بعد از انکه فایل yii را در htdocs ذخیره کردیم باید دسترسی همه ی ان را خواندن و نوشتن کنیم

یک سیستم orm(object relation mapping) در yii وجود دارد

-کنترلر داده هایی که لازم است را در اختیار view قرار میدهد

هرگاه در فایل view بگوییم $this به کنترلر مربوط به ان view دسترسی پیدا میکنیم و فکر نکنیم که با $this به view دسترسی پیدا میکنیم view هیچ ماهیت شی گرایی ندارد.

* با دستور زیر متوجه میشویم که سیستم ما حداقل تنظیمات را داراست یا نه:
* localhost/yii/requirements/index.php

هرکدام از ان قابلیت ها که در جدول فوق می ایند extention هایی هستند که میتوانیم داشته باشیم میتوانیم کلاس هایی که این extention ها ازشان ایجاد شده اند را هم در سایت یی نگاه کنیم. دو مسیر ریشه وجود دارد یکی webroot است که هماه htdocs است یعنی مسیری که سرور را نصب کرده ایم و دیگری yiiroot است که همان جایی است framework در ان قرار دارد

-راه اندازی خط فرمان yii که احتمالا نامش yiic است: برای اینکار باید وارد yiiroot شویم و این دستور را بنویسیم

Php yiic.php webapp name\_of\_projecct

گفتیم یک وب اپلیکیشن جدید میخواهیم

این قسمت(php yiic.php) دارد از طریق مفسر php فایل yiic.php را صدا میزند.

حال اگر بخواهیم ان را اجرا می کنیم.

اما اگرفایلی که برایماان در webroot ساخته میشود را نگاه کنیم

Index.php: ورودی اصلی برنامه است

Index.test.php: برای زمانی است که میخواهیم تستی انجام دهیم این کار را در سرور اصلی انحام نمیدهیم

Assets: مربوط به چیزهایی است که کاربر با ان سروکار دارد

Theme: قالب های سایت است

Protected:راقرارنیست کاربرها ببینند.

نحوه ی فعال کردن gii:

برای اعمال تغییرات روی پروژه نیاز داریم تا ابزار gii را فعال کنیم این ابزاردرواقع کد generator ماست پس وارد main.php میشویم

Protected->config->main.php

درخط 23 gii را از حالت comment خارج میکنیم. در همان خطوط میتوانیم روی این ابزار رمز بگذاریم که ما ان را false قرار میدهیم چون حوصله ی هربار رمز دادن را نداریم لازم به تذکر نیست که false یک کلمه ی رزروی است و نباید درون رشته باشد.

Crud generator: create,read,update,delete (همان query های sql )

مدل و ماژول هم میتواند برایمان بسازد که ما بعدا انگونه که میخواهیم تغییرشان میدهیم.

اکنون ما میخواهیم با این ابزار یک کنترولرجدید بسازیم نام ان message میگذاریم:

روی contoroller generator کلیک میکنیم

نام کنترولر و نام action را میدهیم. که ما با کلیک روی priview میتوانیم پیش نمایشی از ان ببینیم. میتوانیم دو action یا بیشتر ایجاد کنیم برای این کار درمستطیل پایینی نام ها را با کاما از هم جدا کنیم مثلا:

Hello,test

class MessajeController extends Controller  
{  
    public function actionHello()  
    {  
        $this->render('hello');  
    }  
  
    public function actionTest()  
    {  
        $this->render('test');  
    }

خیلی جالبه. این کدها درونش بودند. البته این مال زمانی است که ما ازمایشی دو اکشن گذاشته بودیم همان طور که میبینی نام اکشنها کلمه ی اکشن را در ابتدا دارد یعنی hello در واقع actionhello است و اما کنترلر messaje درواقع messajecontoroler است . هر کنترولری از کلاس contoroller ارث بری میکند.

حال اگرما روی گزینه ی try it now کلیک کنیم خطا میگیریم زیرا اکشنی که ما ایجاد کردیم نامش hello بود اما چون ان را set نکرده ایم دنبال اکشنه ایندکس میگرددکه ما ان را تعریف نکرده ایم. این گشتن به دنبال اکشنه ایندکس را به صورت پیش فرض انجام میدهد که ما میتوانیم ان را تغییر دهیم

به این ادرس نگاه کن:

<http://localhost/iran/index.php?r=messaje/hello>

?r=messaje: نام کنترلر را میگوید

Hello: نام view است که ایجاد کرده ایم.

میدانی این یعنی چه؟ یعنی r علامتی است که کنترلر را مشخص میکند. یعنی ابتدا نام کنترلر وسسپس نام اکشن می اید. یعنی نام کنترلر و نام اکشن بایک / از هم جدا میشوند

-view را برای action میسازند. هر اکشن میتواند view مخصوص به خودش را داشته باشد اما ما میتئانیم با براشتن تیک ان بگوییم برای این اکشن view نساز.

وقتی که ما داریم نام کنترولر را در قسمت contoroler id وارد میکنیم شاهد هستیم که مثال هایی برای ما در قسمت سمت چپ می ایند دو مثال اول که بدیهی هستند میگوید اگر بنویسی post ما برای تو postcontoroler.php را میسازیم که از این دو مثال میتوان دریافت روش نامگزاریشان برای کنترلرها paskal case است یعنی حروف اول کلمات بزرگ خواهد شد. اما در مثال سوم گفته اگر از / در مستطیل contoroler id استفاده کنی چه میشود. مثلا اگر بگویی admin/user باعث میشود در پوشه ی ادمین یک usercontoroler ساخته شود البته به شرطی که ماژول ادمین را ساخته و فعال کرده باشیم در غیر اینصورت فقط کنترلری به نام usercontoriler میسازد و ان قسمت admin/ را نادیده میگیرد اما من چون فعلا علاقه ای به کار با ادمین ندارم از پیگیری ان صرف نظر میکنم. اما اینقدر دانستم که در پوشه ی protected فایلی وجود دارد به ایم views که وقتی درون ان را نگاه کردم فولدری با همان نام کنترلری که خودم ایجاد کرده بودم یعنی message مشاهده کردم و درون ان فایلی با نام hello.php قرار داشت. خیلی جالب است که دارم این ها را یاد میگیرم.

در پوشه ی contorolers دو کنترولر تا کنون داریم یکی sitecontoroller.php که پیش فرض است و دیگری messajecontoroller.php که خودمان ساخته ایم. حال یک بار دیگر به این ادرس دقت کن:

<http://localhost/iran/index.php?r=messaje/hello>

اگر ان را اینگونه بنویسیم:

<http://localhost/iran/index.php>

کنترولر پیشفرض سایت کنترولر و اکشنه پیشفرض ایندکس است.اما اگر نام کنترلر را ذکر بکنیم بدیهی است که سراغ ان میرود.

وارد فولدر message میشویم و ان را باز میکنیم میبینیم که متدی به نام actionhello دارد نام همه ی متدهایی که یک اکشن هستند باید با کلمه ی اکشن همراه باشد و گرنه roter نمیتواند ان را پیدا کند .

تمام کنترولرها باید از کلاس contoroller مشتق شده باشند صرف این که در پوشه ی کنترلرباشند کفایت نمیکند.

خوب حال ما برای تغییر اکشن پیشفرض در کنترولر message این دستور را مینویسیم:

$defaultAction=’hello’;

تا بشود:

class MessajeController extends Controller

{ public $defaultAction='hello';

public function actionHello()

{

$this->render('hello');

}

به این که درکلمه ی default بعد از fحرف a قراردارد دقت نکردیم و خطا گرفتیم.

من فکر میکنیم نوشتن این دستور درواقع سایه زدن متغیر $defaultAction است هرچند فعلا به کد کلاس contoroller که کلاس messagecontoroller از ان مشتق شده دسترسی ندارم .

مطلب دیگر اینکه اگر ?r= باعث میشود که کنترولر ها صدازده شوند پس gii هم یک کنترولراست مثل کنترولرهایی که خودمان نوشته ایم.

دقیقه ی چهل و دو هستیم