Open for Innovation
KNIME

KNIME Workflow Invocation Guide

KNIME AG, Zurich, Switzerland
Version 5.3 (last updated on 2022-08-04)

Table of Contents

Introduction

Workflows can be deployed as REST Services on a KNIME Server, and can be tailored to almost any of our REST based needs and make the KNIME stack a powerful integration tool. We can use workflows to enhance existing third party REST endpoints, create brand new REST services, or expose several systems as one REST interface. We can even use these workflows to manage the KNIME Server itself. You can have information about how to access the KNIME Server REST API in the KNIME Server User Guide.

The following guide, instead, focuses on how to create:

1. *Callee workflows*: These workflows can expose data to a caller, either a *call workflow* or a REST interface
2. *Call workflows*: These workflows can call other workflows either locally or deployed to a KNIME Server

In order to invoke workflows via REST, a KNIME Server installation is needed, but it is possible to build sets of caller/callee workflows which can be used within KNIME Analytics Platform.

Two sets of nodes are available to build these type of workflows. One set of nodes (see KNIME Workflow Invocation (external clients) section) is primarily made to define APIs for 3rd party clients, whereas the set of nodes which are available as Labs starting with KNIME Analytics Platform version 4.5, is for KNIME-use only (see KNIME Workflow Services section).

KNIME Workflow Invocation (external clients)

When uploading KNIME Workflows on KNIME Server they immediately are also deployed as a REST Service. It is then possible to interact with these workflows via the KNIME Server REST API interface from any REST client.

In order to do these operations the workflows that need to be called (*callee workflows*) will need to have special nodes that expose different data types to the REST interface.

For more information about how to build workflows that are able to use this functionality please go to the KNIME Workflow Invocation (external clients) section.

KNIME Workflow Services

Moreover, it is also possible to invoke workflows that are deployed to KNIME Server or that

exist on the local workspace of your KNIME Analytics Platform by using other workflows (*call workflows*).

Workflow Services make it much easier to call KNIME workflows from other workflows. Rather than invoking standardized (but limiting) JSON-based APIs, it is now possible to create and connect directly to KNIME native API endpoints.

This means that it is possible to share many more KNIME data types, beyond what is possible with JSON, e.g., text documents, decision trees, deep learning models, and many more.

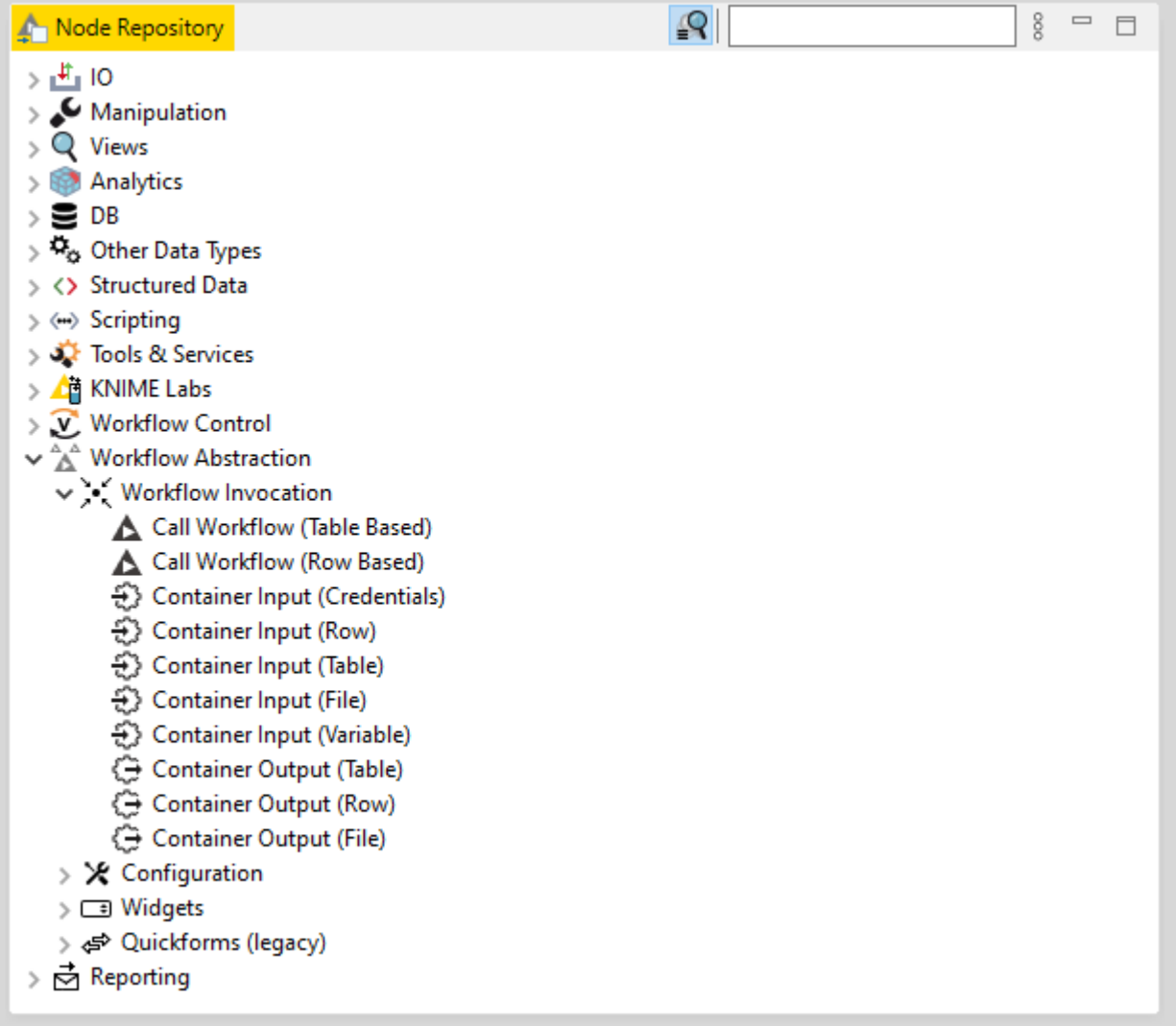
Workflow Services also allow you to define multiple input and output nodes of either consistent or varying data types. These nodes can then be efficiently called with the Call Workflow Service node.

For more information about how to build workflows that are able to use this functionality please go to the KNIME Workflow Services section.

KNIME Workflow Invocation (external clients)

Workflow Invocation nodes

In KNIME Analytics Platform node repository you can find, under *Workflow Abstraction* > *Workflow Invocation*, some useful nodes to interact with your workflows via REST interface and/or by call workflows.



These nodes can be divided into the following categories, depending on their functionality:

- Container nodes. Every type of Container Input node defines the exact format (JSON) the workflow needs to be called with, and the Container Output node how the output looks like.
    - Container Input nodes. Container Input nodes can receive different kind of data formats by an external caller and/or from a REST interface and make them available to the workflow. The following are different kind of inputs the Container

Input nodes can receive:

- JSON
- Row
- Table
- File
- Variable
- Credentials

  - Container Output nodes. Container Output nodes can send different kind of data formats to an external caller and/or the REST interface.

- Call nodes. These nodes can be used to call other workflows that resides either locally or on a KNIME Server. These nodes use the very same interface/mechanism used by the Container Nodes. You can use these nodes to build a *call workflow* which can be used to call a *callee workflow* in order to either test the service deployed for 3rd party clients, or to use a service from within a KNIME workflow that you also want not only to make accessible to 3rd party clients. In case you want to build a workflow service to be use only within KNIME we recommend to use the new set of nodes described in the KNIME Workflow Services section.

Each of the Container nodes is able to expose data (Output) or to receive data (Input) from a set of specific callers (either via REST interface or by/to a caller workflow).

Table 1. Container nodes and type of calls pairings

Container Input	**Caller**
Credentials	Call Workflow (Table Based)
Row	• Call Workflow (Row Based) • REST interface
Table	Call Workflow (Table Based)
JSON	• Call Workflow (Row Based) • REST interface
File	REST interface
Variable	REST interface

Container Output	Caller
Row	• Call Workflow (Row Based) • REST interface
Table	• Call Workflow (Table Based) • REST interface
JSON	• Call Workflow (Row Based) • REST interface
File	REST interface

Container nodes and REST interface

Below is an explanation of the usage of the nodes that can be called only via a REST interface. For clarity we are going to indicate the `curl` command you need to use to invoke the REST endpoints. You can use anyways any of the several tools available to invoke REST endpoints, such as RESTClient, SoapUI, Postman, or KNIME Analytics Platform.

You can read a hands on guide about building workflows for KNIME Server REST API on KNIME blog.

Container Input (Variable) node

This node receives flow variables from an external caller like the REST interface and makes them available to the workflow. A configured parameter makes the Container Input (Variable) visible from the external caller and enables the external caller to send the variables to the Container Input (Variable) node.

In the configuration dialog of the Container Input (Variable) node you can choose a name for the variable input parameter and append to it unique ID values. You can also add a description for the variable input parameter.

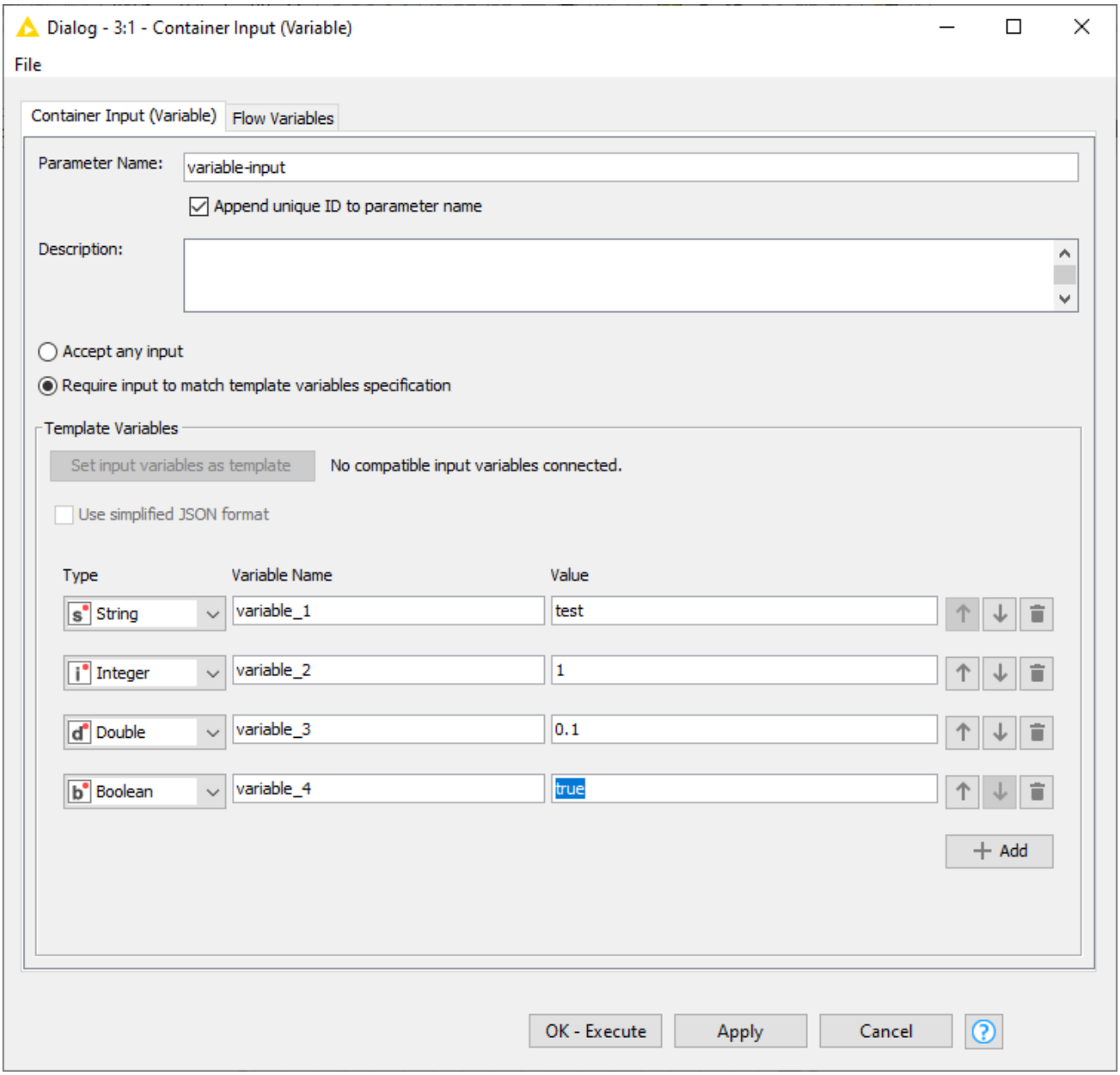
By default the node will accept any input. To be accepted the input will need to be well formed and it can contain an arbitrary amount of variables. The input has to be a JSON object. Each property (key/value) pair that it might contain represents a variable, where the key property is the variable name and the value property is the variable value. The type of the

variable is determined by using the JSON type of the value property.

Four basic data types are supported:

- *String*: A string of characters.
- *Integer*: An integer number.
- *Double*: A floating point decimal number.
- *Boolean*: A truth value that can be either `true` or `false`.

You can also define a number of template variables specification. Only the input that will match those templates will be accepted and their value exposed to the workflow. To do so check the option *Require input to match template variables specification* and click *Add* to add the templates.



If you define only one template variable you can also check the option *Use simplified JSON format*. This will allow the external input format to be simpler by using the value directly instead of an object that defines the variables as properties. For example, if this option is enabled you can define a variable by using the following simplified `InputParameters`:

```
{
    ...
    "parameter-name": <value>,
    ...
}
```

instead of the object notation:

```
{
    ...
    "parameter-name": {
        "variable-name": <value>
    },
    ...
}
```

Please note that in this case the variable will always have the same name as the parameter name without the unique ID appended.

You can also feed flow variables from an another node output, e.g. you can define variables via the Variable Creator node, and set those input variables as template, either by replacing or merging the template variables already defined, if any.

Once you have deployed the workflow where the Container Input (Variable) node is present to your KNIME Server you can execute it via REST API. You can either access it via the SWAGGER UI, by right clicking the deployed workflow from KNIME Explorer in KNIME Analytics Platform and selecting *Show API definition* from the context menu. In SWAGGER UI choose *executing* to see the POST request specifications of the workflow. Under *Request body* section you will see, if defined, the template variables with either simplified JSON or object notation format.

An example of a curl command to execute a POST request with a variable input template is provided, where `{{baseUrl}}` is the URL of your KNIME Server and `{{path}}` is the path relative to the Server file system to the workflow that you want to execute.


```
curl --location -g --request POST '{{baseUrl}}/rest/v4/repository/{{path}}:execution' \
--data-raw '"{
  "variable-input": {
    "variable-input": "test"
  }
}"'
```

Container Input/Output (File) nodes

The Container Input (File) and Contained Output (File) nodes can be used for file upload and download respectively.

Due to some limitations of the SWAGGER UI you can not specify how to call the Container Input (File) node since you would need a different content type for the input file, such as a multi-part content type instead of a JSON type and this is not possible to specify it via the SWAGGER UI.

Instead you will need to use another tool to invoke the REST endpoint. Here, we will give example commands based on `curl`.

Upload a file via REST API call

In order to upload a file to a workflow you will need to add a Container Input (File) node to your workflow. When configuring this node you can give a name to the parameter successively identifying your input file, choose to append to it a unique ID, and give the input parameter a description. Then you can define the name of the Path type flow variable that contains the location of the local copy of the resource. Finally you can also decide to use a default file in case no external resource is uploaded, but an external resource will always take precedence over a default file. Only the path of a default file will be exposed.

Then, to upload a file to a REST API enabled workflow where the Container Input (File) node is present you can use the following `curl` command:

```
curl --location --request POST '{{baseUrl}}/rest/v4/repository/{{path}}:execution' \
--header 'Content-Type: multipart/form-data' \
--form 'input-file=@"{{file-path}}"'
```

where `{{baseUrl}}` is the URL of your KNIME Server and `{{path}}` is the path relative to the Server file system to the workflow that you want to execute. Via this command you are also defining the content type of the file you want to upload as `form-data` type and `{{file-path}}` is the path to that file.

Download a file via REST API call

In order to download a file from a workflow you will need to add a Container Output (File) node to your workflow. When configuring this node you can give a name to the parameter successively identifying your output file, choose to append to it a unique ID, and give the input parameter a description. Then you need to define the variable that contains the path to the resource to be exposed. This needs to be of the Path type.

Since, at the current status, the execution endpoint of the Server REST API deletes the respective job after a successful execution, any output files for download are deleted, too. Thus, in order to download a file from a REST API enabled workflow where the Container Output (File) node is present you will need to go through the following steps:

1. Create a job for the workflow on KNIME Server: To do so send a POST request to the `jobs` endpoint

   ```
   curl --location --request POST '{{baseUrl}}/rest/v4/repository/{{path}}:jobs'
   ```

2. Execute the created job: To do so send a POST request to the `jobs` endpoint, adding the identifying ID of the job `{uuid}` which you will find in the output response of the previous request under `"id"`

   ```
   curl --location --request POST '{{baseUrl}}/rest/v4/jobs/{uuid}'
   ```

3. Download the created file: To do so send a GET request to the `output-resources` endpoint

   ```
   curl --location --request GET curl '{{baseUrl}}/rest/v4/jobs/{uuid}/output-
   resources/{parameter-name}' \
   -o {parameter-name}
   ```

   where `{parameter-name}` is the one configured in the Container Output (File) node configuration dialog and that you can find in the output response of the previous request under `"outputResources"`.

4. Finally, you can optionally delete the executed job: To do so send a DELETE request to the `jobs` endpoint

   ```
   curl --location --request DELETE '{{baseUrl}}/rest/v4/jobs/{uuid}
   ```

KNIME Workflow Services

To make it much easier to call KNIME workflows from other workflows it is possible to use the KNIME Workflow Services set of nodes.

This set of KNIME Labs nodes can be used to build and use workflows as native KNIME services.

One advantage of these nodes is that data being passed between the caller and the callee do not need to be serialized into/from JSON-objects like it is the case when the Container Input/Output nodes are used together with Call (Local) Workflow nodes, making the process more efficient.

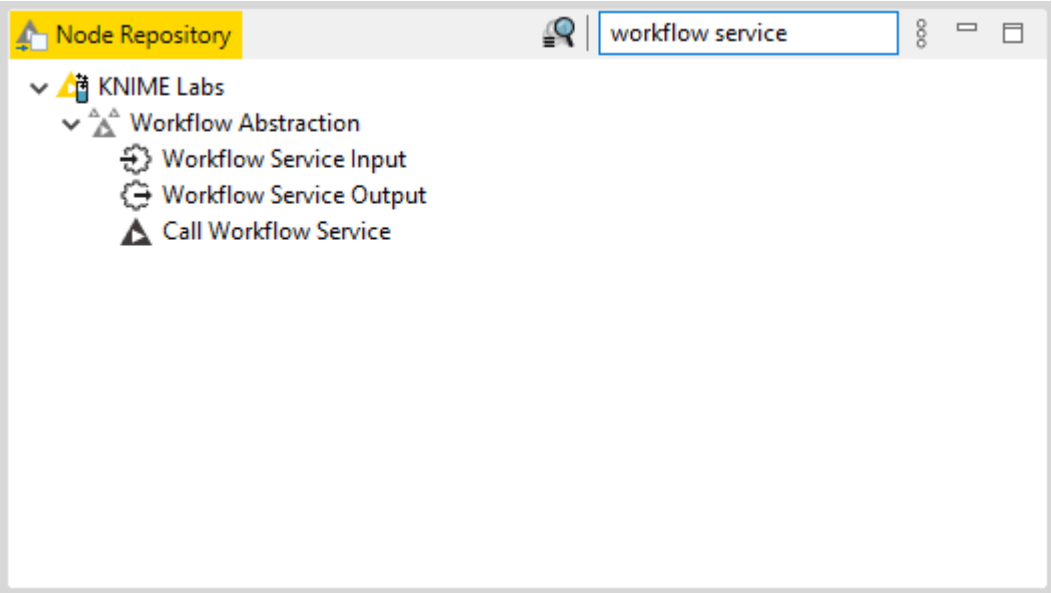
Moreover this new set of nodes makes use of dynamic ports functionality, which allows to use only one node with adjustable port types.

We recommend to use Container Input/Output nodes exclusively in workflows which are intended to be called from external REST clients.

Use the Workflow Service Input/Output nodes instead to build workflows intended to be called from within another workflow (i.e. using Call Workflow Service node).

KNIME Workflow Services nodes

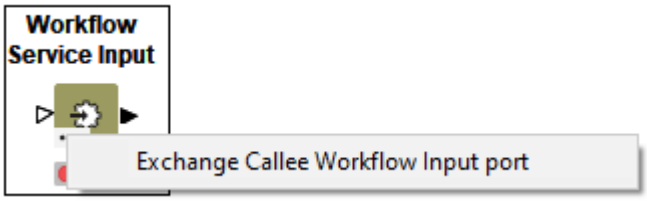
In KNIME Analytics Platform node repository you can find, under *KNIME Labs* > *Workflow Abstraction*, some useful nodes to build workflows that work as native KNIME services.



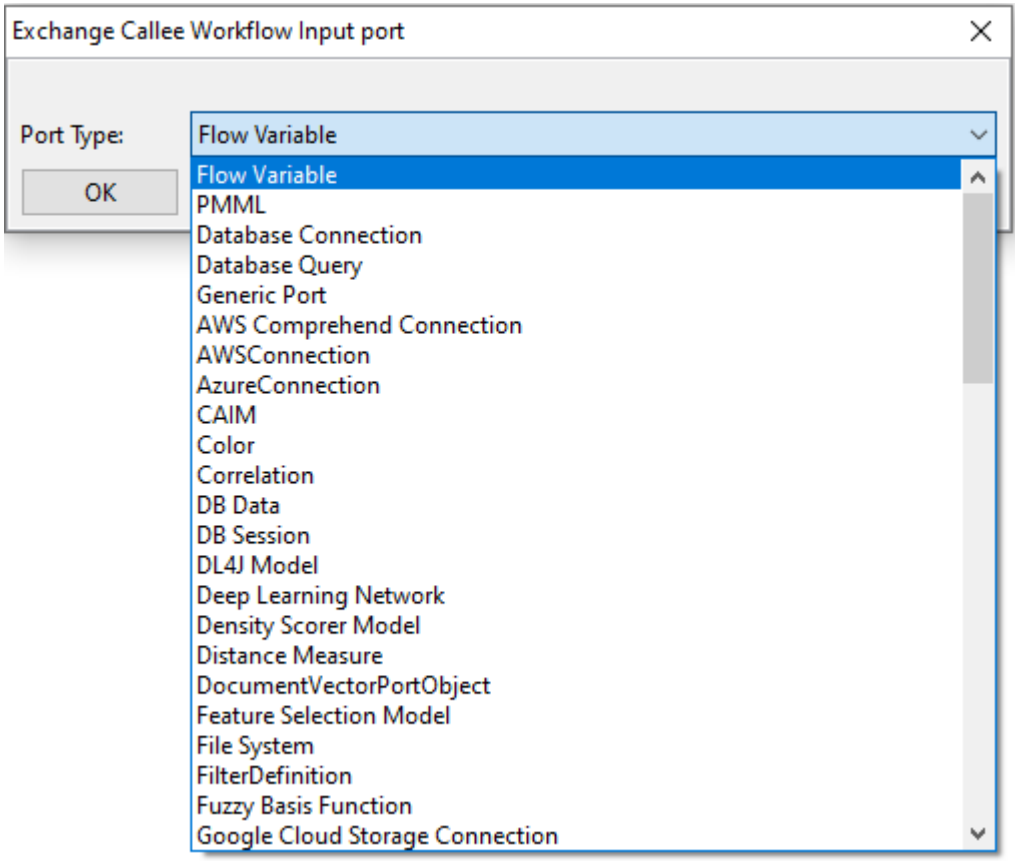
Workflow Service Input/Output nodes

Use these two nodes to receive (Input) or send (Output) a data table or any other port object.

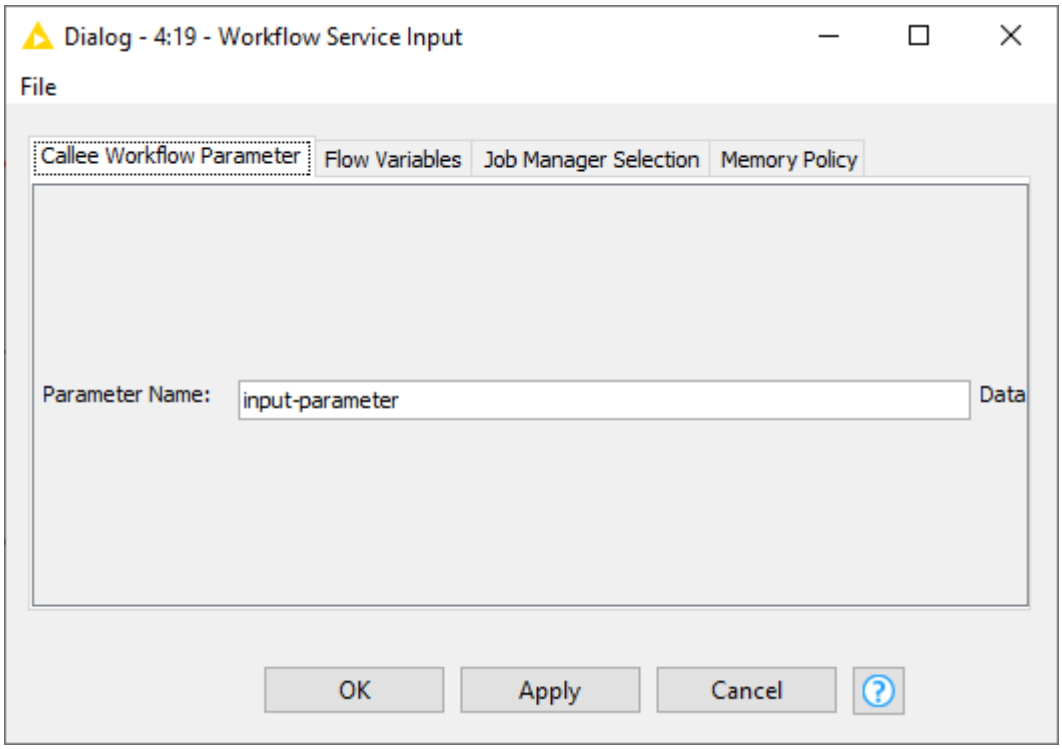
- Workflow Service Input node: This node can receive a data table or any other object from a KNIME workflow that calls this workflow using the Call Workflow Service node. When adding the node to your workflow an optional data table input port and a data table output port are shown. Click the three button dynamic port button on the left bottom corner of the node to *Exchange Callee Workflow Input port* type.



You can then choose between any available port type.

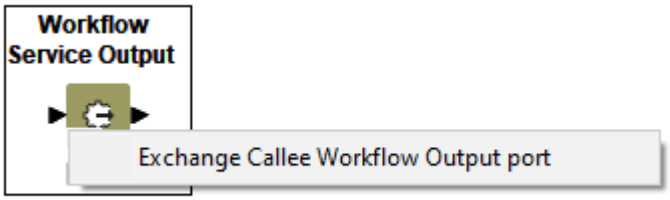


Finally, in the node configuration dialog you can choose a *Parameter name* to assign to the input object.

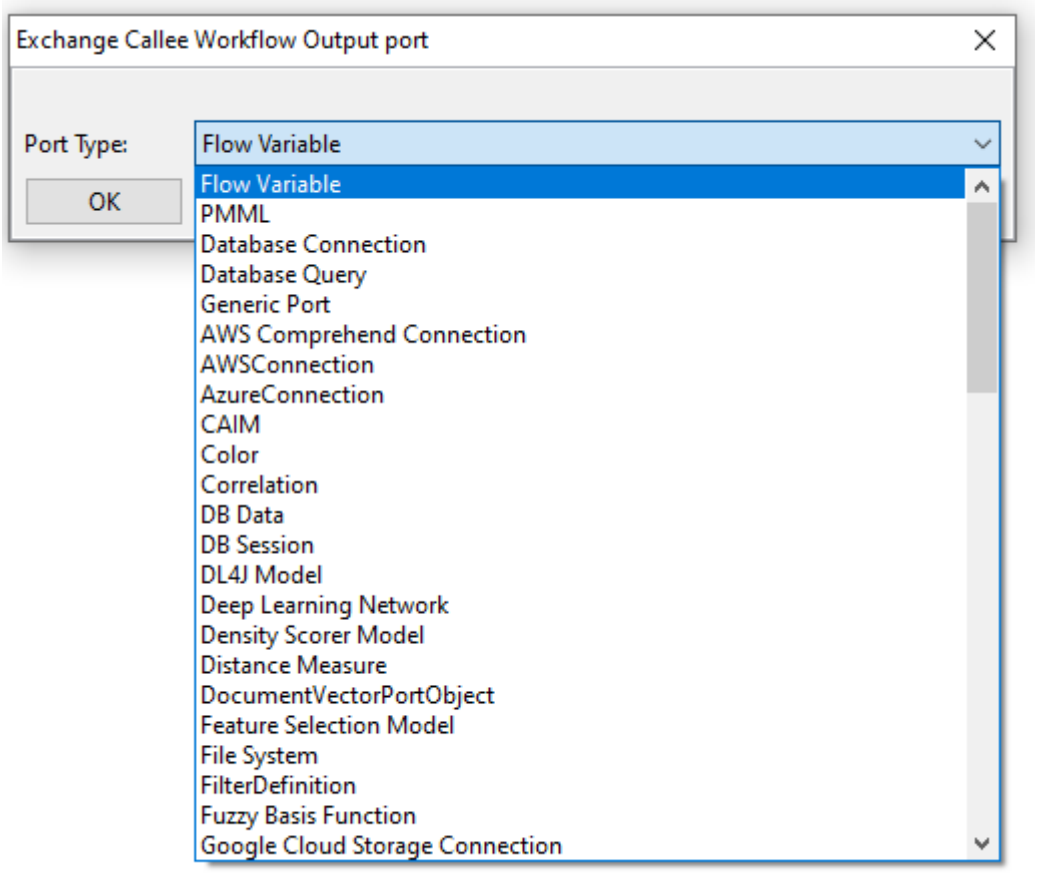


The parameter name is supposed to be unique, but this is not enforced. In case multiple Workflow Service Input nodes define the same parameter name, KNIME will make them unique by appending the node's node ID, e.g., "input-table" becomes "input-table-7".

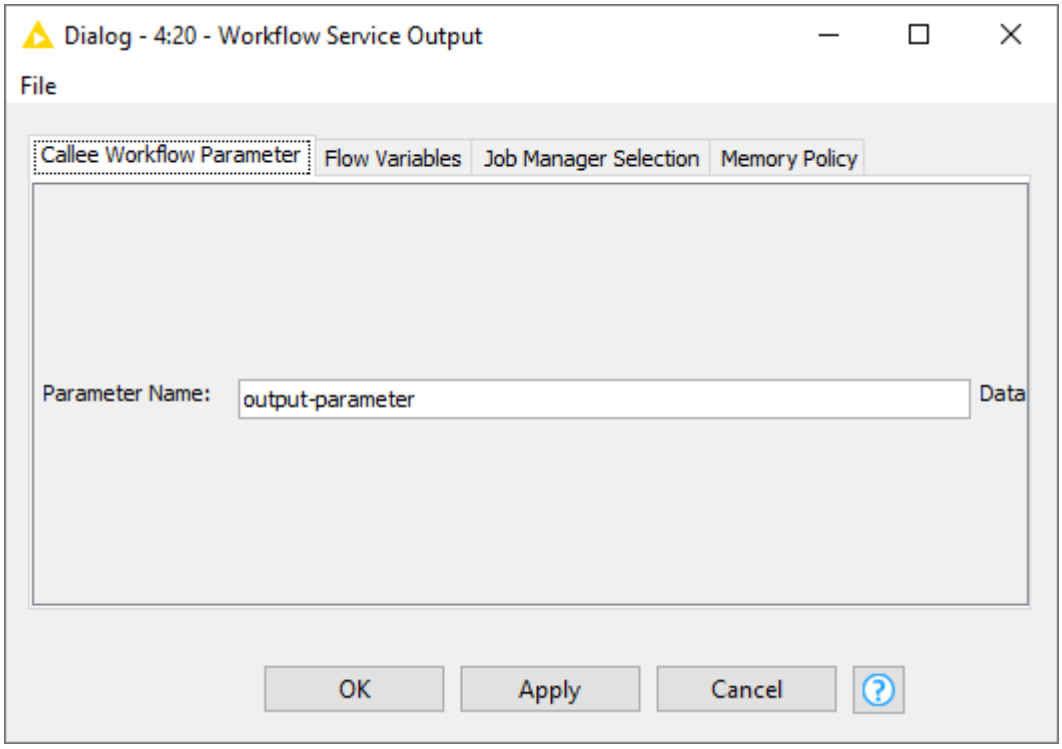
- Workflow Service Output node: This node can send a data table or any other object to a KNIME workflow that executes this workflow using the Call Workflow Service node. When adding the node to your workflow a data table input port and a data table output port are shown. Click the three button dynamic port button on the left bottom corner of the node to *Exchange Callee Workflow Output port* type.



You can then choose between any available port type.



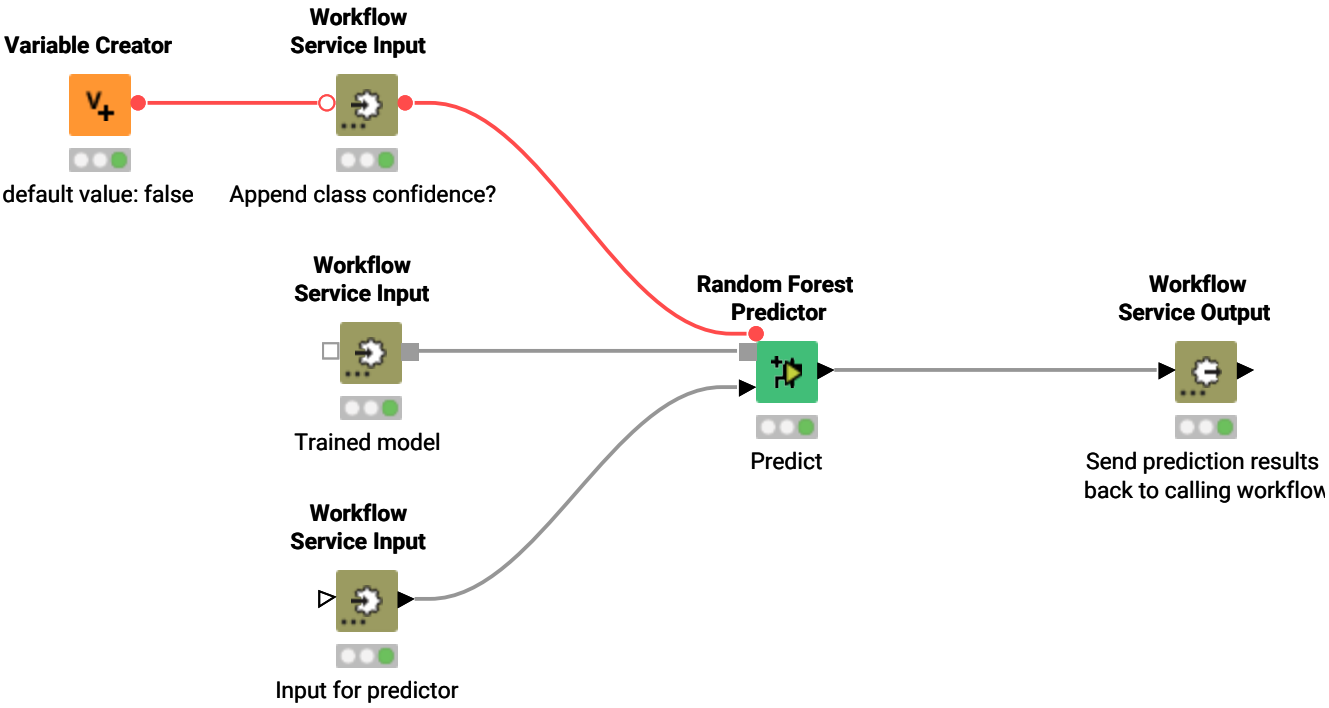
Finally, in the node configuration dialog you can choose a *Parameter name* to assign to the output object.



The parameter name is supposed to be unique, but this is not enforced. In case multiple

Workflow Service Input nodes define the same parameter name, KNIME will make them unique by appending the node's node ID, e.g., "output-table" becomes "output-table-7".

An example of a callee workflow using this set of nodes is available on the KNIME Hub.

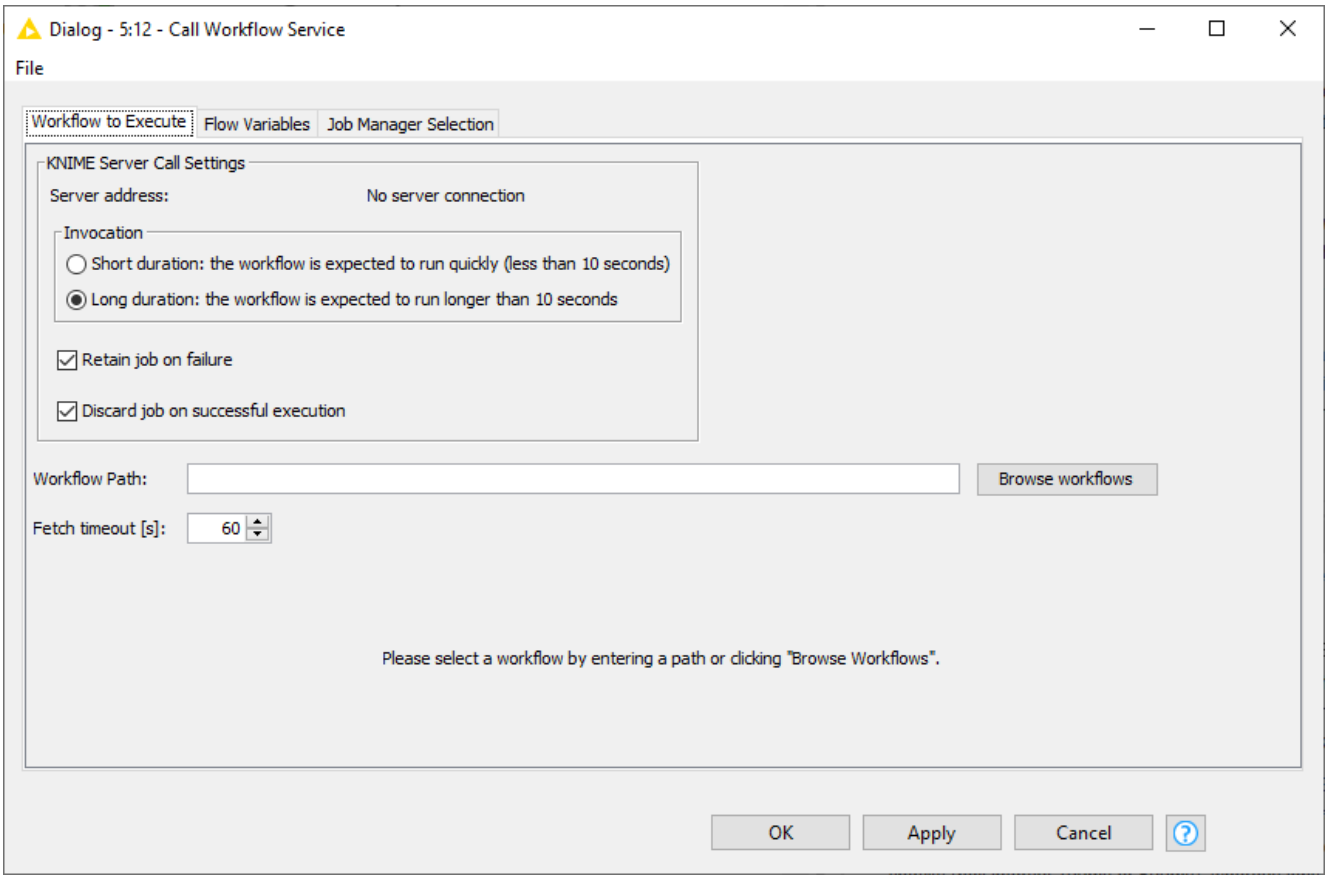


Call Workflow Service node

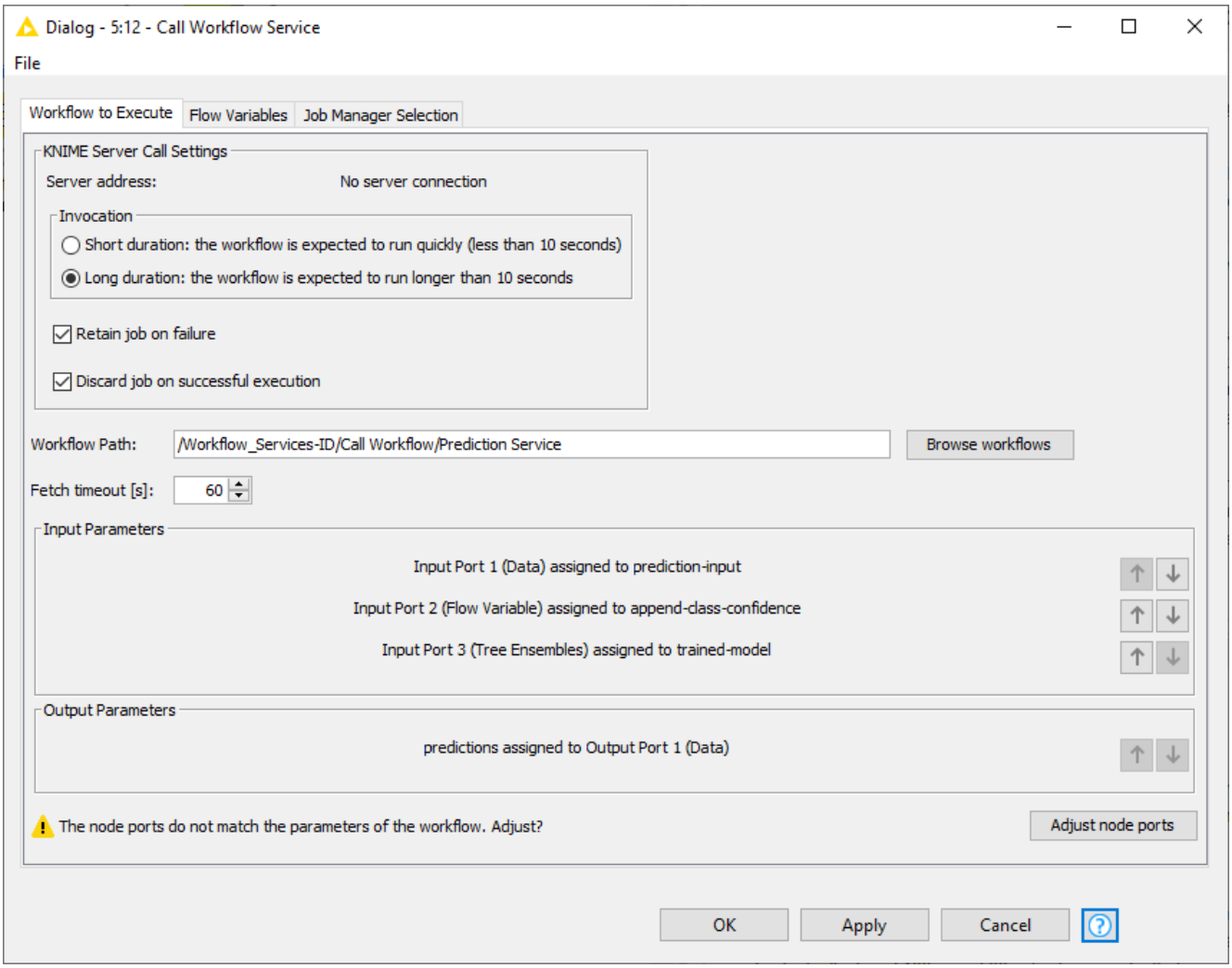
Use this node in a workflow to call another workflow and obtain the results for further processing in the workflow. The workflow will be able to receive inputs via Workflow Service Input nodes and return outputs using Workflow Service Output nodes.

This node comes with an optional KNIME Server connection. Use the KNIME Server Connector node to connect to a KNIME Server in case the callee workflow has been deployed to a KNIME Server installation. In the node configuration dialog you can then set parameters to manage the connection to the KNIME Server. If you want to call a workflow in your local workspace you can just leave this port unconnected.

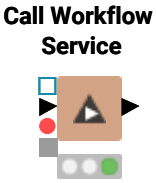
You can insert the *Workflow Path* or *Browse workflows* to choose the callee workflow.



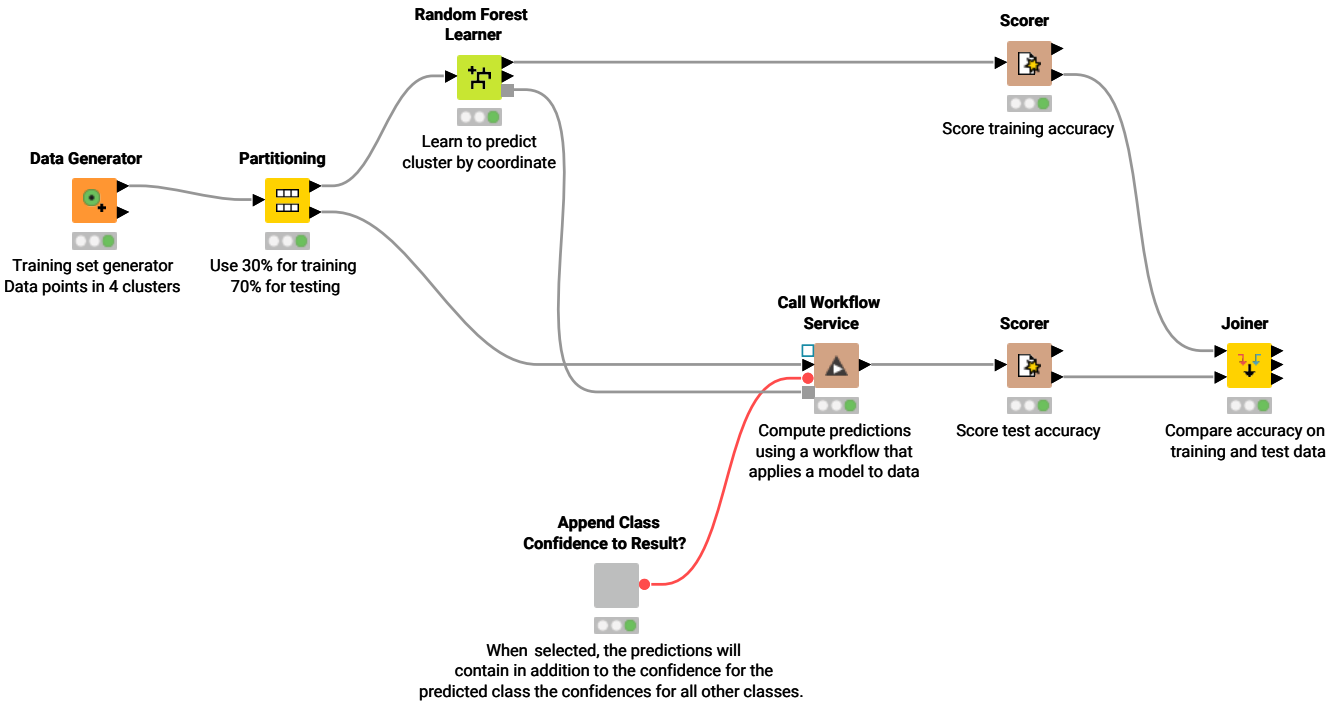
A button *Adjust node ports* in the node configuration dialog will appear in case the node ports do not match the parameters of the callee workflow. Click the button to adjust the node ports.



Each Workflow Service Input node in the workflow to be called will create an input port on this node when finishing the configuration of the Call Workflow Service node. Similarly, each Workflow Service Output node will create an output port on the node.



An example of a call workflow using this set of nodes is available on the KNIME Hub.



KNIME AG
Talacker 50
8001 Zurich, Switzerland
www.knime.com
info@knime.com