PHPUnit\_Framework\_Assert::assertArraySubset(

[intval($product) => intval($pieces)],

{

$this->storage->load()

);

}

private function mockApplication()

{

Yii::$container = new Container(); new Application([

'id' => 'testapp',

'basePath' => DIR ,

'vendorPath' => DIR . '/../../vendor',

]);

}

}

1. Add the features/bootstrap/CartContext/FakeStorage . php file with a fake storage class: <?php

namespace book\cart\features\bootstrap\storage;

use book\cart\storage\StorageInterface;

class FakeStorage implements StorageInterface {

private $items = [];

public function load()

{

return $this->items;

}

public function save(array $items)

{

$this->items = $items;

}

}

1. Add book/cart/behat.yml with contexts definition:

default:

suites:

default:

contexts:

* CartContext
* StorageContext

1. Now we will get the following structure:

book 1 cart

* src

storage

I— SessionStorage.php I— StorageInterface.php Cart.php

* features

bootstrap

* storage

1 FakeStorage.php

* bootstrap.php
* CartContext.php
* StorageContext.php