Processing model fields with AR event-like methods

Active Record implementation in Yii is very powerful and has many features. One of these features is the event-like methods, which you can use to preprocess model fields before putting them into the database or getting them from a database, as well as to delete data related to the model, and so on.

In this recipe, we will link all URLs in the post text and list all existing Active Record event-like methods.

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-jnstallation.html](http://www.yiiframework.com/doc-2.0/guide-start-installation.html).

1. 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,

'title' VARCHAR(255) NOT NULL,

'text' TEXT NOT NULL,

PRIMARY KEY ('id')

);

1. Generate the post model using Gii.

How to do it...

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

/\*\*

* @param bool $insert

\*

* @return bool \*/

public function beforeSave($insert)

{

$this->text = preg\_replace('~((?:https?|ftps?)://.\*?)( |$)~iu',

'<a href="\1">\1</a>\2', $this->text);

return parent::beforeSave($insert);

}

1. That is it. Now, try saving a post containing a link. Create controllers/TestController. php as follows:

<?php

namespace app\controllers;

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

/\*\*

* Class TestController.
* @package app\controllers

\*/

class TestController extends Controller {

public function actionIndex()

{

$post = new Post();

$post->title = 'links test';

$post->text = 'before <http://www.yiiframework.com/> after'; $post->save();

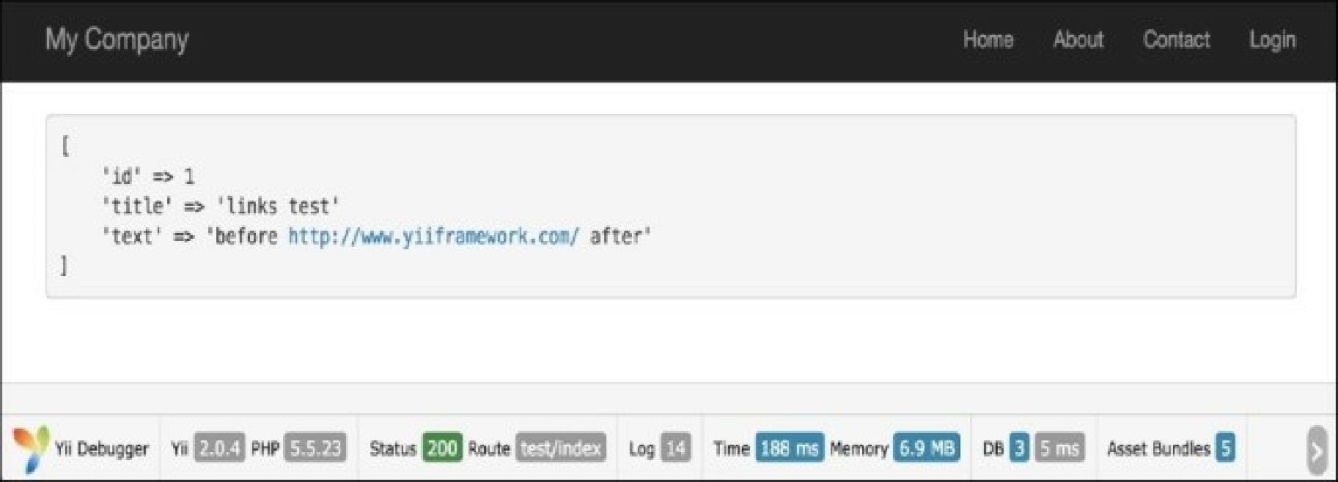
return $this->renderContent(Html::tag('pre', VarDumper::dumpAsString( $post->attributes

)));

}

}

1. That is it. Now, run test/index. You should get the following result:



How it works...

The beforeSave method is implemented in the ActiveRecord class and executed just before saving a model. Using a regular expression, we replace everything that looks like a URL with a link that uses this URL and call the parent implementation, so that real events are raised properly. In order to prevent saving, you can return false.

See also

* [For further information, refer to http://www.yiiframework.com/doc-2.0/guide-db-active-](http://www.yiiframework.com/doc-2.0/guide-db-active-record.html%23active-record-life-cycles) record.html#active-record-life-cycles.
* The Working with events recipe in Chapter 1, Fundamentals
* The Automating timestamps recipe
* The Setting up an author automatically recipe
* The Setting up a slug automatically recipe