if ($this->\_items === null) {

$this->\_items = $this->\_storage->load();

}

}

private function saveItems()

{

$this->\_storage->save($this->\_items);

}

}

2. Create Storageinterface interface in the src/storage subdirectory: <?php

namespace book\cart\storage;

interface StorageInterface {

/\*\*

\* @return array \*/

public function load();

/\*\*

\* @param array $items \*/

public function save(array $items);

}

and SessionStorage class: namespace book\cart\storage; use Yii;

class SessionStorage implements StorageInterface {

public $sessionKey = 'cart';

public function load()

{

return Yii::$app->session->get($this->sessionKey, []);

}

public function save(array $items)

{

Yii::$app->session->set($this->sessionKey, $items);

}

}

3. Now we must get the following structure:

book 1 cart

— src

storage

I— SessionStorage.php I— Storageinterface.php Cart.php

— tests

— .gitignore

— composer.j son

— phpunit.xml.dist

— vendor