Uploading files

Handling file uploads is a pretty common task for a web application. Yii has some helpful classes built in to  
do this. Let’s create a simple form that will allow the upload of ZIP archives and store them in /uploads.

Getting ready

1. Create a new application by using the Composer package manager, as described in the official  
   guide at [http://www. yiiframework. c om/doc-2.0/guide -start-installation .html](http://www.yiiframework.com/doc-2.0/guide-start-installation.html).
2. Create the @app/web/uploads directory.

How to do it...

1. We will start with the model, so create the @app/models/Upload. php model as follows:

<?php

namespace app\models;  
use yii\base\Model;  
use yii\web\UploadedFile;  
class UploadForm extends Model  
{

/\*\*

\* @var UploadedFile

\*/

public $file;

public function rules()

{

return [

['file', 'file', 'skipOnEmpty' => false, 'extensions' => 'zip'],

];

}

public function upload()

{

if ($this->validate()) {

$this->file->saveAs('uploads/' . $this->file->baseName . '.' . $this-

>file->extension);

return true;

} else {

return false;

}

}

}

1. Now we will move on to the controller, so create @app/controllers/UploadController . php as  
   follows:

<?php

namespace app\controllers;  
use Yii;

use yii\web\Controller;  
use app\models\UploadForm;  
use yii\web\UploadedFile;  
class UploadController extends Controller  
{

public function actionUpload()

{

$model = new UploadForm();  
if (Yii::$app->request->isPost) {

$model->file = UploadedFile::getInstance($model, 'file');  
if ($model->upload()) {

return $this->renderContent("File {$model->file->name} is uploaded

successfully");

}

}

return $this->render('index', ['model' => $model]);

}

}

1. Finally, you can view @app/views/upload/index. php as follows:

<?php

use yii\widgets\ActiveForm;  
use yii\helpers\Html;

?>

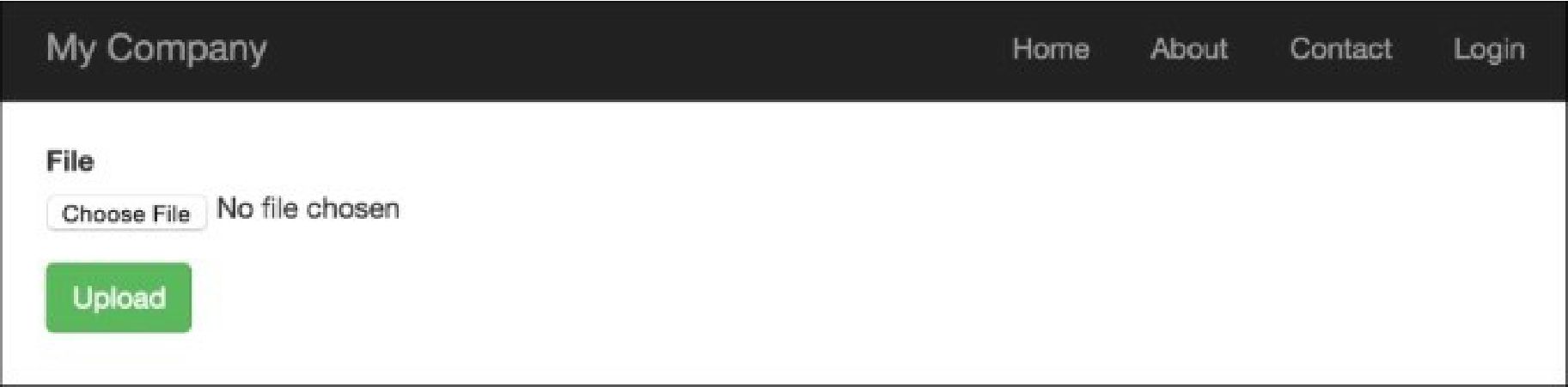
<?php $form = ActiveForm::begin(['options' => ['enctype' => 'multipart/form-  
data']]) ?>

<?= $form->field($model, 'file')->fileInput() ?>

<?= Html::submitButton('Upload', ['class' => 'btn-success'])?>

<?php ActiveForm::end() ?>

1. That is it. Now, run the upload controller and try uploading both ZIP archives and other files, as  
   shown in the following screenshot:

How it works...

The model we use is pretty simple. We define only one field, named $file, and a validation rule that uses  
the FileValidator file validator, which reads only ZIP files.

We create a model instance and fill it with data from $\_post if the form is submitted:

$model->file = UploadedFile::getInstance($model, 'file');  
if ($model->upload()) {

return $this->renderContent("File {$model->file->name} is uploaded successfully");

}

We then use UploadedFile: : getInstance, which gives us access to use the UploadedFile instance. This  
class is a wrapper around the $\_file array that PHP fills when the file is uploaded. We make sure that  
the file is a ZIP archive by calling the model’s validate method, then we save the file using  
UploadedFile::saveAs.

In order to upload a file, the HTML form must meet the following two important requirements:

* Method must be set to post
* The enctype attribute must be set to multipart/form-data

It is important to remember that you add the enctype option to the form so that the file can be properly  
uploaded.

We can generate this HTML using the Html helper or ActiveForm with htmlOptions set. Here, HTML  
was used:

<?= Html::beginForm('', 'post', ['enctype'=>'multipart/form-data'])?>

In the end, we display an error and a field for the model’s file attribute, and render a submit button.

There’s more...

To upload multiple files, Yii2 implements two special methods.

For instance, you have defined $imageFiles in your model in the view file in common all will be the same  
with a little difference:

<?= $form->field($model, 'imageFiles[]')->fileInput(['multiple' => true, 'accept' =>

'image/\*']) ?>

To get all file instances, you have to call UploadedFile : : getInstances() instead of  
UploadedFile::getInstance():

$model->imageFiles = UploadedFile::getInstances($model, 'imageFiles');

Handling and saving multiple files can be done with a simple code snippet:

foreach ($this->imageFiles as $file) {

$file->saveAs('uploads/' . $file->baseName . '.' . $file->extension);

}

See also

For further information, refer to:

* <http://www.yiiframework.com/doc-2.0/guide-input-file-upload.html>
* [http://www.yiiframework.com/doc-2.0/guide-input-file-upload.html#uploadjng-multiple-files](http://www.yiiframework.com/doc-2.0/guide-input-file-upload.html%23uploading-multiple-files)