2015年3月29日

终于搞明白scenario要怎么用了。

这里是yii1.0里的代码，和yii2.0大同小异。

public function rules ()

{

// NOTE: you should only define rules for those attributes that will receive user inputs.

return array (

array ( 'create\_user\_id, update\_user\_id', 'numerical', 'integerOnly'=> true ) ,

array ( 'name', 'length', 'max'=>128 ) ,

array ( 'create\_time, update\_time', 'safe' ) ,

// The following rule is used by search().

// Please remove those attributes that should not be searched.

array ( 'id, name, description, create\_time, create\_user\_id,

update\_time, update\_user\_id', 'safe', 'on'=>'search' ) ,

) ;

scenario(情景)允许你限制验证规则应用在特定的上下文中。一种典型的例子是 insert(插入)或update(更新)。例如：如果被指定为 'on'=>'insert'，这将表明验证规则只适用于模型的插入情景。这同样适用于'update'或其它的任何你希望定义的情景。你可以设置一个模型的 scenario(情景)属性或通过构造函数传给一个模型的实例。

使用fixtures为测试单独配置一个数据库。

一个测试fixture是指在测试运行中的一个系统状态或上下文。我们想多次运行我们的测试，每次运行我们希望能够让他们可以返回重复的结果。夹具的目的是提供一个众所周知在固定的环境中运行验证。通常情况下，一个夹具的主要工作是确保所有参与测试的对象在测试运行中初始化一个指定的状态。一个典型的夹具例子是使用固定和已知的数据加载一个数据库的表。

这里是yii2.0的官方讲解，很清晰

<http://www.yiiframework.com/doc-2.0/guide-test-fixtures.html>

yii2.0中将filter换成了behaviors

现在需要用到用户了，但是用户验证成了一个大问题，所以需要处理用户验证的事。

二进制比较

strcasecmp($user['username'], $username)

我自己做了一个user表，但是直接替换yii2.0 basic里面的那个表就出错。

于是就根据其内容改来改去的。

有几个地方要注意：

1.user表必须要用这个系统接口

implements \yii\web\IdentityInterface

2.表里必须要有authKey和accessToken两个字段，这两个字段主要是yii2.0系统用来加密用户信息的。

3.yii2.0系统里面是通过findIdentityByAccessToken()、getAuthKey()、validateAuthKey()来读取authKey和accessToken两个字段的，所以这几个方法必须要写进user的model中。

设定表格的controller(搞这个搞了一晚上)

[

'class' => 'yii\grid\ActionColumn',

'controller' => 'issue',

],

2015年4月5日

跳过了测试章节，又跳过了权限分配，接下来要实现评论功能了。