

## РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ CMS

<sup>1</sup>Рудоловский М. Н., <sup>2</sup>Кондратёнок Е. А.

<sup>1</sup>*Беларусский национальный технический университет,  
Минск, Беларусь, mrudalovksiy@gmail.com,*

<sup>2</sup>*Беларусский национальный технический университет,  
Минск, Беларусь, elena\_kondr@tut.by*

**Аннотация.** В данной статье рассмотрена возможность разработки серверной части CMS.

**Ключевые слова:** сервер, CMS, WEB-разработчик, MediatR, токен.

**Abstract.** In this article dascusses the possibility of developing a server part of a CMS.

**Key words:** server, CMS, WEB-developer, MediatR, token.

За последние 10 лет произошло множество прорывов в области разработки WEB-приложений. WEB-приложений становились все более функциональными, большими. Для упрощения разработки некоторых WEB-приложений разработчики часто используют CMS (система управления содержимым). CMS – это программа, при помощи которой создаются и наполняются веб-сайты [1].

Основное назначение CMS – предоставить WEB-разработчикам быстрый и удобный инструмент для разработки WEB-приложений.

В процессе изучения информации о CMS были изучены другие похожие системы, были приняты во внимание недостатки и извлечены плюсы из других таких систем таких, как WordPress или Joomla.

В ходе разработки CMS использовались такие технологии, как .Net 6.0, Entity Framework Core, MS SQL, Automapper, Swagger, EF Fluent API, Angular.

CMS обладает следующими функциями:

- предоставление инструментов для создания содержимого, организация совместной работы над содержимым;
- управление содержимым: хранение, соблюдение режима доступа, управление потоком документов;
- публикация содержимого;
- представление информации в виде, удобном для навигации, поиска.

CMS часто называют движком сайта, например проводя аналогию с игровыми движками, которые так же располагают инструментами для разработки игрового продукта.

CMS разработана на .Net 6.0, с помощью языка программирования C#. Было решено использовать чистую архитектуру. Архитектура программы приведена на рис. 1, 2.

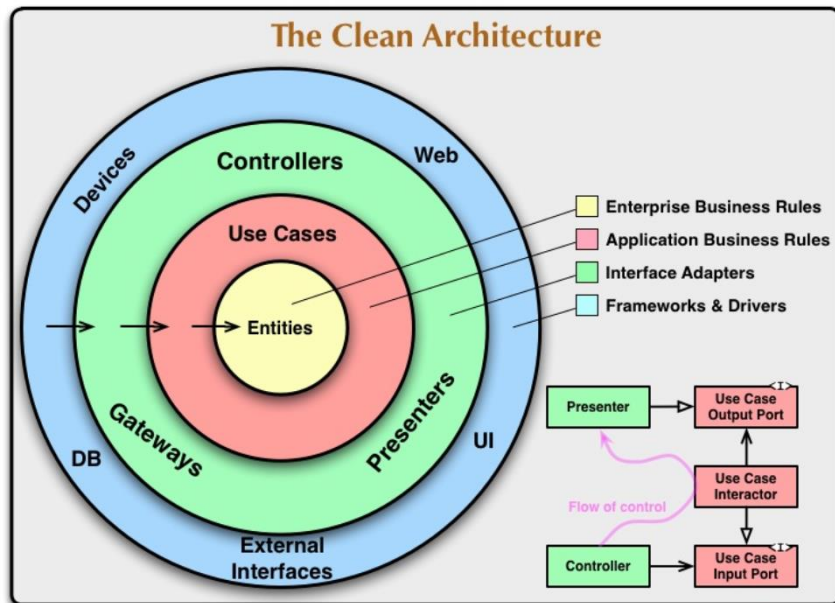


Рисунок 1 – Чистая архитектура [2]

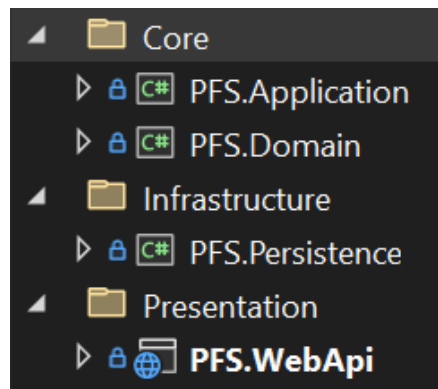


Рисунок 2 – Архитектура проекта

Архитектура состоит из следующих слоев:

1. Core содержит два подслоя: Application, Domain. В Domain хранятся сущности системы. В Application хранятся команды и запросы для сущностей из Domain.

2. Infrastructure содержит всю информацию о базе данных, в которой хранятся сущности слоя Domain.

3. Presentation содержит WebApi системы.

Взаимодействие между уровнями происходит с помощью паттерна MediatR.

MediatR – очень популярная библиотека с открытым исходным кодом для C#. Он следует шаблону CQRS и посреднику, который представляет собой шаблон поведенческого проектирования, который позволяет объектам взаимодействовать друг с другом без тесной связи [3].

В данной программе содержится 3 основных модуля:

- Core;

- Infrastructure;
- WebApi.

В модуле Core содержатся сущности программы и бизнес-логика программы. Для сущности контента предусмотрены следующие функции: создать контент, удалить контент, обновить контент, получить детали контента. Исходя из вышесказанного становится понятно, что бизнес-логика работает по парадигме CQRS. CQRS-принцип или парадигма CQRS разделяет назначение запросов (например при чтении данных) и команд на обработку данных [4]. То есть у сущности контента имеется три команды и один запрос. Структура модуля Core приведена на рис. 3.

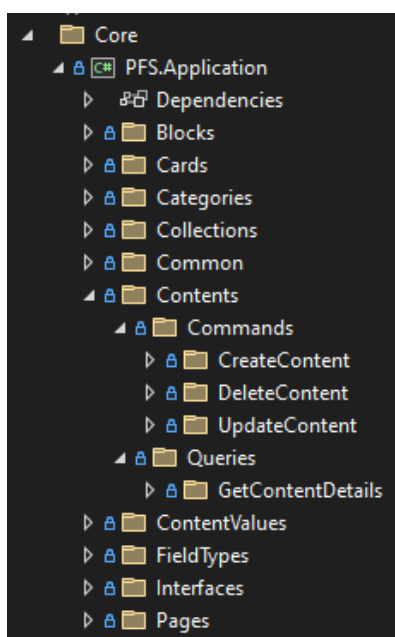


Рисунок 3 – Структура модуля Core

В модуле Infrastructure содержится контекст и инициализатор базы данных, с которой работает программа. Сущности базы данных в этом модуле не содержатся, в этом модуле содержится только «контекстная» информация, то есть какие таблицы в базе данных имеются, конфигурация типов в таблицах. Структура модуля Core приведена на рис. 4.

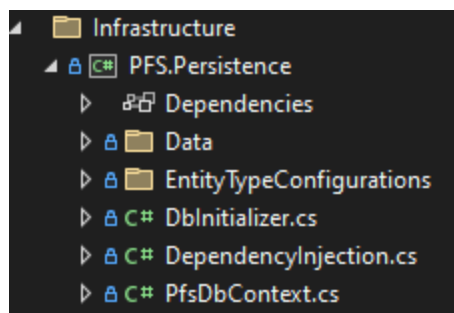


Рисунок 4 – Структура модуля Infrastructure

В модуле WebApi происходит все взаимодействие серверной части с клиентской частью. В данном модуле содержатся контроллеры, логирование, модели, сервисы аутентификации. Контроллеры модуля WebApi лишь отвечают за обработку запроса, они передают данные в модуль Core, получаемые из запроса, а сама логика, как раньше было упомянуто, содержится в модуле Core. Структура модуля Core приведена на рис. 5.

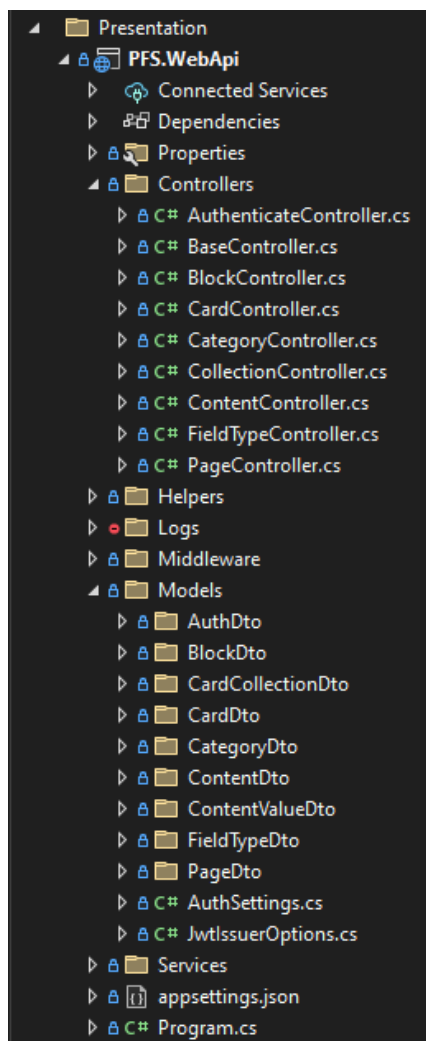


Рисунок 5 – Структура модуля WebApi

Для данной CMS так же предусмотрена аутентификация посредством JWT токена. JSON Web Token (JWT) – это открытый стандарт для создания токенов доступа, основанный на формате JSON. Как правило, используется для передачи данных для аутентификации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения подлинности аккаунта [5].

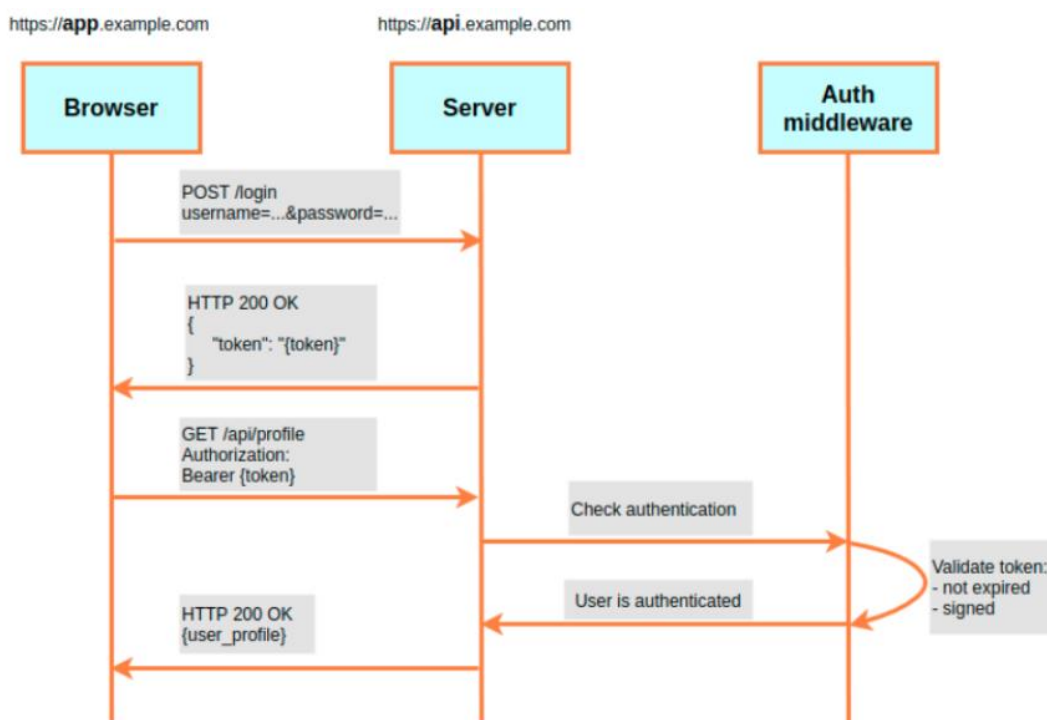


Рисунок 6 – Аутентификация с помощью токенов [4]

Примерный алгоритм создания страницы в системе будет выглядеть следующим образом:

- применить команду «Создать страницу»;
- применить команду «Добавить блок в страницу»;
- применить команду «Создать контент»;
- применить команду «Добавить контент в блок»;

После выполнения вышеуказанных команд будет создана страница с готовым контентом в ней. Страницу, контент, блоки и т. д. можно создавать, изменять, удалять, получать информацию и т. д.

В результате работы был создан программный интерфейс серверной части CMS, также была разработана и описана архитектура CMS, выбраны технологии, программные средства и, непосредственно, язык программирования для реализации поставленной задачи.

Дальнейшее направление развития проекта – это разработка клиентской части приложения, для полноценной работы CMS.

#### Список использованных источников:

1. Что такое система управления содержимым сайта и как ее выбрать [Электронный ресурс]. – Режим доступа: [https://www.umi-cms.ru/support/poleznye-stati/kak\\_sozdat\\_sajtvizitku/#:~:text=D0%A1%D0B8](https://www.umi-cms.ru/support/poleznye-stati/kak_sozdat_sajtvizitku/#:~:text=D0%A1%D0B8). – Дата доступа: 10.11.2023.

2. Чистая архитектура [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/269589/>. – Дата доступа: 10.11.2023.

3. Документация Mediatr [Электронный ресурс]. – Режим доступа: <https://q.agency/blog/simplifying-complexity-with-mediator-and-minimal-apis/#:~:text=MediatR%20is%20a%20very%20popular,other%20without%20being%20>. – Дата доступа: 10.11.2023.

4. Документация CQRS [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/CQRS>. – Дата доступа: 10.11.2023.

5. Документация JSON Web Token [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/JSONWebToken>. – Дата доступа: 10.11.2023.