**1、测试目录结构**

项目根目录下的 tests 文件夹或者 models 下的项目中的 tests 文件夹。

（

执行单元测试命令

phpunit

-------------会执行 所有test 方法

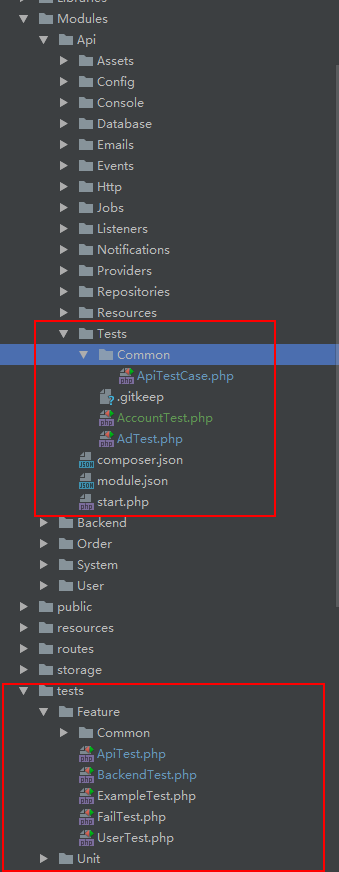
phpunit tests/Feature/BackendTest.php

-------------会执行 所有 BackendTest.php 里的test方法

）

也可以直接 命令行：php artisan make:test DemoTest

（这种情况是直接 测试文件创建在项目根目录下的 tests /Feature ）



**2、测试代码编写**

举个栗子：

--------

class AccountTest extends TestCase

//所有测试类必须基于TestCase

{

/\*\*

\* 测试用户信息详情页接口

\* @return mixed

\*/

//所有测试方法命名，必须以test小写开头命名（或者在注释中写明标注@test）

public function testuserinfo\_detail()

{

$response = $this->get('http://ideabuy.xin.cn/api/userinfo-detail',$this->authHeader)

// $this->assertEquals('ture',$response->getcontent());

//当下面断言失败时，将这行代码取消注释，将下面注释，上面代码最后加上分号，再次运行，即可看到报错具体信息。

->assertJsonFragment([

'msg'=>'查询成功',

//断言返回的JSON值中包含一个 msg：‘查询成功’ 的对象

]);

}

}

------访问接口方式------

get（‘URL地址'，'请求头部信息——登录信息等'）;//get方法访问的参数拼接在URL上

clipboard.png

post（‘URL地址’，'参数'，’请求头部信息——登录信息等‘）;

clipboard.png

//post方法，请求参数放在第二个位置，以数组形式

call（'请求方式（post/get）','请求路由'，’请求参数‘，'请求头部'）；

clipboard.png

json（'请求方式（post/get）','请求路由'，’请求参数‘，'请求头部'）；

clipboard.png

------获取接口返回值------

$response->getcontent();

------返回结果断言------

简单列几个常见的断言方法，具体更多的断言方法下面链接：

<https://phpunit.de/manual/6.3/zh_cn/appendixes.assertions.html>

1.

$this->assertJsonFragment（$params）;// 判断返回的JSON数值里是否包含$params

栗子：

**$this->assertJsonFragment([**

**'msg'=>'头像修改成功',**

**]);**

2.

$this->assertJsonStructure（$params）;// 判断返回的JSON数据里是否包含$params数据结构

栗子：

**$this->assertJsonStructure([**

**'data'=>[],**

**]);**

3.

$this->assertEquals（1，$params）;// 断返回值是否与$params相同

栗子：

**$this->assertEquals(1, $result['code']);**

4、

assertResponseOk()；// 断言客户端返回的响应状态码是否是200

**3、模拟用户登录的坑（还没完全爬出来）**

由于大部分接口都做了用户登录的判断，所以在对接口进行测试的时候需要带上用户登录的信息（token+版本信息）

获取token的时候（即模拟登陆的时候--userlogin方法）如果报 data 错误，先检查一下本地 环境是不是需要全拼路由才能访问接口。

即：

$response = $this->get('http://ideabuy.xin.cn/backend/constype-list', $this->authHeader)

如果token可以打印的情况下，还报错，检查一下setup 方法是否添加了刷新语句。

即：

$this->refreshApplication();

下面的代码举了一个userlogin登录的操作，backend登录可以仿照下面操作，在 Common 里的方法中添加一个adminlogin，然后继承这个类去调用登录后的token,嗯，我还是贴代码吧：

**路径**：test/common/ApiTestCase.php

在Common 里的文件中加这样一段：

public function adminLogin($params = [])

{

$response = $this->post(

'http://ideabuy.xin.cn/backend/admin-login',

$params

);

$admins = $response->getOriginalContent();

return $admins;

}

**路径**：test/BackendTest.php

在backend 测试里的setup 方法

public function setUp()

{

parent::setUp();

$this->params = $this->readyApiParams();

$admin = $this->adminLogin($this->params['login']);

$this->admin = $admin;

$this->authHeader = $this->headers($admin);

$this->refreshApplication();

}

接下来在你写的test方法里的头部信息就可以直接$this->authHeader 进行调用了！

类似于这样

clipboard.png

附上登录用例代码：

**路径**：test/common/ApiTestCase.php

<?php

namespace Modules\Api\Tests\Common;

use Tests\TestCase;

/\*\*

\* Class ApiTestCase

\* @package Modules\Api\Tests\Common

\*/

class ApiTestCase extends TestCase

{

/\*\*

\* User login

\* @param array $params

\* @return mixed

\*/

public function userLogin($params = [])

{

$response = $this->post(

#路由看个人环境配置是否需要加上 http://ideabuy.xin.cn

'http://ideabuy.xin.cn/api/user-login',

$params,

#调用下面拼接token的方法

$this->headers()

);

$users = $response->getOriginalContent();

return $users;

}

/\*\*

\* Setting request header

\* @param null $user

\* @param array $addition

\* @return array

\*/

protected function headers($user = null, $addition = [])

{

#添加版本号头部信息

$headers = ['Accept' => 'application/vnd.ideabuy.v1+json'];

#

if (!is\_null($user)) {

$headers['Authorization'] = 'Bearer '.$user['data']['token'];

}

if($addition){

$headers = array\_merge($headers,$addition);

}

return $headers;

}

}

**路径**：test/AccountTest.php

<?php

namespace Modules\Api\Tests;

use Modules\Api\Tests\Common\ApiTestCase;

class AccountTest extends ApiTestCase

{

public $authHeader;

public $params;

public $data;

public $user;

/\*\*

\* ready for data & params

\*/

public function setUp()

{

#每个test方法之前都会调用一次这个方法

parent::setUp();

$this->params = $this->readyApiParams();

/\*\* Get auth info include token \*/

#调用用户登录的操作

$user = $this->userLogin($this->params['login']);

$this->user = $user;

#拼接token值

$this->authHeader = $this->headers($user);

#刷新应用。该操作由TestCase的setup()方法自动调用，不然会使用过期的token

$this->refreshApplication();

}

/\*\*

\* Ready for test params

\*/

protected function readyApiParams()

{

#这个方法用来放test里需要的参数

$params = [];

/\*\* login params for test \*\*/

$login = [

'user\_mobile' => '13777979098',

'user\_password' => 'a12345678'

];

$params['login'] = $login;

/\*\* register user params for test \*\*/

$register = [

'user\_mobile' => '13777979098',

'user\_password' => 'a12345678',

'confirm\_password' => 'a12345678',

'code' => '1234'

];

$params['register'] = $register;

/\*\* setting user paypassword params for test \*\*/

$payPassword = ['pay\_password' => 'b11111111'];

$params['payPassword'] = $payPassword;

return $params;

}

/\*\*

\* Clean test data

\*/

protected function cleanData()

{

//todo logic data

//Artisan::call('migrate:reset');

}

/\*\*

\* this is case for user login

\*/

public function testUserLogin()

{

$this->assertEquals(1, $this->user['code']);

}

/\*\*

\* This is case for setting pay password

\* @test

\*/

public function SetPayPassword()

{

$response = $this->post(

'http://ideabuy.xin.cn/api/user-setpaypwd',

$this->params['payPassword'],

#获取token

$this->authHeader

);

$result = $response->getOriginalContent();

#getOriginalContent（）是将返回值转成数组，getcontent（）是直接将对象返回

$this->assertEquals(1, $result['code']);

}

/\*\*

\* Drop something test data

\*/

public function tearDown()

{

$this->cleanData();

parent::tearDown();

}

}