

Введение в SeaBattle

SeaBattle - это многопользовательская онлайн-игра "Морской бой", разработанная с использованием современных технологий.

Особенности проекта

- Многопользовательский режим с использованием SignalR для real-time коммуникации
- Современный веб-интерфейс
- Система рейтинга игроков
- История игр и статистика
- REST API для интеграции

Технологии

- **Backend:** ASP.NET Core 8.0
- **Real-time коммуникации:** SignalR
- **База данных:** PostgreSQL с Entity Framework Core
- **API документация:** Swagger/OpenAPI
- **Контейнеризация:** Docker

Структура проекта

- **Controllers/** - REST API контроллеры
- **Models/** - Модели данных с XML-документацией
- **Services/** - Бизнес-логика
- **Hubs/** - SignalR хабы для real-time взаимодействия
- **Data/** - Контекст базы данных и миграции

Документация моделей

В проекте используются следующие основные модели:

1. **Game** - основная модель игры с полями для состояния игры, игроков и игрового поля
2. **GameHistory** - модель для хранения истории игр
3. **GameState** - перечисление состояний игры
4. **Position** - модель позиции на игровом поле
5. **ShotResult** - перечисление результатов выстрела
6. **ShotResultResponse** - модель ответа на выстрел
7. **User** - модель пользователя
8. **PlayerRanking** - модель рейтинга игрока
9. **LobbyInfo** - модель информации о лобби
10. **CellState** - перечисление состояний клетки

Все модели содержат подробную XML-документацию на русском языке.

Начало работы с SeaBattle

Требования к системе

Для работы с проектом необходимо:

- .NET 8.0 SDK
- PostgreSQL 15
- Docker (опционально)
- Visual Studio 2022 или VS Code с C# расширением

Установка и настройка

1. Клонировать репозиторий:

```
git clone https://github.com/Mug1vara97/SeaBattle
cd SeaBattle/Server
```

2. Восстановите зависимости:

```
dotnet restore
```

3. Настройте строку подключения к базе данных в `appsettings.json`:

```
{
  "ConnectionStrings": {
    "DefaultConnection":
      "Host=localhost;Database=seabattle;Username=your_username;Password=your_password"
  }
}
```

4. Примените миграции базы данных:

```
dotnet ef database update
```

5. Запустите приложение:

```
dotnet run
```

Запуск через Docker

1. Соберите Docker образ:

```
docker build -t seabattle .
```

2. Запустите контейнер:

```
docker run -p 5000:80 seabattle
```

Документация API

Документация API доступна через Swagger UI после запуска приложения по адресу:

<http://localhost:5000/swagger>

Структура решения

- **SeaBattle/** - основной проект
 - **Controllers/** - REST API контроллеры
 - **Models/** - модели данных с XML-документацией
 - **Services/** - бизнес-логика
 - **Hubs/** - SignalR хабы
 - **Data/** - работа с базой данных
- **GameServiceTests/** - модульные тесты

XML-документация

Все модели в проекте содержат подробную XML-документацию на русском языке. Документация включает:

- Описание назначения классов
- Описание свойств
- Описание методов и их параметров
- Примеры использования (где применимо)