CodeIgniter Interview Questions

A list of top frequently asked **CodeIgniter interview questions** and answers are given below.

1) What is CodeIgniter?

CodeIgniter is an open source and powerful framework used for developing web applications on PHP. It is loosely based on MVC pattern and similar to Cake PHP. CodeIgniter contains libraries, simple interface and logical structure to access these libraries, plug-ins, helpers and some other resources which solve the complex functions of PHP more easily maintaining high performance. It simplifies the PHP code and brings out a fully interactive, dynamic website at a much shorter time.

2) What are the most prominent features of CodeIgniter?

A list of most prominent features of CodeIgniter:

o   It is an open source framework and free to use.
o   It is extremely light weighted.
o   It is based on the Model View Controller (MVC) pattern.
o   It has full featured database classes and support for several platforms.
o   It is extensible. You can easily extend the system by using your libraries, helpers.
o   Excellent documentation.

3) Explain the folder structure of CodeIgniter.

If you download and unzip CodeIgniter, you get the following file structure/folder structure:

Application

o   cache
o   Config
o   Controllers
o   core
o   errors
o   helpers
o   hooks
o   language

- o libraries
- o logs
- o models
- o third-party
- o views

system
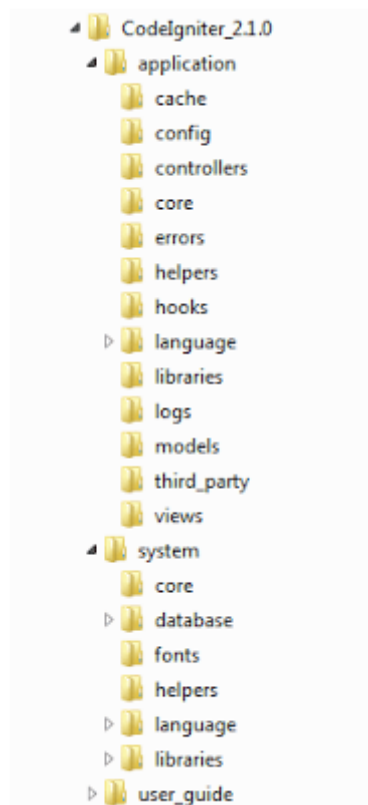
- o core
- o database
- o fonts
- o helpers
- o language
- o libraries
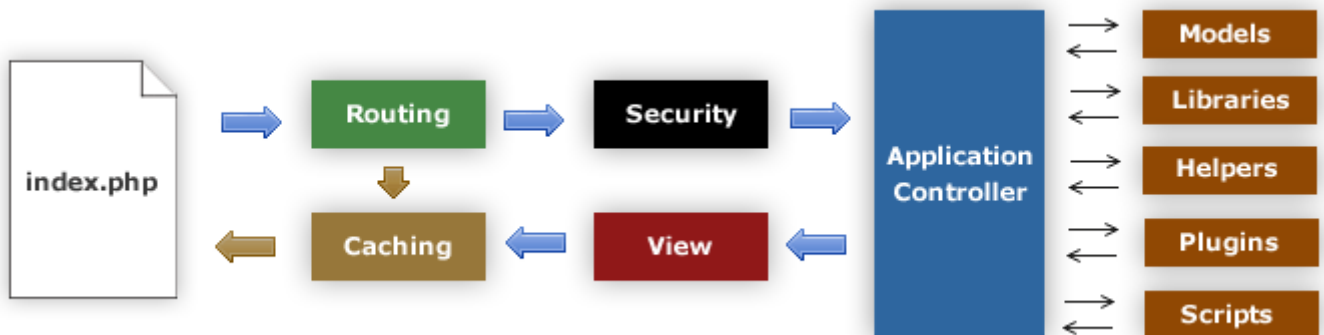

---

4) Explain CodeIgniter architecture.

From a technical point of view, CodeIgniter is dynamically instantiation (light-weighted), loosely coupled (components rely very less on each other) and has a component singularity (each class and functions are narrowly focused towards their purpose).
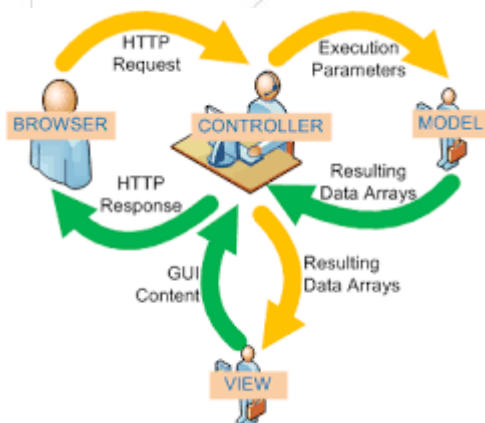
Data flow in CodeIgniter

5) Explain MVC in CodeIgniter.

CodeIgniter framework is based on MVC pattern. MVC is a software that gives you a separate logical view from the presentation view. Due to this, a web page contains minimal scripting.

- o Model - The Controller manages models. It represents your data structure. Model classes contain functions through which you can insert, retrieve or update information in your database.
- o View - View is the information that is presented in front of users. It can be a web page or parts the page like header and footer.
- o Controllers - Controller is the intermediary between models and view to process HTTP request and generates a web page. All the requests received by the controller are passed on to models and view to process the information.

More details.

6) Explain model in CodeIgniter.

Model's responsibility is to handle all data logic and representation and load data in the views. It is stored in application/models folder.

The basic structure of a model file

```
class ModelName extends CI_Model {

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

Here, ModelName is the name of your model file. Remember, the class first letter must be in an uppercase letter followed by other lowercase letters, and it should be the same as your file name. It extends the base CodeIgniter Model so that all the built-in methods of parent Model file gets inherited to the newly created file.

More details.

7) How can you add or load a model in CodeIgniter?

To load models in controller functions, use the following function:

1. $this->load->model('ModelName');

If in case your model file is located in sub-directory of the model folder, then you have to mention the full path. For example, if your file location is application/controller/models/project/ModelName. Then, your file will be loaded as shown below,

1. $this->load->model('project/ModelName');

8) How can you connect models to a database manually?

To connect database manually use following syntax,

1. $this->load->database();
   More details.

9) Explain views in CodeIgniter.

View folder contains all the markup files like header, footer, sidebar, etc. They can be reused by embedding them anywhere in controller file. They can't call directly, and they have to be loaded in the controller's file.

View syntax

Create a file and save it in application/views folder. For example, we have created a file Firstview.php,
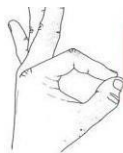
```
<!DOCTYPE html>
<html>
<head>
    <title>View Page</title>
</head>
<body>
        <h1>An example of view page</h1>

</body>
</html>
```

More details.

10) How can you load a view in CodeIgniter?

The View can't be accessed directly. It is always loaded in the controller file. Following function is used to load a view page:

1. $this->load->view('page_name');

Write your view's page name in the bracket. You don't need to specify **.php** unless you are using some other extension.

Now, go to your controller file **(Main.php)** and write this code as shown below.

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

class Main extends CI_Controller {

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

More details.

11) Explain controller in CodeIgniter.

A controller is the intermediary between models and views to process the HTTP request and generates a web page. It is the center of every request on your web application.

Consider following URI,

1. abc.com/index.php/front/

In this URI, CodeIgniter try to find Front.php file and Front class.

Controller Syntax

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

class Main extends CI_Controller {

    public function index()
    {
        |
    }
}
```

Look at the above snapshot, controller's file name is **Main.php** (the first letter has to be in uppercase), and the class name is **Main** (the first letter has to be in uppercase).

More details.

12) What is the default controller in CodeIgniter?

The file specified in the default controller loaded by default when no file name is mentioned in the URL. By default, it is welcome.php which is the first page to be seen after installing CodeIgniter.

With URL

1. localhost/codeigniter/

Welcome.php will be loaded as there is no file name mentioned in the URL.

Although as per your need, you can change the default controller in the file **application/config/routes.php.**

1. $route['default_controller'] = ' ';

Here, specify your file name which you want to be loaded by default.

More details.

13) How will you call a constructor in CodeIgniter?
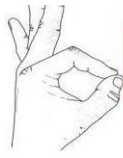
To use a constructor, you need to mention the following line of code,

1. parent::_construct()
   More details.

14) What is the basic CodeIgniter URL structure?

Instead of using 'query-string' approach, it uses a segment based approach.

Its structure is as follows,

1. abc.com/class/function/ID

The **class** represents a controller class that needs to be invoked.

The **function** is the method that is called.

ID is an additional segment that is passed to controllers.

15) What is an inhibitor of CodeIgniter?

In CodeIgniter, Inhibitor is an error handler class that uses native PHP functions like set_exception_handler, set_error_handler, register_shutdown_function to handle parse errors, exceptions, and fatal errors.

16) What is the default method name in CodeIgniter?

By default controller always calls index method. If you want to call a different method, then write it in the controller?s file and specify its name while calling the function.
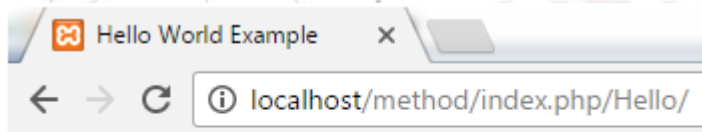
Look at the URL. There is no method name is mentioned. Hence, by default index method is loaded.

More details.

17) Explain the remapping method calls in CodeIgniter.

The Second segment of URI determines which method is being called. If you want to override it, you can use _remap() method. The _remap method always get called even if URI is different. It overrides the URI. For Example:

```
1.  public function _remap($methodName)
2.  {
3.          if ($methodName === 'a_method')
4.          {
```

```
5.          $this->method();
6.      }
7.      else
8.      {
9.          $this->defaultMethod();
10.     }
11. }
```
More details.

18) What is a helper in CodeIgniter? How can a helper file be loaded?

Helpers are the group of functions that are used to assist the user to perform specific tasks.

URL Helpers: used to create the links.

Text Helpers: used for text formatting.

Cookies Helpers: used for reading and setting cookies.

More Details.

19) How can you load multiple helper files?

To load multiple helper files, specify them in an array,

```
1. $this->load->helper(
2. array('helper1', 'helper2', 'helper3')
3. );
```
More details.

20) Explain the CodeIgniter library. How will you load it?

CodeIgniter provides a rich set of libraries. It is an essential part of CodeIgniter as it increases the developing speed of an application. It is located in the system/library.

It can be loaded as follows,

```
1. $this->load->library('class_name');
```
More details.

21) How can you create a library in CodeIgniter?

There are three methods to create a library,

- o Creating an entirely new library
- o Extending native libraries
- o Replacing native libraries

More details.

22) Where is a newly created library stored in CodeIgniter structure?

It should be placed in application/libraries folder.

More details.

23) Can you extend native libraries in CodeIgniter?

Yes, we can add some extended functionality to a native library by adding one or two methods. It replaces the entire library with your version. So it is better to extend the class. Extending and replacing is almost identical with only following exceptions.

- o The class declaration must extend the parent class.
- o New class name and filename must be prefixed with **MY_.**

For example, to extend it to native Calendar, create a
file **MY_Calendar.php** in **application/libraries** folder. Your class declared as **class MY_Calendar extends CI_Calendar}**

More details.

24) How can you extend a class in CodeIgniter?

You have to build a file name application/core/MY_Input.php and declare your class with **Class MY_Input extends CI_Input {}** to extend the native input class in CodeIgniter.

25) What is routing in CodeIgniter?

Routing is a technique by which you can define your URLs according to the requirement instead of using the predefined URLs. Routes can be classified in two ways, either using Wildcards or Regular Expressions.

Wildcards

There are two types of wildcards:

- o :num–series containing only numbers matched.
- o :any–series containing only characters matched.

Regular Expression

Regular expressions are also used to redirect routes.

1. $route['blog'(a-zA-Z0-9]+)'] = 'women/social';

You can create your regular expression to run your URL.

More Details.

26) Why is URL routes need to be configured?

There are many purposes for which the URL routes are configured.

1. To improve the number of page visits.
2. To hide the code complexities from the user.

27) What are the hooks in CodeIgniter?

The Hook is a feature in CodeIgniter that provides a way to change the inner working of the framework without hacking the core files. It facilitates you to execute a script with a particular path within the CodeIgniter. Usually, it is defined in the **application/config/hooks.php** file.

More Details.

28) How to enable CodeIgniter hook?

To enable hook, go to **application/config/config.php/** file and set it TRUE as shown below,

1. $config['enable_hooks'] = TRUE;
   More details.

29) What are different types of hook points in CodeIgniter?

A list of different types of hook points in CodeIgniter:

- o post_controller_constructor - It is called immediately after your controller is started but before any method call.
- o pre_controller - It is called immediately before your controller being called. At this point, all the classes, security checks, and routing have been done.
- o post_sytem - It is called after the final page is sent to the browser at the end of the system execution.
- o pre_system - It is called much before the system execution. Only benchmark and hook class have been loaded at this point.
- o cache_override - It enables you to call your function in the output class.
- o display_override - It is used to send the final page at the end of file execution.
- o post_controller - It is called immediately after your controller is entirely executed.

More Details.

30) What are CodeIgniter drivers?

These are a particular type of library that has a parent class and many child classes. These child classes have access to the parent class, but not to their siblings. Drivers are found in **system/libraries** folder.

More details.

31) How to initialize a driver in CodeIgniter?

To initialize a driver, write the following syntax,

1. $this->load->driver('class_name');

Here, class_name is the driver name.

More details.

32) How to create a driver in CodeIgniter?

There are three steps to create a driver:

1. Making file structure
2. Making driver list
3. Making driver(s)

More details.

33) How to connect multiple databases in CodeIgniter?

To connect more than one database simultaneously, do the following,

1. $db1 = $this->load->database('group_one', TRUE);
2. $db1 = $this->load->database('group_two', TRUE);

More details.

34) How can you print SQL statement in CodeIgniter model?

1. $this>db>insertid();

35) What are CodeIgniter security methods?

CodeIgniter security methods help to create a secure application and process input data. The methods are given below:

- XSS filtering
- CSRF (Cross-site Request Forgery)
- Class reference

More details.

36) What are the XSS security parameters?

XSS stands for cross-site scripting. Codeigniter contains a cross-site scripting hack prevention filter. The XSS filter targets methods to trigger JavaScript or other types of suspicious code. If it detects anything, it converts the data to character entities.

XSS filtering uses xss_clean() method to filer data.

1. $data = $this->security->xss_clean($data);

There is an optional second parameter, is_image, which is used to test images for XSS attacks. When this parameter is set to TRUE, it doesn't return an altered string. Instead, it returns TRUE if an image is safe and FALSE if it contains malicious information.

1. if ($this->security->xss_clean($file, TRUE) === FALSE)
2.   {
3.   //file failed in xss test
4.   }

More Details.

37) How can the CodeIgniter be prevented from CSRF?

There are the various ways by which, we can prevent CodeIgniter from CSRF. The most used method is using the hidden field in each page of the website. The hidden field is stored in the user's session. The filed is changed with every HTTP request. The user can

be detected in its every request to the website. The hidden value is always compared with the one saved in the session. If it is the same, the request is valid.

More Details.

38) How can you enable CSRF?

You can enable protection by editing config.php file and setting it to

To enable CSRF make the following statement TRUE from FALSE
in **application/config/config.php** file.

1. $config['csrf_protection'] = TRUE;
   More Details.

39) What is CSRF attack in CodeIgniter?

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including victim's session cookie and other authentication information, to a web application.

For example, suppose you have a site with a form. An attacker could create a bogus form on his site. This form could contain hidden inputs and malicious data. This form is not sent to the attacker's site, in fact, it comes to your site. Thinking that the form is genuine, your site process it.

Now suppose that the attacker's form point towards the deletion form in your site. If a user is logged in and redirected to the attacker's site and then perform the search, his account will be deleted without knowing him. That is the CSRF attack.

More details.

40) What is a token method in a CSRF attack?

To protect from CSRF, we need to connect both HTTP requests, form request and form submission. There are several ways to do this, but in CodeIgniter hidden field is used which is called the CSRF token. The CSRF token is a random value that changes with every HTTP request sent.

With each request, a new CSRF token is generated. When an object is created, name and value of the token are set.

1. $this->csrf_cookie_name = $this->csrf_token_name;
2. $this->_csrf_set_hash();

The function for it is,

```
1.  function _csrf_set_hash()
2.  {
3.      if ($this->csrf_hash == '')
4.        {
5.  if ( isset($_COOKIE[$this->csrf_cookie_name] ) AND
6.            $_COOKIE[$this->csrf_cookie_name] != '' )
7.          {
8.            $this->csrf_hash = $_COOKIE[$this->csrf_cookie_name];
9.          } else {
10.             $this->csrf_hash = md5(uniqid(rand(), TRUE));
11.         }
12.       }
13.    return $this->csrf_hash;
14. }
```