

Интеграционное тестирование базы данных

№ урока: 6 **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 в том, какие бывают виды интеграционного тестирования базы данных. Разбираться в видах баз данных, применяемых для интеграционного тестирования. Рассмотреть варианты изоляции интеграционных тестов для базы данных. Рассмотрение демо проекта.

Изучив материал данного занятия, учащийся сможет

- Разбираться в том, как правильно писать интеграционные тесты для базы данных.
- Разбираться в том, какие есть виды баз данных, применяемых для интеграционного тестирования.
- Разбираться в том, какие есть варианты изоляции интеграционных тестов для базы данных.
- Рассмотреть на практике, как и с чего начинать писать интеграционные тесты для базы данных.

Содержание урока

1. Напоминание про интеграционные тесты
2. База данных: что это и как это тестировать
3. Демо

Резюме

- **Интеграционный тест** – медленная проверка группы взаимодействующих компонентов программы.
- **База данных** – совокупность логически структурированных данных, хранящихся и обрабатывающихся в вычислительной системе.
- **На практике**, использование базы данных в ASP.NET Core MVC приложениях зачастую начинают с использования ORM Entity Framework Core и создания соответствующей реляционной базы данных MS SQL.
- **Виды баз данных:** продуктовая, SQLite, InMemory.
- **Интеграционное тестирование с помощью продуктовой базы данных** это наиболее оптимальный вариант по реализации, так как даёт максимальное количество гарантий, что всё будет работать так как задумано. Недостаток скорости тестирования данного варианта можно скомпенсировать различными вариантами изоляции тестов.
- **SQLite база данных** хороша для ускорения прохождения интеграционных тестов, но обладает фатальным недостатком – не предоставляет всех возможностей полноценной продуктовой базы данных и может вести себя по-другому в различных мелочах, которые, складываясь в общую картину, полностью ломают поведение приложения. Даёт намного меньше гарантий чем тестирование на полноценной продуктовой базе данных, но и работает намного быстрее.

- **InMemory база данных в Entity Framework Core** предназначена для тестирования EF Core самими разработчиками фреймворками, так что в интеграционных тестах пользователей фреймворка она не используется.
- **Варианты изоляции тестов:** транзакция, схема базы данных, новая база.
- **Вариант изоляции в виде транзакций** идеально подходит для простейших CRUD приложения, но обладает фатальным недостатком невозможности логики, которая уже в себе содержит транзакции. Разумеется, база данных должна поддерживать транзакции для данного вида изоляции.
- **Вариант изоляции в виде новой базы данных** - это самый простой, очевидный и универсальный вариант, но и самый медленный из-за необходимости подготовки новой базы данных.
- **Вариант изоляции в виде схемы базы данных** наиболее оптимальный, так как создать новую схему и применить на неё миграции намного быстрее чем создавать физически новую базу данных, при этом наши тесты изолированы друг от друга. Разумеется, база данных должна поддерживать схемы для данного варианта изоляции.
- **На самом деле, есть ещё нереляционные базы данных**, но их проверка мало чем отличается от реляционных, кроме, разве что, вариантов изоляции тестов, так как в нереляционных базах данных может не быть транзакций или схемы (смотрите CAP теорему для взаимосвязи между различными видами баз данных).

Закрепление материала

- Что такое интеграционный тест?
- Что такое база данных?
- Какие виды баз данных используются в интеграционном тестировании?
- Какие варианты изоляции интеграционных тестов используются для базы данных?

Дополнительное задание

Задание

Разверните простую InMemory базу данных, напишите некоторую логику и интеграционные тесты для неё. Оцените ограничения InMemory базы данных.

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

Создайте базу данных и напишите весь необходимый код для поддержки данной базы в проекте из предыдущих уроков. Используйте Entity Framework Core.

Задание 4

Напишите интеграционные тесты на базу данных из предыдущего задания. Попробуйте все варианты изоляции тестов. Оцените их ограничения. 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/>

EF Core testing

<https://docs.microsoft.com/en-us/ef/core/miscellaneous/testing/>

Database tests isolation

<https://www.thinktecture.com/en/entity-framework-core/isolation-of-integration-tests-in-2-1/>