Episode 5 - Generating CRUD code with Gii

What is CRUD

CRUD stands for Create, Read, Update, and Delete, representing the four common tasks taken with data on most Web sites. CRUD Generator generates a controller and views that implement CRUD operations for the specified data model.

Generating CRUD code

We can generate CRUD code directly in the Gii code generator. To go to CRUD generator click on the "CRUD Generator" button or go to this address:

http://localhost/blog/frontend/web/gii/crud

There are several form fields which we have to fill:

Model class - This is the ActiveRecord class associated with the table that CRUD will be built upon.

Search Model Class - This is the name of the search model class to be generated.

Controller Class - This is the name of the controller class to be generated.

View Path - Specify the directory for storing the view scripts for the controller.

Widget Used in Index Page - This is the widget type to be used in the index page to display list of the models.

Generating CRUD code for Posts

To generate CRUD code for Post model, fill the form with:

 $Model \ Class \ \hbox{-} \ \texttt{common} \\ \verb|\mbox{models} \\ \verb|\mbox{Post}|$

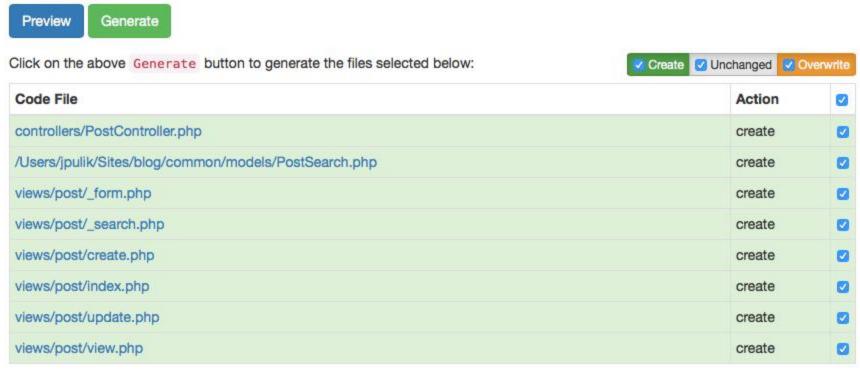
 $Search\ Model\ Class\ \hbox{-}\ \texttt{common}\\ \verb|\mbox{models}\\ \verb|\mbox{PostSearch}|$

 $Controller\ Class\ \hbox{-}\ \texttt{frontend} \verb|\controllers|| PostController$

View Path - @frontend/views/post

Widget Used in Index Page - ListView

After filling the fields, click on the "Preview" button and you should see:



Then click

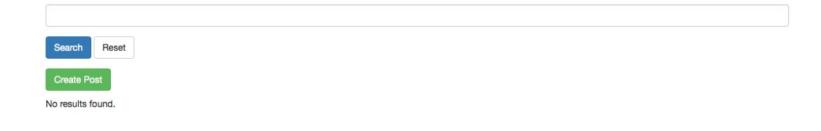
on the "Generate" button to generate CRUD code.

Now go to this address:

http://localhost/blog/frontend/web/post/index

and you should see our newly generated Posts index page:







We will customize generated code in the next episode. In this episode we will only **generate** necessary CRUD code for the models which we generated in the previous episode.

Before we will continue, create new .htaccess file in backend\web\ to route backend requests properly:

```
Options +FollowSymLinks

IndexIgnore */*

RewriteEngine on

RewriteCond %{REQUEST_FILENAME} !-f

RewriteCond %{REQUEST_FILENAME} !-d

RewriteRule . index.php
```

We will do the same for the Category and Tag models.

Generating CRUD code for Categories

```
Fill the Gii form with: Model Class - common\models\Category

Search Model Class - common\models\CategorySearch

Controller Class - backend\controllers\CategoryController
```

View Path - @backend/views/category

Widget Used in Index Page - GridView

Categories index page:

http://localhost/blog/backend/web/index.php?r=category%2Findex

We have generated CRUD code for Categories to the backend because we want to manage categories from there.

Generating CRUD code for Tags

Fill the Gii form with: Model Class - common\models\Tag

Search Model Class - common\models\TagSearch

 $Controller\ Class\ \hbox{-}\ {\tt backend} \verb|\controllers| {\tt TagController}|$

View Path - @backend/views/tag

Widget Used in Index Page - GridView

Tags index page:

http://localhost/blog/backend/web/index.php?r=tag%2Findex

We have also generated CRUD code for Tags to the backend because we want to manage tags from there.

In this episode we have successfully generated all CRUD code which we will need in our blog. Generated CRUD code is just the skeleton (avoids DRY) of the real code which we want to have. So we will customize generated code in the next episodes to bring all the functionality we want. If do you have any questions regarding to this episode, please write them below to the comments section.

Download files from this episode: episode_o5.zip.