Deployment

- Home
- Components
  - InfoSymbolServer.Domain
  - **InfoSymbolServer.Infrastructure**
    - Data Access
    - Background Jobs
    - Notifications
  - InfoSymbolServer.Application
  - InfoSymbolServer.Presentation
  - InfoSymbolServer
- Versioning
- Configuration
- **Deployment** *(current)*

Overview

InfoSymbolServer is containerized using Docker and can be deployed using Docker Compose for development and testing environments. This page explains how to deploy the application using these tools.

Docker Compose Deployment

The project includes a `docker-compose.yml` file that defines the application stack:

```
version: '3.8'

services:
  infosymbolserver:
    image: infosymbolserver
    build:
      context: .
      dockerfile: InfoSymbolServer/Dockerfile
    ports:
      - "61578:61578"
    environment:
      - ASPNETCORE_ENVIRONMENT=Development
      -
ConnectionStrings__DefaultConnection=Host=db;Port=61579;Database=infosymbo
ldb;Username=postgres;Password=postgres
    depends_on:
      db:
```

```
        condition: service_healthy

  db:
    image: postgres:alpine
    container_name: infosymbol-postgres
    environment:
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgres
      - POSTGRES_DB=infosymboldb
      - PGPORT=61579
    volumes:
      - postgres-data:/var/lib/postgresql/data
    ports:
      - "61579:61579"
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U postgres -p 61579"]
      interval: 5s
      timeout: 5s
      retries: 5
      start_period: 10s

volumes:
  postgres-data:
```

Components:

1. **infosymbolserver**: The main application service
   - Built from the Dockerfile in the project
   - Exposes port 61578
   - Depends on the database service
   - Configured via environment variables
2. **db**: PostgreSQL database service
   - Runs on explicitly defined port 61579
   - Includes a health check to ensure readiness
   - Persists data using a named volume
   - Port is exposed to the host for external tools to connect

Deployment Steps:

1. Clone the repository:

```
git clone https://github.com/yourusername/infosymbolserver.git
cd infosymbolserver
```

2. Start the services:

```
docker-compose up -d
```

3. Verify the deployment:

```
docker-compose ps
```

4. Check the logs:

```
docker-compose logs -f infosymbolserver
```

5. Access the API:
   - Swagger UI: http://localhost:61578/swagger
   - API Endpoints: http://localhost:61578/api/v1/...

Stopping the Services:

To stop the services but keep the data volumes:

```
docker-compose down
```

To stop the services and remove the data volumes:

```
docker-compose down -v
```