

Тестирование

№ урока: 11 **Курс:** Веб разработка на PHP Symfony

Средства обучения: Eclipse PHP Development Tools, командная строка, любой современный браузер

Обзор, цель и назначение урока

В данном уроке рассматривается то, как можно использовать PHPUnit в Symfony приложениях для тестирования самого кода приложения. PHPUnit позволяет убедиться в том, что код работает стабильно и хорошо поддерживается.

Изучив материал данного занятия, учащийся сможет:

- Использовать PHPUnit в Symfony для написания небольших тестов, которые будут тестировать код приложения и проверять его работоспособность.
- Уметь пользоваться разделением окружений в Symfony приложении.
- Пользоваться тестовыми двойниками при потребности тестировать репозитории или Entity чтобы убедиться в корректной работе кода, который используется для записи в БД.
- Писать Functional тесты, которые имеют в себе несколько функций, разделяющих тест на несколько этапов при этом.
- Делать фикстуры для тестов, при этом используя тестовую БД.

Содержание урока

1. Вступление в Unit и Function тестирование
2. Первый Unit тест с PHPUnit
3. Dataprovider
4. Тестовые двойники
5. Первый functional тест
6. Разделяем test, prod, dev окружения
7. Фикстуры для тестов

Резюме

- В Symfony приложении уже по умолчанию установлен PHPUnit, фреймворк для PHP который предназначен для того, чтобы писать модульные тесты и тестировать все приложение.
- Unit тестирование - представляет из себя тестирование каких-то небольших частей кода приложения. Для того чтобы запустить все тесты приложения, нужно воспользоваться командой в консоли:
`php bin/phpunit`
Эта команда запустит все тесты, которые на данный момент имеются в папке tests/, которая находится в корневой папке приложения. В случае если нам нужно протестировать какую-то отдельную папку с тестами, нам надо написать команду в таком виде:
`php bin/phpunit tests/Admin`
В результате выполнения данной команды будут выполнены тесты, которые находятся в папке tests/Admin.
- Для того чтобы создать какой-то тест в Symfony приложении, нужно воспользоваться командой, предоставленной от maker-bundle:

php bin/console make:test

После этого откроется генератор тестов и нужно будет выбрать название файла теста и его тип.

- Dataprovider представляет из себя отдельную функцию, которая содержит в себе данные, которые будут использованы в какой-то функции для тестирования определённого кода, но сама функция, которая и будет тестировать определённую часть кода и вызовет указанный dataprovider в аннотациях - чтобы получить данные для тестирования.
- В случае, если нужно сделать тестирование какого-то Entity или же репозитория, то нужно воспользоваться тестовыми двойниками. Тестовые двойники позволяют проводить тестирование работы с БД, при этом не создавая ничего лишнего в самой базе данных во время тестов. Это нужно всегда использовать, когда речь идёт об функциях, где есть любая работа с базами данных и их объектами.
- Function тесты являются тестами, которые имеют в себе несколько функций для выполнения тестов, и чаще всего они делятся на несколько этапов. Например, этап подготовки (функция в которой идёт подбор всех переменных и т.п.), этап выполнения теста (функция выполнения самого теста), этап завершения теста (функция завершает процессы в тесте).
- По умолчанию Symfony приложение уже разделено на 3 окружения, это test, prod и dev. test — это окружение, предназначенное для выполнения тестов и тестирования приложения в общем. dev - окружение используется при разработке приложения. prod - это окружение, которое используется для уже готового приложения к использованию. Чтобы изменить окружение в приложении, нужно зайти в файл .env и поменять значение переменной APP_ENV, вписав там окружение на которое мы хотим переключиться.
- В случае, если нужно будет использовать фикстуры, которые предназначены именно для тестов, то рекомендуется создать для этого тестовую базу данных и уже в неё загружать тестовые фикстуры, данные которых будут использоваться при тестах.

Закрепление материала

- Что такое Unit тесты и как их создавать?
- Для чего нужны dataprovider в тестах?
- Зачем нужно использовать тестовые двойники?
- Как использовать Functional тесты?
- Какие есть окружения в приложении Symfony и как на них переключаться?

Дополнительное задание

Задание

Сначала идёт создание самого обычного Unit теста, в котором проверяется истинность ответа. После этого уже рассматривается создание unit теста который использует внутри себя dataprovider для того, чтобы иметь данные, которые используются во время теста. После того идёт создание тестовых двойников определённого Entity и репозитория, чтобы проверить как выполняется работа через тестовых двойников с работой БД. При этом сначала создаётся объект Entity, а потом уже через тестовый двойник репозитория он достаётся и проверяется его правильность. Узнав, что такое тестовые двойники, переходим к использованию Functional тестов. Для этого создаём functional тест с типом WebTestCase и внутри функции теста прописываем проверку на наличие текста при переходе в определённый маршрут.

Самостоятельная деятельность учащегося

Задание 1

Создать Unit тест, который будет тестировать контроллер с функцией, где есть вывод на страницу фразы "unit testing". В тесте сделать проверку - выводится ли данная надпись при переходе в маршрут.

Задание 2

Создать тест, где будет идти тестирование деления трех цифр. В данном случае нужно создать dataprovider, который предоставит функции, выполняющей тест, данные для теста в виде массива.

Задание 3

Написать functional тест, который будет тестировать репозиторий любого доступного Entity в приложении. Главным будет здесь сделать запрос на создание нового объекта в БД и после этого получение только что созданного объекта, через тестовый двойник репозитория.

Рекомендуемые ресурсы

Официальная документация Symfony

<https://symfony.com/doc/current/index.html>

Документация Symfony на рус. языке (имеет в себе некоторые устаревшие материалы)

<https://symfony.com.ua/doc/current/index.html>