



Codeception

p e n t a l o g . f r

What is Codeception

Automated testing == Java ? Maybe not

Project was started in 2011

It solves the following :

- Bridge between all testing types
- No other language required (only php)
- Extensible
- Covers major frameworks

It's built on top of...

- PHPUnit
- Symfony browser kit
- Selenium

Sounds familiar ? :)

Aimed to be

- Easy to read
- Easy to write
- Easy to debug

How easy you ask ?

This easy

Judge by code

```
$I = new WebGuy($scenario)
$I->wantTo('sign up as a customer');
$I->click('Sign Up');
$I->fillField('name','Michael');
$I->fillField('password','qwerty');
$I->click('Submit');
$I->see('Hello, Michael!');
```

Types of testing

- Acceptance
- Functional
- Unit
- Api

Acceptance Testing

Acceptance testing

- PhpBrowser
- PhantomJs
- Selenium

```
I WANT TO SIGN IN
I am on page '/login'
I fill field 'username', 'davert'
I fill field 'password', 'qwerty'
I click 'LOGIN'
I see 'Welcome, Davert!'
```

```
<?php
$I = new AcceptanceTester($scenario);
$I->wantTo('sign in');
$I->amOnPage('/login');
$I->fillField('username', 'davert');
$I->fillField('password', 'qwerty');
$I->click('LOGIN');
$I->see('Welcome, Davert!');
?>
```


PhpBrowser

This is **the fastest way** to run acceptance tests, since **it doesn't require running an actual browser**. We use a PHP web scrapper, which acts like a browser:
it **sends a request**, then **receives and parses the response**.

```
yaml
class_name: AcceptanceTester
modules:
  enabled:
    - PhpBrowser
    - AcceptanceHelper
    - Db
  config:
    PhpBrowser:
      url: [your site's url]
```

PhpBrowser

Click

Emulates a click on valid anchors. The page from the "href" parameter will be opened. As a parameter you can specify the link name or a valid CSS or XPath selector.

```
<?php
$I->click('Log in');
// CSS selector applied
$I->click('#login a');
// XPath
$I->click('//a[@id=login]');
// Using context as second argument
$I->click('Login', '.nav');
?>
```

Codeception tries to locate element either by its text, name, CSS or XPath. You can specify locator type manually by passing array as a parameter. We call this a **strict locator**.

Forms

Clicking the links is not what takes the most time during testing a web site. If your site consists only of links you can skip test automation. The most routine waste of time goes into the testing of forms. Codeception provides several ways of doing that.

Let's submit this sample form inside the Codeception test.

```
<form method="post" action="/update" id="update_form">
  <label for="user_name">Name</label>
  <input type="text" name="user[name]" id="user_name" />
  <label for="user_email">Email</label>
  <input type="text" name="user[email]" id="user_email" />
  <label for="user_gender">Gender</label>
  <select id="user_gender" name="user[gender]">
    <option value="m">Male</option>
    <option value="f">Female</option>
  </select>
  <input type="submit" value="Update" />
</form>
```

From a user's perspective, a form consists of fields which should be filled, and then an Update button clicked.

```
<?php
// we are using label to match user_name field
$I->fillField('Name', 'Miles');
// we can use input name or id
$I->fillField('user[email]', 'miles@davis.com');
$I->selectOption('Gender', 'Male');
$I->click('Update');
?>
```

Assertions

In the PHP browser you can test the page contents. In most cases you just need to check that the required text or element is on the page.

The most useful command for this is `see`.

```
<?php
// We check that 'Thank you, Miles' is on page.
$I->see('Thank you, Miles');
// We check that 'Thank you Miles' is inside
// the element with 'notice' class.
$I->see('Thank you, Miles', '.notice');
// Or using XPath locators
$I->see('Thank you, Miles', "descendant-or-self::*[contains(concat(' ', normalize
// We check this message is not on page.
$I->dontSee('Form is filled incorrectly');
?>
```

You can check that specific element exists (or not) on a page

```
<?php
$I->seeElement('.notice');
$I->dontSeeElement('.error');
?>
```

We also have other useful commands to perform checks. Please note that they all start with the `see` prefix.

```
<?php
$I->seeInCurrentUrl('/user/miles');
$I->seeCheckboxIsChecked('#agree');
$I->seeInField('user[name]', 'Miles');
$I->seeLink('Login');
?>
```

Conditional Assertions

Sometimes you don't want the test to be stopped when an assertion fails. Maybe you have a long-running test and you want it to run to the end. In this case you can use conditional assertions. Each `see` method has a corresponding `canSee` method, and `dontSee` has a `cantSee` method.

```
<?php
$I->canSeeInCurrentUrl('/user/miles');
$I->canSeeCheckboxIsChecked('#agree');
$I->cantSeeInField('user[name]', 'Miles');
?>
```

Each failed assertion will be shown in test results. Still, a failed assertion won't stop the test.

Grabbers

These commands retrieve data that can be used in test. Imagine, your site generates a password for every user and you want to check the user can log into the site using this password.

```
<?php
$I->fillField('email', 'miles@davis.com');
$I->click('Generate Password');
$password = $I->grabTextFrom('#password');
$I->click('Login');
$I->fillField('email', 'miles@davis.com');
$I->fillField('password', $password);
$I->click('Log in!');
?>
```

Selenium WebDriver

A nice feature of Codeception is that most scenarios can be easily ported between the testing backends. Your PhpBrowser tests we wrote previously can be executed inside a real browser (or even PhantomJS) with Selenium WebDriver.

The only thing we need to change is to reconfigure and rebuild the AcceptanceTester class, to use **WebDriver** instead of PhpBrowser.

Modify your `acceptance.suite.yml` file:

```
class_name: AcceptanceTester
modules:
  enabled:
    - WebDriver
    - AcceptanceHelper
  config:
    WebDriver:
      url: 'http://localhost/myapp/'
      browser: firefox
```

In order to run Selenium tests you need to [download Selenium Server](#) and get it running (Alternatively you may use [PhantomJS](#) headless browser in `ghostdriver` mode).

If you run acceptance tests with Selenium, Firefox will be started and all actions will be performed step by step using browser engine.

In this case `seeElement` won't just check that the element exists on a page, but it will also check that element is actually visible to user.

```
<?php
$I->seeElement('#modal');
?>
```


Multi Session Testing

Codeception allows you to execute actions in concurrent sessions. The most obvious case for it is testing realtime messaging between users on site. In order to do it you will need to launch two browser windows at the same time for the same test. Codeception has very smart concept for doing this. It is called **Friends**.

```
<?php
$I = new AcceptanceTester($scenario);
$I->wantTo('try multi session');
$I->amOnPage('/messages');
$nick = $I->haveFriend('nick');
$nick->does(function(AcceptanceTester $I) {
    $I->amOnPage('/messages/new');
    $I->fillFiled('body', 'Hello all!');
    $I->click('Send');
    $I->see('Hello all!', '.message');
});
$I->wait(3);
$I->see('Hello all!', '.message');
?>
```

In this case we did some actions in second window with `does` command on a friend object.

Configurations

- Yaml
- Extensible

Let's look at
tests\acceptance.suite.yml

Modules

⊗ Added modules appear in our ide :)

```
$I->click('Add new blog');  
$I->see('Add new blog');  
$I->
```

(m) amOnPage (\$page)	TestGuy
(m) attachFile (\$field, \$filename)	TestGuy
(m) checkOption (\$option)	TestGuy
(m) click (\$link)	TestGuy
(m) dontSee (\$text)	TestGuy
(m) dontSeeCheckboxIsChecked (\$checkbox...	TestGuy
(m) dontSeeInField (\$field, \$value)	TestGuy
(m) dontSeeLink (\$text)	TestGuy
(m) fillField (\$field, \$value)	TestGuy
(m) see (\$text)	TestGuy
(m) seeCheckboxIsChecked (\$checkbox)	TestGuy

Press again to see more variants. 22

Modules

➤ Added modules appear in our ide :)

```
$I->click('Add new blog');  
$I->see('Add new blog');  
$I->
```

m amOnPage (\$page)	TestGuy
m attachFile (\$field, \$filename)	TestGuy
m checkOption (\$option)	TestGuy
m click (\$link)	TestGuy
m dontSee (\$text)	TestGuy
m dontSeeCheckboxIsChecked (\$check...	TestGuy
m dontSeeInField (\$field, \$value)	TestGuy
m dontSeeLink (\$text)	TestGuy
m fillField (\$field, \$value)	TestGuy
m see (\$text)	TestGuy
m seeCheckboxIsChecked (\$checkbox)	TestGuy

Press again to see more variants. [»»](#)



Demo

➤ It's demonstration time !

Questions ?

➤ Please ask



Pentalog
HIGH TECH

MERCI !

p e n t a l o g . f r