

Extending CodeIgniter with custom libraries and helpers in CodeIgniter

Topics : [Codeigniter](#)

Written on [March 01, 2024](#)

Extending CodeIgniter with custom libraries and helpers allows you to encapsulate reusable functionality and streamline development across your applications. Here's how you can create and use custom libraries and helpers in CodeIgniter:

Creating Custom Libraries:

1. **Create Library Class:** Create a new PHP class file for your custom library in the application/libraries directory. Your class should extend CI_Library and contain the desired functionality.

```
// application/libraries/Custom_library.php
class Custom_library {
    public function __construct() {
        // Constructor logic
    }

    public function some_method() {
        // Method logic
    }
}
```

2. **Load Library:** Load your custom library in your controller or wherever it's needed using the `$this->load->library()` method.

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

3. **Access Library Methods:** Once loaded, you can access methods of your custom library using the library instance.

```
$this->custom_library->some_method();
```

Creating Custom Helpers:

1. **Create Helper Functions:** Create a new PHP file for your custom helper functions in the application/helpers directory. Define your helper functions as you would any other PHP functions.

```
// application/helpers/custom_helper.php function
custom_helper_function() { // Function logic }
```

2. **Load Helper:** Load your custom helper using the `$this->load->helper()` method in your controller or wherever it's needed.

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

3. **Use Helper Functions:** Once loaded, you can use your custom helper functions throughout your application.

```
custom_helper_function();
```

Autoloading Custom Libraries and Helpers:

You can autoload your custom libraries and helpers to make them available globally throughout your application. Add them to the `autoload.php` configuration file located in `application/config/autoload.php`.

```
$autoload['libraries'] = array('custom_library'); $autoload['helper'] =
array('custom_helper');
```

Passing Parameters to Custom Libraries and Helpers:

You can pass parameters to custom library methods and helper functions by adding parameters to the function/method definition and passing them when calling the function/method.

```
// Custom library method
public function some_method($param1, $param2) {
    // Method logic using $param1 and $param2
}

// Custom helper function
function custom_helper_function($param1, $param2) {
    // Function logic using $param1 and $param2
}
```

ARYATECHNO