

DryRun
GenerateSuite
GherkinSnippets
Init
Console
ConfigValidate
GenerateGroup
GenerateCept
Run
SelfUpdate
GenerateTest
Build
GenerateHelper
Bootstrap
GenerateEnvironment
GenerateFeature
GenerateScenarios
GenerateStepObject
Clean
GherkinSteps
Completion
GenerateCest
GeneratePageObject

Console Commands

DryRun

Shows step by step execution process for scenario driven tests without actually running them.

- `codecept dry-run acceptance`
- `codecept dry-run acceptance MyCest`
- `codecept dry-run acceptance checkout.feature`
- `codecept dry-run tests/acceptance/MyCest.php`

GenerateSuite

Create new test suite. Requires suite name and actor name

- ``
- `codecept g:suite api -> api + ApiTester`
- `codecept g:suite integration Code -> integration + CodeTester`
- `codecept g:suite frontend Front -> frontend + FrontTester`

GherkinSnippets

Generates code snippets for matched feature files in a suite. Code snippets are expected to be implemented in Actor or PageObjects

Usage:

- `codecept gherkin:snippets acceptance` - snippets from all feature of acceptance tests
- `codecept gherkin:snippets acceptance/feature/users` - snippets from feature/users dir of acceptance tests
- `codecept gherkin:snippets acceptance user_account.feature` - snippets from a single feature file
- `codecept gherkin:snippets acceptance/feature/users/user_accout.feature` - snippets from feature file in a dir

Init

Follow @codeception

Console

Try to execute test commands in run-time. You may try commands before writing the test.

- `codecept console acceptance` - starts acceptance suite environment. If you use WebDriver you can manipulate browser with Codeception commands.

ConfigValidate

Validates and prints Codeception config. Use it to debug Yaml configs

Check config:

- `codecept config:check` global config
- `codecept config:unit` check suite config

Load config:

- `codecept config:validate -c path/to/another/config` : from another dir
- `codecept config:validate -c another_config.yml` : from another config file

Check overriding config values (like in `run` command)

- `codecept config:validate -o "settings: shuffle: true"` : enable shuffle
- `codecept config:validate -o "settings: lint: false"` : disable linting
- `codecept config:validate -o "reporters: report: \Custom\Reporter"` --report : use custom reporter

GenerateGroup

Creates empty GroupObject - extension which handles all group events.

- `codecept g:group Admin`

GenerateCept

Generates Cept (scenario-driven test) file:

- `codecept generate:cept suite Login`
- `codecept g:cept suite subdir/subdir/testnameCept.php`
- `codecept g:cept suite LoginCept -c path/to/project`

Run

Executes tests.

Usage:

- `codecept run acceptance` : run all acceptance tests
- `codecept run tests/acceptance/MyCept.php` : run only MyCept
- `codecept run acceptance MyCept` : same as above
- `codecept run acceptance MyCest:myTestInIt` : run one test from a Cest
- `codecept run acceptance checkout.feature` : run feature-file
- `codecept run acceptance -g slow` : run tests from *slow* group
- `codecept run unit,functional` : run only unit and functional suites

Verbosity modes:

- `codecept run -v` :
- `codecept run --steps` : print step-by-step execution
- `codecept run -vv` :
- `codecept run --debug` : print steps and debug information
- `codecept run -vvv` : print internal debug information

Load config:

- `codecept run -c path/to/another/config` : from another dir

Follow @codeception

- `codecept run -c another_config.yml` : from another config file

Override config values:

- `codecept run -o "settings: shuffle: true"` : enable shuffle
- `codecept run -o "settings: lint: false"` : disable linting
- `codecept run -o "reporters: report: \Custom\Reporter"` --report : use custom reporter

Run with specific extension

- `codecept run --ext Recorder` run with Recorder extension enabled
- `codecept run --ext DotReporter` run with DotReporter printer
- `codecept run --ext "My\Custom\Extension"` run with an extension loaded by class name

Full reference:

Arguments:

suite	suite to be tested
test	test to be run

Options:

-o, --override=OVERRIDE	Override config values (multiple values allowed)
--config (-c)	Use custom path for config
--report	Show output in compact style
--html	Generate html with results (default: "report.html")
--xml	Generate JUnit XML Log (default: "report.xml")
--tap	Generate Tap Log (default: "report.tap.log")
--json	Generate Json Log (default: "report.json")
--colors	Use colors in output
--no-colors	Force no colors in output (useful to override config file)
--silent	Only outputs suite names and final results
--steps	Show steps in output
--debug (-d)	Show debug and scenario output
--coverage	Run with code coverage (default: "coverage.serialized")
--coverage-html	Generate CodeCoverage HTML report in path (default: "coverage")
--coverage-xml	Generate CodeCoverage XML report in file (default: "coverage.xml")
--coverage-text	Generate CodeCoverage text report in file (default: "coverage.txt")
--no-exit	Don't finish with exit code
--group (-g)	Groups of tests to be executed (multiple values allowed)
--skip (-s)	Skip selected suites (multiple values allowed)
--skip-group (-x)	Skip selected groups (multiple values allowed)
--env	Run tests in selected environments. (multiple values allowed, environments can be merged with ',')
--fail-fast (-f)	Stop after first failure
--help (-h)	Display this help message.
--quiet (-q)	Do not output any message.
--verbose (-v vv vvv)	Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug
--version (-V)	Display this application version.
--ansi	Force ANSI output.
--no-ansi	Disable ANSI output.
--no-interaction (-n)	Do not ask any interactive question.

SelfUpdate

Auto-updates phar archive from official site: '<http://codeception.com/codecept.phar>' .

- `php codecept.phar self-update`

@author Franck Cassedanne franck@cassedanne.com (<mailto:franck@cassedanne.com>)

GenerateTest

Generates skeleton for Unit Test that extends `Codeception\TestCase\Test` .

- `codecept g:test unit User`
- `codecept g:test unit "App\User"`

Build

Follow @codeception

Generates Actor classes (initially Guy classes) from suite configs. Starting from Codeception 2.0 actor classes are auto-generated. Use this command to generate them manually.

- `codecept build`
- `codecept build path/to/project`

GenerateHelper

Creates empty Helper class.

- `codecept g:helper MyHelper`
- `codecept g:helper "My\Helper"`

Bootstrap

Creates default config, tests directory and sample suites for current project. Use this command to start building a test suite.

By default it will create 3 suites **acceptance**, **functional**, and **unit**.

- `codecept bootstrap` - creates `tests` dir and `codeception.yml` in current dir.
- `codecept bootstrap --empty` - creates `tests` dir without suites
- `codecept bootstrap --namespace Frontend` - creates tests, and use `Frontend` namespace for actor classes and helpers.
- `codecept bootstrap --actor Wizard` - sets actor as `Wizard`, to have `TestWizard` actor in tests.
- `codecept bootstrap path/to/the/project` - provide different path to a project, where tests should be placed

GenerateEnvironment

Generates empty environment configuration file into `envs` dir:

- `codecept g:env firefox`

Required to have `envs` path to be specified in `codeception.yml`

GenerateFeature

Generates Feature file (in Gherkin):

- `codecept generate:feature suite Login`
- `codecept g:feature suite subdir/subdir/login.feature`
- `codecept g:feature suite login.feature -c path/to/project`

GenerateScenarios

Generates user-friendly text scenarios from scenario-driven tests (Cest, Cept).

- `codecept g:scenarios acceptance` - for all acceptance tests
- `codecept g:scenarios acceptance --format html` - in html format
- `codecept g:scenarios acceptance --path doc` - generate scenarios to `doc` dir

GenerateStepObject

Generates StepObject class. You will be asked for steps you want to implement.

- `codecept g:step acceptance AdminSteps`
- `codecept g:step acceptance UserSteps --silent` - skip action questions

Clean

Cleans output directory

- `codecept clean`
- `codecept clean -c path/to/project`

GherkinSteps

Prints all steps from all Gherkin contexts for a specific suite

```
codecept gherkin:steps acceptance
```

Completion

GenerateCest

Generates Cest (scenario-driven object-oriented test) file:

- `codecept generate:cest suite Login`
- `codecept g:cest suite subdir/subdir/testnameCest.php`
- `codecept g:cest suite LoginCest -c path/to/project`
- `codecept g:cest "App\Login"`

GeneratePageObject

Generates PageObject. Can be generated either globally, or just for one suite. If PageObject is generated globally it will act as UIMap, without any logic in it.

- `codecept g:page Login`
- `codecept g:page Registration`
- `codecept g:page acceptance Login`

Looking for commercial support? Request official consulting service (http://sdclabs.com/codeception?utm_source=codeception.com&utm_medium=docs_bottom&utm_term=link&utm_campaign=reference)

Codeception is a BDD-styled PHP testing framework, brought to you by Codeception Team (<http://codeception.com/credits>). Logo by Mr. Adnan (<https://twitter.com/adnanblog>). OpenSource **MIT Licensed**.

Thanks to



(<https://www.jetbrains.com/phpstorm/>)



(<https://www.rebilly.com/>)



(<https://github.com/codeception/codeception>)



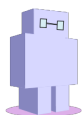
(<https://twitter.com/codeception>)



(<http://www.facebook.com/pages/Codeception/288959711204412>)

- Installation (/install)
- Credits (/credits)
- Releases (/changelog)
- Commercial Services (http://sdclabs.com/codeception?utm_source=codeception.com&utm_medium=bottom_menu&utm_term=link&utm_campaign=reference)
- License (<https://github.com/Codeception/Codeception/blob/master/LICENSE>)

Codeception Family



Robo

(<http://robo.li>)(<http://robo.li>)

Modern PHP **Task Runner**. Allows to declare tasks with zero configuration in pure PHP.



CodeceptJS

(<http://codecept.io>)(<http://codecept.io>)

Codeception for **NodeJS**. Write acceptance tests in ES6 and execute in webdriverio, Selenium WebDriver, and Protractor.