

Top CodeIgniter Interview Questions and Answers

What is CodeIgniter?

CodeIgniter is easy to use an open-source MVC based framework for PHP. It is a loosely coupled framework that is used for the rapid development of websites and mobile APIs. Here you can read about CodeIgniter helpers, sessions, hooks, Routing, Constants ORM supported by Codeigniter and more.

Q1. Explain what is Codeigniter?

CodeIgniter is a powerful **MVC** based PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications.

Q2. What is the current version of Codeigniter?

As on Sept 19, 2019 **CodeIgniter 3.1.11** is the latest version of the framework. You can download it from [here](#)

Q3. How to check the version of CodeIgniter framework?

In system/core/CodeIgniter.php, check CI_VERSION constant value define('CI_VERSION', '3.0.6');

Q4. List Databases supported By Codeigniter Frameworks?

Following Databases supported are supported by Codeigniter Frameworks

- [MySQL](#) (5.1+) via the MySQL (deprecated), MYSQLI and PDO drivers
- [Oracle](#) via the oci8 and PDO drivers
- PostgreSQL via the Postgre and PDO drivers
- [MS SQL](#) via the MsSQL, Sqlsrv (version 2005 and above only) and PDO drivers
- SQLite via the SQLite (version 2), sqlite3 (version 3) and PDO drivers
- CUBRID via the Cubrid and PDO drivers
- Interbase/Firebird via the iBase and PDO drivers
- ODBC via the ODBC and PDO drivers (you should know that ODBC is actually an abstraction layer)

Q5. List some features provided by CodeIgniter?

- Framework with a small footprint
- Simple solutions over complexity
- Clear documentation
- Exceptional performance
- Strong Security
- Nearly zero configuration

Q6. Explain helpers in CodeIgniter and how to load helper file?

As the name suggests, helpers help you with tasks. Each helper file is simply a collection of functions in a particular category. They are simple, procedural functions. Each helper function performs one specific task, with no dependence on other functions.

CodeIgniter does not load Helper Files by default, so the first step in using a Helper is to load it. Once loaded, it becomes globally available in your controller and views.

Helpers are typically stored in your system/helpers, or application/helpers directory

Loading a helper file is quite simple using the following method:

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

Where name is the file name of the helper, without the .php file extension or the “helper” part.

[Read More](#)

Q7. Explain routing in CodeIgniter?

In Software engineering **routing** is the process of taking a URI endpoint (that part of the URI which comes after the base URL) and decomposing it into parameters to determine which module, controller, and action of that controller should receive the request. In CodeIgniter typically there is a one-to-one relationship between a URL string and its corresponding controller class/method. The segments in a URI normally follow this pattern:

example.com/class/function/id/. In CodeIgniter, all routing rules are defined in your application/config/routes.php file.

Q8. What are hooks in CodeIgniter? List them?

CodeIgniter’s Hooks feature provides a way to modify the inner workings or functionality of the framework without hacking the core files.

The following is a list of available hook points.

- **pre_system** Called very early during system execution.
- **pre_controller** Called immediately prior to any of your controllers being called.

- **post_controller_constructor** Called immediately after your controller is instantiated, but prior to any method calls happening.
- **post_controller** Called immediately after your controller is fully executed.
- **display_override** Overrides the _display() method.
- **cache_override** Enables you to call your own method instead of the _display_cache() method in the Output Library. This permits you to use your own cache display mechanism.
- **post_system** Called after the final rendered page is sent to the browser, at the end of system execution after the finalized data is sent to the browser.

[Read More](#)

Q9. [List Common Functions in Codeigniter?](#)

CodeIgniter uses a few functions for its operation that are globally defined, and are available to you at any point. These do not require loading any libraries or helpers.

- is_php(\$version)
- is_really_writable(\$file)
- config_item(\$key)
- set_status_header(\$code[, \$text = ”])
- remove_invisible_characters(\$str[, \$url_encoded = TRUE])
- html_escape(\$var)
- get_mimes()
- is_https()
- is_cli()

Read Best 80+ [AngularJS Interview Questions](#)

Q10. [How do you set default timezone in Codeigniter ?](#)

To set default timezone in Codeigniter open application/config.php file and add below code in it.

```
date_default_timezone_set('your timezone');
```

Q11. [How to add / link an images/CSS/JavaScript from a view in CI?](#)

In Codeigniter, you can link images/CSS/JavaScript by using the absolute path to your resources.

Something like below

```
// References your $config['base_url']

```

Q12. What is inhibitor are Codeigniter?

An inhibitor is an error handling class in Codeigniter. It uses PHP 's native functions like `register_shutdown_function` , `set_exception_handler`, `set_error_handler` to handle **parse errors**, **exceptions**, and **fatal errors**.

Q13. How can you remove index.php from URL in Codeigniter?

Follow below steps to index.php from URL in Codeigniter

- Step 1: Open config.php and replaces

```
$config['index_page'] = "index.php" to $config['index_page'] = ""  
and  
$config['uri_protocol'] ="AUTO" to $config['uri_protocol'] = "REQUEST_URI"
```

- Step 2: Change your .htaccess file to

```
RewriteEngine on  
RewriteCond $1 !^(index\.php|[Javascript / CSS / Image root Folder name(s)]|  
RewriteRule ^(.*)$ /index.php/$1 [L]
```

Q14. How will you add or load a model in Codeigniter?

In Codeigniter, models will typically be loaded and called from within your controller methods. To load a model you will use the following method:

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

If your model is located in a sub-directory, include the relative path from your model's directory. For example, if you have a model located at `application/models/blog/Posts.php` you'll load it using:

```
$this->load->model('blog/Posts');
```

Once your Model loaded, you will access its methods using an object with the same name as your controller:

Example

```

class Blog_controller extends CI_Controller {
    public function blog()
    {
        $this->load->model('blog');
        $data['query'] = $this->blog->get_last_ten_entries();
        $this->load->view('blog', $data);
    }
}

```

Q15. What are Sessions In CodeIgniter? How to read, write or remove session in CodeIgniter?

Sessions in CodeIgniter

In CodeIgniter Session class allows you maintain a user's "state" and track their activity while they are browsing your website.

In order to use session, you need to load Session class in your controller.

\$this->load->library('session'); method is used to sessions in CodeIgniter

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

Once loaded, the Sessions library object will be available using:

```
$this->session
```

Reading session data in CodeIgniter

Use `$this->session->userdata();` method of session class to read session data in CodeIgniter.

Usage

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

You can also read a session data by using the magic getter of CodeIgniter Session Class

Usage

```
$this->session->item
```

Where an item is the name of the key you want to retrieve.

Creating a session in CodeIgniter

Session's Class `set_userdata()` method is used to create a session in CodeIgniter. This method takes an associative array containing your data that you want to add in session.

Example

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

If you want to add userdata one value at a time, `set_userdata()` also supports this syntax:

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

Removing Session Data

Session's Class `unset_userdata()` method is used to remove a session data in CodeIgniter. Below are example usages of same.

Unset particular key

```
$this->session->unset_userdata('some_key');
```

Unset an array of item keys

```
$array_items = array('username', 'email');
$this->session->unset_userdata($array_items);
```

Read More <https://www.codeigniter.com/userguide3/libraries/sessions.html>

Q16. [How to get last inserted id in CodeIgniter?](#)

CodeIgniter DB Class `insert_id()` method is used to get last insert id.

Usage:

```
function add_post($post_data){
    $this->db->insert('posts', $post_data);
    $insert_id = $this->db->insert_id();
    return $insert_id;
}
```

Q17. [How you will use or load CodeIgniter libraries](#)

`$this->load->library('library_name');` method is used to load a library in CodeIgniter.

Usage:

```
//Loading Cart library
```

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

Using Cart library methods

```
$data = array(
    'id'      => 'sku_9788C',
    'qty'     => 1,
    'price'   => 35.95,
    'name'    => 'T-Shirt',
    'options' => array('Size' => 'L', 'Color' => 'Red')
);
$this->cart->insert($data);
```

Q18. What is the CLI? Why we use CLI in Codeigniter?

CLI is a text-based command-line interface for interacting with computers via a set of commands.

In Codeigniter, we can use CLI for

- Run your **cronjobs** without needing to use **wget** or **curl**
- Make your **cron-jobs** inaccessible from being loaded in the URL by checking the return value of `is_cli()`.
- Make interactive “tasks” that can do things like set permissions, prune cache folders, run backups, etc.
- Helps to integrate Codeigniter with other applications in other languages. For example, a random C++ script could call one command and run code in your models!

Q19. How to do 301 redirects in CodeIgniter?

We can use **redirect** helper to do 301 redirects in Codeigniter.

Syntax :

```
redirect($uri = '', $method = 'auto', $code = NULL)
```

Parameter:

```
$uri (string) - URI string
$method (string) - Redirect method ('auto', 'location' or 'refresh')
$code (string) - HTTP Response code (usually 302 or 303)
```

Return type: **void**

Sample Usage:-

```
redirect('/posts/13', 'New location', 301);
```

Q20. How to check a field or column exists in a table or not in Codeigniter?

Code for Checking a field or column exists or not in a Codeigniter table.

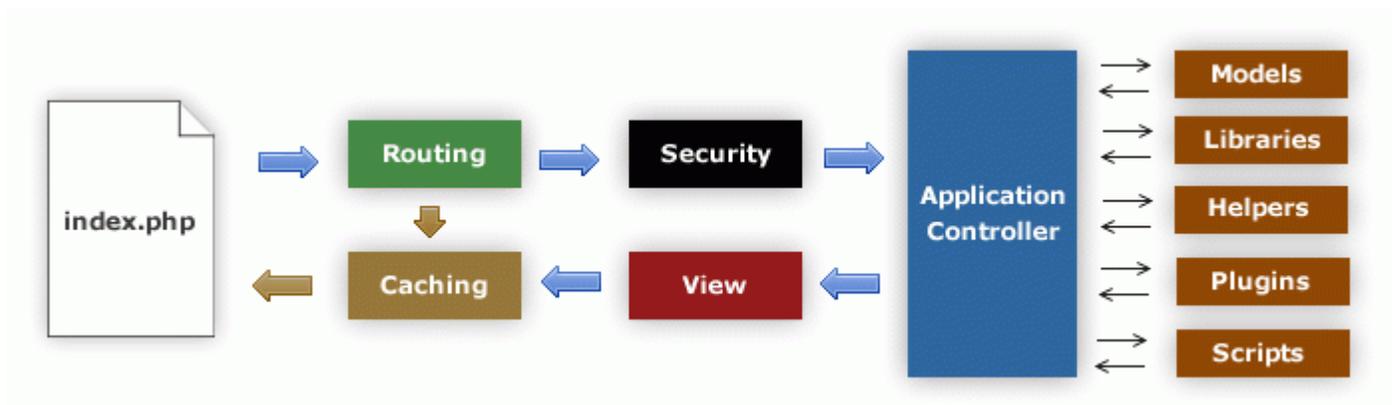
```
if ($this->db->field_exists('field_name', 'table_name'))
{
    // some code...
}
```

Q21. What is cli() method does in Codeigniter?

In Codeigniter `is_cli()` method is used to check request is from the command line or not. Returns TRUE if the application is run through the command line and FALSE if not.

Q22. Explain Codeigniter's application Flow Chart?

Below images give you an understanding of how data flows throughout the system in Codeigniter.



1. The index.php serves as the front controller, initializing the base resources needed to run CodeIgniter.
2. The Router examines the HTTP request to determine what should be done with it.
3. If a cache file exists, it is sent directly to the browser, bypassing the normal system execution.
4. Security. Before the application controller is loaded, the HTTP request and any user submitted data are filtered for security.
5. The Controller loads the model, core libraries, helpers, and any other resources needed to process the specific request.
6. The finalized View is rendered then sent to the web browser to be seen. If caching is enabled, the view is cached first so that on subsequent requests it can be served.

Source: https://www.codeigniter.com/user_guide/overview/appflow.html

Q23. How to set or get config variables in Codeigniter?

In Codeigniter by default all config variables are located at “application/config/config.php” file. Below is the way to set or get a config variable in Codeigniter

```
// Setting a config variable dynamically
$this->config->set_item('variable_name', value);
// Getting value of config variable in Codeigniter.
$this->config->item('variable_name');
```

Q24. How to delete a record in Codeigniter?

In Codeigniter, delete function is used to delete the one or more row data from a table.

```
//DELETE FROM table WHERE id = $id
$conditions = ['id' => $id]
$this->db->delete('table_name', $conditions);
// Deleting records from more than one tables in one go
$id=1;
$tables = array('table1', 'table2', 'table3');
$this->db->where('id', $id);
$this->db->delete($tables);
```

Q25. How to implement validations in Codeigniter?

Q26. What is the default URL pattern used in Codeigniter framework?

In CodeIgniter, URLs are designed to be search-engine and human-friendly. CodeIgniter uses a segment-based approach rather than using “query string” based approach.

abc.com/user/edit/ramesh

The default URL pattern in CodeIgniter consists of 4 main components. They are :

1. A server name (abc.com)
2. A Controller (user)
3. An Action or method (edit)
4. An optional action parameter (ramesh)

Prepare 10 **Essential [Magento Interview Questions](#)**

Q27. How to get random records in MySQL using CodeIgniter?

order_by function is used to order the records from a table in CodeIgniter.

```
// Getting random rows from database in CodeIgniter
$this->db->select('*');
$this->db->from('table_name');
$this->db->order_by("column_name", "random");
$result = $this->db->get()->result();
```

Q28. Why CodeIgniter is called as loosely based MVC framework?

Codeigniter Is Called a loosely based MVC framework, this is because unlike others the controllers' classes such as models and views are not mandatory in CodeIgniter. Moreover, one can modify CodeIgniter to utilize HMVC as well.

Q29. What is an ORM, List ORM's for CodeIgniter?

Object-relational mapping (ORM) is programming technique for converting data between incompatible type systems using object-oriented programming languages. Below is the list of ORM's supported by CodeIgniter Framework

- DataMapper
- Doctrine
- Gas ORM

Q30. Explain URL Helper? Can you list some commonly used URL helpers in Codeigniter?

https://codeigniter-id.github.io/user-guide/helpers/url_helper.html

Q31. List the resources that can be autoloaded in Codeigniter?

The following items can be loaded automatically:

- Classes found in the libraries/ directory
- Helper files found in the helpers/ directory
- Custom config files found in the config/ directory
- Language files found in the system/language/ directory
- Models found in the models/ folder

To autoload resources, open the application/config/autoload.php file and add the item you want loading to the autoload array. You'll find instructions in that file corresponding to each type of item.

Q32. In which files routes are defined in Codeigniter?

All Routing rules in Codeigniter are defined in your application/config/routes.php file.

Q33. In which directory logs are saved in Codeigniter? How to enable error logging in Codeigniter?

By default, all logs in Codeigniter are stored in logs/ directory. To enable error logging you must set the “threshold” for logging in application/config/config.php. Also, your logs/ must be writable.

```
$config['log_threshold'] = 1;
```

Q34. How many types of messages can you log in Codeigniter?

There are three message types in Codeigniter. They are :

- **Error Messages.** These are actual errors, such as PHP errors or user errors.
- **Debug Messages.** These are messages that assist in debugging. For example, if a class has been initialized, you could log this as debugging info.
- **Informational Messages.** These are the lowest priority messages, simply giving information regarding some process.

Q35. How to set csrf token in codeIgniter?

Csrf is used to set the protection in CodeIgniter. To set csrf, you have to put the corresponding config value True.

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