

Production

Initial Setup

1. Install Docker Desktop:

- Go to docker.com and download the app
- Open Docker Desktop

2. Download and configure the Project:

- Go to github.com and download the latest release
- Unzip the project folder and place it somewhere on your machine
- Navigate to **Backend** and copy the file **.env_example** and paste it in the same folder named as **.env_prod**
- Open the file **.env_prod** and replace the variable values as follows:

```
TEST_STAGE=prod
PASSWORD={{correct_email_password}}
SECRET_KEY=Some_random_32byte_key
MONGODB_URL=mongodb://mongo:27017/
```

Build and Run the TalkTix Apps

1. **Build and start the containers:** open a terminal/shell (e.g., Terminal, Powershell, Bash, etc.) in the project`s root folder (amos2023ws01-ticket-chat-ai/) and execute (Copy and Paste into terminal and press Enter):

```
docker-compose up
```

2. **Use the apps:** go to the [User Documentation](#)
3. **Stop the containers and clean up:** open a shell in the project`s root folder (e.g., Terminal, Powershell, Bash, etc.) and execute:

```
docker-compose down
docker image prune -f -a
```

Development

Initial Setup

Clone the Repository:

```
git clone git@github.com:amosproj/amos2023ws01-ticket-chat-ai.git
```

Build and Deploy the Backend

Setup

1. **Install Python:** go to python.org
2. **Install MongoDB Community Server:** go to mongodb.com
3. **Navigate to Backend:**

```
cd amos2023ws01-ticket-chat-ai/Backend
```

4. **Install the Dependencies:**

```
pip install -r requirements.txt
```

5. **Set the Email Account Data:**

- Navigate into `./backend` directory. Rename `.env_example` to `.env` and set your Email Password inside the `.env`
- Open the `config.ini` and set your Email Account Name, IMAP- and SMTP Server

Running and Testing the Project

1. **Start Database:**

- Mac:

```
brew services start mongodb-community@7.0
```

- Windows:

```
elevate -w net start MongoDB
```

Alternative:

1. Add the path to `mongod.exe` to the environment variables in order to use it globally.
2. Create a storage directory at any location, for example, `...\data\db`, if you have not already done so.

3. Run **mongod** with path to storage location for example:

```
mongod --dbpath E:\workspace\Uni\ws23-24\AMOS\data\db
```

- Linux:

```
sudo systemctl start mongod
```

2. **Start the Backend API:**

```
uvicorn app.main:app --reload
```

3. **Use the endpoints:** go to [SwaggerUI](#) **OR** try [Postman](#) with the base URL <http://localhost:8000/api/v1> + router path

4. **Run the tests:**

```
pytest test/
```

Run Test Model

1. Navigate into `./backend/app/models/t5` directory.
2. Train the model by running the following command:

```
python train_t5_model.py
```

3. Test the trained model by running the following command:

```
python use_trained_t5_model.py
```

4. Test the untrained T5 model by running the following command:

```
python train_t5_model.py
```

Run the Email Proxy

1. Start Backend API

2. Navigate to `app/email` and run:

```
python main.py
```

Build and Deploy the Frontend

Setup

1. **Install newest Version of Node.js:** go to nodejs.org or on Linux:

```
sudo apt install npm
```

```
npm install -g n
```

```
n lts
```

2. **Navigate to Frontend:**

```
cd amos2023ws01-ticket-chat-ai/Frontend
```

3. **Install Angular CLI** (Optional but recommended):

```
npm install -g @angular/cli
```

4. **Install the Dependencies:**

```
npm install
```

Build

Run `ng build` to build the project. The build artifacts will be stored in the `dist/` directory.

Running unit tests

Run `ng test` to execute the unit tests via [Karma](#).

Running end-to-end tests

Run `ng e2e` to execute the end-to-end tests via a platform of your choice. To use this command, you need to first add a package that implements end-to-end testing capabilities.