مدیریت کاربران:

ما دو کاربر تاحلا داشته ایم که یکی ادمین و دیگری دمو.

اول روی جدول کاربران مدل میسازیم و سپس crud(خودش هم گفت کرود) برایش میسازیم.

Rewrite کردن یک کلاس مدل:

گاهی پیش می اید هنگامی که در حال ساخت یک کلاس مدل برای یک جدول هستیم بعداز کلیک روی گزینه ی preview مشاهده کنیم که گزینه ی rewrite که میتواند تیک خورده باشد یا نباشد هم در کنارش قرار میگیرد.امدن این گزینه نشان از این ندارد که ماقبلا یک کلاس کدل برای این موجودیت ساخته ایم و حالا قصد ساخت دومی را داریم. این گزینه را تیک میزنیم تا نسخه ای به روز شده ساخته شود. میتوانیم قبل از کلیک روی generate تعییرات را مشاهد کنیم.

-کاربر دمو نمیتواند manage delete کند.

مقداردهی خودکار به فیلدها:

فیلدهایی که در بردارنده ی زمان ایجاد و زمان اپدیت یا کسی که ایجاد کرده یا کسی که اپدیت کرده چون جز فیلدهای جدول هستند پس ییی هنگامی که فرم را ساخته ان هارا هم اورده اما ما نمی خواهیم اینگونه باشد ما میخواهیم طوری باشد که اگر اعمالی مرتبط با فیلدهای فوق انجام شدهند خودشان به صورت خودکار مقدار بگیرند. ما میتوانیم دستورات مقداردهی خودکار به انهارا برای هرکدام بنویسیم اما چون این فیلدها در همه ی جداولمان مشترک هستند و منطق مشترکی هم دارند ما طوری این دستورات را خواهیم نوشت که برای همه وجود داشته باشند. ما برای این کار چند روش را میتوانیم دنبال کنیم که اولی ایجاد کلاس پایه ی مشترک است و دومی کامپوننت ها هستند کامپوننت ها اشیایی هستند که از کلاس ccomponent یا فرزندی از این کلاس مشتق میشوند بیشتری حجم کلاس هایی که ما مینویسیم مربوط به همین کامپوننت هاست. این ها بسیار کاربردی تر هستند و کارهایشان از طریق رفتارهایی که به کامپوننت ها نسبت میدهند انجام میدهند رفتارها در ییی کلاس هایی هستند که رابط ibihavior را پیاده سازی میکنند. در این روش ما به جای انکه بیاییم کامپوننت ها را گسترش دهیم به ان رفتار نسبت میدهیم. هر رفتار را میتوان به

کامپوننت نسبت داد و هر کامپوننت میتواند چند رفتارداشته باشد این که هر رفتار میتواند برای چند کامپوننت استفاده شود قابلیت استفاده ی مجدد ان هارا نشان میدهد و این که هر کامپوننت میتواند چند رفتاررا پیاده سازی کند قابلیت ارث بری چند گانه را به یاد می اورد.

ما رفتارها رو روی فیلدها مینویسیم و میگوییم این فیلدها هرجا که باشند این رفتار هم رویشان اعمال میشود. به جای اصطلاح نسبت دادن بهتر است از اصطلاح پیوست کردن یا attach کردن استفاده شود یعنی ما رفتارها را به کامپوننت attach میکنیم. پس از ان که با متد attachBihavior پیوست صورت گرفت کامپوننت ما میتواند از متدهای درون رفتار استفاده کند

ایجاد کلاس پایه مشترک برای همه ی موجودیت ها(یک راه رایج برای مقداردهی به فیلدهای مشترک):

Zii: کتابخانه ی افزونه های yii

این کتابخانه یک رفتار اماده برای مقداردهی خودکار دارد

فعلا میخواهیم به دو فیلد create\_time update\_time در جدول user به طور خودکار مقدارداده شود پس وارد کلاس مدل ان میشویم و این متد را مینویسیم:

public function behaviors()

{

return array(

'CTimestampBehavior'=>array(

'class'=>'zii.behaviors.CTimestampBehavior',

'createAttribute'=>'create\_time',

'updateAttribute'=>'update\_time',

'setUpdateOnCreate'=>true,

)(;}

حالا دیگر با create کردن هر user این فیلد ها خود به خود مقدار میگیرند البته بعدا این کد را از این جا برداشته و درکلاسی قرار میدهیم که برای هرسه جدول کار کند.

CTimestampBehavior اسم رفتار است. ان 4 تا هم فیلدهایش هستند.

مقداردهی با فیلدها با کلاس پایه:برای این کار از سیستم مدیریت رویداد استفاده کنیم. ذخیره شدن یک فیلد در جدول اعمال اعتبارسنجی و حذف و ... همه رویداد حساب می ایند و ما میتوانیم یک سری کارها را قبل و بعد از این رویدادها انجام دهیم

مثلا beforesafe aftersafe

ایجاد کلاس پایه مشترک :

<?php

abstract class TrackStarActiveRecord extends CActiveRecord {

protected function beforeSafe(){

if (Yii::app()->user!==null) {

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

}

else{

$id=1;

}

if($this->isnewRecord){

$this->create\_user\_id=$id;}

$this->update\_user\_id=$id;

return parent::beforeSafe();

}

public function behaviors()

{

return array(

'CTimestampBehavior'=>array(

'class'=>'zii.behaviors.CTimestampBehavior',

'createAttribute'=>'create\_time',

'updateAttribute'=>'update\_time',

'setUpdateOnCreate'=>true,

)

);

}

}

?>

حال باید به کلاس های دیگربگوییم به جای cactiverecord از کلاس فوق مشتق شوند. دقیقه 30

حال view های این ها را تغییر میدهیم چون نمیخواهیم این مقادیر را از کاربر بگیرند.

Last\_login\_time برای جدول کاربران را هم از فرمم حدف میکنیم.

چون این مقادر را از فرم حذف کردیم پس زمان ان است تا در متد rules هم validation ها یشان را حذف کنیم.

Bus asinment یا انتساب انبوه زمانی کار میکند که rules برای فیلد مربوطه نوشته باشیم.بررسی کن که اگر فیلدی در فرمی باشد و برایش اعتبارات را حذف کرده باشیم وقتی درج انجام دهیم ایا ان فیلد درجدولش مقدار میگیرد یا نمیگیرد؟

ینی اگر validation نباشد submit نمی شود؟

این حذف اتبارات باعث صرفه جویی در حافظه میشود

چند اعتبارگزاری:

array('email,username','uniqe'),

array('email','email'),

نهادن گزینه ی confirm:

در کلاس مدل میگوییم:

public $password\_repeat;

و در rules میگوییم:

array('password','compare'),

اگر در ییی نگوییم چرابا چه مقایسه کن دنبالی فیلدی همنام با ان با پسوند \_repeat میگردد.

و این خطوط هم ایجاد یا تغییر کردند

array('username, email, password,$password\_repeat', 'required'),

array('username, email, password,$password\_repeat', 'length', 'max'=>255),

array('$password\_repeat','safe'),

این ها اشتباه تغییر کرده اند این را که نوشتیم خطا گرفتیم.نباید از $ استفاده می کردیم:

public function rules()

{

// NOTE: you should only define rules for those attributes that

// will receive user inputs.

return array(

array('username, email, password,password\_repeat', 'required'),

array('username, email, password,password\_repeat', 'length', 'max'=>255),

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

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

array('id, username, email, password, last\_login\_time, create\_time, create\_user\_id, update\_time, update\_user\_id', 'safe', 'on'=>'search'),

array('email,username','unique'),

array('email','email'),

array('password','compare'),

array('password\_repeat','safe'),

);

}

لابد میدانی که در فرم هم باید برایش کاری کرد. در فرم کاربران از پسورد کپی میگیریم و تغییراتی میدهیم فقط در حد تغییرنام میشود ان په میخواستیم:

<div class="row">

<?php echo $form->labelEx($model,'password\_repeat'); ?>

<?php echo $form->passwordField($model,'password\_repeat',array('size'=>60,'maxlength'=>255)); ?>

<?php echo $form->error($model,'password\_repeat'); ?>

</div>

استفاده از رویداد aftervalidate برای هش کردن:

دستور زیر را در کلاس مدل کاربر مینویسیم:

return parent::model($className);

}

protected function afterValidate(){

parent::afterValidate();

if(!$this->hasErrors())

{$this->password=$this->hashpassword($this->password);

}

}

public function hashpassword($password){

return md5($password);

}

این متد هش کردن بعدا به کار میاید در جاهای دیگر به همین دلیل ان را جدا و عمومی نوشتیم.

سرعت کانفیرم و هش کردن کد بهتر است کم باشد وی یک بار کانفیرم میکند هکر کارش بیشتر طول میکشد

in logکردن با کاربران:

میخواهیم از اطلاعات پایگاه داده ی خودمان برای log in کردن استفاده شود.

فعال یا غیرفعال بودن کوکی ها:

True false بودن در کد زیر موجب قابلیت فوق میشود این کد در main.php قرار دارد.

'components'=>array(

'user'=>array(

// enable cookie-based authentication

'allowAutoLogin'=>true,

),

در دستور فوق یوزر یک component است پس هرجا از پروزه میتوانیم با دستور زیر به ان دسترسی پیدا کنیم:

Yii::user()->id

البته نمی دانم این id متعلق به کدام کاربر است. من فکر میکنم رفیقمان app() را فراموش کرده است.

دقیقه ی شصت:

وقتی وارد main.php میشویم مشاهده میکنیم که :

'components'=>array(

'user'=>array(

// enable cookie-based authentication

'allowAutoLogin'=>true,

),

این نشان میدهد که کامپوننت user دارد در برنامه ی ما استفاهد میشود اگر ما بخواهیم از کامپوننت های دیگری هم استفاهده کنیم بدیهی است که باید ان هارا اینجا اسم ببریم اما این که یک کامپوننت چه کاری انجام میدهد در پوشه ی componenets قرار دارد انجا میتوانی محتویات userرا مشاهده کنی.

ما به این دلیل user را به صورت componenets تعریف میکنیم چون میخواهیم در هر جا از پروزه به ان دسترسی داشته باشیم

'allowAutoLogin'=>true,

میگوید که سیستم کوکی فعال باشد اگر این را false کنیم همه چیز در session ذخیره میشود و به محض بستن مرورگرازبین میزود

قابلیتی وجود دارد که با ان میتوان بدون اینکه تغییری در برنامه ایجاد کرد log in کردن و چگونگی ان را تغییر داد مثلا تا دیروز از پایگاه داده log in می کردند حالا میخواهیم open … باشد . این کارزمانی شدنی است که ما منطق اعتبارسنجی کاربر برای وزود را از قسمت های دیکربرنامه جداکنیم.

در کلاس UserIdentityکه در پوشه ی componenets قرار دارد داریم:

public function authenticate()

{

$users=array(

// username => password

'demo'=>'demo',

'admin'=>'admin',

);

if(!isset($users[$this->username]))

$this->errorCode=self::ERROR\_USERNAME\_INVALID;

elseif($users[$this->username]!==$this->password)

$this->errorCode=self::ERROR\_PASSWORD\_INVALID;

else

$this->errorCode=self::ERROR\_NONE;

return !$this->errorCode;

}

نگاه کن به ان $this->username که اشاره به فیلدی از این کلاس ندارد. من میگویم ندارد چون فیلدی به اسم username در این کلاس وجود ندارد. حالا شاید میپرسی پس به چه اشاره دارد؟ این به مقداری که از طرف کاربر در ان مستطیل وارد میشود اشاره دارد و چون در ارایه ی users تنها دو اندیس وجود دارد اگر انچه وارد کرده غیرازاندوباشد isset false برمیگرداند و نقیض ان راست میشود و ... و به همین نحو شرط دوم هم کار میکند.انچه من هنوز نمیدانم این است که چرا$this->username به انچه در کادر نوشته میشود اشاره دارد و اینها کجا تعیین شده اند؟ شاید در کلاس cuseridentity تعریف شده اند.غیر اینگونه نیست این مقادیر از طریق کلاس loginform که در models قرار دارد به این کلاس ارسال میشوند این مطلب در ادامه ی این درس میاید.

self::ERROR\_PASSWORD\_INVALID این ثوابت درکلاس والد تعریف شده اند

self::ERROR\_NONE این ثابت مقدار false دارد اگر خطایی تولید نشود$this->errorCode مقدار false خواهد داشت که نقیض ان true میشود.

اگر true برگرداند کاربر با موفقیت اعتارسنجی شده انگاه بقیه ی سیستم شروع به کار میکند و برای وی سشن میسازد اگر ان تیک را زده باشد کوکی هم برایش میشازد و ادامه ی ماجرا.

حالا ما بررسی میکنیم که جریان چیست و چه دارد رخ میدهد:

فرایند از لحظه ای شروع میشود که شخص گزینه ی log in را میزند. انگاه نوار ادرس مرورگر این شکلی میشود:

<http://localhost/ta/index.php?r=site/login>

این r که میبینی یعنی router الان روتر برابر سایت شده است. خوب چه میشود؟ site در contorolers وجود دارد.همان گونه که وقتی مینوشتیم ?r=project و کنترولری به اسم project داشتیم حالا هم کنترولری به اسم sitecontorolers داریم. و کلمه ی login موجب صدا زدن متد actionlogin در ان کلاس میشود (برایم جالب است هرمتدی که در این کلاس وجو دارد با کلمه ی action شروع میشود) وقتی پیاده سازی actionlogin را نگاه میکنیم میبینیم که در سطر اول ان امده:

$model=new LoginForm;

من دارم می اموزم که این $model مرتب دارد ساخته میشود و هربار شی ای از یک کلاس است و من تا به حال هروقت دیده ام شی $model ساخته اند از کلاس های مدل ساخته اند. الان هم همین طور بود و دیدیم که در فولدر models کلاسی با همین نام وجود دارد.

نکته ی کم ربط: اینگونه نیست که هر فیلدی که در rules از نوعbooleanباشد حتما در فرم ایجاد check box میشود.

public function rules()

{

return array(

// username and password are required

array('username, password', 'required'),

// rememberMe needs to be a boolean

array('rememberMe', 'boolean'),

// password needs to be authenticated

array('password', 'authenticate'),

);

}

اعتبارسنجی ها که واضح اند فقط در مورد اخری بایدگفت: 'authenticate' جز اعتبارسنج های خود yii نیست پس از کجا می اید؟ در این شرایط دنبال متدی با همین نام در ان کلاس میگردد(ما هم یاد بگیریم که به این طریق میتوان اعتبارسنج نوشت البته این مطلب را در جلسه ی 4 خوانده بودیم(

پیاده سازی متد 'authenticate' به این شکل است:

public function authenticate($attribute,$params)

{

if(!$this->hasErrors())

{

$this->\_identity=new UserIdentity($this->username,$this->password);

if(!$this->\_identity->authenticate())

$this->addError('password','Incorrect username or password.');

}

}

میبینی که دو پارامتر دارد. الان password در $attribute و اما در $paramsهیچ چیز قرا نگرفته اگر میخواستیم در$paramsهم مقداری قرار گیرد باید در rules اینگونه می نوشتیم :

array('password', 'authenticate',id=>’test’),

انگاه به صورت یک ارایه در params قرار میگرفتند.

if(!$this->hasErrors())

بررسی میکند که ایا در اعتبارسنجی های قبلی خطایی رخ داده یا نه. اینکه چگونه این کاررا انجام میدهد را نمیدانم.

$this->\_identity فیلدی است که در همین کلاس اعلان شده و با دستور زیر:

$this->\_identity=new UserIdentity($this->username,$this->password);

این فیلد شی ای از کلاس UserIdentity میشود و مقادیری که کاربر در username password وارد کرده هم ازطریق سازنده به کلاس ذکرشده ارسال میشوند.کلاس UserIdentity همان کلاسی است که در پوشه ی components قرارداشت و ما در همین جلسه باهاش اشنا شدیم.عجیبه هرچند در این کد سازنده ی کلاس UserIdentity صدازده شده اما وقتی بدنه ی این کلاس را نگاه کردیم دیدم اصلا سازنده ندارد ما این مطلب را برای شهرکی نمیفرستیم چون امیدواریم با یادگیریه phpاین مشکلات رفع شود.

الان دیگر $this->\_identity شی ای از کلاس UserIdentity است پس با ان میتوان متد authenticate() را از کلاس UserIdentity صدا زد.

if(!$this->\_identity->authenticate())

$this->addError('password','Incorrect username or password.');

متد adderror کجا تعریف شده؟

در پایان و با کد فوق بررسی کرده که اگر اعتبارسنجی با موفقیت انجام نشده بود پیغام خطا چاب شود

متد loginهم در کلاس loginform اینگونه تعریف میشود هرچند هنوز نمیدانم که کی این متد صدا زده شده؟:

public function login()

{

if($this->\_identity===null)

{

$this->\_identity=new UserIdentity($this->username,$this->password);

$this->\_identity->authenticate();

}

if($this->\_identity->errorCode===UserIdentity::ERROR\_NONE)

{

$duration=$this->rememberMe ? 3600\*24\*30 : 0; // 30 days

Yii::app()->user->login($this->\_identity,$duration);

return true;

}

else

return false;

}

که شرط دوم بررسی میکند که اگر با موفقیت login کرده چک کن که اگر ان گزینه را تیک زده برایش کوکی بساز اگر نزده هم که هیچ زمان کوکی را صفر میگذارد یعنی به مصض بشتن مرورگر هرچه هست پاک میشود. خط پایانی هم کلاس sitecontoroler متد actionlogin را صدا میزند . از روی چه حسابی خدا میداند.

کلاس useridentity به این شکل درمیاید:

<?php

/\*\*

\* UserIdentity represents the data needed to identity a user.

\* It contains the authentication method that checks if the provided

\* data can identity the user.

\*/

class UserIdentity extends CUserIdentity

{ private $\_id;

public function authenticate()

{

$user=User::model()->find('LOWER(username)=?',array(strtolower($this->username)));

if($user===null)

{

$this->errorCode=self::ERROR\_USERNAME\_INVALID;

}

elseif (!$user->validatePassword($this->password)) {

$this->errorCode=self::ERROR\_PASSWORD\_INVALID;

}

else

{

$this->\_id=$user->id;

$this->username=$user->username;

$this->setState('lastLogin',date('m/d/y g:i A',strtotime($user->last\_login\_time)));

$user->saveAttributes(

array('last\_login\_time'=>date('Y-m-d H:i:s',time()),));

$this->errorCode=self::ERROR\_NONE;

}

return $this->errorCode==self::ERROR\_NONE;

}

public function getId(){

return $this->\_id;

}

}

متد validatepassword را باید در کلاس مدل یوزر تعریف کنیم:

public function validatePassword($password){

return $this->hashPassword($password)===$this->password;

}

مرحله ی بعدی تغییر فایل index.php در پوشه ی views برای sitecontoroler است:

<?php

/\* @var $this SiteController \*/

$this->pageTitle=Yii::app()->name;

?>

<?php if (!Yii::app()->user->isGuest):?>

<p>

you last login on<?php echo Yii::app()->user->lastLogin;?>

</p>

<?php endif; ?>