

KNIME Server User Guide

KNIME AG, Zurich, Switzerland
Version 4.13 (last updated on 2022-08-19)



Table of Contents

Introduction	1
Further reading	1
Connect to KNIME Server	2
KNIME Explorer	2
Inspect and edit a workflow from KNIME Server	6
Versioning	7
Create a snapshot	7
Server History view	8
KNIME Workflow Comparison	9
Remote execution of workflows	10
Standard remote execution options	10
Actions	11
Scheduling the remote execution	19
Parametrization of Configuration nodes	20
Workflow jobs	21
Execution retries	24
Remote workflow editor	25
Install Remote Workflow Editor Extension	26
Edit remote workflows	26
Remote Workflow Editor preferences	27
Managing access to Server items	29
Roles on KNIME Server	29
Access rights	29
Set up Server items permissions	31
Workflow pinning	35
Prerequisites for workflow pinning	35
Setting <code>executor.requirements</code> property for a workflow	35
Removing <code>executor.requirements</code> property for a workflow	36
Webservice interfaces	37
KNIME Server REST API	37
SwaggerUI for Workflows	38
Workflow REST API nodes	39
Recycle bin	45

Introduction

KNIME Server is the enterprise software for team based collaboration, automation, management, and deployment of data science workflows, data, and guided analytics. Non experts are given access to data science via KNIME WebPortal or can use REST APIs to integrate workflows as analytic services to applications and IoT systems. A full overview is available [here](#).

This guide contains information on how to connect to KNIME Server from KNIME Analytics Platform.



For an overview of use cases, see our [solutions page](#). Presentations at KNIME Summits about usage of the KNIME Server can be found [here](#).

Further reading

If you are looking for detailed explanations around the configuration options for KNIME Server, you can check the [KNIME Server Administration Guide](#).

If you are looking to install KNIME Server, you should first consult the [KNIME Server Installation Guide](#).

For guides on connecting to KNIME Server using KNIME WebPortal please refer to the guides:

- [KNIME WebPortal User Guide](#)
- [KNIME WebPortal Admin Guide](#)

An additional resource is also the [KNIME Server Advanced Setup Guide](#).

Connect to KNIME Server

To connect to KNIME Server from KNIME Analytics Platform you have to first add the relative **mount point** to the **KNIME Explorer**.

KNIME Explorer

The KNIME Explorer, on the left-hand side of KNIME Analytics Platform, is the point of interaction with all the mount points available.

By default the KNIME Explorer displays:

- The LOCAL mount point, which has access to the chosen **workspace**, a folder on your local system
- The **EXAMPLES Server** where a number of example workflows made available by KNIME are stored
- The **KNIME Hub** mount point where you can connect to your profile on the Hub and have access to your Spaces

From here you can also access a KNIME Server repository, by adding the Server mount point.



For a detailed guide to the functionalities of the KNIME Explorer please refer to the **KNIME Explorer section** of the KNIME Workbench Guide.

Setup a mount point

Go to the KNIME Explorer window and click the preferences button in the toolbar as shown in **Figure 1**.



Figure 1. The Preferences button of the KNIME Explorer toolbar

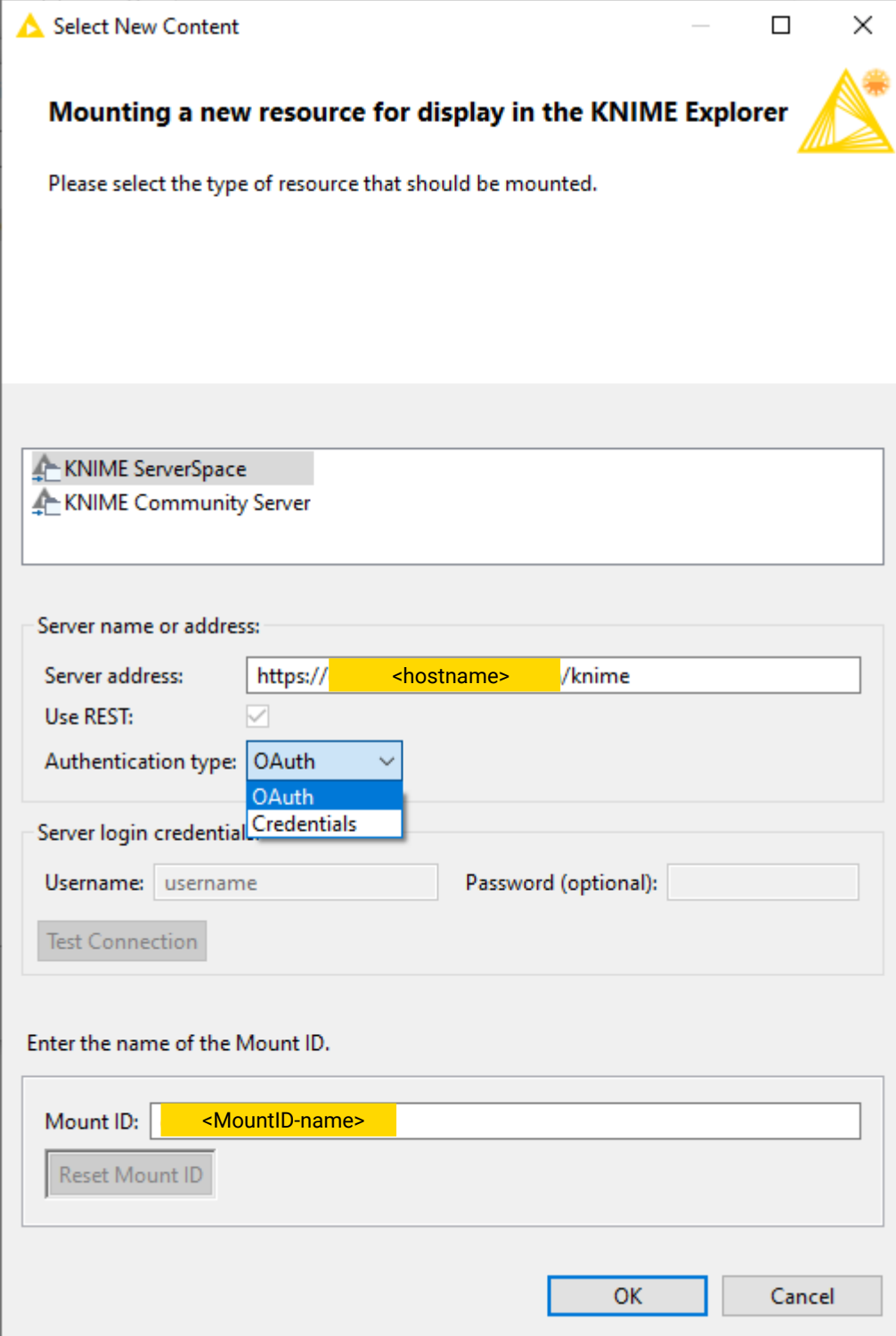


Since KNIME Analytics Platform version 4.2 the KNIME ServerSpace extension is already installed on the client. If needed this can be installed navigating to **File → Install KNIME Extensions...** where you will find it under KNIME Server Extensions or from **KNIME Hub**.

In the preferences window click **New...** to add the new Server mount point. In the **Select New**

Content window, shown in [Figure 2](#), select KNIME ServerSpace and insert the Server address:

https://<hostname>/knime/



Select New Content

Mounting a new resource for display in the KNIME Explorer

Please select the type of resource that should be mounted.

KNIME ServerSpace
KNIME Community Server

Server name or address:

Server address: https://<hostname>/knime/

Use REST: ☒

Authentication type: OAuth

Server login credentials:

Username: username Password (optional):

Test Connection

Enter the name of the Mount ID.

Mount ID: <MountID-name>

Reset Mount ID

OK Cancel

Figure 2. Select New Content window to add a new Server mount point



The KNIME Community Server instead is a Server dedicated to the developer community for test purposes. For more information, please refer to the testing section of the [developers page](#) on the KNIME website.

Choose the desired authentication type. Finally, click the *Mount ID* field where the name of the mount point will be automatically filled, and select *OK*.

Through the Mount ID you can reference the KNIME Server in your workflow. For example you might want to read or write a file from or to a specific location in the KNIME Server mount point. You can do this by using [mount point relative URLs](#) with the new File Handling framework, available for KNIME Analytics Platform version 4.3+, or with the [knime protocol](#). Using these features with KNIME Server requires then the Mount ID to be consistent. For this reason we also do not recommend to change the Mount ID to anything different from the default Mount ID.



The [default Mount ID](#) can be changed by the KNIME Server Administrator.

Click *Apply and Close* and a new mount point will be available in the KNIME Explorer, as shown in [Figure 3](#).

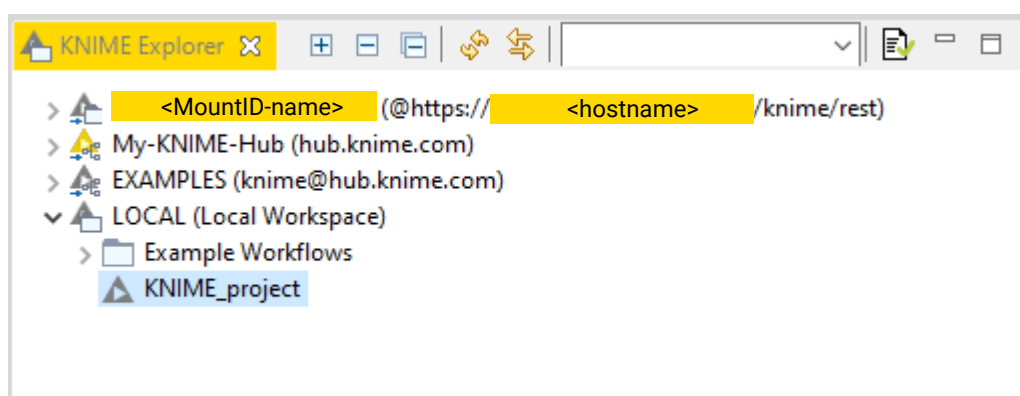


Figure 3. A Server mount point in the KNIME Explorer

Open the Server mount point in the explorer and double-click *Double-click to connect to server*. If you are using [OAuth authentication](#) a browser page will open where you can connect via your identity provider. If you are using Credentials authentication, instead, a login window opens (see [Figure 4](#)) where you can insert your login information and click *OK*.

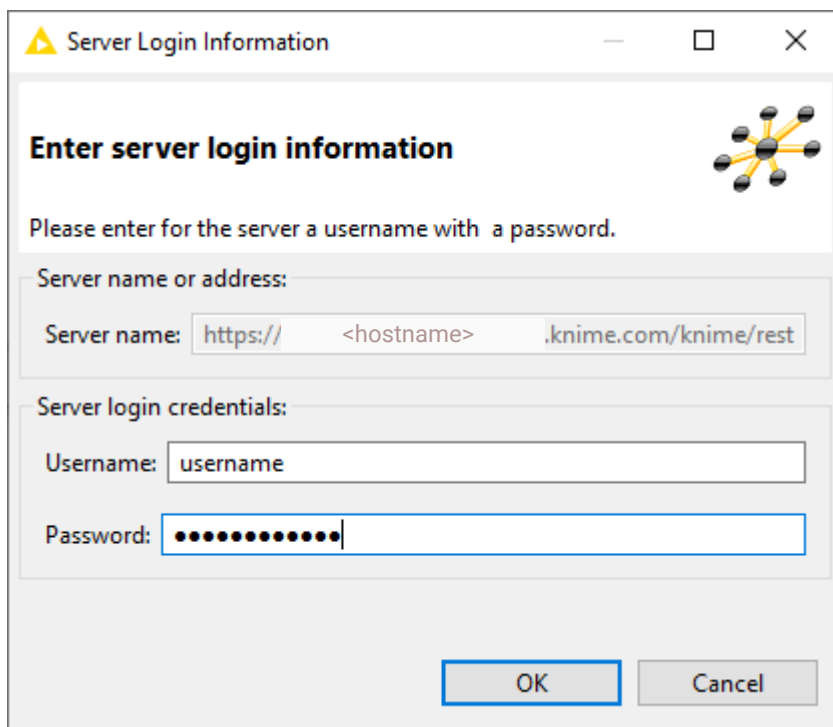


Figure 4. The Server Login information window to connect to KNIME Server using credentials

Store items in the KNIME Server repository

Now, you can use the KNIME Server repository to store your workflows, workflow groups, shared components and metanodes, and data.



For a detailed guide about shared components and metanodes please refer to the [KNIME Components Guide](#).

In order to copy the desired item to the Server repository you can:

- Drag and drop: items can be moved between the repositories in the same way as in any other file explorer, i.e. you can drag an existing item from your LOCAL workspace and drop to the desired location in the Server repository
- Copy/paste: items can be copied from a repository to the Server repository, into the desired location
- Deploy to Server: right-click the item and select *Deploy to Server* from the context menu. A window like the one shown in [Figure 5](#) will open where you can select the desired Server mount point folder where to place the item, given that you are logged in to the respective Server.

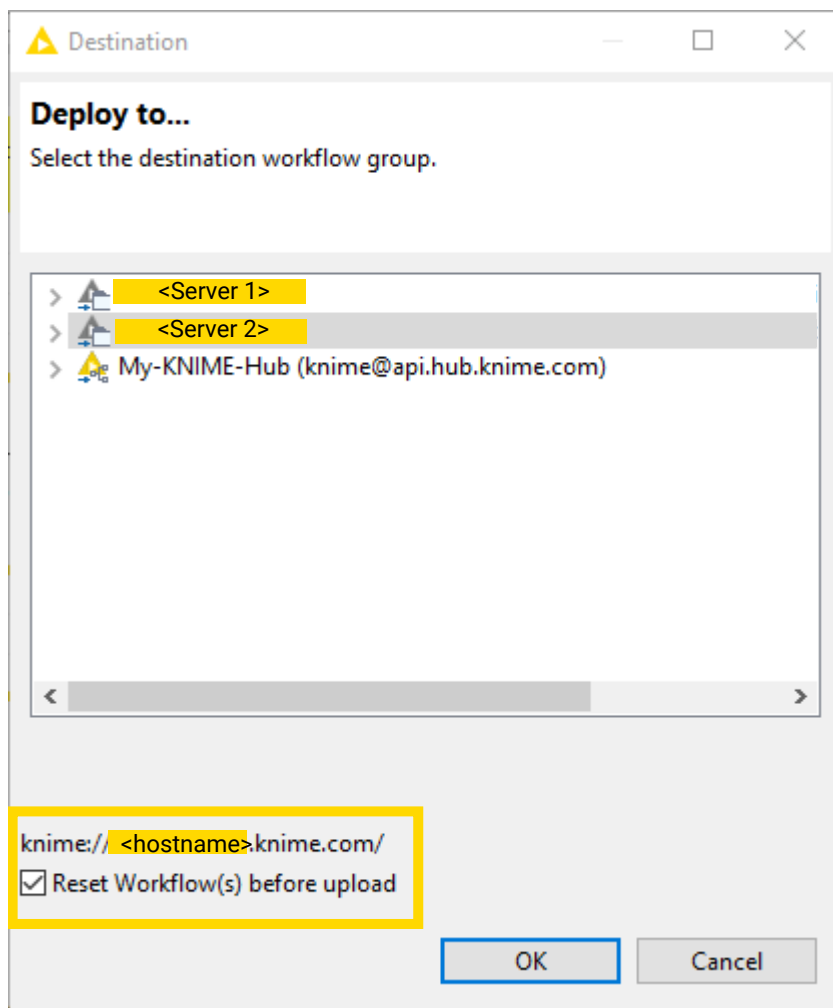


Figure 5. The Deploy to... window

Here you can select to *Reset Workflow(s) before upload* to avoid exposure of sensitive data, e.g. login credentials.

Inspect and edit a workflow from KNIME Server

By double-clicking a workflow on KNIME Server repository, the client downloads it to a temporary location and subsequently opens it automatically. A yellow bar at the top of the workflow editor indicates that this is a temporarily downloaded Server workflow. Now, you can make your changes to the workflow, e.g. you can change the configuration of the nodes, and then you can either Save the workflow to overwrite it to its location on the Server or you can choose the Save as... option to save the workflow locally and store the changed workflow back to a the Server repository. In both cases a **snapshot** of the workflow can be created.

Versioning

It is possible to create a history of items on KNIME Server. To do that, you can create snapshots of workflows, data files and shared components. These are stored with a timestamp and a comment.

Create a snapshot

To manually create a snapshot of an item saved on KNIME Server, right-click the item in the KNIME Explorer and select *Create snapshot* from the context menu.

You can choose to insert an optional comment and a window shows where the location and name of the newly created snapshot is shown.

You can also create a snapshot when overwriting an item on KNIME Server. Check the option *Create snapshot before overwriting?* on the confirmation dialog and insert an optional comment, as shown in [Figure 6](#).

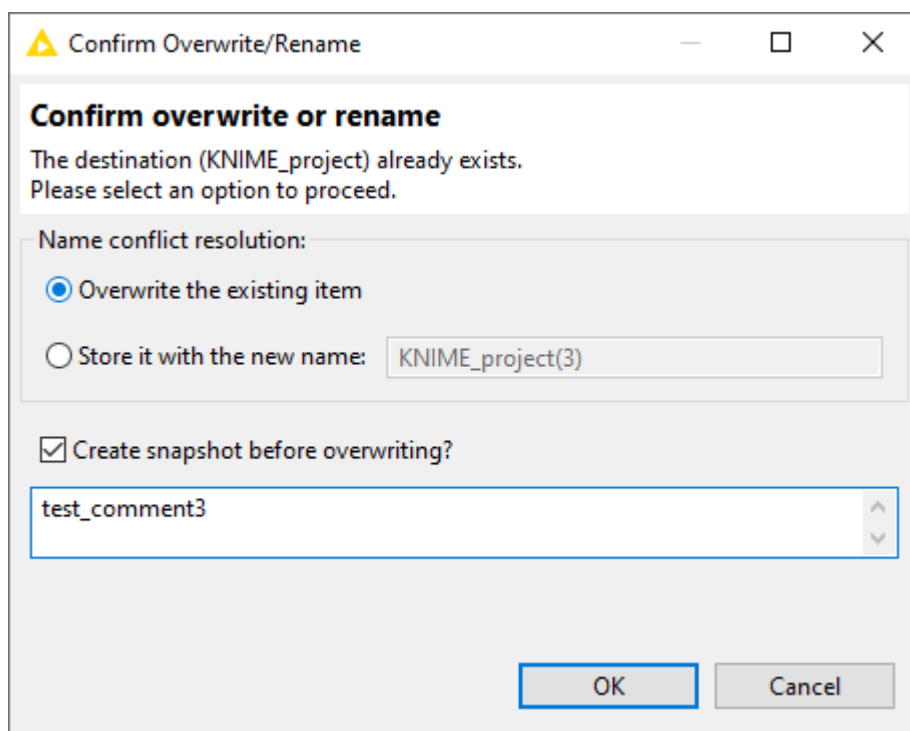


Figure 6. The confirmation dialog



You need write permissions on any item you want to create a snapshot of. It is also not possible to create a snapshot of a workflow group.

When overwriting any Server item with another item, the latest version will be overwritten but its history will not. This means that when you overwrite an item without creating a snapshot,

you only discard the latest version of it.

Note, though, that:

- The history is not available in the local workspace and thus will be lost on downloaded items.
- The history is completely overwritten, if you replace a Server item by dragging and dropping another Server item under the same path. This means, the server-side drag and drop copies and replaces the whole instance including the snapshots.

Server History view

All the snapshots created for an item on KNIME Server are visible in the Server History view. To enable the view go to *View* → *Other...* and, under *KNIME Views*, select *Server History*.

The view will be visible as a tab on the same panel of the Console and Node Monitor views, as shown in [Figure 7](#).

To view all the snapshot of an item select the item in the KNIME Explorer.

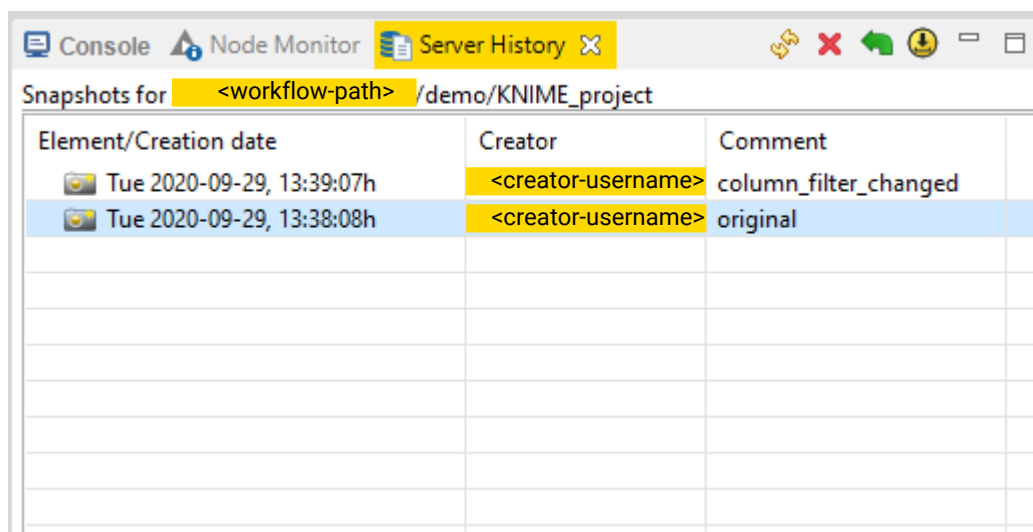


Figure 7. The Server History view showing a created snapshot for a workflow saved on KNIME Server

By right-clicking a snapshot in the Server History view and using the context menu or selecting it and using the buttons on the upper right corner of the view you can:

- Refresh the view to the latest state
- Delete a snapshot
- Replace an item with the selected snapshot, restoring a certain item to a previous state

- Download an item as a workflow, saved in the LOCAL mount point.

KNIME Workflow Comparison

The Workflow Comparison feature provides tools and views that compare workflow structures and node settings. Workflow Comparison, combined with the possibility to create snapshots for your workflows on the Server, allows you to view changes in different versions of a workflow. The feature allows users to spot insertions, deletions, substitutions or similar/combined changes of nodes. The node settings comparison makes it possible to track changes in the configuration of a node.

You can compare not only all items (except data) that can be shown in KNIME Explorer i.e. workflows, components, metanodes, but also snapshots from the Server History view.

Select the items you want to compare, right-click and select *Compare* from the context menu. A Workflow Comparison view will open and you will be able to track differences between the items and between the nodes the two items contain.

An extensive explanation of both [Workflow Comparison](#) and [Node Comparison](#) is given in the KNIME Workbench Guide.

Remote execution of workflows

You can execute the workflows, saved on the Server repository from KNIME Analytics Platform. From KNIME Explorer right-click the workflow you want to execute and select *Execute...* from the context menu. The *Server Execution Options* window open, as shown in [Figure 8](#).

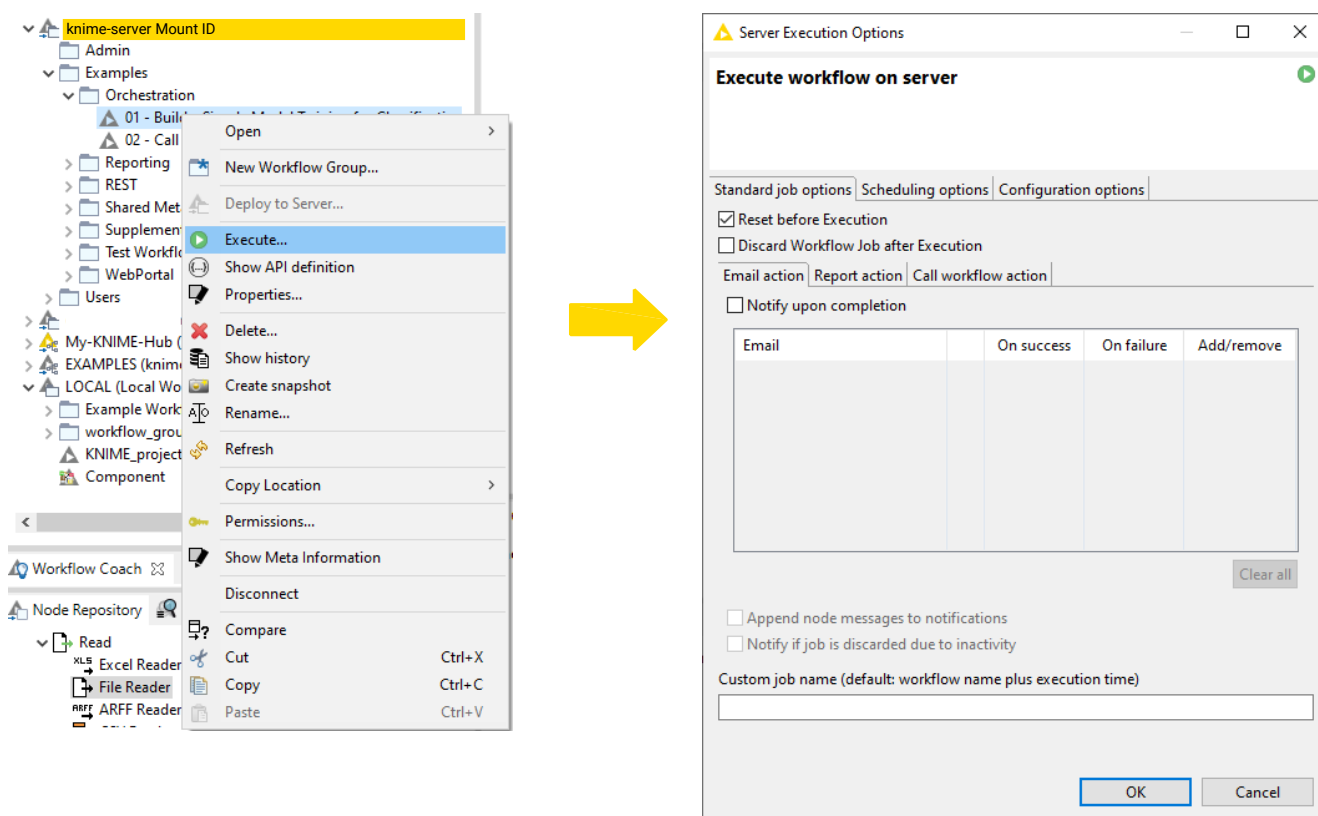


Figure 8. The remote execution of a workflow on KNIME Server

Here you can set some **standard execution behavior** for the job in the top pane and then you have three tabs:

- **Actions**: where you can control basic options of the execution of the workflow on KNIME Server
- **Scheduling options**: where you can schedule the workflow execution
- **Configuration options**: where you can change and control the parametrization of top-level Configuration nodes

Standard remote execution options

Here you can:

- Check *Reset before Execution*: to reset the workflow before execution, meaning that all nodes will be reset to their configured state, including File and DB Reader nodes
- Check *Discard Workflow Job after successful Execution*: to delete the execute job immediately after execution is finished successfully. The job is not saved on the Server.
- Check *Discard Workflow Job after failed Execution*: to delete the execute job immediately after execution is finished unsuccessfully. The job is not saved on the Server.

Actions

The *Actions* tab is shown in [Figure 9](#).

Server Execution Options

Execute workflow on server

☒ Reset before Execution

☐ Discard Workflow Job after successful Execution

☐ Discard Workflow Job after failed Execution

Custom job name (default: workflow name plus execution time)

Actions | Scheduling options | Configuration options

Email action | Report action | Call workflow action

☐ Notify upon completion

Email	On success	On failure	Add/remove
-------	------------	------------	------------

Clear all

☐ Append node messages to notifications

☐ Notify if job is discarded due to inactivity

OK Cancel

Figure 9. The Actions tab for remote execution

Here you have three tabs:

- *Email action* tab: Check *Notify upon completion* to activate the *Email action* tab and set up the email to send out upon completion of the execution (see [Figure 10](#))

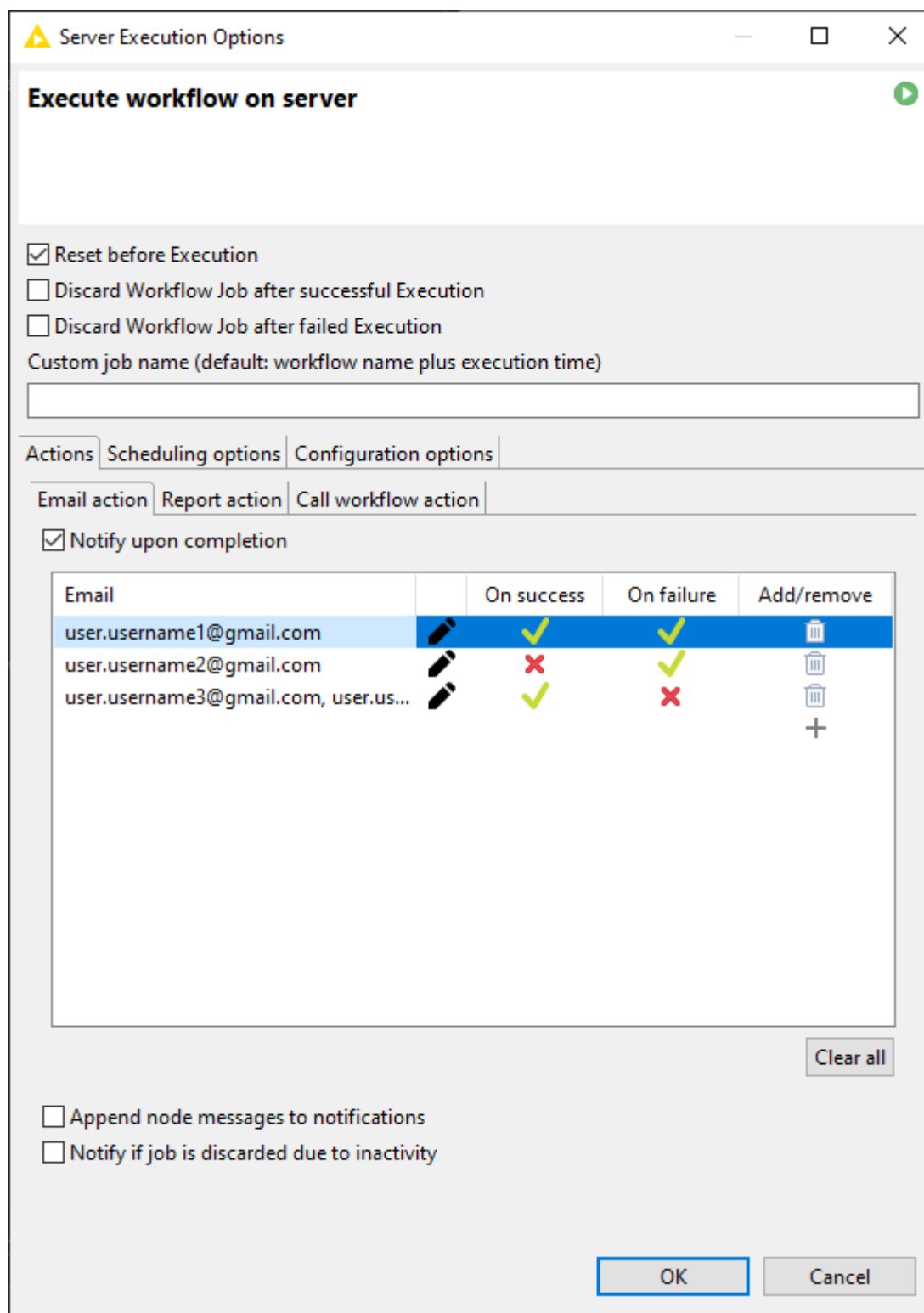


Figure 10. The Email action tab

- Click the plus symbol under *Add/remove* column, then click the pencil symbol to add the email address you want to be notified. You can also insert a list of email addresses by separating them with a comma, a semicolon or a new line.
- Click the green check or the red cross symbol to notify the email address(es) in the corresponding field:

- On success: only if the job is completed successfully
- On failure: only if the job is not completed successfully
- Check *Append node messages to notifications* and/or *Notify if job is discarded due to inactivity* to add information to the email notifications
- *Report action* tab: if your workflow is also creating a report, you can check *Save report on server* (see [Figure 11](#))

Server Execution Options

Execute workflow on server

☒ Reset before Execution

☐ Discard Workflow Job after successful Execution

☐ Discard Workflow Job after failed Execution

Custom job name (default: workflow name plus execution time)

Actions | **Scheduling options** | Configuration options

Email action | **Report action** | Call workflow action

☒ Save report on server

Custom report file name (default: workflow name)

☐ Append "Created At" time

☐ Overwrite

Report location

/ Select destination

Report formats to save

☒ pdf ☐ docx ☐ html ☐ pptx ☐ xlsx

OK Cancel

Figure 11. The Report action tab

Here you can choose:

- A *Custom report file name* to rename the report file saved on the Server
 - Append: to save the report file with a suffix indicating the time of creation of the file

- **Overwrite:** to overwrite the report file in case an existing one with the same name is present
- Insert the location where you want to save your report file by manually writing the path relative to the workflow location on the Server in the field under *Report location* or by clicking *Select destination* to navigate through the Server repository mount point
- Finally, you can choose which format(s) you want to use for your report
- *Call workflow action* tab: if you want to execute another workflow on the Server as a consequence of success or failure of the current workflow (see [Figure 12](#)). Check *Call workflows upon completion* and the field is activated.

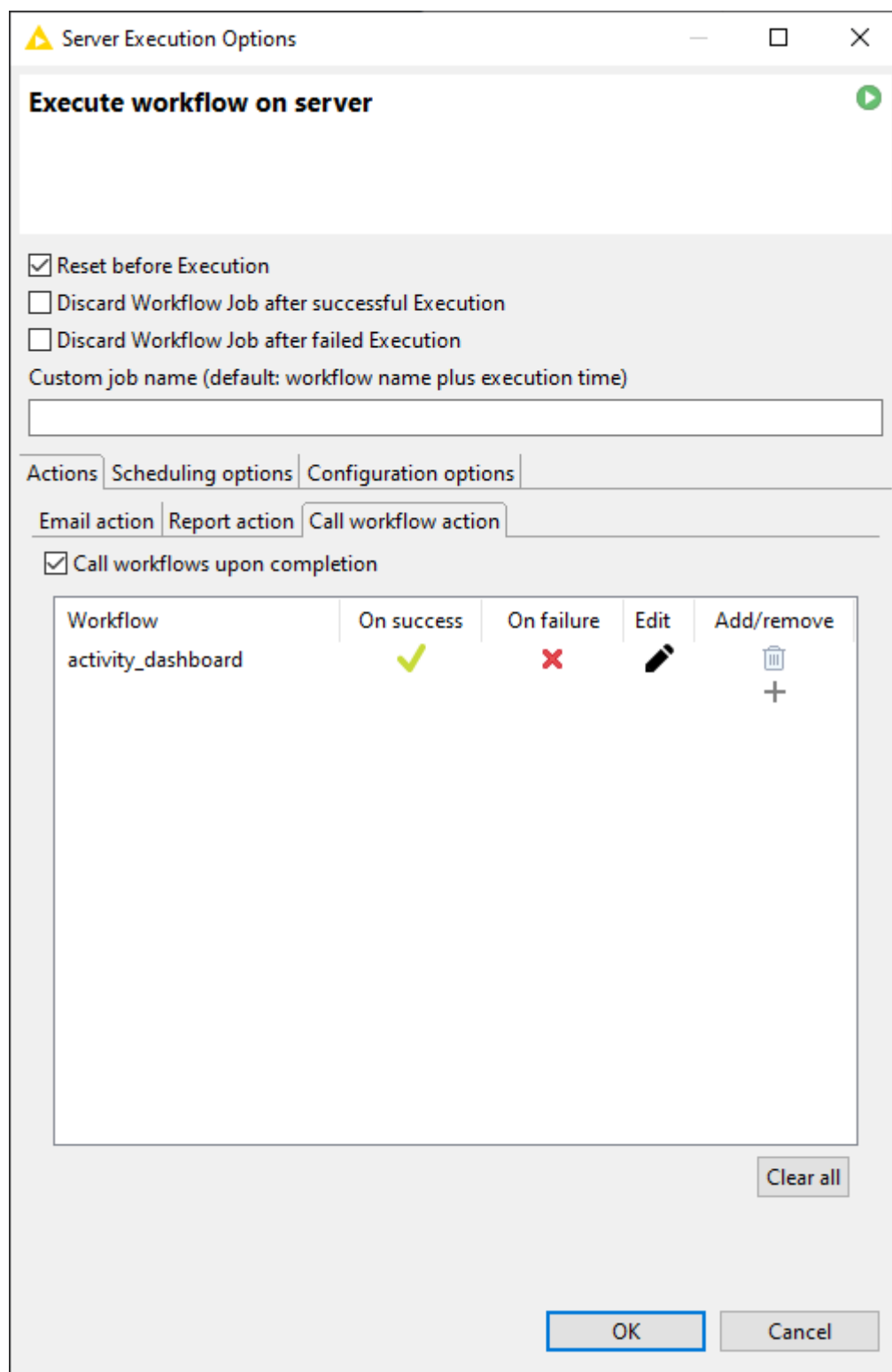


Figure 12. The Call workflow action tab

Here you can:

- Add or remove a workflow: to do this click the plus symbol under *Add/remove* column and the window shown in [Figure 13](#).

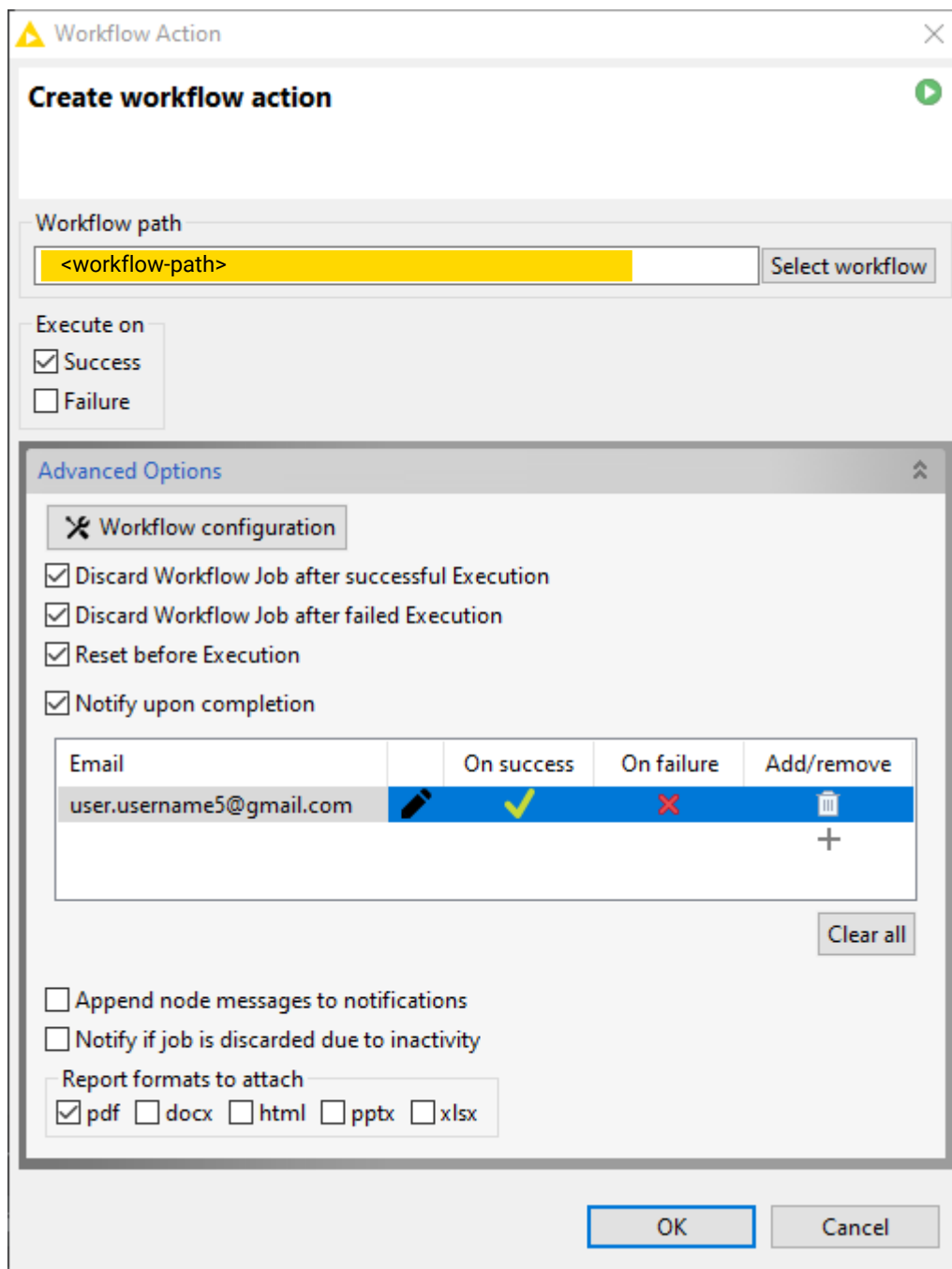


Figure 13. The Advanced Options in the Call workflow action tab

When selecting the called workflow you can choose to execute it on *Success* or on *Failure*. Click *Advanced Options* to choose execution options for the called workflow, e.g. discard or save called workflow job after execution, email notification or report attachment, if available. Also for the called workflow is possible to choose parameters to fill top-level Configuration nodes by clicking *Workflow configuration*. For more information on this functionality please refer to [Parametrization of Configuration nodes](#). Click *OK* to apply the selected options.

- Finally, at the bottom of the window, you can give the job a custom name. The default name is the same as the workflow name with a suffix of the execution time.

Scheduling the remote execution

In the *Scheduling options* tab, shown in [Figure 14](#), you can check the option *Schedule job* to choose the day and time of the *First execution* and then choose either to repeat the execution every n minutes, hours, or days, or to repeat it at fixed start times.

Server Execution Options

Execute workflow on server

☒ Reset before Execution

☐ Discard Workflow Job after successful Execution

☐ Discard Workflow Job after failed Execution

Custom job name (default: workflow name plus execution time)

Actions | **Scheduling options** | Configuration options

☒ Schedule job

First execution 12/11/2020 15:33

Last execution 12/11/2020 15:33

☒ Repeat execution

☒ Repeat every 1 hours

☐ Repeat at fixed start times

Restrict months, days, and times for execution

Days of week | Days of month | Months | Time frames

☒ Monday ☒ Tuesday ☒ Wednesday

☒ Thursday ☒ Friday ☒ Saturday

☒ Sunday

☐ Skip execution if previous job is still running

☐ Disable schedule

Next execution at Thu 11/12/20 4:33 PM

OK Cancel

Figure 14. The Scheduling options tab

You can also choose to execute the job for a last execution day and time. In the second section you can choose to restrict the execution at certain *Days of week*, *Days of month*, *Months*. Here you can also choose *Time frames* during which your job will be allowed to run.

Finally, you can choose to skip the scheduled execution in case the previous job is still running or disable the schedule for the next execution. At the bottom of the window the next execution day and time according to your settings, is shown.

Parametrization of Configuration nodes

Configuration nodes are usually used inside **KNIME Components**, which are like KNIME nodes and are made by wrapping some nodes to perform a specific more complex task. In components, Configuration nodes are used to expose externally some of the parameters that can be configured via the component configuration dialog.

However, Configuration nodes can also be inserted in the root of a workflow, i.e. not as part of a component, in order to perform a number of operations that can be pre-configured with certain parameters. This can be done by choosing the default value in the configuration dialog of this type of nodes. Moreover, the parameters of the top-level Configuration nodes can also be set in the *Server Execution Options* window, from the *Configuration options* tab, shown in **Figure 15**. Here, you can choose to restore all the Configuration nodes to their default parameters, and you can choose the parameters for the available Configuration nodes.

Server Execution Options

Execute workflow on server

☒ Reset before Execution

☐ Discard Workflow Job after successful Execution

☐ Discard Workflow Job after failed Execution

Custom job name (default: workflow name plus execution time)

Actions | **Scheduling options** | Configuration options

Here you can change the values of the top level configuration nodes for this job. If no values are changed in a field, the job will be executed using the respective default settings
Note: For arrays each line counts as a value.

Restore defaults

Test integer input

0

Test double input

0

Test string input

default input

Test credentials input

Username

Password

OK Cancel

Figure 15. The Configuration options tab for the execution of a workflow with different top-level Configuration nodes

Workflow jobs

Remotely executed workflows are run as jobs. A workflow job is a copy of the workflow with specific settings and input data. Jobs are tied to the version from when the job was created,

and they are shown in the KNIME Explorer nested under the workflow and can have different status. A workflow can be executed multiple times, creating multiple workflow jobs.

Workflow job status

The workflow job icon in the explorer can have different status, as shown in [Figure 16](#):

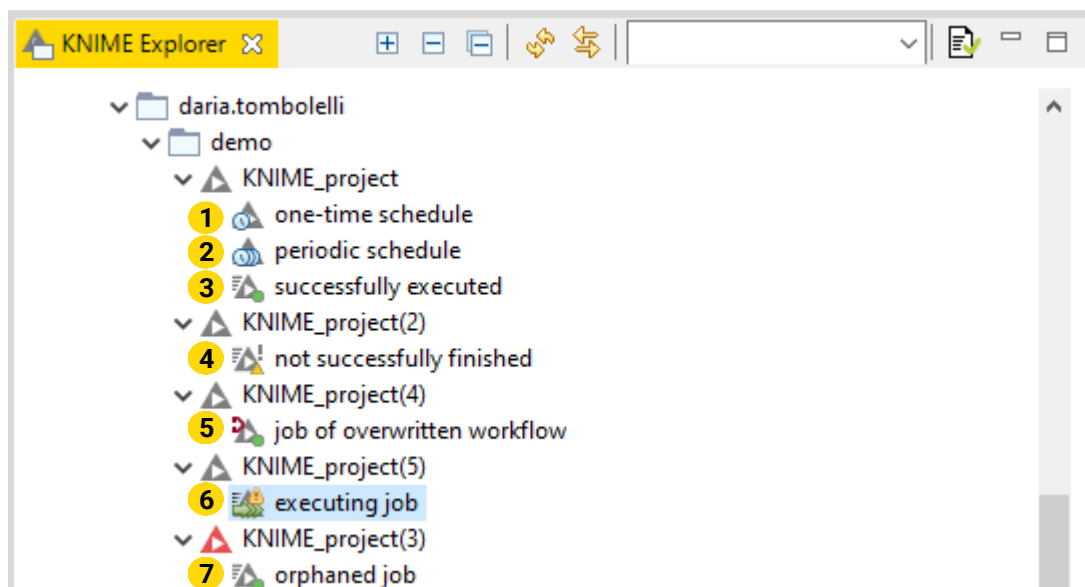


Figure 16. The workflow job status

1. One-time scheduled: the job is scheduled for a one-time execution
2. Periodic scheduled: the job is scheduled for periodic execution

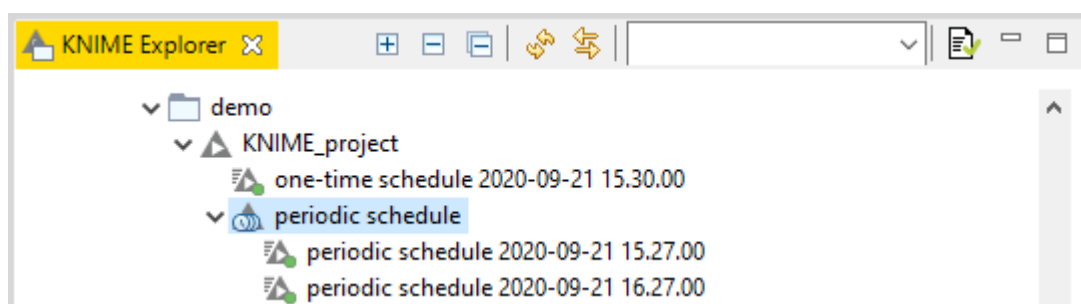


Figure 17. The scheduled jobs run in the KNIME Explorer

3. Successfully executed: the job is finished and the execution was successful
4. Unsuccessfully finished: the job is not successfully finished, meaning that something in the configuration of some node is not correct. An example could be that you have a workflow where a Breakpoint node is configured to throw an error in case an empty table is passed to it. In this case the error message can be visualized by right-clicking the job in the explorer and selecting *Show workflow messages* from the context menu. A window pops up where the error thrown by the corresponding node is shown. An example is shown in [Figure 18](#)

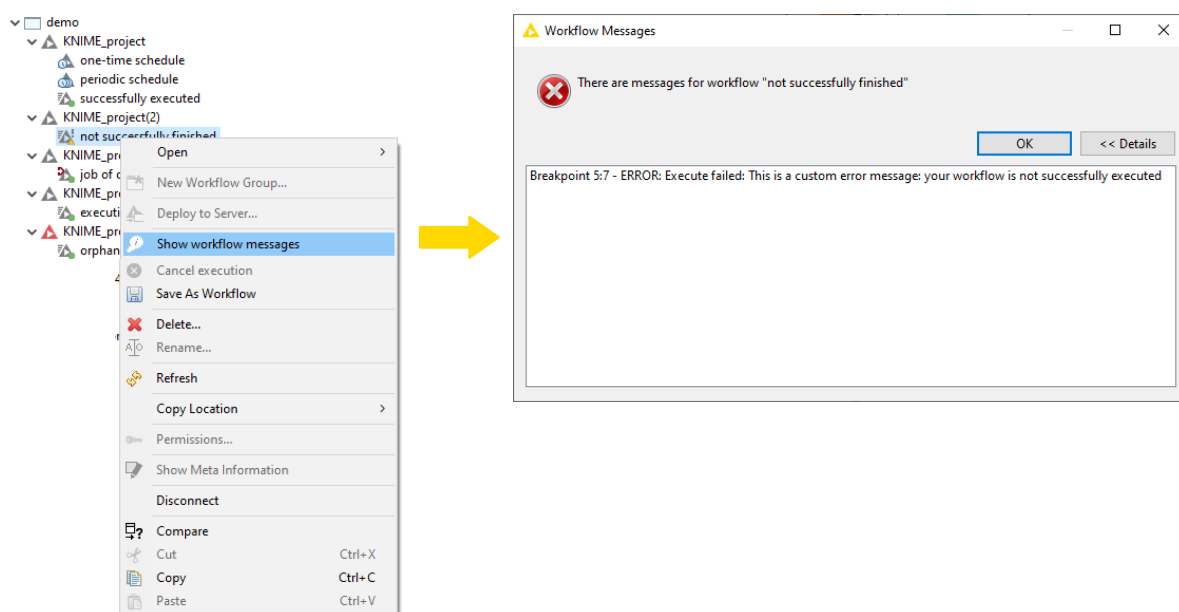


Figure 18. An error message can be visualized in case of not successfully finished jobs

5. Overwritten workflow: the workflow from which the job was initiated has been overwritten and the job relates to the previous version
6. Currently executing: the job is currently executing on KNIME Server
7. Orphaned job: the workflow from which the job was initiated has been deleted but its related job is kept in memory.

Jobs context menu

Workflow jobs stay in the main memory of the Server after execution (unless you checked *Discard Workflow Job after Execution*) until they reach their maximum time they are **automatically discarded** or you remove them manually.

Right-click the job icon in the KNIME Explorer to open the context menu, with the following options:

- Show workflow messages: as explained in the previous section this option is available in case of **a job not successfully finished** to visualize the error message
- Cancel execution: to stop the execution of a job
- Save As Workflow: to save the job as workflow for data provenance and debugging
- Show/Edit Schedule...: this option is only available in case of **scheduled jobs**, to edit or visualize their schedule settings.

The result of the workflow job can currently be inspected:

- By right-clicking the job in the KNIME Explorer and selecting *Open* → *Job* from the

context menu. This option is available only upon installation of the [Remote workflow editor](#).

- By saving it as workflow in the repository and then opening it on the KNIME Analytics Platform (double-click the job saved as workflow or right-click and select *Open → _as Local Copy_*)
- By looking at the resulting views, if any, on [KNIME WebPortal](#).

If the Server is not configured to run multiple KNIME Executor instances (see [KNIME Server Installation Guide](#)), a workflow is executed on the Server under the same user the Server was started with. If it writes to a file, this user must have write permissions at the destination location. If the workflow submits jobs into a cluster they are submitted by this user.

Execution retries

For scheduled jobs it is also possible to define the number of execution retries in case a job fails. The following points have to be considered when using this feature:

- A retry will create and execute a new job with a new ID
- The retried job will contain the ID of the previously failed job as `parentId`
- Each one of the retried jobs triggers all defined post execution actions of the schedule (e.g. Email action, Call Workflow action)
- Failed jobs with retries won't get discarded automatically if `Discard Workflow Job after failed Execution` isn't enabled
- If `Skip execution if previous job is still running` is enabled execution of the schedule will also be skipped if there is still a retry in progress, even if the retry is still waiting to be executed (see `com.knime.server.job.execution_retry_waiting_time` in the [KNIME Server Administration Guide](#)).
- The number of failures (`numFailures`) of the scheduled job will only be increased if all retries fail
- A retry will be also triggered if the original job fails to load

Remote workflow editor

The KNIME Remote Workflow Editor enables users to investigate the status of jobs and edit them directly on the Server. Whenever a workflow is executed on the KNIME Server, it is represented as a job on the Server. This instance of your workflow will be executed on the KNIME Server, which can be helpful in cases where the Server hardware is more powerful than your local hardware, the network connection to external resources such as databases is faster, and does not require traversing firewalls/proxies.

The Remote Workflow Editor looks just like your local workflow editor, apart from the fact that it is labelled and the canvas has a watermark to help identify that the workflow is running on the KNIME Server, as shown in [Figure 19](#).

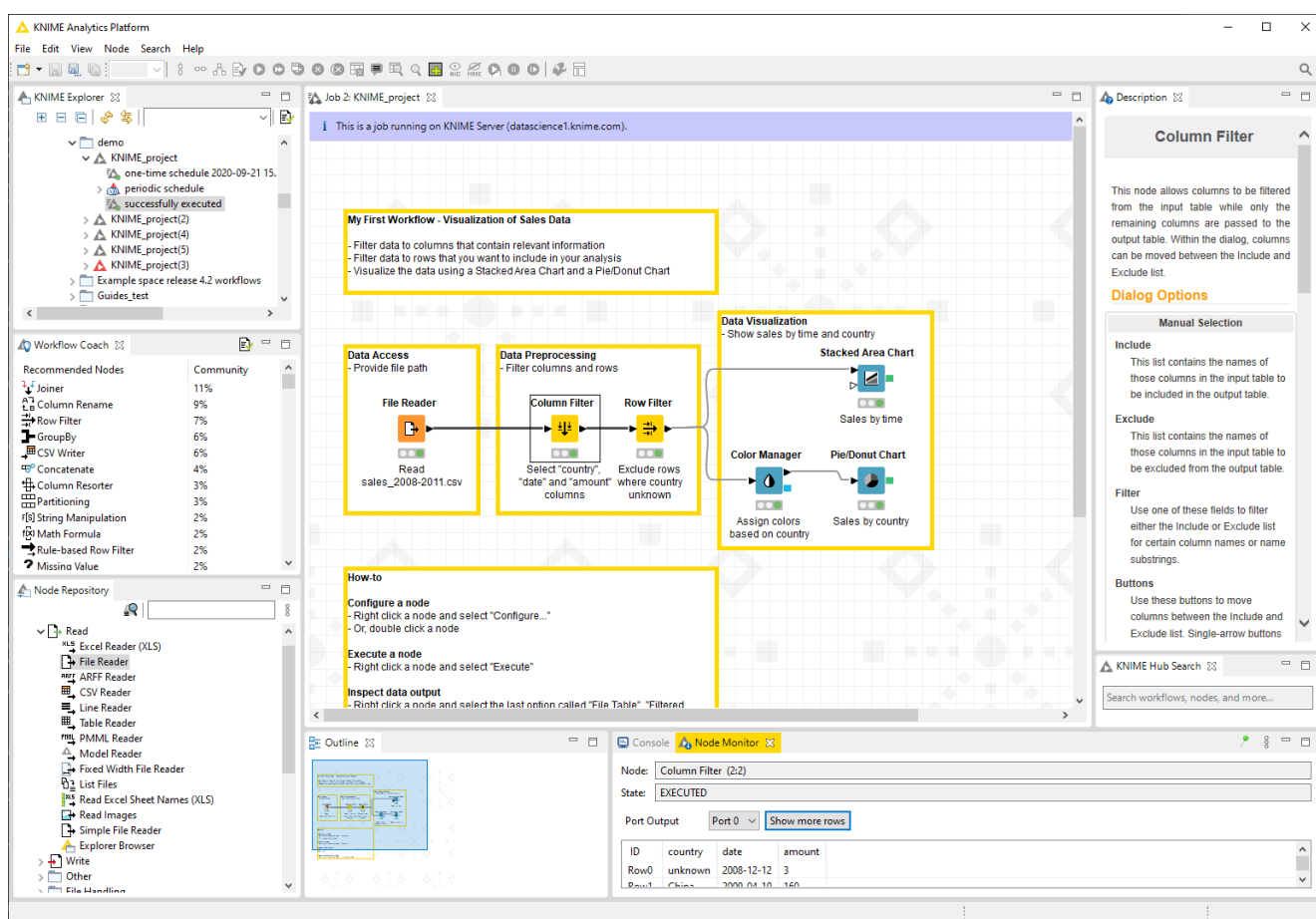


Figure 19. A workflow job on KNIME Server opened from KNIME Analytics Platform via the Remote Workflow Editor

Most of the edit functionality that you would expect from editing a workflow locally on your machine is possible.

Notable cases that are not yet supported are:

- Creating and expanding metanodes/components.

- Adding a component via drag and drop from KNIME Hub.
- Performing loop actions (resume, pause, step, select loop body...).
- Editing and creating node and workflow annotations.
- File drag'n'drop that results in a reader node.
- Copying nodes from a local workflow to a remote workflow (and vice-versa).
- Browse dialog for file reader/writer nodes browses the local filesystem rather than the remote filesystem.

Install Remote Workflow Editor Extension

The Remote Workflow Editor feature needs to be installed in the KNIME Analytics Platform. You can drag and drop the extension from [KNIME Hub](#) or go to *File → Install KNIME Extensions*, and select KNIME Remote Workflow Editor under KNIME Server Extensions.

For detailed instructions on how to install the Remote Workflow Editor on the KNIME Server please refer to the [KNIME Server Administration Guide](#).

Edit remote workflows

It is possible to create and open a new job from a workflow saved on KNIME Server. From KNIME Analytics Platform, go to the KNIME Explorer, right-click on the workflow saved on the KNIME Server mount point and select *Open → as new Job on Server* from the context menu.

Jobs that are already created, a scheduled job, or a job started in the WebPortal, can be visualized by right-clicking the job and selecting *Open → Job* from the context menu.

All jobs, executed from KNIME Analytics Platform, or via KNIME WebPortal, can be inspected. You can see node execution progress, number of rows/columns generated, any warning/error messages, and view the configuration settings of each node.

For jobs that are executed on Server from KNIME Analytics Platform it is also possible to:

- Edit the configuration settings, of nodes as you would if the workflow was on your local workspace



Currently the configuration of some file paths is not supported in some file reader nodes.

- Move, add, remove connections and nodes from the [workbench](#).

- View data via the normal data view and views by using the **View nodes**
- See which nodes are currently executing, which are already executed, and which are queued to be the next in execution
- Check errors and warning in the workflow by mouse-over on the respective sign.

Remote Workflow Editor preferences

To open the Remote Workflow Editor preferences go to *File* → *Preferences* and select *KNIME* → *KNIME GUI* → *Remote Workflow Editor* from the left pane, as shown in **Figure 20**.

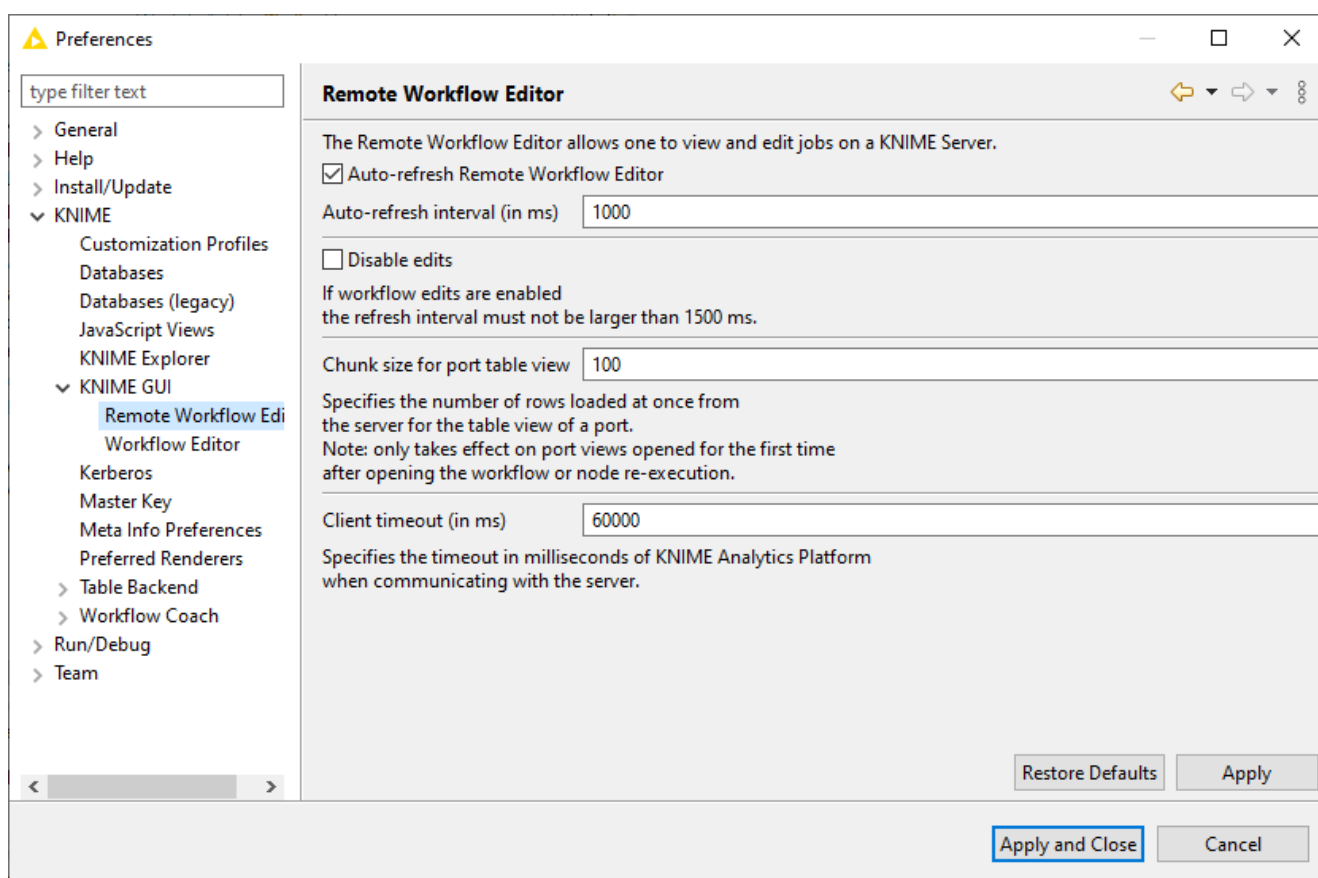


Figure 20. The Remote Workflow Editor preferences window

Here you can set:

- The auto-refresh interval in milliseconds
- Check *Disable edits* to enforce view only mode for all workflows
- The *Chunk size for port table view* defining the number of rows loaded at once from the Server for the table view of a port
- The *Client timeout* in milliseconds to specify the timeout of KNIME Analytics Platform when communicating with KNIME Server.

You can also change the maximum for the refresh interval in milliseconds by adding the option `-Dorg.knime.ui.workflow_editor.timeout` to the `knime.ini` file. Please notice that if workflow edits are enabled the refresh interval must not be set to a value larger than the value specified by this option.

Managing access to Server items

You can assign access permissions to each item saved on the Server like files, workflows or workflow groups, and components to control the access of other users to your saved items.

Roles on KNIME Server

The owner

The Server stores the owner of each Server item, which is the user that created the item. When you upload a workflow, copy a workflow, save a workflow job or create a new workflow group you are assigned to the new item as owner. When a new Server item is created, you can set the permissions how you want this item to be available to other users. Only the owner can change permissions on an item.

User groups

When the KNIME Server administrator defines the users that have access to the KNIME Server, the users are assigned to groups. Groups can be defined as needed, for example one group per department, or per research group, etc. Each user must be in at least one group, and could be in many groups.

Server administrator

Server administrators are not restricted by any access permissions. Administrators always have the right to perform any action usually controlled by user access rights. They can always change the owner of an item, change the permissions of an item, they see all workflow jobs (while as a regular user you can only see your own jobs) and they can delete all jobs and items.

Access rights

There are two different access rights that control access to a workflow group and three access rights to a workflow.

Workflow group permissions

Read	Allows the user to see the content of the workflow group. All workflows and subgroups contained are shown in the repository view.
Write	If granted, the user can create new items in this workflow group. He can create new subgroups and can store new workflows or shared components in the group. Also deletion of the group is permitted.



In order to access a workflow it is not necessary to have read-permissions in the workflow group the flow is contained in. Only the listing of contained flows is controlled by the read-right. Also, a flow can be deleted without write permission in a group (if the corresponding permission on the flow is granted). Also, in order to add a flow to a certain group, you only need permissions to write to that particular group, not to any parent group.

Workflow permissions

Execute	Allows the user to execute the flow, to create a workflow job from it. It does not include the right to download that job, or even store the job after it finishes (storing requires the right to download).
Write	If granted, the user can overwrite and delete the workflow.
Read	Allows the user to download the workflow (including all data stored in the flow) to its local desktop repository and inspect the flow freely.



Executing or downloading/reading a flow does not require the right to read in the group that contains the flow. In fact, there is currently no right controlling the visibility of a single flow (there is no "hidden" attribute).

Workflow jobs and scheduled jobs permissions

The owner of a scheduled job, i.e. the user that created the job, can see the job in the KNIME Explorer and can delete it.

Permissions for scheduled job can also be set by the owner to grant access to the scheduling options.

In order to store a workflow job as new workflow in the Server repository, the user needs the right to download the original workflow (the flow the job was created from).



This is a requirement, because the newly created workflow is owned by the user that stores the job and the owner can easily grant itself the right to download the flow. Thus, if the original flow did not have the download right set, the user that is allowed to execute the flow could easily work around the missing download right.

Set up Server items permissions

To change permissions to a specific Server item you own, right-click the item in the KNIME Explorer and select *Permissions* from the context menu. In the window that opens, you can uncheck the *Inherit permissions from parent* option and the bottom pane will activate, as shown in [Figure 21](#).



If *Inherit permissions from parent* option is checked the access rights for the current item are taken over from the parent workflow group. If you move the item, the permissions might change (as its parent may change).

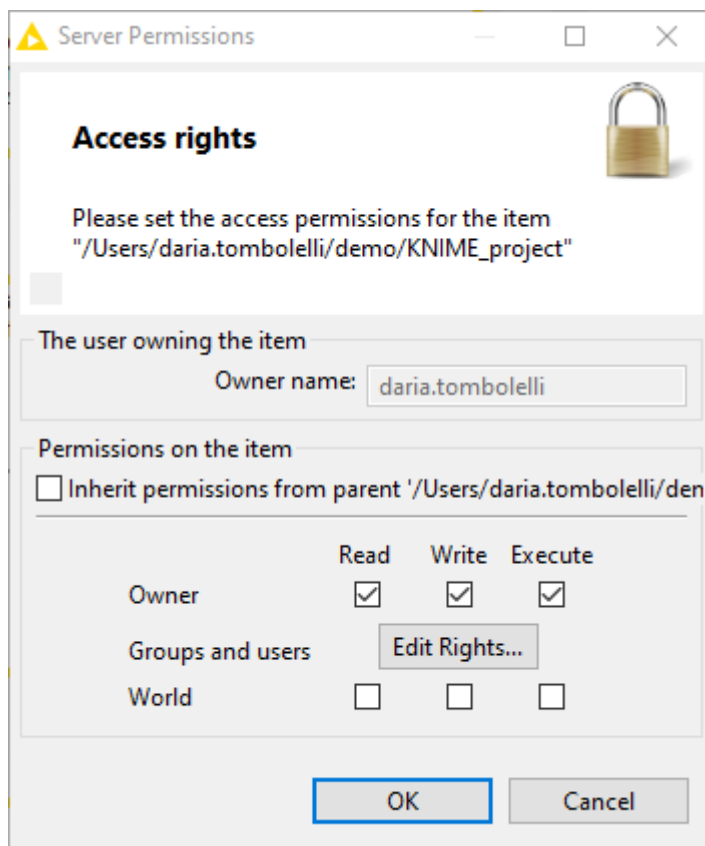


Figure 21. The Server Permissions window

As the owner of a Server item (workflow, shared component or workflow group) you can grant access rights to specific users or groups.

Owner rights

As a owner you can assign permissions to yourself to protect a flow from accidental deletion. You can change your own permissions at any time.

Groups and users rights

As a owner of a Server item, you can assign permissions to groups or users. Click *Edit Rights...* button and the window shown in [Figure 22](#) opens.

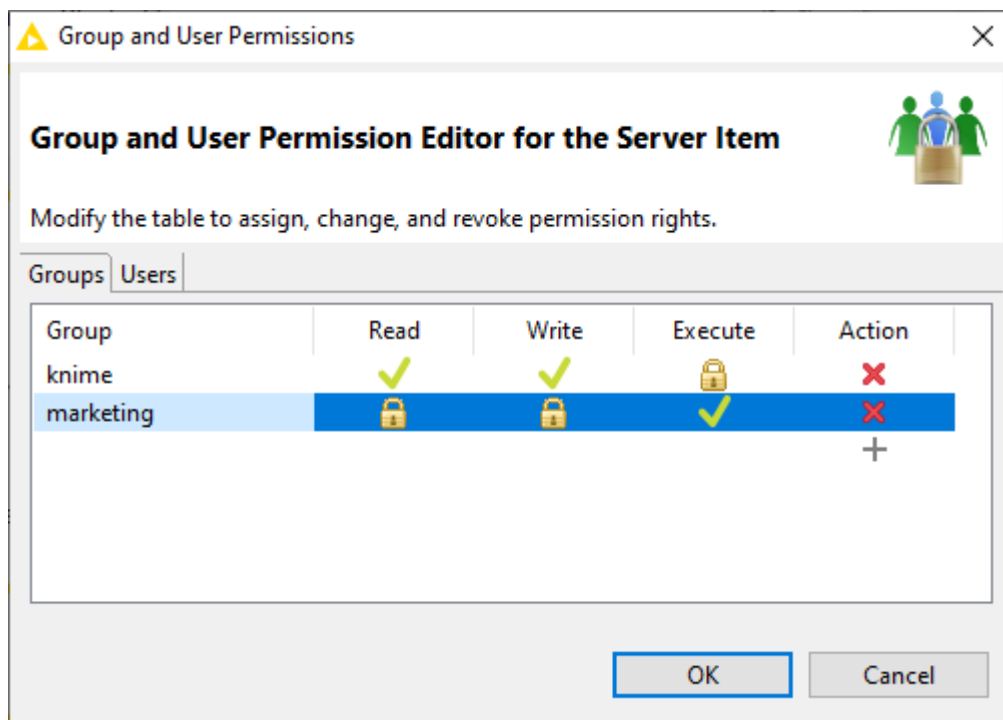


Figure 22. Group and User Permission Editor for the Server Item window

Here you can change permissions for:

- **Groups.** Under the *Groups* tab you can set permissions to different groups of users by adding them to the list. Click the plus symbol under *Action* column to add a new group or the red cross symbol to delete existing groups permissions. If an access right is granted to a group, all users that are assigned to that group have the access right.
- **Users.** Under the *Users* tab, you can set permissions to specific users.

World rights

Permissions can be set to all users that are not the owner and that are not specified under groups and users rights.



Access rights are adding up and can not be withdrawn. I.e. if you grant the right to execute a flow to other users and you define permissions for a certain group of users not including the execute right, the users of that group are still able to execute that flow, as they obtain that right through the other permissions.

Recursive permissions for workflow groups

At the bottom of the *Server Permissions* window that opens when *Permissions* is select from the context menu of a workflow group (see [Figure 23](#)) you can check the option to apply permissions to the selected workflow group and to all its children.

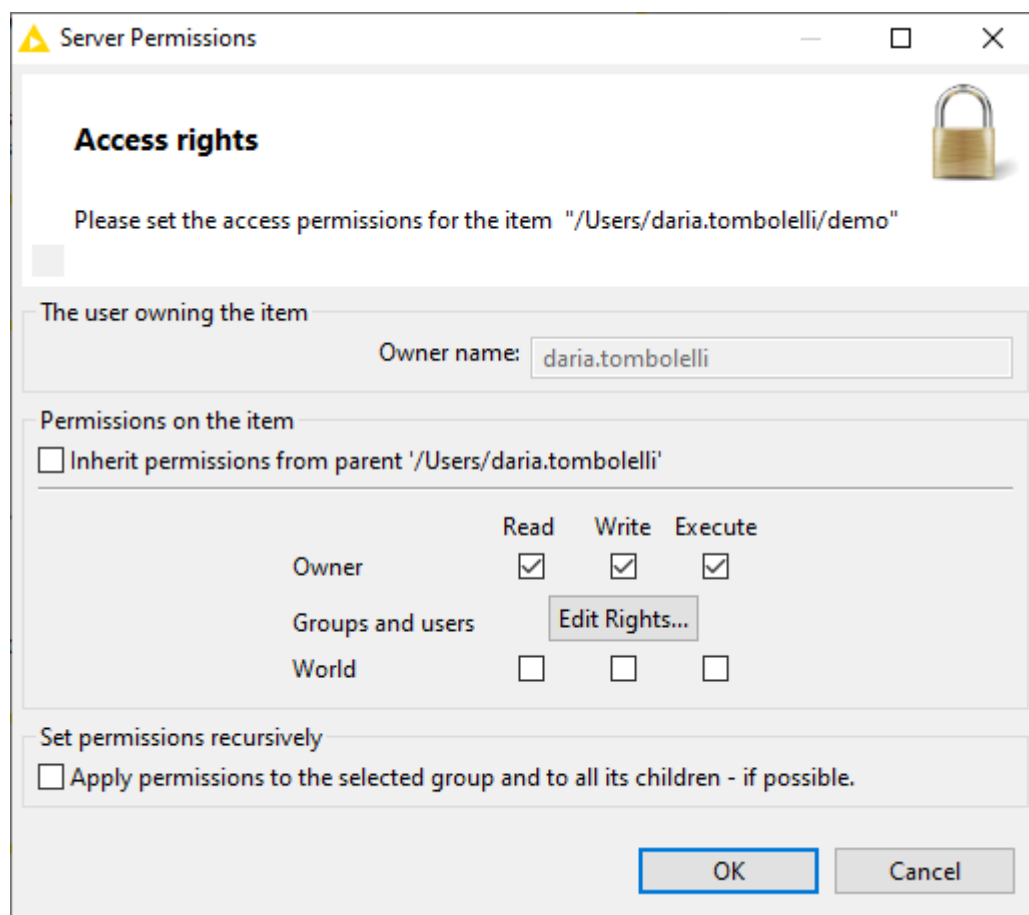


Figure 23. The Server Permissions window for workflow groups

Workflow pinning

Workflow Pinning can be used to let workflows only be executed by a specified subset of the available KNIME Executors when distributed [KNIME Executors](#) are enabled. Please be aware that this is only supported with a KNIME Server Large license.

For workflows that need certain system requirements (e.g. specific hardware, like GPUs, or system environments, like Linux) it is now possible (starting with KNIME Server 4.9.0) to define such Executor requirements per workflow. Only KNIME Executors that fulfill the Executor requirements will accept and execute the workflow job. To achieve this behavior, a property has to be set for the workflows.



The system admin of the KNIME Executors has also to specify a property for each Executor separately, as described in the [KNIME Server Administration Guide](#).

The properties consist of values that define the Executor requirements, set for a workflow, and Executor resources, set for an Executor, respectively.

Prerequisites for workflow pinning

In order to use workflow pinning, the [KNIME Server Distributed Executors](#) must be enabled and [RabbitMQ](#) must be installed. Otherwise, the set Executor requirements are ignored.

Setting `executor.requirements` property for a workflow

Executor requirements for a workflow can be defined by setting a property on the workflow. The Executor requirements are a simple comma-separated list of user-defined values.

To set workflow properties right-click a workflow in KNIME Explorer and select *Properties...* from the context menu. A dialog opens, as shown in [Figure 24](#), where you can view and edit the properties of the workflow.

Property Name	Description	Type	Default Value	User Specified Value
jobpool.size	The maximum number of idle jobs in the job pool for this workflow.	INT	0	0
executor.requirements	A comma seperated list of the executor resource requirements.	STRING		large-GPU, Linux, 32GB-RAM

Figure 24. The Server Item Properties dialog

Alternatively, workflow properties can also be set via a REST call, e.g. using `curl`:

```
curl -X PUT -u <user>:<password> http://<server-
address>/knime/rest/v4/repository/<workflow>:properties?com.knime.enterprise.server.executor.requirements=<executor requirements>
```

This will set the executor requirements *executor-requirements* for the workflow *workflow*.

Removing `executor.requirements` property for a workflow

Executor requirements can be removed by setting the property to an empty field. This can be done either in the KNIME Explorer or via a REST call:

```
curl -X PUT -u <user>:<password> http://<server-
address>/knime/rest/v4/repository/<workflow>:properties?com.knime.enterprise.server.executor.requirements=
```

Webservice interfaces

KNIME Server REST API

With KNIME Server, you can use REST to give access to your workflows to external applications. That might mean that you build a workflow to predict something, and then an external application can trigger that workflow to predict on the data that it is interested in.

A typical use case for a REST service would be to integrate the results from KNIME workflows into an existing, often complex IT infrastructure.



You can read a hands on guide about getting started with KNIME Server REST API on [KNIME blog](#).

The entry point for the REST interface is `https://<server-address>/knime/rest/`.

The interface is based on a hypermedia-aware JSON format called Mason.

Specific REST documentation is available at this address within your Server:

```
https://<server-address>/knime/rest/doc/index.html
```

The usual starting point to query the repository and to execute operations is:

```
https://<server-address>/knime/rest/v4/repository/
```

The returned document also contains links to further operations.

All KNIME Server functionalities are available via REST API, allowing you to:

- Upload/download/delete resources
- Upload licenses
- Empty trash or restore items
- Execute workflows
- Schedule jobs
- Set permissions
- Create user and groups

SwaggerUI for Workflows

The KNIME Server automatically generates SwaggerUI pages for all workflows that are present on the KNIME Server.

The **SwaggerUI** interface allows you to document and test your web services.

From the KNIME Analytics Platform you can access this functionality by right-clicking the item of interest in KNIME Explorer and selecting *Show API definition* from the context menu, as shown in **Figure 25**.

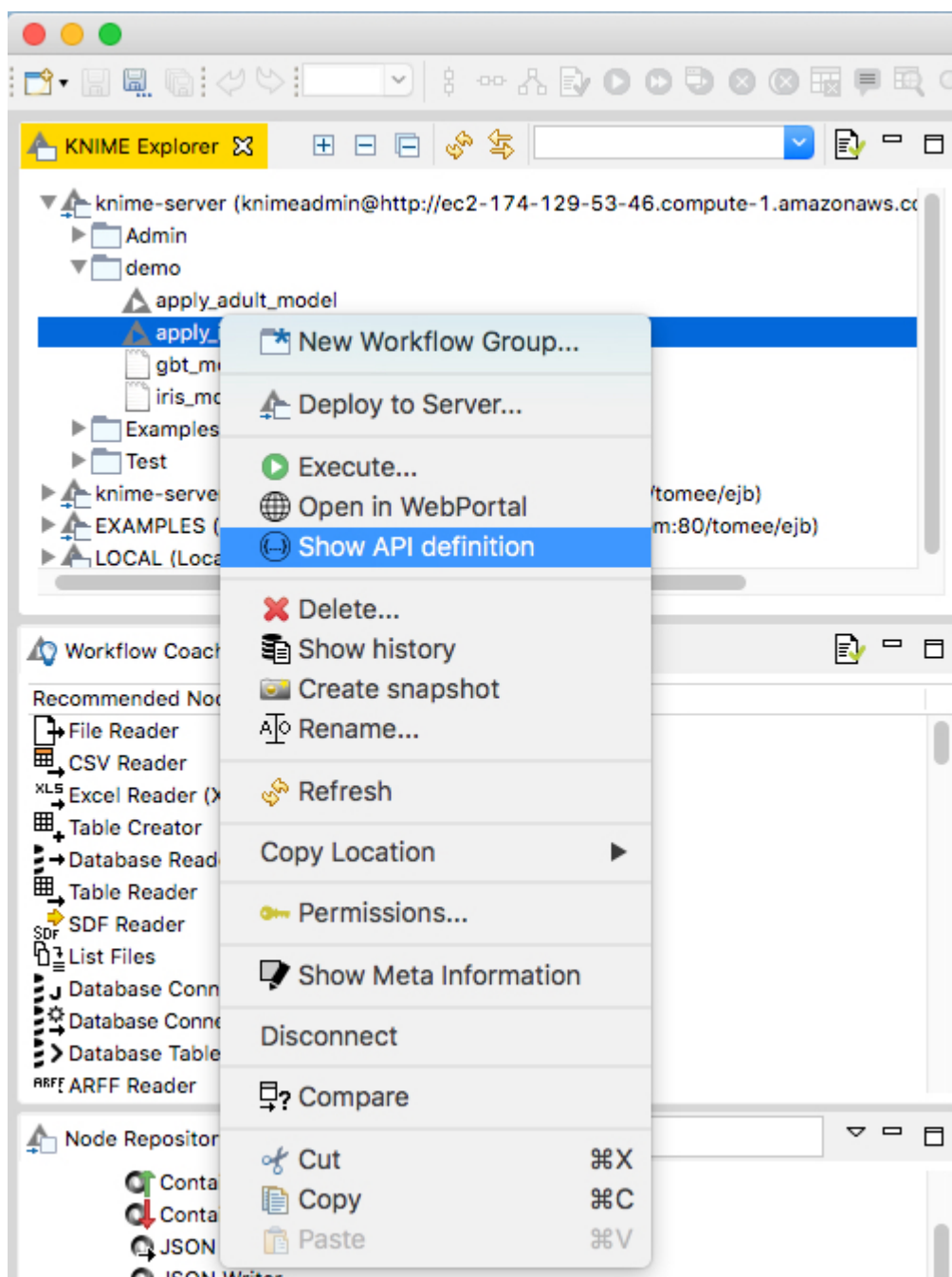


Figure 25. Select *Show API definition* from the context menu

A SwaggerUI page for that workflow will open in your browser, as shown in [Figure 26](#). It is also possible to browse to that page using the REST API as described in the above section.

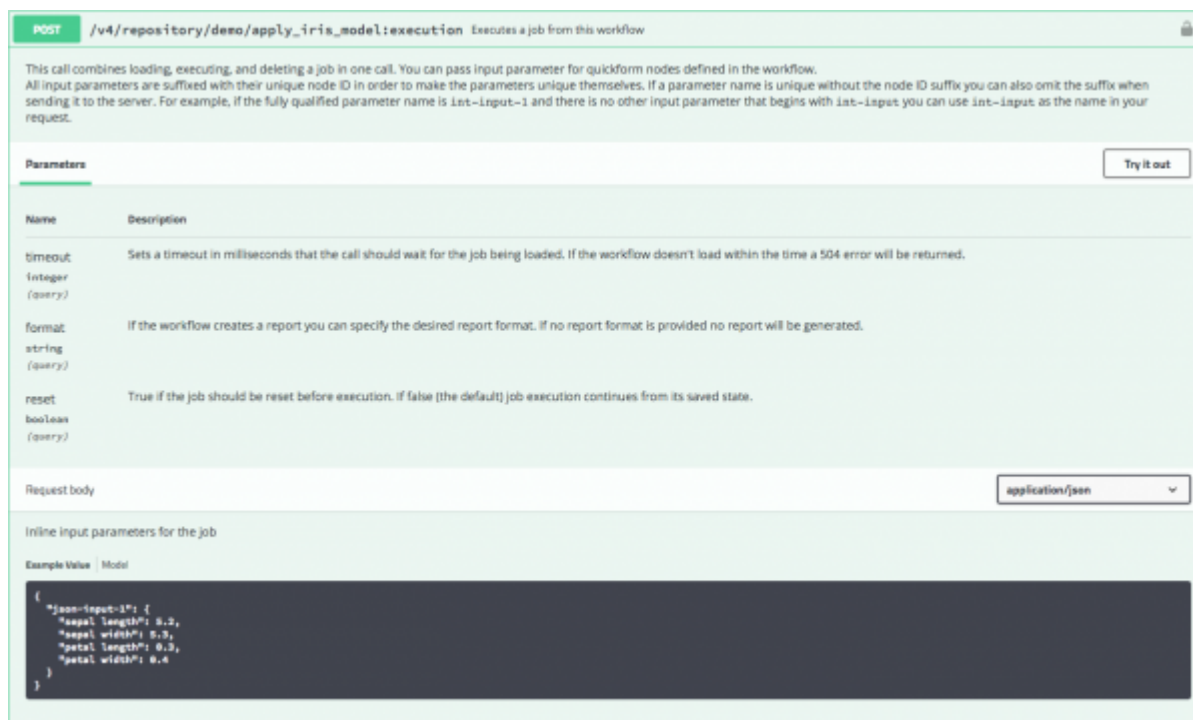


Figure 26. An example of SwaggerUI page

Workflow REST API nodes

In KNIME Analytics Platform node repository you can find, under *Workflow Abstraction* > *Workflow Invocation*, some useful nodes to interact with your workflows via the KNIME Server REST API.

Below is an explanation of the usage of some of these nodes together with the REST API 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.

Dialog - 3:1 - Container Input (Variable)

File

Container Input (Variable) Flow Variables

Parameter Name: variable-input

☒ Append unique ID to parameter name

Description:

☐ Accept any input

☒ Require input to match template variables specification

Template Variables

Set input variables as template No compatible input variables connected.

☐ Use simplified JSON format

Type	Variable Name	Value
String	variable_1	test
Integer	variable_2	1
Double	variable_3	0.1
Boolean	variable_4	true

+ Add

OK - Execute Apply Cancel ?

Figure 27. The Container Input (Variable) node configuration dialog with defined template variables

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 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}'
```

Recycle bin

KNIME Server also offers a recycle bin feature.

Every time you delete an item on KNIME Server, the item is actually moved to the Recycle Bin, unless otherwise selected in the confirmation dialog, shown in [Figure 28](#).

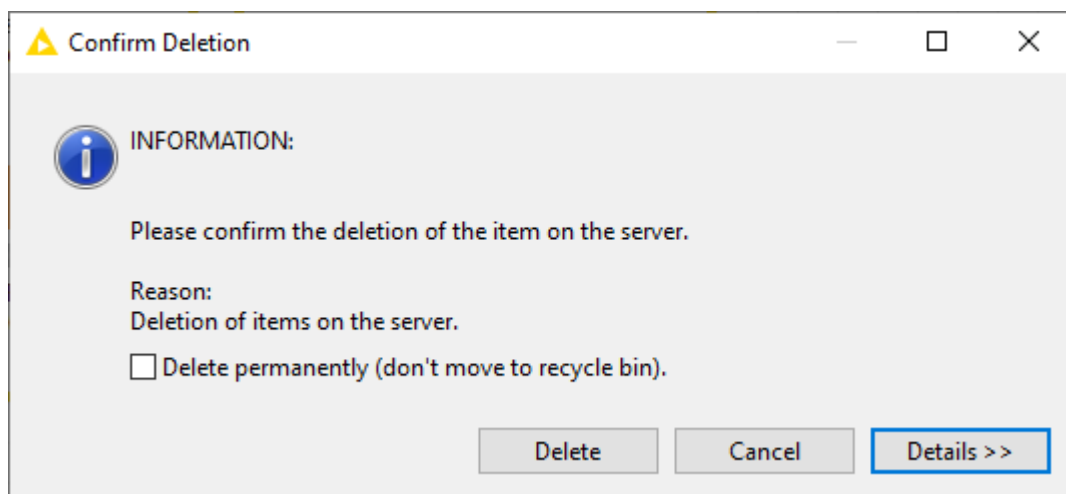


Figure 28. The confirmation dialog shown when deleting items from KNIME Server

You can see the contents of the Recycle Bin in the Server Recycle Bin view. To open this view go to *View* → *Other...* and select *Server Recycle Bin* under *KNIME Views* category.

In this view, shown in [Figure 29](#), you can see all deleted items of the Server instance currently logged in. You can restore or permanently delete items from the *Server Recycle Bin* view and additionally show the original contents of deleted workflow groups.

<server-mountID>				
Name	Item Type	Original Location	Date Deleted	User
▲ KNIME_project(5)	workflow	/ <item-path>	2020-10-12 15:59:26	<username>
▲ Deployment_Workflow	workflow	/ <item-path>	2020-10-12 15:59:19	<username>
▲ Capture-0_97	workflow	/ <item-path>	2020-10-12 15:59:09	<username>
▲ 01_Modelling_Workflow	workflow	/ <item-path>	2020-10-12 15:58:57	<username>

Figure 29. The Server Recycle Bin view

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