University of Minho
School of Engineering

ISLab

Machine Learning with Knime

Similarity Based Systems

Perfil ML:FA@MiEI/4º ano - 1º Semestre
@MES/2º ano - 1º Semestre
Bruno Fernandes, Paulo Novais

17/10/2019

Contents

- Flow Variables
- Metanode Templates and Components
- Data Visualization
  - JavaScript (and Java) integration
  - Composite Views
- PMML
- Hands On

Flow Variables

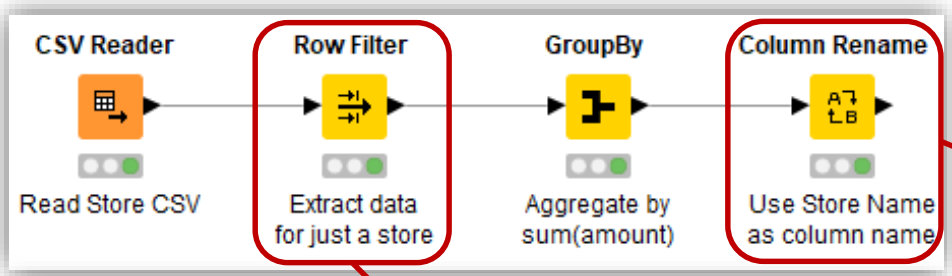
Flow Variables are variables used inside a workflow to dynamically overwrite node settings, i.e., to parameterize workflows when node settings need to be determined dynamically

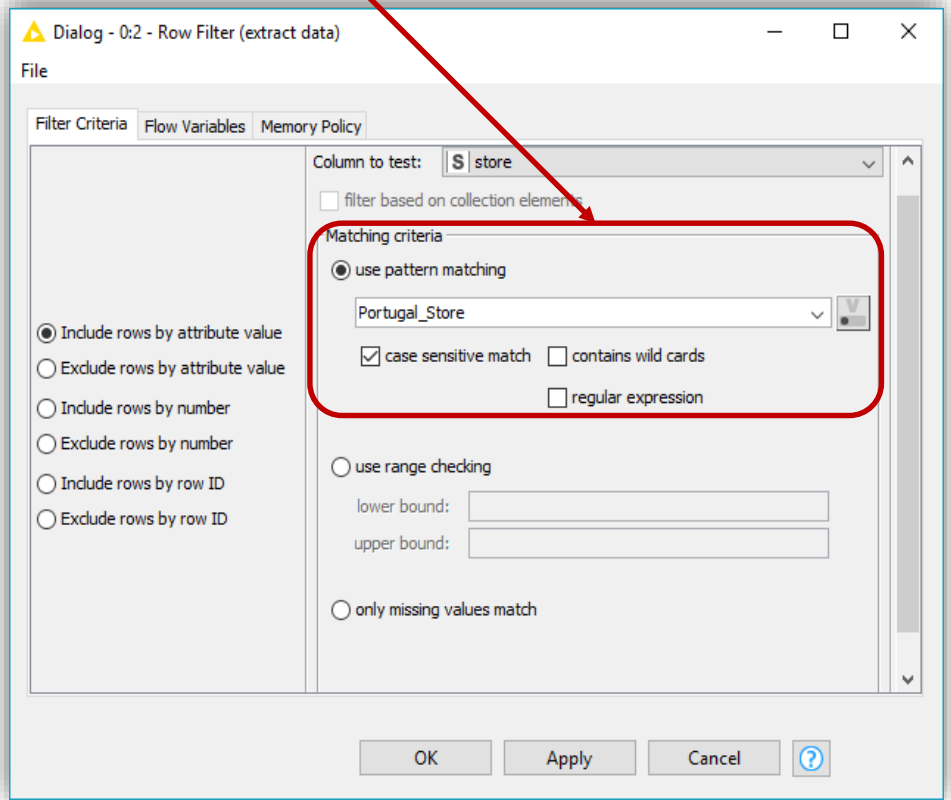
Two of the most important applications for flow variables are the configuration of Loops and Metanodes

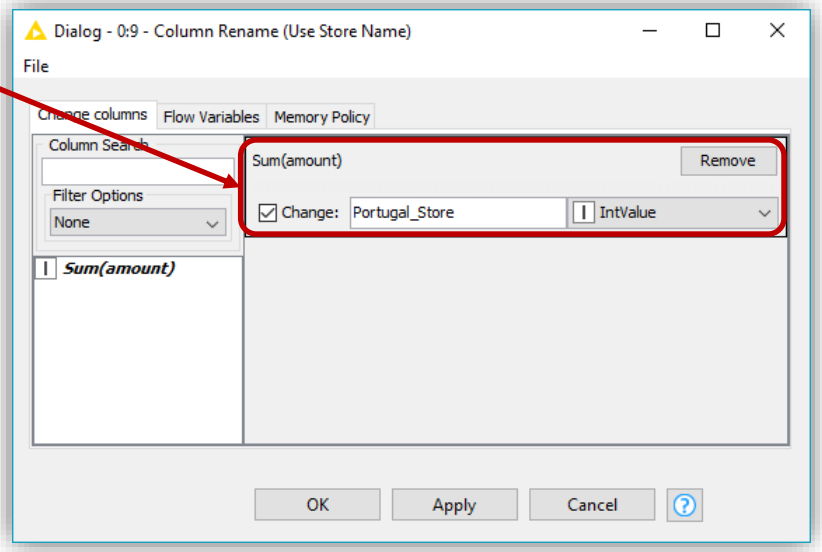
Two ways of creating workflow variables:

- At Workflow Level (visible to all nodes)
- or Inside the Workflow (visible to the nodes that appear after the variable creation)

Without Flow Variables

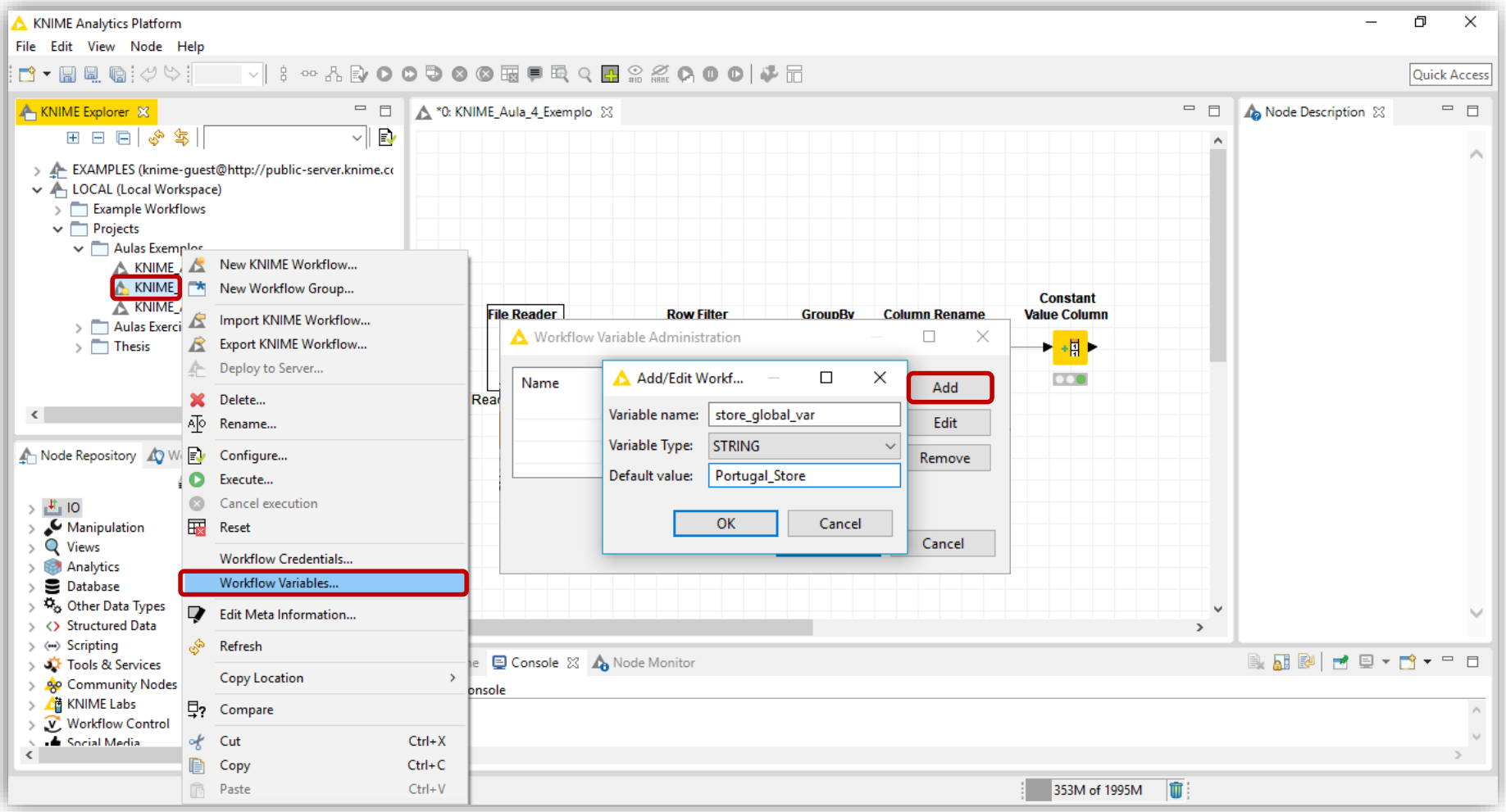




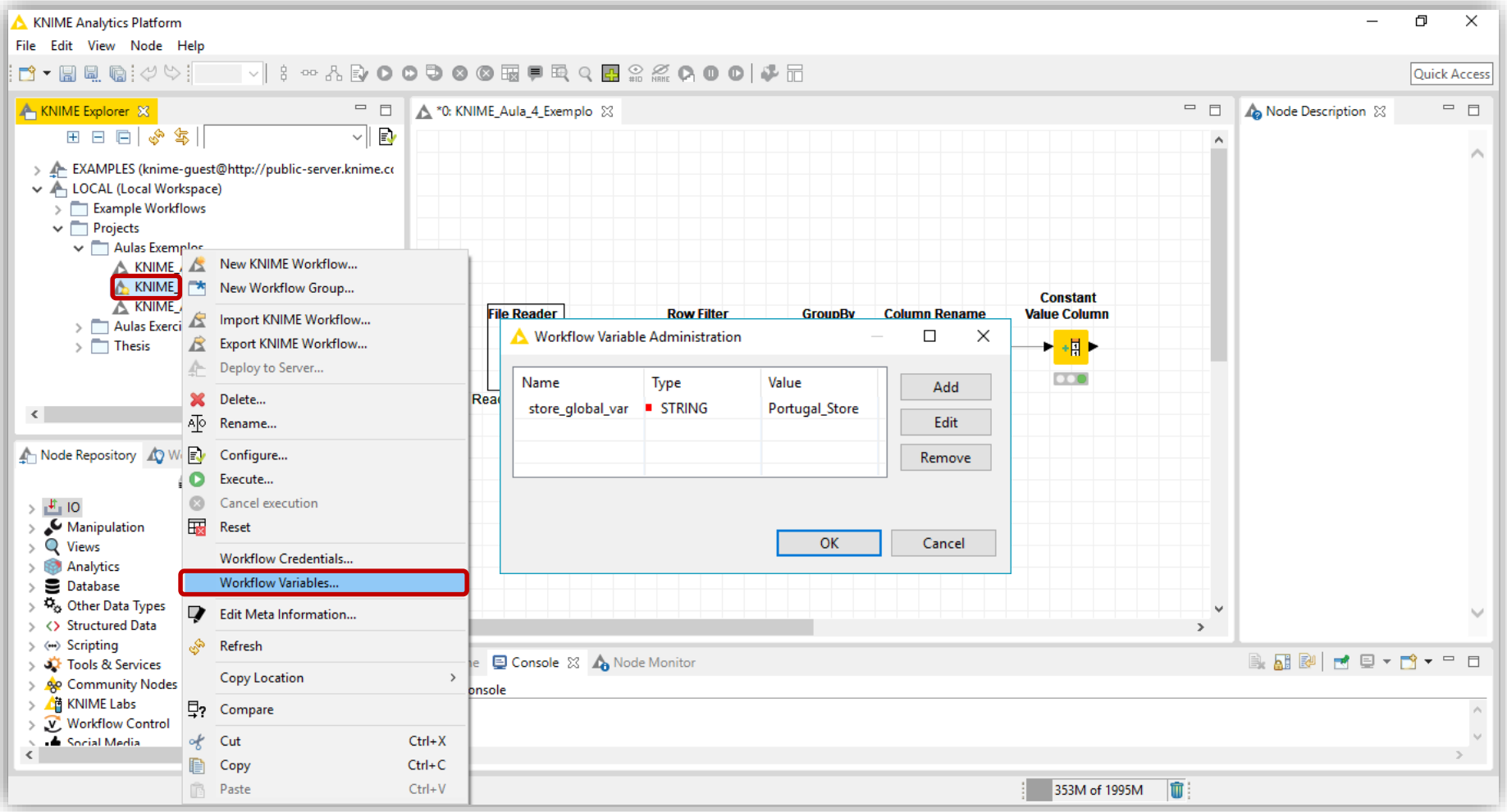


And this is a simple flow… A bigger one with more nodes would imply a lot of work just to rename all parameters!

Flow Variables at Workflow Level

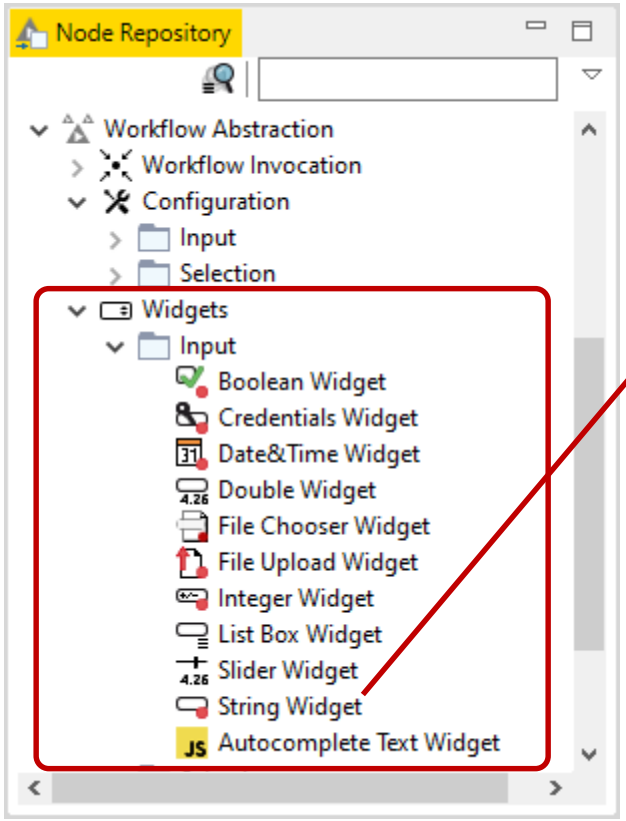


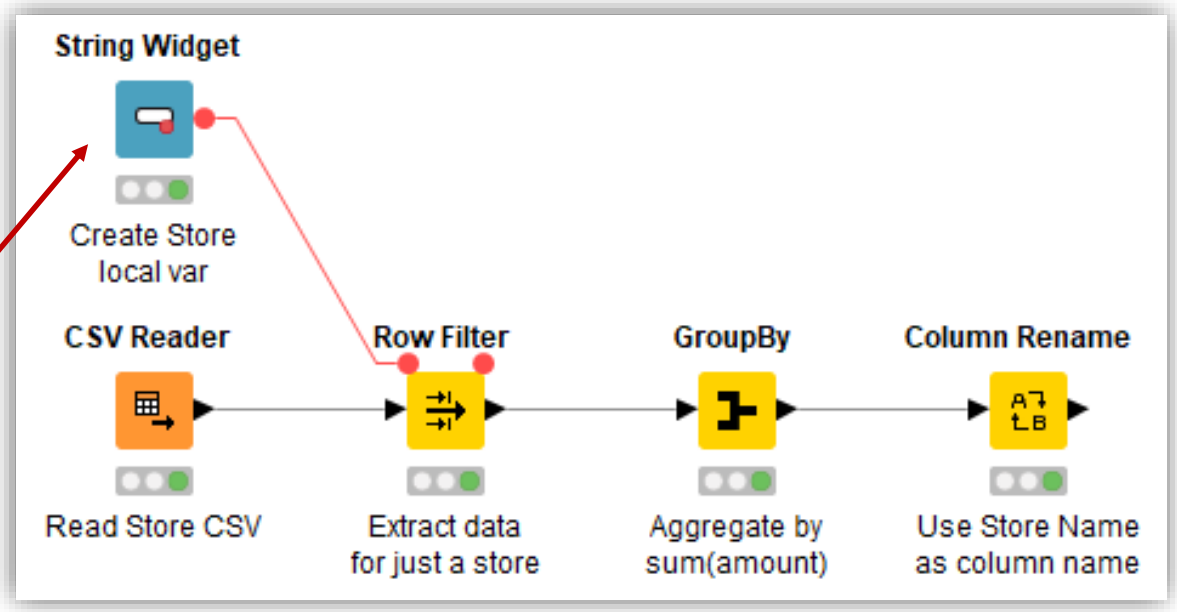
Flow Variables at Workflow Level



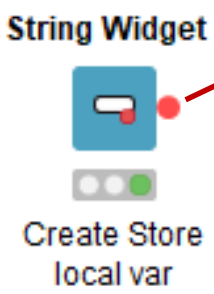
Flow Variables Inside the Workflow

To create a flow variable inside the workflow we should use Widgets!





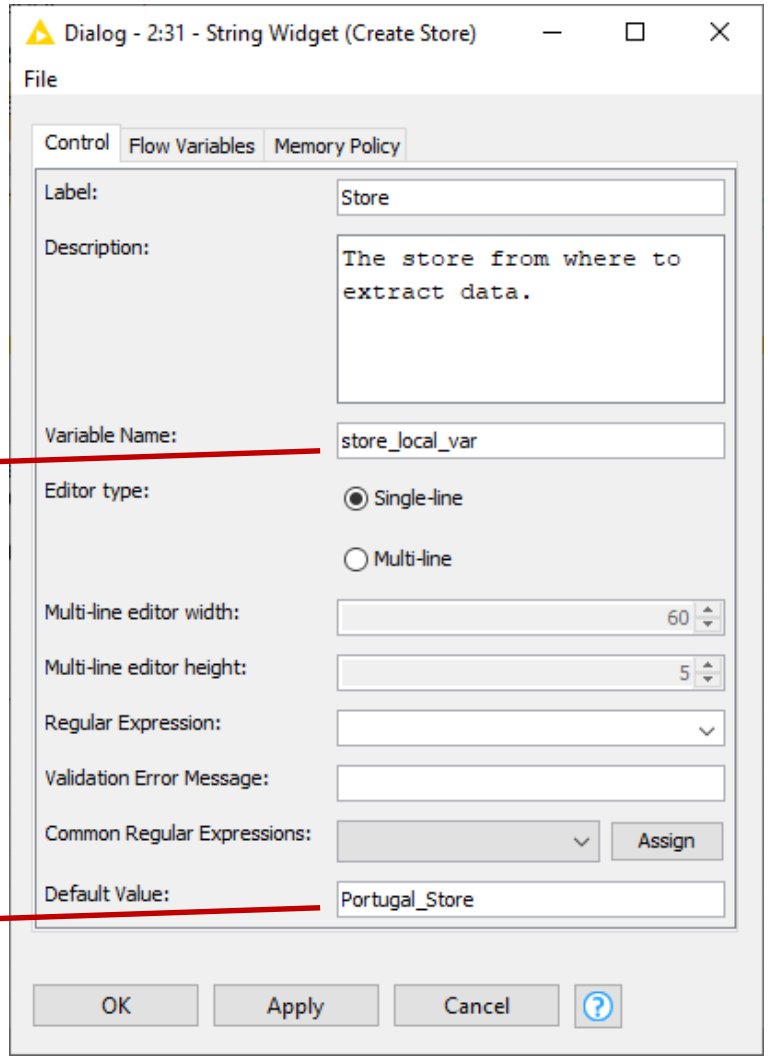
Flow Variables Inside the Workflow



Indicates a Flow Variable!

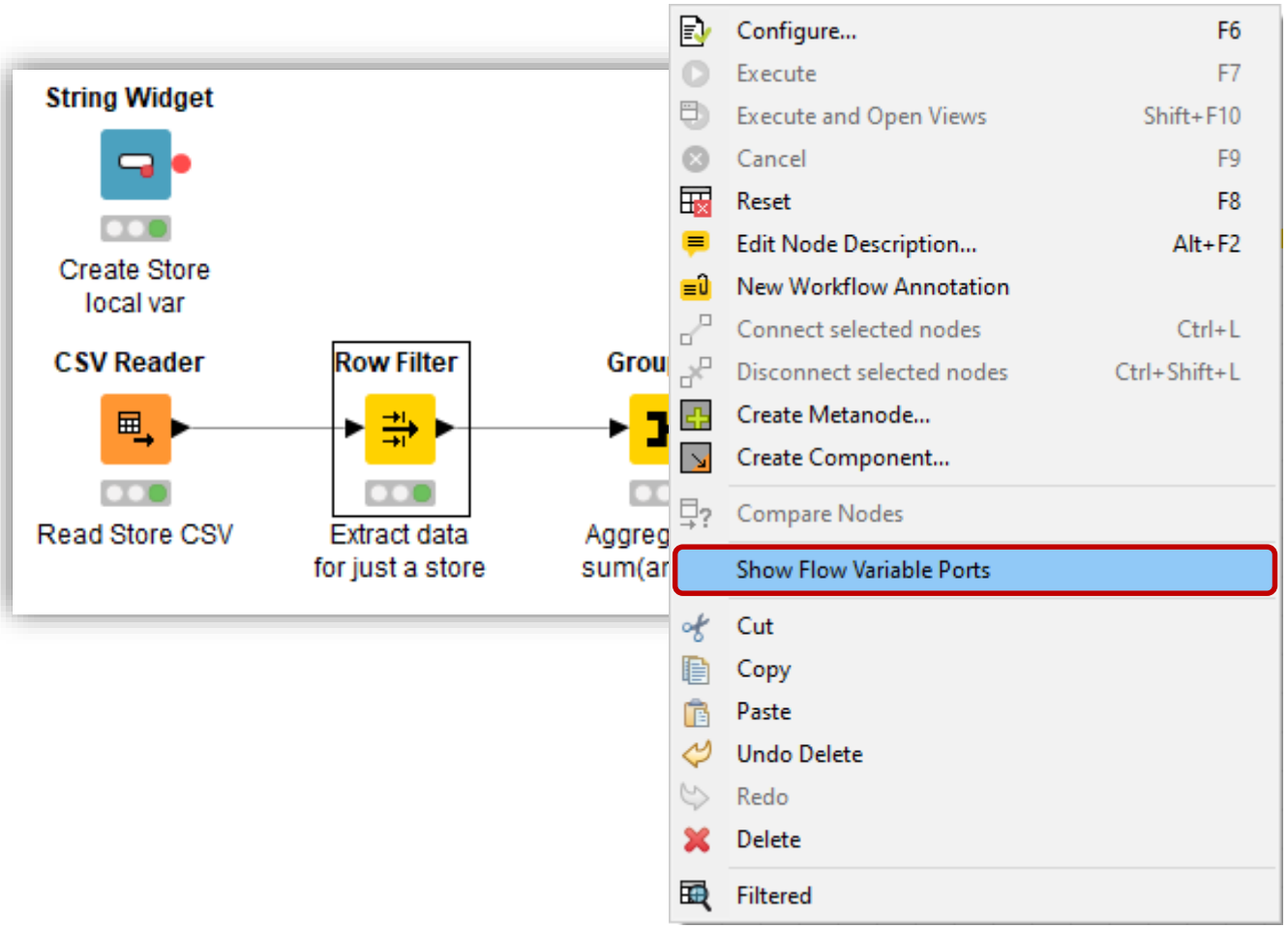
Defines the variable's name

Defines the default value



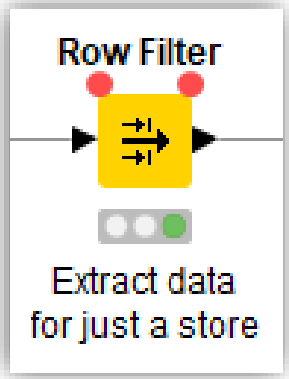
Flow Variables Inside the Workflow

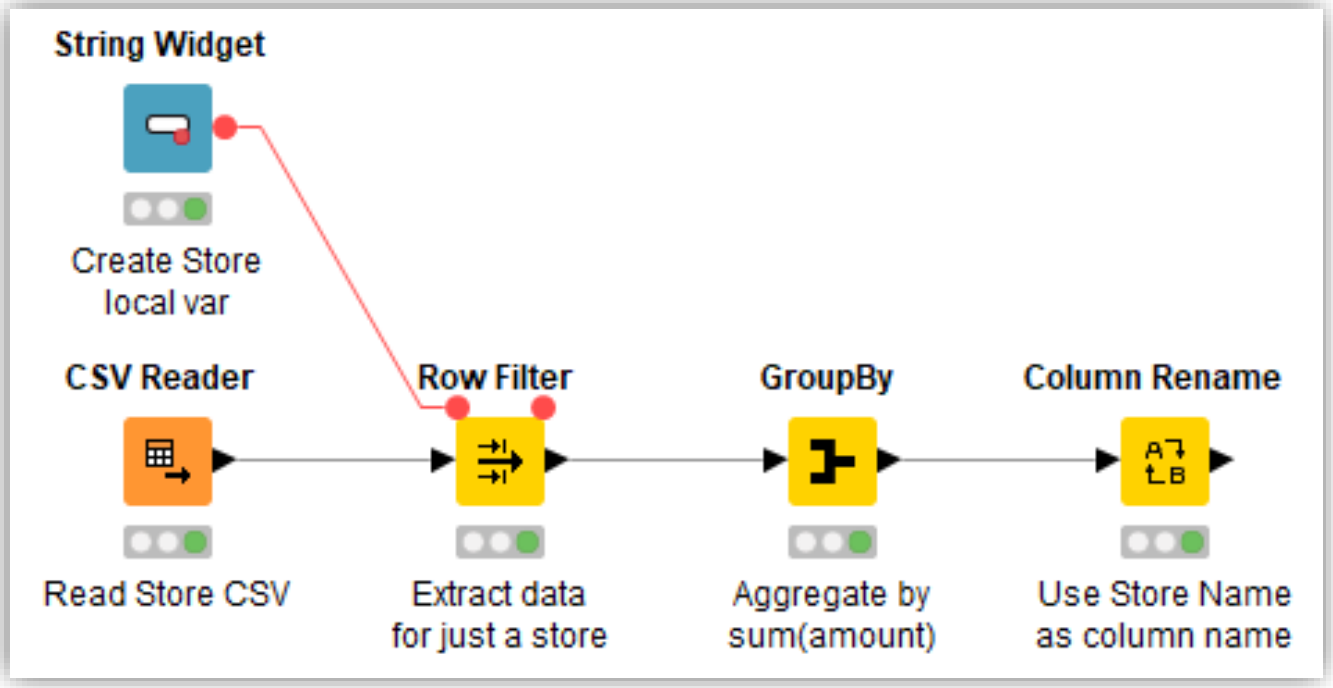
But… How to connect the flow variables output of the widget to a node's input?



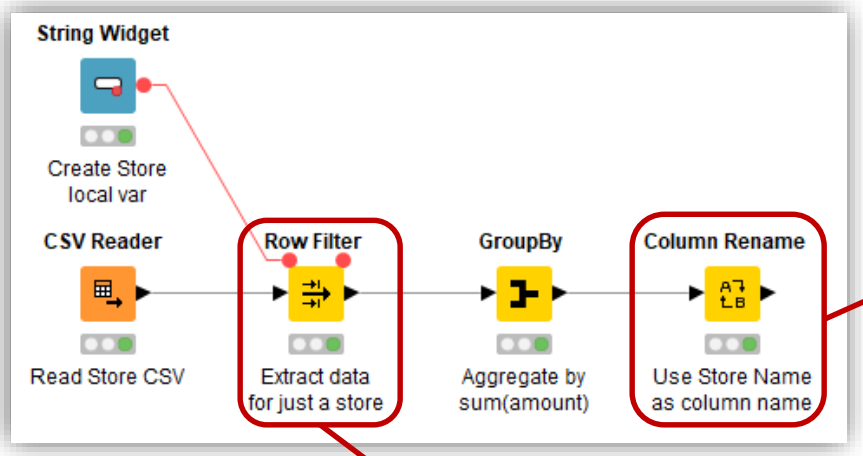
Flow Variables Inside the Workflow

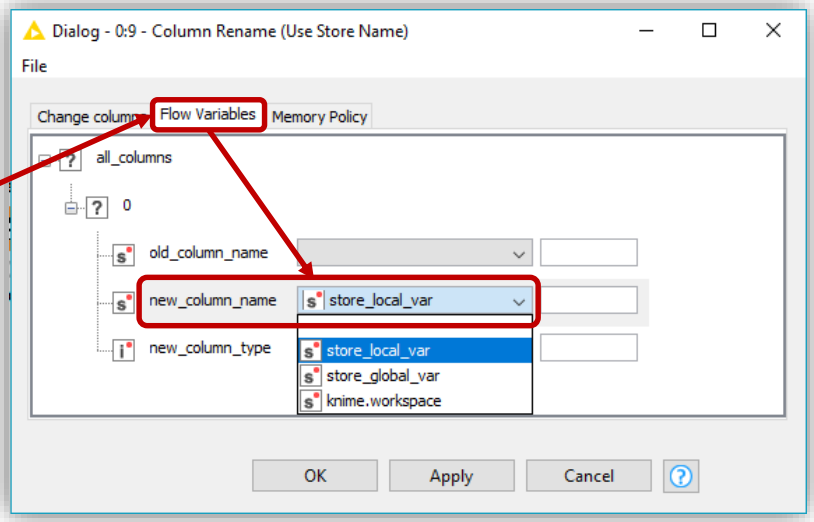
The variables' input and output of the selected node appears! Then, just connect.

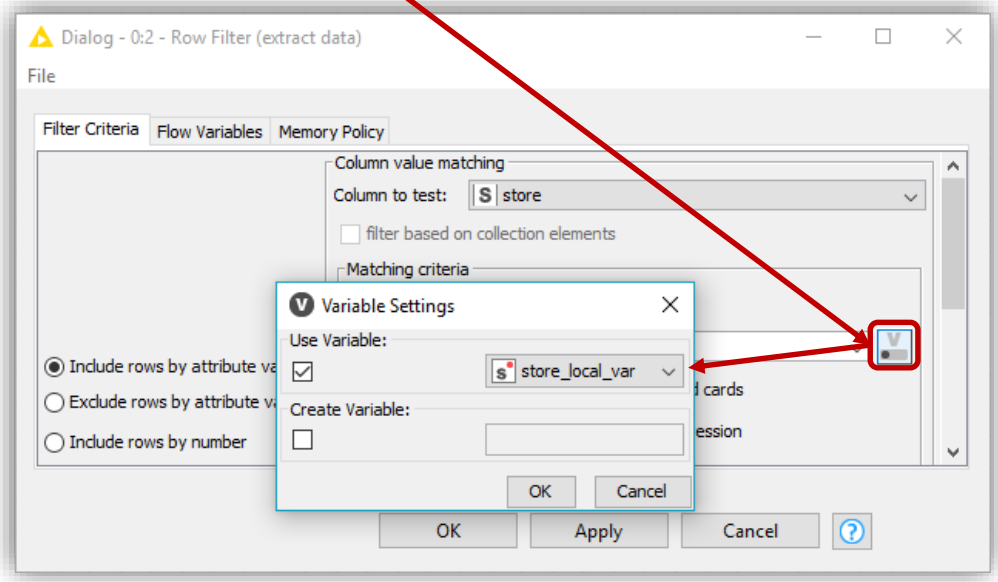




How to use the created Flow Variables

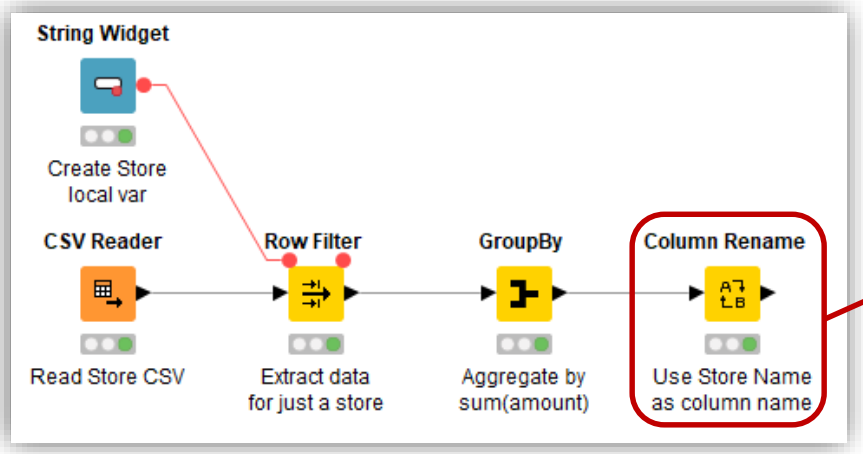


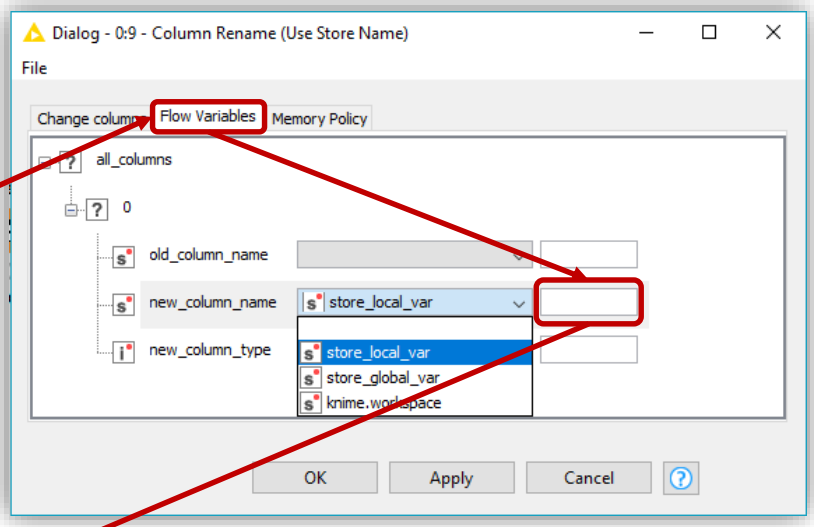




Everything is now parameterizable using Flow Variables! We can now execute the flow and gather results for the variable's value!

Create More Flow Variables

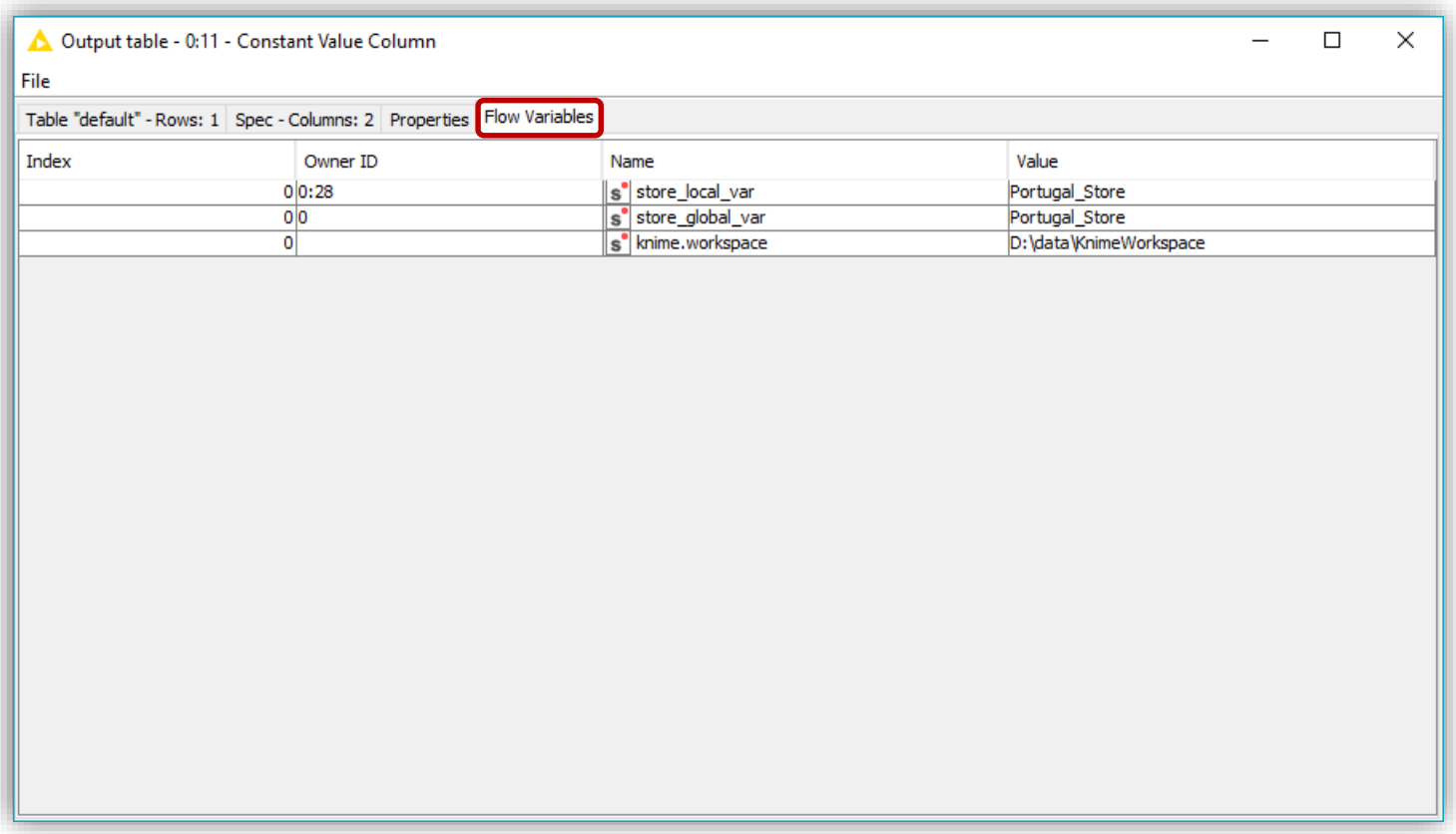




We can also create new Flow Variables with the value of a specific node setting. For that just define the name of the new flow variable in the input text near to the desired setting!

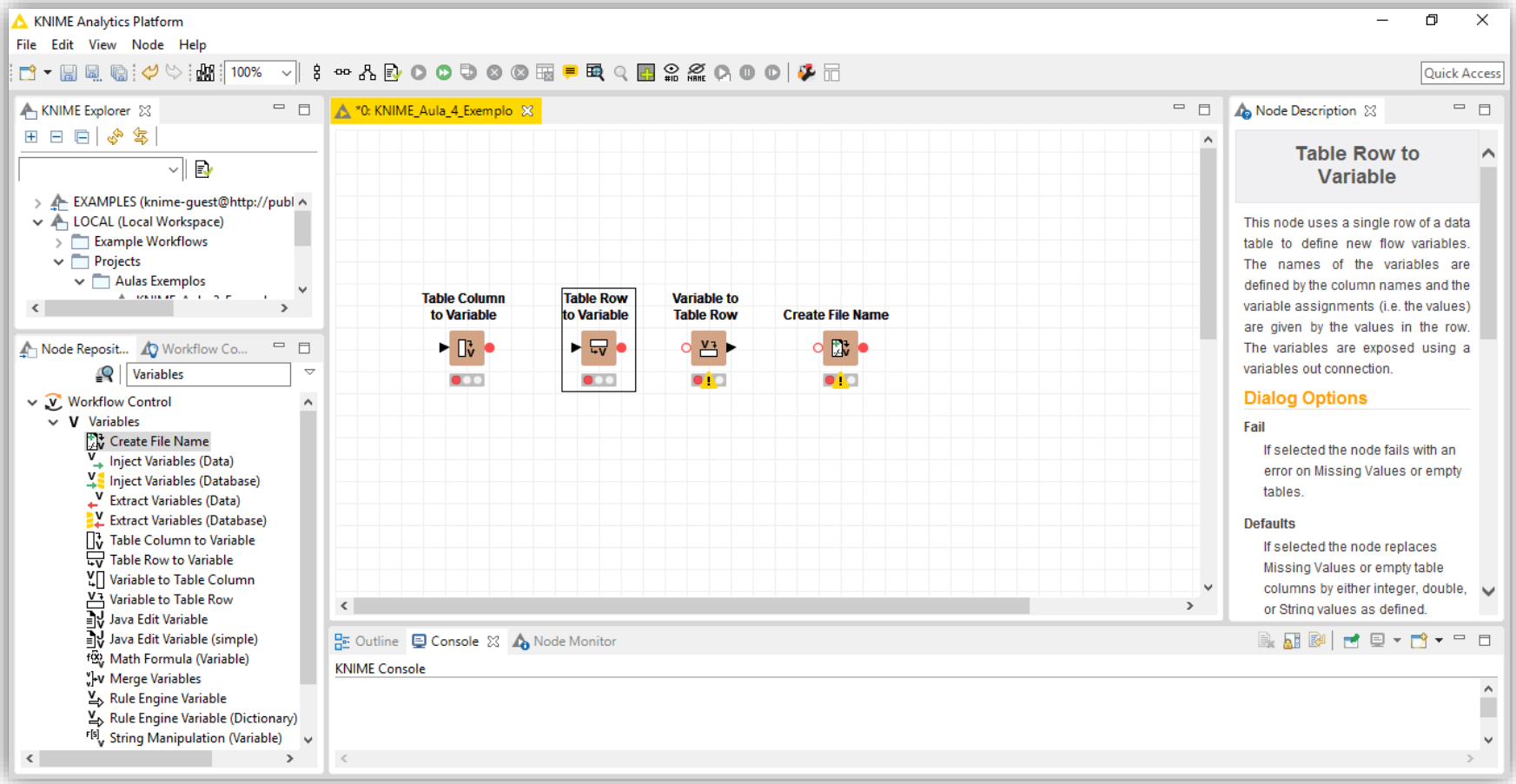
Flow Variables available to a Node

Last but not the least, we can check which flow variables a node sees (and may use) in the table view of the node!



Data Cells into Variables

These nodes allows us to convert values from a table into flow variables with the row IDs or column names as variable names.



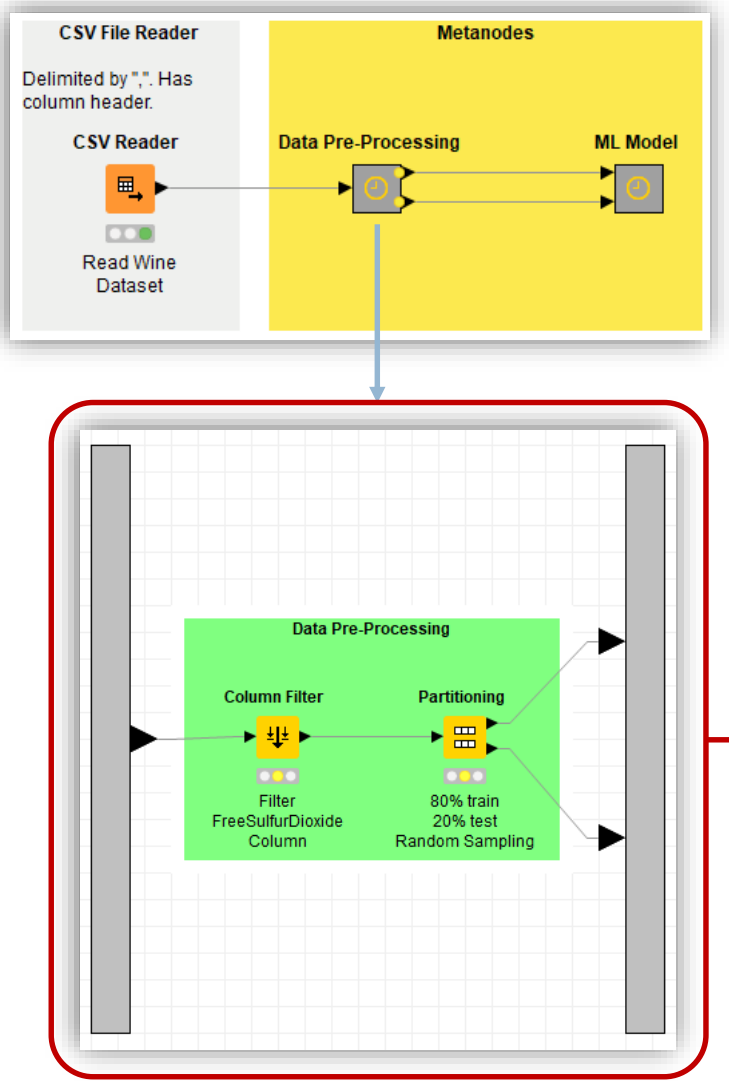
Metanode Templates

We have seen that a Metanode is a node with other nodes inside! We use them for a Tidy Workflow!

However…

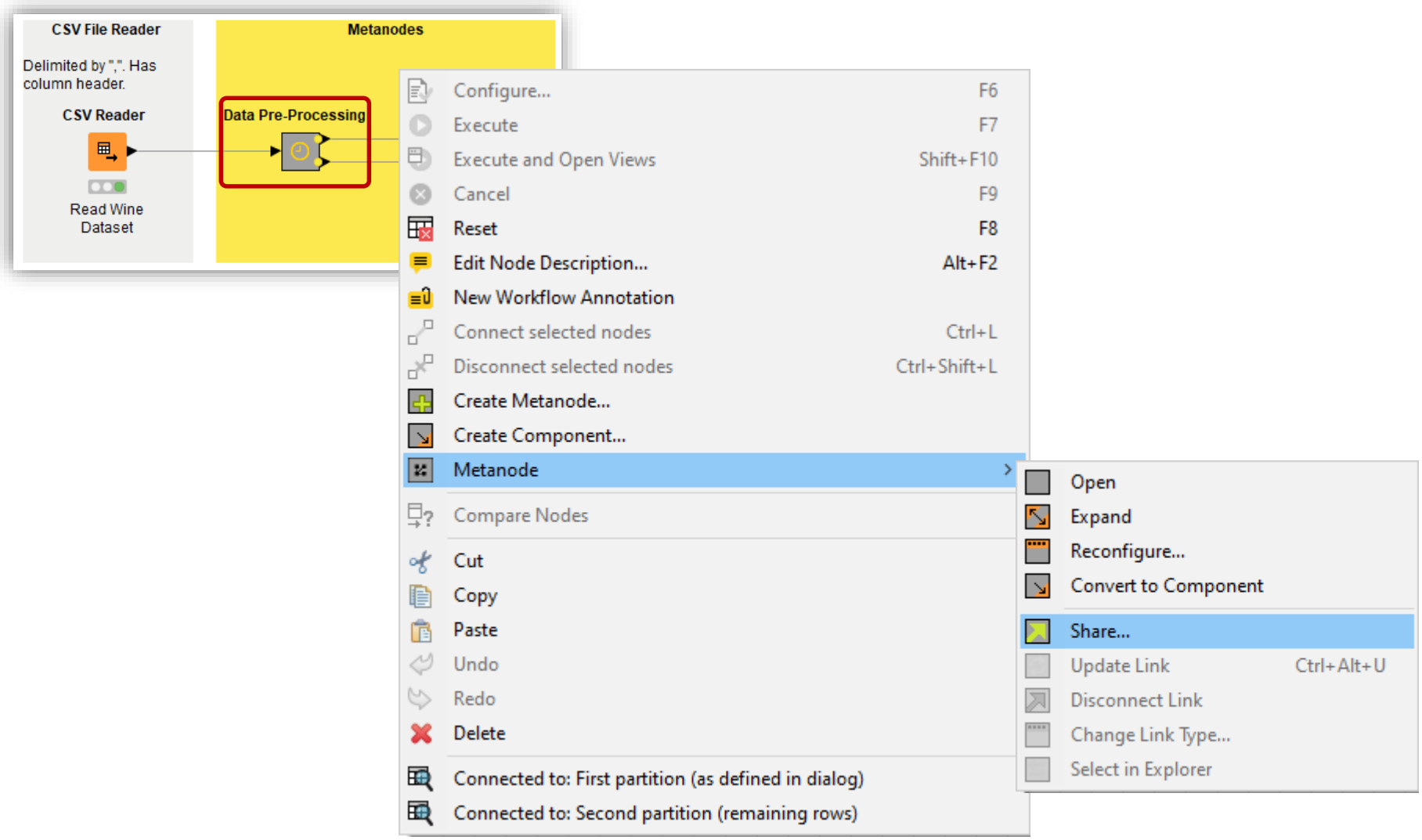
We can, and should (!), have metanodes promoting reusability!! If a metanode holds a self-contained logical block of nodes, it can be reused for the same task in other workflows by defining a Metanode Template.

Metanode Templates

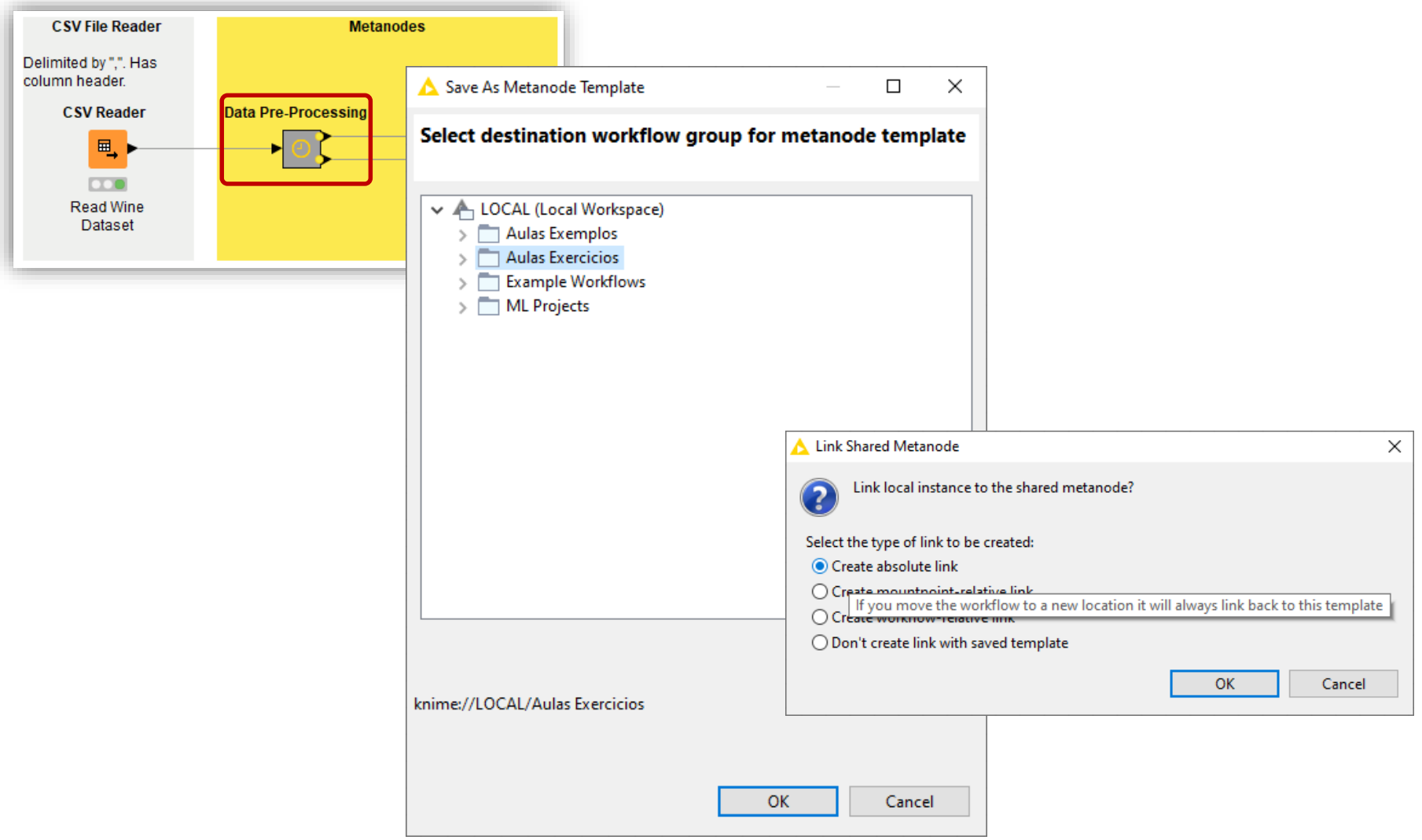


We may want to have this logic replicated in more workflows. Therefore, we should create a Metanode Template!

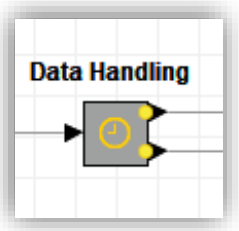
Metanode Templates



Metanode Templates

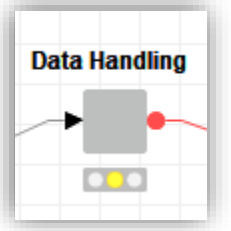


Components (~~Wrapped Metanodes~~)



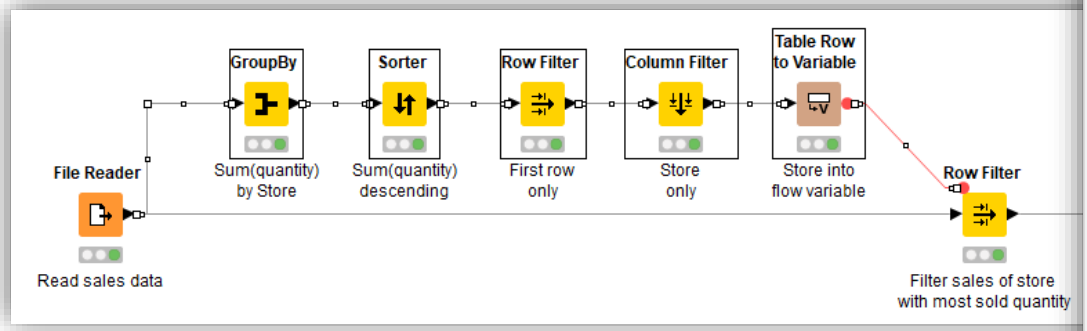
Components are new to Knime and were added in Knime's last major release (*4.0.0, June 2*019). They were introduced to enhance and replace old Wrapped Metanodes, which have evolved from simple metanodes! Components can:

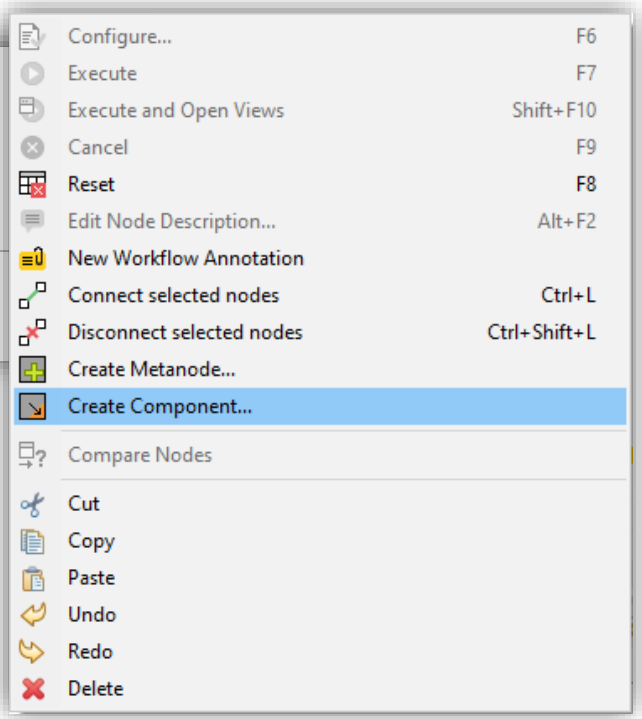
- Acquire a configuration window via Widget nodes
- Have their own dialog and their own interactive views
- Represent web pages in an Analytical Application
- They can filter flow variables, i.e., flow variables don't enter or exit a component automatically! We need to allow the flow variable to enter and exit the component!

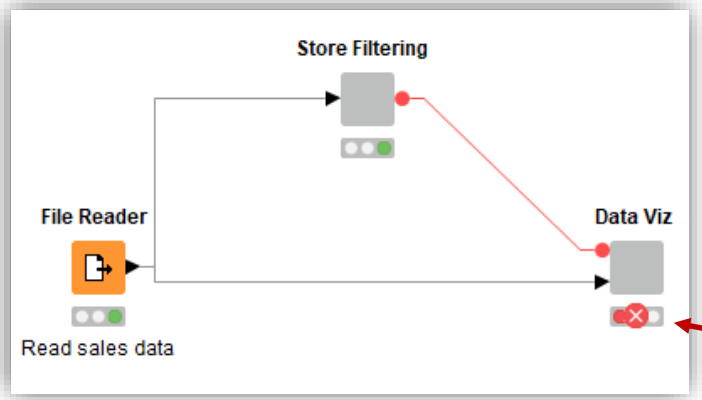


(https://www.knime.com/changelog-v40)

Components

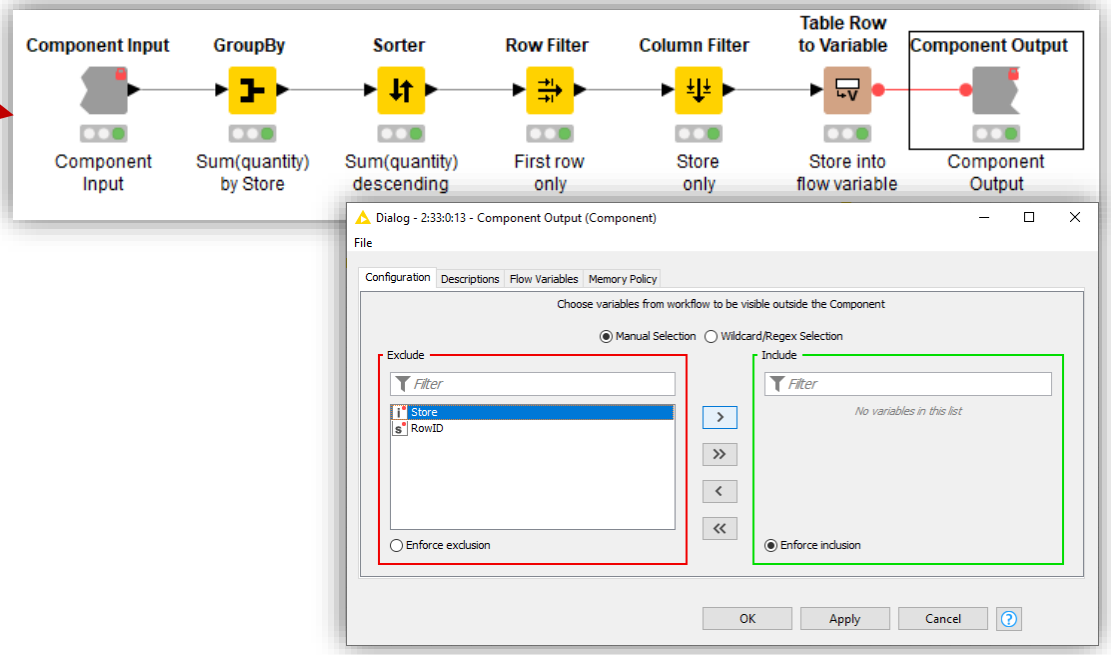






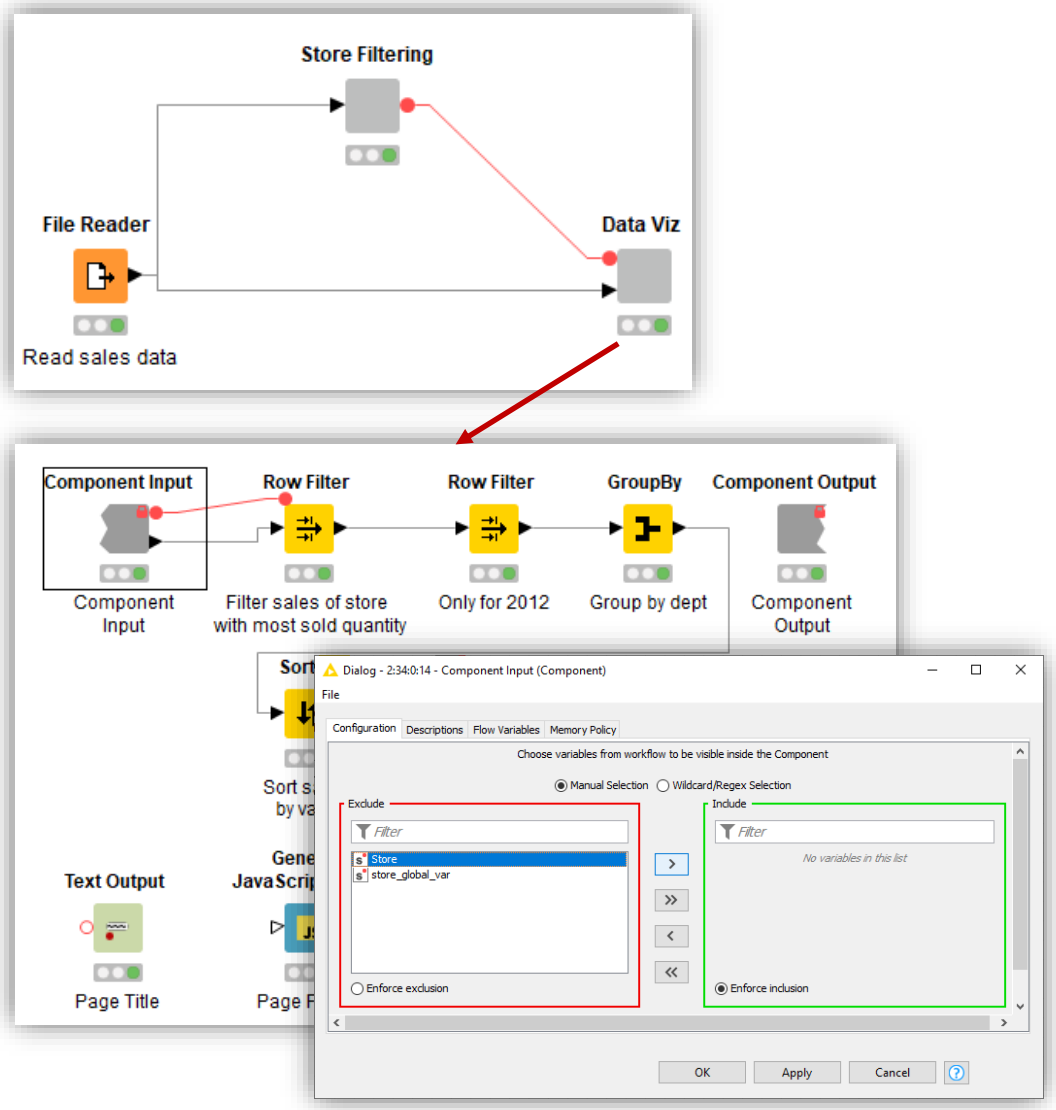
Data Viz component will not execute… Why?

Components



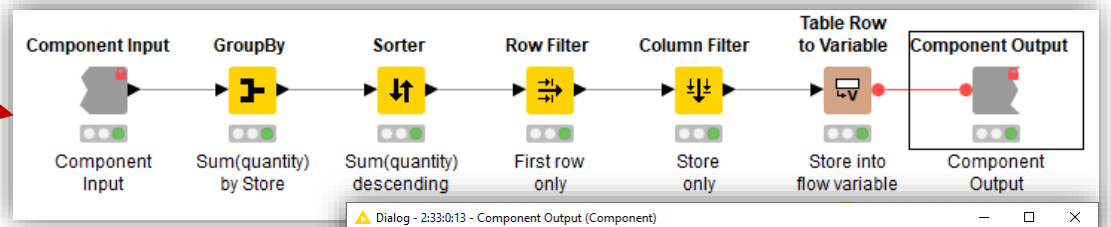
We have to allow the flow variable created in the Data Handling component to exit such component and then allow it to enter the Data Viz one.

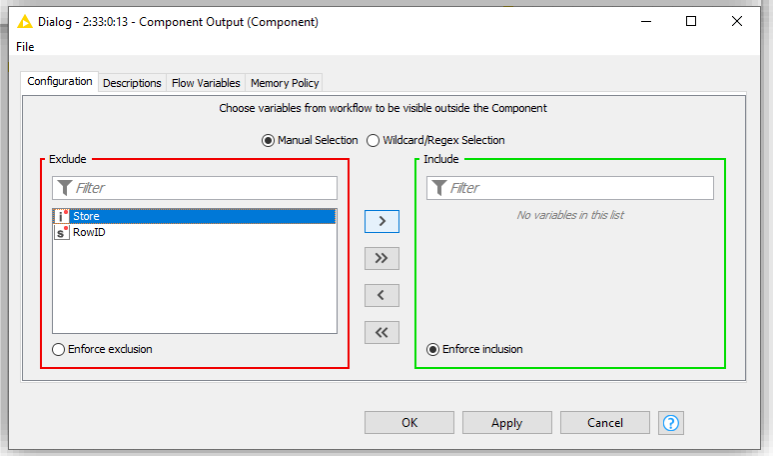
Components

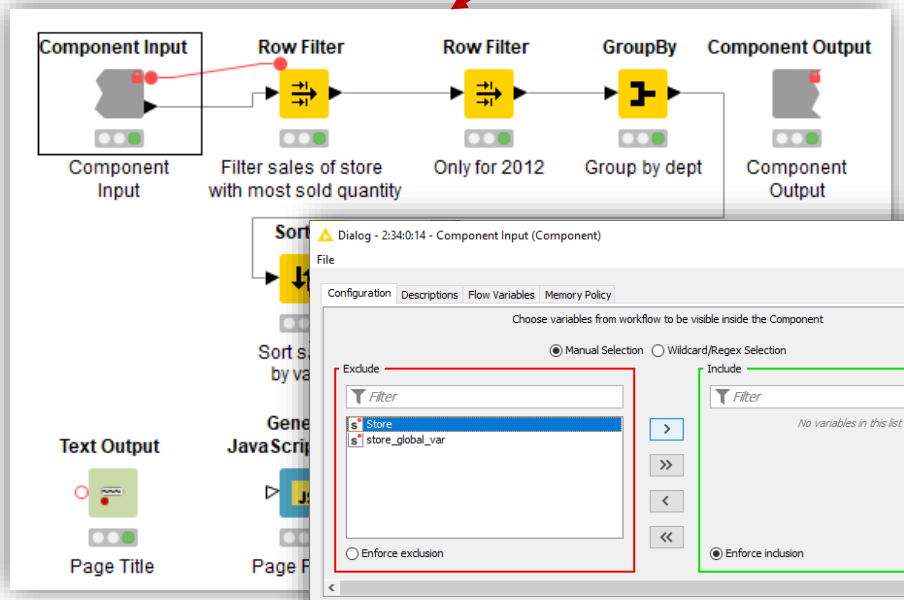


We have to allow the flow variable created in the Data Handling component to exit such component and then allow it to enter the Data Viz one.

Components

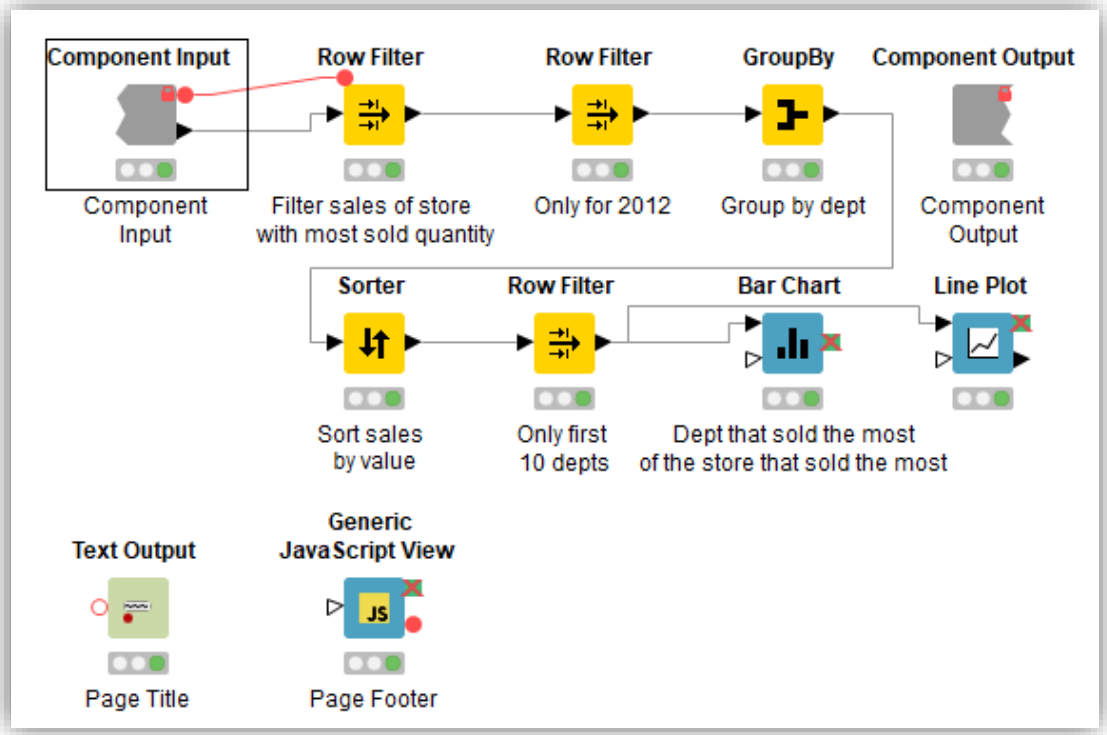


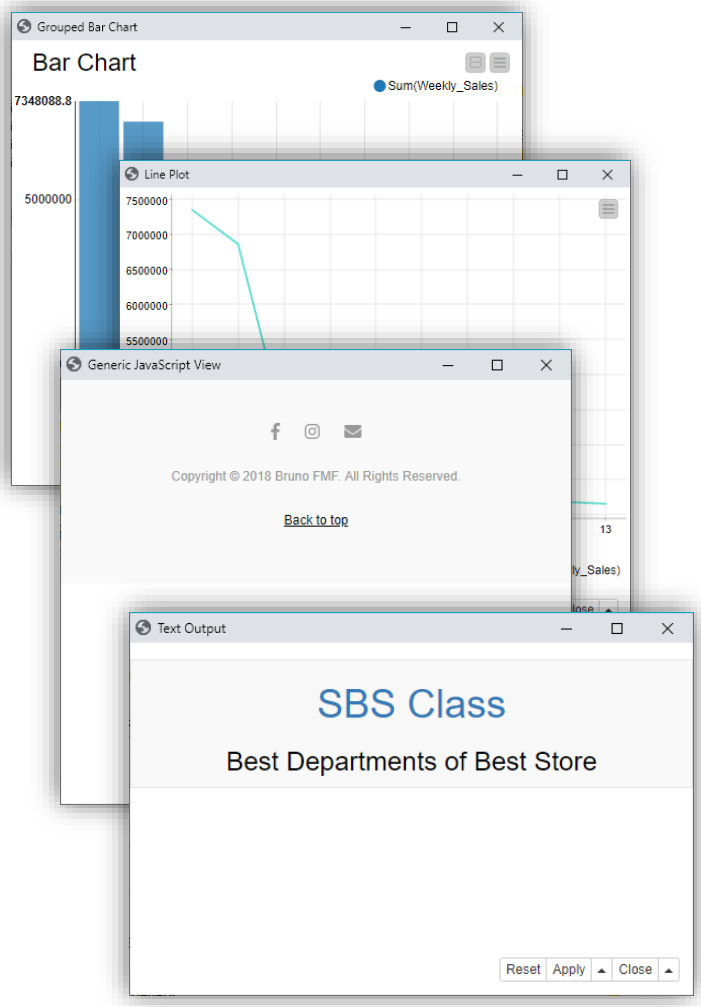




We have to allow the flow variable created in the Data Handling component to exit such component and then allow it to enter the Data Viz one.

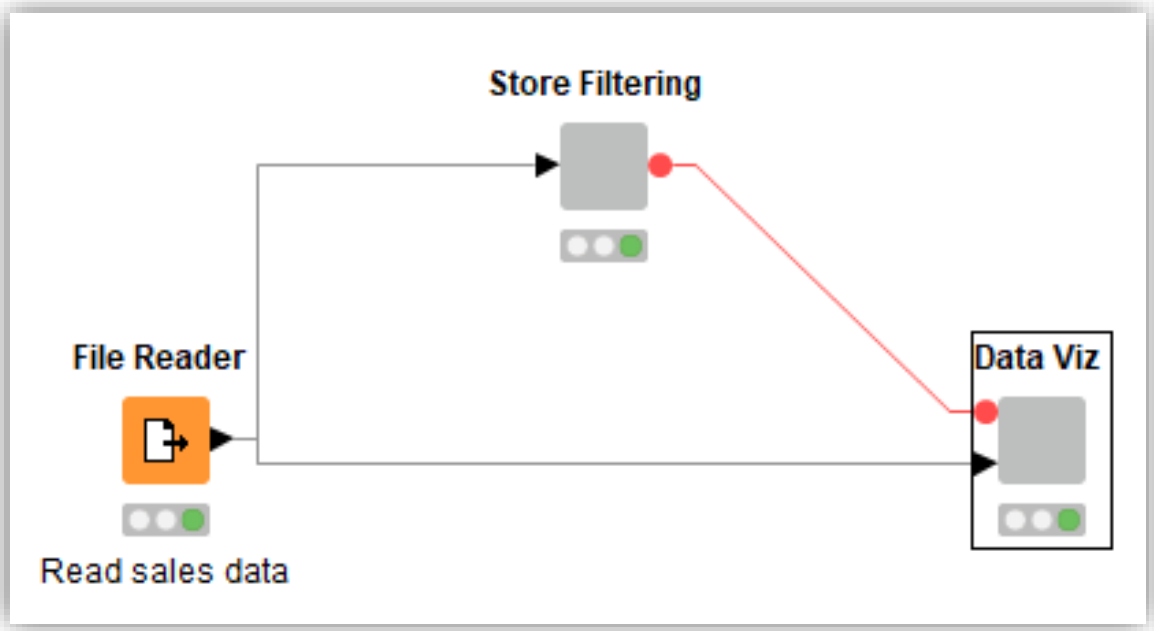
Components and Data Viz

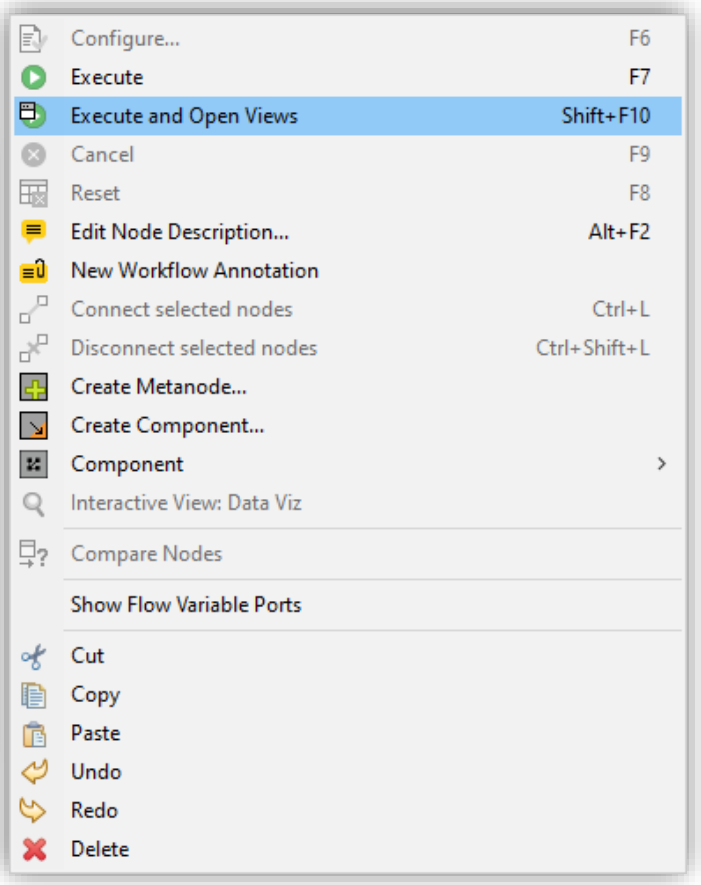




Each plot, chart and javascript widget has its own view.
However…

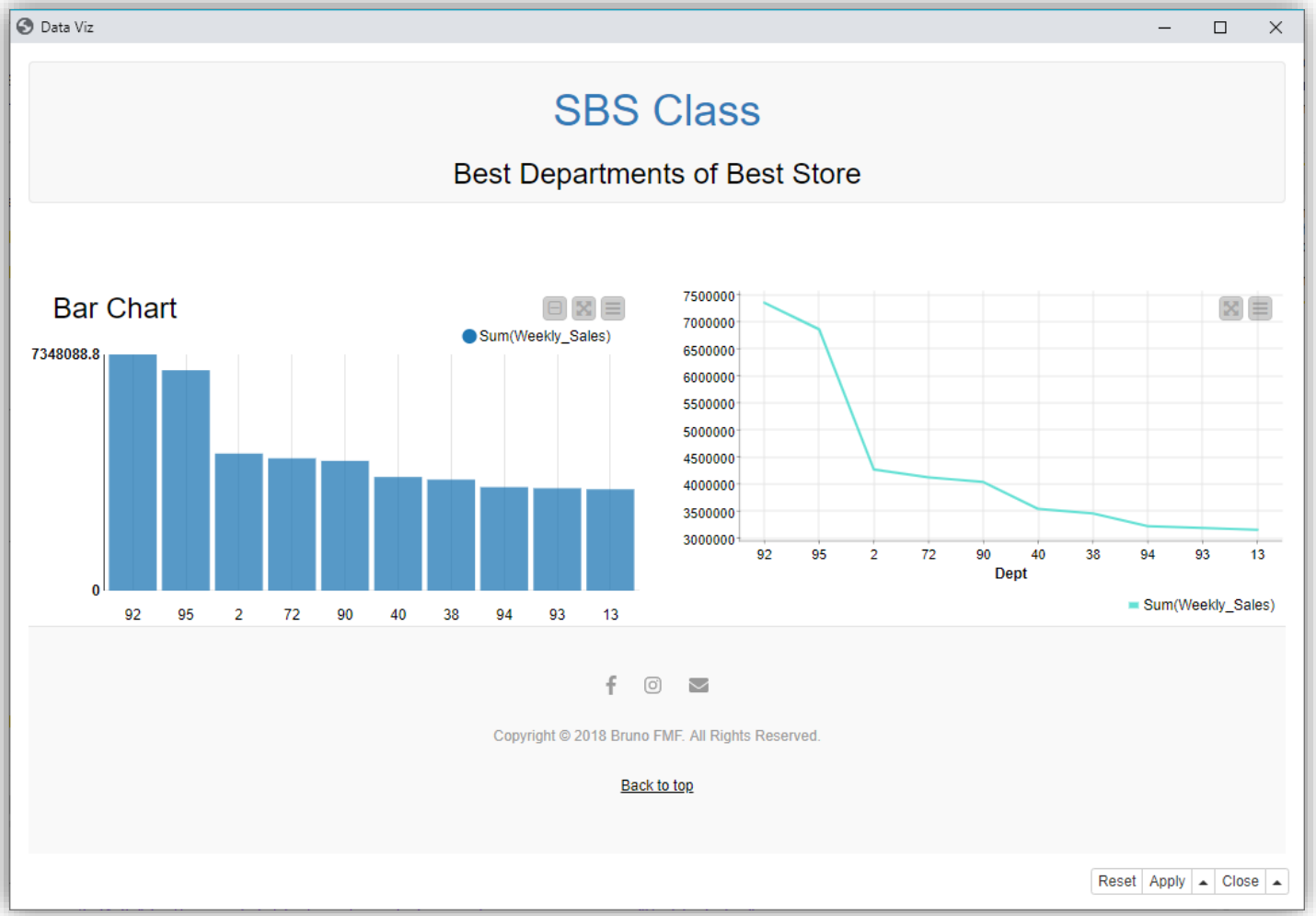
Components and Data Viz



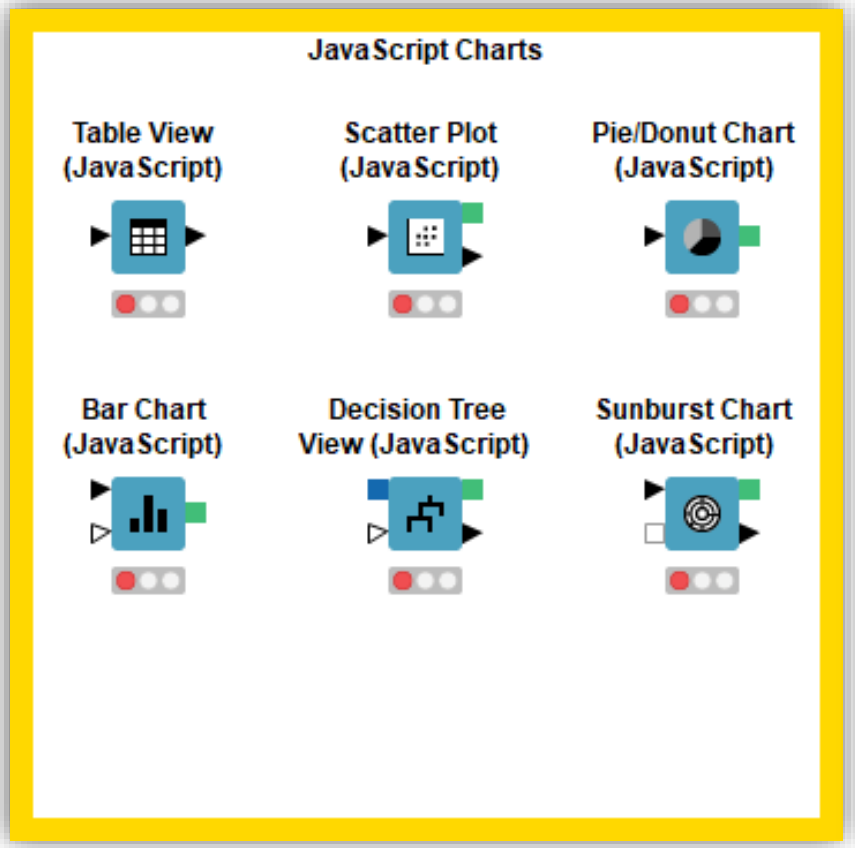


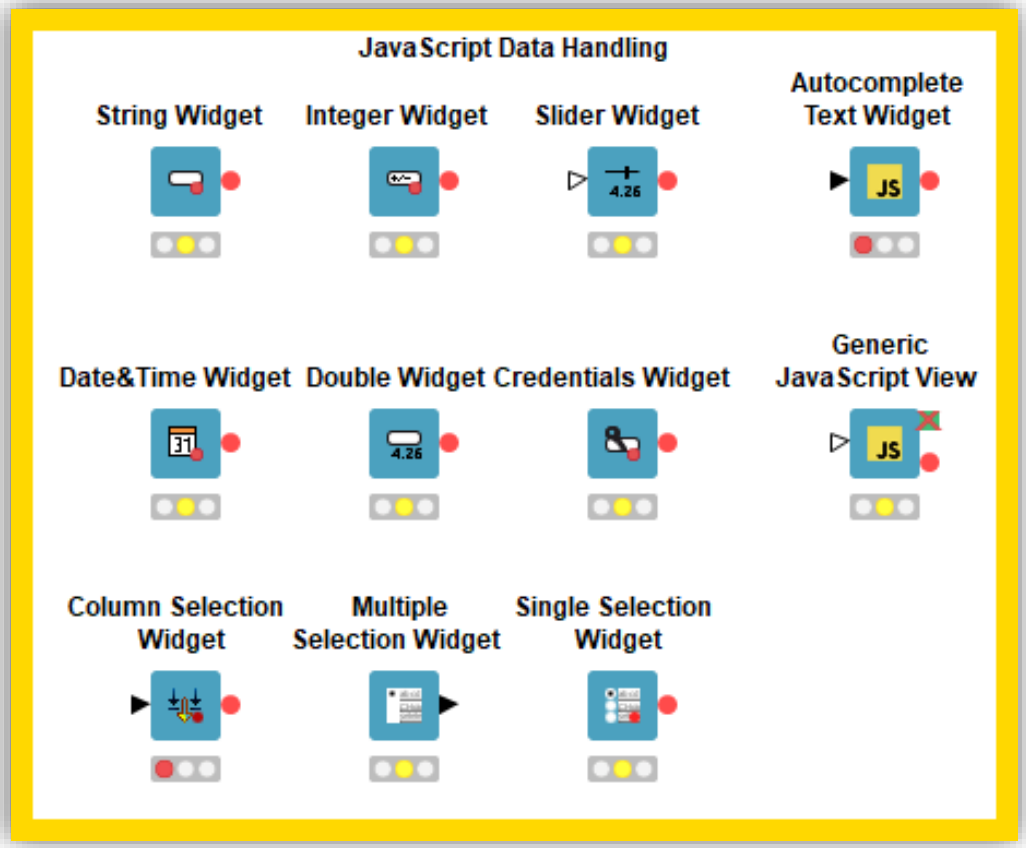
Wrapping the plots/charts/widgets into a Component and executing it…

Components and Data Viz

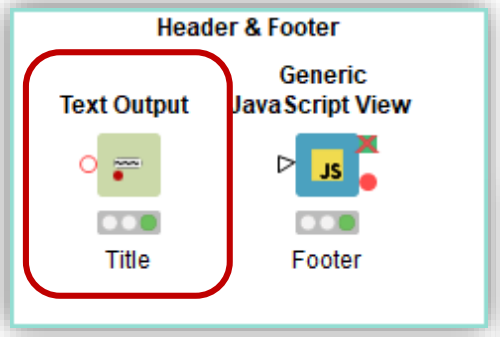


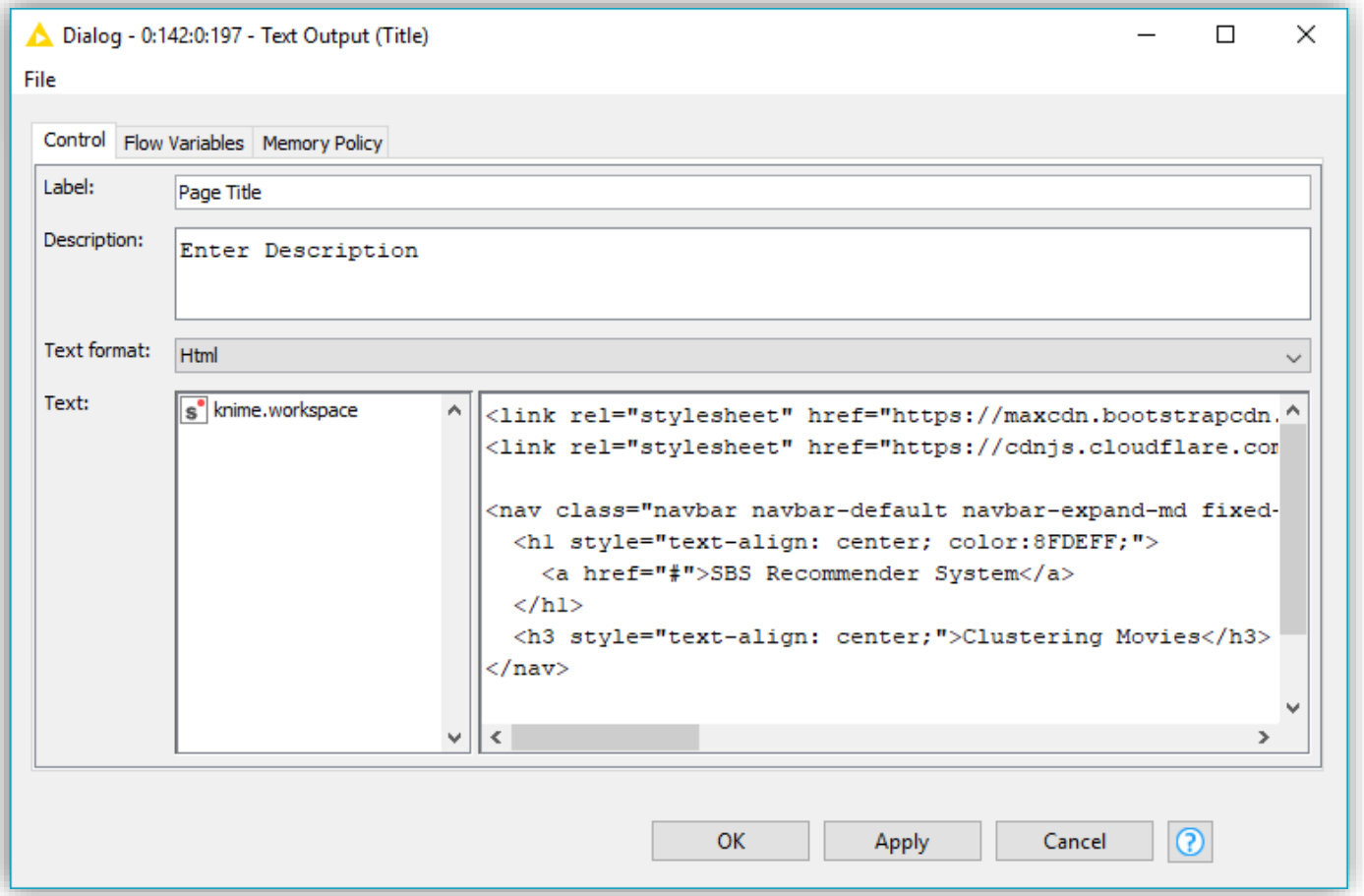
JavaScript Integration



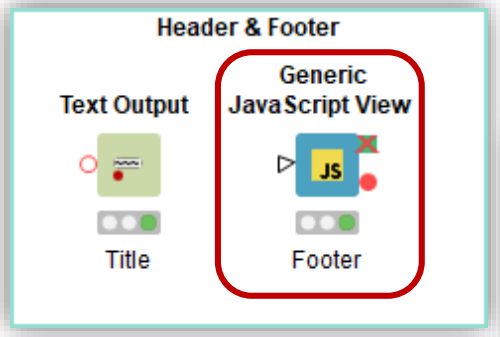


JavaScript Integration





JavaScript Integration



Dialog - 0:142:0:179 - Generic JavaScript View (Footer)

File

JavaScript View | Image Generation | Flow Variables | Memory Policy

Maximum number of rows: 2.500

Flow Variables

knime.workspace

Dependencies

- D3 - Version 4.2.6
- JSFreeChart - Version 0.5.0
- jQuery - Version 3.1.1
- jQuery UI - Version 1.12.1

CSS

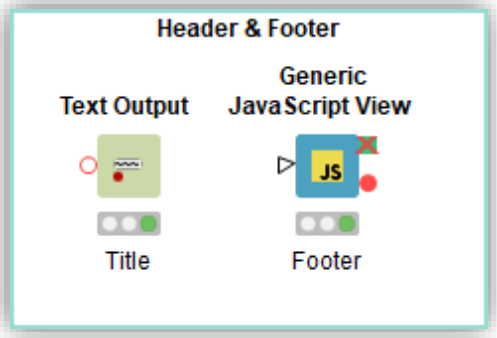
```
/*
 * Footer
 */
.my-footer {
  font-family: 'Raleway', Helvetica, Arial, sans-
  padding: 40px 0;
  color: #999;
  text-align: center;
  font-size: 12px;
  background-color: #f9f9f9;
  border-top: 1px solid #e5e5e5;
}

.my-footer .socialMedia {
  padding: 10px;
  color: #999;
  transition: all .5s ease;
}
```

JavaScript

```
var body = document.getElementsByTagName('body')[
var html = '';

//Load External Scripts - Font Awesome
var external_script = document.createElement('scr
external_script.setAttribute('src','https://use.f
document.head.appendChild(external_script);

//knimeDataTable.getNumRows() or knimeDataTable.g
html += '<footer class="my-footer">   <div>     <

body.innerHTML = html;
```

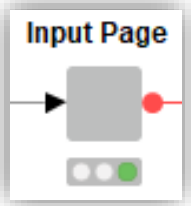
OK | Apply | Cancel

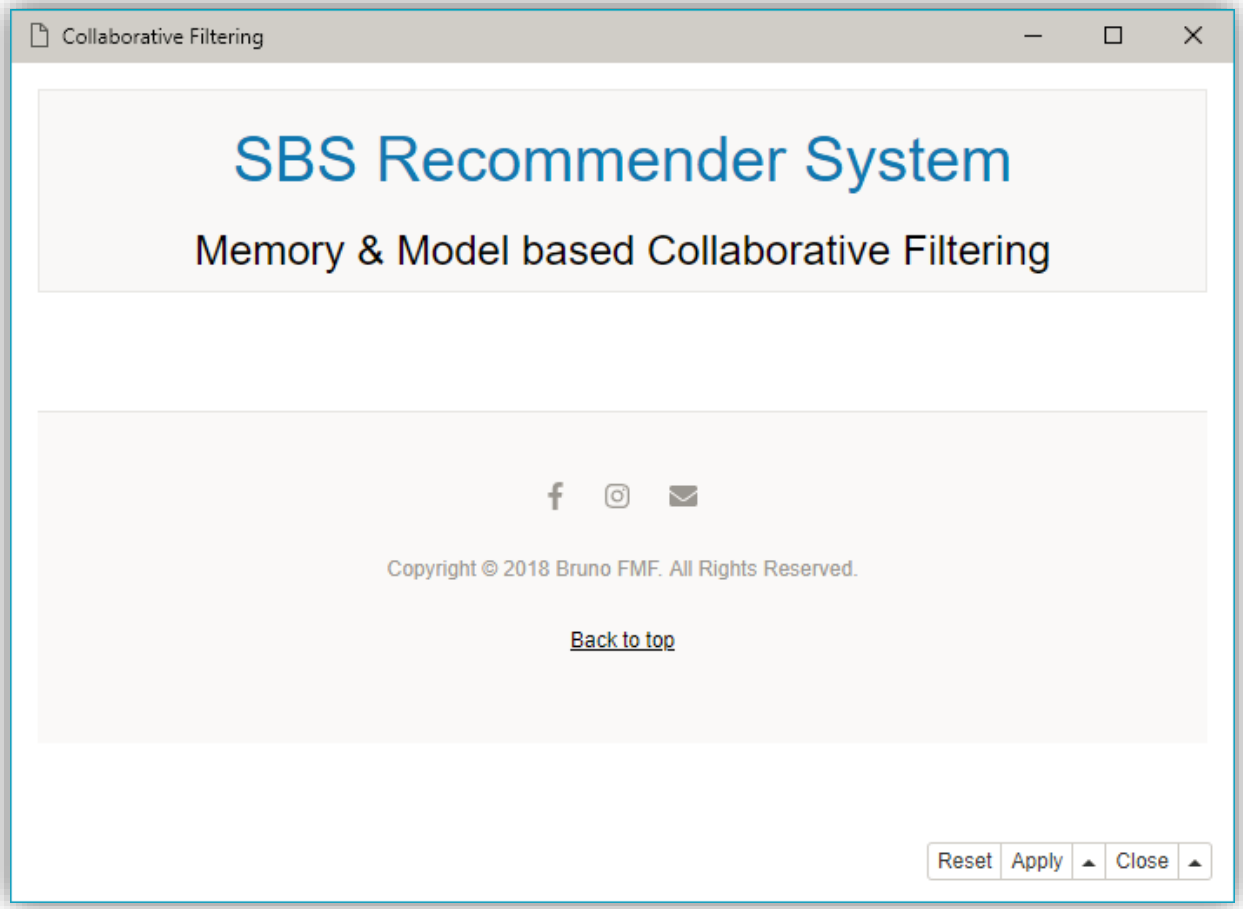
Composite Views



Encapsulated inside a

Component

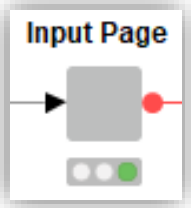


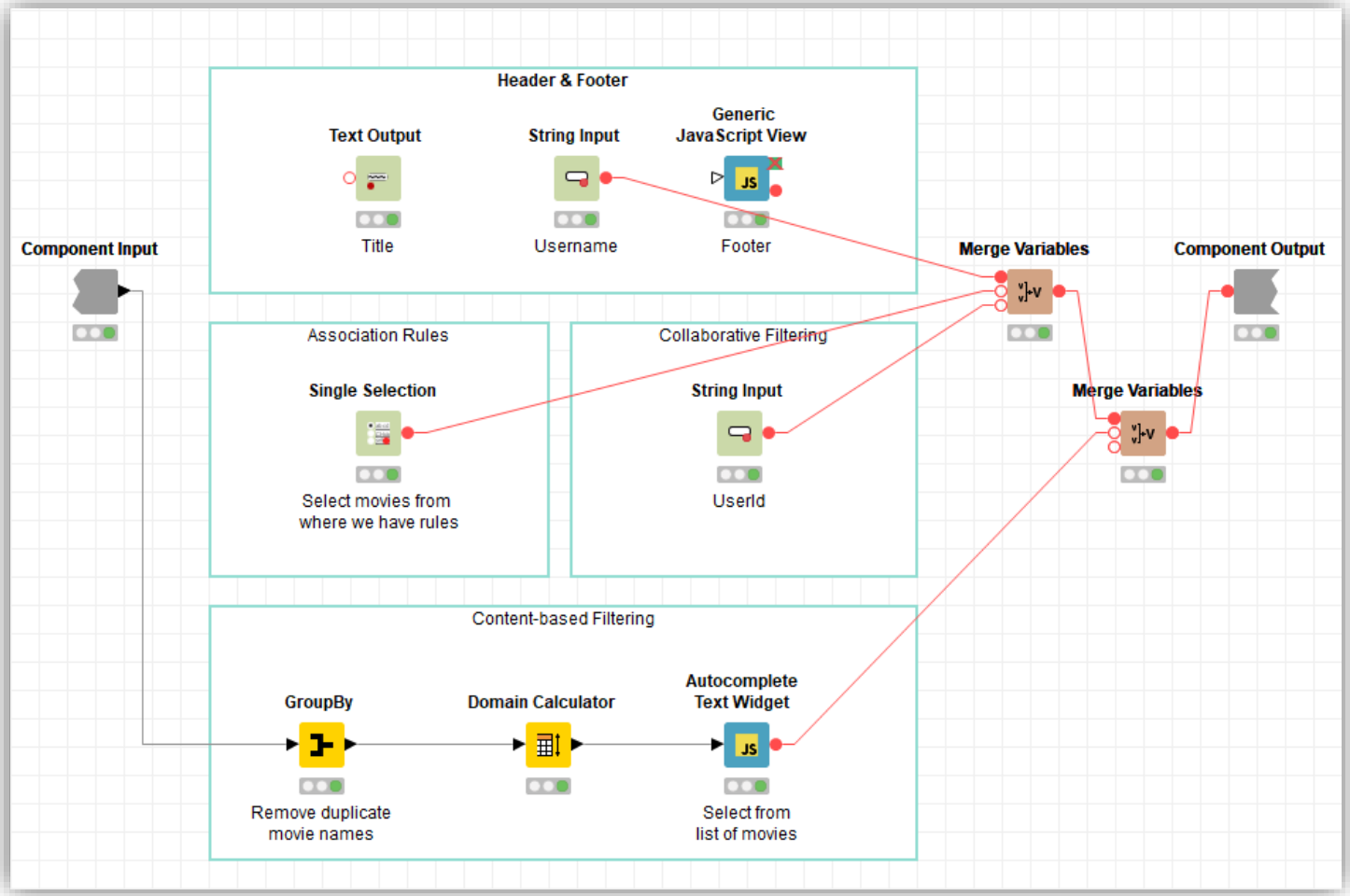


Composite Views

Use JS nodes inside

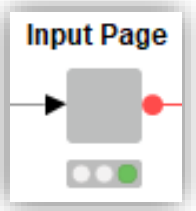
Components!

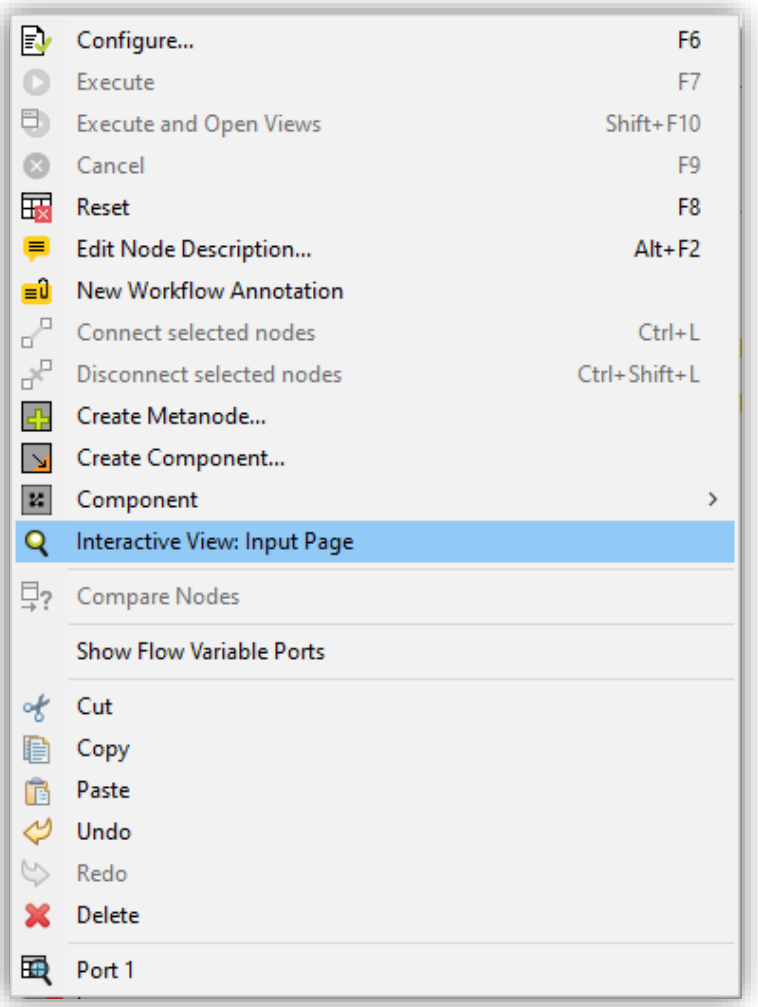




Composite Views

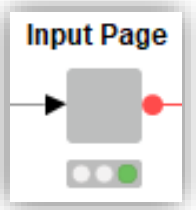
Use JS nodes inside Components!

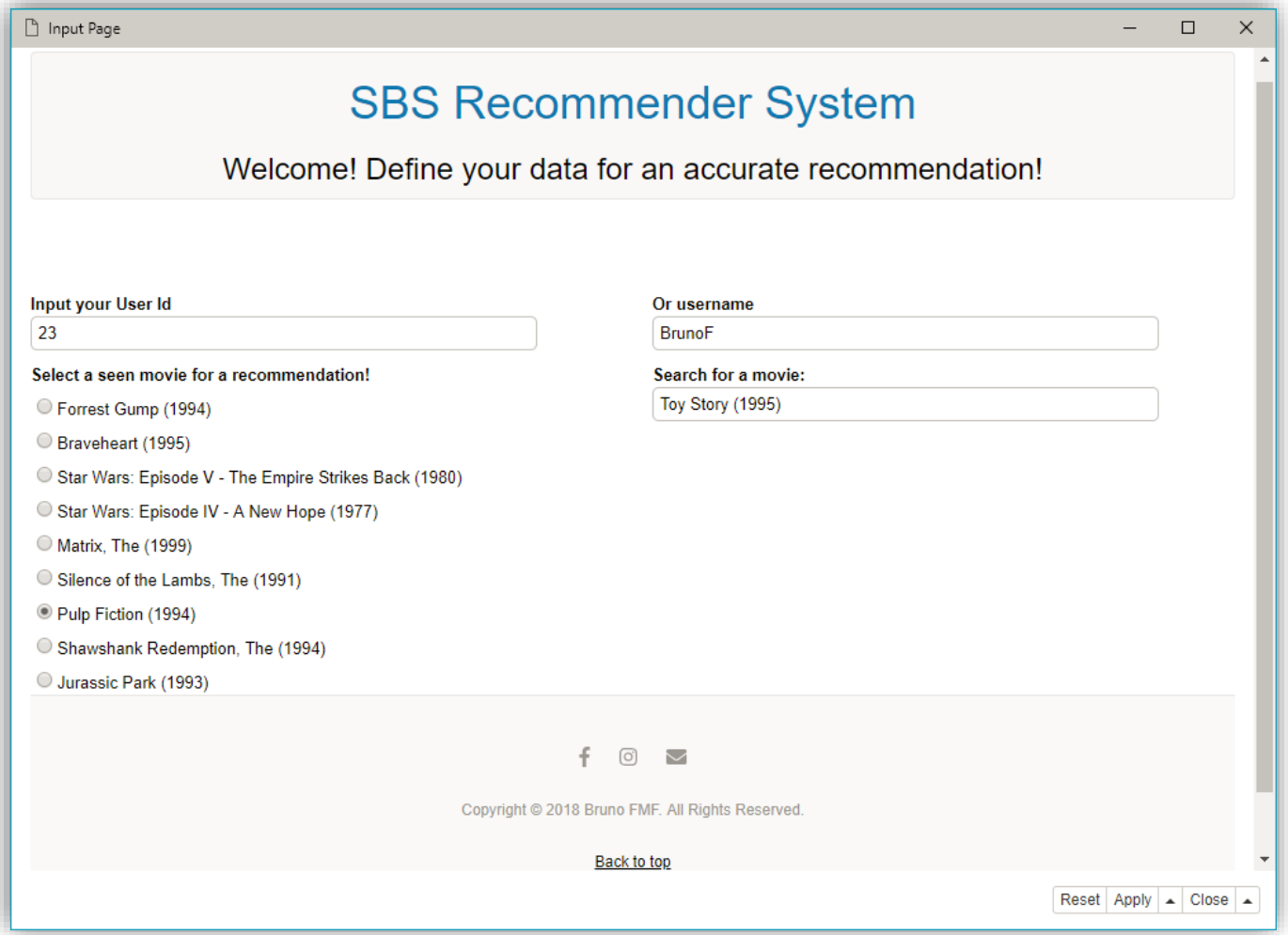




Composite Views

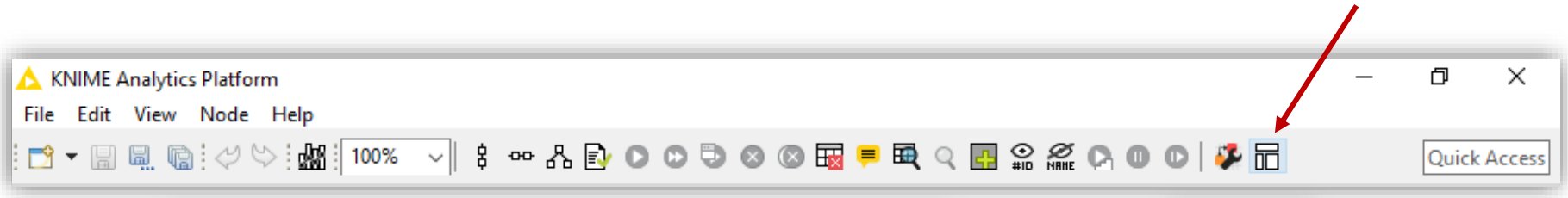
Use JS nodes inside Components!

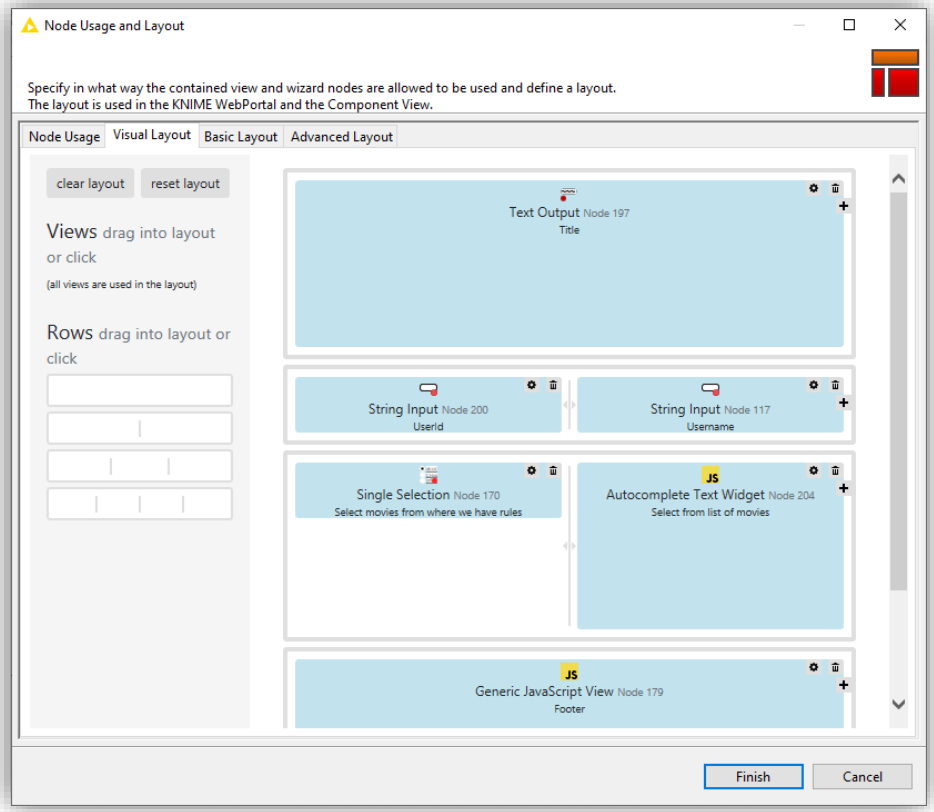


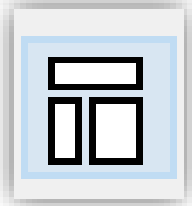


Composite Views

Define the order of the elements inside the page

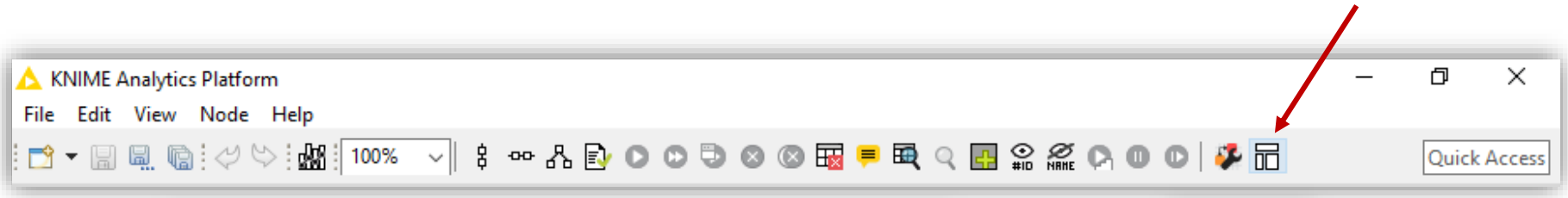


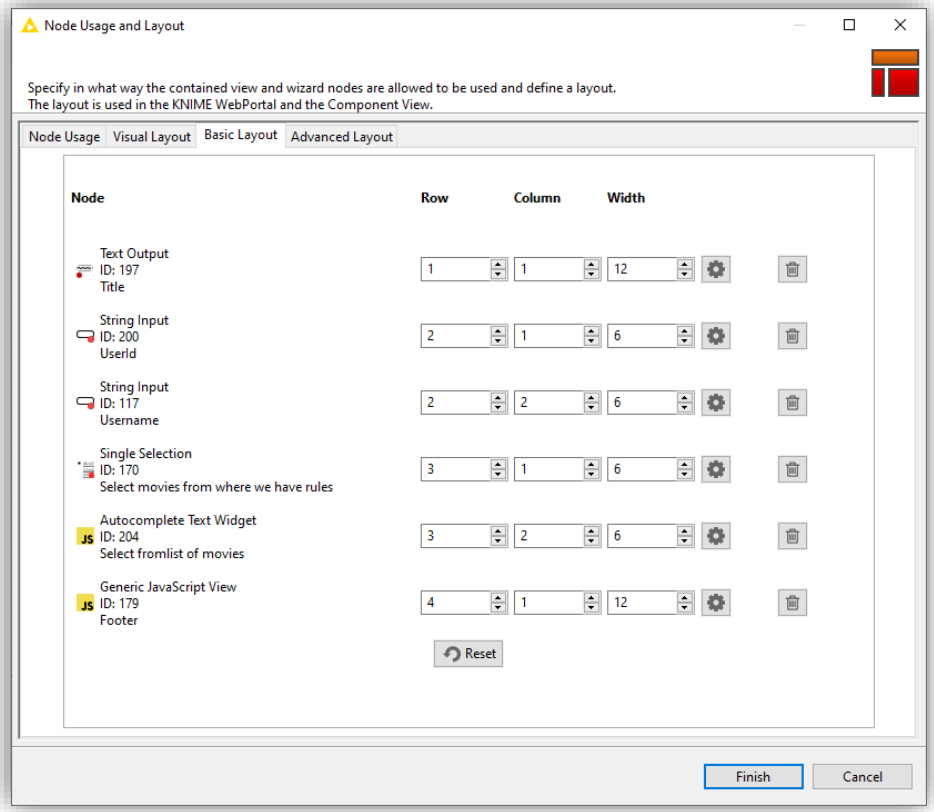


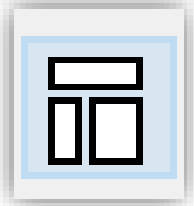


Composite Views

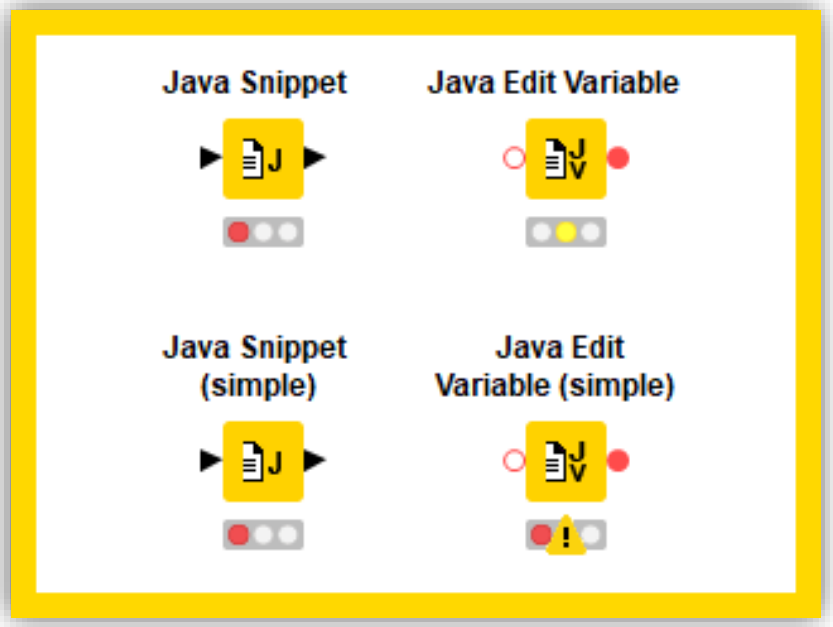
Define the order of the elements inside the page



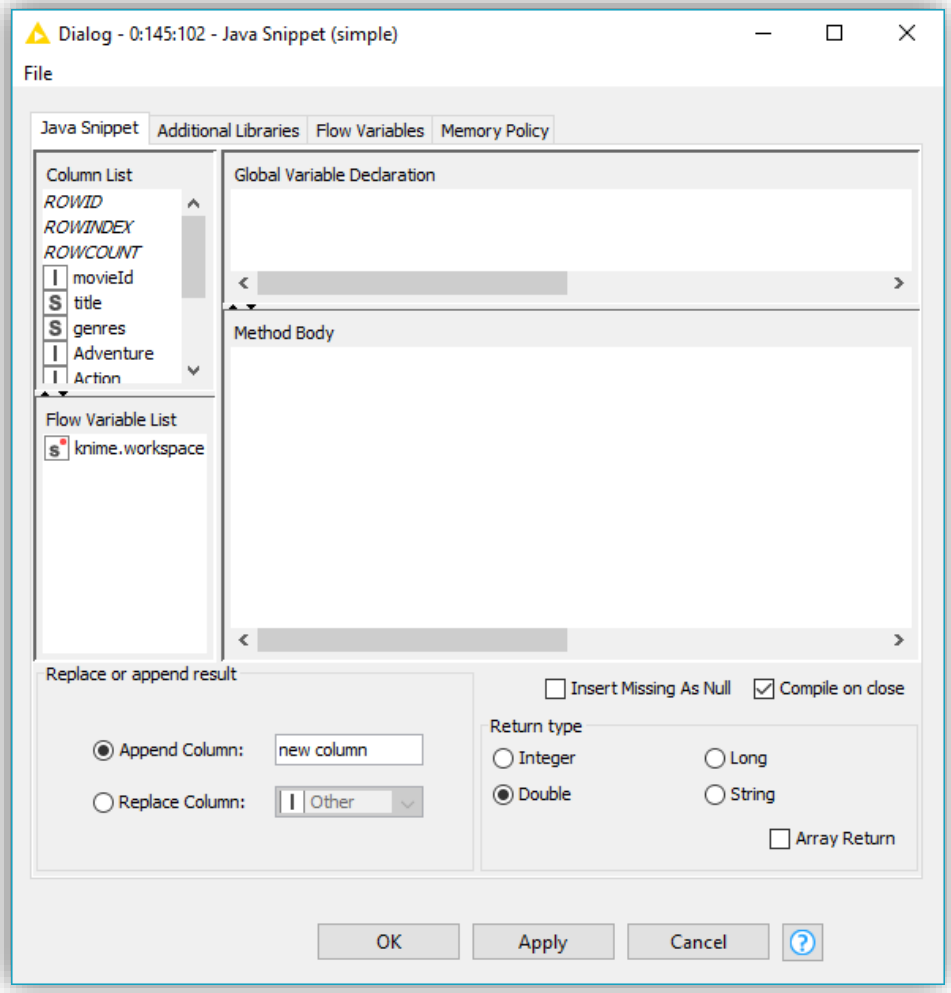




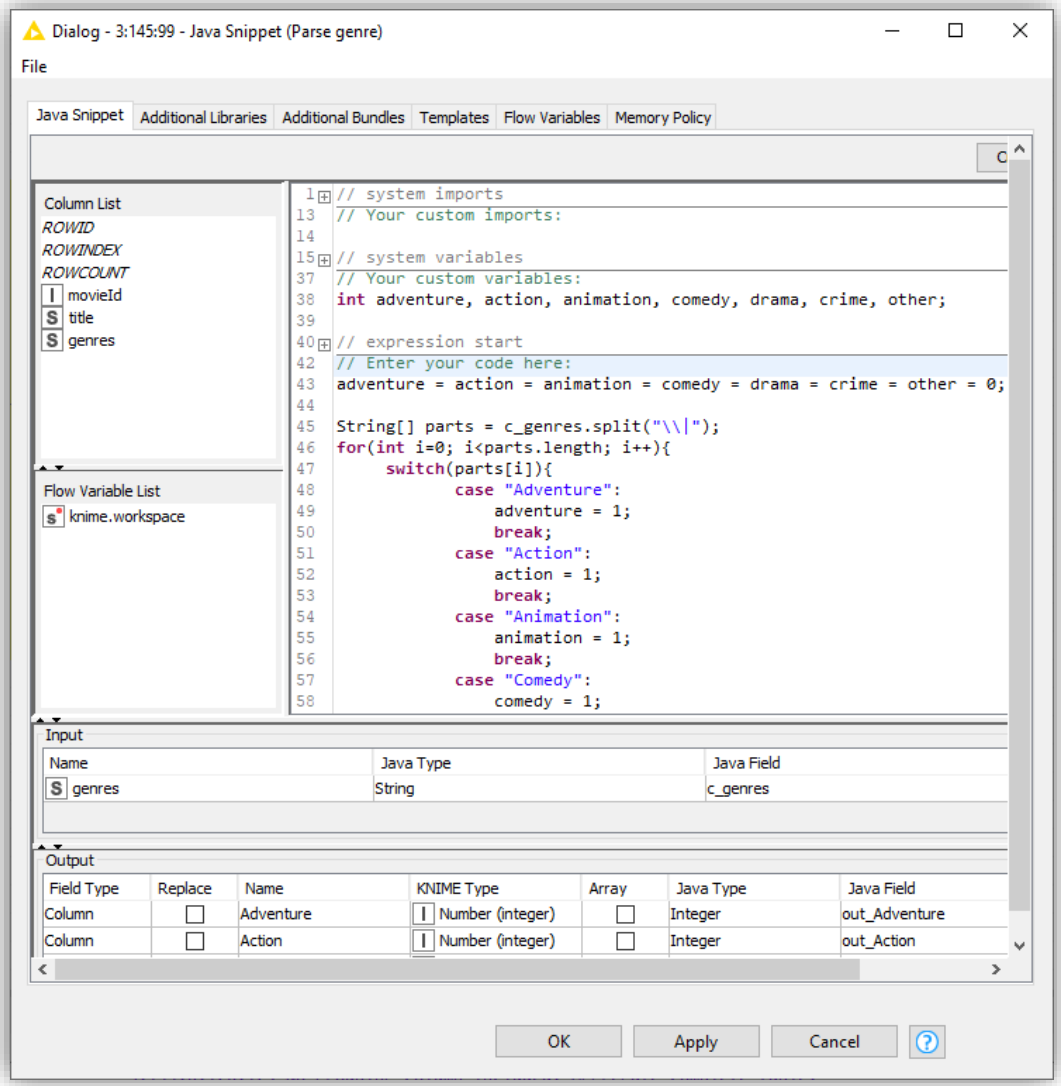
Java Integration



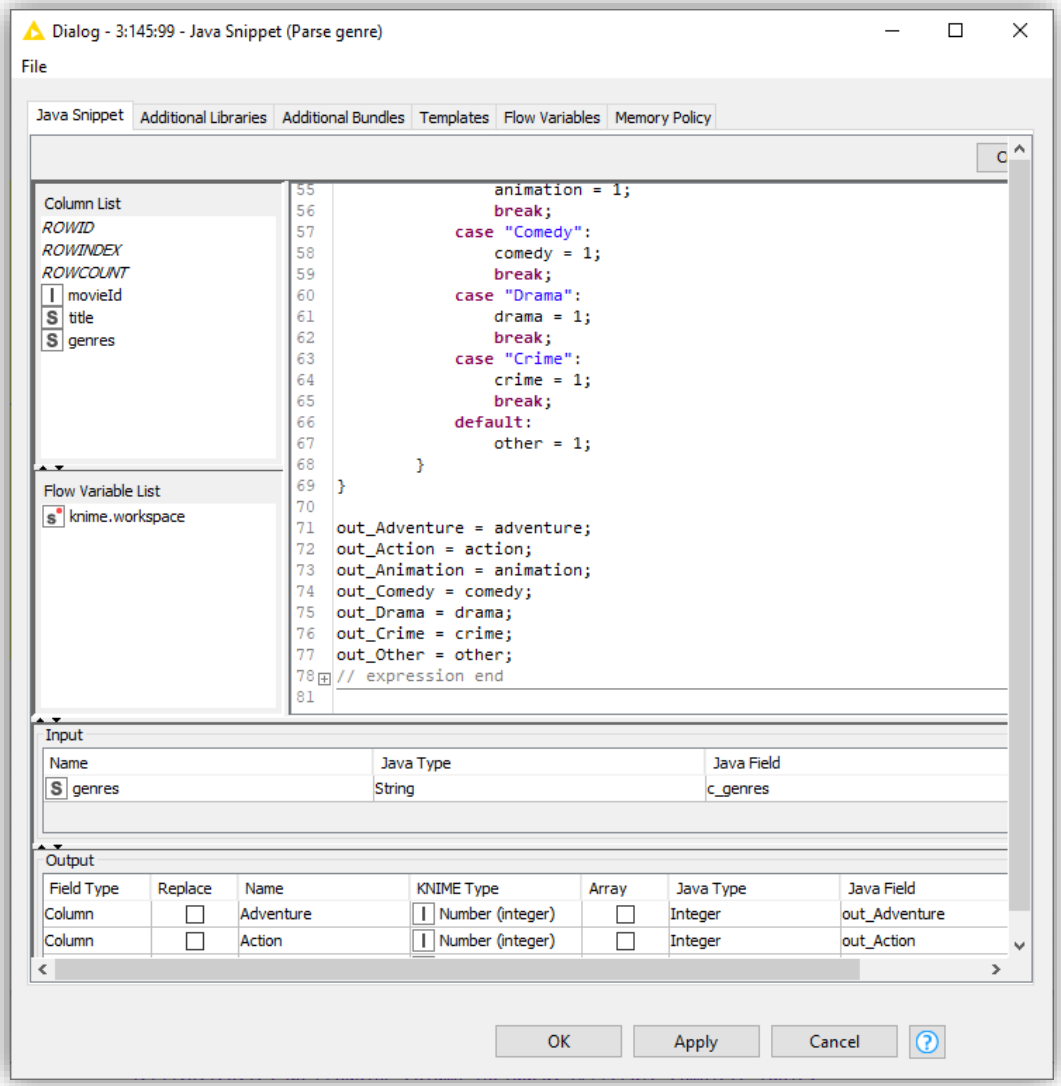
Java Integration



Java Integration



Java Integration



Java Integration

Appended table - 3:145:99 - Java Snippet (Parse genre)

File Hilite Navigation View

Table "default" - Rows: 9742 | Spec - Columns: 10 | Properties | Flow Variables

Row ID	movieId	title	genres	Advent...	Action	Animation	Comedy	Drama
Row0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1	0	1	1	0
Row1	2	Jumanji (1995)	Adventure|Children|Fantasy	1	0	0	0	0
Row2	3	Grumpier Old M...	Comedy|Romance	0	0	0	1	0
Row3	4	Waiting to Exh...	Comedy|Drama|Romance	0	0	0	1	1
Row4	5	Father of the B...	Comedy	0	0	0	1	0
Row5	6	Heat (1995)	Action|Crime|Thriller	0	1	0	0	0
Row6	7	Sabrina (1995)	Comedy|Romance	0	0	0	1	0
Row7	8	Tom and Huck (...	Adventure|Children	1	0	0	0	0
Row8	9	Sudden Death ...	Action	0	1	0	0	0
Row9	10	GoldenEye (1995)	Action|Adventure|Thriller	1	1	0	0	0
Row10	11	American Presi...	Comedy|Drama|Romance	0	0	0	1	1
Row11	12	Dracula: Dead ...	Comedy|Horror	0	0	0	1	0
Row12	13	Balto (1995)	Adventure|Animation|Children	1	0	1	0	0
Row13	14	Nixon (1995)	Drama	0	0	0	0	1
Row14	15	Cutthroat Islan...	Action|Adventure|Romance	1	1	0	0	0
Row15	16	Casino (1995)	Crime|Drama	0	0	0	0	1
Row16	17	Sense and Sen...	Drama|Romance	0	0	0	0	1
Row17	18	Four Rooms (1...	Comedy	0	0	0	1	0

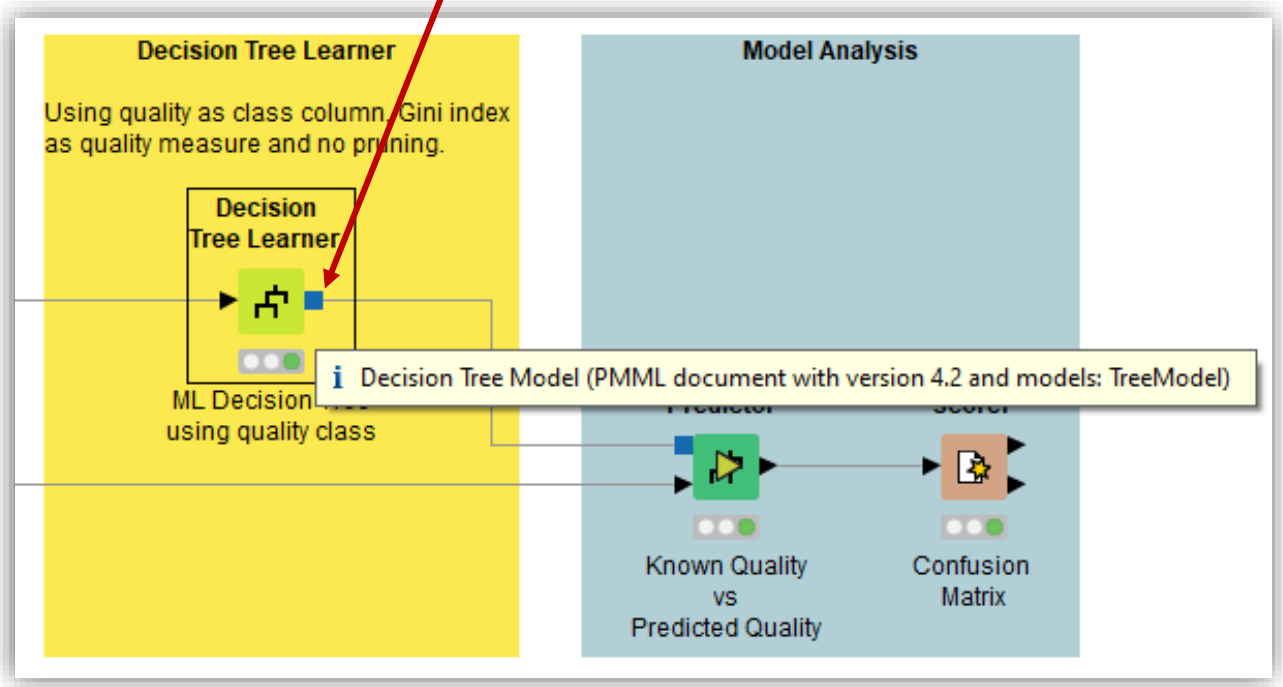
PMML

PMML stands for Predictive Model Markup Language

- It is the leading standard on Data Mining and Machine Learning models representation
- Being developed by the Data Mining Group (DMG)
- It is a XML-based format for interchanging predictive models among different applications
- Enables the instant deployment of predictive solutions
- The most popular software tools for ML already contain features to import/export PMML models

PMML

Most Knime ML nodes export a PMML model (you can save it to the file system using the PMML Writer node)



PMML Structure

Header

Information about who created the PMML, the used software, copyright information and a timestamp

Data Dictionary

Information about the variables in the dataset: their name, type and range

Transformation Dictionary

Includes the pre and post-processing actions to be applied within the predictive model such as normalization, discretization, value mapping, text indexing, functions and/or aggregation

Predictive Model

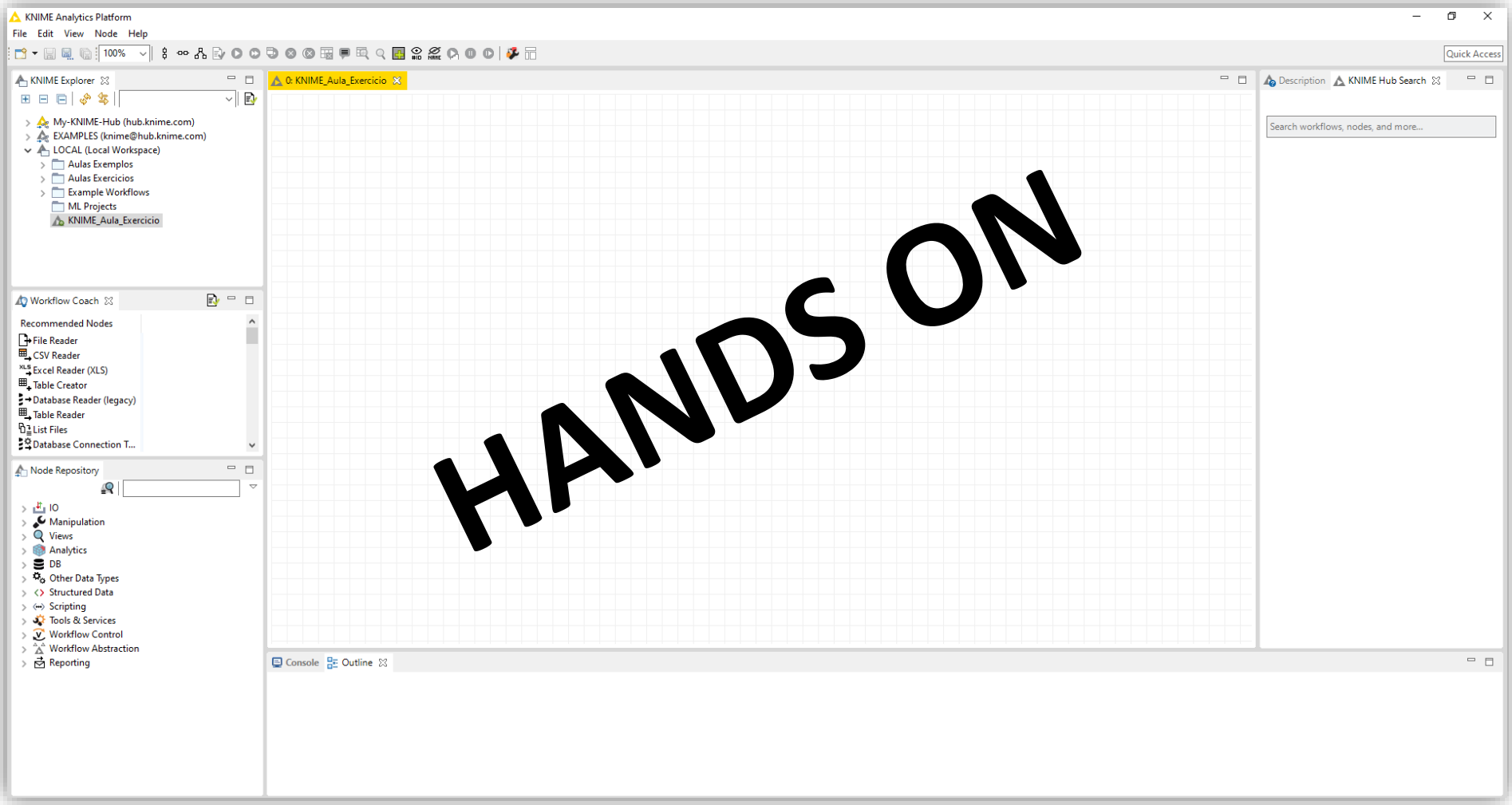
Represents the predictive model itself. For example, a Multi-Layered Feedforward Neural Network is represented by a *NeuralNetwork* element containing attributes such as *Model Name*, *Function Name*, *Algorithm Name* and *Number of Layers*, followed by elements such as *Neural Inputs*, *Neural Layer* and *Neural Outputs*

(http://dmg.org/pmml/v4-3/GeneralStructure.html)

PMML Structure


```
<?xml version="1.0" encoding="UTF-8"?>
<PMML version="4.2" xmlns="http://www.dmg.org/PMML-4_2">
  <Header copyright="user">
    <Application name="KNIME" version="3.6.1"/>
  </Header>
  <DataDictionary numberOfFields="11">
  <TreeModel modelName="DecisionTree" functionName="classification" splitCharacteristic="binarySplit"
  ="returnLastPrediction">
    <MiningSchema>
      <MiningField name="fixed acidity" invalidValueTreatment="asIs"/>
      <MiningField name="volatile acidity" invalidValueTreatment="asIs"/>
      <MiningField name="citric acid" invalidValueTreatment="asIs"/>
      <MiningField name="residual sugar" invalidValueTreatment="asIs"/>
      <MiningField name="chlorides" invalidValueTreatment="asIs"/>
      <MiningField name="total sulfur dioxide" invalidValueTreatment="asIs"/>
      <MiningField name="density" invalidValueTreatment="asIs"/>
      <MiningField name="pH" invalidValueTreatment="asIs"/>
      <MiningField name="sulphates" invalidValueTreatment="asIs"/>
      <MiningField name="alcohol" invalidValueTreatment="asIs"/>
      <MiningField name="quality" invalidValueTreatment="asIs" usageType="target"/>
    </MiningSchema>
    <Node id="0" score="=5" recordCount="1279.0">
      <True/>
      <ScoreDistribution value="=5" recordCount="556.0"/>
      <ScoreDistribution value="=6" recordCount="499.0"/>
      <ScoreDistribution value="=7" recordCount="164.0"/>
      <ScoreDistribution value="=4" recordCount="40.0"/>
      <ScoreDistribution value="=8" recordCount="15.0"/>
      <ScoreDistribution value="=3" recordCount="5.0"/>
      <Node id="1" score="=5" recordCount="786.0">
      <Node id="116" score="=6" recordCount="493.0">
    </Node>
  </TreeModel>
</PMML>
```

Hands On


KNIME Analytics Platform
File Edit View Node Help
KNIME Explorer
My-KNIME-Hub (hub.knime.com)
EXAMPLES (knime@hub.knime.com)
LOCAL (Local Workspace)
Aulas Exemplos
Aulas Exercicios
Example Workflows
ML Projects
KNIME_Aula_Exercicio
Workflow Coach
Recommended Nodes
File Reader
CSV Reader
Excel Reader (XLS)
Table Creator
Database Reader (legacy)
Table Reader
List Files
Database Connection T...
Node Repository
IO
Manipulation
Views
Analytics
DB
Other Data Types
Structured Data
Scripting
Tools & Services
Workflow Control
Workflow Abstraction
Reporting
KNIME_Aula_Exercicio
HANDS ON
Description
KNIME Hub Search
Search workflows, nodes, and more...
Console
Outline
Quick Access