

Project information

Created on
February 19, 2026

Name	Last update
 Backend.Reservationtool	2 weeks ago
 Frontend.Reservationstool	2 weeks ago
 resources	1 month ago
 .gitignore	25 minutes ago
 Makefile	3 weeks ago
 README.md	40 seconds ago
 run-load-test.ps1	2 minutes ago

 [README.md](#)

Reservationstool

A full-stack room reservation system featuring an ASP.NET Core backend and a React-based frontend.

Architecture

- **Backend:** ASP.NET Core with Identity, EF Core (PostgreSQL), and JWT authentication.
 - `Backend.Business` : Core business logic.
 - `Backend.Common` : Shared models, DTOs, and enums.
 - `Backend.DataAccess` : Database context and repository implementations.
 - `Backend.Reservationtool` : API controllers and host application.
- **Frontend:** React (Vite) with TypeScript, Tailwind CSS, and FullCalendar.
 - Uses [Orval](#) for automated API client generation from OpenAPI.
- **Infrastructure:** Dockerized environment with Traefik as a reverse proxy and PostgreSQL for data storage.

Getting Started

Prerequisites

- [Docker Desktop](#)
- [.NET 10.0 SDK](#)
- [Node.js \(v22+\)](#)

Environment Setup

1. Navigate to the `Backend.Reservationtool` folder.
2. Create a `.env` file (or copy from `.env.prod` for defaults). It should contain:

```
ASPNETCORE_ENVIRONMENT=Development
JWT_SECRET_KEY=YourSuperSecretKeyHere
JWT_Issuer=http://localhost:8080
JWT_Audience=http://localhost:8080
DB_PASSWORD=YourStrongDbPassword
```

```
DATA_VOLUME_NAME=res_db_local_data
```

Running the Application

The project includes a `Makefile` for simplified execution:

```
# Install frontend dependencies, build everything, and start containers
make run
```

Access the application at:

- **Frontend/API:** `http://localhost:8080`
- **Traefik Dashboard:** `http://localhost:8081`

Development

Database Migrations

To manage migrations locally, you must temporarily change the database host in your connection string to `localhost` since tools like `dotnet ef` run outside the Docker network.

Create a migration:

```
Add-Migration -StartupProject Backend.Reservationtool -Project Backend.DataAccess <MigrationName>
```

Remove last migration:

```
Remove-Migration -StartupProject Backend.Reservationtool -Project Backend.DataAccess
```

API Client Generation

The frontend automatically generates its API client using Orval.

To regenerate after backend changes:

1. Ensure the backend is running.
2. Run in `Frontend.Reservationtool`:

```
npm run generate-api
```

Load Testing

The project includes a load testing setup using `hey`:

1. Open PowerShell in the root directory.
2. Execute the load test script:

```
./run-load-test.ps1
```

3. The script will prompt for credentials, obtain a JWT, and run `hey` against the API.

Domain Model

