Using standalone actions

In Yii, you can define controller actions as separate classes and then connect them to your controllers.

This will help you to reuse some common functionality.

For example, you can move the backend for autocomplete fields to an action and save some time by not having to write it over and over again.

Another example is that we can create all CRUD operations as separate standalone actions. We will write, create, view, and delete operations of the model and view the list operation of models.

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. Let’s create post table. Create migration for this using the following command:

./yii migrate/create create\_post\_table

3. Update the just-created migration’s methods and list of imported classes as follows:

<?php

use yii\db\Schema; use yii\db\Migration;

class m150719\_152435\_create\_post\_table extends Migration {

const TABLE\_NAME = '{{%post}}';

public function up()

{

$tableOptions = null;

if ($this->db->driverName === 'mysql') {

$tableOptions = 'CHARACTER SET utf8 COLLATE utf8\_general\_ci ENGINE=InnoDB';

}

$this->createTable(self::TABLE\_NAME, [

'id' => Schema::TYPE\_PK,

'title' => Schema::TYPE\_STRING.'(255) NOT NULL',

'content' => Schema::TYPE\_TEXT.' NOT NULL',

], $tableOptions);

for ($i = 1; $i < 7; $i++) {

$this->insert(self::TABLE\_NAME, [

'title' => 'Test article #'.$i,

'content' => 'Lorem ipsum dolor sit amet, consectetur adipiscing

elit. '

.'Sed sit amet mauris est. Sed at dignissim dui. '

.'Phasellus arcu massa, facilisis a fringilla sit amet, '

. ' rhoncus ut enim.',

]);

}

}

public function down()