



# CodeIgniter Framework Basic Tutorial

---

*CodeIgniter is open souce PHP Framework.*

This article is abstracted from the link(<http://www.phpeveryday.com/articles/CodeIgniter-Framework-Basic-Tutorial-2841.html>) "PHPEveryDay" and summarized by "LwinMyo Thu".

# ***CodeIgniter Framework Basic Tutorial***

CodeIgniter is open source PHP Framework.

## **What is CodeIgniter?**

- Cool toolkit for building web.
- Simple and elegant framework to create web applications.
- Good documented PHP framework

## **Why CodeIgniter?**

- Cool performance
- Small footprint
- Almost zero configuration framework
- Framework with good and clear documentation

## **Where You can get it?**

- You can download at <http://codeigniter.com/>

## **Whenever you need CodeIgniter?**

- You need exceptional performance.
- You need broad compatibility with standard hosting accounts that run a variety of PHP versions and configurations.
- You do not want to be forced to learn a templating language (although a template parser is optionally available if you desire one)
- You eschew complexity, favoring simple solutions

## **Learn CodeIgniter Framework Basic**

- (i) Introduction to CodeIgniter Framework
- (ii) CodeIgniter: Installation
- (iii) Creating First Application at CodeIgniter
- (iv) Creating and Sending Parameters Between Controller and View
- (v) Getting Parameters From GET
- (vi) Setting Database Configuration
- (vii) Preparing Database
- (viii) Showing Simple All Data
- (xi) Showing Simple One Data
- (x) Choosing a Data From GET

## **Learn CodeIgniter Framework Form**

1. Creating Skeleton
2. File Structure

3. Creating Layout Code
4. Creating Menu Library
5. Putting Text at Header and Footer
6. Centralizing \$data
7. Adding CSS
8. Creating Form HTML
9. Creating Beautiful Form Code
10. CodeIgniter HTML Style
11. Preparing Table at Database
12. Creating Insert Data
13. Creating List Data Use Table Library
14. Creating Table List Without Table Library
15. Using URL Helper
16. Showing Single Data for Form Edit
17. Something Happen With Form Parameters
18. Updating Data
19. Deleting Data

## Learn CodeIgniter Framework Basic

### (i) CodeIgniter: Introduction to CodeIgniter Framework

CodeIgniter Step By Step Tutorial: CodeIgniter (CI) is one of popular php framework. If you are already building PHP Application, CodeIgniter will help you to do it better and more easily. With CodeIgniter, you can save time, make your web more robust, your code will be easier to read and maintenance. It is free, lightweight, and simple to install. In this post, we will know more deep about CodeIgniter before write code.

Nice, CodeIgniter is small and lightweight framework. The zipped download for version 1.5 is only 737 KB. You don't need long time to download it. You can get it from <http://www.codeigniter.com>. CI was written by Rick Ellis, rock musician turned programmer.

With CodeIgniter, you can cut down the amount of code you need to type. This is not just good for lazy, but: less type, fewer mistake, and less time for spend debugging.

But, CodeIgniter is not everything. We will not find 'engine generator' that can build page self. Several frameworks have features like that. For example, they can create web page (that to do basic Create, Read, Update, and Delete operation) automatically. CodeIgniter doesn't do this.

This, I copy from their help page: "Our goal for CodeIgniter is **maximum performance, capability, and flexibility in the smallest, lightest possible package.**

...

From an technical and architectural standpoint, CodeIgniter was created with the following objectives:

- **Dynamic Instantiation.** In CodeIgniter, components are loaded and routines executed only when requested, rather than globally. No assumptions are made by the system regarding what may be

needed beyond the minimal core resources, so the system is very light-weight by default. The events, as triggered by the HTTP request, and the controllers and views you design will determine what is invoked.

- **Loose Coupling.** Coupling is the degree to which components of a system rely on each other. The less components depend on each other the more reusable and flexible the system becomes. Our goal was a very loosely coupled system.
- **Component Singularity.** Singularity is the degree to which components have a narrowly focused purpose. In CodeIgniter, each class and its functions are highly autonomous in order to allow maximum usefulness.

..."

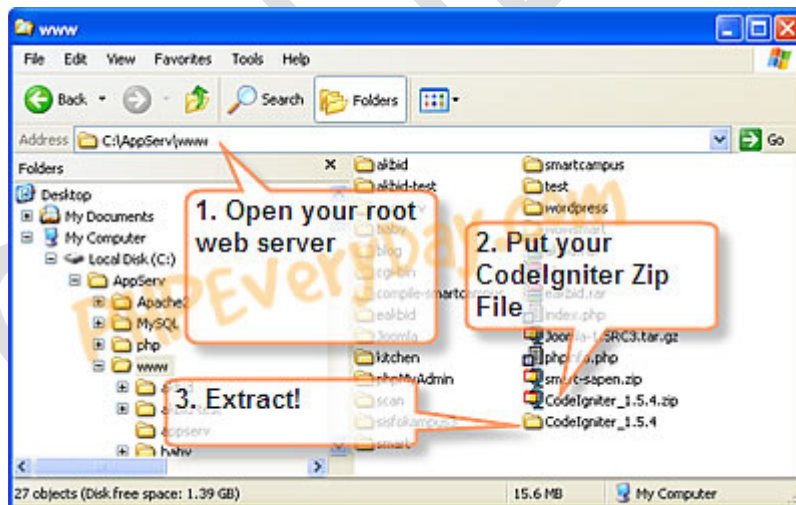
Nice feature, CodeIgniter is very flexible. We can apply at PHP 4.3.2 and above, or PHP 5. It will really help us when our web hosting still don't support PHP 5. CI support several database: MySQL, MySQLi, MS SQL, Postgre, Oracle, SQLite, and ODBC.

Ok, next post, we will learn how to install CodeIgniter.

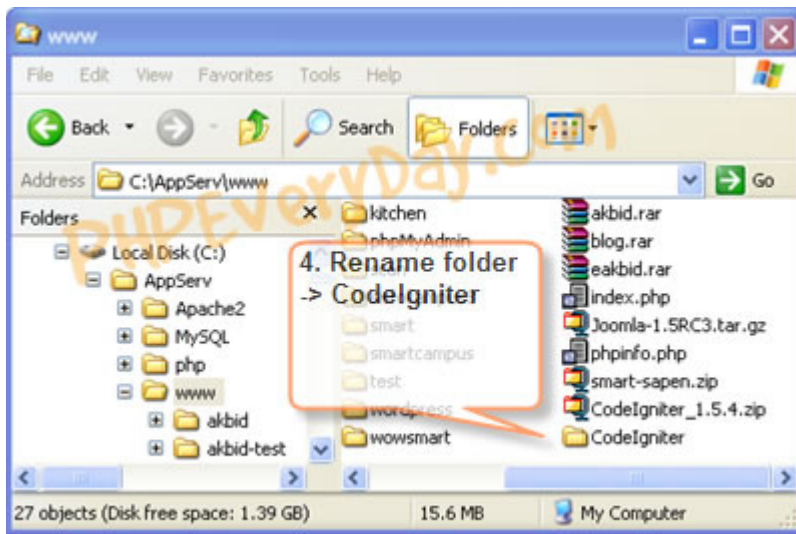
## (ii) CodeIgniter: Installation

CodeIgniter Step By Step Tutorial: After understand about codeIgniter at this, now, we learn how to install CodeIgniter. We will install to our local computer. Before follow this instruction, please download CodeIgniter at [www.codeigniter.com](http://www.codeigniter.com).

1. Open your root web server.
2. Put CodeIgniter downloaded.
3. Extract it, you will get a folder named "codeigniter\_[version]".



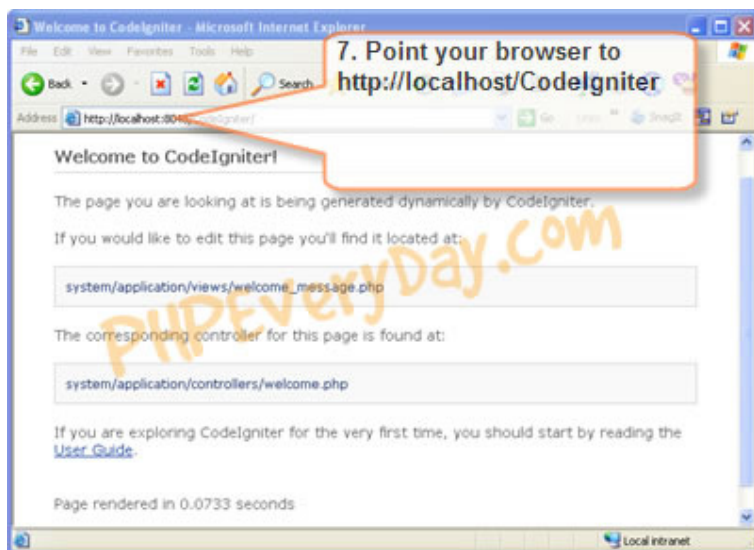
4. For simple, rename the folder to be "CodeIgniter" only.



5. Now, open config.php within CodeIgniter\system\application\config.
6. Change base site url at line 14:



7. Point your browser to <http://localhost/CodeIgniter>. You should get like this:



### (iii) CodeIgniter: Creating First Application at CodeIgniter

CodeIgniter Step By Step Tutorial: As usually, we will create first application by build hello application. But, before write code, we must know that we will build application use Model Controller View (MVC) pattern. We ever talk about MVC at [here](#). Hope you read that posting if you still don't know about MVC. Although that post about Joomla, it doesn't matter. It is same.

1. First, we make controller, create a file name "hello.php" within: \system\application\controllers. Enter following code:

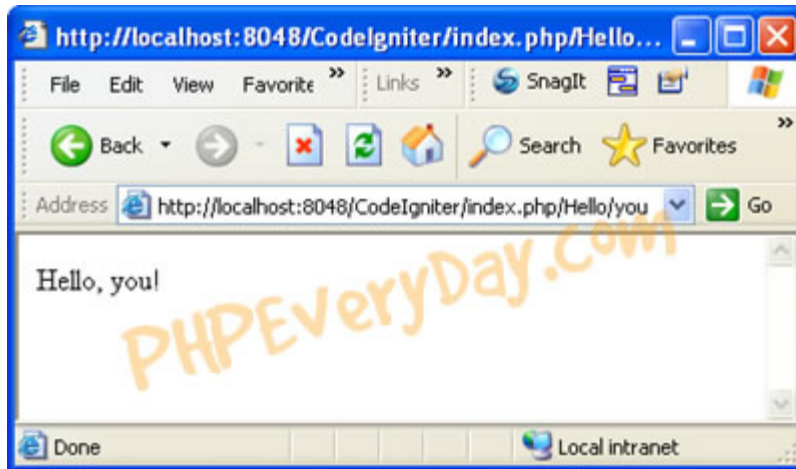
```
1 <?php
2
3 class Hello extends Controller {
4
5     function Hello()
6     {
7         parent::Controller();
8     }
9
10    function you()
11    {
12        $this->load->view('you_view');
13    }
14 }
15 ?>
```

2. Next step, make a view. Create you\_view.php within CodeIgniter\system\application\views. Enter just simple line code like:

```
1 Hello, you!
```



- Now, test your application. Point your browser to **<http://localhost/CodeIgniter/index.php/Hello/you>** You should get like this:



See, this flow:



#### (iv) CodeIgniter: Creating and Sending Parameters Between Controller and View

CodeIgniter Step By Step Tutorial: We still work with our first application at CodeIgniter. But we will build controller and view better. Create parameters in controller and send it to view. It will make our application more flexible.

Open your `hello.php` within `CodeIgniter/system/application/controller`. Overwrite become like this:

```
1 <?php
2
3 class Hello extends Controller {
4
5     var $name;
6     var $color;
7
8
9     function Hello()
10    {
11        parent::Controller();
12        $this->name = 'Andi';
13        $this->color = 'red';
14    }
15
16    function you()
17    {
18        $data['name'] = $this->name;
19        $data['color'] = $this->color;
20        $this->load->view('you_view', $data);
21    }
22 }
23 ?>
```

declare variables or class properties

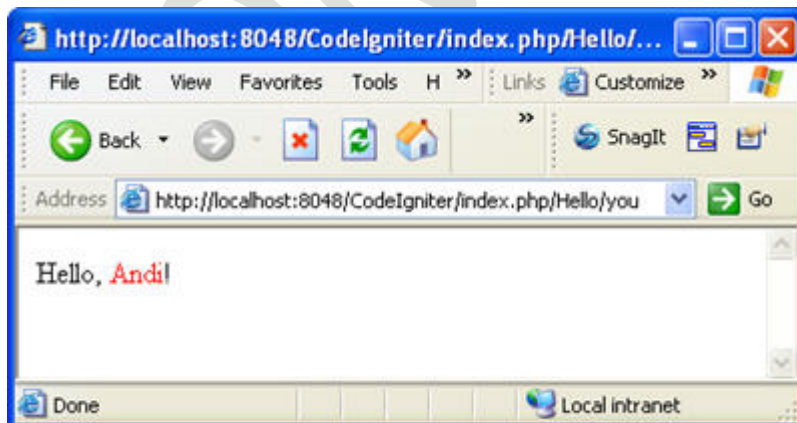
give default value

define variabel sent to views

Now, open your you\_view.php within CodeIgniter/system/application/views. Rewrite with following code:

```
1 Hello,
2 <font color="<?=$color?>"><?=$name?></font>!
```

Point your browser to <http://localhost/CodeIgniter/index.php/hello/you>.



This is analysis:



<http://localhost/codeigniter/index.php/hello/you/Imanda/fa>

```
controller: hello.php
function you($first,$second){
    ...
}
```

## (v) CodeIgniter: Getting Parameters From GET

CodeIgniter Step By Step Tutorial: If [previous post](#) we learn how to send parameter to view, now, we will learn how to get parameter from GET. As we know, we can see this parameter from Url. Example: <http://localhost/index.php?name=Andi>. Clear, we can get parameter name. How about in CodeIgniter?

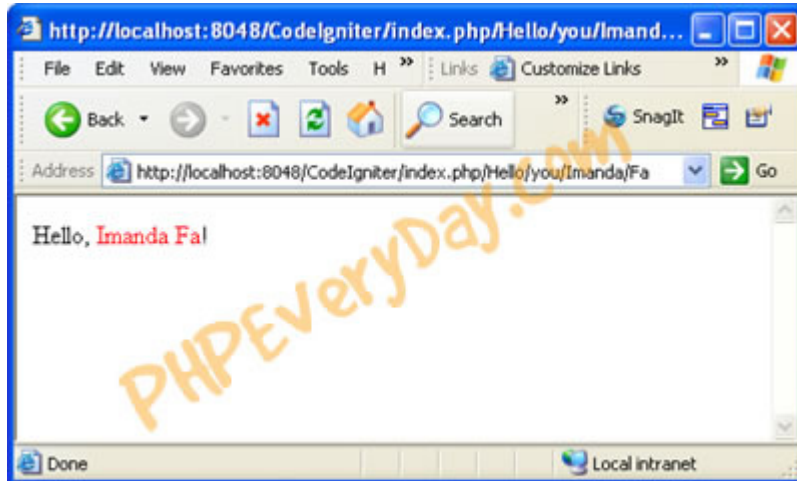
Ok, we learn by doing. Open again your hello.php within CodeIgniter/system/application/controller. Rewrite with following code:

```
1  <?php
2
3  class Hello extends Controller {
4
5      var $name;
6      var $color;
7
8
9      function Hello()
10     {
11         parent::Controller();
12         $this->name = 'Andi';
13         $this->color = 'red';
14     }
15
16     function you($firstname='', $lastname='')
17     {
18         $data['name'] =
19             ($firstname) ? $firstname.' '.$lastname :
20             $this->name;
21         $data['color'] = $this->color;
22         $this->load->view('you_view', $data);
23     }
24 }
25
26
27 ?>
```

Give variabel that can catch GET value

Use it

Now, try point your browser to <http://localhost/CodeIgniter/index.php/hello/you/Imanda/Fa>



This is analysis:

`http://localhost/codeigniter/index.php/hello/you/imanda/fa`

```
controller: hello.php
function you($first,$second){
    ...
}
```

## (vi)CodeIgniter: Setting Database Configuration

CodeIgniter Step By Step Tutorial: [Previous post](#), we have build simple first application. That application do not use database. In this post, we try to set database for CodeIgniter.

please, open database.php within CodeIgniter\system\application\config. Set config like below:

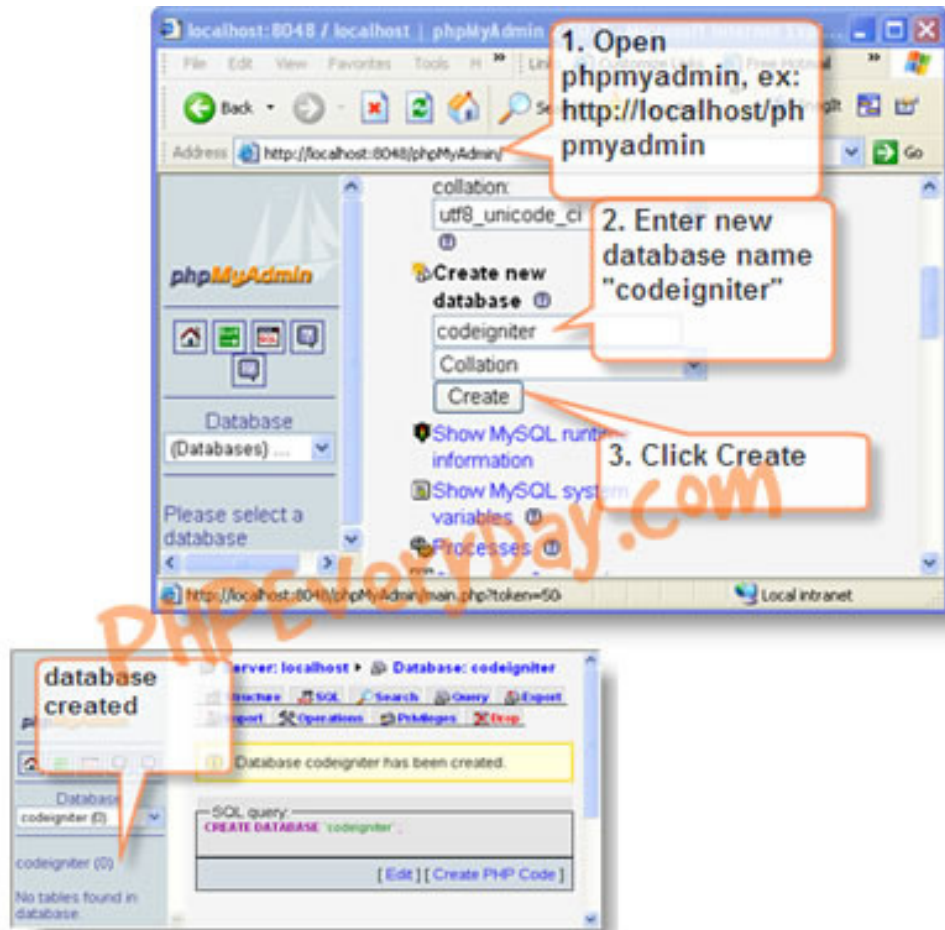
```
33
34 $active_group = "default";
35
36 $db['default']['hostname'] = "localhost";
37 $db['default']['username'] = "root";
38 $db['default']['password'] = "password";
39 $db['default']['database'] = "codeigniter";
40 $db['default']['dbdriver'] = "mysql";
41 $db['default']['dbprefix'] = "";
42 $db['default']['active_r'] = TRUE;
43 $db['default']['pconnect'] = TRUE;
44 $db['default']['db_debug'] = TRUE;
45 $db['default']['cache_on'] = FALSE;
46 $db['default']['cachedir'] = "";
47
```

Make sure, it match with your database.

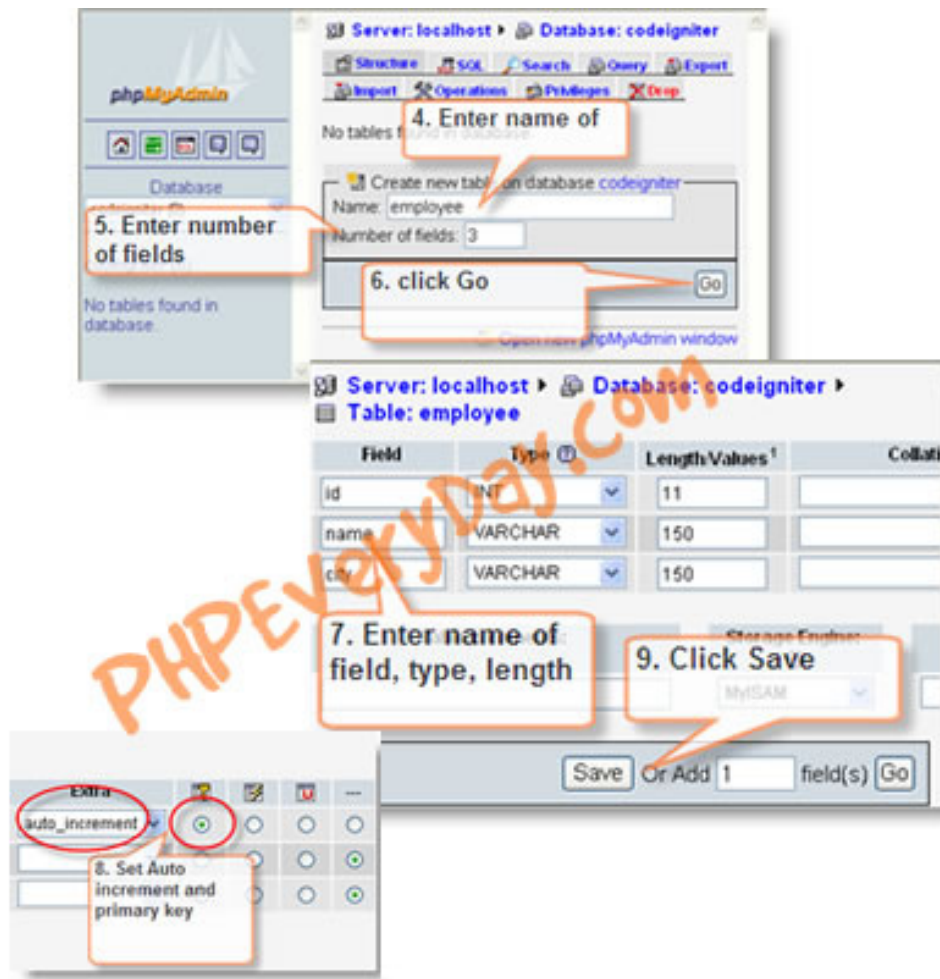
### (vii) CodeIgniter: Preparing Database

CodeIgniter Step By Step Tutorial: After [set database configuration](#), we will learn about showing data from database in CodeIgniter. But, before that, we prepare a database for practice. This post create a database named "codeigniter" and a table named "hello". We use phpMyAdmin for easy.

1. Open your phpmyadmin.
2. Enter database name "codeigniter" in **create new database** field.
3. Click Create button. Your database will be created.



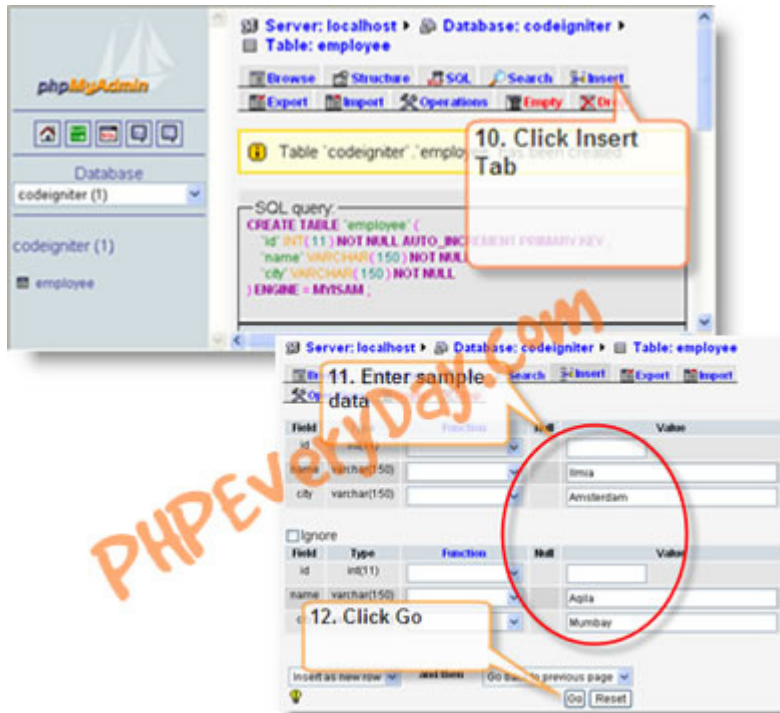
4. Create new table by entering name of new table at create new table field.
5. Enter number of fields.
6. Click Go button.
7. Enter name of field, type, and length
8. Choose auto increment and check primary key for primary key field.
9. Click Save button.



10. Now, we insert data. Click Insert tab.

11. Enter data.

12. Click Go for saving.



### (viii) CodeIgniter: Showing Simple All Data

CodeIgniter Step By Step Tutorial: We have set [database](#) and create [table](#) for CodeIgniter practice. Now, we learn how to show data in CodeIgniter. As we know, CodeIgniter use MVC pattern. We will use Model for retrieve data from database.

First, build a model. Create a file named "employee\_model.php" within CodeIgniter/CodeIgniter/system/application/models. Enter following code:



```
1 <?
2 class Employee_model extends Model {
3
4     function Employee_model()
5     {
6         parent::Model();
7     }
8
9     function employee_getall()
10    {
11        $this->load->database();
12        $query = $this->db->get('employee');
13        return $query->result();
14    }
15
16 }
17
18 ?>
```

Next, we make a view. Create a file named "employee\_viewall.php" within CodeIgniter\system\application\views. Enter following code:

```
1 <?php
2 foreach($query as $row){
3     print $row->id;
4     print $row->name;
5     print $row->city;
6     print "<br>";
7 }
8 ?>
```

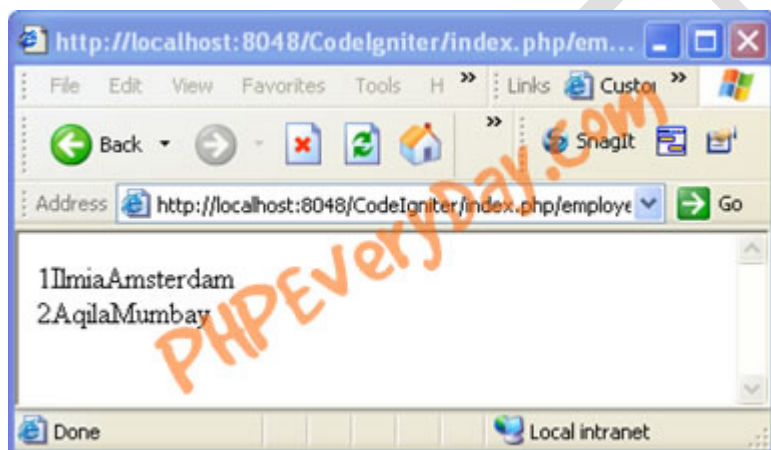
Build controller. Create a file named "employee.php" within CodeIgniter\system\application\controllers. Enter following code:

```
1 <?php
2 class employee extends Controller {
3
4     function employee()
5     {
6         parent::Controller();
7     }
8
9     function GetAll()
10    {
11        $this->load->model('employee_model');
12        $data['query'] =
13            $this->employee_model->employee_getall();
14        $this->load->view('employee_viewall', $data);
15    }
16 }
17
18 ?>
```

call function at model

load model

Now, try to point your browser to <http://localhost/CodeIgniter/index.php/employee/getall>



### (xi) CodeIgniter: Showing Simple One Data

CodeIgniter Step By Step Tutorial: After [show all data](#), We learn how to show one data. We want a data with certain ID, example ID = 1. We still use same files with previous practice.

For model: Open "employee\_model.php" within CodeIgniter\system\application\models. Add following function:

```

1  <?
2  class Employee_model extends Model {
3
4      function Employee_model()
5      {
6          parent::Model();
7      }
8
9      function employee_getall()
10     {
11         $this->load->database();
12         $query = $this->db->get_where('employee', array('id' => 1));
13         return $query->result();
14     }
15
16     function employee_get(){
17         $this->load->database();
18         $query = $this->db->get_where('employee', array('id' => 1));
19         return $query->row_array();
20     }
21 }
22
23
24  ?>

```

Show a data with id=1

return array model

For Controller: Open "employee.php" within CodeIgniter\system\application\controllers. Add following function:

```

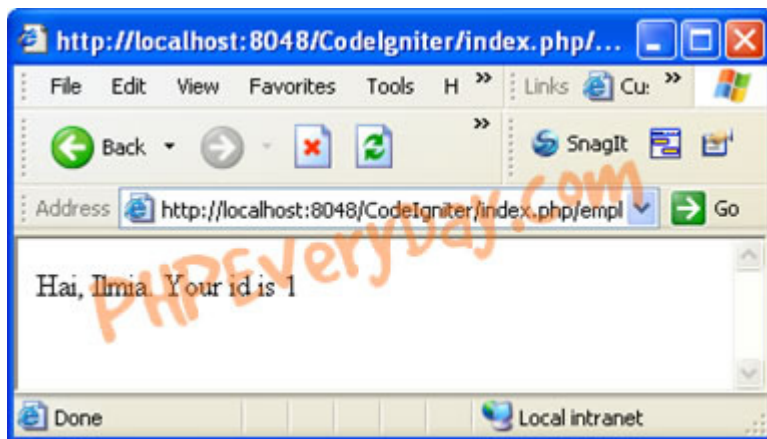
1  <?php
2  class employee extends Controller {
3
4      function employee()
5      {
6          parent::Controller();
7      }
8
9      function GetAll()
10     {
11         $this->load->model('employee_model');
12
13         $data['query'] =
14             $this->employee_model->employee_getall();
15
16         $this->load->view('employee_viewall', $data);
17     }
18
19     function Get(){
20         $this->load->model('employee_model');
21
22         $data['query'] =
23             $this->employee_model->employee_get();
24
25         $this->load->view('employee_view', $data);
26     }
27 }
28  ?>

```

For View: Open "employee\_view.php" within CodeIgniter\system\application\views. Enter following code:

```
1 Hai, <?=$query['name']?>.  
2 Your id is <?=$query['id']?>
```

Now, test it. Point your browser to <http://localhost/CodeIgniter/index.php/employee/get>.



### (x) CodeIgniter: Choosing a Data From GET

CodeIgniter Step By Step Tutorial: At [previous tutorial](#), we learn show a data. But we have define ID that we want to show. In this post, we learn how to show a data that ID defined from GET.

For model: Open "employee\_model.php" within CodeIgniter\system\application\models. Add following function:

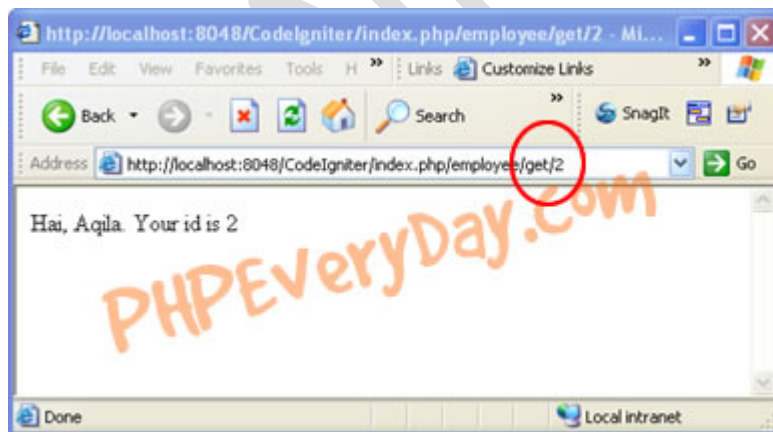
```
1 <?php  
2 class Employee_model extends Model {  
3  
4     function Employee_model()  
5     {  
6         parent::__construct();  
7     }  
8  
9     function employee_get($id)  
10    {  
11        $this->load->database();  
12        $query = $this->db->getwhere('employee', array('id' => $id));  
13        return $query->row_array();  
14    }  
15  
16    function employee_get($id){  
17        $this->load->database();  
18        $query = $this->db->getwhere('employee', array('id' => $id));  
19        return $query->row_array();  
20    }  
21  
22 }  
23  
24 <?php
```

value this flexible ID come from controller

For Controller: Open "employee.php" within CodeIgniter\system\application\controllers. Add following function:

```
1 <?php
2 class employee extends Controller {
3
4     function employee()
5     {
6         parent::Controller();
7     }
8
9     function GetAll()
10    {
11        $this->load->model('employee_model');
12
13        $data['query'] =
14            Catch from GET model->employee_getall();
15
16        $this->load->view('employee_viewall', $data);
17    }
18
19    function Get($id){
20        $this->load->model('employee_model');
21
22        $data['query'] =
23            Will be send to
24            model
25            $this->employee_model->employee_get($id);
26
27        $this->load->view('employee_view', $data);
28    }
29 }
```

Now, test it. Point your browser to <http://localhost/CodeIgniter/index.php/employee/get/2>.



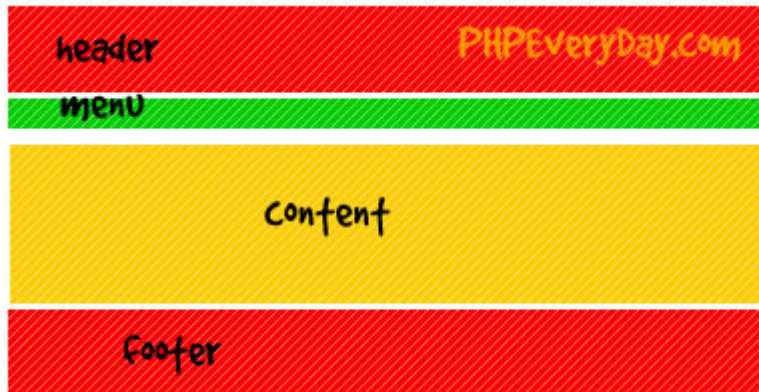


## Learn CodeIgniter Framework Form

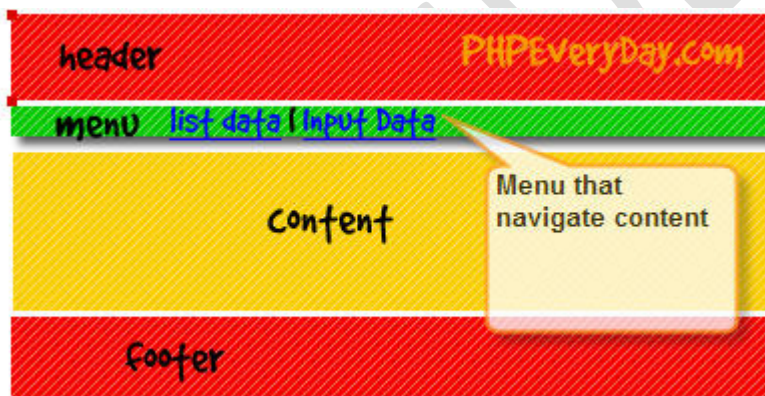
### 1. CodeIgniter - Form: Creating Skeleton

CodeIgniter Form Step By Step Tutorial - Part 1: We ever talk introduction of CodeIgniter at [this post](#). At that post, we learn about first application and simple data display. In this post, we will talk more deep about creating form. We will learn libraries such as form, html, etc.

In this tutorial series, we will create simple application with layout like this:



This application, we called "books". It have two pages: input data and display list data.



Ok, we will continue with involved file at next post.

### 2. CodeIgniter - Form: File Structure

CodeIgniter Form Step By Step Tutorial - Part 2: For [this practice](#), we need 4 main files: books.php (controller), books\_display.php (model), and books\_main.php, books\_input.php (view). Then, there are 3 adding files: books\_header.php, books\_footer.php, books\_menu.php, and mystyles.css (as css). Create that files and put within folders like this:



- + CodeIgniter
  - **mystyles.css**
- + system
  - + application
    - + controllers
      - **books.php**
    - + models
      - **books\_model.php**
    - + views
      - **books\_footer.php**
      - **books\_header.php**
      - **books\_input.php**
      - **books\_main.php**
      - **books\_menu.php**

These are elementary files. But in the middle of tutorial, we will add some files as according to discussion in that time.

### 3. CodeIgniter - Form: Creating Layout Code

CodeIgniter Form Step By Step Tutorial - Part 3: At this post, we will write simple code for showing our layout. Follow this step:

1. Open books.php within CodeIgniter\system\application\controllers. Enter following code:

```
01 <?
02 classBooks extendsController{
03
04     functionBooks(){
05         parent::Controller();
06     }
07
08     functionmain(){
09         $this->load->view('books_main');
10     }
11
12     functioninput(){
13         $this->load->view('books_input');
14     }
15 }
16 ?>
```

2. Open books\_main.php within CodeIgniter\system\application\views.

```
01 <html>
02 <head></head>
03 <body>
```

```
04 <div id="header">
05 <? $this->load->view('books_header'); ?>
06 </div>
07 <div id="menu">
08 <? $this->load->view('books_menu'); ?>
09 </div>
10
11 test list
12
13 <div id="footer">
14 <? $this->load->view('books_footer'); ?>
15 </div>
16
17 </body>
18 </html>
```

I think, it is simple to understand for you. Right?

3. Open books\_input.php within CodeIgniter\system\application\views.

```
01 <html>
02 <head></head>
03 <body>
04 <div id="header">
05 <? $this->load->view('books_header'); ?>
06 </div>
07 <div id="menu">
08 <? $this->load->view('books_menu'); ?>
09 </div>
10
11 test input
12
13 <div id="footer">
14 <? $this->load->view('books_footer'); ?>
15 </div>
16
17 </body>
18 </html>
```

4. Open books\_menu.php within CodeIgniter\system\application\views.

1 menu :

```
2 <a href="main">List</a> | <a href="input">Input</a>
```

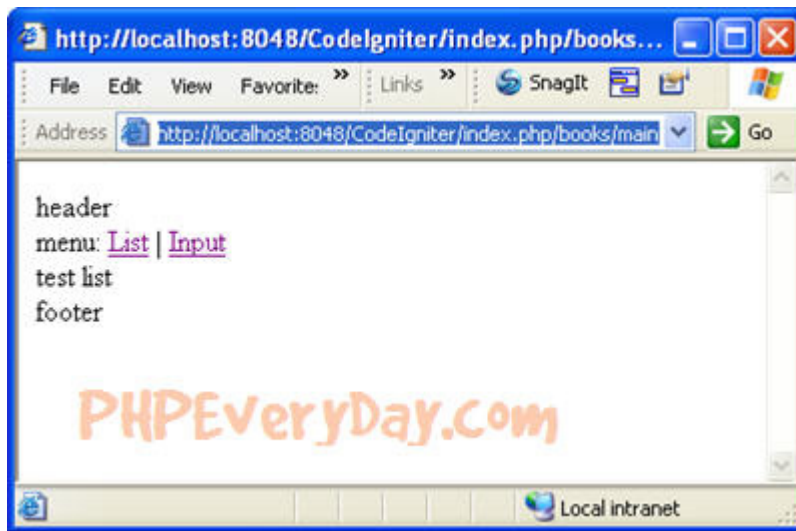
5. Open books\_header.php within CodeIgniter\system\application\views.

```
1 test header
```

6. Open books\_footer.php within CodeIgniter\system\application\views.

```
1 test footer
```

Now, point your browser to <http://localhost/CodeIgniter/index.php/books/main>



#### 4. CodeIgniter - Form: Creating Menu Library

CodeIgniter Form Step By Step Tutorial - Part 4: After create [simple layout](#), at this step, we will learn about library. For practice, we put our menu to library.

First, create a file named "MyMenu.php" within CodeIgniter\system\application\libraries. Enter following code to that file:

```
01 <?php
02 classMyMenu{
03     functionshow_menu(){
04         $obj=&get_instance();
05         $obj->load->helper('url');
06         $menu = "<ul>";
07         $menu.= "<li>";
08         $menu.= anchor("books/main","List of Books");
```

```
09         $menu.= "</li>";
10         $menu.= "<li>";
11         $menu.= anchor("books/input","Input Book");
12         $menu.= "</li>";
13         $menu.= "</ul>";
14
15         return$menu;
16     }
17 }
18 ?>
```

At creating of menu library, we need a class (we give name "MyMenu"). It have a function show\_menu(). This function access other CI classes and helpers (URL helper). URL helper will help you make url easier:

```
1 anchor("books/input","Input Book")
2 =
3 <a href="http://localhost/CodeIgniter/index.php/
4 books/input">Input Book</a>
```

This menu, we use <ul> and <li>. We will modify layout of menu use css (at next topic about css).

Now, open our controller: books.php within CodeIgniter\system\application\controllers. Update like following code:

```
01 <?
02 classBooks extendsController{
03
04     functionBooks(){
05         parent::Controller();
06     }
07
08     functionmain(){
09         $this->load->library('MyMenu');
10         $menu= newMyMenu;
11         $data['menu'] = $menu->show_menu();
12         $this->load->view('books_main',$data);
13     }
14
15     functioninput(){
16         $this->load->library('MyMenu');
17         $menu= newMyMenu;
18         $data['menu'] = $menu->show_menu();
```

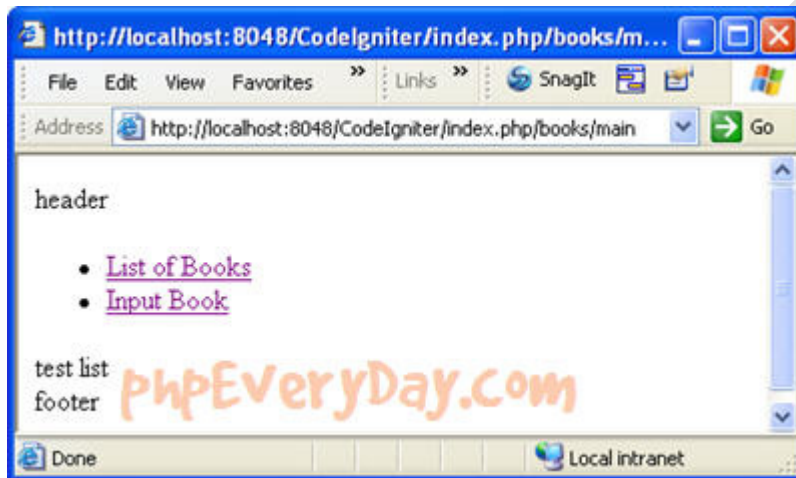
```
19         $this->load->view( 'books_input' , $data );
20     }
21 }
22 ?>
```

If you still don't understand about this code, please read [this tutorial series](#).

Next, open "books\_menu.php" within CodeIgniter\system\application\views. Replace all code with:

```
1 <?php echo$menu; ?>
```

Point your browser to <http://localhost/CodeIgniter/index.php/books/main>



## 5. CodeIgniter - Form: Putting Text at Header and Footer

CodeIgniter Form Step By Step Tutorial - Part 5: We put [menu](#) below header. But we still don't talk about header. Now, we will put some text at header. If you remember, this step is like [this post](#).

Open your controller, "books.php" within CodeIgniter\system\application\controllers. Update like this:

```
01 <?
02 classBooks extendsController{
03
04     functionBooks(){
05         parent::Controller();
06     }
07
08     functionmain(){
```

```
09         $this->load->library('MyMenu');
10         $menu= newMyMenu;
11         $data['menu']      = $menu->show_menu();
12         $data['webtitle']  = 'Book Collection';
13         $data['websubtitle']= 'We collect
14 all title of books on the world';
15         $data['webfooter'] = '© copyright
16 by step by step php tutorial';
17
18         $this->load->view('books_main',$data);
19     }
20
21     functioninput(){
22         $this->load->library('MyMenu');
23         $menu= newMyMenu;
24         $data['menu'] = $menu->show_menu();
25         $data['webtitle']  = 'Book Collection';
26         $data['websubtitle']= 'We collect
27 all title of books on the world';
28         $data['webfooter'] = '© copyright
29 by step by step php tutorial';
30
31         $this->load->view('books_input',$data);
32     }
33 }
34 ?>
```

Then update header, "books\_header.php" within CodeIgniter\system\application\views.

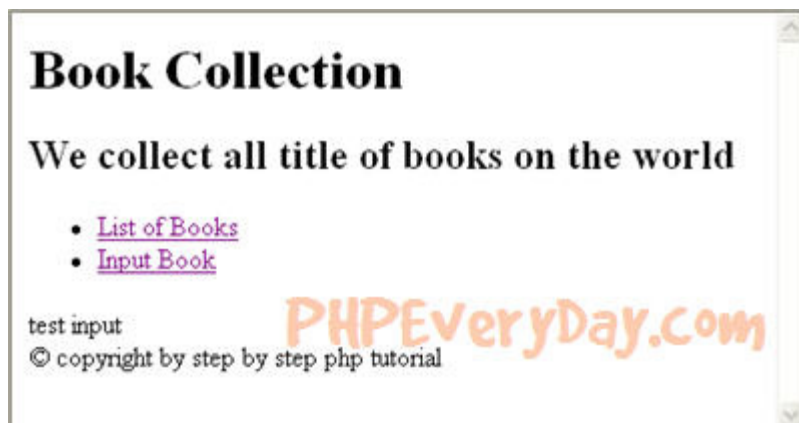
```
01 <h1 id="title">
02 <?php echo$webtitle; ?>
03 </h1>
04 <h2>
05 <?php echo$websubtitle; ?>
06 </h2>
07
```

08 Last, update footer, "books\_footer.php" within  
CodeIgniter\system\application\views.

```
09 <code>
10 <?php echo$webfooter; ?>
11 </code>
```



Result like this:



## 6. CodeIgniter - Form: Centralizing \$data

CodeIgniter Form Step By Step Tutorial - Part 6: At [previous post](#), we always write code that call library every page at controller. It is not efficient. We can centralize all that always need to display every pages (header and footer too). In this post, we use model for place them.

First, add new function called general() at model. Open "books\_model.php" within CodeIgniter\system\application\models. Add like following code:

```
01 <?
02 classbooks_model extendsModel{
03
04     functionbooks_model(){
05         parent::Model();
06         $this->load->helper('url');
07     }
08     functiongeneral(){
09         $this->load->library('MyMenu');
10         $menu= newMyMenu;
11         $data['menu']      = $menu->show_menu();
12         $data['webtitle']  = 'Book Collection';
13         $data['websubtitle']= 'We collect
14 all title of books on the world';
15         $data['webfooter'] = '© copyright
16 by step by step php tutorial';
17
18         return$data;
19     }
```

```
20 }
21 ?>
22
23 Then, update controller: books.php:
24 <code><?
25 classBooks extendsController{
26
27     functionBooks(){
28         parent::Controller();
29     }
30
31     functionmain(){
32         $this->load->model('books_model');
33         $data= $this->books_model->general();
34
35         $this->load->view('books_main',$data);
36     }
37
38     functioninput(){
39         $this->load->model('books_model');
40         $data= $this->books_model->general();
41
42         $this->load->view('books_input',$data);
43     }
44 }
45 ?></code>
```

```

<?
class Books extends Controller{

    function Books(){
        parent::Controller();
    }

    function main(){
        $this->load->model('books_model');
        $data = $this->books_model->general();
        $this->load->view('books_main',$data);
    }
}

<?
class books_model extends Model(
    );
    function books_model(){
        parent::Model();
        $this->load->helper('url');
    }

    function general(){
        $this->load->library('MyMenu');
        $menu = new MyMenu;
        $data['menu'] = $menu->show_menu();
        $data['webtitle'] = 'Book Collection';
        $data['websubtitle'] = 'We collect all titles';
        $data['webfooter'] = 'copy: copyright by PHPEveryDay.com';
        return $data;
    }
}
?>

```

## 7. CodeIgniter - Form: Adding CSS

CodeIgniter Form Step By Step Tutorial - Part 7: At [previous post](#), we talk about how to collect all parameters at one general method at model. Now, we see another advantage of that way. For example, we will add a css. Follow these steps:

1. define general css that used. Open "config.php" within directory CodeIgniter\system\application\config.
2. Add following code. The line posisi is up to you. I add at last line before ?>

```
1 $config['css'] = 'mystyles.css';
```

3. Create a file named mystyles.css within root application: \CodeIgniter\.
4. Enter example code like this:

```

01 h1{
02     font-family:Verdana, Arial, Helvetica, sans-serif;
03     font-size:30px;
04 }
05 p{
06     font-family:Georgia, "Times New Roman", Times, serif;
07     font-size:12px;
08 }

```

```
09 #menu{
10     height:50px;
11 }
12 #menu ul{
13     list-style-type:none;
14     margin-left:0px;
15 }
16 #menu li{
17     float:left;
18     padding:10px;
19 }
20 #footer{
21     font-family:Arial, Helvetica, sans-serif;
22     font-size:10px;
23 }
24 ?>
```

5. Open model file named "books\_model.php" within CodeIgniter\system\application\models. Add two lines like line 12 and 13 following code:
6. Then, update "books\_main.php" within CodeIgniter\system\application\views. Update <head></head>, like this:

```
01 <html>
02 <head>
03 <link rel="stylesheet" type="text/css"
04     href="<?php echo "$base/$css"?>">
05 </head>
06 <body>
07 <div id="header">
08 <? $this->load->view('books_header'); ?>
09 </div>
10 <div id="menu">
11 <? $this->load->view('books_menu'); ?>
12 </div>
13
14 test list
15
16 <div id="footer">
17 <? $this->load->view('books_footer'); ?>
18 </div>
19
20 </body>
```

```
21 </html>
```

7. Last, update "books\_input.php" within CodeIgniter\system\application\views. Update <head></head>, like this:

```
01 <html>
02 <head>
03 <link rel="stylesheet" type="text/css"
04     href="<?php echo "$base/$css"?>">
05 </head>
06 <body>
07 <div id="header">
08 <? $this->load->view('books_header'); ?>
09 </div>
10 <div id="menu">
11 <? $this->load->view('books_menu'); ?>
12 </div>
13
14 test input
15
16 <div id="footer">
17 <? $this->load->view('books_footer'); ?>
18 </div>
19
20 </body>
21 </html>
```

Point your browser to <http://localhost/CodeIgniter/index.php/books/main>. May be get like this:



## 8. CodeIgniter - Form: Creating Form HTML

CodeIgniter Form Step By Step Tutorial - Part 8: Now, we begin create a form for input data. We will use HTML Form CodeIgniter style. It is more save your time!

First, we need load Form helper. Put at controller (books.php within CodeIgniter\system\application\controllers), input() method. You can see line 16 at following code

```
<?
class Books extends Controller{

function Books(){
    parent::Controller();
}

function main(){
    $this->load->model('books_model');
    $data = $this->books_model->general();

    $this->load->view('books_main',$data);
}

function input(){
    $this->load->helper('form');
    $this->load->model('books_model');
    $data = $this->books_model->general();

    $this->load->view('books_input',$data);
}
}
?>
```

Open "books\_input.php" within CodeIgniter\system\application\views. Update like following code:

```
<html>
<head>
<link rel="stylesheet" type="text/css"
href="<?php echo "$base/$css"?>">
</head>
<body>
<div id="header">
<? $this->load->view('books_header'); ?>
</div>
<div id="menu">
<? $this->load->view('books_menu'); ?>
</div>

<? echoform_open('books/input'); ?>
<? echo $title; ?>:
<? echoform_input('title'); ?>
</br>
<? echo $author; ?>:
<? echoform_input('author'); ?>
</br>
<? echo $publisher; ?>:
```



```

<? echoform_input('publisher'); ?>
</br>
<? echo $year; ?>:
<? echoform_dropdown('year',$years); ?>
</br>
<? echo $available; ?>:
<? echoform_checkbox('available','yes',TRUE); ?>
</br>
<? echo $summary; ?>:
<? echoform_textarea('summary'); ?>
</br>
<? echoform_submit('mysubmit','Submit!'); ?>
<? echoform_close(); ?>

<div id="footer">
<? $this->load->view('books_footer'); ?>
</div>

</body>
</html>

```

Do You can feel its difference with ordinary HTML? Yup, it more simple and save your time. The code still not work. We must define parameters at model.

Edit your model, and add parameter

```

<?
classbooks_model extends Model{

functionbooks_model(){
    parent::Model();
    $this->load->helper('url');
}

function general(){
    $this->load->library('MyMenu');
    $menu = new MyMenu;
    $data['base']      = $this->config->item('base_url');
    $data['css']       = $this->config->item('css');
    $data['menu']      = $menu->show_menu();
    $data['webtitle']  = 'Book Collection';
    $data['websubtitle']= 'We collect all title of
                        books on the world';
    $data['webfooter'] = '@ copyright by step
                        by step php tutorial';

    $data['title']     = 'Title';
    $data['author']    = 'Author';
    $data['publisher'] = 'Publisher';
    $data['year']      = 'Year';
    $data['years']     = array('2007'=>'2007',
                                '2008'=>'2008',
                                '2009'=>'2009');

    $data['available'] = 'Available';
    $data['summary']   = 'Summary';
}
}

```

```

        return $data;
    }
}
?>

```

Now, point your browser to <http://localhost/CodeIgniter/index.php/books/input>. May be you will get like this:

**Book Collection**

We collect all title of books on the world

[List of Books](#) [Input Book](#)

Title:

Author:

Publisher:

Year: 2007

Available: ☒

Summary:

© copyright by step by step php tutorial

## 9. CodeIgniter - Form: Creating Beautiful Form Code

CodeIgniter Form Step By Step Tutorial - Part 9: We have learned create form at previous post. Now, we will modify that code in order to more clean and beautiful.

First, open again your "books\_input.php" within views. Update like following code:

```

01 <html>
02 <head>
03 <link rel="stylesheet" type="text/css"
04     href="<?php echo "$base/$css"?">">

```

```

05 </head>
06 <body>
07 <div id="header">
08 <? $this->load->view('books_header'); ?>
09 </div>
10 <div id="menu">
11 <? $this->load->view('books_menu'); ?>
12 </div>
13
14 <? echoform_open('books/input'); ?>
15 <? echo$title      .' : '.form_input($ftitle); ?></br>
16 <? echo$author     .' : '.form_input($fauthor); ?></br>
17 <? echo$publisher   .' : '.form_input($fpublisher); ?></br>
18 <? echo$year        .' : '.form_dropdown('year',$years); ?></br>
19 <? echo$available    .' : '.form_checkbox($favailable); ?></br>
20 <? echo$summary.' : '.form_textarea($fsummary); ?></br>
21 <? echoform_submit('mysubmit','Submit!'); ?>
22 <? echoform_close(); ?>
23
24 <div id="footer">
25 <? $this->load->view('books_footer'); ?>
26 </div>
27
28 </body>
29 </html>

```

You can see such as line 15. Before, we write like this

```
1 form_input('title');
```

Now, we write like this:

```
1 form_input($ftitle);
```

Why we do this? Because we want to control parameters for <input> tag. We will add parameters like this

```

1 $data['ftitle']      = array('name'=>'title',
2                          'size'=>30
3                          );

```

Can you see what I mean? Ok, for more clear, update your model like following code:

```

01 <?
02 classbooks_model extendsModel{
03

```

```
04 functionbooks_model(){
05     parent::Model();
06     $this->load->helper('url');
07 }
08
09 functiongeneral(){
10     $this->load->library('MyMenu');
11     $menu= newMyMenu;
12     $data['base']      = $this->config->item('base_url');
13     $data['css']       = $this->config->item('css');
14     $data['menu']      = $menu->show_menu();
15     $data['webtitle']  = 'Book Collection';
16     $data['websubtitle']= 'We collect all title of
17                             books on the world';
18     $data['webfooter'] = '© copyright by step
19                             by step php tutorial';
20
21     $data['title']     = 'Title';
22     $data['author']    = 'Author';
23     $data['publisher'] = 'Publisher';
24     $data['year']      = 'Year';
25     $data['years']     = array('2007'=>'2007',
26                               '2008'=>'2008',
27                               '2009'=>'2009');
28     $data['available'] = 'Available';
29     $data['summary']   = 'Summary';
30
31     $data['ftitle']    = array('name'=>'title',
32                               'size'=>30
33                               );
34     $data['fauthor']   = array('name'=>'author',
35                               'size'=>30
36                               );
37     $data['fpublisher'] = array('name'=>'publisher',
38                               'size'=>30
39                               );
40     $data['favailable'] = array('name'=>'available',
41                               'value'=>'yes',
42                               'checked'=>TRUE
43                               );
44     $data['fsummary']  = array('name'=>'summary',
45                               'rows'=>5,
```

```

46             'cols'=>30
47         );
48     return$data;
49 }
50 }
51

```

?&gt;

See line 29 and rest. You will understand what I mean. It is more simple and clearly to write HTML code.

## Book Collection

We collect all title of books on the world

[List of Books](#) [Input Book](#)

Title :

Author :

Publisher :

Year :

Available : ☒

Summary :

© copyright by step by step php tutorial

### 10. CodeIgniter - Form: CodeIgniter HTML Style

CodeIgniter Form Step By Step Tutorial - Part 10: We have learn create form using HTML at [previous post](#). CodeIgniter have library for other HTML tag. For use it, we must call HTML helper.

First, load HTML helper at controller. You can see at line 17:

```

01 <?
02 classBooks extendsController{
03
04     functionBooks(){
05         parent::Controller();

```

```
06  }
07
08  functionmain(){
09      $this->load->model('books_model');
10      $data= $this->books_model->general();
11
12      $this->load->view('books_main',$data);
13  }
14
15  functioninput(){
16      $this->load->helper('form');
17      $this->load->helper('html');
18      $this->load->model('books_model');
19      $data= $this->books_model->general();
20
21      $this->load->view('books_input',$data);
22  }
23 }
24 ?>
```

Next, we try to implement HTML CodeIgniter style. Update your books\_input.php

```
01 <html>
02 <head>
03 <link rel="stylesheet" type="text/css"
04     href="<?php echo "$base/$css"?>">
05 </head>
06 <body>
07 <div id="header">
08 <? $this->load->view('books_header'); ?>
09 </div>
10 <div id="menu">
11 <? $this->load->view('books_menu'); ?>
12 </div>
13 <? echoheading($forminput,3) ?>
14 <? echoform_open('books/input'); ?>
15 <? echo$title    .' : '.
16     form_input($ftitle).br(); ?>
17 <? echo$author   .' : '.
18     form_input($fauthor).br(); ?>
19 <? echo$publisher .' : '.
```

```

20         form_input($fpublisher).br(); ?>
21 <? echo$year      .' : '.
22         form_dropdown('year',$years).br(); ?>
23 <? echo$available .' : '.
24         form_checkbox($favailable).br(); ?>
25 <? echo$summary.' : '.
26         form_textarea($fsummary).br(); ?>
27 <? echoform_submit('mysubmit','Submit!'); ?>
28 <? echoform_close(); ?>
29
30 <div id="footer">
31 <? $this->load->view('books_footer'); ?>
32 </div>
33
34 </body>
35 </html>

```

For example, line 13, heading(\$forminput,3) to change <h3>\$forminput</h3>. Line 16, br() to change <br>.

For complete this practice, add definition of \$forminput at model (line 28 at following code):

```

01 <?
02 classbooks_model extendsModel{
03
04     functionbooks_model(){
05         parent::Model();
06         $this->load->helper('url');
07     }
08
09     functiongeneral(){
10         $this->load->library('MyMenu');
11         $menu= newMyMenu;
12         $data['base']      = $this->config->item('base_url');
13         $data['css']       = $this->config->item('css');
14         $data['menu']      = $menu->show_menu();
15         $data['webtitle']  = 'Book Collection';
16         $data['websubtitle']= 'We collect all title of
17                             books on the world';
18         $data['webfooter'] = '© copyright by step
19                             by step php tutorial';
20

```

```
21 $data['title']      = 'Title';
22 $data['author']     = 'Author';
23 $data['publisher']  = 'Publisher';
24 $data['year']       = 'Year';
25 $data['years']      = array('2007'=>'2007',
26                             '2008'=>'2008',
27                             '2009'=>'2009');
28 $data['available']  = 'Available';
29 $data['summary']    = 'Summary';
30 $data['forminput']  = 'Form Input';
31
32 $data['ftitle']     = array('name'=>'title',
33                             'size'=>30
34                             );
35 $data['fauthor']    = array('name'=>'author',
36                             'size'=>30
37                             );
38 $data['fpublisher'] = array('name'=>'publisher',
39                             'size'=>30
40                             );
41
42 $data['favailable'] = array('name'=>'available',
43                             'value'=>'yes',
44                             'checked'=>TRUE
45                             );
46 $data['fsummary']   = array('name'=>'summary',
47                             'rows'=>5,
48                             'cols'=>30
49                             );
50 return $data;
51 }
52 }
53 ?>
```

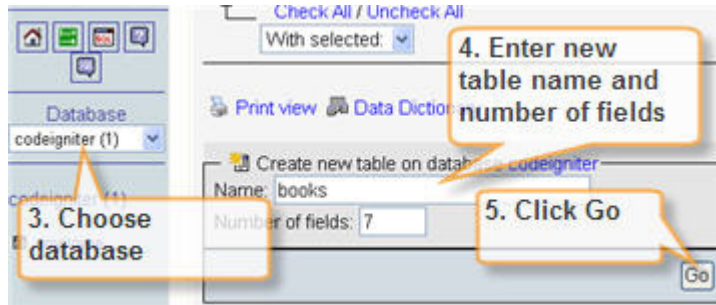
## 11. CodeIgniter - Form: Preparing Table at Database

CodeIgniter Form Step By Step Tutorial - Part 11: After [create form](#) for input data, now, we will prepare table at database. We use configuration that we ever create at [this post](#). For this practice, we will create a table named "books" at codeigniter database.

1. Open your phpmyadmin by point your browser to <http://localhost/phpmyadmin>.
2. Enter username and password (if any).
3. Choose CodeIgniter database.



- Enter name of new table and number of fields.



- Click Go button.
- Enter fields like this:

Server: localhost ▶ Database: codeigniter ▶  
Table: books

Field	Type ?	Length/Values <sup>1</sup>	
id	INT	11	
title	VARCHAR	250	
author	VARCHAR	150	
publisher	VARCHAR	150	
year	INT	4	
available	VARCHAR	3	
summary	TEXT		

- Don't forget to add auto increment and primary key like this:

Extra				---	
auto_increment ▼	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>
▼	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>
▼	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>
▼	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="checkbox"/>
▼	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>
▼	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>
▼	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>

- Last, click Save button.

## 12. CodeIgniter - Form: Creating Insert Data

CodeIgniter Form Step By Step Tutorial - Part 12: We have had [a form](#) and [a table](#). Now, we build function for insert data.

First, update our controller, books.php within controller folder. Add line 21,22,23 like below:

```
01 <?
02 classBooks extendsController{
03
04     functionBooks(){
05         parent::Controller();
06     }
07
08     functionmain(){
09         $this->load->model('books_model');
10         $data= $this->books_model->general();
11
12         $this->load->view('books_main',$data);
13     }
14
15     functioninput(){
16
17         $this->load->helper('form');
18         $this->load->helper('html');
19         $this->load->model('books_model');
20
21         if($this->input->post('mysubmit')){
22             $this->books_model->entry_insert();
23         }
24
25         $data= $this->books_model->general();
26
27         $this->load->view('books_input',$data);
28     }
29
30 }
31 ?>
```

If someone click submit button, we will call entry\_insert() from model books.

Ok, now, update our model books\_model.php within models folder. Add a method named entry\_insert() like below:

```
01 <?php
02 classbooks_model extendsModel{
03
04     functionbooks_model(){
05         parent::Model();
06         $this->load->helper('url');
07     }
08
09     functionentry_insert(){
10         $this->load->database();
11         $data= array(
12             'title'=>$this->input->post('title'),
13             'author'=>$this->input->post('author'),
14             'publisher'=>$this->input->post('publisher'),
15             'year'=>$this->input->post('year'),
16             'available'=>$this->input->post('available'),
17             'summary'=>$this->input->post('summary'),
18         );
19         $this->db->insert('books',$data);
20     }
21
22     functiongeneral(){
23         $this->load->library('MyMenu');
24         $menu= newMyMenu;
25         $data['base']      = $this->config->item('base_url');
26         $data['css']       = $this->config->item('css');
27         $data['menu']      = $menu->show_menu();
28         $data['webtitle']  = 'Book Collection';
29         $data['websubtitle']= 'We collect all title of
30                             books on the world';
31         $data['webfooter'] = '© copyright by step
32                             by step php tutorial';
33
34         $data['title']     = 'Title';
35         $data['author']    = 'Author';
36         $data['publisher'] = 'Publisher';
37         $data['year']      = 'Year';
38         $data['years']     = array('2007'=>'2007',
39                                   '2008'=>'2008',
```

```

40             '2009'=>'2009' );
41     $data['available'] = 'Available';
42     $data['summary']   = 'Summary';
43     $data['forminput'] = 'Form Input';
44
45     $data['ftitle']     = array('name'=>'title',
46                               'size'=>30
47                               );
48     $data['fauthor']    = array('name'=>'author',
49                               'size'=>30
50                               );
51     $data['fpublisher'] = array('name'=>'publisher',
52                               'size'=>30
53                               );
54     $data['favailable'] = array('name'=>'available',
55                               'value'=>'yes',
56                               'checked'=>TRUE
57                               );
58     $data['fsummary']   = array('name'=>'summary',
59                               'rows'=>5,
60                               'cols'=>30
61                               );
62     return $data;
63 }
64 }
65 ?>

```

Ok. You can test it. Input some data from input form.

### 13. CodeIgniter - Form: Creating List Data Use Table Library

CodeIgniter Form Step By Step Tutorial - Part 13: At this post, we will build list of data. We don't build as manual, but, use CodeIgniter table library. It is so simple and easy.

First, update our model (books\_model.php within models). Add getall() method like below (line 22-29):

```

01 <?
02 class books_model extends Model {
03
04     function books_model() {
05         parent::Model();
06         $this->load->helper('url');
07     }

```

```
08
09 functionentry_insert(){
10     $this->load->database();
11     $data= array(
12         'title'=>$this->input->post('title'),
13         'author'=>$this->input->post('author'),
14         'publisher'=>$this->input->post('publisher'),
15         'year'=>$this->input->post('year'),
16         'available'=>$this->input->post('available'),
17         'summary'=>$this->input->post('summary'),
18     );
19     $this->db->insert('books',$data);
20 }
21
22 functiongetall(){
23     $this->load->database();
24     $this->load->library('table');
25
26     $query= $this->db->query('SELECT * FROM books');
27     $table= $this->table->generate($query);
28     return$table;
29 }
30
31 functiongeneral(){
32     $this->load->library('MyMenu');
33     $menu= newMyMenu;
34     $data['base']      = $this->config->item('base_url');
35     $data['css']       = $this->config->item('css');
36     $data['menu']      = $menu->show_menu();
37     $data['webtitle']  = 'Book Collection';
38     $data['websubtitle']= 'We collect all title of
39         books on the world';
40     $data['webfooter'] = '© copyright by step
41         by step php tutorial';
42
43     $data['title']     = 'Title';
44     $data['author']    = 'Author';
45     $data['publisher'] = 'Publisher';
46     $data['year']      = 'Year';
47     $data['years']     = array('2007'=>'2007',
48         '2008'=>'2008',
49         '2009'=>'2009');
```

```
50 $data['available'] = 'Available';
51 $data['summary']   = 'Summary';
52 $data['forminput'] = 'Form Input';
53
54 $data['ftitle']     = array('name'=>'title',
55                             'size'=>30
56                             );
57 $data['fauthor']    = array('name'=>'author',
58                             'size'=>30
59                             );
60 $data['fpublisher'] = array('name'=>'publisher',
61                             'size'=>30
62                             );
63 $data['favailable'] = array('name'=>'available',
64                             'value'=>'yes',
65                             'checked'=>TRUE
66                             );
67 $data['fsummary']   = array('name'=>'summary',
68                             'rows'=>5,
69                             'cols'=>30
70                             );
71 return $data;
72 }
73 }
74 ?>
```

Then, update main() at controller: books.php. Add line 12-13 like below

```
01 <?
02 class Books extends Controller{
03
04     function Books(){
05         parent::Controller();
06     }
07
08     function main(){
09         $this->load->library('table');
10         $this->load->model('books_model');
11
12         $data= $this->books_model->general();
13         $data['table'] = $this->books_model->getall();
14     }
```

```

15     $this->load->view('books_main',$data);
16 }
17
18 functioninput(){
19
20     $this->load->helper('form');
21     $this->load->helper('html');
22     $this->load->model('books_model');
23
24     if($this->input->post('mysubmit')){
25         $this->books_model->entry_insert();
26     }
27
28     $data= $this->books_model->general();
29
30     $this->load->view('books_input',$data);
31 }
32
33 }
34 ?>

```

Point your browser to <http://localhost/CodeIgniter/index.php/books/main>

## Book Collection

We collect all title of books on the world

[List of Books](#) [Input Book](#)

id	title	author	publisher	year	available	summary
1	PHP Undercover	Wiwit	Elexmedia	2007	yes	test
2	PHP Superstar	Singleton	MedioRock	2008	yes	test

© copyright by step by step php tutorial

### 14. CodeIgniter - Form: Creating Table List Without Table Library

CodeIgniter Form Step By Step Tutorial - Part 14: We have learned create table use table library at [here](#). In this post, we try to create as manual (not use table library).

First, add a method books\_getall() at model (books\_model.php) like this:

```
1 function books_getall()
2 {
3     $this->load->database();
4     $query= $this->db->get('books');
5     return $query->result();
6 }
```

Update controller (books.php), like this:

```
01 function main(){
02     $this->load->library('table');
03     $this->load->helper('html');
04     $this->load->model('books_model');
05
06     $data= $this->books_model->general();
07     $data['query'] = $this->books_model->books_getall();
08
09     $this->load->view('books_main',$data);
10 }
```

Last, update views (books\_main.php) like this:

```
01 <html>
02 <head>
03 <link rel="stylesheet" type="text/css"
04     href="<?php echo "$base/$css"?>">
05 </head>
06 <body>
07 <div id="header">
08 <? $this->load->view('books_header'); ?>
09 </div>
10 <div id="menu">
11 <? $this->load->view('books_menu'); ?>
12 </div>
13 <? echo heading('List of Books',3); ?>
14 <table border="1">
15 <tr>
16 <th>ID</th>
17 <th>Title</th>
18 <th>Author</th>
19 <th>Year</th>
```



```
20 </tr>
21 <?php
22 foreach($queryas$row){
23     echo"<tr>";
24     echo"<td>". $row->id ."</td>";
25     echo"<td>". $row->title ."</td>";
26     echo"<td>". $row->author ."</td>";
27     echo"<td>". $row->year ."</td>";
28     echo"</tr>";
29 }
30 ?>
31 </table>
32
33 <div id="footer">
34 <? $this->load->view('books_footer'); ?>
35 </div>
36
37 </body>
38 </html>
```

Result like this:

## Book Collection

We collect all title of books on the world

[List of Books](#) [Input Book](#)

PHPEveryDay

### List of Books

ID	Title	Author	Year
1	PHP Undercover	Wiwit	2007
2	PHP Superstar	Singleton	2008

© copyright by step by step php tutorial

## 15. CodeIgniter - Form: Using URL Helper

CodeIgniter Form Step By Step Tutorial - Part 15: After create table at here, we will add links for edit and delete data. We use URL Helper to build the links.

Update books\_main.php (views) with following code (add line 29 & 30):

```
01 <html>
02 <head>
03 <link rel="stylesheet" type="text/css"
04     href="<?php echo "$base/$css"?>">
05 </head>
06 <body>
07 <div id="header">
08 <? $this->load->view('books_header'); ?>
09 </div>
10 <div id="menu">
11 <? $this->load->view('books_menu'); ?>
12 </div>
13 <? echoheading('List of Books',3); ?>
14 <table border="1">
15   <tr>
16     <th>ID</th>
17     <th>Title</th>
18     <th>Author</th>
19     <th>Year</th>
20     <th colspan="2">Action</th>
21   </tr>
22 <?php
23 foreach($queryas$row){
24   echo"<tr>";
25   echo"<td>". $row->id ."</td>";
26   echo"<td>". $row->title ."</td>";
27   echo"<td>". $row->author ."</td>";
28   echo"<td>". $row->year ."</td>";
29   echo"<td>". anchor('books/input/'.$row->id,'Edit') ."</td>";
30   echo"<td>". anchor('books/del/'.$row->id,'Delete') ."</td>";
31   echo"</tr>";
32 }
33 ?>
34 </table>
35
36 <div id="footer">
37 <? $this->load->view('books_footer'); ?>
38 </div>
39
40 </body>
41 </html>
```

You can see, than write line like this

```
1 <a href="http://localhost/CodeIgniter/
2 index.php/books/input/1">Edit</a>
```

Better:

```
1 <?php echoanchor('books/input/1','Edit'); ?>
```

## List of Books

ID	Title	Author	Year	Action
1	PHP Undercover	Wiwit	2007	<a href="#">Edit</a> <a href="#">Delete</a>
2	PHP Superstar	Singleton	2008	<a href="#">Edit</a> <a href="#">Delete</a>

© copyright by step by step php tutorial

## 16. CodeIgniter - Form: Showing Single Data for Form Edit

CodeIgniter Form Step By Step Tutorial - Part 16: Now, we want to show single data at form for editing. User can choose a data from table that we create at [here](#).

First, create a function at model(books\_model.php) that will display single data.

```
function get($id){
    $this->load->database();
    $query = $this->db->getwhere('books',array('id'=>$id));
    return $query->row_array();
}
```

Then, update input() at controller (books.php). Do like this:

```
function input($id = 0){

    $this->load->helper('form');
    $this->load->helper('html');
    $this->load->model('books_model');

    if($this->input->post('mysubmit')){
        $this->books_model->entry_insert();
    }
    $data = $this->books_model->general();

    if((int)$id > 0){
        $query = $this->books_model->get($id);
        $data['fid']['value'] = $query['id'];
        $data['ftitle']['value'] = $query['title'];
    }
```

```

        $data['fauthor']['value'] = $query['author'];
        $data['fpublisher']['value'] = $query['publisher'];
        $data['fyear']['value'] = $query['year'];
        if($query['available']=='yes'){
            $data['favailable']['checked'] = TRUE;
        }else{
            $data['favailable']['checked'] = FALSE;
        }
        $data['fsummary']['value'] = $query['summary'];
    }

    $this->load->view('books_input',$data);
}

```

Now, input() can accept parameter from GET. If there is a value from GET, it will retrieve data from get at model (line 13). We enter the value to form (line 14 - 24).

Next, update view books\_input() become like this:

```

<html>
<head>
<link rel="stylesheet" type="text/css"
href="<?php echo "$base/$css"?>">
</head>
<body>
<div id="header">
<? $this->load->view('books_header'); ?>
</div>
<div id="menu">
<? $this->load->view('books_menu'); ?>
</div>
<? echo heading($forminput,3) ?>
<? echoform_open('books/input'); ?>
<? echoform_hidden('id',$fid['value']); ?>
<? echo $title      .' : ' .
form_input($ftitle).br(); ?>
<? echo $author     .' : ' .
form_input($fauthor).br(); ?>
<? echo $publisher   .' : ' .
form_input($fpublisher).br(); ?>
<? echo $year        .' : ' .
form_dropdown('year',$years,$fyear['value']).br(); ?>
<? echo $available   .' : ' .
form_checkbox($favailable).br(); ?>
<? echo $summary     .' : ' .
form_textarea($fsummary).br(); ?>
<? echoform_submit('mysubmit','Submit!'); ?>
<? echoform_close(); ?>

<div id="footer">
<? $this->load->view('books_footer'); ?>
</div>

</body>
</html>

```

We add input type hidden at line 15 for editing key. Then accomodate dropdown object at line 23.

Now, try to edit a data by click edit link at table. May be you get a data like:

Address <http://localhost:8048/CodeIgniter/index.php/books/input/2>

## Book Collection

We collect all title of books on the world

[List of Books](#) [Input Book](#)

### Form Input

Title :

Author :

Publisher :

Year :

Available : ☒

Summary :

© copyright by step by step php tutorial

## 17. CodeIgniter - Form: Something Happen With Form Parameters

CodeIgniter Form Step By Step Tutorial - Part 17: At [previous post](#), we talk about creating edit form. This form is normal when we sent a parameter get to show single data. But, when we didn't send any value, it raise error. Why?

Ok, try open form with <http://localhost/CodeIgniter/index.php/books/input>

## Form Input

### A PHP Error was encountered

Severity: Notice

Message: Undefined variable: fid

Filename: views/books\_input.php

Line Number: 15

Title :

Author :

Publisher :

### A PHP Error was encountered

Severity: Notice

Message: Undefined variable: fyear

Filename: views/books\_input.php

Line Number: 23

Yes, you can guess what happen. We didn't declare two parameter: \$fid and \$fyear. We define when there is id at GET. See again following controller:

```
01 function input($id= 0){
02
03     $this->load->helper('form');
04     $this->load->helper('html');
05     $this->load->model('books_model');
06
07     if($this->input->post('mysubmit')){
08         $this->books_model->entry_insert();
09     }
10     $data= $this->books_model->general();
11
12     if((int)$id> 0){
13         $query= $this->books_model->get($id);
14         $data['fid']['value'] =
15             $query['id'];
16         $data['fyear']['value'] = $query['year'];
17     }
18 }
```

```
16     $data['fauthor']['value'] = $query['author'];
17     $data['fpublisher']['value'] = $query['publisher'];
18     $data['fyear']['value'] = $query['year'];
19     if($query['available']=='yes'){
20         $data['favailable']['checked'] = TRUE;
21     }else{
22         $data['favailable']['checked'] = FALSE;
23     }
24     $data['fsummary']['value'] = $query['summary'];
25 }
26
27
28 $this->load->view('books_input',$data);
29 }
```

So we must create bait. I create it at model by add 2 line (13 & 14):

```
01 functiongeneral(){
02     $this->load->library('MyMenu');
03     $menu= newMyMenu;
04     $data['base']      = $this->config->item('base_url');
05     $data['css']       = $this->config->item('css');
06     $data['menu']      = $menu->show_menu();
07     $data['webtitle']  = 'Book Collection';
08     $data['websubtitle']= 'We collect all title of
09                          books on the world';
10     $data['webfooter'] = '© copyright by step
11                          by step php tutorial';
12
13     $data['fid']['value']      = 0;
14     $data['fyear']['value']    = 0;
15
16     $data['title']           = 'Title';
17     $data['author']          = 'Author';
18     $data['publisher']       = 'Publisher';
19     $data['year']            = 'Year';
20     $data['years']           = array('2007'=>'2007',
21                                     '2008'=>'2008',
22                                     '2009'=>'2009');
23     $data['available']       = 'Available';
24     $data['summary']         = 'Summary';
25     $data['forminput']       = 'Form Input';
```



```
26
27 $data['ftitle']      = array('name'=>'title',
28                               'size'=>30
29                               );
30 $data['fauthor']     = array('name'=>'author',
31                               'size'=>30
32                               );
33 $data['fpublisher']  = array('name'=>'publisher',
34                               'size'=>30
35                               );
36 $data['favailable']  = array('name'=>'available',
37                               'value'=>'yes',
38                               'checked'=>TRUE
39                               );
40 $data['fsummary']    = array('name'=>'summary',
41                               'rows'=>5,
42                               'cols'=>30
43                               );
44 return $data;
45 }
```

Our form become normal again.

## 18. CodeIgniter - Form: Updating Data

CodeIgniter Form Step By Step Tutorial - Part 18: After create [edit form](#), now we add updating function.

First, create entry\_update() at model.

```
01 function entry_update(){
02     $this->load->database();
03     $data= array(
04         'title'=>$this->input->post('title'),
05         'author'=>$this->input->post('author'),
06         'publisher'=>$this->input->post('publisher'),
07         'year'=>$this->input->post('year'),
08         'available'=>$this->input->post('available'),
09         'summary'=>$this->input->post('summary'),
10     );
11     $this->db->where('id',$this->input->post('id'));
12     $this->db->update('books',$data);
13 }
```

Then, update our input() at controller:

```
01 functioninput($id= 0){
02
03     $this->load->helper('form');
04     $this->load->helper('html');
05     $this->load->model('books_model');
06
07     if($this->input->post('mysubmit')){
08         if($this->input->post('id')){
09             $this->books_model->entry_update();
10         }else{
11             $this->books_model->entry_insert();
12         }
13     }
14     $data= $this->books_model->general();
15
16     if((int)$id> 0){
17         $query= $this->books_model->get($id);
18         $data['fid']['value'] =
19 $query['id'];
20         $data['ftitle']['value'] = $query['title'];
21         $data['fauthor']['value'] = $query['author'];
22         $data['fpublisher']['value'] = $query['publisher'];
23         $data['fyear']['value'] = $query['year'];
24         if($query['available']=='yes'){
25             $data['favailable']['checked'] = TRUE;
26         }else{
27             $data['favailable']['checked'] = FALSE;
28         }
29         $data['fsummary']['value'] = $query['summary'];
30     }
31
32     $this->load->view('books_input',$data);
33 }
```

If there is id that be sent by post, it will load entry\_update (line 9).

## 19. CodeIgniter - Form: Deleting Data

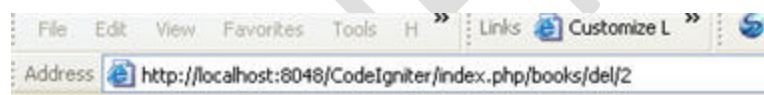
CodeIgniter Form Step By Step Tutorial - Part 19: We have created link for delete data at [data list](#). Now, we create deleting function.

First, create a function for deleting data at model.

```
1 functiondelete($id){
2   $this->load->database();
3   $this->db->delete('books', array('id'=> $id));
4 }
```

Then, create a function for deleting data at controller:

```
01 functiondel($id){
02   $this->load->library('table');
03   $this->load->helper('html');
04   $this->load->model('books_model');
05
06   if((int)$id> 0){
07     $this->books_model->delete($id);
08   }
09
10   $data= $this->books_model->general();
11   $data['query'] = $this->books_model->books_getall();
12
13   $this->load->view('books_main',$data);
14 }
```



## Book Collection

We collect all title of books on the world

[List of Books](#)   [Input Book](#)

### List of Books

ID	Title	Author	Year	Action
1	PHP Undercover	Wiwit	2007	<a href="#">Edit</a> <a href="#">Delete</a>

© copyright by step by step php tutorial