la protección CSRF configurando las coincidencias para ellas en la matriz de configuración usando la clave `'csrf_exclude_uris'` como se muestra a continuación. También puedes usar expresiones regulares.

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

Eliminar datos de abuso de la entrada del usuario

```
// 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); ?>
```

Prevención XSS en la entrada del usuario

No confíe en ninguna entrada del usuario. el usuario ingresa todo como la etiqueta `<script>` o cualquier `alert()`; `javascript alert()`; así que tenemos que evitar que todos los datos no se ejecuten en nuestro navegador. por lo que tenemos que usar el método de prevención xss para restringir nuestros datos seguros y mantenerlos en manos de piratas informáticos, y también es

responsabilidad del desarrollador la validación de las entradas del usuario y la resolución de errores mediante programación.

Entonces, verifique que este es un ejemplo de prevención xss en 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
```

por lo tanto, después de agregar `xss_filtering`, no tenemos ningún problema para ejecutar ningún código de abuso que ingrese el usuario. y CodeIgniter reemplaza esta etiqueta de abuso con la palabra clave `[removed]` .

Lea Asegurando su aplicación web en línea:

<https://riptutorial.com/es/codeigniter/topic/9857/asegurando-su-aplicacion-web>

Capítulo 4: Autenticación

Examples

Cargando su biblioteca de Autores para cada controlador

vaya a codeigniter / application / library / cree o reemplace sus archivos de biblioteca aquí.

vaya a codeigniter / application / core / cree un nuevo archivo php llamado como MY_Controller.php

dentro 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
    }
}
```

Y luego, en cada archivo de controlador que necesite para extender MY_Controller.

Ejemplo de un controlador; Vaya a codeigniter / application / controllers y cree un archivo 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('/'));
        }
    }
}
```

Lea Autenticación en línea: <https://riptutorial.com/es/codeigniter/topic/7722/autenticacion>

Capítulo 5: Ayudante de matriz

Introducción

El archivo de ayuda de arreglos contiene funciones que ayudan a trabajar con arreglos.

Examples

Cargando este Ayudante

Este ayudante se carga utilizando el siguiente código:

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

Las siguientes funciones están disponibles:

elemento()

Le permite obtener un elemento de una matriz. La función comprueba si el índice de matriz está establecido y si tiene un valor. Si existe un valor se devuelve. Si un valor no existe, devuelve FALSO, o lo que haya especificado como el valor predeterminado a través del tercer parámetro. Ejemplo:

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

random_element ()

Toma una matriz como entrada y devuelve un elemento aleatorio de ella. Ejemplo de uso:

```
$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);
```

elementos()

Le permite obtener una serie de elementos de una matriz. La función comprueba si cada uno de los índices de matriz está establecido. Si no existe un índice, se establece en FALSE, o lo que haya especificado como el valor predeterminado a través del tercer parámetro. Ejemplo:

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

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

Lo anterior devolverá la siguiente matriz:

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

Puede configurar el tercer parámetro a cualquier valor predeterminado que desee:

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

Lo anterior devolverá la siguiente matriz:

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

Esto es útil cuando se envía la matriz `$_POST` a uno de sus Modelos. Esto evita que los usuarios envíen datos POST adicionales a ser ingresados en sus tablas:

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

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

Esto garantiza que solo los campos de identificación, título y contenido se envíen para ser actualizados.

Lea Ayudante de matriz en línea: <https://riptutorial.com/es/codeigniter/topic/8068/ayudante-de-matriz>

Capítulo 6: CAPTCHA ayudante

Examples

Cargando este Ayudante

Este ayudante se carga utilizando el siguiente código:

En el propio controlador (* se puede repetir una y otra vez *)

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

En config/autoload.php (*cargar solo una vez*)

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

create_captcha (\$ data)

Toma una matriz de información para generar el CAPTCHA como entrada y crea la imagen según sus especificaciones, devolviendo una matriz de datos asociativos sobre la imagen.

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

La "imagen" es la etiqueta de imagen real:

```

```

El "tiempo" es la marca de tiempo micro utilizada como nombre de la imagen sin la extensión de archivo. Será un número como este: 1139612155.3422

La "palabra" es la palabra que aparece en la imagen captcha, que si no se proporciona a la función, será una cadena aleatoria.

Usando el ayudante de CAPTCHA

Una vez cargado puedes generar un captcha como este:

```
$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 función captcha requiere la librería de imágenes GD.
- Solo se requieren `img_path` e `img_url`.
- Si no se proporciona una "palabra", la función generará una cadena ASCII aleatoria. Puede armar su propia biblioteca de palabras de la que puede dibujar al azar.
- Si no especifica una ruta de acceso a una fuente TRUE TYPE, se utilizará la fuente GD feo nativa. La carpeta "captcha" debe ser de escritura (666 o 777)
- La "caducidad" (en segundos) significa cuánto tiempo permanecerá una imagen en la carpeta captcha antes de que se elimine. El valor predeterminado es dos horas.

Ejemplo completo

Aquí hay un ejemplo de uso con una base de datos. En la página donde se mostrará el CAPTCHA tendrá algo como esto:

```

$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="" />';

```

Luego, en la página que acepta el envío tendrás algo como esto:

```

// 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";
}
```

Lea CAPTCHA ayudante en línea: <https://riptutorial.com/es/codeigniter/topic/7902/captcha-ayudante>

Capítulo 7: CodeIgniter - Internacionalización

Introducción

La clase de idiomas en CodeIgniter proporciona una manera fácil de admitir múltiples idiomas para la internacionalización. Hasta cierto punto, podemos usar archivos de diferentes idiomas para mostrar texto en muchos idiomas diferentes.

Examples

Ejemplo

Creación de archivos de lenguaje

Para crear un archivo de idioma, debe finalizarlo con `_lang.php` . Por ejemplo, si desea crear un archivo de idioma para el idioma francés, debe guardarlo con `french_lang.php` . Dentro de este archivo puede almacenar todos sus textos de idioma en clave, combinación de valor en `$lang` array como se muestra a continuación.

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

Cargando archivo de idioma

Para usar cualquiera de los idiomas en su aplicación, primero debe cargar el archivo de ese idioma en particular para recuperar varios textos almacenados en ese archivo. Puede utilizar el siguiente código para cargar el archivo de idioma.

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

nombre de archivo : es el nombre del archivo que desea cargar. No use la extensión del archivo aquí, solo el nombre del archivo. **Idioma** : es el conjunto de idiomas que lo contiene.

Obtención de texto de idioma

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

Para obtener una línea del archivo de idioma, simplemente ejecute el siguiente código. Donde **language_key** es el parámetro clave utilizado para obtener el valor de la clave en el archivo de idioma cargado.

Autoload Languages

Si necesita un idioma global, puede cargarlo `application/config/autoload.php` archivo `application/config/autoload.php` como se muestra a continuación.

```

| -----
|   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();

```

Simplemente, pase los diferentes idiomas para ser cargados automáticamente por CodeIgniter.

Cree un controlador llamado `Lang_controller.php` **y guárdelo en** `application/controller/Lang_controller.php`

```
<?php
```

la clase `Lang_controller` extiende `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);
}
}
?>

```

Cree un archivo de vista llamado `lang_view.php` **y guárdelo en** `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');

```

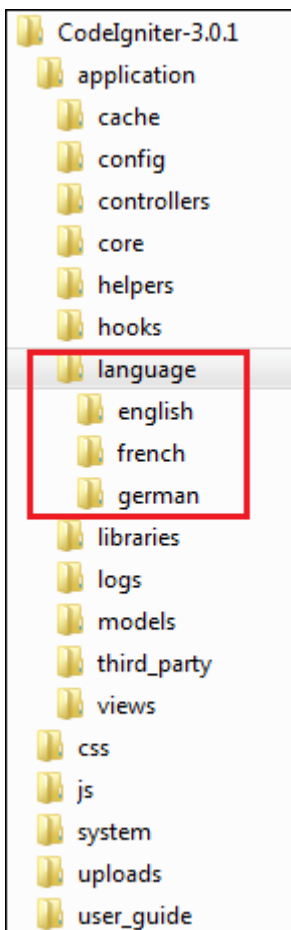
```

?>
<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>

```

Cree tres carpetas llamadas **inglés** , **francés** y **alemán** en la **aplicación / idioma** como se muestra en la siguiente figura.



Copie el código dado a continuación y guárdelo en el archivo `english_lang.php` en la carpeta `application/language/english` .

```

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

```

```
?>
```

Copie el código dado a continuación y guárdelo en el archivo `french_lang.php` en la carpeta `application/language/French` .

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

Copie el código dado a continuación y guárdelo en el archivo `german_lang.php` en la carpeta `application/language/german` .

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

Cambie el archivo `routes.php` en `application/config/routes.php` para agregar la ruta para el controlador anterior y agregue la siguiente línea al final del archivo.

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

Ejecute la siguiente URL en el navegador para ejecutar el ejemplo anterior.

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

A continuación, compruebe en su navegador. Gracias.

Lea [CodeIgniter - Internacionalización en línea](https://riptutorial.com/es/codeigniter/topic/9864/codeigniter---internacionalizacion):

<https://riptutorial.com/es/codeigniter/topic/9864/codeigniter---internacionalizacion>

Capítulo 8: CodeIgniter Carrito de compras

Introducción

Podemos utilizar la biblioteca de carritos de compras de CI cuando estamos construyendo un sitio de comercio electrónico. podemos configurar agregar al carrito, actualizar artículos del carrito, eliminar elementos del carrito e incluso borrar las funciones del carrito utilizando esta biblioteca.

Desde CodeIgniter Doc: La clase de carro permite agregar elementos a una sesión que permanece activa mientras un usuario navega por su sitio. Estos artículos se pueden recuperar y mostrar en un formato estándar de "carrito de compras", lo que permite al usuario actualizar la cantidad o eliminar artículos del carrito.

Examples

Agregar artículos en el carrito

Debe crear funciones en un controlador como insertar, actualizar, eliminar y borrar carro, etc. Por ejemplo: para insertar un nuevo artículo en el carrito, escriba debajo del código que acepte valor.

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

Y cree funciones en el modelo para tareas de carrito como insertar, actualizar, eliminar, borrar, etc.

Ej .: para insertar artículos en el carrito

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

El método insert () devolverá el \$ rowid si inserta con éxito un solo elemento. para que pueda verificar que el elemento se haya insertado o no y mostrar un mensaje relacionado al usuario.

y puede insertar varios elementos como este

```
$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);

```

Elementos básicos de la cesta de compras de CI

Como podemos agregar varios elementos en la matriz del Carro y luego agregarlo a la sesión del carro, pero hay 4 elementos básicos que la clase del Cart requiere para agregar datos con éxito en la sesión del carro.

1. id (cadena)
2. cantidad (número)
3. precio (número, decimal)
4. nombre (String)

Y si desea agregar más opciones con respecto al producto, puede usar el quinto elemento, que es "opciones". Puedes configurar un conjunto de opciones en este elemento.

Se verá así:

```

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

```

Mostrar artículos de carrito

Puede mostrar los artículos del carrito en bucle a través del carrito o puede mostrar un solo artículo desde el carrito.

```

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

```


Esto devolverá una matriz de elementos del carrito para que pueda recorrer esta matriz utilizando el bucle 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>";
}
```

Puede formatear estos datos como celda de tabla o algún div y luego mostrarlos en la vista.

Actualizar artículo de carrito

Rowid: el ID de fila es un identificador único generado por el código del carrito cuando se agrega un artículo al carrito. La razón por la que se crea una ID única es para que el carrito pueda gestionar productos idénticos con diferentes opciones.

Cada artículo en el carrito tiene un elemento rowid y por rowid puede actualizar el artículo del carrito.

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

y luego debajo del código

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

Eliminar artículos del carrito

Al utilizar el elemento rowid, puede eliminar un artículo del carrito. solo tienes que poner la cantidad del artículo a 0

```
$deleteItem = array(
    'rowid' => 'b99ccdf16028f015540f341130b6d8ec',
    'qty'   => 0
);

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

esto eliminará el elemento con este rowid.

Lea CodeIgniter Carrito de compras en línea:

<https://riptutorial.com/es/codeigniter/topic/9372/codeigniter-carrito-de-compras>

Capítulo 9: CodeIgniter URI Segment

Examples

Segmentos URI:

Por ejemplo, considere la siguiente URI:

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

El segmento permite recuperar una cadena URI de forma de segmento específica donde n es un número de segmento. Los segmentos están numerados de izquierda a derecha. Por ejemplo, el siguiente código:

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

Se utiliza para recuperar un segmento específico de la URI donde n es el número del segmento.

```
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
```

Obtener el último y último segmento de URI

Obtener el último segmento

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

Obtener antes del último segmento

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

Más información: [<http://stackoverflow.com/questions/9221164/code-igniter-get-before-last-uri-segment> ♦ []

Lea CodeIgniter URI Segment en línea:

<https://riptutorial.com/es/codeigniter/topic/5402/codeigniter-uri-segment>

Capítulo 10: Cómo configurar la zona horaria en CodeIgniter

Examples

Cómo configurar la zona horaria en CodeIgniter

Colocando `date_default_timezone_set('Asia/Kolkata');` en `config.php` encima de la base URL también funciona.

[Lista de PHP de zonas horarias soportadas](#)

aplicación / config.php

```
<?php

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

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

Otra forma que he encontrado útil es si desea establecer una zona horaria para cada usuario:

- Crea un archivo `MY_Controller.php` .
- Cree una columna en su tabla de `user` que puede nombrarle zona horaria o cualquier cosa que desee. De esa manera, cuando el usuario selecciona su zona horaria, puede configurarse en su zona horaria cuando inicia sesión.

aplicación / 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;
            }
        }
    }
}
```

```
}  
}
```

Además, para obtener la lista de zonas horarias en PHP:

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

Otra forma de configurar la zona horaria en codeigniter

Establecer la zona horaria en Codeigniter extendiendo la ayuda de la fecha es una forma alternativa. Para hacer eso necesitas seguir la siguiente actividad de dos pasos.

1. Extender la fecha ayudante con la siguiente función:

```
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. Ahora configure la zona horaria como un valor de `time_reference` de `config.php` como:

```
$config['time_reference'] = 'Asia/Dhaka';
```

Todo esto está configurado para usar la zona horaria.

Para su información: Lista de zonas horarias se agrega en el primer ejemplo.

[Lea Cómo configurar la zona horaria en CodeIgniter en línea:](#)

Capítulo 11: Cómo utilizar las bibliotecas y el ayudante de CI

Examples

Usando bibliotecas y ayudantes

El ejemplo es para fines ilustrativos del uso de bibliotecas y ayudantes y no un código válido. No lo copie / pegue en sus proyectos.

HELPER helpers / 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();
    }
}
```

BIBLIOTECA bibliotecas / 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);
        .....
    }
}

```

CONTROLADOR

```

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);

    }
}

```

Ayudante

Carga automática de su función auxiliar. Si usas mucho tiempo en tu proyecto.

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

Use el formulario de ayuda a la vista

```

<?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='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 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>";

```

?>

Lea Cómo utilizar las bibliotecas y el ayudante de CI en línea:

<https://riptutorial.com/es/codeigniter/topic/3776/como-utilizar-las-bibliotecas-y-el-ayudante-de-ci>

Capítulo 12: Creando cronjob en codeigniter en un servidor de alojamiento linux

Examples

Llamando a un controlador CodeIgniter desde 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 en 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";  
        }  
    }  
}  
?>
```

Llame esto desde su administrador de cpanel / cron de la siguiente manera (agregué más formas de llamarlo):

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

O

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

O

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

En mi caso: wget thing está trabajando en plesk y cpanel (wget creando archivos en el servidor en su directorio raíz). php-cli funciona tanto en plesk como en cpanel.

Lea [Creando cronjob en codeigniter en un servidor de alojamiento linux en línea](https://riptutorial.com/es/codeigniter/topic/4980/creando-cronjob-en-codeigniter-en-un-servidor-de-alojamiento-linux):

<https://riptutorial.com/es/codeigniter/topic/4980/creando-cronjob-en-codeigniter-en-un-servidor-de-alojamiento-linux>

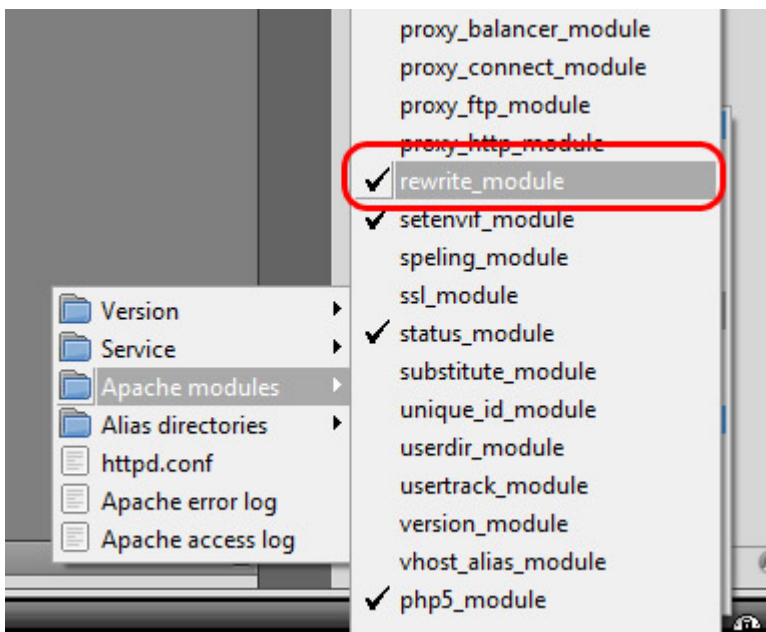
Capítulo 13: Eliminando index.php usando WAMP y CodeIgniter

Examples

Cómo eliminar el index.php de url con el uso de wamp y codeigniter

Lo primero que debe hacer es habilitar la reescritura de mod en wamp ir a los módulos de Apache y desplazarse hacia abajo en la lista

Si no se muestra, active la casilla y luego reinicie todos los servidores.



Los usuarios de Linux también pueden usar el siguiente comando de terminal para habilitar el módulo de reescritura

```
sudo a2enmod rewrite
```

Luego reinicie apache usando:

```
sudo service apache2 restart
```

Luego, al lado de su carpeta de aplicaciones, cree un archivo llamado .htaccess

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

Pruebe este código a continuación

```
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 no es aquí hay algunos [ejemplos](#) más htaccess

Luego vaya al archivo config.php. Configura tu base_url y deja en blanco el index_page

```
$config['base_url'] = ((isset($_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'] = '';
```

Espero que esto te ayude a usar los archivos htaccess de ejemplos para otros.

Lea [Eliminando index.php usando WAMP y CodeIgniter en línea:](#)

<https://riptutorial.com/es/codeigniter/topic/4340/eliminando-index-php-usando-wamp-y-codeigniter>

Capítulo 14: Empecemos: Hola Mundo.

Examples

Una aplicación muy simple de Hello World.

A partir de una nueva instalación de Codeigniter 3, aquí hay una manera simple de comenzar con una aplicación Hello World, para romper el hielo con este marco sólido de PHP.

Para ello, puede comenzar a crear la vista que queremos que se muestre para nuestra aplicación Hello World.

Vamos a ponerlo en tu carpeta de aplicación, aquí:

En `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>
```

Es solo un simple contenido HTML.

Ahora, para poder mostrar esta vista, necesitamos un **controlador** . El controlador es el que recordará la vista para que se muestre su contenido.

Para que funcione correctamente, el controlador debe ir a la carpeta de controladores adecuada.

Aquí es donde vamos a colocar nuestro controlador Hello World:

`/application/controllers/Hello_world.php`

(El nombre del controlador generalmente es *snake_case* con la primera letra en mayúscula)

```
<?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 función predeterminada para un controlador es la función de índice.

Ahora podrá ver el contenido de su página Hello World accediendo a la siguiente dirección:

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

o, en caso de que haya aplicado la solución utilizando .htaccess (vuelva a la página de instalación de la solución)

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

(Si está trabajando localmente, lo más probable es que la dirección donde encontrará su página sea: `http://localhost/hello_world`)

La URL realmente se forma llamando a su clase de controlador (en este caso `Hello_world` , pero usando todas las minúsculas en la URL). En este caso es suficiente, ya que utilizamos la función de índice. Si hubiéramos usado un nombre de función diferente (digamos `greetings`), deberíamos haber usado una URL como esta:

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

Que está estructurado como `/[controller_name]/[method_name]`

¡Aquí tienes! ¡Tu primera aplicación Codeigniter está funcionando!

Vamos a usar el controlador un poco más

Ahora vamos a probar un ejemplo un poco más complejo, utilizando las capacidades del controlador para completar la vista.

Aquí está nuestra opinión: `/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>
```

Ahora tenemos un marcador de posición para mostrar nuestros saludos.

Aquí es cómo cambiamos el controlador para que esto funcione:

```
<?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);
    }
}
```

La matriz `$data` se prepara con la información que se inyectará en la vista, utilizando la misma etiqueta (`greetings`) que se ha recuperado dentro de la vista.

El resultado final es el mismo que con el primer ejemplo, ¡pero ahora estamos usando más potencial del marco!

Vamos a elegir nuestros saludos: ¿Hola Mundo o Adiós Mundo o ...?

Digamos que queremos tener un saludo alternativo que sea accesible a través de una URL diferente. Podríamos crear una nueva función o incluso un nuevo controlador para eso, pero una buena práctica es optimizar lo que ya tenemos, para que funcione de la mejor manera.

Para hacer esto, mantendremos la misma vista que en los ejemplos anteriores, pero introduciremos un parámetro en nuestra función, para que pueda elegir entre dos saludos diferentes:

```
<?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);  
}  
}
```

Ahora tenemos múltiples opciones de saludos! Para poder visualizarlos, agregaremos el parámetro en la URL, de la siguiente manera:

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

Esto nos mostrará el mensaje: "Good Bye World".

La estructura de la URL es la siguiente:

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

En este caso, para volver a nuestro viejo "Hola Mundo", basta con llamar a la antigua URL, sin parámetros:

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

Puede agregar múltiples parámetros a su función (por ejemplo, si necesita 3 de ellos):

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

y se pueden llenar usando la url de la siguiente manera:

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

por ejemplo, `http://localhost/hello_world/greetings/goodbye/italian/red`

De esta manera, se le pueden pasar los parámetros directamente desde la URL que afectará el contenido de lo que se mostrará.

Para obtener más información acerca de cómo pasar parámetros a través de la URL, ¡es posible que desee ver el tema de enrutamiento!

Lea Empecemos: Hola Mundo. en línea:

<https://riptutorial.com/es/codeigniter/topic/2411/empecemos--hola-mundo->

Capítulo 15: Enviando email

Observaciones

En CodeIgniter 3 tienes que incluir el parámetro:

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

Simplemente no funcionará sin él.

Si no le importan las nuevas líneas y está utilizando CodeIgniter 2, este parámetro de configuración es opcional.

Examples

Cargar la biblioteca de correo electrónico

Primero necesitas cargar la biblioteca de correo electrónico.

Haga esto en el archivo del controlador que enviará el correo electrónico:

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

O cárguelo globalmente en el archivo autoload.php en la carpeta de configuración:

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

Mientras esté allí, es posible que desee cargar el asistente de correo electrónico si desea utilizar algunos de los accesos directos incorporados de CodeIgniter:

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

El asistente de correo electrónico se puede cargar en el archivo del Controlador de manera similar a la biblioteca de correo electrónico:

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

Establezca sus parámetros de configuración de correo electrónico

Cree un nuevo archivo en la carpeta application / config llamada email.php

Establecer los parámetros para el envío de correo electrónico. Estos se cargarán cuando envíe su correo electrónico.

```
$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
```

Crea tu correo electrónico

```
$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!');
```

En el método 'desde', el primer parámetro es la dirección de correo electrónico desde la que está enviando, el segundo parámetro es el nombre que desea que el receptor vea.

En el método 'a', usted define a quién se envía el correo electrónico.

El método 'asunto' define el asunto del correo electrónico.

El método de 'mensaje' define lo que estará en el cuerpo de su correo electrónico.

Cualquiera de estos podría ser un dato que un usuario envió a su sitio. Así que puede tener una variable aquí que contiene datos publicados. Para que se vean más como esto:

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

Envía tu e-mail

```
$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.";
}
```

Enviar un correo electrónico HTML

Pero no solo quieres un correo electrónico de texto plano. Quieres un bonito email en html.

Establece tu archivo de configuración como html:

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

Si desea pasar datos (como un nombre de usuario, por ejemplo) al correo electrónico html, colóquelos en una matriz:

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

Luego, al enviar, apunte su 'mensaje' a una vista. Luego pasa tu matriz de datos a ella:

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

En su carpeta de aplicación / vista cree su vista.

En este caso se llama 'new_user.php'.

Puedes peinar esto de cualquier manera que quieras. Aquí hay un ejemplo rápido:

```
<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>
```

Formulario de contacto

Controlador (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');
    }
}

```

Vistas (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>

```

```
<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>
```

Lea Enviando email en línea: <https://riptutorial.com/es/codeigniter/topic/5403/enviando-email>

Capítulo 16: Estructura de consulta

Examples

Seleccionando datos

Las siguientes funciones le permiten crear instrucciones SQL SELECT.

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

Esto ejecuta la consulta de selección y devuelve el resultado. Se puede usar solo para recuperar todos los registros de una tabla:

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

Los parámetros segundo y tercero le permiten establecer una cláusula de límite y compensación:

```
$query = $this->db->get('tablename', 10, 20);  
  
// Executes: SELECT * FROM tablename LIMIT 20, 10  
// (in MySQL. Other databases have slightly different syntax)
```

Seleccionando datos

Seleccionando datos con condicion

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

Selección de datos con múltiples condiciones.

```
$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();
```

Seleccionar datos con condición y límite.

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

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

Seleccionar datos con condición, filas máximas y orden descendente

```
$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();
```

Selección de datos con el segundo parámetro opcional

Normalmente no estamos usando el segundo parámetro en `select([$select = '*', $escape = NULL])` en CodeIgniter. Si lo configura en `FALSE`, CodeIgniter no intentará proteger sus nombres de campo o tabla.

En el siguiente ejemplo, vamos a seleccionar el campo de tipo de fecha y hora mediante el formateo utilizando la consulta sql y lo estableceremos en `FALSE` (al hacer esto, le diremos a CI que no escape de la consulta automáticamente).

```
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 no lo configuramos en `FALSE`, automáticamente se escapará y romperá la consulta.

Unir tablas usando el generador de consultas

A veces necesitamos unir varias tablas para obtener datos agregados a cambio. aquí es cómo podemos lograr lo mismo utilizando 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();
}
```

Aquí usamos `join()` para unir varias tablas y podemos cambiar el tipo de unión en el tercer parámetro como "interior", "izquierda", "derecha", etc.

Lea Estructura de consulta en línea: <https://riptutorial.com/es/codeigniter/topic/3769/estructura-de-consulta>

Capítulo 17: Hacer API en Codeigniter

Introducción

Codeigniter proporciona una clase de salida autoinicializada que es muy útil para crear API y diferentes tipos de salida de documentos como .pdf, .csv, .image, etc.

NOTA: - El tipo de documento predeterminado de Codeigniter es HTML; cámbielo a application / json; la API debe ser del tipo json requerido

Examples

crear el nuevo controlador con nombre 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'
        );
    }
}
?>
```

Recupere algunos datos de la API agregue la siguiente función en el controlador API

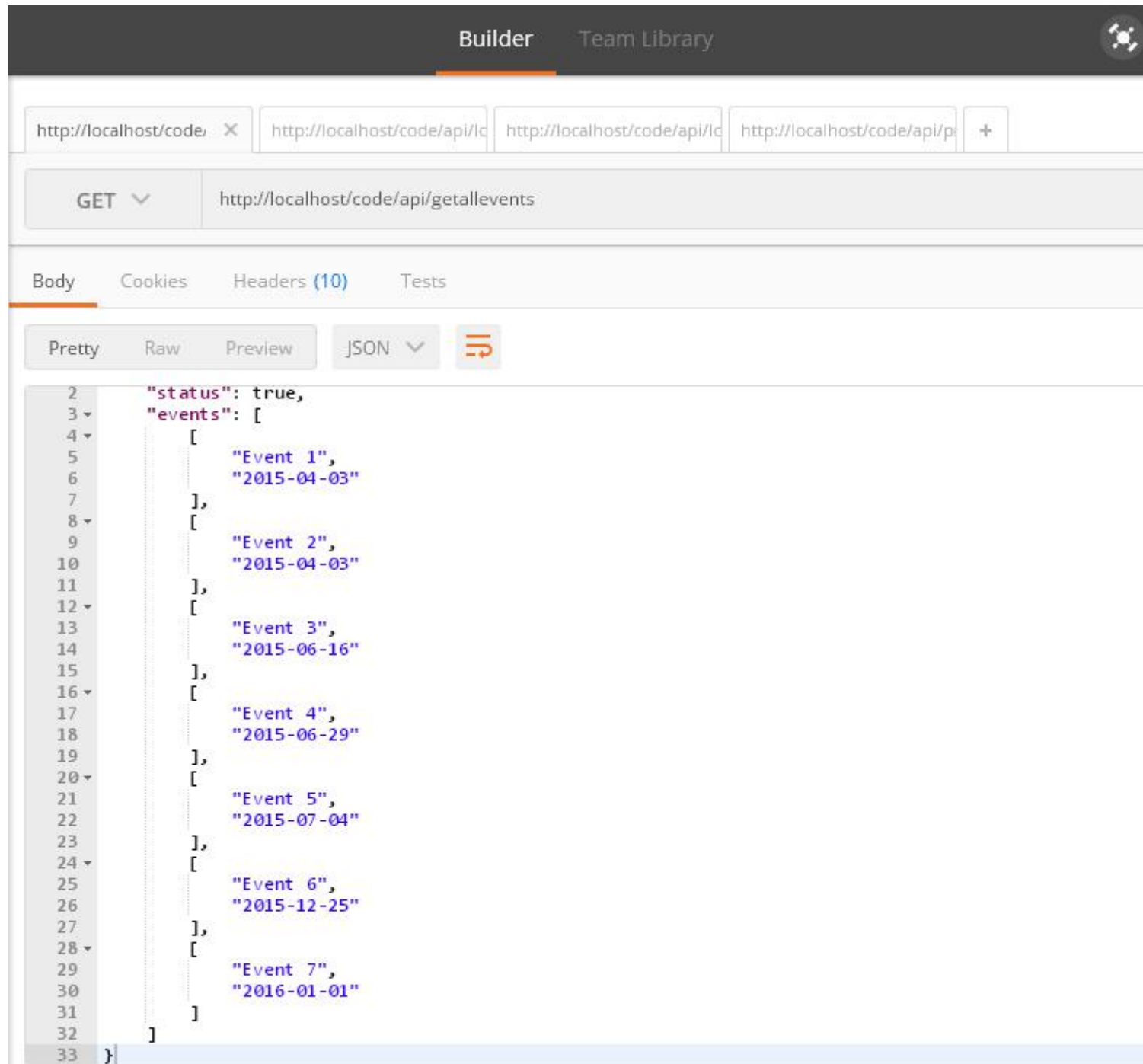
```
/******
@return all events
*****/
public function getallevents(){
    //get data from model
    $events = array(
        array('Event 1', '2015-04-03'),
        array('Event 2', '2015-04-03'),
        array('Event 3', '2015-06-16'),
```

```

    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)));
}

```

Vista del cartero



The screenshot shows a web browser interface for a REST client. The top bar has 'Builder' and 'Team Library' tabs. Below the address bar, the URL is `http://localhost/code/api/getallevents`. The 'GET' method is selected. The response body is displayed in JSON format, showing a status of `true` and a list of 7 events. The events are:

- Event 1, 2015-04-03
- Event 2, 2015-04-03
- Event 3, 2015-06-16
- Event 4, 2015-06-29
- Event 5, 2015-07-04
- Event 6, 2015-12-25
- Event 7, 2016-01-01

inicie sesión en la API de usuario para permitir el acceso de algunos datos privados para un usuario particular

```

/*****
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')));
    }
}

```

http://localhost/code/welco http://localhost/code/ X http://localhost/code/api/lc http://localhost/code/api/p +

POST ▼ http://localhost/code/api/login

Authorization Headers **Body** ● Pre-request Script Tests

☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary

	Key	Value	Description
<input checked="" type="checkbox"/>	username	admin	
<input checked="" type="checkbox"/>	password	test@123	
	New key	Value	Description

Body Cookies Headers (10) Tests

Pretty Raw Preview JSON ▼

```

1 {
2   "status": true,
3   "msg": "log in successfully",
4   "data": {
5     "username": "admin",
6     "email": "domain@test.com",
7     "logged_in": true
8   }
9 }

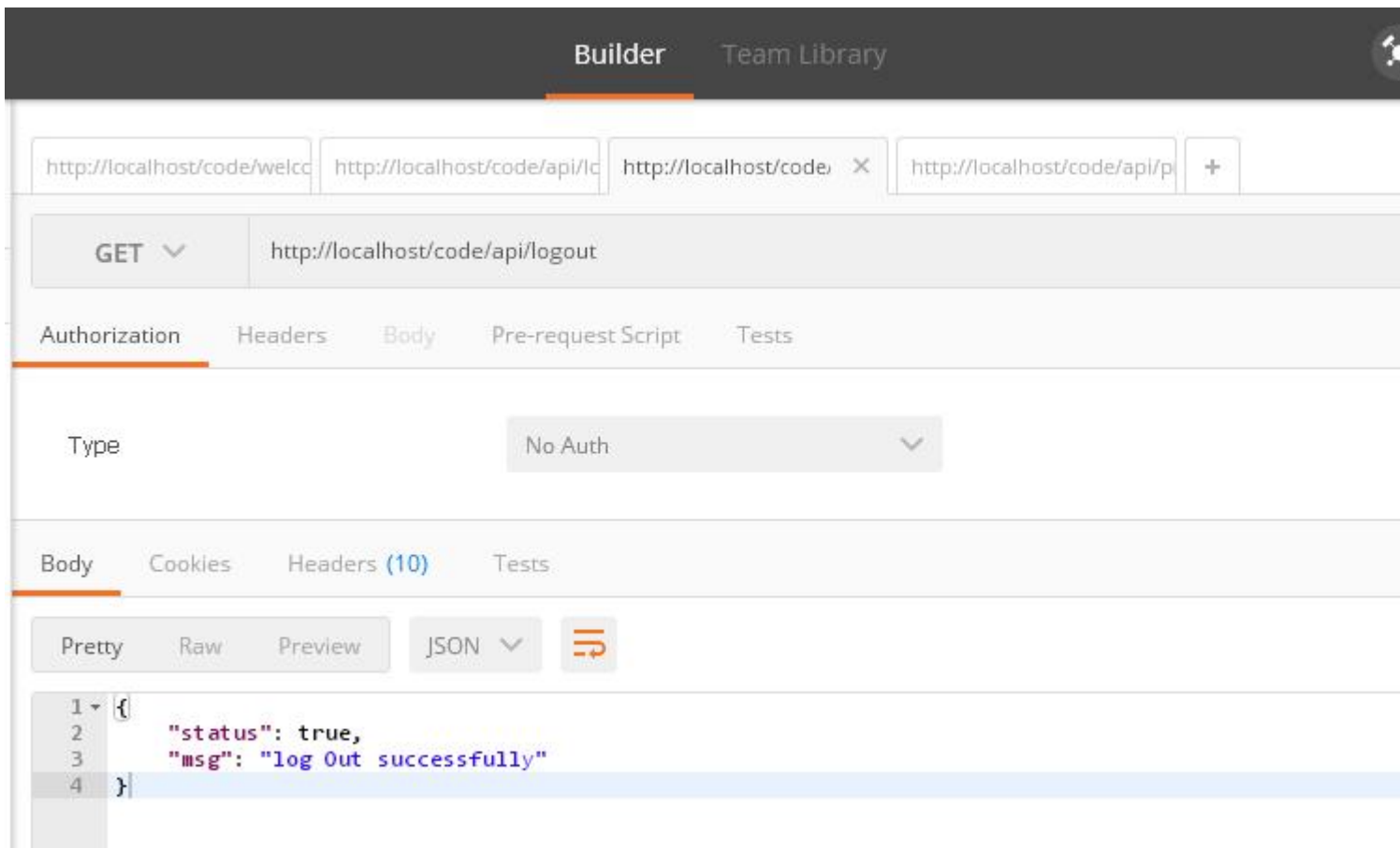
```

el usuario cierra sesión en la API para destruir la sesión del usuario que ha iniciado sesión

```

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

```



crear api protegido

Esta API no es accesible para el usuario público, se requiere autenticación

```
/*  
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')));  
    }  
}
```

Builder Team Library


http://localhost/code/welcc http://localhost/code/api/lc http://localhost/code/api/lc http://localhost/code/ X +

GET ▼ http://localhost/code/api/protectedapi

Authorization Headers Body Pre-request Script Tests

Type No Auth ▼

Body Cookies Headers (10) Tests

Pretty Raw Preview JSON ▼ 

```
1 {  
2   "status": true,  
3   "msg": "Access allowed"  
4 }
```

Lea Hacer API en Codeigniter en línea: <https://riptutorial.com/es/codeigniter/topic/10903/hacer-api-en-codeigniter>

Capítulo 18: Image / File Uploader en Codelgniter

Observaciones

No es necesario que tenga que usar los mismos nombres para (Controlador, Archivo, Clase, ID) o lo que sea. Todo lo que he usado es para comprender el propósito del flujo de codificación y mis suposiciones. Depende del desarrollador que tome el código y edite el código / nombre de acuerdo con su deseo y luego aloje el código y tenga éxito.

Examples

Un solo archivo / cargador de imágenes

Ahora veremos cómo funciona el código de carga de imagen / archivo en el método de CI nativo con la ayuda de los formularios que ha propuesto la forma de CI.

Carga de archivos en PHP tiene dos escenarios. Se menciona a continuación de la siguiente manera.

- Imagen única / cargador de archivos: se puede guardar con la ayuda de la variable normal en el atributo de formulario. (Por ejemplo) `<input type="file" name="image" />`
- Imagen múltiple / Cargador de archivos: esto se puede guardar solo con la ayuda de la variable de matriz para el nombre en el tipo de archivo. (Por ejemplo) `<input type="file" name="image[]" />`.

La variable de matriz a saber `name="profile[]"` puede también ser mantenido para la `single image` de subida, así como la `multi-image` **cargador** también.

Por lo tanto, el código de la Imagen única / Cargador de archivos en el formato Native Codelgnitor es el siguiente:

Ver parte:

```
<?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();
?>
```

Por lo tanto si presentamos el formulario iremos a la

- Employee - Controlador y búsqueda de la función llamada `addemployee`
- Si necesita el atributo requerido para el código del cargador de archivos, puede agregar el atributo HTML5 llamado `required` a la etiqueta de entrada.

A continuación se muestran los dos ejemplos de cómo usar el atributo requerido, pero ambos métodos también son los mismos.

1. Método uno: `<input type="file" name="photo" required="required" />`
2. Método dos: `<input type="file" name="photo" required />`

Por lo tanto, estos son algunos de los consejos importantes que se deben seguir en la parte de la vista del cargador de imágenes / archivos.

Parte del controlador:

```
<?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
    }

    class
        $this->load->helper('form'); //This will load up all the form attributes that are need by
        the form.
    }

    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
    }
}
?>

```

Nota: de forma predeterminada, la rutina de carga espera que el archivo provenga de un campo de formulario llamado `userfile`, y el `form` debe ser de tipo `multipart`.

- Por lo tanto, irá a `employee_model` con la matriz `$data_value` - array y guardará los datos bajo la función llamada `saveemployee`.
- Si desea establecer su propio nombre de campo, simplemente pase su valor al método `do_upload()`
- Usando la clase de carga de archivos, podemos cargar archivos y también podemos restringir el tipo y tamaño del archivo que se va a cargar.
- `display_errors()` : recupera cualquier mensaje de error si el método `do_upload()` devuelve falso. El método no se hace eco automáticamente, devuelve los datos para que pueda asignarlos como necesite.

Notaciones:

Estas son las notaciones que están disponibles en el CI y podemos definirlo en el `index.php` como una definición corta y podemos usarlo en el proyecto completo.

```

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

```

Parte del modelo:

```

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

```

- Se guardarán los datos en la tabla de `employee` con el nombre de la imagen cargada.
- Y la imagen cargada se guardará en el directorio que hemos creado en la carpeta raíz o en cualquier otra carpeta que especifiquemos.

Lea Image / File Uploader en CodeIgniter en línea:

<https://riptutorial.com/es/codeigniter/topic/7450/image---file-uploader-en-codeigniter>

Capítulo 19: Juega con la palabra inglesa con el ayudante INFLECTOR

Introducción

Inflector es un ayudante muy útil para cambiar / convertir palabras inglesas a singular, plural, funda de camello, humanizar, etc. El ayudante también ayuda a verificar si una palabra tiene versión plural o no.

Examples

Auxiliar de carga del inflector

Para usar el método de ayuda de inflector, primero cargue el ayudante como todos los demás ayudantes con el siguiente código:

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

Hacer una palabra singular

Función `singular($string)` , convierte una palabra plural a singular. Para obtener el parámetro de resultado perfecto, `$string` debe ser una sola palabra. La función devolverá la `string` .

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

Comprobar que una palabra tiene plural

`is_countable($string)` se usa para verificar si una palabra tiene forma plural o no. El tipo devuelto será `boolean` significa que si la palabra dada tiene forma plural devolverá `true` , de lo contrario devolverá `false` .

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

Hacer una palabra plural

Para obtener la forma plural de cualquier palabra en inglés, la función `plural($string)` es útil. Al igual que el `singular($string)` , la función `plural($string)` también devuelve el resultado del `string` .

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

Camelizado la cuerda

Camel Case es la práctica de escribir palabras o frases compuestas donde cada palabra comienza con mayúscula, sin espacio entre palabras. La función `camelize($string)` ayuda a hacer una cadena camelizada. Convierte una cadena de palabras separadas por espacios o guiones bajos a camel case.

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

Eliminar / Añadir delimitador entre palabras

Eliminar delimitador

La función `humanize($words)` , toma varias palabras separadas por guiones bajos y agrega espacios para los guiones bajos con mayúsculas en cada palabra.

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

La función también puede reemplazar cualquier separador / delimitador declarado. En este caso, el delimitador será el segundo parámetro.

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

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

Añadir subrayado

Por otro lado, la función de `underscore($words)` reemplaza el espacio entre las palabras con subrayado (`_`).

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

Lea Juega con la palabra inglesa con el ayudante INFLECTOR en línea:

<https://riptutorial.com/es/codeigniter/topic/8057/juega-con-la-palabra-inglesa-con-el-ayudante-inflector>

Capítulo 20: Llamar a un método modelo en una vista

Introducción

A veces es más útil hacer una llamada al método de un modelo desde nuestro punto de vista, por lo que esta es una manera de hacerlo

Examples

Guardar una llamada de método en una variable

En el controlador:

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

En vista:

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

Lea Llamar a un método modelo en una vista en línea:

<https://riptutorial.com/es/codeigniter/topic/8163/llamar-a-un-metodo-modelo-en-una-vista>

Capítulo 21: Manejo de errores

Introducción

CodeIgniter le permite crear informes de errores en sus aplicaciones utilizando las funciones que se describen a continuación. Además, tiene una clase de registro de errores que permite que los mensajes de error y depuración se guarden como archivos de texto.

Examples

`show_error ()`

Esta función mostrará el mensaje de error que se le proporcionó utilizando la siguiente plantilla de error:

Ruta - `application/errors/error_general.php`

El parámetro opcional `$ status_code` determina qué código de estado HTTP debe enviarse con el error.

Sintaxis

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

Parámetros:

- `$message` (mixed) - Mensaje de error
- `$status_code` (int) - Código de estado de respuesta HTTP
- `$heading` (string) - encabezado de página de error

Tipo de devolución: void

Fuente

1. [show_error](https://codeigniter.com/user_guide/errors/show_error.html) en codeigniter.com

`show_404 ()`

Esta función mostrará el mensaje de error 404 que se le proporcionó usando la siguiente plantilla de error:

Ruta - `application/errors/error_404.php`

La función espera que la cadena que se le pase sea la ruta del archivo a la página que no se encuentra. Tenga en cuenta que CodeIgniter muestra automáticamente los mensajes 404 si no se encuentran los controladores.

CodeIgniter registra automáticamente cualquier `show_404()`. Al establecer el segundo parámetro opcional en `FALSE` se omitirá el registro.

Sintaxis

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

Parámetros:

- `$page` (cadena) - cadena URI
- `$log_error` (bool) - Si registrar el error

Tipo de devolución: void

Fuente

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

`log_message()`

Esta función le permite escribir mensajes en sus archivos de registro. Debe proporcionar uno de los tres "niveles" en el primer parámetro, indicando qué tipo de mensaje es (depuración, error, información), con el mensaje en sí en el segundo parámetro.

Ejemplo:

```
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.');
```

Sintaxis

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

Parámetros:

- `$level` (string) - Nivel de registro: 'error', 'depuración' o 'información'
- `$message` (string) - Mensaje para registrar

Tipo de devolución: void

Hay tres tipos de mensajes:

- **Mensajes de error** . Estos son errores reales, como errores de PHP o errores de usuario.
- **Mensajes de depuración**. Estos son mensajes que ayudan en la depuración. Por ejemplo, si una clase se ha inicializado, podría registrar esto como información de depuración.
- **Mensajes informativos** . Estos son los mensajes de prioridad más baja, simplemente brindando información sobre algún proceso. CodeIgniter no genera ningún mensaje de información de forma nativa, pero es posible que desee hacerlo en su aplicación.

Nota: Para que el archivo de registro se escriba, los "registros" de la carpeta deben poder escribirse. Además, debe establecer el "umbral" para iniciar sesión en `application/config/config.php` . Es posible que, por ejemplo, solo desee que se registren los mensajes de error y no los otros dos tipos. Si lo configura a cero, el registro será deshabilitado.

Lea Manejo de errores en línea: <https://riptutorial.com/es/codeigniter/topic/8699/manejo-de-errores>

Capítulo 22: Paginación del codificador

Examples

En esta sección asumo que usted conoce al ayudante que llama, en el controlador.

```
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');
```

en la vista

paginación-> create_links (); ?>

Lea Paginación del codificador en línea:

<https://riptutorial.com/es/codeigniter/topic/9393/paginacion-del-codificador>

Capítulo 23: Resolución de problemas de Codeigniter

Introducción

Para la depuración y la solución de problemas en Codeigniter, puede usar **Profiler** , parte de la biblioteca de resultados

Examples

Solución de problemas

Si descubre que no importa lo que ponga en su URL, solo se está cargando su página predeterminada, es posible que su servidor no admita la variable `PATH_INFO` necesaria para servir URLs compatibles con los motores de búsqueda.

Como primer paso, abra el archivo `application/config/config.php` y busque la información del `URI Protocol` . Le recomendará que intente un par de configuraciones alternativas.

Si aún no funciona después de haber intentado esto, deberá obligar a Codeigniter a agregar un signo de interrogación a sus URLs . Para hacer esto, abra su `application/config/config.php` y cambie esto:

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

A esto:

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

Lea Resolución de problemas de Codeigniter en línea:

<https://riptutorial.com/es/codeigniter/topic/7901/resolucion-de-problemas-de-codeigniter>

Capítulo 24: sesión setflashdata

Examples

Cómo configurar los datos flash de sesión en el controlador

Puede configurar los datos flash en el controlador simplemente utilizando esta sintaxis

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

Aquí 'mensaje' es el identificador de los datos de acceso a la vista. Puede establecer más de un mensaje simplemente cambiando el identificador.

por ejemplo

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

Cómo mostrar Flashdata en la vista

Simplemente puede acceder a los fashdata en vista como esta

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

Para acceder a múltiples mensajes solo cambia identificador

Por ej.

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

Lea sesión setflashdata en línea: <https://riptutorial.com/es/codeigniter/topic/9688/sesion-set-flashdata>

Capítulo 25: sufijo de url

Examples

sufijo de url

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

cambia todo lo que quieras como html o asp, esto funcionará después de tu rwmoving index.php en config.php

Lea sufijo de url en línea: <https://riptutorial.com/es/codeigniter/topic/9379/sufijo-de-url>

Capítulo 26: URL base en Codeigniter

Examples

Configuración de su url base en Codeigniter

Necesitará establecer su URL base en `application/config/config.php`

Si no está configurado, Codeigniter intentará adivinar el protocolo y la ruta de acceso a su instalación, pero por razones de seguridad, el nombre de host se configurará en `$_SERVER['SERVER_ADDR']` si está disponible, o localhost de lo contrario. El mecanismo de autodetección existe solo por conveniencia durante el desarrollo y NO DEBE utilizarse en la producción.

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

Debe ser archivado como

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

Siempre es bueno usar `/` al final de `base_url`

Cuando no configura su URL base, puede encontrarse con algunos errores en los que no puede cargar su CSS, imágenes y otros elementos activos. Y también es posible que tenga problemas para enviar formularios, ya que algunos usuarios se han dado cuenta.

Actualizar

Si no desea configurar su URL base de otra manera es.

Cree un nuevo archivo de núcleo en `application/core/MY_Config.php`

Y pega este código

```
<?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);
    }
}

```

Algo más sobre base_url

¿Qué pasa si no configuro `base_url` ?

No obtendrá ningún error de Impotencia para configurar esto y continuar. Puede continuar sin configurar, pero debe saber acerca de [la inyección del encabezado HTTP](#)

Si no lo puse ¿qué aparecerá?

Obtendrá `http://[::1]/` lugar de su URL real.

¿Qué significa esto `http://[::1]/` ??

Esta es la URL temporal que establece por CI por defecto. Esto apuntará a la raíz de su documento.

`::1` - Dirección del servidor (localhost) [Lea más sobre esto](#)

¿Cómo establecer una `base_url()` adecuada `base_url()` ?

La URL base siempre debe apuntar a la raíz de la carpeta de su proyecto. (fuera de la carpeta de la aplicación)

```
$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)
```

¿Cómo usar `base_url()` ?

El uso más común es encontrar la ruta correcta a sus archivos js o css.

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

Agregar el código de arriba en tu vista producirá HTML como se muestra a continuación:

```
<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>
```

Campo de golf

1. Ayudante de URL

Forma inteligente de configurar el `base_url`

Las siguientes líneas de código son una forma más inteligente de configurar `base_url` en 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']);
```

Recomendado es

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

Porque todo el mundo conoce el espacio de alojamiento. Entonces, si configura de esta manera, ***puede evitar la inyección en su sitio / host***.

Lea URL base en Codeigniter en línea: <https://riptutorial.com/es/codeigniter/topic/3643/url-base-en-codeigniter>

Capítulo 27: Usando el modelo en codeigniter

Examples

Creando Modelo

Ir a la `application/model`

Nombre de archivo - **Home_model.php**

Dentro del archivo

```
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();
    }
}
```

Y cuando necesites cargar este modelo:

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

O si desea que su modelo se asigne a un nombre de objeto diferente, puede especificarlo de esta manera:

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

Cargando modelo

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

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

Si desea que su modelo se asigne a un nombre de objeto diferente, puede especificarlo a través del segundo parámetro del método de carga:

Sintaxis -

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

```
$this->foobar->method();
```

Practica -

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

Función de modelo de llamada

Sintaxis

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

Práctica

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

Pasando datos al modelo

Sintaxis

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

Práctica

```
$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);
```

Recibiendo datos del controlador

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

Cuidado con el orden que pasa del controlador al modelo.

Devolver datos al controlador

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

esto devolverá el resultado con el ID y el nombre de usuario coincidentes al controlador.

Lea Usando el modelo en codeigniter en línea:

<https://riptutorial.com/es/codeigniter/topic/3777/usando-el-modelo-en-codeigniter>

Capítulo 28: Usando Sesiones

Observaciones

La clase **Sesiones** de Codeigniter utiliza cookies del navegador para guardar datos que persistirán en las cargas de varias páginas.

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

Examples

Crear una sesión

Para inicializar una sesión, simplemente puede cargarla en su controlador, esto usualmente se coloca dentro de las construcciones del controlador, pero también se puede cargar automáticamente en la matriz que se encuentra dentro de application / config / autoload.php:

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

Manejo de datos de sesión

Una sesión es simplemente una matriz que consta de la siguiente información del usuario:

1. El ID de sesión único del usuario (esta es una cadena estadísticamente aleatoria con una entropía muy fuerte, con hash con MD5 para portabilidad y regenerada (por defecto) cada cinco minutos)
2. La dirección IP del usuario
3. Los datos del agente de usuario del usuario (los primeros 120 caracteres de la cadena de datos del navegador)
4. La "última actividad" marca de tiempo.

Fuente ([lo que es información de sesión](#))

Para recuperar datos de sesión

como el SessionID:

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

Nota: para Codeigniter 3.x, puede usar la sintaxis anterior, pero se introdujo el concepto o los captadores mágicos, donde puede usar `$this->session->session_id`.

Recuerde que el `userdata()` devuelve NULL si el elemento de sesión no existe.

Para recuperar todos los datos de la sesión

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

Para establecer datos de sesión

El método `set_userdata()` permite configurar datos en su sesión, el siguiente ejemplo muestra una matriz de ejemplo que desea insertar:

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

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

También puede configurar un dato a la vez, por ejemplo:

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

O

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

Para eliminar sesión y datos de sesión

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

Este método también acepta una matriz de claves de elementos para desarmar:

Para Codeigniter 3.x:

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

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

Para Codeigniter 2.x (esta sintaxis heredada no admite 3.x):

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

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

Lea Usando Sesiones en línea: <https://riptutorial.com/es/codeigniter/topic/5793/usando-sesiones>

Capítulo 29: Uso de ganchos

Examples

Habilitar ganchos

La función de ganchos se puede habilitar / deshabilitar globalmente configurando el siguiente elemento en el archivo `application/config/config.php` :

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

Definiendo un gancho

Los ganchos se definen en el archivo `application/config/hooks.php` . Cada gancho se especifica como una matriz con este prototipo.

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

El índice de matriz se correlaciona con el nombre del punto de enlace particular que desea utilizar. En el ejemplo anterior, el punto de `pre_controller` es `pre_controller` . Una lista de puntos de gancho se encuentra a continuación. Los siguientes elementos deben definirse en su matriz de gancho asociativo:

clase El nombre de la clase que desea invocar. Si prefiere utilizar una función de procedimiento en lugar de una clase, deje este elemento en blanco.

función El nombre de la función (o método) al que desea llamar.

nombre de archivo El nombre del archivo que contiene su clase / función.

ruta-archivo El nombre del directorio que contiene su script.

params Cualquier parámetro que desee pasar a su script. Este artículo es opcional.

Puntos de gancho

`pre_system`

Llamado muy temprano durante la ejecución del sistema. En este punto, solo se han cargado la clase de referencia y los ganchos. Ningún enrutamiento u otros procesos han ocurrido.

`pre_controller`

Llamado inmediatamente antes de que se llame a cualquiera de sus controladores. Se han realizado todas las clases de base, enrutamiento y controles de seguridad.

`post_controller_constructor`

Llamado inmediatamente después de que su controlador sea instanciado, pero antes de que ocurra cualquier llamada de método.

`post_controller`

Llamado inmediatamente después de que su controlador esté completamente ejecutado.

`display_override`

`_display()` método `_display()` , utilizado para enviar la página finalizada al navegador web al final de la ejecución del sistema. Esto le permite utilizar su propia metodología de visualización. Tenga en cuenta que necesitará hacer referencia al superobjeto de CI con `$this->CI =& get_instance()` y luego los datos finalizados estarán disponibles llamando a `$this->CI->output->get_output()` .

`cache_override`

Le permite llamar a su propio método en lugar del método `_display_cache()` en la Biblioteca de resultados. Esto le permite utilizar su propio mecanismo de visualización de caché.

`post_system`

Se llama después de que la página procesada final se envía al navegador, al final de la ejecución del sistema después de que los datos finalizados se envían al navegador.

Ejemplo de Pre Controller Hook usando CodeIgniter

En la carpeta `application/hooks` , cree un archivo con el nombre `Blocker.php` y pegue el siguiente código.

```
<?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;
        }
    }
}
```

```
}  
?>
```

En `application/config/hooks.php` , declare el siguiente `application/config/hooks.php` .

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

En `application/config/config.php` , establezca el siguiente valor como verdadero

Definiendo un gancho

Los ganchos se definen en la `application/config/hooks.php` . Cada gancho se especifica como una matriz con este prototipo:

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

- **CLASS** : la clase que desea invocar si se trata de un código de procedimiento, déjelo en blanco.
- **FUNCTION** - El nombre de la función que desea llamar.
- **FILENAME DE ARCHIVO**: el nombre de archivo que contiene su clase / función.
- **FILEPATH** - Ubicación del archivo de `FILEPATH` .
- **PARAMS** Parámetro `PARAMS` si es necesario es opcional.

Lea **Uso de ganchos en línea**: <https://riptutorial.com/es/codeigniter/topic/3953/uso-de-ganchos>

Capítulo 30: Validación de formularios

Examples

Validar ejemplo de formulario

```
// 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
}
```

[Enlazar](#)

Lea Validación de formularios en línea: <https://riptutorial.com/es/codeigniter/topic/7398/validacion-de-formularios>

Creditos

S. No	Capítulos	Contributors
1	Primeros pasos con codeigniter	Abdulla Nilam , Ariful Islam , BIBIN JOHN , Bilal Ahmad , cfnerd , Community , emstawicki , gabe3886 , karel , Mitul , Prakash , Shiva127 , zur4ik
2	¿Cómo usar las bibliotecas y ayudantes de CI?	Abdulla Nilam , karel , Yaseen Ahmad
3	Asegurando su aplicación web	ankit suthar , BIBIN JOHN , ImBS , moopet
4	Autenticación	Ahmet Can Boyraz
5	Ayudante de matriz	Abdulla Nilam , Rana Ghosh
6	CAPTCHA ayudante	Abdulla Nilam , Rana Ghosh
7	CodeIgniter - Internacionalización	ImBS
8	CodeIgniter Carrito de compras	Abdulla Nilam , Hemant Sankhla , Muhamad Riyan
9	CodeIgniter URI Segment	Adrian P. , Md. Khairul Hasan
10	Cómo configurar la zona horaria en CodeIgniter	Abdulla Nilam , Ariful Islam , Nadim Sheikh , RamenChef , rap-2-h , wolfgang1983
11	Cómo utilizar las bibliotecas y el ayudante de CI	Abdulla Nilam , Adrian P. , karel , Mitul , NAW_AB , shantanu
12	Creando cronjob en codeigniter en un servidor de alojamiento linux	Abdulla Nilam , kishor10d , Tim Duncklee
13	Eliminando index.php usando WAMP y CodeIgniter	Ali , Ariful Islam , Gaurav , wolfgang1983

14	Empecemos: Hola Mundo.	Abdulla Nilam , clami219 , David , KuroKo3 , Shiva127 , Vishal
15	Enviando email	gabe3886 , Lucifer MorningStar , Miles , Rahamathullah MK
16	Estructura de consulta	abdurRahaman , anju , DogeAmazed , Fabio Widmer , Hemant Sankhla , kishor10d , MAZux , Rahmat
17	Hacer API en Codeigniter	Kundan Prasad
18	Image / File Uploader en Codeigniter	Andrey , Hanthony Tagam , Naresh Kumar .P
19	Juega con la palabra inglesa con el ayudante INFLECTOR	Ariful Islam
20	Llamar a un método modelo en una vista	Saul
21	Manejo de errores	Abdulla Nilam , Rana Ghosh
22	Paginación del codificador	Muhamad Riyan
23	Resolución de problemas de Codeigniter	Abdulla Nilam , ankit suthar , Mahdi Majidzadeh , Rana Ghosh
24	sesión set flashdata	Gopal Bhuva
25	sufijo de url	Muhamad Riyan
26	URL base en Codeigniter	Abdulla Nilam , Adrian P. , Ariful Islam , Blubberguy22 , sudopower , wolfgang1983
27	Usando el modelo en codeigniter	Abdulla Nilam , anju , gabe3886 , Lucifer MorningStar , MAZux , Russ_AB
28	Usando Sesiones	Abdulla Nilam , ElmerCat , Prakash
29	Uso de ganchos	Abdulla Nilam , ankit suthar , kishor10d , liamja , Mitul , Rafiquel Islam
30	Validación de formularios	Abdulla Nilam , ankit suthar , Lucifer MorningStar , Mitul , Murilo