欢迎继续我们的学习。

在本课的内容中，我们将重点介绍Mocha这个测试框架。首先要说明的是，Mocha是javascript的通用测试框架，不仅仅是针对以太坊的。

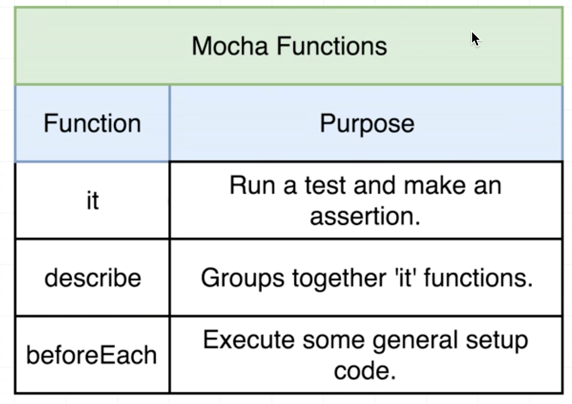
首先来了解下Mocha中最常用的三个函数。

[1.it](http://1.it)

用于执行测试，以及创建一个断言

2.describe

用于将it函数组织在一起



3.beforeEach

用于执行一些常规的设置代码。

接下来让我们尝试编写自己的测试代码。

首先创建一个Car类：

class Car{

park(){

return 'stopped';

}

drive(){

return 'vroom';

}

}

再次强调一下，本教程是基于javascript的ES6版本（ECMAScript6）。

接下来编写测试代码：

describe('Car', () => {

it('can park', () => {

const car = new Car();

assert.equal(car.park(),'stopped');



});

});

在以上代码中，我们首先使用describe将多个it(或一个）函数组织在一起。

然后就是it函数的具体使用，我们创建了一个Car类的实例对象，然后使用assert函数创建了一个断言。

在继续之前，让我们更改项目文件夹中的package.json。

当前的代码如下：

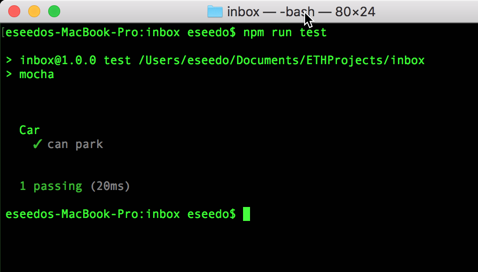
将红色框线中的代码更改为如下：

然后打开Terminal终端（或windows系统中的cmd)，

进入项目文件夹，并输入如下命令：

npm run test

等待几秒之后就会看到结果如下：



接下来让我们有意更改测试代码的内容，

describe('Car', () => {

it('can park', () => {

const car = new Car();

assert.equal(car.park(),'stoaapped');

});

});

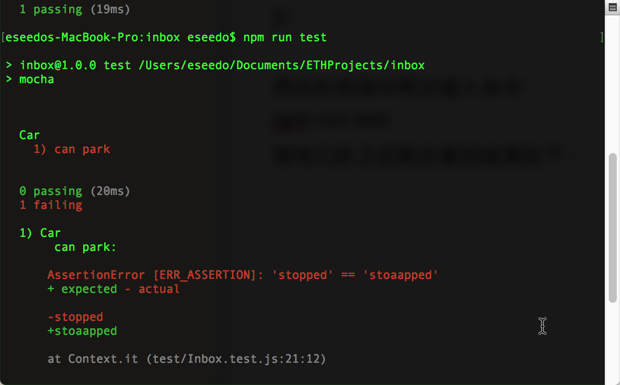
然后在终端中再次输入命令：

npm run test

等待几秒之后就会看到结果如下：

显然，这一次的测试以失败告终。

接下来在describe中添加另外一个测试函数。回到Atom，更改describe函数中的代码如下：



describe('Car', () => {

it('can park', () => {

const car = new Car();

assert.equal(car.park(),'stopped');

});

it('can drive', () =>{

const car = new Car();

assert.equal(car.drive(),'vroom');

});

});

再次回到终端，输入命令：

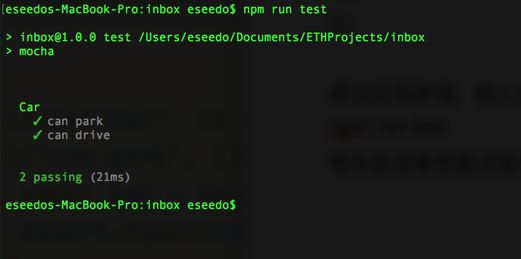
npm run test

很快就会看到测试结果如下：

接下来让我们对describe函数中的代码优化一下。

在当前的代码中，在每个it函数中，我们都用到了同一行代码，也就是创建Car类的实例变量。

对于比较复杂的应用，这种方式是非常低效的，所以我们可以考虑使用beforeEach进行一些通用的常规设置。



在Inbox.test.js的describe函数之前添加以下代码：

beforeEach(() => {

const car = new Car();

});

然后将describe函数中的代码修改如下：

describe('Car', () => {

it('can park', () => {

assert.equal(car.park(),'stopped');

});

it('can drive', () =>{

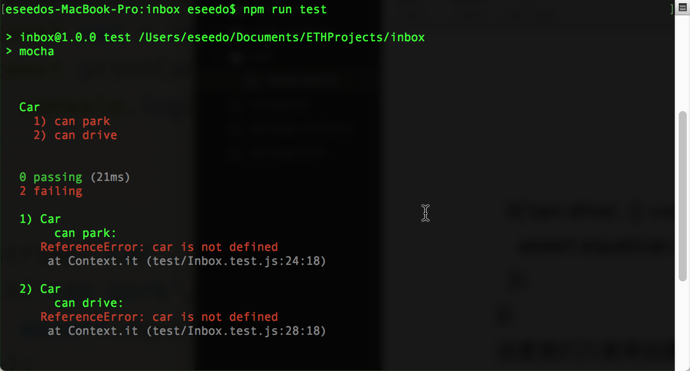
assert.equal(car.drive(),'vroom');

});

});

这里我们只是将创建Car实例对象的代码从it函数中删掉了。

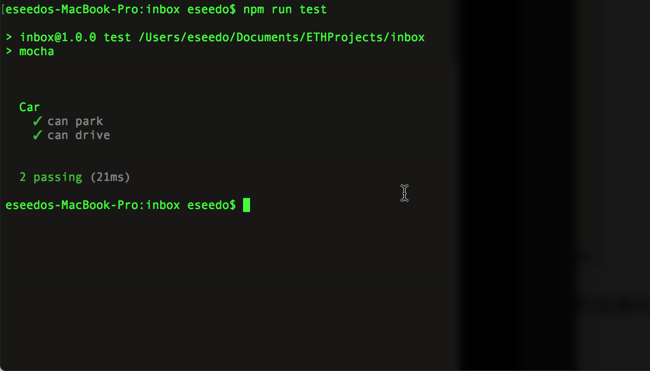
但是需要注意的是，如果此时运行测试是会报错的。



这是因为在beforeEach函数中定义的car作用域仅限于beforeEach函数中，在describe函数中是无法识别的。

因此，我们应该修改这部分的代码如下：

这里我们在函数体之外声明了car，然后在beforeEach函数中创建了Car类的实例对象，并赋予car这个变量。



之所以这里使用let而不是const，是因为我们需要修改car的值。

再次运行测试，会发现一切正常了。

在继续之前，让我们删除刚才的测试代码，此时Inbox.test.js中的代码如下：

好了，本课的内容就到这里了。

相信你对如何使用Mocha进行测试有了基本的概念。

在下一课的内容中，我们将带领大家学习如何使用Mocha对智能合约代码进行测试。



我们下一课再见。

讨论群-笨猫学编程QQ群：

375143733

答疑论坛：

<http://www.vr910.com/forum.php?mod=forumdisplay&fid=157>

知乎专栏：

<https://zhuanlan.zhihu.com/kidscoding>

新浪博客:

<http://blog.sina.com.cn/eseedo>

Github:

<https://github.com/eseedo>

个人网站：

<http://icode.ai/>