無料電子ブック

学習

zend-framework2

Free unaffiliated eBook created from
Stack Overflow contributors.

#zend-framework2

1
1: zend-framework2 2
2
Examples 2
2
- 2
Git 3
HTTP 3
1 - PHP CLI 3
2 - HTTP 3
Hello World 4
5
2: Zend Expressive 7
Examples 7
Hello World 7
3: 10
Examples 10
github 10
11

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

1: zend-framework2をいめる

Zend Framework 2ZF2は、WebがさまざまなさのwebアプリケーションをできるようにするのなPHPフレームワークです。 Zend Frameworkのスポンサーは[Zend Technologies]です.[Zend Technologies]はにくしています。 ZF1としてこの2のバージョンには2つのきながあります。まず、モジュールベースのアーキテクチャがデフォルトでされました。これはモジュールへのをとするきなサイズのWebアプリケーションをするにです。に、ZF2は、にをできるPHP5.3 +のすべてのをしています。のバージョンでは、コントローラクラスのはのようになっています。

```
class IndexController extends Zend_Controller_Action
{

}
```

このじクラスは、のようにZF2にきされています。

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

class IndexController extends AbstractActionController
{

}
```

はZF2ののエキサイティングなです

-
- イベントマネージャ
- サービスマネージャ

Examples

インストールまたはセットアップ

Zend Framework 2をセットアップまたはインストールするためのな。フレームワークをインストールするにはさまざまながあります。はそれらのです

の - される

すると、 `composer`されたインストールターゲットボックスに。

スケルトンMVCアプリケーションをインストールするには、でして、されたにしいzend framework 2プロジェクトをします。

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

```
    zendframework/skeleton-application path/to/install
```

の ZF2Zend MVC +のいくつかのをでインストーールするには、コマンドラインでします。

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

またはフルてるさ d ZF2+64モジューールのために

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

のオプションは、のアプリケーーションディレクトリとともに、にするアプリケーーションをするインストーーラをすることにしてください。のオプションをすると、アプリケーーションをからすることができます。これは、するZF2モジューールをするだけです。

Gitサブモジューールの

のコマンドをしてzf2をクローーンし、それはGithubからにします。

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

HTTPサーーバーのセットアップ

なWebアプリケーーションでは、をアプリケーーションにし、をしてするポートは80をするHTTPサーービスをするがあります。

> Zend Framework 2では、HTTPサーーバをとせずに、コンソーールアプリケーーションをすることもできます。

オプション1 - PHP CLIサーーバ

PHP 5.4をしている、もなは、ルートディレクトリでPHPクライアントサーーバをすることです。

プロジェクトディレクトリにし、のコマンドをします。

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

これで、ポート`8080`でみみのcli-serverがし、すべてのネットワークインターーフェイスにバインドされます。

オプション2 - カスタムHTTPサーーバー

Apache または Microsoft IIS Server または Nginxでホストをし、HTTPリクエストをアプリケーーシ

ョンにします。

シンプルなHello World

コマンドラインで、プロジェクトをするディレクトリにし、のようにします `composer create-project zendframework/skeleton-application helloWorldTest` 。 インストーールに、のインストーールがかどうかをねられます。ここでyesとえましょう。ただテストしています。

にするために、みみのPHP CLIサーバーをします。コマンドラインから、のプロジェクトのルートディレクトリ `helloWorldTest` に`helloWorldTest` 、 `php -S 0.0.0.0:8080 -t public/ public/index.php` `helloWorldTest` `php -S 0.0.0.0:8080 -t public/ public/index.php`ます。 Webブラウザをいてhttp// localhost /にすると、ZF2スケルトンアプリケーションのようこそページがされます。

そうしたら、しいページをセットアップします。 `module/Application/config/module.config.php`では、アプリケーションのサブフォルダのルートがにされていることがわかります。

```
return [
    'router' => [
        'routes' => [
            'home' => [
                ...
            ],
            'application' => [
                'type'    => Segment::class,
                'options' => [
                    'route'    => '/application[/:action]',
                    'defaults' => [
                        'controller'    => Controller\IndexController::class,
                        'action'        => 'index',
                    ],
                ],
            ],
        ],
    ],
```

`module/Applicaiton/src/Controller/IndexController.php`しいアクション" `helloWorldAction()` "をして `module/Applicaiton/src/Controller/IndexController.php`

```
class IndexController extends AbstractActionController
{
    public function indexAction()
    {
        ...
    }

    public function helloWorldAction()
    {
        return new ViewModel();
    }
}
```

に、ののビューファイル`module/Application/view/application/index/hello-world.phtml`をします。

```
<?php
echo "Hello World !";
```

さて、 http// localhost / application / hello-worldにき、ZF2にこんにちは

をる

クラスにハードをたせるがある、そのがファクトリをしてされるコンストラクタパターンをすることがベストプラクティスです。

`MyClass`がアプリケーションのからするがある`$dependency`しているとしましょう。

```
<?php
namespace Application\Folder;

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

class MyClass
{
    protected $dependency;

    public function __construct($dependency)
    {
        $this->dependency = $dependency;
    }
}
```

このをするには、ファクトリクラスがされます。このファクトリは、configからのをし、クラスのにをし、をします

```
<?php
namespace Application\Factory;

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;
    }
}
```

ファクトリクラスがされたので、キーファクトリのにあるモジュールファイル`module.config.php`のサービスマネージャコンフィグ`module.config.php`にするがあり`module.config.php`。 クラスとファクトリのにじをすることをおめします。そのため、プロジェクトフォルダツリーでにをつけることができます。

```
<?php

namespace Application;

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

あるいは、クラスをしてクラスをすることもできます。

```
<?php

namespace Application;

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

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

クラスは、そのクラスのファクトリをするにしたキーをして、サービスマネージャーでできます。

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

または

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

サービスマネージャは、ファクトリをつけてしてし、がされたクラスインスタンスをします。

オンラインでzend-framework2をいめるをむ https://riptutorial.com/ja/zend-framework2/topic/1304/zend-framework2をいめる

2: Zend Expressive

Examples

シンプルなHello World

コンポーザーをして、アプリケーションがインストールされるディレクトリで、のコマンドをします `composer create-project zendframework/zend-expressive-skeleton expressive-skeleton`。

インストールプロセスに、さまざまなをするようめられます。

1. デフォルトのインストールのについては、no [n] とえてください。
2. ルータの、Aura Router [1] をしてみましょう。
3. コンテナについては、Zend ServiceManager [3] をしてみましょう。
4. テンプレートについては、Zend View [3] をってみましょう。
5. に、エラーハンドラについては、Whoops [1] をしましょう。

インストールがしたら、ルートディレクトリである`expressive-skeleton`にり、みみのPHP CLIサーバーをします `php -S 0.0.0.0:8080 -t public public/index.php`。 ブラウザでhttp// localhost8080 /にすると、アプリケーションがしてするはずです。

しいミドルウェアへのしいパスをしましょう。まず、 `config/autoload/routes.global.php`ルータのファイルをき、のをします

```
<?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`のコンテンツをれてください

```
<?php

namespace App\Action;

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

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

        return new HelloWorldAction($template);
    }
}
```

に、 `src/App/Action/HelloWorldAction.php`このコンテンツ

```
<?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(Template\TemplateRendererInterface $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'));
    }
}
```

に、のように`templates/app/hello-world.phtml`

```
<?php echo 'Hello World'; ?>
```

たちはです http// localhost8080 / hello-worldにし 、Zend Expressiveに "hi"とえてください

オンラインでZend Expressiveをむ https://riptutorial.com/ja/zend-framework2/topic/6109/zend-expressive

3: インストール

Examples

でインストールするgithub

```
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
```

オンラインでインストールをむ https://riptutorial.com/ja/zend-framework2/topic/6458/インストール

クレジット

S. No		Contributors
1	zend-framework2をいめる	Community, edigu, Hassan, Sanjeev kumar, Shirraz, Wilt
2	Zend Expressive	Shirraz
3	インストール	edigu, Waqar Haider