Yii2中如何使用CodeCeption

Yii2和CodeCeption

CodeCeption是一个全栈的PHP测试框架，关于CodeCeption的介绍见：CodeCeption官方文档 (https://github.com/Codeception/Codeception/tree/2.0/docs)。

Yii2官方增加了对CodeCeption的支持，这里主要讲解Yii2里如何基于CodeCeption进行单元测试和功能测试。

知识准备

- Composer基础：Composer官方文档 (http://docs.phpcomposer.com/00-intro.html)
- CodeCeption基础：CodeCeption官方文档 (https://github.com/Codeception/Codeception/tree/2.0/docs)
- Yii2-app-basic中的CodeCeption例子：yii2-app-basic (https://github.com/yiisoft/yii2-app-basic/blob/master/tests/README.md)
- Yii2-app-basic项目的地址：https://github.com/yiisoft/yii2-app-basic (https://github.com/yiisoft/yii2-app-basic)

执行后面操作的前提是本地已经正确安装配置composer。

通过脚手架开始一个Yii2项目

通过Composer，我们可以很简单的在本地基于脚手架创建一个Yii2项目：

- 进入一个可以通过web服务器访问的目录： `cd <webroot>`
- 执行： `composer global require "fxp/composer-asset-plugin:~1.0.0"` 命令验证必须插件是否全局安装
- 执行： `composer create-project yiisoft/yii2-app-basic basic` 命令在当前目录的basic目录创建一个”yiisoft/yii2-app/basic”脚手架项目。注意，如果是本地开发环境，你也可以增加 `--prefer-dist --stability=dev` 选项，参考stability选项 (https://github.com/5-say/composer-doc-cn/blob/master/cn-introduction/04-schema.md#minimum-stability)，这样，composer会到github上拉取最新的开发包下来。

通过以上操作，你的本地目录应该大致如下：

```
webroot/basic
        |--- assets/
        |--- commands/
        |--- config/
        |--- controllers/
        |--- mail/
        |--- models/
        |--- runtime/
        |--- tests/
        |--- vendor/
        |--- views/
        |--- web/
        |--- .gitignore
        |--- composer.json
        |--- composer.lock
        ...
```

其中，tests就是框架默认创建的测试代码目录，里面有框架提供的一些测试的例子，你可以按如下步骤测试一下：

- `cd basic/tests/` 进入CodeCeption测试用例所在目录
- `codecept build` 将构建测试用例（根据cept生成tester）
- `codecept run` 运行测试用例

如果你终端提示codecept命令未知，请执行以下命令安装codeception扩展：

```
composer global require "codeception/codeception=2.0.*"
composer global require "codeception/specify=*"
composer global require "codeception/verify=*"
composer require --dev yiisoft/yii2-faker:*
```

正常安装后，再执行 `codecept run` 时，如果看到类似如下的报错：

```
1) Failed to ensure that about works in [1mAboutCept[22m (D:\php\basic\tests\codeception\acceptance\AboutCept.php)
Can't be on page "/index-test.php?r=site%2Fabout":
GuzzleHttp\Exception\ConnectException: cURL error 7: Failed to connect to localhost port 8080: Connection refused
```

提示在xx端口连接拒绝，这里，我们需要修改一下配置文件:

- 修改 `basic/tests/codeception.yml` 里面的 `config/test_entry_url` 配置，为你实际的项目入口地址
- 修改 `basic/tests/codeception/acceptance.suite.yml` 里面的 `modules/config/PhpBrowser` 配置为你实际的地址

完成后，再执行 `codecept run` ，你应该可以看到终端没有报错了。

我们来看看tests目录的结构：

```
webroot/basic/tests
        |--- codeception/
        |               |--- _output/
        |               |--- _pages/
        |               |--- acceptance/
        |               |--- bin/
        |               |--- config/
        |               |--- fixtures/
        |               |--- functional/
        |               |--- templates/
        |               |--- unit
        |               |--- _bootstrap.php
        |               |--- acceptance.suite.yml
        |               |--- functional.suite.yml
        |               |--- unit.suite.yml
        |--- codeception.yml
```

其实，这里acceptance、functional、unit是Yii2默认为我们创建的三个suite，顾名思义，分别用于验收，功能，单元测试。

而执行 `codecept run` 时，会依次将codeception目录的所有suite运行，故，你可以通过 `codecept run suitename` 的方式制定执行某个suite；同理，可以执行 `codecept run suitename testname` 的方式执行某个test。

你可以仿照functional，unit，acceptance里面的例子写你自己的测试用例。

> 注：通过脚手架创建的Yii2项目会自动增加gitignore，将vendor中的内容从版本库中忽略，如果你需要提交，请手动修改gitignore文件。

从已有的Yii2项目开始CodeCeption

对于一个已有的Yii2项目，我们需要遵循如下几步来配置CodeCeption。

- 在项目根目录执行 `composer init` 来初始化composer。
- 执行如下指令确保codeception的扩展包已经全局安装
    - `composer global require "codeception/codeception=2.0.*"`
    - `composer global require "codeception/specify=*"`
    - `composer global require "codeception/verify=*"`
    - `composer require --dev yiisoft/yii2-faker:*`
- 在项目合适的目录创建一个codeception目录作为codeception的测试代码目录
- 进入codeception目录，执行 `codecept bootstrap` 来初始化生成测试代码脚手架。
- 仿照yii2-app-basic，修改codeception.yml
- 仿照yii2-app-basic，修改codeception/tests/_bootstrap.php文件
- 仿照yii2-app-basic，修改codeception/tests/*.suite.yml
- 增加一个codeception/tests/config目录，以存储配置文件。仿照yii2-app-basic的形式增加config.php,common.php

> 具体的配置内容依实际情况而定，具体可以参考yii2-app-basic提供的例子。

附：如何基于yii2-app-basic按照module来组织项目代码

前面所述，通过脚手架创建yii2-app-basic开发环境后，我们的代码目录结构为：

```
webroot/basic
        |--- assets/
        |--- commands/
        |--- config/
        |--- controllers/
        |--- mail/
        |--- models/
        |--- runtime/
        |--- tests/
        |--- vendor/
        |--- views/
        |--- web/
        |--- .gitignore
        |--- composer.json
        |--- composer.lock
        ...
```

这个结构是Yii2默认创建的目录，我们可以看到它并没有安装module进行划分。我们可以对其进行调整：

- 新建一个modules目录，假设默认的module为demo
- 在modules目录下面新建demo目录作为demo模块代码存放目录
- 将上面的controllers，models，views三个目录移到demo目录下面

- 在demo目录下新建一个Module.php，新建一个类继承自[[\yii\base\Module]],代码见后。
- 在config目录下面增加一个modules.php，增加对demo目录的配置
- 修改web.php，增加对modules.php的引用
- 修改index.php，增加@modules的别名
- 对应调整modules/demo/里面的namespace

按照以上步骤调整后的路径如下：

```
webroot/basic
        |--- assets/
        |--- commands/
        |--- config/
        |--- modules/
        |               |--- demo/
        |               |               |--- controllers/
        |               |               |--- views/
        |               |               |--- models/
        |               |--- ...
        |--- mail/
        |--- runtime/
        |--- tests/
        |--- vendor/
        |--- web/
        |--- .gitignore
        |--- composer.json
        |--- composer.lock
        ...
```

各文件的内容如下：

basic/modules/demo/Module.php

```
<?php
 /**
  * Module.php
  *
  * @author        fangliang
  * @create_time           2015-06-16
  */

namespace modules\demo;


class Module extends \yii\base\Module
{
    public $layout = "main";
    public $controllerNamespace = 'modules\demo\controllers';

    public function init()
    {
        parent::init();
        //do something init here
    }
}
```

basic/web/index.php

```
<?php

// comment out the following two lines when deployed to production
defined('YII_DEBUG') or define('YII_DEBUG', true);
defined('YII_ENV') or define('YII_ENV', 'dev');

require(__DIR__ . '/../vendor/autoload.php');
require(__DIR__ . '/../vendor/yiisoft/yii2/Yii.php');

$config = require(__DIR__ . '/../config/web.php');

Yii::setAlias("@modules",dirname(__DIR__) .'/modules' );

(new yii\web\Application($config))->run();
```

basic/config/web.php

```
<?php

$params = require(__DIR__ . '/params.php');

$config = [
    'id' => 'basic',
    'basePath' => dirname(__DIR__),
    'bootstrap' => ['log'],
    'components' => [
        'request' => [
            // !!! insert a secret key in the following (if it is empty) - this is required by cookie validation
            'cookieValidationKey' => '0_wkmOLxql9rlIaqjYNPFL3pYDfLNuLk',
        ],
        'cache' => [
            'class' => 'yii\caching\FileCache',
        ],
        'user' => [
            'identityClass' => 'modules\demo\models\User',
            'enableAutoLogin' => true,
        ],
        'errorHandler' => [
            'errorAction' => 'demo/site/error',
        ],
        'mailer' => [
            'class' => 'yii\swiftmailer\Mailer',
            // send all mails to a file by default. You have to set
            // 'useFileTransport' to false and configure a transport
            // for the mailer to send real emails.
            'useFileTransport' => true,
        ],
        'log' => [
            'traceLevel' => YII_DEBUG ? 3 : 0,
            'targets' => [
                [
                    'class' => 'yii\log\FileTarget',
                    'levels' => ['error', 'warning'],
                ],
            ],
        ],
        'db' => require(__DIR__ . '/db.php'),
    ],
    'params' => $params,
    'modules'=> require(__DIR__.'/modules.php'),
];

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;
```

basic/config/modules.php

```
<?php

return [
    'demo' =>[
        'class'=>'modules\demo\Module',
    ]
];
```

在完成以上的配置后，基本的配置就完成了。

如果是一个全新的项目，你可以直接fork我的这个脚手架：Yii2模块化Web应用脚手架 (https://github.com/hustnaive/yii2-app-modular)。