Использование нескольких конфигураций для упрощения развертывания

Шаблон расширенного приложения использует различные файлы конфигурации для каждого из своих приложений:

common

config

main . php  
main-local.php  
params.php  
params-local.php

console

config

main . php  
main-local.php  
params.php  
params-local.php

backend

config

main . php  
main-local.php  
params.php  
params-local.php

frontend

config

main . php  
main-local.php  
params.php  
params-local.php

Каждая запись скрипта web/index.php объединяет собственный набор конфигурационных файлов:

$config = yii\helpers\ArrayHelper::merge(

require(\_\_DIR\_\_ .'/../../common/config/main.php'),

require(\_\_DIR\_\_ .'/../../common/config/main-local.php'),

require(\_\_DIR\_\_ .'/../config/main.php'),

require(\_\_DIR\_\_ .'/../config/main-local.php')

);

$application = new yii\web\Application($config);

$application->run();

Каждый файл config/main.php объединяет параметры:

<?php

$params = array\_merge(

require(\_\_DIR\_\_ .'/../../common/config/params.php'),

require(\_\_DIR\_\_ .'/../../common/config/params-local.php'),

require(\_\_DIR\_\_ .'/params.php'),

require(\_\_DIR\_\_ .'/params-local.php')

);

return [

// ...

'params' => $params,

];

Эта система позволяет настроить как общие, так и специфические свойства приложений и компонентов приложений. И мы можем хранить файлы конфигурации по умолчанию в системе управления версиями и игнорировать все \* - local.php -файлы.

Все локальные шаблоны файлов подготавливаются в каталоге среды. При запуске php init в консоли и выборе среды работы этот сценарий инициализации создает копии соответствующих файлов и помещает их в целевые папки.

Но базовый шаблон приложения не содержит гибкой системы конфигурации и предоставляет только следующие файлы:

config

console.php  
web.php  
db.php  
params.php

Попробуем добавить расширенную систему настройки в шаблон приложения yii2-app-basic.

Подготовка

Создайте новое yii2-app-basic приложение с помощью диспетчера пакетов Composer, как описано в официальном руководстве по адресу  
<http://www.yiiframework.com/doc-2.0/guide-start-installation.html>.   
По русски <http://yiiframework.domain-na.me/doc/guide/2.0/ru/start-installation>

Как это сделать...

1. Создайте config/common.php файл:

<?php

$params = array\_merge(

require(\_\_DIR\_\_ . '/params.php'),

require(\_\_DIR\_\_ . '/params-local.php')

);

return [

'basePath' => dirname(\_\_DIR\_\_),

'components' => [

'cache' => [

'class' => 'yii\caching\FileCache',

],

'mailer' => [

'class' => 'yii\swiftmailer\Mailer',

],

'db' => [],

],

'params' => $params,

];

1. Создайте файл config/common-local.php:

<?php  
return [

'components' => [

'db' => [

'class' => 'yii\db\Connection',

'dsn' => 'mysql:host=localhost;dbname=yii2basic',

'username' => 'root',

'password' => '',

' charset' => ' utf8',

],

'mailer' => [

'useFileTransport' => true,

],

],

];

1. Удалите config/db.php файл.
2. Удалите повторяющийся код из config/console.php:

<?php

Yii::setAlias('@tests', dirname(\_\_DIR\_\_) . '/tests');

return [

'id' => 'basic-console',

'bootstrap' => ['log', 'gii'],

'controllerNamespace' => 'app\commands',

'modules' => [

'gii' => 'yii\gii\Module',

],

'components' => [

'log' => [

'targets' => [

[

'class' => 'yii\log\FileTarget',  
'levels' => ['error', 'warning'],

],

],

],

],

];

1. Создайте config/console-local.php файл с пустым массивом:

<?php  
return [

];

1. Изменить config/web.php файл:

$config = [

'id' => 'basic',

'bootstrap' => ['log'],

'components' => [

'user' => [

'identityClass' => 'app\models\User',  
'enableAutoLogin' => true,

],

'errorHandler' => [

'errorAction' => 'site/error',

],

'log' => [

'traceLevel' => YII\_DEBUG ? 3 : 0,

'targets' => [

[

'class' => 'yii\log\FileTarget',  
'levels' => ['error', 'warning'],

],

],

],

],

];

if (YII\_ENV\_DEV) {

// configuration adjustments for 'dev' environment  
$config['bootstrap'][] = 'debug';  
$config['modules']['debug'] = 'yii\debug\Module';

$config['bootstrap'][] = 'gii';  
$config['modules']['gii'] = 'yii\gii\Module';

}

return $config;

1. Переместите конфигурацию запроса в config/web-local.php:

<?php  
return [

'components' => [

'request' => [

'cookieValidationKey' => 'TRk9G1La5kvLFwqMEQTp6PmC1NHdjtkq',

],

],

];

1. Удалите идентификатор электронной почты из config/params.php:

<?php  
return [

'adminEmail' => '',

];

1. Вставьте ID в config/params-local.php:

<?php  
return [

'adminEmail' => 'admin@example.com',

];

1. Удалите строку dsn из tests/codeception/config/config.php:

<?php

/\*\*

\* Application configuration shared by all test types  
\*/

return [

'controllerMap' => [

// ...

],

'components' => [

'db' => [

'dsn' => '',

],

'mailer' => [

'useFileTransport' => true,

],

'urlManager' => [

'showScriptName' => true,

],

],

];

1. Поместите строку в новый tests/codeception/config/config-local.php:

<?php  
return [

'components' => [

'db' => [

'dsn' => 'mysql:host=localhost;dbname=yii2\_basic\_tests',

],

],

];

1. Добавьте объединение конфигурации в web/index.php файл:

$config = yii\helpers\ArrayHelper::merge(

require(\_\_DIR\_\_ .'/../config/common.php'),

require( \_\_DIR\_\_ .'/../config/common-local.php'),

require(\_\_DIR\_\_ .'/../config/web.php'),

require(\_\_DIR\_\_ .'/../config/web-local.php')

);

1. Добавить слияние конфигурации в сценарий ввода консоли, yii:

$config = yii\helpers\ArrayHelper::merge(

require(\_\_DIR\_\_ .'/config/common.php'),

require(\_\_DIR\_\_ .'/config/common-local.php'),

require(\_\_DIR\_\_ .'/config/console.php'),

require(\_\_DIR\_\_ .'/config/console-local.php')

);

1. Добавить объединение конфигураций в конфигурации тестирования , функциональных и приемочных тестов из tests/codeception/config:

return yii\helpers\ArrayHelper::merge(

require(\_\_DIR \_\_ .'/../../../config/common.php'),

require(\_\_DIR\_\_ . '/../../../config/common-local.php'),

require(\_\_DIR \_\_ .'/../../../config/web.php'),

require(\_\_DIR \_\_ .'/../../../config/web-local.php'),

require(\_\_DIR \_\_ .'/config.php'),

require(\_\_DIR \_\_ .'/config-local.php'),

[

// ...

]

);

1. Добавление слияния конфигурации в сценарий входа консоли среды тестирования tests/codeception/bin/yii:

$config = yii\helpers\ArrayHelper::merge(

require(YII\_APP\_BASE\_PATH . '/config/common.php'),  
require(YII\_APP\_BASE\_PATH . '/config/common-local.php'),  
require(YII\_APP\_BASE\_PATH . '/config/console.php'),  
require(YII\_APP\_BASE\_PATH . '/config/console-local.php'),

require( DIR . '/../config/config.php'),

require( DIR . '/../config/config-local.php')

);

1. В результате необходимо получить следующее содержимое в каталоге конфигурации:

config

common.php  
common-local.php  
console.php  
console-local.php  
web.php  
web-local.php  
params.php  
params-local.php

1. После всего можно добавить новое в файл .gitignore с этим содержимым в каталогах config и tests/codeception/config позволяет игнорировать локальные файлы конфигурации системой управления версиями Git:

/\*-local.php

Как это работает...

Можно сохранить конфигурацию общих компонентов приложения в config/common.php файл, а также установить конкретные конфигурации для веб-и консольных приложений. Вы можете поместить ваши временные и безопасные данные конфигурации в \* - local.php файлы.

Кроме того, вы можете скопировать сценарий оболочки инициализации из yii2-app-advanced.

1. Создайте новый каталог сред и скопируйте в него шаблоны:

environments

dev

config

common-local.php  
console-local.php  
web-local.php  
params-local.php

web

index.php  
index - test.php  
tests

codeception

config

config.php  
config-local.php

yii

prod

config

common-local.php  
console-local.php  
web-local.php  
params-local.php

web

index.php

yii

1. Создайте environments/index.php файл с этим кодом:

<?php  
return [

'Development' => [

'path' => ' dev',

'setWritable' => [

'runtime',

'web/assets',

],

'setExecutable' => [

'yii',

'tests/codeception/bin/yii',

],

'setCookieValidationKey' => [

'config/web-local.php',

],

],

'Production' => [

'path' => ' prod ',

'setWritable' => [

'runtime',

'web/assets',

],

'setExecutable' => [

'yii',

],

'setCookieValidationKey' => [

'config/web-local.php',

],

],

];

1. Удалите конфигурацию Installer::postCreateProject из composer.json:  
   "extra": {

"asset-installer-paths": {

"npm-asset-library": "vendor/npm",

"bower-asset-library": "vendor/bower"

}

}

1. [Скопируйте init и init.bat из расширенного шаблона  
   , https ://github. c om/yiisoft/yii2-](https://github.com/yiisoft/yii2-app-advanced)app-advanced а запустить процесс инициализации можно с помощью команды php init после клонирования проекта из репозитория.

Смотрите так же

Для получения дополнительной информации о конфигурации см.

<https://www.yiiframework.com/doc/guide/2.0/en/concept-configurations>

по русски <https://www.yiiframework.com/doc/guide/2.0/ru/concept-configurations>