

Итоговая аттестация

Технологическая
специализация:

.Net разработчик

Итоговая аттестация по программе “.Net разработчик”

Уважаемые студенты, перед вами задание для итогового проекта, который вы должны выполнить после прохождения курсов “Разработчик C#”,

Описание итоговой аттестации:

- 1) Цель задания: *повторить и отработать на практике материал, изученный в ходе программы.*
- 2) Проект включает в себя **обязательное задание** и **дополнительные задания**.
 - Обязательное задание необходимо выполнить для получения диплома. За него ставится оценка.
 - Дополнительные задание можно выполнить для портфолио и также сдать и получить обратную связь. Его выполнение полезно, но не влияет на оценку.
- 3) Инструменты, которые обязательно нужно использовать для **обязательного задания**: C#, PostgreSQL
- 4) Формат сдачи: ссылка на подписанную и доступную для просмотра копию данного шаблона с выполненными заданиями **в синих полях**. [Инструкция](#).

Обязательное задание

Проект бекенд “мессенджер-приложения” на ASP.Net

Краткое описание задания

Используя фреймворк ASP Net Core создайте набор бекенд сервисов посредством которых пользователи смогут регистрироваться и обмениваться сообщениями. Приложение должно содержать минимум два микро-сервиса: API регистрации пользователей и API получения и обмена сообщениями. Организуйте доступ к сервисам через API Gateway

Подробное описание задания

Создайте проект ASP.Net Core и приложение(я) почтового ящика. Создайте два микросервера: работа с пользователями и работа с сообщениями.

Сервис пользователей

Обладает следующим функционалом:

- Сервис должен уметь регистрировать пользователей предоставляя соответствующий метод API. В качестве имени пользователя нужно использовать email, в качестве пароля произвольную строку
- Сервис должен уметь добавлять и аутентифицировать пользователей на основе асимметричного шифрования RSA
- Методы
 - Добавить администратора (первый пользователь в системе)
 - Добавить пользователя (обязательна проверка на дублирующиеся имена/адреса)
 - Получить список пользователей
 - Удалить пользователя (доступ только у администратора), администратор не может удалить сам себя
 - Метод возвращающий ID пользователя при обращении с JWT-токеном (для проверки работоспособности API)
- База данных
 - Пользователи

- Роли
- ID пользователя, который будет использоваться в сервисе обмена сообщениями должен быть добавлен в JWT(Claim) для использования в качестве ID в сервисе сообщений

Сервис сообщений

Предназначен для обмена сообщениями между авторизованными пользователями. Обладает следующим функционалом:

- Сервис позволяет отправлять сообщения от имени авторизованного пользователя.
- Сервис позволяет получать сообщения предназначенные для авторизованного пользователя
- Сервис помечает полученные сообщения во избежание повторной отправки
- Сервис получает только те сообщения которые не помечены как полученные
- Методы
 - Получить сообщения
 - Отправить сообщение
- База данных
 - Сообщения
- Пользователи
 - Авторизуются через сервис пользователей
 - Claim'ы JWT должны содержать Guid пользователя

Инструменты и рекомендации

Используется следующие инструменты:

- Autofac
- AutoMapper (не обязательно, но будет плюс)
- Postgresql в качестве базы данных

Реализуйте:

- Конфигурацию базы данных через конфиг-файл
- AutoMapper (будет плюс)
- Проверку длины пароля (будет плюс)
- Проверку сложности пароля
- Проверку валидности имени пользователя (email в соответствии с шаблоном)

Тестирование

Напишите Unit-тесты для вашего решения.

Postman

Для тестирования вашего решения подготовьте коллекцию в Postman покрывающую всех API-вызовы обоих сервисов.

Дополнительное задание

Реализуйте API-Gateway объединяющий методы обоих сервисов, настройте Swagger для API Gateway

Описание задания 1

Используя Ocelot реализуйте API-Gateway сервис предоставляющий доступ ко всем методам сервиса основного задания

Описание задание 2*

Настройте Swagger for Ocelot таким образом чтобы у после запуска API Gateway Swagger можно было получить доступ ко всем методам обоих сервисов