

Примечание:

Задание сформулировано для проекта по курсу БД, как самого недавнего для студентов. В качестве объекта тестирования может быть выбран любой многокомпонентный проект, написанный или изученный студентом.

Лабораторная работа № 2. (После лекции №4)

Задание:

1. Написать integration-тесты для компонентов доступа к данным и бизнес логики курсовой работы по БД
2. Написать E2E-тест для демонстрационного сценария (например того, который был использован при защите курсовой работы)
3. Организовать запуск тестов из лабораторной работы №1 и вновь добавленных в рамках работы №2 в любой, на усмотрение студента, CI\CD среде (GitLab, Jenkins, TeamCity, etc.)
4. Проимитировать действия из E2E теста с помощью средства для отправки запросов, снять лог с помощью средства захвата траффика

Требования

1. Интеграционные тесты должны включать в себя взаимодействие с БД
2. Для тестов должен инициализироваться отдельный инстанс БД
3. Инстанс БД поднимать с помощью последовательного запуска скриптов – генерация таблиц, заведение тестовых данных, откат состояния БД на то, которое было до прогона тестов
4. Порядок запуска тестов в CI\CD: unit → integration → e2e
5. Если один из этапов свалился, последующие запускать не нужно
6. Если свалился integration или e2e этап, то требуется принудительно откатить БД на состояние до запуска тестов (но не на состояние, когда таблицы ещё не созданы)
7. Если используется service bus / message broker / шина / и т.д. и т.п. то после аварийного или успешного завершения тестов требуется произвести вычитывание всех сообщений из очереди
8. Если в проекте используется хранение активных сессий пользователей в том или ином виде, то требуется предусмотреть завершение всех сессий после аварийного или успешного завершения тестов
9. Не требуется включать этап тестирования GUI в E2E тест
10. Убедиться, что тесты могут быть запущены несколькими разработчиками локально одновременно без влияния на результаты друг друга
11. Рекомендации по структуре файлов с тестами:
 - a. один файл – один тест из нескольких шагов (секций act, assert)
 - b. arrange по возможности желательно сделать один раз в начале теста;

с. если для unit-тестов рекомендуется использовать суффикс `test\Test_test\test_` и т.д. в зависимости от языка программирования, то для интеграционных также лучше использовать соответствующий суффикс, например для Java это `ITCase` (сокращение от `Integration Test Case`) – уточняйте для выбранного языка программирования сами

12. Желательно оформить запуск тестов в виде отдельного контейнера `docker`:
выкачивается репозиторий с кодом, устанавливаются все внешние зависимости и всё для запуск тестов на используемом языке программирования, контейнер вызывается внутри выбранной `CI\CD` среды