

- 1. [About](#)
  - 1.1. [Описание](#)
- 2. [Build & Run](#)
- 3. [Stack](#):
  - 3.1. [Frontend](#):
  - 3.2. [Backend](#):
  - 3.3. [Infrastructure](#):
- 4. [Media](#)

# Forum/Blog Project

---

Автор: Мрясов М.С

Тема: Форум/Блог

## 1. About

---

Форум/блог - простой проект блога с возможностью регистрации пользователей, создания постов, коментирования постов.

особенности проекта, аутентификация и авторизация пользователей, разграничения доступа пользователем к части функционала по роли. Построен проект на бэкенде ASP.NET Rest Api и фронтенде Node.js React.

### 1.1. Описание

#### **инфраструктурная часть:**

- Comment - класс сущности БД.
- Post - класс сущности БД.
- ApplicationDbContext - класс контекста БД работающий сущностями.

#### **Контроллеры и DTO:**

- PostController - контроллер управления постами.
- CommentController - контроллер управления комментариями.
- AccountController - контроллер управления аккаунтами.
- PostModel - DTO постов.
- CommentModel - DTO комментариев.
- LoginModel - DTO данных авторизации.
- RegisterModel - DTO данных регистрации.

#### **Страницы и компоненты:**

- main.jsx и App.jsx - точка входа и основной компонент приложения.
- AuthContext.jsx - Контекст авторизации, сохраняющий **кред'ы** (авторизационные данные).
- CreatePost.jsx - страница создания поста.

- Home.jsx - домашняя страница.
- Register.jsx - страница регистрации.
- Login.jsx - страница авторизации.
- Post.jsx - страница поста.
- CommentList.jsx - компонент списка комментариев используется в Post.jsx.
- Header.jsx - компонент заголовка используется со всеми страницами.
- PostList.jsx - компонент списка постов используется в Home.jsx.

#### Эндпоинты:

- POST /api/account/login
  - POST /api/account/register
  - POST /api/create-role
- 
- POST /api/comments/{postId}
  - GET /api/comments/{postId}
  - PUT /api/comments/{commentId}
  - DELETE /api/comments/{commentId}
- 
- GET /api/posts/all
  - GET /api/posts/{id}
  - PUT /api/posts/{id}
  - DELETE /api/posts/{id}
  - POST /api/posts/create

## 2. Build & Run

### Download

```
git clone https://github.com/Plasmat1x/Term
cd ./Term
```

### Backend

```
cd ./Backend
dotnet build Backend.csproj
dotnet run Backend.csproj
```

### Frontend

```
cd ./Frontend
npm i
npm run dev
```

## 3. Stack:

---

### 3.1. Frontend:

- JS
- React
- Vite
- Tailwindcss
- Antd
- Reactrouter
- axios

### 3.2. Backend:

- .NET 7
- ASP.NET Core
- Identity
- EntityFramework
- Swashbuckle

### 3.3. Infrastructure:

- MSSQL 2016

## 4. Media

---



ERD диаграмма БД.

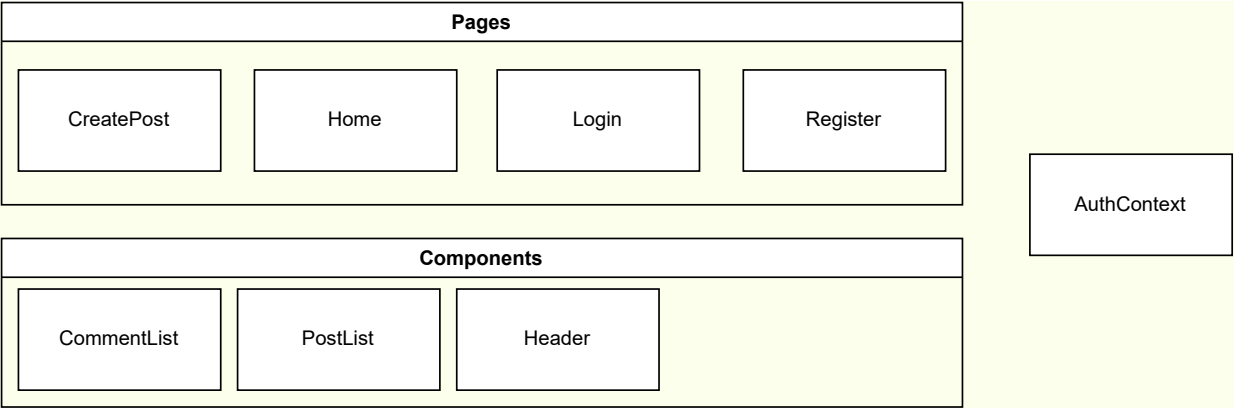


диаграмма компонентов фронтенда.

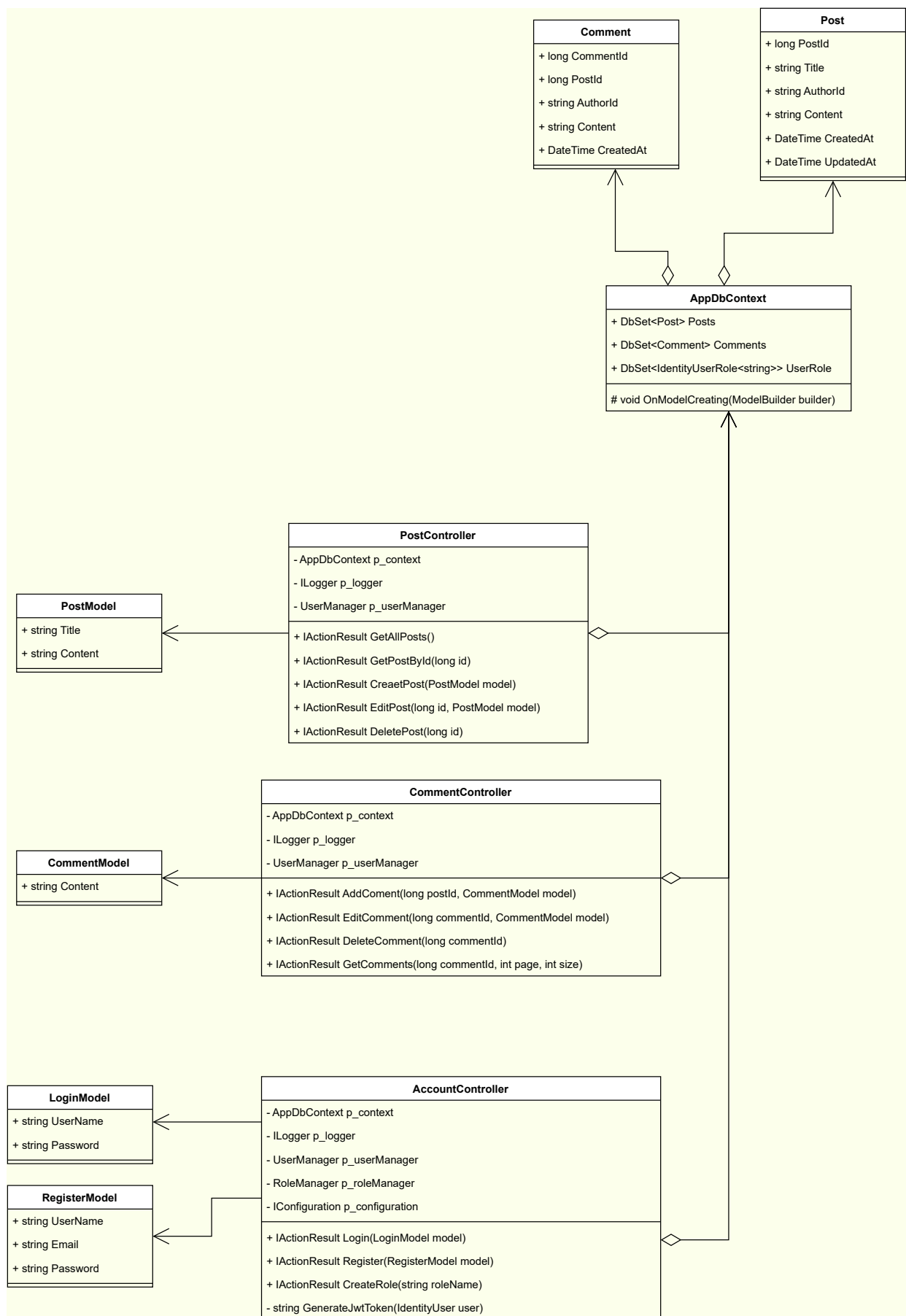


диаграмма классов бэкенда.

Account		^
POST	/api/account/login	✓
POST	/api/account/register	✓
POST	/api/account/create-role	✓
POST	/api/account/logout	✓
Comment		^
POST	/api/comments/{postId}	✓
GET	/api/comments/{postId}	✓
PUT	/api/comments/{commentId}	✓
DELETE	/api/comments/{commentId}	✓
Post		^
GET	/api/posts/all	✓
GET	/api/posts/{id}	✓
PUT	/api/posts/{id}	✓
DELETE	/api/posts/{id}	✓
POST	/api/posts/create	✓

## Эндпоинты api

Schemas		^
<pre> CommentModel {   content string nullable: true } </pre>		
<pre> LoginModel {   userName* string minLength: 1   password* string minLength: 1 } </pre>		
<pre> PostModel {   title string nullable: true   content string nullable: true } </pre>		
<pre> RegisterModel {   userName* string minLength: 1   email* string minLength: 1   password* string minLength: 1 } </pre>		

## схема api