Encrypting/Decrypting data

The Yii2 framework contains a special security component that provides a set of methods for handling common security-related tasks. The \yii\base\security class requires the openssL PHP extension instead of mcrypt.

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. Set up the database connection and create a table named order, as follows:

DROP TABLE IF EXISTS 'order';

CREATE TABLE IF NOT EXISTS 'order' (

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

'client' VARCHAR(255) NOT NULL,

'total' FLOAT NOT NULL,

'encrypted\_field' BLOB NOT NULL,

PRIMARY KEY ('id')

);

1. Generate an Order model using Gii.

How to do it...

1. Add an additional key parameter to config/params. php, as follows:

<?php return [

'adminEmail' => 'admin@example.com',

'key' => 'mysecretkey'

];

1. Add the behaviors and helper properties to the Order model as follows:

public $encrypted\_field\_temp;

public function behaviors()

{

return [

[

'class' => AttributeBehavior::className(),

'attributes' => [

ActiveRecord::EVENT\_BEFORE\_INSERT => 'encrypted\_field', ActiveRecord::EVENT\_BEFORE\_UPDATE => 'encrypted\_field',

],

'value' => function ($event) {

$event->sender->encrypted\_field\_temp = $event->sender- >encrypted\_field;

return Yii::$app->security->encryptByKey(  
$event->sender->encrypted\_field,

Yii::$app->params['key']

);

},

],

[

'class' => AttributeBehavior::className(),