X`ПИС-3, ПОИТ

**Лабораторная работа 10**

**ASPA** – приложение ASP.NET CORE

♣ – задания, требующие самостоятельного изучения (нет в лекциях)

Оформление html-разметки на усмотрение разработчика

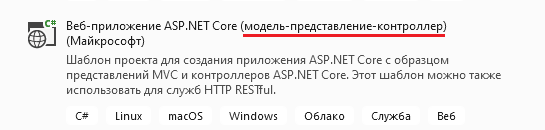
**Задание 1.**

1. Разработайте ASPA, применив следующий шаблон.

Имя решения: **ASPA**

Имя проекта: **ASPA0010\_1**

Версия .NET: **7.0** или **8.0**



1. Разработайте REST-сервис на основе технологии ASP.NET CORE WEB API.
2. Ресурс представляет собой) коллекцию RESULTScoстроковыми (string)значениями и целочисленными ключами, ключи формируются автоматически (auto increment)**.**
3. Сервис должен обрабатывать следующие запросы.

|  |  |  |
| --- | --- | --- |
| Метод | URI | Response В формате JSON |
| GET | /api/Results | вся коллекция  допустимые статусы: 200, 204, 401 |
| GET | /api/Results/{k:int} | элемент с ключом k  допустимые статусы: 200, 401, 404 |
| POST | /api/Results/  Body: {“value”: *value*} | добавленный новый элемент  допустимые статусы: 201, 400, 401 |
| PUT | /api/Results/{k:int}  Body:{“value”: *value*} | измененный элемент с ключом k  допустимые статусы: 200, 400, 401,404 |
| DELETE | /api/Results/{k:int} | удаленный элемент с ключом k  допустимые статусы: 200, 401, 404 |

1. Коллекция сохраняется в файловой системе сервера в JSON-формате.
2. Доступ к коллекции осуществить с помощью потокобезопасного (блокировка JSON-файла) Transient-сервиса c именем ***Results***. Сервис должен быть внедрен в контроллер ***ResultsController***. Сервис должен быть реализован как отдельная библиотека (***ResultsCollection.dll***), оформлен как Nuget-пакет (BSTU.Results.Collection) в локальном Nuget-репозитории.
3. Добавьте в приложение **ASPA0010\_1** поддержку OpenAPI.
4. Протестируйте работоспособность приложения с помощью OpenAPI.

**Задание 2.**

1. Сервис **ASPA0010\_1** должен поддерживать Cookie-аутентификацию (Identity Framework) и две роли: READER (метод GET), WRITER (методы POST, PUT, DELETE). Аутентификация должна быть реализована с помощью Scope-сервиса с именем ***Authenticate***. Сервис должен быть внедрен в контроллер ***ResultsController***. Сервис должен быть реализован как отдельная библиотека (***Results Authenticate.dll***), оформлен как Nuget-пакет (BSTU.Results.Authenticate) в локальном Nuget-репозитории. Добавьте конечные точки

|  |  |  |
| --- | --- | --- |
| Метод | URI | Response |
| POST | /api/Results/SignIn  Body: {“login”: *login*, “password”: *password*} | Cookie-токен  допустимые статусы: 200, 400, 404 |
| GET | /api/Results/SignOut | JWT-токен  допустимые статусы: 200, 401 |

1. Протестируйте работоспособность приложения с помощью OpenAPI.
2. **КОНЕЦ**