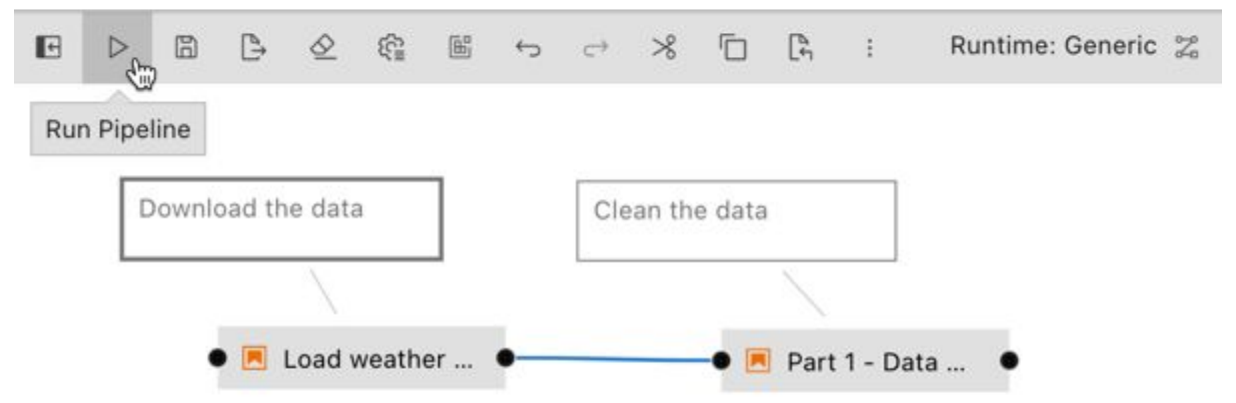
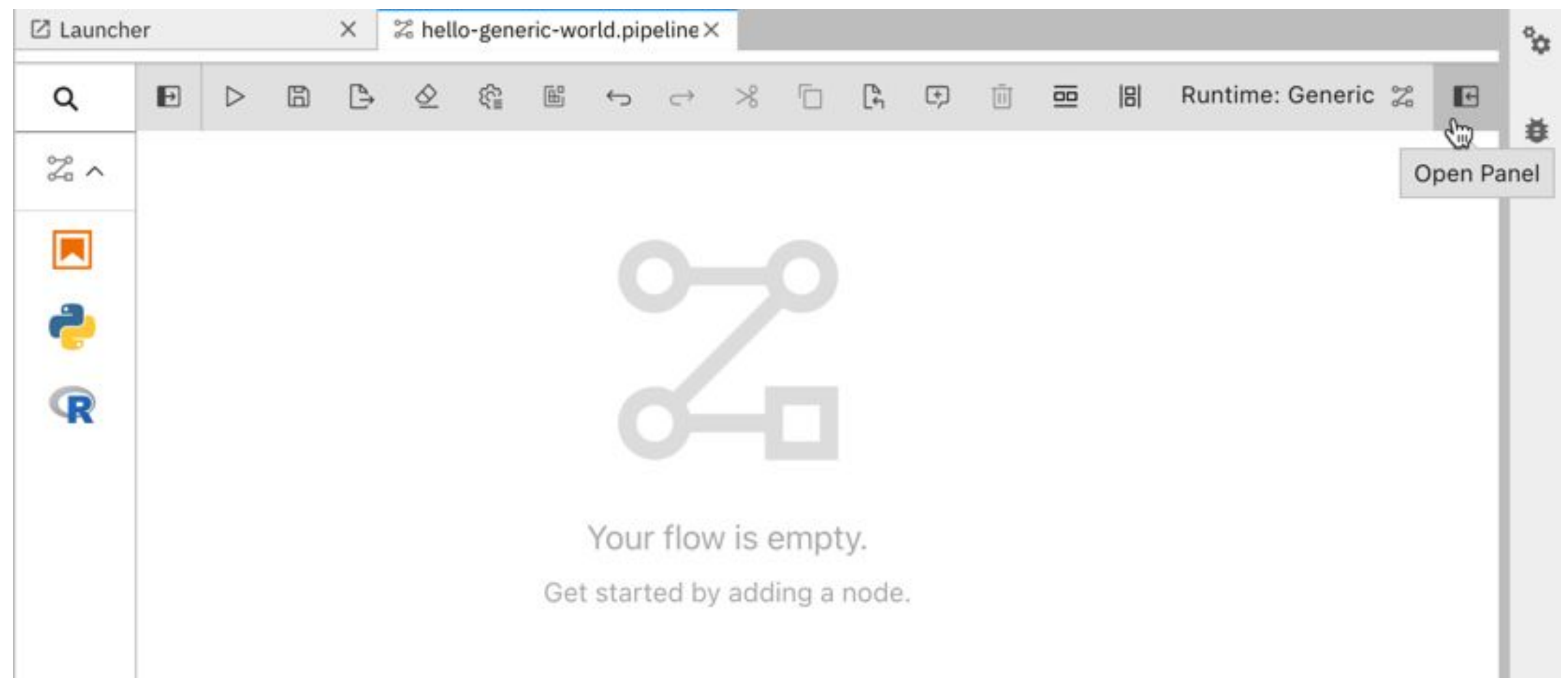
Elyra
Extends JupyterLab with an AI Centric Approach

Xin Chao Chen & Kai Chen
RCOS + IBM 2024

User Interface





Pipeline Editor

The pipeline function of Elyra empowers users to orchestrate complex workflows by connecting individual nodes, each representing a specific task or process, to define execution dependencies. These nodes are implemented through components, which can be Jupyter notebooks, Python scripts, or R scripts.

In a basic scenario, a pipeline consists of interconnected nodes, each performing a distinct function such as data loading, processing, model training, or even non-technical tasks like sending emails.

Elyra's generic pipeline functionality extends across multiple runtime environments, including local/JupyterLab, Kubeflow Pipelines, and Apache Airflow. This versatility allows users to develop pipelines that seamlessly transition between these environments.

- Prerequisites: Ensure you have JupyterLab 3.x with Elyra extension v3.x installed.
- Setup: Clone the introductory sample from the Elyra GitHub repository.
- Create a Generic Pipeline: Use the Generic Pipeline Editor in JupyterLab to create a new pipeline. Give it a descriptive name and description.
- Add Nodes: Add notebooks or scripts to the pipeline. You can mix and match different types of files based on your requirements.
- Configure Nodes: Customize each node by specifying properties such as file location, runtime image, resource requirements, file dependencies, environment variables, and output files.
- Connect Nodes: Establish dependencies between nodes by connecting output ports of one node to input ports of another.
- Run Pipeline: Execute the pipeline either locally from the pipeline editor or using the command line interface provided by Elyra.
- Monitor and Inspect Results: Keep track of the pipeline run progress and inspect the output artifacts generated.

Elyra

Elyra AI is an innovative open-source extension to JupyterLab, designed to facilitate the development and execution of AI and data science workflows. By integrating a Visual Pipeline Editor, Elyra enables users to construct complex pipelines from notebooks, Python, and R scripts with ease. The project supports batch job execution, reusable code snippets, and hybrid runtime environments through Jupyter Enterprise Gateway. With features like auto-generated outlines for notebook navigation, integrated Python script debugger, and Git version control, Elyra AI empowers developers to streamline their AI projects. Hosted under the LF AI & Data Foundation, Elyra AI is at the forefront of making AI tools more accessible and efficient for the open-source community.

Semester Goals

The results of this experiment showed that temperature and coacervates production does not correlate, and temperature change does not effect the coacervates production.

Team Formation

Major features are implemented by members of Elyra Community

Luciano Resende
Rome Kienzler

Implementation

npm
ESLint
TypeScript
jupyterlab
React JS

Join Elyra

Github

Slack

SCAN ME

Gitter

SCAN ME

Timeline

January 2024:
- Review codebase and understand underlying technology.
- Complete Project Proposal for Elyra Project.
- Determine additional tools and technologies needed for project contribution.
- Create a cookie cutter extension for Jupyter Lab for understanding extension creation.

February 2024:
- Start migration of Elyra from JupyterLab 3.x to 4.x.
- Refactor Elyra packages for compatibility.
- Resolve build errors, type errors, and import issues.
- Begin work on UI components.
- Fix issues related to missing properties and import errors.
- Solve migration errors due to breaking changes.
- Update Python editor to address type errors.

March 2024:
- Resolve build errors for UI components.
- Update metadata handling to use new interfaces for JupyterLab 4.x.
- Refactor scala-editor and r-editor to accommodate changes in CodeEditor.
- Resolve type errors in imported components.
- Update build errors for code-snippet and pipeline-editor.

April 2024:
- Continue updating Elyra for compatibility with JupyterLab 4.x.
- Work on building Elyra and resolve integration issues with different components.
- Remove code viewer from Elyra and utilize JupyterLab's code viewers.
- Update Elyra to use elyra-canvas 12.40.1.

Others

Conclusions