

Deploy Documentation

Getting Started 🚩

1. Clone the repository:

```
git clone https://github.com/amosproj/amos2024ss05-knowledge-graph-extractor.git
```

2. Change directory into the project:

```
cd amos2024ss05-knowledge-graph-extractor
```

3. Copy the `.env.example` file to `.env` and update the values as needed:

```
cp Project/backend/.env.example Project/backend/.env
```

Initial Setup ⚙️

Ensure you have Docker and Python 3.11 or higher installed before proceeding with the setup.

Development Prerequisites

1. Create a virtual environment:

```
python -m venv venv
```

2. Activate the virtual environment:

```
source venv/bin/activate
```

3. Install the development requirements specific to your IDE for enhanced functionality and support:

```
pip install -r Project/backend/codebase/requirements.txt
```

Build the image and run the Container:

- If buildkit is not enabled, enable it and build the image:

```
DOCKER_BUILDKIT=1 COMPOSE_DOCKER_CLI_BUILD=1 docker-compose -f  
Project/backend/docker-compose.yml up --build -d
```

- If buildkit is enabled, build the image:

```
docker-compose -f Project/backend/docker-compose.yml up --build -d
```

Start Frontend server:

1. Navigate to the frontend directory:

```
cd Project/frontend
```

2. Install the frontend dependencies:

```
npm install
```

3. Build the frontend:

```
npm run dev
```

Alternatively, you can use the Makefile to run the Container and the frontend server from the **Project directory**:

- Start the Container and the Frontend Server:

```
make build-dev
```

Only start the Frontend Server:

```
make frontend-build-dev
```

Only start the Container:

```
make backend-build-dev
```

You can now access the application at http://localhost:8000. The development environment allows for immediate reflection of code changes.

Shortcuts

This project includes several shortcuts to streamline the development process.

Starting the Development Environment from the Root Directory

To build and run the development environment (backend and frontend at the same time), you can run the following command in the root directory of the project:

```
make build-dev
```

To stop the development environment (frontend and backend), press **Ctrl + C** to stop the npm server and run the following command:

```
make stop-dev
```

Separately Starting the Components

If you want to start the components separately, you can use the prefix **make backend-<command>/make frontend-<command>** for the separate backend and frontend commands or by navigating to the respective directories and running the Makefile commands.

Start only the frontend:

```
make frontend-build-dev
```

Example to start the backend separately:

```
make backend-build-dev
```

You can run backend commands using the following format:

```
make backend-<command>
```

Example: To run the tests in the backend:

```
make backend-tests
```

Or navigate to the backend directory and run the command:

```
cd Project/backend  
make build-dev
```

Example to start the frontend separately:

```
cd Project/frontend  
make start-dev
```

Show Help

You can always run the following command to see the available commands:

```
make help
```