

 免費電子書

學習

# zend-framework2

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#zend-

framework2

.....	1
<b>1: zend-framework2</b> .....	<b>2</b>
.....	2
Examples.....	2
.....	2
Composer - .....	2
Git.....	3
<b>HTTP</b> .....	<b>3</b>
1 - PHP CLI.....	3
2 - HTTP.....	3
Hello World.....	3
.....	4
<b>2: Zend Expressive</b> .....	<b>6</b>
Examples.....	6
Hello World.....	6
<b>3:</b> .....	<b>9</b>
Examples.....	9
composergithub.....	9
.....	10

---

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [zend-framework2](#)

It is an unofficial and free zend-framework2 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official zend-framework2.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# 1: zend-framework2

Zend Framework 2 ZF2 PHP Web Web。 Zend Framework [Zend Technologies](#)。 ZF1。 。 Web。 ZF2 PHP 5.3 +。

```
class IndexController extends Zend_Controller_Action
{
}
}
```

## ZF2

```
namespace Application\Controller;
use Zend\Mvc\Controller\AbstractActionController;

class IndexController extends AbstractActionController
{
}
}
```

## ZF2

- 
- 
- 

## Examples

Zend Framework 2。 。

## Composer -

composer 。

## MVC zend framework 2

```
php composer.phar create-project -sdev \
--repository-url="https://packages.zendframework.com" \
zendframework/skeleton-application path/to/install
```

## ZF2 Zend MVC +

```
composer require zendframework/zend-mvc
```

## ZF2+64

```
composer require zendframework/zendframework`
```

- ZF2◦

## Git

### Githubzf2

```
git clone git://github.com/zendframework/ZendSkeletonApplication.git --recursive
```

## HTTP

### WebHTTP80◦

Zend Framework 2HTTP◦

## 1 - PHP CLI

### PHP 5.4PHP cli-server◦

```
php -S 0.0.0.0:8080 -t public/ public/index.php`.
```

### 8080cli-server◦

## 2 - HTTP

### Apache Microsoft IIS Server NginxHTTP◦

### Hello World

```
composer create-project zendframework/skeleton-application helloWorldTest ◦ “”◦
```

PHP CLI◦ helloWorldTest php -S 0.0.0.0:8080 -t public/ public/index.php ◦ Web[http:// localhost /](http://localhost/)  
ZF2 Skeleton Application◦

- module/Application/config/module.config.php

```
return [  
    'router' => [  
        'routes' => [  
            'home' => [  
                ...  
            ],  
        ],  
        'application' => [  
            'type' => Segment::class,  
            'options' => [  
                'route' => '/application[:action]',  
                'defaults' => [  
                    ...  
                ],  
            ],  
        ],  
    ],  
];
```



```

use Zend\ServiceManager\FactoryInterface;
use Zend\ServiceManager\ServiceLocatorInterface;

class MyClassFactory implements FactoryInterface
{
    public function createService(ServiceLocatorInterface $serviceLocator)
    {
        $config = $serviceLocator->get('Config');
        $dependency = $config['dependency'];
        $myClass = new MyClass($dependency);
        return $myClass;
    }
}

```

module.config.php<sup>o</sup>

```

<?php

namespace Application;

return array(
    //...
    'service_manager' => [
        'factories' => [
            'Application\Folder\MyClass' => 'Application\Factory\MyClassFactory'
        ]
    ],
    //...
);

```

```

<?php

namespace Application;

use Application\Folder\MyClass;
use Application\Factory\MyClassFactory;

return array(
    //...
    'service_manager' => [
        'factories' => [
            MyClass::class => MyClassFactory::class
        ]
    ],
    //...
);

```

```
$serviceManager->get('Application\Folder\MyClass');
```

```
$serviceManager->get(MyClass::class);
```

o

[zend-framework2](https://riptutorial.com/zh-TW/zend-framework2/topic/1304/zend-framework2) <https://riptutorial.com/zh-TW/zend-framework2/topic/1304/zend-framework2>

# 2: Zend Expressive

## Examples

### Hello World

`composer create-project zendframework/zend-expressive-skeleton expressive-skeleton`

◦

1. `no` ;
2. `1` ;
3. `Zend ServiceManager` ;
4. `Zend View` ;
5. `Whoops` ;

`expressive-skeleton PHP CLI php -S 0.0.0.0:8080 -t public public/index.php` [http:// localhost 8080 /](http://localhost:8080/)

◦ `config/autoload/routes.global.php`

```
<?php

return [
    'dependencies' => [
        ...
    ],

    'routes' => [
        [
            'dependencies' => [
                'invokables' => [
                    ...
                ],
                'factories' => [
                    ...
                    // Add the following line
                    App\Action\HelloWorldAction::class => App\Action\HelloWorldFactory::class,
                ],
            ],
        ],
        // Following lines should be added
        [
            'name' => 'hello-world',
            'path' => '/hello-world',
            'middleware' => App\Action\HelloWorldAction::class,
            'allowed_methods' => ['GET'],
        ],
    ],
];
```

`src/App/Action/HelloWorldFactory.php`



```

<?php

namespace App\Action;

use Interop\Container\ContainerInterface;
use Zend\Expressive\Template\RendererInterface;

class HelloWorldFactory
{
    public function __invoke(ContainerInterface $container)
    {
        $template = ($container->has(RendererInterface::class))
            ? $container->get(RendererInterface::class)
            : null;

        return new HelloWorldAction($template);
    }
}

```

src/App/Action/HelloWorldAction.php

```

<?php

namespace App\Action;

use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;
use Zend\Diactoros\Response\HtmlResponse;
use Zend\Diactoros\Response\JsonResponse;
use Zend\Expressive\Template;
use Zend\Expressive\Plates\PlatesRenderer;
use Zend\Expressive\Twig\TwigRenderer;
use Zend\Expressive\ZendView\ZendViewRenderer;

class HelloWorldAction
{
    private $template;

    public function __construct(RendererInterface $template = null)
    {
        $this->template = $template;
    }

    public function __invoke(ServerRequestInterface $request, ResponseInterface $response,
        callable $next = null)
    {
        $data = [];

        return new HtmlResponse($this->template->render('app::hello-world'));
    }
}

```

templates/app/hello-world.phtml

```

<?php echo 'Hello World'; ?>

```

<http://localhost8080/hello-world> Zend Expressive“hi”



---

3:

## Examples

### composergithub

```
1 cd my/project/dir
2 git clone git://github.com/zendframework/ZendSkeletonApplication.git
3 cd ZendSkeletonApplication
4 php composer.phar self-update
5 php composer.phar install
```

<https://riptutorial.com/zh-TW/zend-framework2/topic/6458/>

---

S. No		Contributors
1	zend-framework2	<a href="#">Community</a> , <a href="#">edigu</a> , <a href="#">Hassan</a> , <a href="#">Sanjeev kumar</a> , <a href="#">Shirraz</a> , <a href="#">Wilt</a>
2	Zend Expressive	<a href="#">Shirraz</a>
3		<a href="#">edigu</a> , <a href="#">Waqar Haider</a>