[Установка Codeception 1](#_Toc503255561)

[Первый вариант командами composer 1](#_Toc503255562)

[Глобальная установка. 1](#_Toc503255563)

[Локально: 2](#_Toc503255564)

[Второй вариант composer.json 3](#_Toc503255565)

[Третий вариант Phar 3](#_Toc503255566)

[Установка Selenium 3](#_Toc503255567)

[Первый вариант программой установки 4](#_Toc503255568)

[Второй вариант руками 4](#_Toc503255569)

[Третий вариант запускающий скрипт через Composer 4](#_Toc503255570)

[Проблемы при установке Selenium 5](#_Toc503255571)

[Инициализация окружения для тестирования Codecept 8](#_Toc503255572)

[Файловая структура 8](#_Toc503255573)

[Один загрузчик для нескольких приложений 9](#_Toc503255574)

[Пространства имен 10](#_Toc503255575)

[Bootstrap 10](#_Toc503255576)

[Настройки Codeception: 11](#_Toc503255577)

[codeception.yml 12](#_Toc503255578)

[Сообщения об ошибках 13](#_Toc503255579)

[Фреймворки 13](#_Toc503255580)

[functional.suite.yml 15](#_Toc503255581)

[acceptance.suite.yml 15](#_Toc503255582)

[Модули и помощники 16](#_Toc503255583)

[Стандартные модули 17](#_Toc503255584)

[https://github.com/Codeception/Codeception/tree/master/docs/modules 17](#_Toc503255585)

[Помощники 17](#_Toc503255586)

[МОДУЛЬ DB 21](#_Toc503255587)

# Установка Codeception

## Первый вариант командами composer

### Глобальная установка.

Если вы планируете использовать фреймворк Codeception для нескольких проектов, то лучше установить его глобально (не в каталог vendor текущего проекта, а в каталог vendor пользователя, откуда Composer так же подключает пакеты):

composer global require "codeception/codeception=2.0.\*"

composer global require "codeception/specify=\*"

composer global require "codeception/verify=\*"

Исполняемый файл **codecept.bat** при этом автоматически запишется в папку vendor\bin пользователя (для Win), которая должна быть указана в системной переменной **PATH** (как и папка с интерпретатором php) и таким образом вы сможете обращаться к фреймворку только по названию исполняемого файла:

codecept

Если в PATH не указан путь к **bin**, обязательно укажите. Чтобы проверить наберите в консоли:

echo %PATH%

Если планируете использовать кириллические символы (русский) в тестах, то чтобы они выводились корректно в консоли, можно добавить строку

@chcp 65001

в файл codecept.bat находящийся в папке пользователя, например: C:\Users\имя\_вашего\_пользователя\AppData\Roaming\Composer\vendor\bin\codecept.bat

При запуске команды

codecept

Мы должны увидеть приветствие:

Codeception version 2.0.16

### Локально:

Установка в текущий проект осуществляется командой:

composer require "codeception/codeception=2.0.\*"

composer require "codeception/specify=\*"

composer require "codeception/verify=\*"

Запускающий файл будет находиться в папке vendor\bin проекта. Чтобы не набирать для тестирования «vendor\bin\codecept»:  
**1) для Linux** создаем алиас:

alias codecept="./vendor/bin/codecept"

**2) для Win**  
создать в корне проекта файл codecept.bat с содержимым:

@echo off

@setlocal

@chcp 65001

set CODECEPT\_PATH=vendor/bin/

"%CODECEPT\_PATH%codecept.bat" %\*

@endlocal

строкой @chcp 65001 я указал кодировку чтобы кириллические символы отображались корректно.

При запуске команды

codecept

Мы должны увидеть приветствие:

Codeception version 2.0.16

Далее я буду описывать процесс исходя из того, что установлен Windows.  
Теперь для запуска из корня проекта достаточно писать просто

codecept

Если вы не планируете проводить приемочное тестирование, то можете установить базовую версию **Codeception** без загрузки библиотек Webdriver и Guzzle:

composer require codeception/base --dev

Для ознакомления с консольными командами **Codeception** можно воспользоваться справкой:

вывести описание доступных консольных команд

codecept

вывести справку по нужной команде

codecept название команды –h

## Второй вариант composer.json

Установка приложения локально для проекта

В файле composer.json добавляем

"require": {

"codeception/codeception": "2.0.\*",

}

И запускаем команду

composer update

## Третий вариант Phar

Скачайте phar-архив Codeception с http://codeception.com/thanks

wget http://codeception.com/codecept.phar .

# Установка Selenium

**PhpBrowser** не позволяет полноценно протестировать страницу на которой используется **JavaScript**, т.к. данный модуль использует библиотеку Curl. Для таких целей подключают модуль WebDriver, который использует **Selenium**.

Selenium — это проект, предоставляющий драйвера для разных браузеров, которые встраиваются в них и позволяют управлять ими. Также, Selenium содержит сервер, который позволяет управлять большим числом разных браузеров и распределять задания между ними. Selenium сервер написан на Яве, потому она понадобится чтобы его запустить.  
  
При использовании Selenium для тестов будет запускаться браузер и вы увидите, как происходит заполнение полей форм, их отправка на сервер, клики по элементам и прочие действия, которые обычно совершаются пользователем приложения.

## Первый вариант

Переходим на http://www.seleniumhq.org/download/ и нажимаем по ссылке после слов «Download version».  
Для того чтобы скачать предыдущие версии Selenium переходим по ссылке http://selenium-release.storage.googleapis.com/index.html и там скачиваем файл такого типа: selenium-server-standalone-X.X.X.jar

<http://chromedriver.storage.googleapis.com/index.html>

Создать start-server.bat

java -jar c:\selenimu-server\selenium-server-standalone{version}.jar -Dwebdriver.chrome.driver=c:\selenimu-server\chromedriver.exe

Драйвера можно посмотреть во здесь <http://docs.seleniumhq.org/download/>

Далее, для использования **Firefox** скачиваем драйвер <https://github.com/mozilla/geckodriver/releases> для своей операционной системы и сохраняем его в каталог указанный в системной переменной **PATH**. Я сохранил в D:\OpenServer 5.2.7\modules, эта папка указана у меня в PATH. Вы можете указать нужный путь к каталогу с драйвером в переменной PATH. В эту же папку я переместил и скачанный модуль Selenium.  
  
Установка описана в комментариях файла модуля - vendor\codeception\codeception\src\Codeception\Module\WebDriver.php  
  
Перед выполнением тестирования должен быть запущен сервер. Запускаем командой:

java -jar -Dwebdriver.gecko.driver="D:\OpenServer 5.2.7\modules\geckodriver.exe" "D:\OpenServer 5.2.7\modules\selenium-server-standalone-3.4.0.jar"

или

java -jar selenium-server-standalone-3.8.1.jar -Dwebdriver.chrome.driver=chromedriver.exe

где нужно указать свои пути к скачанным файлам и в конце изменить версию **Selenium**.  
Тут я указал путь к **geckodriver.exe**, но если он у вас лежит в папке которая указана в системной переменной PATH, то его можно не указывать. Получится:

java -jar "D:\OpenServer 5.2.7\modules\selenium-server-standalone-3.4.0.jar"

Так же вы можете создать файл selenium-server-standalone.bat в папке vendor\bin с указание своих путей к исполняемым файлам для удобства запуска модуля одной короткой строкой по примеру установки с помощью Composer.

## Второй вариант запускающий скрипт через Composer

Но чтобы не возиться со скачиванием файла вручную и созданием ярлыка для его запуска лучше просто установим сам сервер и запускающий скрипт через Composer:

composer global require se/selenium-server-standalone

Или с указанием конкретной версии:

composer global require --dev se/selenium-server-standalone 2.53.1

вы установите сервер **Selenium** глобально в папку текущего пользователя, у меня это C:\Users\User\AppData\Roaming\Composer\vendor

если у вас путь к bin уже прописан в PATH, будем запускать в консоли просто по этому же имени:

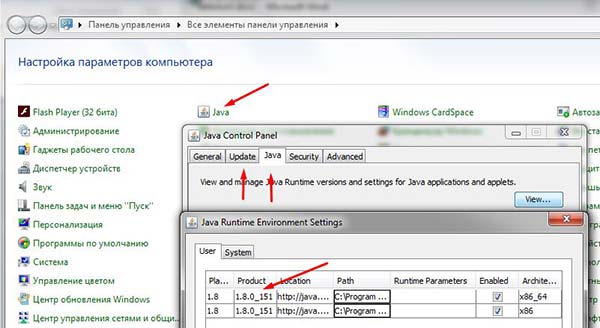
selenium-server-standalone

# Проблемы при установке Selenium

Если вы запускаете из Windows и у вас возникает такая ошибка:  
/usr/bin/bash: "C:\Users\...\AppData\Roaming\Composer\vendor\bin\/../se/selenium-server-standalone/bin/selenium-server-standalone": No such file or directory  
то нужно подкорректировать файл C:\Users\{USERNAME}\AppData\Roaming\Composer\vendor\bin\selenium-server-standalone.bat (тут показан путь при глобальной установке Selenium), заменить 2 последние строки на

SET BIN\_TARGET=%~dp0/../se/selenium-server-standalone/bin/selenium-server-standalone.jar

java -jar "%BIN\_TARGET%" %\*

т.к. прописанная там по-умолчанию команда **sh** предназначена для выполнения на **Unix** подобных системах.  
  
А если при выполнении команды возникнет ошибка такого плана:  
Exception in thread "main" java.lang.UnsupportedClassVersionError: org/openqa/grid/selenium/GridLauncherV3 : Unsupported major.minor version 52.0…  
значит **Selenium** не может работать с установленной версией **Java** и нужно ее обновить или установить другую версию Selenium. В **Win-7** для просмотра версии или обновления Java можно открыть в «Панели управления»:   
  
  
Если все равно будут ошибки при запуске Selenium установите версию пониже.  
  
Так же, при экспериментировании с версиями, уже при запуске тестов возможна ошибка такого плана:   
[Facebook\WebDriver\Exception\UnknownServerException] Unable to connect to host 127.0.0.1 on port 7055 after 45000 ms.…  
Для устранения данной ошибки нужно обновить (или подобрать) версию Firefox с которой работает скаченный модуль Selenium. Например с selenium-server-standalone-2.53.1.jar будет работать с Firefox 47.0.1

Если возникнет ошибка примерно с таким содержанием:  
[Facebook\WebDriver\Exception\SessionNotCreatedException] Unable to create new remote session. desired capabilities = Capabilities [{firefoxOptions=org.openqa.selenium.firefox.FirefoxOptions@7a99c504, browserName=firefox, moz:firefoxOptions=org.openqa.selenium.firefox.FirefoxOptions@7a99c504}], required capabilities = Capabilities [{}]Build info: version: '3.0.0', revision: '350cf60', time: '2016-10-13 10:48:16 -0700'System info: host: 'MICROSOFT-PC', ip: '192.168.56.1', os.name: 'Windows 7', os.arch: 'amd64', os.version: '6.1', java.version: '1.8.0\_40'Driver info: driver.version: FirefoxDriver  
значит запускаемый файл Selenium не соответствует установленной версии Java и скорее всего придется попробовать Selenium более старой версии.  
  
См. так же ошибки описанные в пункте установки с помощью Composer.  
  
У меня работала такая связка:  
Java 1.8.0\_40 (1.8.0\_51)selenium-server-standalone-3.4.0.jarи Firefox 56.0.2  
хотя текущая на момент написания статьи была версия selenium-server-standalone-3.6.0.jar

При использовании Selenium, метод **see()** позволяет не просто убедится, что текст внутри элемента существует, но также проверит, что элемент действительно виден пользователю.

<?php

$I->see('Confirm','#modal');

?>

Например на странице используется «Bootstrap Accordion» т.е. выводятся в ряд заголовки, кликнув по которым открывается блок соответствующего им текста. До клика по нужному заголовку, блок принадлежащего ему текста скрыт. При использовании **PhpBrowser** проверка текста из скрытого блока на то, что он в данное время не видим пользователю:

$I->dontSee('скрытый текст');

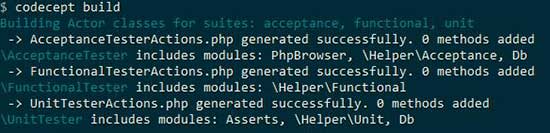
завершится с ошибкой, т.к. PhpBrowser не умеет работать с скрытым текстом и он будет считаться видимым. А вот **Selenium** с этим справится.

Основной концепцией Codeception является представление тестов как действий человека. Мы будем называть этого человека "**парнем**" (Guy). У нас есть **CodeGuy**, который выполняет функции/методы и тестирует код. Также у нас есть **TestGuy** - опытный тестировщик, который тестирует приложение целиком и знает кое-что о его внутреннем устройстве. Еще у нас есть **WebGuy**, пользователь, который работает с нашим приложением через интерфейс, который мы предоставляем.

Каждый из этих "парней" представляет из себя PHP класс с набором действий, который он может совершать. Как Каждый из этих "парней" имеет различные способности. Эти способности не постоянны, Вы можете расширять их. Вы даже можете создавать новых "парней", но запомните: один "парень" на один набор тестов.

Классы "парней" не пишутся, они генерируются следующей командой после изменения конфигурационных файлов:

codecept build

это сгенерирует трейты с методами для тестирования из модулей Codeception которые появится в папке tests\\_support\\_generated.   
При этом в информации, которая будет выведена в консоль после выполнения команды, можно увидеть кол-во добавленных методов и подключенных модулей для каждого типа теста:  
  
  
Сам класс конфигурации находится в файле vendor\codeception\codeception\src\Codeception\Configuration.php

# Инициализация окружения для тестирования Codecept

Сodecept bootstrap

эта команда сгенерирует папку с тестами и установит дерево нужных каталогов с файлами.

Команда инициализации создаст следующую структуру файлов и каталогов в указанной вами директории.

### Файловая структура

app/tests/codeception Корневой каталог фреймворка

├── codeception.yml Основной файл конфигурации

└── tests Каталог с тестами

├── \_bootstrap.php Общий файл инициализации

├── unit.suite.yml Конфигурация Unit-тестов

├── unit Каталог с Unit-тестами

│ ├── \_bootstrap.php Файл инициализации при запуске Unit-тестов

│ └── CodeEngineer.php Базовый класс с общими методами необходимыми в Unit-тестах

├── functional.suite.yml Конфигурация модулей функциональных тестов

├── functional Каталог с функциональными тестами

│ ├── \_bootstrap.php

│ └── TestEngineer.php

├── acceptance.suite.yml Конфигурация модулей приемочных тестов

├── acceptance Каталог с приемочными тестами

│ ├── \_bootstrap.php Файл инициализации при запуске приемочных тестов

│ ├── WebEngineer.php Базовый класс с кастомными тестовыми методами

│ └── ManageUserCest.php Класс тестирует интерфейс управления пользователями

├── \_data Каталог для файлов, необходимых при тестировании

│ ├── database.dump.sql Дамп БД для восстановления тестового окружения приложения

│ ├── books.csv Вы можете держать здесь любые необходимые длятестов файлы..

│ └── testing.png Например, изображения...

├── \_pages Каталог для файлов описанных данных для заполнения страниц

│ └── AboutPage.php Данные для заполнения на странице About

│ └── contactPage.php Данные для заполнения на странице Contact

├── \_helpers

│ ├── CodeHelper.php

│ ├── TestHelper.php

│ └── WebHelper.php

└── \_log Логи, отчеты, скриншоты о прохождении тестов

├── report.json

├── report.tap.log

└── ManageUserCest.create.fail.png

## \_Bootstrap файлы

Каждый набор тестов имеет свой собственный стартовый (bootstrap) файл. Он расположен в директории, содержащей набор тестов и называется \_bootstrap.php. Этот файл исполняется перед каждым тестом из соответствующего набора. Любые подготовительные операции для набора тестов следует писать именно в этом файле

В папке tests в файле \_bootstrap.php необходимо написать конфигурацию общую для 3 типов тестов.

В файле codeception/\_bootstrap.php переопределяем путь к базе данных. Для тестов мы можем использовать SQL

К примеру

<?php  
ini\_set('xdebug.max\_nesting\_level', 1000);

defined('YII\_DEBUG') **or** define('YII\_DEBUG', **true**);  
defined('YII\_ENV') **or** define('YII\_ENV', 'test');  
  
**require**(*\_\_DIR\_\_* . '/../vendor/autoload.php');  
**require**(*\_\_DIR\_\_* . '/../vendor/yiisoft/yii2/Yii.php');  
  
Yii::*setAlias*('@tests', dirname(*\_\_DIR\_\_*) . '/tests');

Изначально нас интересуют эти файлы:

tests

└── codeception

├── acceptance

│ └── \_bootstrap.php

├── functional

│ └── \_bootstrap.php

└── unit

└── \_bootstrap.php

Файлы acceptance/\_bootstrap.php

**new** yii\web\Application(**require**(dirname(\_\_DIR\_\_) . '/config/acceptance.php'));

и:

functional/\_bootstrap.php

**new** yii\web\Application(**require**(dirname(\_\_DIR\_\_) . '/config/functional.php'));

Внутри же третьего unit/\_bootstrap.php такого кода нет, так как создание и завершение приложения находятся в методах setUp() и tearDown() самого класса yii\codeception\TestCase, от которого будут наследоваться классы наших тестов из папки unit.

# Пространства имен

Для того, чтобы избежать конфликтов имен между Guy и Helper классами, они должны быть добавлены в пространства имен.  
Для создания набора тестов с использованием пространств имен, добавьте опцию --namespace к команде bootstrap.

codecept bootstrap --namespace frontend

Это создаст новый проект с параметром namespace: frontend в файле конфигурации codeception.yml.  
Помощники будут использовать пространство имен frontend\Codeception\Module а классы Guy пространство имен frontend.  
Сгенерированные классы будут выглядеть примерно так:

<?php use frontend\WebGuy;

$I = new WebGuy($scenario);

//...

?>

Codeception имеет утилиты для того, чтобы обновить тесты существующего проекта, добавив в них использование пространств имен. Этого можно добиться выполнив следующую команду

codecept refactor:add-namespace frontend

Вы получите guy классы и помощники, а так же cept тесы обновленные для использования с пространствами имен. Запомните, что Cest файлы должны быть обновлены вручную. Коме того опция namespace не изменит пространство имен тестов или Cest классов. Она используется только для Guys классов или классов Помощников.

В то время, когда каждое приложение (бандл) имеет собственное пространство имен и разные классы помощники и guy классы, вы можете выполнить их с помощью одной команды.  
Используйте meta-config который мы создали выше для запуска тестов как вы делаете это обычно.

codecept run

Это выполнит тесты для всех 3 приложений и сольет отчеты в один. По существу, это может быть довольно удобно, в случае, если вы запускаете тесты на сервере непрерывной интеграции и хотите получить один отчет в формате JUnit или HTML. Отчет о покрытии кода тестами, будет так же объединен в один.

# Настройки Codeception:

Для каждого типа тестов (приемочные, функциональные, модульные) существует дополнительный, специализированный конфигурационный файл.

Codeception имеет глобальный файл конфигурации codeception.yml и отдельный конфигурационный файл для каждого набора тестов. Также поддерживаются конфигурационные файлы .dist. Если у Вас в проекте несколько разработчиков - разместите общие параметры конфигрурации в файле codeception.dist.yml, а индивидуальные настройки каждого разработчика в файле codeception.yml. Тоже самое справедливо и для конфигураций наборов тестов. Например, файл unit.suite.yml будет объединен с файлом unit.suite.dist.yml.

tests

├── codeception

│ ├── acceptance.suite.yml

│ └── acceptance.suite.dist.yml

│ ├── functional.suite.yml

│ └── functional.suite.dist.yml

│ ├── unit.suite.yml

│ └── unit.suite.dist.yml

│ ├ codeception.yml

└codeception.dist.yml

## Один загрузчик для нескольких приложений

В случае, если проект состоит из нескольких приложений (frontend, admin, api) или вы используете бандлы Symfony2,  
вы можете захотеть, чтобы тесты для всех приложений запускались одним загрузчиком.  
В таком случае вы получите один отчет для всего проекта.

Начиная с версии Codeception 1.6.3 появилась возможность создать meta-config, включающий конфигурационные файлы из разных каталогов.

Поместите codeception.yml в корень проекта, и укажите пути к других codeception.yml конфигам, которые вы хотите подключить.

include:

- frontend

- admin

- api/rest

paths:

log: log

settings:

colors: false

Кроме того, вам следует указать путь к каталогу log, в котором будут храниться отчеты и логи всего проекта.

### codeception.yml

В общем файле codeception.yml настроим анализатор покрытия (test coverage) нашего кода. Включаем только папки с кодом и исключаем всё лишнее. Другие секции нас вполне устраивают, и мы будем запускать тесты по адресу http://localhost:8080, поэтому всё остальное оставляем как есть:

actor: Tester

coverage:

enabled: true

whitelist:

include:

- ../components/\*

- ../mail/\*

- ../modules/\*

- ../views/\*

exclude:

- ../modules/admin/messages/\*

- ../modules/main/messages/\*

- ../modules/user/messages/\*

blacklist:

include:

- ../assets/\*

- ../config/\*

- ../runtime/\*

- ../environments/\*

- ../messages/\*

- ../vendor/\*

- ../web/\*

- ../tests/\*

paths:

# где хранятся модули

tests: codeception

# там будут записаны журналы и выходные данные отладки

log: codeception/\_output

# директория для данных фикстур

data: codeception/\_data

# директория для пользовательских модулей (helpers)

helpers: codeception/\_support

settings:

# название bootstrap файла который будем использовать

# Каждый файл должен быть в своем каталоге

#.

bootstrap: \_bootstrap.php

# При необходимости можно расширить класс

suite\_class: \PHPUnit\_Framework\_TestSuite

# Тесты (особенно функциональные) могут занять много памяти

# Мы установили верхний предел для них по умолчанию.

memory\_limit: 1024M

# Если журнал должен быть написан.

# Каждое действие в тесте регистрируется.

#Журналы хранятся в течение 3 дней.

log: true

# по умолчанию выключенно в Windows

# use [ANSICON](http://adoxa.110mb.com/ansicon/) to colorize output.

colors: true

config:

# the entry script URL (with host info) for functional and acceptance tests

# PLEASE ADJUST IT TO THE ACTUAL ENTRY SCRIPT URL

test\_entry\_url: <http://localhost:8080/index-test.php>

# Global modules configuration.

modules:

config:

Db:

dsn: 'pgsql:host=localhost;dbname=DBNAME'

user: 'USER'

password: 'PASS'

dump: tests/\_data/database.dump.sql

populate: true

cleanup: true

### Сообщения об ошибках

По умолчанию Codeception использует значение E\_ALL & ~E\_STRICT & ~E\_DEPRECATED.  
В функциональных тестах вы можете захотеть сменить эти значения в зависимости от используемого фреймворка.  
Сообщения об ошибках могут быть настроены в конфигурационном файле набора:

class\_name: TestGuy

modules:

enabled: [Yii1, TestHelper]

error\_level: "E\_ALL & ~E\_STRICT & ~E\_DEPRECATED"

error\_level может быть установлен глобально в файле codeception.yml.

## functional.suite.yml

Теперь подключим модуль фикстур для наших функциональных и приёмочных тестов. Откроем functional.suite.yml и добавим его в массив активных модулей:

class\_name: FunctionalTester

modules:

enabled:

- Filesystem

- Yii2

- tests\codeception\\_support\FixtureHelper

config:

Yii2:

configFile: 'codeception/config/functional.php'

## acceptance.suite.yml

Аналогично добавим модуль фикстур в acceptance.suite.yml. Заодно и вместо curl-эмулятора браузера PhpBrowser подключим Facebook WebDriver, предназначенный для работы с Selenium Server, и настроим его на запуск браузера Firefox:

class\_name: AcceptanceTester

modules:

enabled:

- WebDriver

- tests\codeception\\_support\FixtureHelper

config:

WebDriver:

url: http://localhost:8080

browser: firefox

restart: true

window\_size: 1024x768

**Возможные настройки модуля WebDriver**:

* **url** - Стартовый URL для вашего приложения.
* **browser** - Браузер для запуска.
* **host** - хост сервера Selenium (по умолчанию 127.0.0.1).
* **port** - порт сервера Selenium (по умолчанию 4444).
* **restart** - установите в «false» (по умолчанию) для использования одного и того же окна браузера для всех тестов или установите «true» для создания нового окна для каждого теста. В любом случае, когда все тесты завершены, окно браузера закрывается.
* **start** - автозапуск браузера для тестирования. Может быть отключен, если сеанс браузера начинается с «\_initializeSession» внутри помощника.
* **window\_size** - Начальный размер окна. Установите значение «maximize» или пропишите в формате «640x480».
* **clear\_cookies** - Установите значение «false», чтобы сохранить файлы cookie, или установите значение «true» (по умолчанию), чтобы удалить все куки между тестами.
* **wait** (по умолчанию: 0 секунд) - всякий раз, когда элемент требуется и не находится на странице, подождите несколько секунд, чтобы найти его перед сбоем.
* **capability** - Настраивает Selenium [желаемые возможности] (https://github.com/SeleniumHQ/selenium/wiki/DesiredCapabilities). Должен быть массив значений ключа.
* **connection\_timeout** - тайм-аут для открытия соединения с удаленным сервером (по умолчанию 30 секунд).
* **request\_timeout** - тайм-аут для запроса вернуть что-то из удаленного сервера selenium (по умолчанию 30 секунд).
* **pageload\_timeout** - время ожидания загрузки страницы до того, как будет выдана ошибка (по умолчанию 0 секунд).
* **http\_proxy** - устанавливает URL-адрес прокси-сервера HTTP для тестирования удаленного сервера.
* **http\_proxy\_port** - устанавливает порт HTTP-прокси-сервера
* **debug\_log\_entries** - сколько записей Selenium для печати с «debugWebDriverLogs» или при сбое (по умолчанию 15).
* **log\_js\_errors** - установите значение «true», чтобы включить возможный отчет JavaScript в HTML, или установите значение «false» (по умолчанию) для деактивации.

Подробнее можно посмотреть в блоке ## Configuration файла vendor\codeception\codeception\src\Codeception\Module\WebDriver.php

На этом конфигурирование завершено.

Если вы создадите в каталоге проекта файл **.scrutinizer.yml и поместите туда вот такое содержание**

imports:

- php

tools:

external\_code\_coverage:

timeout: 2100 # Timeout in seconds.

# disable copy paste detector and similarity analyzer as they have no real value

# and a huge bunch of false-positives

php\_sim: false

php\_cpd: false

**то ваш проект на GitHub будет тестировать сервис scrutinizer-ci.com**