{

private $items = [];

public function load()

{

return $this->items;

}

public function save(array $items) {

$this->items = $items;

}

}

It will store items into a private variable instead of working with a real session. It allows to run tests independently (without real storage driver) and also improves testing performance.

4. Add the cartTest class:

<?php

namespace book\cart\tests; use book\cart\Cart;

use book\cart\tests\storage\FakeStorage;

class CartTest extends TestCase {

/\*\*

\* @var Cart \*/

private $cart;

public function setUp()

{

parent::setUp();

$this->cart = new Cart(['storage' => new FakeStorage()]);

}

public function testEmpty()

{

$this->assertEquals([], $this->cart->getItems()); $this->assertEquals(0, $this->cart->getCount()); $this->assertEquals(0, $this->cart->getAmount());

} public function testAdd()

{

$this->cart->add(5, 3);

$this->assertEquals([5 => 3], $this->cart->getItems()); $this->cart->add(7, 14);

$this->assertEquals([5 => 3, 7 => 14], $this->cart->getItems()); $this->cart->add(5, 10);

$this->assertEquals([5 => 13, 7 => 14], $this->cart->getItems());

} public function testSet()

{

$this->cart->add(5, 3);

$this->cart->add(7, 14);

$this->cart->set(5, 12);

$this->assertEquals([5 => 12, 7 => 14], $this->cart->getItems());

}