Writing extension tests

To conduct the extension test, follow these steps:

1. Add the book/cart/tests/bootstrap.php entry script for PHPUnit:

<?php

defined('YII\_DEBUG') or define('YII\_DEBUG', true); defined('YII\_ENV') or define('YII\_ENV', 'test');

require( DIR . '/../vendor/autoload.php');

require( DIR . '/../vendor/yiisoft/yii2/Yii.php');

1. Create a test base class by initializing the Yii application before each test and by destroying the application afterwards:

<?php

namespace book\cart\tests;

use yii\di\Container; use yii\web\Application;

abstract class TestCase extends \PHPUnit\_Framework\_TestCase {

protected function setUp()

{

parent::setUp();

$this->mockApplication();

}

protected function tearDown()

{

$this->destroyApplication(); parent::tearDown();

}

protected function mockApplication()

{

new Application([

'id' => 'testapp',

'basePath' => DIR ,

'vendorPath' => dirname( DIR ) . '/vendor',

]);

}

protected function destroyApplication()

{

\Yii::$app = null;

\Yii::$container = new Container();

}

}

1. Add a memory-based clean fake class that implements the StorageInterface interface:

<?php

namespace book\cart\tests\storage;

use book\cart\storage\StorageInterface;

class FakeStorage implements StorageInterface