**Zend Forms**

Adding new albums

Create the file module/Album/src/Form/AlbumForm.php with the following contents:

namespace Album\Form;

use Zend\Form\Form;

class AlbumForm extends Form

{

public function \_\_construct($name = null)

{

// We will ignore the name provided to the constructor

parent::\_\_construct('album');

$this->add([

'name' => 'id',

'type' => 'hidden',

]);

$this->add([

'name' => 'title',

'type' => 'text',

'options' => [

'label' => 'Title',

],

]);

$this->add([

'name' => 'artist',

'type' => 'text',

'options' => [

'label' => 'Artist',

],

]);

$this->add([

'name' => 'submit',

'type' => 'submit',

'attributes' => [

'value' => 'Go',

'id' => 'submitbutton',

],

]);

}

}

module/Album/src/Model/Album.php:

namespace Album\Model;

// Add the following import statements:

use DomainException;

use Zend\Filter\StringTrim;

use Zend\Filter\StripTags;

use Zend\Filter\ToInt;

use Zend\InputFilter\InputFilter;

use Zend\InputFilter\InputFilterAwareInterface;

use Zend\InputFilter\InputFilterInterface;

use Zend\Validator\StringLength;

class Album implements InputFilterAwareInterface

{

public $id;

public $artist;

public $title;

// Add this property:

private $inputFilter;

public function exchangeArray(array $data)

{

$this->id = !empty($data['id']) ? $data['id'] : null;

$this->artist = !empty($data['artist']) ? $data['artist'] : null;

$this->title = !empty($data['title']) ? $data['title'] : null;

}

/\* Add the following methods: \*/

public function setInputFilter(InputFilterInterface $inputFilter)

{

throw new DomainException(sprintf(

'%s does not allow injection of an alternate input filter',

\_\_CLASS\_\_

));

}

public function getInputFilter()

{

if ($this->inputFilter) {

return $this->inputFilter;

}

$inputFilter = new InputFilter();

$inputFilter->add([

'name' => 'id',

'required' => true,

'filters' => [

['name' => ToInt::class],

],

]);

$inputFilter->add([

'name' => 'artist',

'required' => true,

'filters' => [

['name' => StripTags::class],

['name' => StringTrim::class],

],

'validators' => [

[

'name' => StringLength::class,

'options' => [

'encoding' => 'UTF-8',

'min' => 1,

'max' => 100,

],

],

],

]);

$inputFilter->add([

'name' => 'title',

'required' => true,

'filters' => [

['name' => StripTags::class],

['name' => StringTrim::class],

],

'validators' => [

[

'name' => StringLength::class,

'options' => [

'encoding' => 'UTF-8',

'min' => 1,

'max' => 100,

],

],

],

]);

$this->inputFilter = $inputFilter;

return $this->inputFilter;

}

}

module/Album/src/Controller/AlbumController.php:

// Add the following import statements at the top of the file:

use Album\Form\AlbumForm;

use Album\Model\Album;

class AlbumController extends AbstractActionController

{

/\* ... \*/

/\* Update the following method to read as follows: \*/

public function addAction()

{

$form = new AlbumForm();

$form->get('submit')->setValue('Add');

$request = $this->getRequest();

if (! $request->isPost()) {

return ['form' => $form];

}

$album = new Album();

$form->setInputFilter($album->getInputFilter());

$form->setData($request->getPost());

if (! $form->isValid()) {

return ['form' => $form];

}

$album->exchangeArray($form->getData());

$this->table->saveAlbum($album);

return $this->redirect()->toRoute('album');

}

/\* ... \*/

}

module/Album/view/album/album/add.phtml (without CSS)

**<?php**

//

$title = 'Add new album';

$this->headTitle($title);

**?>**

<h1>**<?=** $this->escapeHtml($title) **?>**</h1>

**<?php**

$form->setAttribute('action', $this->url('album', ['action' => 'add']));

$form->prepare();

echo $this->form()->openTag($form);

echo $this->formHidden($form->get('id'));

echo $this->formRow($form->get('title'));

echo $this->formRow($form->get('artist'));

echo $this->formSubmit($form->get('submit'));

echo $this->form()->closeTag();

module/Album/view/album/album/add.phtml (with CSS)

**<?php**

$title = 'Add new album';

$this->headTitle($title);

**?>**

<h1>**<?=** $this->escapeHtml($title) **?>**</h1>

**<?php**

// This provides a default CSS class and placeholder text for the title element:

$album = $form->get('title');

$album->setAttribute('class', 'form-control');

$album->setAttribute('placeholder', 'Album title');

// This provides a default CSS class and placeholder text for the artist element:

$artist = $form->get('artist');

$artist->setAttribute('class', 'form-control');

$artist->setAttribute('placeholder', 'Artist');

// This provides CSS classes for the submit button:

$submit = $form->get('submit');

$submit->setAttribute('class', 'btn btn-primary');

$form->setAttribute('action', $this->url('album', ['action' => 'add']));

$form->prepare();

echo $this->form()->openTag($form);

**?>**

**<?php** // Wrap the elements in divs marked as form groups, and render the

// label, element, and errors separately within ?>

<div class="form-group">

**<?=** $this->formLabel($album) **?>**

**<?=** $this->formElement($album) **?>**

**<?=** $this->formElementErrors()->render($album, ['class' => 'help-block']) **?>**

</div>

<div class="form-group">

**<?=** $this->formLabel($artist) **?>**

**<?=** $this->formElement($artist) **?>**

**<?=** $this->formElementErrors()->render($artist, ['class' => 'help-block']) **?>**

</div>

**<?php**

echo $this->formSubmit($submit);

echo $this->formHidden($form->get('id'));

echo $this->form()->closeTag();

## Editing an album

module/Album/src/Controller/AlbumController.php

// module/Album/src/Controller/AlbumController.php:

// ...

public function editAction()

{

$id = (int) $this->params()->fromRoute('id', 0);

if (0 === $id) {

return $this->redirect()->toRoute('album', ['action' => 'add']);

}

// Retrieve the album with the specified id. Doing so raises

// an exception if the album is not found, which should result

// in redirecting to the landing page.

try {

$album = $this->table->getAlbum($id);

} catch (\Exception $e) {

return $this->redirect()->toRoute('album', ['action' => 'index']);

}

$form = new AlbumForm();

$form->bind($album);

$form->get('submit')->setAttribute('value', 'Edit');

$request = $this->getRequest();

$viewData = ['id' => $id, 'form' => $form];

if (! $request->isPost()) {

return $viewData;

}

$form->setInputFilter($album->getInputFilter());

$form->setData($request->getPost());

if (! $form->isValid()) {

return $viewData;

}

$this->table->saveAlbum($album);

// Redirect to album list

return $this->redirect()->toRoute('album', ['action' => 'index']);

}

// Add the following method in module/Album/src/Model/Album.php

public function getArrayCopy()

{

return [

'id' => $this->id,

'artist' => $this->artist,

'title' => $this->title,

];

}

module/Album/view/album/album/edit.phtml

**<?php**

// module/Album/view/album/album/edit.phtml:

$title = 'Edit album';

$this->headTitle($title);

**?>**

<h1>**<?=** $this->escapeHtml($title) **?>**</h1>

**<?php**

$album = $form->get('title');

$album->setAttribute('class', 'form-control');

$album->setAttribute('placeholder', 'Album title');

$artist = $form->get('artist');

$artist->setAttribute('class', 'form-control');

$artist->setAttribute('placeholder', 'Artist');

$submit = $form->get('submit');

$submit->setAttribute('class', 'btn btn-primary');

$form->setAttribute('action', $this->url('album', [

'action' => 'edit',

'id' => $id,

]));

$form->prepare();

echo $this->form()->openTag($form);

**?>**

<div class="form-group">

**<?=** $this->formLabel($album) **?>**

**<?=** $this->formElement($album) **?>**

**<?=** $this->formElementErrors()->render($album, ['class' => 'help-block']) **?>**

</div>

<div class="form-group">

**<?=** $this->formLabel($artist) **?>**

**<?=** $this->formElement($artist) **?>**

**<?=** $this->formElementErrors()->render($artist, ['class' => 'help-block']) **?>**

</div>

**<?php**

echo $this->formSubmit($submit);

echo $this->formHidden($form->get('id'));

echo $this->form()->closeTag();

## Deleting an album

module/Album/src/Album/Controller/AlbumController.php

// module/Album/src/Album/Controller/AlbumController.php:

//...

// Add content to the following method:

public function deleteAction()

{

$id = (int) $this->params()->fromRoute('id', 0);

if (!$id) {

return $this->redirect()->toRoute('album');

}

$request = $this->getRequest();

if ($request->isPost()) {

$del = $request->getPost('del', 'No');

if ($del == 'Yes') {

$id = (int) $request->getPost('id');

$this->table->deleteAlbum($id);

}

// Redirect to list of albums

return $this->redirect()->toRoute('album');

}

return [

'id' => $id,

'album' => $this->table->getAlbum($id),

];

}

//...

module/Album/view/album/album/delete.phtml

**<?php**

// module/Album/view/album/album/delete.phtml:

$title = 'Delete album';

$url = $this->url('album', ['action' => 'delete', 'id' => $id]);

$this->headTitle($title);

**?>**

<h1>**<?=** $this->escapeHtml($title) **?>**</h1>

<p>

Are you sure that you want to delete

"**<?=** $this->escapeHtml($album->title) **?>**" by

"**<?=** $this->escapeHtml($album->artist) **?>**"?

</p>

<form action="**<?=** $url **?>**" method="post">

<div class="form-group">

<input type="hidden" name="id" value="**<?=** (int) $album->id **?>**" />

<input type="submit" class="btn btn-danger" name="del" value="Yes" />

<input type="submit" class="btn btn-success" name="del" value="No" />

</div>

</form>