



Module Management

Module Structure

A Zend Framework module typically includes the following directories:

- `config/`: Configuration files (module.config.php)
- `src/`: Source code (controllers, models, forms, etc.)
- `view/`: View scripts
- `Module.php`: Module class

Module Configuration

<code>module.config.php</code>	Main configuration file for the module. Defines routes, controllers, services, and view configurations.
<code>Module.php</code>	Module class that defines the <code>getConfig()</code> method to merge module-specific configurations.
Loading Modules	Modules are typically loaded in the <code>config/application.config.php</code> file under the <code>modules</code> key.

Example: module.config.php

```
return [
    'controllers' => [
        'factories' => [
            'MyModule\Controller\Index'
            =>
            'MyModule\Factory\IndexControllerFactory',
        ],
    ],
    'router' => [
        'routes' => [
            'mymodule' => [
                'type' => 'Literal',
                'options' => [
                    'route' =>
                    '/mymodule',
                    'defaults' => [
                        'controller' =>
                        'MyModule\Controller\Index',
                        'action' =>
                        'index',
                    ],
                ],
            ],
        ],
    ],
    'view_manager' => [
        'template_path_stack' => [
            'mymodule' => __DIR__ .
            '/../view',
        ],
    ],
];
```

Controllers and Actions

Controller Basics

Controllers handle incoming requests and return responses.

- Extend `AbstractActionController`.
- Define action methods (e.g., `indexAction()`).
- Retrieve request parameters using `$this->params()`.

Action Methods

<code>indexAction()</code>	Default action method. Typically displays a list of resources or a homepage.
<code>createAction()</code>	Handles the creation of a new resource, typically via a form submission.
<code>updateAction()</code>	Handles the updating of an existing resource, typically via a form submission.
<code>deleteAction()</code>	Handles the deletion of a resource.

Example: Controller

```
namespace MyModule\Controller;

use
Zend\Mvc\Controller\AbstractActionController;
use Zend\View\Model\ViewModel;

class IndexController extends
AbstractActionController
{
    public function indexAction()
    {
        return new ViewModel();
    }
}
```

View Layer

View Scripts

View scripts are PHP files that generate the HTML output.

- Located in the `view/` directory of a module.
- Use the `.phtml` extension.
- Access variables passed from the controller.

View Helpers

<code>\$this->url()</code>	Generates a URL based on a route name.
<code>\$this->form()</code>	Renders a form.
<code>\$this->escapeHtml()</code>	Escapes HTML entities in a string.
<code>\$this->translate()</code>	Translates a string using the translation service.

Example: View Script

```
<h1>Welcome!</h1>

<p>This is a sample view script.</p>

<a href="<?php echo $this->url('mymodule'); ?>">MyModule Home</a>
```

Layouts

Layouts provide a consistent structure for multiple pages. Configure the layout in the `view_manager` section of your

`module.config.php` or `application.config.php`.

Set the default layout with `'template' => 'layout/layout'`.

Forms and Input Filters

Form Basics

Forms are used to handle user input.

- Extend `Zend\Form\Form`.
- Add elements with appropriate input types.
- Validate input using input filters.

Example: Form

```
namespace MyModule\Form;

use Zend\Form\Form;
use Zend\Form\Element;

class MyForm extends Form
{
    public function __construct($name = null, $options = [])
    {
        parent::__construct($name, $options);

        $this->add([new Element\Text('username')]);
        $this->add([new Element\Email('email')]);
        $this->add([new Element\Submit('submit', ['label' => 'Submit'])]);
    }
}
```

Input Filters

<code>Zend\InputFilter\InputFilter</code>	Used to define validation rules for form elements.
Validators	Examples: <code>StringLength</code> , <code>EmailAddress</code> , <code>NotEmpty</code>
Filters	Examples: <code>StringTrim</code> , <code>StripTags</code> , <code>ToInt</code>