## Установка PHPUnit

Phpunit 4.8 требует PHP 5.4 Остальные версии требуют более поздних версии. Поэтому для Yii2 ставим версию 4.8.

Страница официальной документации по установке PHPUnit на английском языке: <https://phpunit.de/manual/current/en/installation.html>   
Рассмотрим способы установки подробно.  
  
1. Composer

Узнать текущую версию PHPUnit можно [тут](https://phpunit.ru/). В примерах установки мы будем устанавливать версию 5.7. Более поздние версии требуют наличия PHP 7+ А PHPUnit 5.7 поддерживается на PHP 5.6, PHP 7.0 и PHP 7.1

### 1.1 Установка локально

Итак, команда для Composer устанавливающая PHPUnit версии 5.7:

composer require --dev phpunit/phpunit ^5.7

Установить дополнительные компоненты

composer require --dev phpunit/dbunit

composer require "phpunit/phpunit-selenium=\*"  
composer require "phpunit/phpunit-skeleton-generator=\*"  
composer require "phpunit/phpunit-story=\*"

Установив можно запустить тестирование командой из корня проекта:  
vendor\bin\phpunit tests

тут имеется ввиду, что тесты будут храниться в папке tests, расположенной в корне приложения на уровне файла composer.json в противном случае нужно указать другой путь к папке с тестами.  
  
Чтобы не набирать каждый раз строку vendor\bin\phpunit, можно:  
  
 Если нужно только для текущего проекта - создать файл **phpunit.bat** в корневой папке приложения с содержимым:

@ECHO OFF

setlocal DISABLEDELAYEDEXPANSION

SET BIN\_TARGET=%~dp0vendor\phpunit\phpunit\phpunit

php "%BIN\_TARGET%" %\*

После этого можно будет запускать тестирование так:  
phpunit unit

Если файл, методы которого хотим протестировать, не подключаются автоматически в тестирующем его классе, то нужно сделать это самостоятельно. Т.к. используется Composer, то делать это лучше всего с помощью его автозагрузчика. Для этого нужно отредактировать объект «**autoload**» в файле **composer.json**. Например, кое-где для демонстрации тестирования я классам присвоил пространство имен main, а сами файлы положил в корень проекта. В таком случае получилось:

"autoload": {

"psr-4": {

"main\\": ""

}

}

Теперь классы с пространством имен «main» автоматически будут найдены в корне проекта.   
Или, например, создал в корне каталог app с файлами классов, которые нужно тестировать. Для того, что бы эти классы автоматически подключались нужно прописать:

"autoload": {

"classmap": ["app"]

}

После внесения изменений в composer.json нужно обновить автозагрузчик:

composer dump-autoload -o

### 1.2 Установка глобально

2) Установить phpunit глобально, чтобы вызывать простой командой (phpunit) из любой директории любого проекта:

composer global require --dev phpunit/phpunit ^5.7

Это установит phpunit в директорию текущего пользователя, например:  
C:\Users\username\AppData\Roaming\Composer\vendor  
и сохранит запускающий файл в папке vendor\bin пользователя. Далее нужно прописать путь к этому файлу в системную переменную PATH (если нет еще), добавить путь к папке bin:  
C:\Users\username\AppData\Roaming\Composer\vendor\bin

где username заменить на имя пользователя.  
Установить дополнительные компоненты

**composer global require "phpunit/dbunit"  
composer global require "phpunit/phpunit-selenium=\*"  
composer global require "phpunit/phpunit-skeleton-generator=\*"  
composer global require "phpunit/phpunit-story=\*"  
composer global require 'phpunit/php-invoker=\*'**

## Если PHPUnit устанавливался ****глобально****, то предварительно нужно подключить автозагрузчик приложения (обычно это автозагрузчик Composer), чтобы файлы классов, которые нужно протестировать, автоматически подключались в классах тестов. PHPUnit позволяет выполнить определенный код, а в данном случае загрузить автозагрузчик Composer до выполнения тестирования с помощью параметра ****bootstrap****, который должен указывать на файл подключающий этот автозагрузчик и/или что-то еще.

## Bootstrap

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

Параметр bootstrap можно:  
**1)** указывать **в командной строке** при вызове phpunit:  
phpunit --bootstrap tests/bootstrap.php

или можно подключить сразу автозагрузчик Composer напрямую:  
phpunit --bootstrap vendor/autoload.php

**2)** указать в конфигурационном файле **phpunit.xml**:

<phpunit bootstrap="./tests/bootstrap.php">…

или, если нужно только лишь подключить автозагрузчик Composer:

<phpunit bootstrap="./vendor/autoload.php">…

Если указываем подключение файла bootstrap.php, то создаем его в папке tests:

<?php

require \_\_DIR\_\_ .'/../vendor/autoload.php';

теперь классы приложения будут подключены.

Если при создании приложения не используется php-фреймворк, а PHPUnit установлен не глобально (в папку vendor проекта) - не забываем сами подключить автозагрузчик Composer в "точке входа", обычно это **index.php**

require\_once './vendor/autoload.php';

тут рассчитано, что индексный файл у вас находится на уровне каталога vendor (который появляется после использования Composer), т.е., возможно, понадобится подняться на уровень-два выше к данному каталогу.  
Используя Composer вам не придется использовать аналогичную конструкцию типа require для подключения тестируемого класса, как это часто указано в примерах использования PHPUnit в интернете.  
  
Осталось добавить, что при написании тестов в PHPUnit установленного с помощью Composer, ваш тестирующий класс (содержащий тесты) должен наследовать от класса phpunit:

class ClassnameTest extends PHPUnit\Framework\TestCase {}

## 2. Ручная установка

### 2.1 Установка вручную для Windows.

Скачать файл с последней версией PHPUnit <https://phar.phpunit.de/phpunit.phar>  
  
Последний релиз требует PHP 7+, чтобы получить поддержку PHP 5.6, скачать:  
<https://phar.phpunit.de/phpunit-5.7.17.phar>   
  
Устанавливать будем в папку с php. При этом путь к интерпретатору php должен быть прописан в системную переменную **PATH**.  
Итак, поместить файл в каталог с интерпретатором php (где находится файл php.exe), например W:\modules\php\PHP-5.6.  
Перейти в каталог с **PHP.exe**, например у меня на **OpenServer**:

cd W:\modules\php\PHP-5.6

создать файл **phpunit.bat** в данном каталоге, чтобы получить доступ к phpunit из любой папки (например из корня своего приложения), а не только находясь в папке с php куда он установлен:

@php "%~dp0phpunit-5.7.17.phar" %\*

где вместо 5.7.17 указать версию скачанного файла.  
Или так:

echo @php "%~dp0phpunit.phar" %\*>phpunit.bat

Проверим появился ли доступ:

phpunit --version

Файл с тестируемым классом нужно подключить в классе который его будет тестировать:

require\_once dirname(\_\_FILE\_\_).'/../User.php';

или использовать автозагрузчик классов. Можно так же для автозагрузки подключить и использовать Composer, правда в таком случае проще и установку phpunit делать с его же помощью.  
  
Тестирующий класс (содержащий тесты) должен наследовать от класса phpunit:

class ClassnameTest extends PHPUnit\_Framework\_TestCase{}

### 2.2 Установка вручную для Linux.

Делаем согласно документации [https://phpunit.de](https://phpunit.de/manual/5.7/en/installation.html)

wget https://phar.phpunit.de/phpunit-5.7.phar

chmod +x phpunit-5.7.phar

sudo mv phpunit-5.7.phar /usr/local/bin/phpunit

Тут первой строкой скачиваем файл с phpunit, далее даем файлу права на запуск b потом переименовываем файл **phpunit-5.7.phar** в **phpunit** для удобства.

Так же можно скачать самую последнюю версию phpunit командой:

wget https://phar.phpunit.de/phpunit.phar

После установки проверяем phpunit командой:

phpunit --version

## 3. Установка с помощью PEAR

**PEAR** - это предшественник Composer, репозиторий PHP-классов. В настоящее время в данном репозиторий находится устаревшая версия PHPUnit, тем не менее опишу и данный способ установки данного фреймворка для тестирования.  
  
Прежде всего нужно установить (активировать) PEAR.  
Опишу установку для OpenServer, он содержит в себе все необходимые компоненты для этого.  
В папке с интерпретатором php (например W:\modules\php\PHP-5.6) находится файл go-pear.bat и папка PEAR.  
Для установки перейти из командной строки в корень папки php:

cd W:\modules\php\PHP-5.6

выполнить файл go-pear.bat, для этого набрать в консоли:

go-pear

выбрать в диалогах:  
(system|local) [system] : localPlease confirm local copy by typing 'yes' : yesнажать EnterWould you like to alter php.ini <W:\modules\php\PHP-5.6\php.ini>? [Y/n] : y  
нажать Enter  
  
проверить установился ли PEAR:

pear list

Теперь устанавливаем PHPUnit:

pear install PHPUnit

проверить появился ли доступ:

phpunit --version

Тестирующий класс (содержащий тесты) должен наследовать от класса phpunit:

class ClassnameTest extends PHPUnit\_Framework\_TestCase{}