# The Pipeline in XNAT

**Pipeline is a method of implementation for automated data processing** in XNAT, using XML descriptors and server-side scripts/tools.

In XNAT, a **pipeline** is used to automate data processing workflows on imaging and related data stored in the system. Pipelines allow researchers and developers to define and execute repeatable, structured sequences of tasks such as data conversion, quality control, analysis, and post-processing directly on XNAT-managed data, without manual intervention.

Typical uses of pipelines in XNAT include:

- **Image format conversion**
- **Running analysis tools**
- **Automated quality control**
- **Extracting features**
- **Machine learning inference**
- **Semi-automated workflows**

Overall, pipelines are used in XNAT to ensure **reproducibility**, **standardization**, and **automation** of data analysis tasks, especially in multi-subject or longitudinal imaging studies.

-To run the Pipeline in XNAT, the first thing we need to do is to install the **Pipeline Engine**. By default, the Pipeline engine is located in a directory next to XNAT_HOME: **/data/xnat/pipeline**, which is called the Pipeline home.

Pipeline engine can be located anywhere on the file system, as long as the PIPELINE_HOME folder has appropriate permissions for the user who launches tomcat and the non-default path to the PIPELINE_HOME should be set using Administer -> More Options -> Default Settings.

**How to connect XNAT with the Pipeline:**
Here:
https://wiki.xnat.org/documentation/xnat-pipeline-development-schema#:~:text=By%20default%2C%20this%20location%20is,are%20mentioned%20in%20this%20documentation.

It written that “By default, this location is /data/xnat/pipeline/, and the default pipelines directory is /data/xnat/pipeline/pipelines/.”
which means that XNAT is expecting the Pipeline under
/data/xnat/pipeline/pipelines/

The XML file:
- we define the name and the description of the Pipeline.

-The command <Command>

- where in XNAT we want to run the Pipeline

**Then The Pipeline can be registered in XNAT:**

To register the Pipeline the Documentation suggest to use the :
./bin/register-pipeline -dir /data/xnat/pipeline/pipelines/my_pipeline/

**How XNAT recognize the Pipeline:**

https://wiki.xnat.org/documentation/installing-pipelines-in-xnat

First of all, we need to pack the pipeline separately in a Pipeline file:

```
mkdir -p /data/xnat/pipeline/pipelines/my_pipeline
```

the copy the there:

```
cp my_pipeline.xml /data/xnat/pipeline/pipelines/my_pipeline/
cp run.sh          /data/xnat/pipeline/pipelines/my_pipeline/
chmod +x run.sh     # ausführbar machen
```

*Explanation:*
The commands shown are part of the process of setting up a classic XML-based pipeline in XNAT.

The first command copies the file my_pipeline.xml, which contains the XML-based definition of the pipeline workflow, into the directory /data/xnat/pipeline/pipelines/my_pipeline/, where XNAT expects all pipeline descriptors to be stored.

The second command copies the file run.sh—a shell script that performs the actual data processing defined in the pipeline into the same directory. This ensures that both the pipeline definition and the script it executes are located together in the correct structure required by the XNAT Pipeline Engine. Finally, the third command makes the run.sh script executable using chmod +x, which is essential because XNAT will later attempt to run this script as a command. Without executable permissions, XNAT would not be able to start the script, and the pipeline execution would fail.

The fact that these pipelines require direct access to the server via SSH in order to manually copy the XML descriptor and associated scripts to a specific directory (/data/xnat/pipeline/pipelines/). This approach is not feasible in secure or restricted settings where developers or researchers do not have shell access to the production server.

**After that we can add the Pipeline in XNAT:**
under Admin>Pipelines>Register new Pipelines> **And then, in the appropriate field, we enter:**

```
/data/xnat/pipeline/pipelines/my_pipeline/my_pipeline.xml
```

Then we have to enable the Pipeline on the web site and on the Project.

**Can we generalize or automate pipelines in XNAT using a Python script?**

After reviewing the available Pipeline REST API endpoints, we found that XNAT supports multiple operations via REST. The most relevant endpoint for automation is

: POST /pipelines/launch/{pipelineName}

This allows us to programmatically launch an already registered pipeline in XNAT. However, for full automation, additional endpoints would be needed, such as one to enable a pipeline globally in XNAT and another to assign or activate a pipeline for a specific project. These functionalities are currently **not exposed via the REST API**, which means that while the **execution** of pipelines can be automated, the **full deployment and project-level configuration** still require manual steps.

In conclusion, a **fully automated pipeline workflow using only a Python script is not currently possible**. Only the **launching** of existing, pre-registered pipelines can be automated via REST.