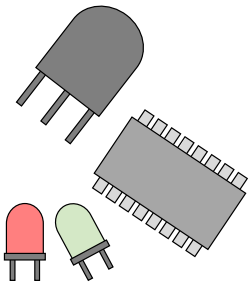


Explainable Similarity Detector

Das AMOS Projekt - WS 2021/2022

Explainable Similarity Detector

Build Documentation

Project Team 06

Content

Getting Started

Content

1. Install Mendix

Go to: https://marketplace.mendix.com/link/studiopro/ and download Mendix Studio Pro.

2. Clone Git Repository

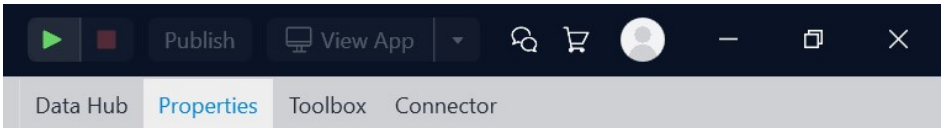
```sh
# SSH
git clone git@github.com:Re-Krass/amos2021ws06-exp-similarity-detector-frontend.git
# HTTPS
git clone https://github.com/Re-Krass/gitamos2021ws06-exp-similarity-detector-frontend.git
```

- Or download the latest release and unzip the package from:
  latest
- Or you can also download the `Amos_Project_6_Exp_Similarity_Detector.mpr` file directly from the latest release.
  1. Unpack the `Amos_Project_6_Exp_Similarity_Detector.mpk` file an empty folder.

3. Open project in Mendix

Open the `Amos_Project_6_Exp_Similarity_Detector.mpr` file in Mendix.

4. Start local development server



1. Click on the green play button
2. When the build is finished click on the *View App* button
3. Go to: http://localhost:8080/index.html

Edit this page on GitHub

Last Updated: 2/9/2022, 10:14:31 PM

← Showcase

Software Architecture →

Deployment

Content

To deploy the infrastructure and code follow the following steps:

For the initial setup use Steps 0. (Clone Git Repository) - 7. (Set Up Environment Variables On GitHub Optional). These Steps must be executed only once at the beginning.

If it has already been deployed initially, proceed to step 6. Use `act` To Build And Deploy The Container - Part 2 (Deploy and restart).

0. Clone Git Repository

```sh
# SSH
git clone git@github.com:Re-Krass/amos2021ws06-exp-similarity-detector-frontend.git
# HTTPS
git clone git clone https://github.com/Re-Krass/gitamos2021ws06-exp-similarity-detector.git
```

- Or download the latest release and unzip the package from:
  latest
- Or you can also download the `Amos_Project_6_Exp_Similarity_Detector.mpk` file directly from the latest release.
  - Unpack the `Amos_Project_6_Exp_Similarity_Detector.mpk` file an empty folder.

1. Setup Infrastructure On Azure - Part 1 (Azure Container Registry)

1. Set up Azure Container Registry
   1. Set up an Azure Container Registry. For example the name is `testregistry`.
   2. Set up the desired `Resource Group` (e.g. `amos-ws-21-22`).
   3. Get the login data in your created registry under: `Settings` > `Access keys`.
   4. Enable `Admin Username`.
   5. Save the credentials for `Registry name` (e.g. `testregistry`), `Login server` (e.g. `testregistry.azurecr.io`), `Username` (e.g. `testregistry`) and `Password` (e.g. `password1234`).
   6. The username and password is to login into the ACR.
   7. Settings > Access key.

2. Set Environment Variables - Part 1

1. Copy the file `template.secrets` to a file named `.secrets` and replace `"<Please set this variable>"` with the right variables. See table below.

`ACR_URL`	Name of Azure Container Registry	`Login server`	`testregistry.azurecr.io`
`DOCKER_USERNAME`	Username for Azure Container Registry	`Username`	`testregistry`
`ACR_PW`	Password for Azure Container Registry	`Password`	`password1234`

3. Use `act` To Build And Deploy The Container - Part 1 (Initially)

0. Install act.
1. Execute command (from the `amos2021ws06-exp-similarity-detector-frontend` directory):
   `act --secret-file .secrets -j build_and_deploy_initial`

4. Setup Infrastructure On Azure - Part 2 (Azure App Service / Database)

1. Set up Azure App Service
   1. Basics
      1. Set up an Azure App Service.
      2. Select the desired `Resource Group` (e.g. `amos-ws-21-22`).
      3. Set the `Name` of the instance (e.g. `similarity-detector`).
      4. Select under `Publish` `Docker Container`.
      5. Use `Linux` as `Operating System`.
      6. Set up the other required settings.
      7. Click on the `Next: Docker >` button.
   2. Docker
      1. Select as `Image Source` `Azure Container Registry`.
      2. Under `Registry` select the previously created registry (e.g. `testregistry`).
      3. Select the right `Image` (e.g. `similaritydetectordocker/mendixdocker`).
      4. Select the right `Tag` (e.g. `dev`).
2. Set up database
   1. Set up database named `mendix` (for example on Azure)
3. Set up environment variables on Azure
   1. Go to the previously created App Service (`similarity-detector`).
   2. Go to `Settings` > `Configuration` > `Application Settings`.
   3. Add the following environment variables (For Mendix login):

Variable name	Description	Example
`ADMIN_PASSWORD`	Password for the Mendix website login	
`DATABASE_ENDPOINT`	Connection string for database	`postgres://...`
`WEBSITES_PORT`	Port for Mendix frontend	`8080`

5. Set Environment Variables - Part 2

Go to the `.secrets` and replace the missing variables values `"<Please set this variable>"` with the right variables values.

Variable name	Description	Example
`AZURE_CREDENTIALS`	Credentials for Azure	`{"clientId": "...", "clientSecret: "..." ...}`

`AZURE_APP_SERVICE_NAME`	Name of the Azure WebApp	`similarity-detector`
`AZURE_APP_SERVICE_RESOURCE_GROUP`	Name of the Azure Resource Group	`amos-ws-21-22`

To get the `AZURE_CREDENTIALS` fill out the variables in the script `create_credentials.sh` (`.github/workflows/create_credentials.sh`) and execute it. Copy and paste the output in the `AZURE_CREDENTIALS` secret variable on GitHub or copy it in a single line for local execution.

6. Use `act` To Build And Deploy The Container - Part 2 (Deploy and restart)

Execute command (from the `amos2021ws06-exp-similarity-detector-frontend` directory):
`act --secret-file .secrets -j build_and_deploy`

7. Set Up Environment Variables On GitHub (Optional)

Set these variables in the *Secrets* section on GitHub if you like to use GitHub Actions CI to build and deploy the website.
For descriptions see the sections above.

Variable name
`ACR_URL`
`DOCKER_USERNAME`
`ACR_PW`
`AZURE_CREDENTIALS`
`AZURE_APP_SERVICE_NAME`
`AZURE_APP_SERVICE_RESOURCE_GROUP`

Edit this page on GitHub

Last Updated: 2/9/2022, 10:14:31 PM

← Software Architecture Getting Started →

Development

Content

0. Clone Git Repository

```sh
# SSH
git clone git@github.com:Re-Krass/amos2021ws06-exp-similarity-detector.git
# HTTPS
git clone https://github.com/Re-Krass/gitamos2021ws06-exp-similarity-detector.git
```

Or download the latest release and unzip the package from:
latest

1. Install poetry

Install Poetry (Python dependency management):

- https://github.com/python-poetry/poetry

```sh
# osx / linux / bashonwindows install instructions
curl -sSL https://install.python-poetry.org | python3 -
```

```
# Windows
(Invoke-WebRequest -Uri https://install.python-poetry.org -UseBasicParsing).Content | python -
```

2. Install dependencies

Go to the folder: `Code/BackendAPI/` .

For local dependencies install all requirements and the development dependencies into the virtual environment:

```sh
poetry install
```

Links

- https://github.com/python-poetry/poetry
- https://python-poetry.org/docs/

Activate virtual environment:

```sh
poetry shell
```

4. Set Environment Variables

1. Copy the file `template.secrets` and rename it to: `.secrets`.
2. Replace the variable placeholders.

The tables below explains the most important variables:

Configuration

Database

Name	Type	Description
`DB_USERNAME`	`str`	The database username
`DB_DATABASE`	`str`	The name of the database
`DB_PORT`	`int`	The database port
`DB_DRIVER`	`str`	The database driver (Microsoft SQL Server)
`DB_HOST`	`str`	The server for the database
`DB_PASSWORD`	`str`	The password for the user of the database

OpenAPI Serving

Name	Type	Description
`BLOB_AZURE_ACCESS_KEY`	`str`	Access key for Azure Blob Storage. This is used to access the blob storage on which the OpenAPI file ist stored in
`BLOB_OPENAPI_URL`	`str`	Location for the OpenAPI Specification on a Azure Blob Storage. It is directly used within the `ReadOpenApi` Azure Function to serve the OpenAPI Specification as a `.yaml` file
`OPENAPI_URL`	`str`	Location for the OpenAPI specification provided through an endpoint. In this case its the URL which points to the `ReadOpenApi` Azure function

Machine Learning API

Name	Type	Description
`ML_ACCESS_TOKEN`	`str`	Token which is needed to access the Azure Machine Learning Api for the model
`ML_URL`	`str`	Location on which the api for the machine learning model is deployed

5. IDE (optional)

5.1 Install IDE (optional)

We recommend Visual Studio Code (VSCode). Download it from:
https://code.visualstudio.com/Download

5.2 Install extensions (optional)

- Azure Tools
- Azure Account
- Azure API Management

Edit this page on GitHub

Last Updated: 2/9/2022, 10:14:31 PM

← Software Architecture

Deployment →

Deployment

Content

To deploy the infrastructure and code follow the following steps:

For the initial setup use Terraform. Terraform creates all necessary services on Azure. The Terraform script must be executed only once at the beginning. If it has already been deployed, proceed to step 2 (2. Deploy Code To Azure).

0. Clone Git Repository

```sh
# SSH
git clone git@github.com:Re-Krass/amos2021ws06-exp-similarity-detector.git
# HTTPS
git clone https://github.com/Re-Krass/gitamos2021ws06-exp-similarity-detector.git
```

Or download the latest release and unzip the package from:
latest

1. Setup Infrastructure With Terraform

1.1 Install Terraform

Download and install Terraform from: https://learn.hashicorp.com/tutorials/terraform/install-cli

1.2 Setup Environment Variables

Go to the folder: `Code/BackendAPI/terraform/` .

Copy the file `terraform.template.tfvars` and rename it to: `terraform.tfvars` .
Replace the variable placeholders. A description of these is given by the following table:

Variable name	Description
`collectionname`	Identifier for every Azure resource
`adminemail`	E-Mail to receive notifications

1.3 Initialize Terraform Project

```
terraform -chdir="terraform" init
```

1.4 Deploy Azure infrastructure

2. Deploy Code To Azure

2.1 Login To The Azure CLI

```sh
az login
```

2.2 Deploy Function App to Azure

Go to the folder: `Code/BackendAPI/terraform/` .

```sh
func azure functionapp publish <function-app-name>

# Example
func azure functionapp publish fa-similarity-detector-func
```

2.3 Set Environment Variables

Copy the file `template.secrets` to a file named `.secrets` and replace `"<Please set this variable>"` with the right variables.

Azure

1. Go to Azure Portal.
2. Go to `Home > Function App > <Function App Name>` .
3. Go to the `Settings` column and `Configuration` .
4. Click on `Advanced edit` and paste the output from the following command under the existing variables (or replace the old ones.)

   ```sh
   python ./scripts/gh_secret_env_vars.py --format_azure
   ```

5. Click on `OK` and `Save` (Don't forget to save!).

Edit this page on GitHub

Last Updated: 2/9/2022, 10:14:31 PM

← Development CI / CD →