

Интеграционное тестирование компонентов ASP.NET Core MVC

№ урока: 4 **Курс:** Тестирование ASP.NET Core MVC приложений

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

Вспомнить, что такое интеграционный тест. Разобраться со структурой интеграционных тестов, рассмотреть признаки хорошего интеграционного теста и типичные ошибки. Вспомнить компоненты ASP.NET Core MVC и проследить взаимосвязь компонентов и интеграционных тестов. Рассмотрение демо проекта.

Изучив материал данного занятия, учащийся сможет

- Разбираться в том, как правильно писать интеграционные тесты для компонент фреймворка.
- Разбираться в том, каковы признаки хорошего интеграционного теста.
- Разобраться в том, как избежать типичных ошибок при реализации интеграционного теста.
- Рассмотреть на практике, как и с чего начинать писать интеграционные тесты для компонентов фреймворка.

Содержание урока

1. Что такое интеграционный тест
2. Признаки хорошего интеграционного теста
3. Структура интеграционного теста
4. Что может проверять интеграционный тест
5. Типичные ошибки в интеграционных тестах
6. Компоненты фреймворка: что это и как это тестировать
7. Демо

Резюме

- **Интеграционный тест** – медленная проверка группы взаимодействующих компонентов программы.
- **Интеграционный тест всегда требует** поднятие группы из нескольких компонентов программы для их тестирования. Из этого вытекает то, что он медленный, относительно unit-тестов.
- **Внешние зависимости** бывают контролируемые и неконтролируемые.
- **Контролируемые внешние зависимости** — это те, которые полностью зависят от нас, используются реальные экземпляры внешних зависимостей для интеграционного тестирования.
- **Неконтролируемые внешние зависимости** — это те, которые даже частично от нас не зависят. Так как мы не можем гарантировать их поведение полностью, используются моки этих внешних зависимостей для интеграционного тестирования, для эмуляции их полного контроля.
- **Хороший интеграционный тест** – не использует лишних внешних зависимостей и не дублирует unit-тесты!

- **Плохой интеграционный тест** – требует слишком много внешних ресурсов, работает за нестабильное время, не приносит дополнительных гарантий!
- **Интеграционный тест обязан** предоставлять гарантии больше, чем просто сумма unit-тестов, так как система состоит из её unit-ов и взаимодействия между ними.
- **Компоненты ASP.NET Core MVC** – Routing, Validation, Configuration, Security.
- **Специфика интеграционных тестов компонентов такова**, что интеграционные тесты этих компонентов, а также интеграционные тесты Controller & Middleware, можно смело объединять в один тест. В случае интеграционных тестов это оправданно, по самой их природе проверки группы взаимодействующих компонентов программы.
- **Необходимо выбрать что важнее** – если скорость получения ответа, тогда unit-тесты, если же важно проверить систему в целом, и добиться более высоких гарантий – тогда необходимо выбирать интеграционные тесты.
- **Фокусируйтесь больше** на интеграционных тестах, которые проверяют логику конкретно вашего приложения. Сам фреймворк уже проверен его разработчиками, не стоит делать лишнюю работу.

Закрепление материала

- Что такое интеграционный тест?
- Опишите каждый термин из определения.
- Назовите признаки хорошего интеграционного теста.
- Что может проверять интеграционный тест?
- Какова типичная структура интеграционного теста?
- Каковы типичные ошибки в интеграционных тестах?
- Какие бывают внешние зависимости?
- Опишите структуру типичного интеграционного теста.
- Перечислите основные компоненты ASP.NET Core MVC и расскажите про их взаимосвязь с интеграционными тестами.

Дополнительное задание

Задание

Напишите по простому примеру для каждого компонента фреймворка и напишите для них интеграционные тесты.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Дополнительно найдите сопутствующую информацию по теме. Составьте сводную таблицу по основным понятиям, найдите схожие и отличающиеся моменты в описаниях, постарайтесь сформулировать своё понимание данной темы.

Задание 3

Реализуйте необходимые компоненты фреймворка для проекта из предыдущих уроков. Настройте сначала роутинг, добавьте валидацию входящих параметров, добавьте всю необходимую конфигурацию. Добавьте также любую безопасность по вашему выбору.

Задание 4

Напишите интеграционные тесты на компоненты из предыдущего задания. Используйте любой тестовый фреймворк на языке программирования C# и любые библиотеки, помогающие написанию интеграционных тестов.

Рекомендуемые ресурсы

Test ASP.NET Core MVC apps

<https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/test-asp-net-core-mvc-apps>

Integration tests in ASP.NET Core

<https://docs.microsoft.com/en-us/aspnet/core/test/integration-tests?view=aspnetcore-3.1>

Test naming

<https://enterprisecraftsmanship.com/posts/you-naming-tests-wrong/>

Routing

<https://docs.microsoft.com/en-us/aspnet/core/fundamentals/routing?view=aspnetcore-3.1>

Model validation

<https://docs.microsoft.com/en-us/aspnet/core/mvc/models/validation?view=aspnetcore-3.1>

Configuration

<https://docs.microsoft.com/en-us/aspnet/core/fundamentals/configuration/?view=aspnetcore-3.1>

Security

<https://docs.microsoft.com/en-us/aspnet/core/security/?view=aspnetcore-3.1>