Unit Test Documentation

SaltOS 4.0 r761

March 2024

Contents

1 Lib

1.1 Autoload file for the unit tests

`lib/autoload.php`

This file contains the code that initialize the unit tests

1.1.1 Main autoloader code

This code emmulates the index.php by loading all autoload files excep the zindex.php, initialize the timer, the random and executes the check_system function.

1.2 CLI helper function

`lib/clilib.php`

This file contains the function used by the web unit tests to communicate with the SaltOS app, using the two interfaces that SaltOS provides.

1.2.1 Test CLI helper

`function test_cli_helper($rest, $data, $token): array`

This function allow to execute SaltOS using the CLI SAPI, to do it, the function detects if data is provided, and executes the command and getting the output of the execution. If data exists, then the contents are stored in a file and passed the contents of the file to the stdin of the php process to emmulate the input channel used by the apache server.

As an example, this functions tries to execute the command using the follow formula:

1) php index.php $rest

2) cat /tmp/input | php index.php $rest

In addition, the token field is used to define the TOKEN environment variable that is used by SaltOS as variable to emmulate the TOKEN used by the apache for authenticate the SaltOS app.

- @rest => The rest request, like update/customers/3
- @data => The data used as json in the SaltOS app
- @token => The token used if authentication is required

1.3 WEB helper function

`lib/weblib.php`

This file contains the function used by the web unit tests to communicate with the SaltOS app, using the two interfaces that SaltOS provides.

1.3.1 Test WEB helper

```
function test_web_helper($rest, $data, $token): array
```

This function performs the action defined by the rest verb sendind the data if it is provided and using the token for authentication actions.

As you can see in the code, the function detects if data is provided and send the request using GET or POST, in addition, an application/json content-type header is send when POST is used.

The token is sent using the TOKEN header to be used in the authentication process.

- @rest => The rest request, like update/customers/3
- @data => The data used as json in the SaltOS app
- @token => The token used if authentication is required

2 Utest

2.1 Test cli customers

```
test_cli_customers.php
```

This test performs all actions of the customers app suck as: create, insert, list, view, edit, update and delete, using the cli sapi interface

2.1.1 Importing namespaces

```
use PHPUnit\Framework\TestCase;
```

2.1.2 Loading helper function

```
require_once "lib/clilib.php";
```

This file contains the needed function used by the unit tests

2.1.3 Main class of this unit test

```
final class test_cli_customers extends TestCase
```

2.1.4 Create

```
public function test_create(array $json): array
```

This function execute the creates rest request, and must to get the json with the layout without data

2.1.5 Insert

```
public function test_insert(array $json): array
```

This function execute the insert rest request, to do it send the json with the data that they want to insert and must to get the json with the status and the create_id.

2.1.6 List

```
public function test_list(array $json): array
```

This function execute the list rest request, to do it send the json with the search that they want to use in the list filter and receives the json with the data used to populate the table.

2.1.7 View

```
public function test_view(array $json): array
```

This function execute the view rest request, intended to retrieve the detail of the app with the layout needed to render it.

2.1.8 Edit

```
public function test_edit(array $json): array
```

This function execute the view rest request, intended to retrieve the detail of the app with the layout needed to render it.

2.1.9 Upgrade

```
public function test_update(array $json): array
```

This function execute the update rest request, to do it send the json with the data that they want to update and must to get the json with the status and the updated_id.

2.1.10 Delete

```
public function test_delete(array $json): void
```

This function execute the delete rest request, they must to get the json with the status and the deleted_id.

2.2 Test cli invoices

```
test_cli_invoices.php
```

This test performs all actions of the invoices app suck as: create, insert, list, view, edit, update and delete, using the cli sapi interface

2.2.1 Importing namespaces

```
use PHPUnit\Framework\TestCase;
```

2.2.2 Loading helper function

```
require_once "lib/clilib.php";
```

This file contains the needed function used by the unit tests

2.2.3 Main class of this unit test

```
final class test_cli_invoices extends TestCase
```

2.2.4 Create

```
public function test_create(array $json): array
```

This function execute the creates rest request, and must to get the json with the layout without data

2.2.5 Insert

```
public function test_insert(array $json): array
```

This function execute the insert rest request, to do it send the json with the data that they want to insert and must to get the json with the status and the create_id.

2.2.6 List

```
public function test_list(array $json): array
```

This function execute the list rest request, to do it send the json with the search that they want to use in the list filter and receives the json with the data used to populate the table.

2.2.7 View

```
public function test_view(array $json): array
```

This function execute the view rest request, intended to retrieve the detail of the app with the layout needed to render it.

2.2.8 Edit

```
public function test_edit(array $json): array
```

This function execute the view rest request, intended to retrieve the detail of the app with the layout needed to render it.

2.2.9 Upgrade

```
public function test_update(array $json): array
```

This function execute the update rest request, to do it send the json with the data that they want to update and must to get the json with the status and the updated_id.

2.2.10 Delete

```
public function test_delete(array $json): void
```

This function execute the delete rest request, they must to get the json with the status and the deleted_id.

2.3 Test cli tokens (first part)

```
test_cli_tokens1.php
```

This test performs some part of the actions related with the tokens suck as authtoken and checktoken, using the cli sapi interface

2.3.1 Importing namespaces

```
use PHPUnit\Framework\TestCase;
```

2.3.2 Loading helper function

```
require_once "lib/clilib.php";
```

This file contains the needed function used by the unit tests

2.3.3 Main class of this unit test

```
final class test_cli_tokens1 extends TestCase
```

2.3.4 Authtoken

```
public function test_authtoken(): array
```

This function execute the authtoken rest request, and must to get the json with the valid token to continue in the nexts unit tests

2.3.5 Checktoken

```
public function test_checktoken(array $json): array
```

This function execute the checktoken rest request, and must to get the json with the ok about the valid token that you are trying to check

2.4 Test cli tokens (second part)

```
test_cli_tokens2.php
```

This test performs some part of the actions related with the tokens suck as deauthtoken and checktoken, using the cli sapi interface

2.4.1 Importing namespaces

```
use PHPUnit\Framework\TestCase;
```

2.4.2 Loading helper function

```
require_once "lib/clilib.php";
```

This file contains the needed function used by the unit tests

2.4.3 Main class of this unit test

```
final class test_cli_tokens2 extends TestCase
```

2.4.4 Deauthtoken

```
public function test_deauthtoken(array $json): array
```

This function execute the deauthtoken rest request, and must to get the json with the ok about the valid token that you are deauthenticate

2.4.5 Checktoken ko

```
public function test_checktoken_ko(array $json): void
```

This function execute the checktoken rest request, and must to get the json with the ko about the invalid token that you are trying to check

2.5 Test database drivers

```
test_database.php
```

2.5.1 Importing namespaces

```
use PHPUnit\Framework\TestCase;
```

2.5.2 Main class of this unit test

```
final class test_database extends TestCase
```

2.5.3 Helper

```
private function test_helper($obj): void
```

This function executes the follow queries and checks the correctness of the driver by comparing the results with the expected results.

The tests that performs are the follow:

- SELECT GROUP_CONCAT(a) test FROM (SELECT 1 a UNION SELECT 2 a UNION SELECT 3 a) a;
- SELECT REPLACE('abc', 'b', 'c') test
- SELECT LPAD('123', '5', '0') test
- SELECT CONCAT('a', 'b', 'c') test
- SELECT CONCAT_WS(',','a','b','c',null,true,false) test
- SELECT UNIX_TIMESTAMP('2024-02-01 12:34:56') test
- SELECT FROM_UNIXTIME(1706787296) test
- SELECT YEAR('2024-02-01 12:34:56') test
- SELECT MONTH('2024-02-01 12:34:56') test
- SELECT WEEK('2024-02-01 12:34:56', 1) test
- SELECT TRUNCATE(1.2345, 2) test
- SELECT DAY('2024-02-01 12:34:56') test
- SELECT DAYOFYEAR('2024-02-01 12:34:56') test
- SELECT DAYOFWEEK('2024-02-01 12:34:56') test
- SELECT HOUR('2024-02-01 12:34:56') test
- SELECT MINUTE('2024-02-01 12:34:56') test
- SELECT SECOND('2024-02-01 12:34:56') test
- SELECT MD5('fortuna') test
- SELECT REPEAT('abc',3) test
- SELECT FIND_IN_SET(3,'1,2,3,4,5') test
- SELECT FIND_IN_SET(6,'1,2,3,4,5') test
- SELECT FIND_IN_SET(3,'12345') test
- SELECT IF(true, 'ok', 'ko') test
- SELECT IF(false, 'ok', 'ko') test
- SELECT IF(null, 'ok', 'ko') test
- SELECT POW(2, 8) test
- SELECT DATE_FORMAT('2024-02-01 12:34:56', '%Y-%m-%d %H:%i:%s') test
- SELECT NOW() test

2.5.4 PDO MySQL driver

```
public function test_pdo_mysql(): void
```

This function checks the correctness of the sqlite3 driver by creating a database connection, sendint queries validating the expected results and closing the connection.

2.5.5 MySQL improved driver

```
public function test_mysqli(): void
```

This function checks the correctness of the sqlite3 driver by creating a database connection, sendint queries validating the expected results and closing the connection.

2.5.6 PDO SQLite driver

```
public function test_pdo_sqlite(): void
```

This function checks the correctness of the sqlite3 driver by creating a database connection, sendint queries validating the expected results and closing the connection.

2.5.7 SQLite3 driver

```
public function test_sqlite3(): void
```

This function checks the correctness of the sqlite3 driver by creating a database connection, sendint queries validating the expected results and closing the connection.

2.6 Test roundcube library

```
test_roundcube.php
```

2.6.1 Importing namespaces

```
use PHPUnit\Framework\TestCase;
```

2.6.2 Main class of this unit test

```
final class test_roundcube extends TestCase
```

2.6.3 html2text

```
public function test_html2text(): void
```

This function checks the correctness of the html2text method provided by the roundcube library.

2.7 Test web customers

```
test_web_customers.php
```

This test performs all actions of the customers app suck as: create, insert, list, view, edit, update and delete, using the apache sapi interface

2.7.1 Importing namespaces

```
use PHPUnit\Framework\TestCase;
```

2.7.2 Loading helper function

```
require_once "lib/weblib.php";
```

This file contains the needed function used by the unit tests

2.7.3 Main class of this unit test

```
final class test_web_customers extends TestCase
```

2.7.4 Create

```
public function test_create(array $json): array
```

This function execute the creates rest request, and must to get the json with the layout without data

2.7.5 Insert

```
public function test_insert(array $json): array
```

This function execute the insert rest request, to do it send the json with the data that they want to insert and must to get the json with the status and the create_id.

2.7.6 List

```
public function test_list(array $json): array
```

This function execute the list rest request, to do it send the json with the search that they want to use in the list filter and receives the json with the data used to populate the table.

2.7.7 View

```
public function test_view(array $json): array
```

This function execute the view rest request, intended to retrieve the detail of the app with the layout needed to render it.

2.7.8 Edit

```
public function test_edit(array $json): array
```

This function execute the view rest request, intended to retrieve the detail of the app with the layout needed to render it.

2.7.9 Upgrade

```
public function test_update(array $json): array
```

This function execute the update rest request, to do it send the json with the data that they want to update and must to get the json with the status and the updated_id.

2.7.10 Delete

```
public function test_delete(array $json): void
```

This function execute the delete rest request, they must to get the json with the status and the deleted_id.

2.8 Test web invoices

```
test_web_invoices.php
```

This test performs all actions of the invoices app suck as: create, insert, list, view, edit, update and delete, using the apache sapi interface

2.8.1 Importing namespaces

```
use PHPUnit\Framework\TestCase;
```

2.8.2 Loading helper function

```
require_once "lib/weblib.php";
```

This file contains the needed function used by the unit tests

2.8.3 Main class of this unit test

```
final class test_web_invoices extends TestCase
```

2.8.4 Create

```
public function test_create(array $json): array
```

This function execute the creates rest request, and must to get the json with the layout without data

2.8.5 Insert

```
public function test_insert(array $json): array
```

This function execute the insert rest request, to do it send the json with the data that they want to insert and must to get the json with the status and the create_id.

2.8.6 List

```
public function test_list(array $json): array
```

This function execute the list rest request, to do it send the json with the search that they want to use in the list filter and receives the json with the data used to populate the table.

2.8.7 View

```
public function test_view(array $json): array
```

This function execute the view rest request, intended to retrieve the detail of the app with the layout needed to render it.

2.8.8 Edit

```
public function test_edit(array $json): array
```

This function execute the view rest request, intended to retrieve the detail of the app with the layout needed to render it.

2.8.9 Upgrade

```
public function test_update(array $json): array
```

This function execute the update rest request, to do it send the json with the data that they want to update and must to get the json with the status and the updated_id.

2.8.10 Delete

```
public function test_delete(array $json): void
```

This function execute the delete rest request, they must to get the json with the status and the deleted_id.

2.9 Test web tokens (first part)

```
test_web_tokens1.php
```

This test performs some part of the actions related with the tokens suck as authtoken and checktoken, using the apache sapi interface

2.9.1 Importing namespaces

```
use PHPUnit\Framework\TestCase;
```

2.9.2 Loading helper function

```
require_once "lib/weblib.php";
```

This file contains the needed function used by the unit tests

2.9.3 Main class of this unit test

```
final class test_web_tokens1 extends TestCase
```

2.9.4 Authtoken

```
public function test_authtoken(): array
```

This function execute the authtoken rest request, and must to get the json with the valid token to continue in the nexts unit tests

2.9.5 Checktoken

```
public function test_checktoken(array $json): array
```

This function execute the checktoken rest request, and must to get the json with the ok about the valid token that you are trying to check

2.10 Test web tokens (second part)

```
test_web_tokens2.php
```

This test performs some part of the actions related with the tokens suck as deauthtoken and checktoken, using the apache sapi interface

2.10.1 Importing namespaces

```
use PHPUnit\Framework\TestCase;
```

2.10.2 Loading helper function

```
require_once "lib/weblib.php";
```

This file contains the needed function used by the unit tests

2.10.3 Main class of this unit test

```
final class test_web_tokens2 extends TestCase
```

2.10.4 Deauthtoken

```
public function test_deauthtoken(array $json): array
```

This function execute the deauthtoken rest request, and must to get the json with the ok about the valid token that you are deauthenticate

2.10.5 Checktoken ko

```
public function test_checktoken_ko(array $json): void
```

This function execute the checktoken rest request, and must to get the json with the ko about the invalid token that you are trying to check