Working with events

Yii’s events provide a simple implementation, which allows you to listen and subscribe to various events that occur in your web-application. For example, you may wish to send a notification about a new article to followers each time you publish new material.

Getting ready

1. Create a new application by using the Composer package manager, as described in the official guide at [http://www. yiiframework. c om/doc-2.0/guide -start-installation .html](http://www.yiiframework.com/doc-2.0/guide-start-installation.html).

2. Execute the following SQL code on your server to create the article table:

CREATE TABLE 'article' (

'id' int(11) NOT NULL AUTO\_INCREMENT,

'name' varchar(255) DEFAULT NULL,

'description' text,

PRIMARY KEY ('id')

) ENGINE=InnoDB AUTO\_INCREMENT=29 DEFAULT CHARSET=utf8;

3. Generate the Article model using Gii.

4. Run your webserver by ./yii serve command.

How to do it...

1. Add an action test to \controllers\SiteController:

public function actionTest()

{

$article = new Article();

$article->name = 'Valentine\'s Day\'s coming? Aw crap! I forgot to get a girlfriend again!';

$article->description = 'Bender is angry at Fry for dating a robot. Stay away from our women.

You've got metal fever, boy. Metal fever';

// $event is an object of yii\base\Event or a child class $article->on(ActiveRecord::EVENT\_AFTER\_INSERT, function($event) {

$followers = ['john2@teleworm.us', 'shivawhite@cuvox.de', 'kate@dayrep.com'

];

foreach($followers as $follower) {

Yii::$app->mailer->compose()

->setFrom('techblog@teleworm.us')

->setTo($follower)

->setSubject($event->sender ->name)

->setTextBody($event->sender->description)

->send();

}

echo 'Emails has been sent';

});

if (!$article->save()) {

echo VarDumper::dumpAsString($article->getErrors());

};

}

2. Update the config/web.php component mailer using the following code.