Open for Innovation
KNIME

KNIME WebPortal Administration Guide

KNIME AG, Zurich, Switzerland
Version 4.13 (last updated on 2022-08-22)

Table of Contents

Introduction

KNIME WebPortal is an extension to KNIME Server. It provides a web interface that lists all accessible workflows, enables their execution and investigation of results.

This guide covers in detail the configuration options for KNIME WebPortal to run it as administrator.

If you are looking for a guide on how to use KNIME WebPortal please refer to the KNIME WebPortal User Guide.

For a more general explanation about how to install KNIME Server, or about the other configuration options specific to KNIME Server please refer to the following guides and additional resources:

- KNIME Server Installation Guide
- KNIME Server Administration Guide
- KNIME Server Advanced Setup Guide.

KNIME WebPortal page

The KNIME WebPortal can be accessed with any standard browser (see below for a list of supported browsers) at:

```
https://<server-address>/knime/webportal/
```

To change the `/knime/` part (Context Root) of the URL please refer to the KNIME Server Installation Guide.

Please be aware that with the release of KNIME WebPortal version 4.11+ it is necessary to **always** specify a Context Root.

Supported browsers

The following browsers are supported by KNIME WebPortal version > 4.11. We do not actively test or support KNIME WebPortal in older browser versions.

Google Chrome	Version 87+
Microsoft Edge	Version 44+
Firefox	Version 78+
Safari	Version 14+

Most WebPortal functionality may work with older browsers, although this is not tested.

KNIME WebPortal version < 4.11 also supports Internet Explorer version 11.0+. Please note that IE8 and below are not supported.

Theming

The look of KNIME WebPortal can be modified in order to customize it with corporate colors, logo, and other elements.

In this section you will find information about how to modify the login page or the general look of KNIME WebPortal.

Customizing the login page

The login page is customized in a `knime_template_login.html` file. In the server repository, under `\workflow_repository\config\webportalTemplate.goldMining\` you will find an example for this file.

Please note that the file `knime_template_login.html` will be read from a folder with name and path corresponding to `<knime-server-repository>\workflow_repository\config\webportalTemplate\`.

Figure 1. A customized login page for KNIME WebPortal

Use the following placeholders to create your own `knime_template_login.html` file:

Location ID	Mandatory	Placeholder for…
`knime-login-image`		The KNIME logo shown on the login page
`knime-version-label`		A label displaying the current server version

Location ID	Mandatory	Placeholder for…
`knime-login-message`	Yes	Additional login message (e.g. session expired)
`knime-input-username`	Yes	The login input field for the username
`knime-input-password`	Yes	The login input field for the password
`knime-login-button`	Yes	The login button

Please be aware that to be able to load the images correctly, they should be referenced in the `knime_template_login.html` file using the scheme `<img src="/CONTEXT/VAADIN/themes/knime/layouts/custom/myLoginImage.jpg" />` where `/CONTEXT/` is the Context Root (usually the default one `/knime/`) used to define the KNIME WebPortal URL. In case you want to use this customized login page on a Server which is installed with a different Context Root you would need to change this scheme to the correct one. Also worth noting is that if the Server is supposed to be used behind a proxy this approach would also have to list the proxy address correctly.

Customizing the general look of KNIME WebPortal

In your KNIME Server installation go to the server repository and navigate to `config` folder located inside the `workflow_repository`.

Here you will find an example folder `webportal2Theme.theCompany` which contains some theming files. You can modify these files or create your own theming folder.

To personalize the theming of your WebPortal open the file `knime-server.config` located in `workflow_repository/config` with any editor and add the name of your theming folder next to `com.knime.server.webportal2.theme_folder`. In case the theming folder is not located under `workflow_repository/config` you will need to add the theming folder path relative to the `knime-server.config` file.

```
com.knime.server.webportal2.theme_folder=<relative-path-to>/webportal2Theme.theCompany
```

We recommend using the browser Developer Tools to develop and test theming and then copy it into the `theme.config` to avoid frequent server restart.

`theme.config` file

Inside the theming folder you will find a `theme.config` file.

Here you can define the following properties using the corresponding property keys:

- `title`: A title for your WebPortal
- `css_file`: The relative path of a custom css file to style KNIME WebPortal
- `logo_file`: The relative path of an svg image for the logo on the top left corner in the WebPortal pages
- `header_image`: The relative path of an image for the header
- `favicon_svg`; `favicon_png`: The relative path of an svg or png image for the `favicon`.

css file

In the css file you can add explicit fonts and styling for KNIME WebPortal pages and for some of the Widget nodes.

In the example file `the-company.css`, that you will find located in the example theming folder `webportal2Theme.theCompany`.

Here, you can define different css elements.

In the file `the-company.css`, you will find some of these elements, listed below. You can either make your changes in this file, or create your own css file.

- Fonts:

```
@font-face {
  font-family: 'OpenSans Light';
  src: url(font/OpenSans/OpenSans-Light.ttf);
}
```

- Header background color and font:

```
/* Header */
--theme-header-background-color: #942e16;
--theme-header-background-color: #FFFFFF;
--theme-header-font-family: 'Roboto Slab'
```

- Logo size:

```
/* Logo */
--theme-logo-width: 150px;
--theme-logo-height: 50px;
```

- Headlines font, color, and font weight:

```
/* Headlines */
--theme-headlines-font-family: 'Roboto Slab';
--theme-headlines-color: black;
--theme-headlines-font-weight: 300;
```

- Text font and color:

```
/* Text */
--theme-text-normal-font-family: 'OpenSans Light';
--theme-text-normal-color: #3E3A39;

--theme-text-medium-font-family: 'OpenSans Bold';
--theme-text-medium-color: #6E6E6E;

--theme-text-bold-font-family: 'Roboto Slab Bold';
--theme-text-bold-color: black;
```

- Default, small, split, and function button color and radius:

```
/* Default Button */
--theme-button-background-color: #000000;
--theme-button-background-color-hover: #942e16;
--theme-button-background-color-focus: #942e16;
--theme-button-foreground-color: white;
--theme-button-foreground-color-hover: white;
--theme-button-foreground-color-focus: white;
--theme-button-border-radius: 0;
--theme-button-border-color: transparent;
--theme-button-border-color-hover: transparent;
--theme-button-border-color-focus: transparent;

/* Small Button */
--theme-button-small-background-color: black;
--theme-button-small-background-color-hover: #707070;
--theme-button-small-background-color-focus: #707070;
--theme-button-small-foreground-color: white;
--theme-button-small-foreground-color-hover: white;
--theme-button-small-foreground-color-focus: white;
--theme-button-small-border-radius: 0;
--theme-button-small-border-color: transparent;
--theme-button-small-border-color-hover: transparent;
--theme-button-small-border-color-focus: transparent;

/* Split Button */
--theme-button-split-background-color: #000000;
--theme-button-split-background-color-hover: #440E07;
--theme-button-split-background-color-focus: #440E07;
--theme-button-split-foreground-color: white;
--theme-button-split-foreground-color-hover: white;
--theme-button-split-foreground-color-focus: white;
--theme-button-split-border-radius: 0;
--theme-button-split-border-color: transparent;
--theme-button-split-border-color-hover: transparent;
--theme-button-split-border-color-focus: transparent;
--theme-button-split-divider-color: white;

/* Function button */
--theme-button-function-border-radius: 0;
--theme-button-function-background-color: transparent;
--theme-button-function-background-color-hover: #942e16;
--theme-button-function-background-color-focus: #942e16;
--theme-button-function-background-color-active: #942e16;
--theme-button-function-foreground-color: #6E6E6E;
--theme-button-function-foreground-color-hover: white;
--theme-button-function-foreground-color-focus: white;
--theme-button-function-foreground-color-active: white;
```

- Widget nodes elements colors and style, including:

- Dropdown

```
/* Dropdown (global) */
--theme-dropdown-background-color: transparent;
--theme-dropdown-background-color-focus: #942e1666;
--theme-dropdown-background-color-hover: #942e1666;
--theme-dropdown-background-color-selected: #942e16;
--theme-dropdown-foreground-color: #3E3A39;
--theme-dropdown-foreground-color-focus: #3E3A39;
--theme-dropdown-foreground-color-hover: #3E3A39;
--theme-dropdown-foreground-color-selected: white;
```

- Text links

```
/* Text links */
--theme-text-link-background-color: transparent;
--theme-text-link-background-color-hover: #942e16;
--theme-text-link-background-color-focus: #942e16;
--theme-text-link-foreground-color: #942e16;
--theme-text-link-foreground-color-hover: white;
--theme-text-link-foreground-color-focus: white;
--theme-text-link-text-decoration: none;
--theme-text-link-text-decoration-hover: none;
--theme-text-link-text-decoration-focus: none;
```

- Tooltip

```
/* Tooltip */
--theme-tooltip-background-color: #942e16;
--theme-tooltip-foreground-color: white;
```

- Checkbox

```
/* Checkbox */
--theme-checkbox-border-color: #888888;
--theme-checkbox-border-color-focus: #3E3A39;
--theme-checkbox-border-color-hover: #888888;
--theme-checkbox-border-color-selected: #942e16;
--theme-checkbox-border-color-selected-focus: #942e16;
--theme-checkbox-border-color-selected-hover: #888888;
--theme-checkbox-background-color: white;
--theme-checkbox-background-color-focus: white;
--theme-checkbox-background-color-hover: #942e1666;
--theme-checkbox-background-color-selected: #942e16;
--theme-checkbox-background-color-selected-focus: white;
--theme-checkbox-background-color-selected-hover: #942e1666;
--theme-checkbox-foreground-color-selected: white;
--theme-checkbox-foreground-color-selected-focus: #942e16;
--theme-checkbox-foreground-color-selected-hover: #3E3A39;
```

- Selection list

```
/* Selection List */
--theme-select-control-background-color: transparent;
--theme-select-control-background-color-hover: #942e1666;
--theme-select-control-background-color-focus: #707070;
--theme-select-control-foreground-color: #6E6E6E;
--theme-select-control-foreground-color-hover: #3E3A39;
--theme-select-control-foreground-color-focus: white;
```

- Radio buttons

```
/* Radio buttons */
--theme-radio-border-color: #888888;
--theme-radio-border-color-hover: #888888;
--theme-radio-border-color-selected: #00677F;
--theme-radio-border-color-selected-hover: #888888;
--theme-radio-background-color: white;
--theme-radio-background-color-hover: #942e1666;
--theme-radio-background-color-selected: #942e16;
--theme-radio-background-color-selected-hover: #942e1666;
--theme-radio-foreground-color-selected: white;
--theme-radio-foreground-color-selected-hover: #3E3A39;
```

- Slider

```
/* Slider */
--theme-slider-border-radius: 0;
--theme-slider-border-color: #888888;
--theme-slider-border-color-hover: #888888;
--theme-slider-border-color-focus: #707070;
--theme-slider-foreground-color: white;
--theme-slider-foreground-color-hover: #942e16;
--theme-slider-foreground-color-focus: #993c2766;
--theme-slider-bar-radius: 3.5px;
--theme-slider-background-color: #942e16;
}
```

Monitoring and administration portal

Through KNIME WebPortal it is also possible to access the monitoring and administration portal. This can be reached from the WebPortal by clicking on the corresponding tab at the top of the page.

- The monitoring portal is available for all the users. However, only users with administrator privileges are allowed to display the active jobs and schedules owned by all the users.
- The administration portal is accessible only to users with administrator privileges.

Users can be granted administrator privileges via the server configuration file (please see Server administrator section in the KNIME Server Administrator Guide).

For an overview of the functionalities of the monitoring portal available for users without administrator privileges, please refer to the KNIME WebPortal User Guide.

The monitoring portal provides details about jobs, schedules and executors and allows to download KNIME Server logs.

The administration portal, instead:

- Provides details about the Server's status and allows to upload the Server license
- Allows to manage local users and groups
- Allows to change the Server's configurations

For a more comprehensive guide on how to use the monitoring and administration portal as a KNIME Server administrator please refer to the Monitoring and administration portal section on the KNIME Server Administration Guide.

Server configuration files and options

KNIME Server can be configured:

- Via the administration portal accessible through the KNIME WebPortal
- By manually changing the options on the `knime-server.config` file

When changing the Server configurations from the administration portal the `knime-server.config` file is automatically overwritten.

All the options that are listed in the next section are configurable via the administration portal.

KNIME Server configuration file

When manually configuring KNIME Server you need access to the `knime-server.config` file. The file can be found in `<knime-server-repository>/config/knime-server.config`. Most of the parameters defined in this file can be changed at runtime and will take effect as soon as possible. Default values will be used for empty or missing configuration options.

The section KNIME Server configuration file options for KNIME WebPortal contains a list of all configuration options and explanations valid for KNIME WebPortal. For a list of all configuration options for KNIME Server please refer to the KNIME Server configuration file options section of the KNIME Server Administration Guide.

KNIME Server configuration file options for KNIME WebPortal

Below you will find a table with all supported configuration options (in alphabetical order). Some of them are described in more detail in later sections. The options can be set in the file `<knime-server-repository>/config/knime-server.config`.

For Windows users: For paths in the server configuration file either use forward slashes ("/") or double backslashes ("\\"). A single backslash is used to escape characters.

The following annotations to the table, provide some additional information about which Executor type is affected, and whether changes take effect at runtime, or require a server restart.

[ST]	changes take effect after a restart of KNIME Server
[RT]	changes can take effect at runtime

Some options can be set as property in the `knime-server.config` file as well as by defining an environment variable (Env). The environment variable changes will only take effect after a restart of KNIME Server. If the environment variable for an option is set, the property in the configuration file will be ignored.

`com.knime.server.webportal.csp=<CSP statement>` **[RT]**

Specifies a custom Content Security Policy for the WebPortal. It may be necessary to override the default if you are using custom JavaScript views that load external resources. The default works for all standard KNIME views. For more information about how to write the CSP statement, please refer to this resource.

`com.knime.server.webportal.debug=<true|false>` **[RT]**

Enables or disables a debug mode for the WebPortal. In debug mode JavaScript and CSS sources are included in their non-minified version and log messages might be printed to the console of the browser.

`com.knime.server.webportal.disable_legacy=<true|false>` **[ST]**

Disables the legacy WebPortal (version <4.11) and forwards all requests to KNIME WebPortal version 4.11. Default is that the legacy WebPortal is disabled.

`com.knime.server.webportal.disable_report_preview=<true|false>` **[RT]**

Disables the report preview in the KNIME WebPortal.
Default is to show report previews.

`com.knime.server.webportal.disable_warning_messages=<true|false>` **[RT]**

Disables warnings messages at the end of a workflow execution on the KNIME WebPortal.
Default is to show warnings messages.

`com.knime.server.webportal.hide_version=<true|false>` **[RT]**

Hides the server's version in the KNIME WebPortal.
Default is to show the version number.

`com.knime.server.webportal.ie_compatibility=<IE version identifier>` **[RT]**

This option allows you to set the IE compatibility mode that the KNIME WebPortal sends to the browser. Default is not to send any compatibility information.

`com.knime.server.webportal.restrict_x_frame_options=<value>` **[RT]**

Sets the value of the HTTP-header `X-Frame-Options`. `<value>` must be one of `DENY`, `SAMEORIGIN` or `ALLOW-FROM xxx`, where `xxx` needs to be replaced with the URL of the embedding page. If this option is not present in the configuration file, the HTTP-header `X-Frame-Options` is not sent. See also avoiding clickjacking attacks section in KNIME Server Administration Guide.

`com.knime.server.webportal.sketcher_page=<relative URL of Sketcher Page>` **[RT]**

Define the location of the main sketcher html-document. Helpful when the sketcher is deployed as static resources under a different context root. Note that the KNIME WebPortal and the sketcher need to be in the same domain. Otherwise cross-domain scripting would occur, which is considered a security threat in all major browsers and thus not allowed.

`com.knime.server.webportal.sketcher_size=<width x height, e.g. 300x300>` **[RT]**

Define the size of the sketcher iframe.
`300x300` is the default value for the Marvin Sketcher.

`com.knime.server.webportal.title_label=<Server Name>` **[RT]**

Define an additional title label displayed to the right of the KNIME WebPortal logo.
-

`com.knime.server.webportal2.theme_folder=<theme folder>` **[ST]**

Define a folder inside the config folder which contains theme information for the KNIME WebPortal version 4.11. For further information see the Theming section.

Managing user and consumer access

It is possible to restrict which groups (or which individual users) are eligible to log in as either users or consumers. In this context, a user is someone who logs in from a KNIME Analytics

Platform client to e.g. upload workflows, set schedules, or adjust permissions. On the other hand, a consumer is someone who can only execute workflows from either the KNIME WebPortal or via the KNIME Server REST API.

In order to control who is allowed to log in as either user or consumer, the following settings need to be adjusted in the `knime-server.config`:

`com.knime.server.login.allowed_groups`: This setting has to include **all** groups that should be allowed to login to KNIME Server, regardless of whether they are users or consumers.

`com.knime.server.login.consumer.allowed_groups`: List of groups that should be allowed to use the WebPortal or REST API to execute workflows.

`com.knime.server.login.user.allowed_groups`: List of groups that should be allowed to connect to KNIME Server from a KNIME Analytics Platform client.

Usage Example

```
com.knime.server.login.allowed_groups=marketing,research,analysts

com.knime.server.login.consumer.allowed_groups=marketing,research,analysts

com.knime.server.login.user.allowed_groups=research
```

In the above example, we first restrict general access to KNIME Server to individuals in the groups `marketing`, `research`, and `analysts`. All individuals who are not in any of these groups won't be able to access KNIME Server at all. Next, we allow all three groups to login as consumers via WebPortal or REST API. Finally, we define that only individuals in the group `research` should be able to log in as users from a KNIME Analytics Platform client.

By default, these settings are left empty, meaning that as long as users are generally able to login to your KNIME Server (e.g. because they are in the allowed AD groups within your organization), they can log in as either users or consumers. Since the number of available user licenses is typically lower than the number of consumers, it is recommended to restrict user access following the above example.

HTML and JavaScript Sanitization

Some nodes, such as the Table View, Table Editor, or Text Output Widget, allow a workflow developer to render HTML, either by specifying it directly in the node configuration or by passing data in HTML format. This is useful for rendering SVGs or formatted text in the

WebPortal. However, if the data or the workflow developer cannot be trusted, it is sensible to enable sanitization of the output in order to prevent the execution of JavaScript code in the consumer's browser. This can for example happen by adding `<script>` tags into the output or other HTML tags with events such as `onclick` or `onload`.

Sanitization of output in the WebPortal is controlled in the executor via system properties that can be set in the `knime.ini` file in the executor's installation directory. The following properties are available:

`-Djs.core.sanitize.clientHTML=<true|false>`

: If true (default false), user input/data in view/wizard pages will be sanitized for potential vulnerabilities before rendering in the browser. The default sanitization behavior can be overridden with the additional sanitize properties. These additional properties have no effect if this property is false or empty (default).

`-Djs.core.sanitize.allowElements=<tag1>,<tag2>,⋯`

: A comma separated list of valid HTML element tags which should be allowed in the sanitized data. Any non-empty value overrides the default element policy defined by OWASP. An extensive list of elements that can be whitelisted can be found here.

`-Djs.core.sanitize.allowAttributes=<attr1>,<attr2>,⋯`

: A comma separated list of valid HTML attribute tags which should be allowed in the sanitized data. Any non-empty value overrides the default attribute policy defined by OWASP. An extensive list of attributes that can be whitelisted can be found here.

`-Djs.core.sanitize.allowNodesPath=<absolute_path>`

: The optional absolute path to a file containing new-line delimited node names (as seen in the node description; e.g. "Table View") to exclude from sanitization before being transferred to the client. Default is empty, sanitizing all node data.

Email notification

Email notification service available in KNIME WebPortal for finished workflow jobs can be activated on KNIME Server. The emails are sent from a single email address which can be configured as part of the web application's mail configuration. If you don't want to enable the email notification feature, no email account is required. You can always change the configuration and enter the account details later.

Setting up the server's email resource

The email configuration is defined in the `<knime-server-repository>/config/knime-server.config`. The installer has already created this file. In order to change the email configuration, you have to modify or add configuration properties. The table below shows the list of supported parameters (see also the JavaMail API documentation).

Name	Value
`mail.smtp.from`	Address from which all mails are sent
`mail.smtp.host`	SMTP server, required
`mail.smtp.port`	SMTP port, default `25`
`mail.smtp.auth`	Set to `true` if the mail server requires authentication; optional
`mail.smtp.user`	Username for SMTP authentication; optional
`mail.password`	Password for SMTP authentication; optional
`mail.smtp.starttls.enable`	If `true`, enables the use of the STARTTLS command (if supported by the server) to switch the connection to a TLS-protected connection before issuing any login commands. Defaults to false.
`mail.smtp.ssl.enable`	If set to true, use SSL to connect and use the SSL port by default. Defaults to `false`.

Note that the mail configuration file contains the password in plain text. Therefore, you should make sure that the file has restrictive permissions.

Installing a molecule sketcher

The KNIME WebPortal can be used with an integrated molecular sketcher. In order to use one of the following sketchers, configure as described below.

WAR files for each of these sketchers are available here. In order to use one of these sketchers copy the corresponding WAR file into `<apache-tomcat>/webapps` and configure as described below. The WAR file should automatically be extracted into a folder of the same name. You may need to restart Tomcat before the WAR is extracted.

Marvin JS sketcher

The current version of the Marvin Sketcher provided by Chemaxon is available at this link. Download the sketcher code and extract its contents to

```
<apache-tomcat>/webapps/com.knime.enterprise.sketcher.marvinJS/marvinJS/
```

Change the server configuration in `knime-server.config` and set

```
com.knime.server.webportal.sketcher_page=/com.knime.enterprise.sketcher.marvinJS/sketcher.html
```

Marvin sketcher applet (Deprecated)

You are most likely interested in the newer Marvin JS functionality, in which case check the previous section Marvin JS sketcher. If you're absolutely sure that this is the functionality that you're interested in, then please read on.

The current version of the Marvin Sketcher provided by Chemaxon is available at this link. Download the sketcher applet and extract its contents to

```
<apache-tomcat>/webapps/com.knime.enterprise.sketcher.marvin/marvin/
```

Change the server configuration in `knime-server.config` and set

```
com.knime.server.webportal.sketcher_page=/com.knime.enterprise.sketcher.marvin/sketcher.html
```

GGA Ketcher

Change the server configuration in `knime-server.config` and set

```
com.knime.server.webportal.sketcher_page=/com.knime.enterprise.sketcher.ketcher/sketcher
.html
```

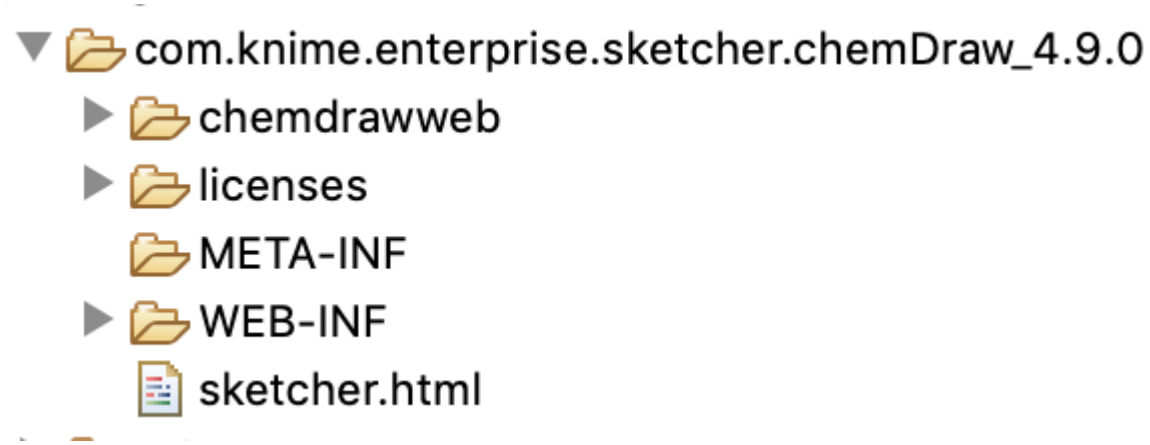
JSME Molecule Editor

Change the server configuration in `knime-server.config` and set

```
com.knime.server.webportal.sketcher_page=/com.knime.enterprise.sketcher.jsme/sketcher.ht
ml
```

ChemDraw JS Sketcher

Extract the package file containing the ChemDraw JS sketcher and locate the `chemdrawweb` folder. Copy this folder into the extracted web application for the sketcher integration in your Tomcat `webapps` folder. Also copy your license file for ChemDraw JS in a `licenses` folder. Your folder structure should look similar to the following picture:



chemDraw JS Setup

Change the server configuration in `knime-server.config` and set

```
com.knime.server.webportal.sketcher_page=/com.knime.enterprise.sketcher.chemDraw/sketche
r.html
```

Adapt the path if you have deployed the sketcher integration under a different context root.

Credits

KNIME WebPortal uses open source software components. We say thanks to the developers of these components and acknowledge their work. Under `https://<server-address>/knime/webportal/open-source-credits` we list all components which may be contained in portions in the WebPortal. Please refer to the individual component source for detailed information.

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