

Clone

Normal HTTPS Link:

- `git clone https://github.com/amosproj/amos2024ss06-health-ai-framework.git`

SSH Link:

- `git clone git@github.com:amosproj/amos2024ss06-health-ai-framework.git`

Build

Because we use python, there is no building. We use the package and dependency manager PDM. (Overlay <https://pdm-project.org/en/latest/>)

To install PDM, which is a dependency manager for project, you can use the following command:

- **prerequisite:** python 3.10
- `pip install pdm`

To install all dependencies run

- `pdm install`

pyproject.toml

All project dependencies are stored under [dependencies]

To run any script from the root we have structured the code into packages for which we can predefined commands to run a certain package. These are specified under [tool.pdm.scripts]

Deploy

Use the following command to build the Singularity container:

- `singularity build Apptainer.sif Apptainer.def`

You can opens a shell within the container, allowing you to interact with its environment:

- `singularity shell Apptainer.sif`

Test

We currently have no automated component tests.

We use the tool ruff for

code linting (checking the code against rules to ensure coding standards)

formatting (applying certain rules automatically)

- `pdm run lint`
- `pdm run format`

these 2 commands are automatically executed on commit (locally)

and also (remotely) via a github action when somebody does a pull request

so that code that doesn't fulfill code standards is not pushed to main

A GitHub Action is a feature provided by GitHub that allows you to automate workflows directly within your GitHub repository. These workflows are defined by YAML configuration files and can be triggered by various events such as pushing code, creating pull requests, or on a schedule.

Example under `.github/workflows` we find

- `check_codestyle.yml`

It does the automatic linting and formatting **on pull request** and executes the **jobs** which will in the end run *ruff check* and *ruff format*

Example2 also in `.github`

- `release_drafter.yml`

release draft template for automatic release notes listing for all the issues solved (show on github). These are created from an external github app that takes the template for it.

Branch for showing automatic linting and formatting

- `build_video`

Run

So far we have successfully developed data acquisition pipeline which can be started by:

- `pdm build-config`
- `pdm run-orchestrator`