

Extensions

Code

The code for the extensions can be found on [GitHub](#). (module `streampipes-extensions`)

Prerequisites

Install:

- Java
- Maven
- Docker + Docker-Compose
- Here we describe the setup in IntelliJ (we do provide run configuration files to ease the configuration setup), but you can use an IDE of your choice

Development Environment

In this setup we will start extensions such as pipeline elements in your IDE. The backend, UI and all the other services run in docker.

Set-Up StreamPipes Services

1. Clone Project [\[Link\]](#)

Prepare Installer

```
# SSH
git clone git@github.com:apache/incubator-streampipes.git

# HTTPS
git clone https://github.com/apache/incubator-streampipes.git

# switch to development branch
git checkout dev
```

2. Open CLI directory

Prepare Installer

```
cd incubator-streampipes/installer/cli/
```

3. Configure StreamPipes for backend development

Prepare Installer

```
./streampipes env --set pipeline-element
# Updates the file 'pipeline-element'. All StreamPipes services required for development are selected
```

4. Start third party services (e.g. CouchDB, Kafka, ... in docker)

Prepare Installer

```
./streampipes up -d
```

5. Open the project in your IDE

Set-Up for Extensions (Pipeline Elements, Connect Adapter)

We provide an environment file for each service in the folder `development`. This contains the environment variables that must be set to run the service.

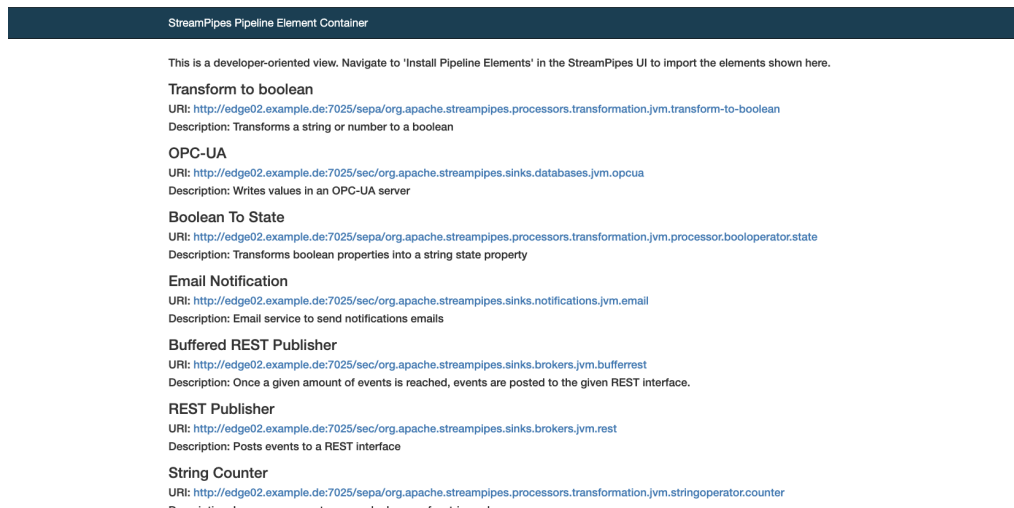
For IntelliJ users: we ship run configurations with the repo located at `.idea/runConfigurations`

For other IDE users: You need to manually set environment variables for the corresponding service according to the specification found in the run configurations for IntelliJ

Example walkthrough for pipeline-elements-jvm-all in IntelliJ:

This is a service that wraps all pipeline elements written with the standard Java wrapper, hence the the notion of "jvm-all".

1. You should automatically have a run configuration "all-pipeline-elements-jvm" as we ship in with the repository
2. (Optional) Modify environment variables depending on your OS (this is due to the dev setup of partly IDE and partly Docker, we need the local services to be reachable from within the Docker network)
 - a. **For Windows/Mac user:** you are good to go
 - b. **For Linux user:** You must change the environment variables stating **host.docker.internal** to you local machine's IP
3. Run the service
4. Navigate to "<http://localhost:7023/>" to see if the backend is running. You should see something similar to the following:



Where to go from here?

You can start writing own pipeline elements, that means adapters, processors, sinks.

For example, see the example of a Greeter processor <https://github.com/wipatrick/apachecon-demo-processor>

We also do provide maven archetypes to easily setup a Maven project skeleton for you, see <https://streampipes.apache.org/docs/docs/dev-guide-archetype/>