Customizing the ActiveQuery class

By default, all Active Record queries are supported by yii\db\ActiveQuery. To use a customized query class in an Active Record class, you should override the yii\db\ActiveRecord: : find() method and return an instance of your customized query class.

Getting ready

1. Create a new application using the Composer package manager, as described in the official guide at <http://www.yiiframework.com/doc-2.0/guide-start-installation.html>.
2. Set up the database connection and create a table named post, as follows:

DROP TABLE IF EXISTS 'post';

CREATE TABLE IF NOT EXISTS 'post' (

'id' INT(10) UNSIGNED NOT NULL AUTO\_INCREMENT,

'lang' VARCHAR(5) NOT NULL DEFAULT 'en',

'title' VARCHAR(255) NOT NULL,

'text' TEXT NOT NULL,

PRIMARY KEY ('id')

);

INSERT INTO 'post'('id','lang','title','text')

VALUES (1,'en\_us','Yii news','Text in English'),

(2,'de','Yii Nachrichten','Text in Deutsch');

1. Generate a Post model using Gii with an enabled Generate ActiveQuery option that generates the PostQuery class.

How to do it...

1. Add the following method to models/PostQuery .php:

<?php

namespace app\models;

/[[1]](#footnote-1) [[2]](#footnote-2)

* This is the ActiveQuery class for [[Post]].

\*

\* @see Post \*/

class PostQuery extends \yii\db\ActiveQuery {

/\*\*

namespace app\controllers;

use app\models\Post; use yii\helpers\Html; use yii\web\Controller;

/\*\*

* Class DbController.
* @package app\controllers \*/

class DbController extends Controller {

public function actionIndex()

{

// Get posts written in default application language $posts = Post::find()->all();

echo Html::tag('h1', 'Default language'); foreach ($posts as $post) {

echo Html::tag('h2', $post->title); echo $post->text;

}

// Get posts written in German

$posts = Post::find()->lang('de')->all();

echo Html::tag('h1', 'German'); foreach ($posts as $post) {

echo Html::tag('h2', $post->title); echo $post->text;

}

}

}

3. Now, run db/index and you should get an output similar to the one shown in the following screenshot:



How it works...

We have rewritten the find method in the Post model and extended the ActiveQuery class. The lang method returns ActiveQuery with the specified language value. In order to support chained calls, lang returns the model instance by itself.

There’s more.

According to the Yii2 Guide, in Yii 1.1, there was a concept called scope. Scope is no longer directly supported in Yii 2.0, and you should use customized query classes and query methods to achieve the same goal.

See also

For further information, refer to the following URLs:

* [http://www.yiiframework.com/doc-2.0/guide-db-active-record.html#cnstomi7ing-query-classes](http://www.yiiframework.com/doc-2.0/guide-db-active-record.html%23customizing-query-classes)
* [http://www.yiiframework.com/doc-2.0/guide-intro-upgrade-from-v1.html#active-record](http://www.yiiframework.com/doc-2.0/guide-intro-upgrade-from-v1.html%23active-record)

1. @param $lang

   \*

   \* @return $this \*/

   public function lang($lang) [↑](#footnote-ref-1)
2. {

   return $this->where([ 'lang' => $lang ]);

   }

   }

   1. That is it. Now, we can use our model. Create controllers/DbController.php as follows:

   <?php [↑](#footnote-ref-2)