用户验证（Authentication）

用户验证，原文网址：

http://www.yiiframework.com/doc-2.0/guide-security-authentication.html

用户验证（Authentication）用于验证一个用户的身份标识(identity)。通常会使用一个标识符(identifier)和一个密令(secret token)去判定一个客户是否是他声称的那个身份。用户验证是登录操作的基础。  
标识符，如：用户名(username)或邮箱地址(email address)  
密令，如：密码或口令  
Yii2提供了一个用户验证框架，通过这个框架可以很好的支持登录，并将各个组件上串联起来。要使用这个框架，你只需做两项工作：  
1、配置user应用组件 yii\web\User  
2、创建一个实现yii\web\IdentityInterface接口的类

### 1、配置yii\web\User（Configuring yii\web\User）

User(用户)应用组件管理着用户的验证状态，需要你定义一个indetity类，此类中包含着实际的验证逻辑。在下面的应用配置中，user的idetity类配置为app\models\User，如何去实现将在下一节中详述。

**return** [

'componnets'=>[

'user'=>[

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

],

],

];

//配置实例：  
D:\phpwork\b2b\imall\frontend\config\main.php

return [

'id' => 'app-frontend',

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

'bootstrap' => ['log'],

'controllerNamespace' => 'frontend\controllers',

'components' => [

'user' => [

'identityClass' => 'frontend\models\User',

'enableAutoLogin' => true,

'loginUrl' => ['site/login'],

],

......

];

### 2、实现yii\web\IdentityInterface接口（Implementing yii\web\IdentityInterface）

findIdentity()，根据给定的user ID查找用户身份类的实例，此方法在通过session维护登录状态时使用。  
findIdentityByAccessToken()，使用定义的口令来查找用户身份类的实例，此方法在使用单一口令验证用户身份时使用，多数用于RESTful应用中。  
getId()，获取当前用户身份实例的ID。  
getAuthKey()，获取一个key，用于验证基于cookie的登录。这个key是存储在登录cookie中，并将与服务器端的版本相比较以确定此cookie是有效的。  
validateAuthKey()，它用于实现验证基于cookie登录的key的逻辑。

如果无需实际的方法，你可以使用一个空体实现它。例如，如果你的应用是一个单纯的无状态RESTful应用，你只需实现findIdntityByAccessToken()和getId()方法，而其他的方法体为空即可。

在接下来的这个例子中，一个identity类继承自Active Record类，并与user数据表相关联：

<?php

**use** **yii**\**db**\**ActiveRecord**;

**use** **yii**\**web**\**IdentityInterface**;

**class** **User** **extends** **ActiveRecord** **implements** **IdentityInterface**{

**public** **static** **function** **tableName**(){

**return** 'user';

}

/\*\*

\* Finds an identity by the given ID.

\*

\* **@param** string|integer $id the ID to be looked for

\* **@return** IdentityInterface|null the identity object that matches the given ID.

\*/

**public** **static** **function** **findIdentity**($id){

**return** **static**::findOne($id);

}

/\*\*

\* Finds an identity by the given token.

\*

\* **@param** string $token the token to be looked for

\* **@return** IdentityInterface|null the identity object that matches the given token.

\*/

**public** **static** **function** **findIdentityByAccessToken**($token, $type = null){

**return** **static**::findOne(['access\_token' => $token]);

}

/\*\*

\* **@return** int|string current user ID

\*/

**public** **function** **getId**(){

**return** **$this**->id;

}

/\*\*

\* **@return** string current user auth key

\*/

**public** **function** **getAuthKey**(){

**return** **$this**->auth\_key;

}

/\*\*

\* **@param** string $authKey

\* **@return** boolean if auth key is valid for current user

\*/

**public** **function** **validateAuthKey**($authKey){

**return** **$this**->getAuthKey() === $authKey;

}

}

如前所述，你只需实现getAuthKey()和validateAuthKey()，如果你的应用使用的是基于cookie的登录，可以使用以下代码去为每一个用户生成一个验证密钥，并将它存储在use表中。

**class** **User** **extends** **ActiveRecord** **implements** **IdentityInterface**

{

......

**public** **function** **beforeSave**($insert){

**if** (**parent**::beforeSave($insert)) {

**if** (**$this**->isNewRecord) {

**$this**->auth\_key = \Yii::$app->security->generateRandomString();

}

**return** **true**;

}

**return** **false**;

}

}

注意：不要困惑于User标识类和yii\web\User，前者是验证逻辑的实现类，它通常继承自一个AR类，并与用户关键信息的永久存储相关；后者是一个应用组件类，负责管理用户的验证状态。

### 3、使用yii\web\User（Using yii\web\User）使用yii\web\User的主要方式是使用user应用组件。 你可以使用表达式Yii::$app->user->identity检测当前用户的身份标识，它将返回一个identity类的实例，这个实例代表当前的登录用户，如果当前用户是未验证的，则返回null，也就是访客(guest)。以下代码演示了如何从yii\web\User获取其他验证相关信息：

//获取用户的identity

$identity=Yii::$app->user->identity;

//获取用户的id

$id=Yii::$app->user->id;

//获取用户是否是访客身份

$isGuest=Yii::$app->user->isGuest;

### 用户登录

//查找$username的用户身份标识，此前应进行密码或口令验证

$identity=User::findOne(['userName'=>$username]);

//以$identity身份登录系统

Yii::$app->user->login($identity);

yii\web\User::login()方法设置当前用户的身份标识identity到yii\web\User中，如果session有效，此方法将会保存identity到session中，这样用户验证状态将在整个会话过程中被维护。如果基于cookie的登录有效，如"remeber me"登录，它将保存identity到cookie中，这样用户验证状态将在cookie有效期内被cookie所覆盖。  
要使基于cookie的登录生效，必须在应用配置中设置yii\web\User::$enableAutoLogin 为true。同时你还需要在调用yii\web\User::login()方法时提供一个保存时长的参数。

### 退出登录

Y**ii**::$app->**user**->logout();

注意：会员退出登录仅当session有效时有意义。logout()方法将从内存和session中清除用户登录状态，同时它也会清除掉该用户所有的session数据。如果你想保留session数据，可以调用Yii::$app->user->logout(false)即可。

### 4、验证事件(Authentication Events)

yii\web\User类为login和logout增加了几个事件：

EVENT\_BEFORE\_LOGIN，在yii\web\User::login()调用之前运行，如果此事件处理器设置事件对象的isValid属性为false，登录操作将被取消。

EVENT\_AFTER\_LOGIN，在成功登录后运行。

EVENT\_BEFORE\_LOGOUT，在yii\web\User::logout()调用之前运行，如果此事件处理器设置事件对象的isValid属性为false，退出操作将被取消。

EVENT\_AFTER\_LOGOUT，在成功退出后运行。

你可以通过这些事件去实现一些特性，如：登录审计，在线用户统计。举例来说：在EVENT\_AFTER\_LOGIN处理程序中，你可以在user表中记录登录时间和IP地址。

# 关于Yii 2.0 认证（Authentication）的理解

认证（authentication），一开始对我来说很抽象，应该很多像我一样的新手都有一样的感受。说简单些，认证就是验证你的用户名和密码是否正确，是否匹配。在实际的应用中，则体现为登录。（官方文档：Authentication is the basis of the login feature.）以下，我用自己的理解，结合文档，分析一下使用Yii2实现登录和自动登录的思路。  
一、准备工作

Configure the yii\web\User application component;

//在配置文件中配置好user应用组件

**Create** a **class** that **implements** the yii\web\IdentityInterface **interface**.

//实现IdentityInterface接口

二、关于user

1、配置好**user**后，可以在全局使用Yii::$app->user访问；

2、**user**的重要属性，$isGuest:判断用户是否登录；$id,保存登录用户的唯一识别标识，由我们自己设置，具体如何设置，且看后文分解；

3、**user**的重要方法，login()：将用户信息保存在session、cookie中；logout():将用户信息从session、cookie中销毁；

Note that User only maintains the **user** authentication status. It does NOT handle how **to** authenticate a **user**. The logic of how **to** authenticate a **user** should be done **in** the class implementing yii\web\IdentityInterface.

//**user**组件的工作只是维护用户的认证状态（即简单的操作session和cookie）。它并不负责认证用户。认证过程的逻辑实现统统由一个实现了IdentityInterface接口的类实现；

三、关于IdentityInterface；

1、这个接口定义了5个函数：

findIdentity();

getId();

findIdentityByAccessToken();

getAuthKey();

validateAuthKey;

这5个函数当中，最重要的可能就是getId()了。这个函数由你实现，由传递给user的对象在login的时候调用，用于设置user的id属性值！所以，在实现这个函数的时候，你返回的是什么，那么user组件记住的user id就是什么。其他几个函数的具体实现，请参照文档；

四、登录

假设我们定义了一个名为User的AR模型类（注意应用组件user的区别），User对应了一张我们保存有用户登录信息的数据表；如果我们要使用User来完成用户认证，User就必须实现上文提及的IdentityInterface；

1、服务器接受到用户提交的用户名和密码；

2、生成User实例$model;

3、赋值:$model->load(Yii::$app->request->post());

4、检查用户输入以及验证用户名和密码：$model->validate();

5、登录Yii::$app->user->login($model,$duration);

5、自动登录

1、要实现自动登录，需要设置参数$enableAutoLogin；

2、$duration必须大于零；

3、使用$isGuest判断用户是否已经登录；