Setting up an author automatically

The Blameable behavior allows you to update one or more authors’ fields automatically. This is primarily used to populate data into the created\_by and updated\_by fields. Similar to the Timestamp behavior, you can easily specify some special parameters and essential events for this behavior.

Let us return to the example from the previous section. We also have posts in our blog application. For example, let’s assume that our blog model is called Blogpost. The model has author\_id, the field which points to who created this post, and updater\_id, the field which points to who updated it. We would like to populate these attributes automatically during the create/update model events. Now you can learn how to do it.

Getting ready

1. Create a new application using the Composer package manager, as described in the official guide at

[http://www.viiframework.com/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 blog\_post, as follows:

DROP TABLE IF EXISTS 'blog\_post';

CREATE TABLE IF NOT EXISTS 'blog\_post' (

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

'author\_id' INT(10) UNSIGNED DEFAULT NULL,

'updater\_id' INT(10) UNSIGNED DEFAULT NULL,

'title' VARCHAR(255) NOT NULL,

'text' TEXT NOT NULL,

PRIMARY KEY ('id')

);

3. Use Gii to create the BlogPost model for the blost\_post table.

How to do it...

1. Add the following behaviors method to models/BlogPost. php: <?php

namespace app\models; use Yii;

use yii\db\BaseActiveRecord;

/\*\*

\* This is the model class for table "blog\_post".

\*

\* @property integer $id

\* @property integer $author\_id

\* @property integer $updater\_id

\* @property string $title

\* @property string $text \*/

class BlogPost extends \yii\db\ActiveRecord {

/\*\*

\* @return array \*/