

# Uploading files in CodeIgniter

Topics : [Codeigniter](#)

Written on [March 01, 2024](#)

In CodeIgniter, you can handle file uploads easily using the built-in file upload library. Here's a step-by-step guide on how to upload files in CodeIgniter:

## 1. Configure File Upload Preferences:

Set up file upload preferences in application/config/config.php:

```
$config['upload_path'] = './uploads/'; // Specify the directory where
uploaded files will be stored
$config['allowed_types'] = 'gif|jpg|png'; // Specify allowed file types
$config['max_size'] = 100; // Specify maximum file size in kilobytes
$config['max_width'] = 1024; // Specify maximum image width
$config['max_height'] = 768; // Specify maximum image height
```

## 2. Create Upload Directory:

Create a directory named "uploads" in your CodeIgniter project root directory. Ensure that it has appropriate permissions for file uploads.

## 3. Create a Form:

Create a form in your view file (upload\_form.php) to allow users to select and upload files:

```
<!-- upload_form.php -->
<form method="post" action="<?php echo base_url('upload/do_upload'); ?>"
enctype="multipart/form-data">
    <input type="file" name="userfile" size="20" />
    <br /><br />
    <input type="submit" value="Upload" />
</form>
```

## 4. Create Controller Method to Handle Upload:

Create a controller (Upload.php) and define a method to handle file uploads:

```
// Upload.php controller
```

```

class Upload extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->helper(array('form', 'url'));
    }

    public function index() {
        $this->load->view('upload_form', array('error' => ' ' ));
    }

    public function do_upload() {
        $config['upload_path'] = './uploads/';
        $config['allowed_types'] = 'gif|jpg|png';
        $config['max_size'] = 100;
        $config['max_width'] = 1024;
        $config['max_height'] = 768;

        $this->load->library('upload', $config);

        if (!$this->upload->do_upload('userfile')) {
            $error = array('error' => $this->upload->display_errors());
            $this->load->view('upload_form', $error);
        } else {
            $data = array('upload_data' => $this->upload->data());
            $this->load->view('upload_success', $data);
        }
    }
}

```

## 5. Handle Upload Success:

Create a view (upload\_success.php) to display success message or handle the uploaded file:

```

<!-- upload_success.php -->
<h3>File uploaded successfully!</h3>
<ul>
    <?php foreach ($upload_data as $item => $value): ?>
        <li><?php echo $item; ?>: <?php echo $value; ?></li>
    <?php endforeach; ?>
</ul>
<p><?php echo anchor('upload', 'Upload Another File!'); ?></p>

```

## 6. Test Your File Upload:

Access the upload form through the browser and test file uploads.