IDE User Guide

Introduction

Introduction

eXo IDE is a powerful Web application which provides a rich environment for developing different content, scripts, and services. It requires no additional installations, runs in a browser and allows to access and work with files online from anywhere. eXo IDE offers:

- Work with the Remote File System through *Virtual File System* including Browsing, File Locking, File Search, Versioning.
- Code Editor with syntax highlighting for many popular languages and advanced features such as: Code Autocomplete, Code Outline and WYSIWYG Editor for HTML
- Support for multiple programming and descriptive languages such as: JavaScript, HTML, XML, CSS, Groovy, etc.
- Tools for developing client side applications including technologies such as Netvibes Widgets, Google Gadgets and Groovy Templates;
- Development of server-side applications, running, debugging, and interaction with the client side via REST Service.

The eXo IDE User Guide aims at providing explanations and step-by-step instructions of functions in eXo IDE.

eXo IDE serves two types of J2EE roles: **administrators** and **developers**. Before working with the eXo IDE, make sure (ask your system admin) that your credentials contain one of the listed roles. eXo IDE offers pretty the same functionality for them except that **developers** can not deploy theirs REST services on common environment and uses kind of "sandbox" to deploy them.

Requirements

Supported Web browsers:

- Mozilla Firefox 3.0
- Internet Explorer 7.0
- Safari 3.2
- Google Chrome 5.0

Basic Operations

When the application is launched the first thing you see is the window like this:

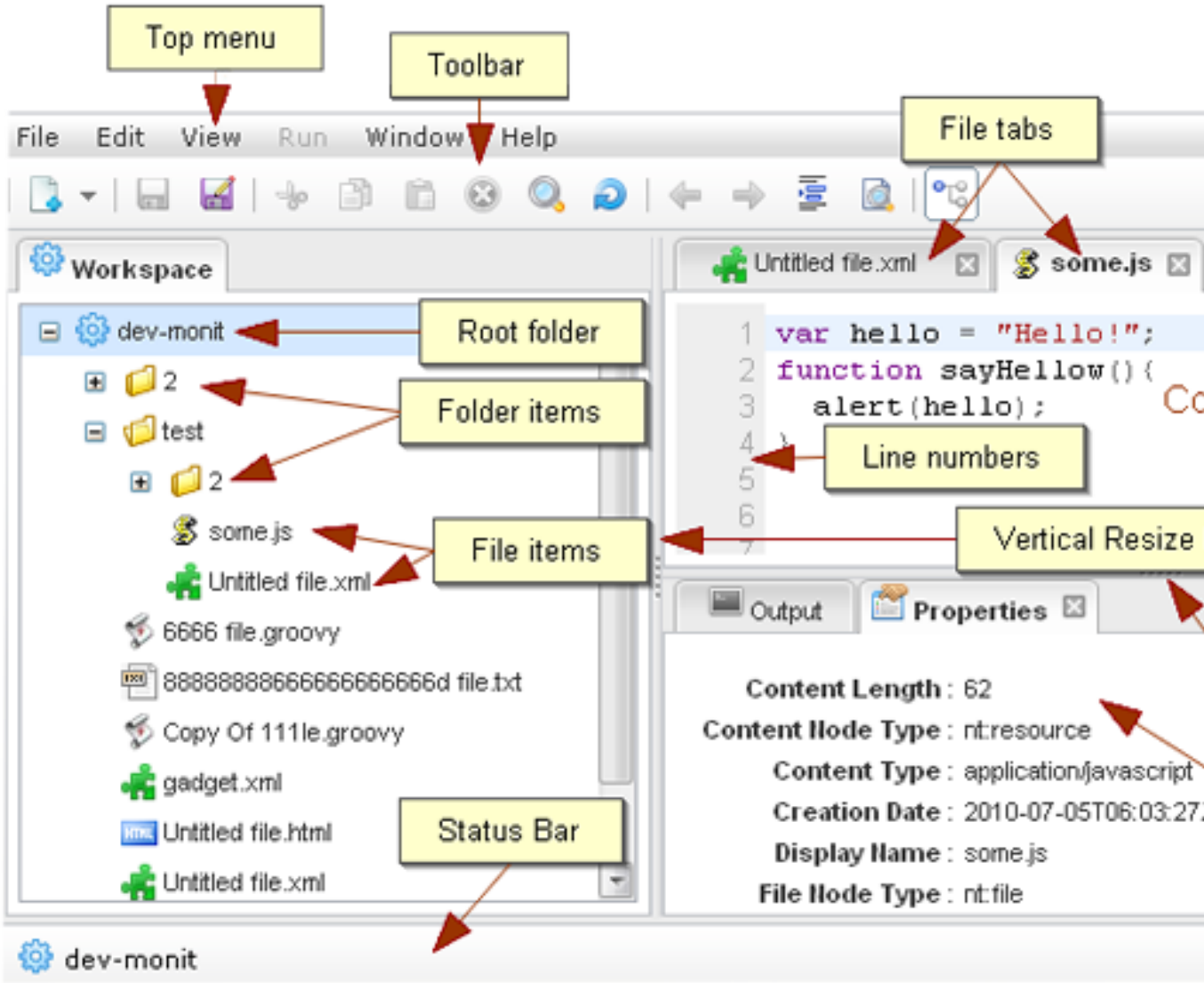


Illustration. The eXo IDE interface with main parts

Exo IDE has customizable interface with enhanced windows and editors. The *Virtual File System* is the physical location (file path) you are working in. It is displayed as the *Workspace Panel* containing folders and files.

The *Content Panel* consists of two horizontal panels:

- The *Top Panel* shows files in several file tabs with their content.
- The *Bottom Operation Panel* may contain several tabs with the list of file properties, REST Service, Google Gadget or Netvibes Widget output messages, HTML, Google Gadget, ECM

Template or Netvibes Widget files preview. To view these tabs, just use special buttons at the right part of the toolbar, or select **View** or **Run** from the top menu.

These panels are divided by *Horizontal Resize Bar*. Also, both *Workspace Panel* and *Content Panel* are divided by the *Vertical Resize Bar*. Both *Workspace Panel* and *Content Panel* are also divided by the *Vertical Resize Bar*.

You can maximize or minimize one of the panels by clicking the special button at the top right corner of this panel.

You can do some basic actions, such as browsing, creating, editing, coping, renaming, uploading, downloading, moving, deleting files and folders. At the right column are shown the *Outline Panel* and *Version Panel*:

- The *Outline Panel* enables you to access tags, variables, functions and other file elements quickly.

- The *Version Panel* is to navigate to the file versions.

- The active panel is highlighted with one special thick border around the panel window.

Select Workspace

Each workspace has its unique URL.

eXo IDE may provide access to one or more workspaces on the server.

Note

In a case if system allowed only one workspace the following option is hidden.

To select a workspace, do as follows:

- **Step 1:** Go to **Window > Select Workspace...** from the top menu.
- **Step 2:** Select your desired workspace and click **OK** to go to the appropriate workspace; or simply double-click your desired workspace.

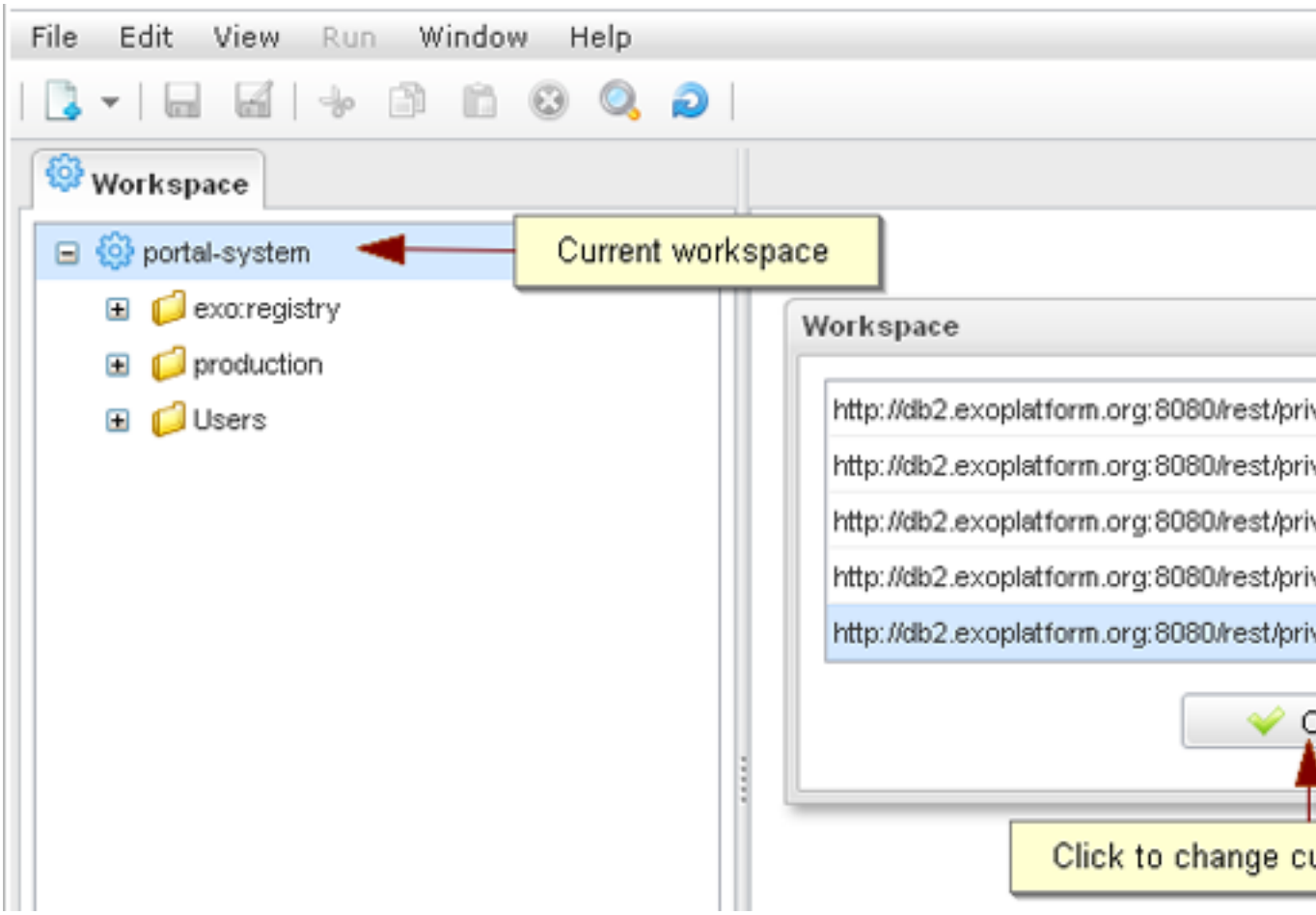


Illustration. Select Workspace window

If you open eXo IDE without any current workspaces, the dialog asking you to go to **Window > Select Workspace...** appears as below.

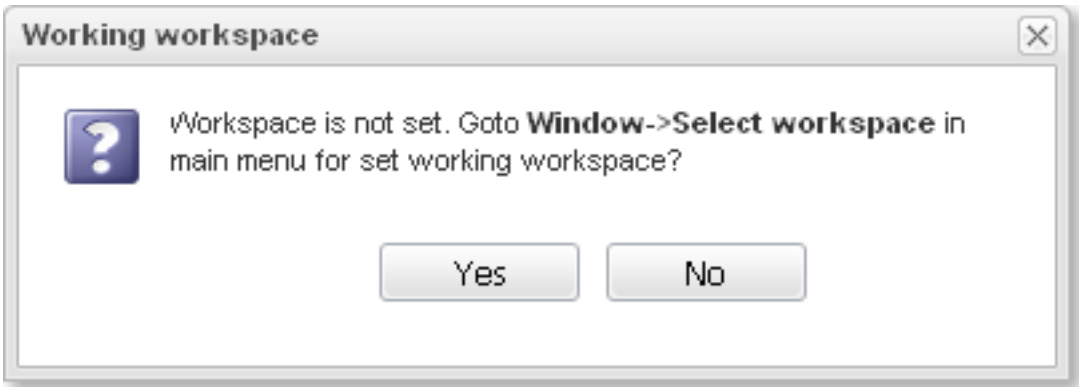


Illustration. The dialog window in case of no any workspaces set

Click **Yes** to open the **Workspace** dialog window.

Open Folder

To open a folder, do as follows:

- **Step 1:** Select your desired folder in the *Workspace Panel*.
- **Step 2:** Click the plus icon located next to the folder name, or double-click the folder item.

The path pointing to the current folder is displayed at the left corner of the *Status Bar*.

Open File

To open a file in the *Content Panel*, double-click the file item in the *Workspace Panel*. The history of opened files is saved in browser cookies, so you can see all the file tabs opened even after refreshing the browser window.

Refresh Selected Folder

To refresh content of the selected folder in the *Workspace Panel*, click the **Refresh Selected Folder** button on the toolbar, or go to **File > Refresh**.

View and Copy File's URL

To view the absolute URL of a file/folder and copy it to the clipboard, do as follows:

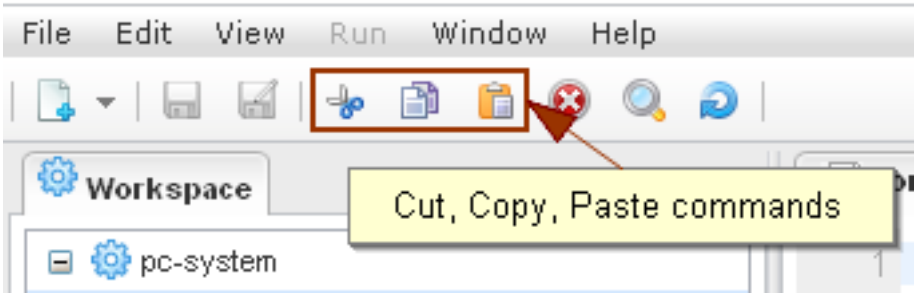
- **Step 1:** Select the appropriate item in the *Workspace Panel*.
- **Step 2:** Go to **View > Get URL...** from the top menu to open the **Item URL** window.
- **Step 3:** Click **OK** to accept copying the target URL to the clipboard.

Note

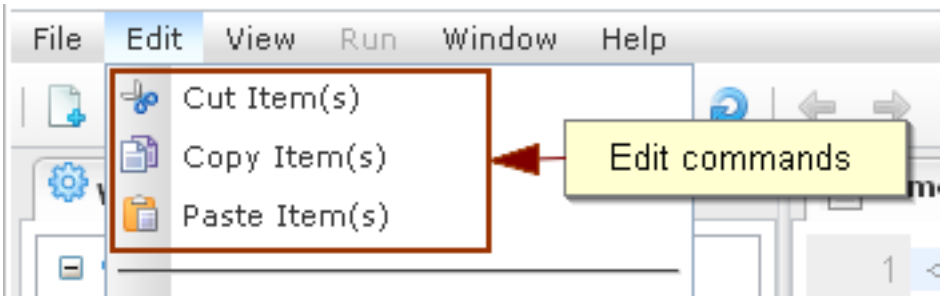
eXo IDE does not support copying multiple URLs to the clipboard at once. The "Get URL..." function is disabled in case of multiple files/folders selected.

Copy/Cut/Paste Files and Folders

- **Step 1:** Click your desired files/folders in the *Workspace Panel*.
- **Step 2:** Select corresponding icons on the toolbar;



Or, select **Edit** on the top menu and then click your desired action from the drop-down menu.



Use the **Paste** function to paste your selected files or folders from one folder to another.

Note

- The "Paste" function can only be enabled once "Cut" or "Copy" has been performed.
- eXo IDE does not support "Copy" and "Cut" functions for folders and/or files placed in the different folders. These functions are disabled if any item is selected at different levels.

Operations With Folder

Create Folder

To create a folder, do as follows:

- **Step 1:** Click your desired target folder in the workspace.
- **Step 2:** Click the **New** button on the toolbar, then select **Folder...** from the drop-down menu; or go to **File > New > Folder...** from the top menu.
- **Step 3:** Enter the folder name in the **Create folder** window.
- **Step 4:** Click **Create** to complete creating your new folder.

Note

- If any file is selected in the Workspace Panel, there will be a folder created in the parent folder of this file.
- In case no item is selected in the Workspace Panel, the created folder is placed in the root folder.
- In case of multiple selections, this function is disabled.

Delete Folder

To delete a folder, do as follows:

- **Step 1:** Click your desired folder in the *Workspace Panel*.
- **Step 2:** Click the **Delete Item(s)...** icon on the toolbar, or go to **File > Delete...** from the top menu.
- **Step 3:** Click **Yes** to confirm your deletion in the **Delete Item(s)** dialog.

Files opened in the *Content Panel* are closed after the folder containing them has been removed.

Note

- You cannot remove:

- Workspace root folder;

- Multiple items from different hierarchic levels at the same time.

- The "Delete Item(s)" function is disabled in case of selecting the root item and/or multiple items on different levels in the "Workspace Panel".

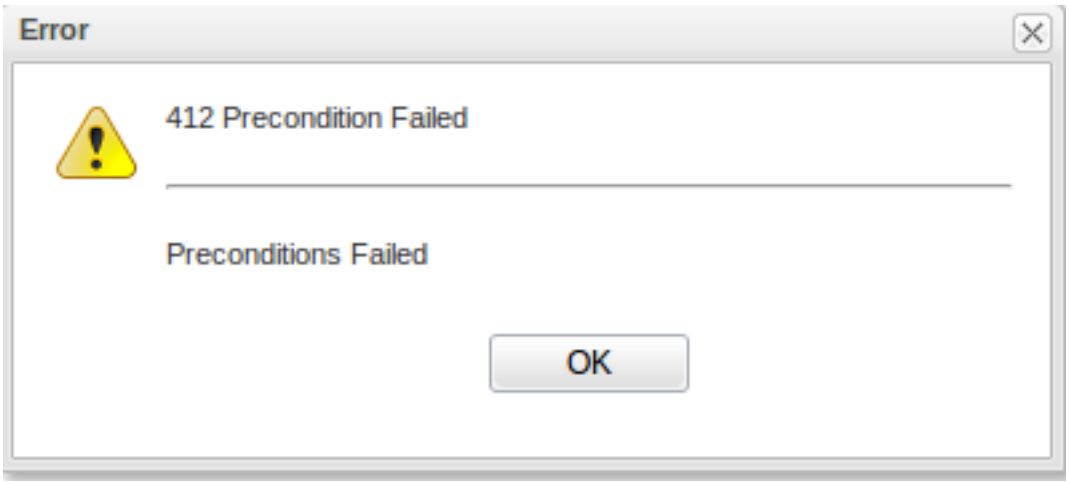
Rename Folder

To rename a folder, do as follows:

- **Step 1:** Click the folder you want to rename in the *Workspace Panel*.
- **Step 2:** Go to **File > Rename...** from the top menu to open the **Rename item** dialog.
- **Step 3:** Enter the new name into the **Rename item to:** field.
- **Step 4:** Click **Rename** to accept.

You can rename folders and sub folders containing opened but unsaved files. You can save them after renaming the folder, no data will be lost.

If the new folder name has been existing in the target folder, you will receive an error message as below.



Note

You cannot rename the workspace root folder.

Remember to undeploy all earlier deployed REST Services within the moved folder or sub-folder before renaming the folder.

Download Zipped Folder

If you want to download some folders from the *JCR Repository*, do as follows:

- **Step 1:** Click the target folder in the *Workspace Panel*.
- **Step 2:** Go to **File > Download Zipped Folder...** from the top menu.

The zip-archive with folder content is prepared on the server and sent to the client. There will be a suggestion window for you to save the archive on your local computer.

Note

The archive uses UTF-8 encoding in entry names. Be sure that your zip extractor supports UTF-8 in file names. The Windows Explorer does not support it, you may use 7zip to unzip the archive. In Linux, the unzip (6.0 or below) command also does not support UTF-8 (depending on distributive). If you use KDE 4 as DE, you may configure Dolphin to use archive as folder by going to "Settings > Configure Dolphin.. > Navigation", and then selecting the "Open archive as folder" checkbox.

Operations With File

Create File

To create a file, do as follows:

- **Step 1:** To create a new file click the **New** button on the left of the toolbar. Select the required file type in drop down menu.
- **Step 2:** Save the new file in some folders of the repository by selecting the target folder in the *Workspace Panel*, then click the **Save As** icon on the toolbar; or go to **File > Save As...** from the top menu; or press **Ctrl+S** keys.
- **Step 3:** Type the name of new file in the **Save file as...** window.
- **Step 4:** Click **OK** to accept creating the new file.

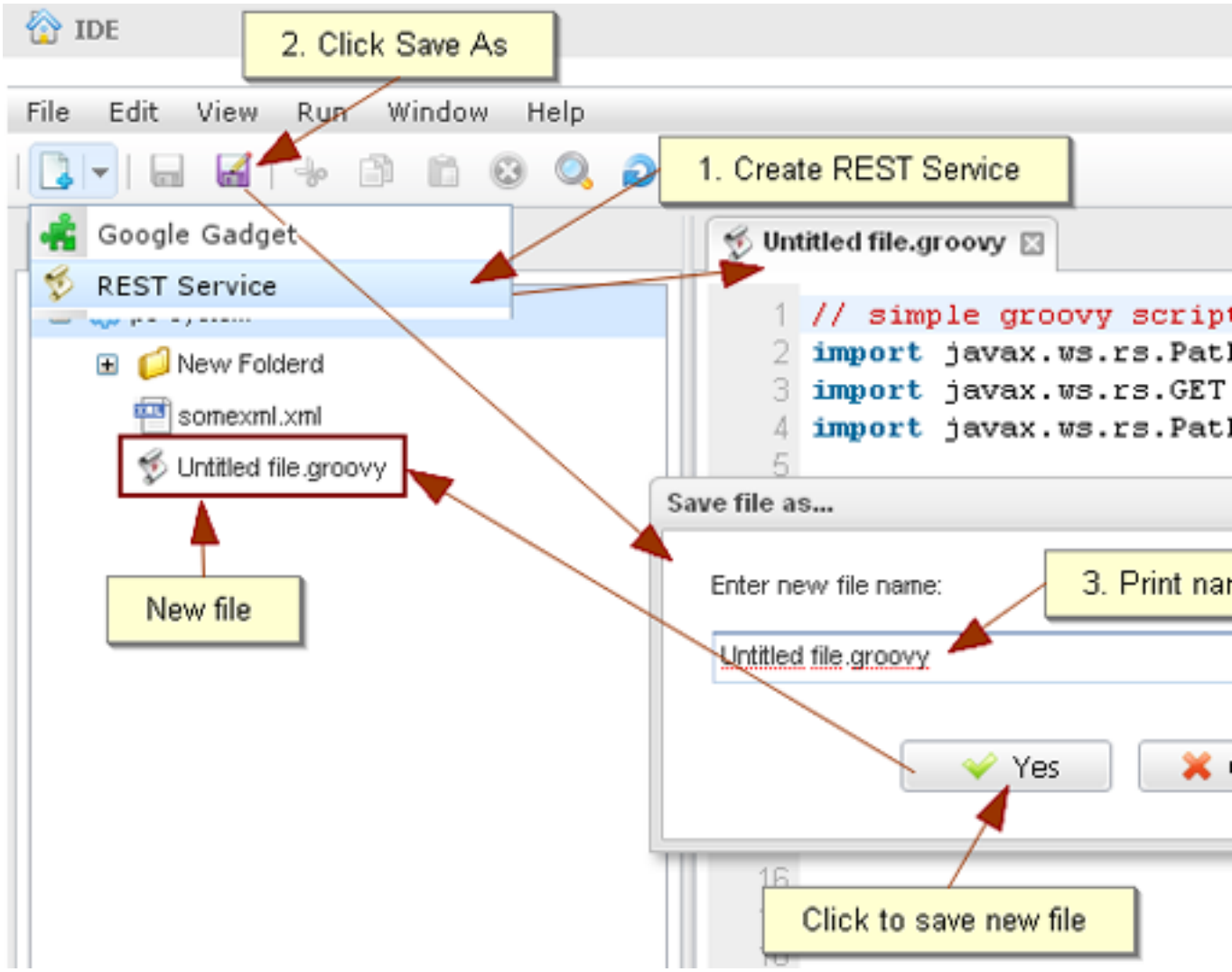


Illustration. Create new REST Service

Save/Save As... File

Once you have changed content of the opened file, its tab title will be marked by * and the **Save** button will be enabled. So, you can save changed content by clicking **Save** or press **Ctrl+S**, or going to **File > Save** from the top menu. To save the file into another folder or with another name, select the target folder in the *Workspace Panel*. Next, click the **Save As...** button on the toolbar; or select **File > Save As...** from the top menu.

Note

If there is any file with the same name in the target folder, it will be rewritten by the "Save As..." command.

Save All Files

If there is one or more files created, opened and changed in the *Content Panel* before, you can save all of them simply by selecting **File > Save All** from the top menu. The **Save All** command does not have any effect on new files.

Delete File

To delete a file, do as follows:

- **Step 1:** Select the target file in the *Workspace Panel*.
- **Step 2:** Click the **Delete Item(s)...** button on the toolbar; or go to **File > Delete...** from the top menu.
- **Step 3:** Confirm the deletion in the **Delete** dialog.

Files opened in the *Content Panel* are closed after being removed. You can view the full path to the opened file in the tip of file tab in the *Content Panel*.

Note

In eXo IDE, you cannot delete items from different folders simultaneously. The "Delete..." function is disabled in such cases.

Rename File

To rename a file, do as follows:

- **Step 1:** Select your desired file in the *Workspace Panel*.
- **Step 2:** Go to **File > Rename...** from the top menu.

- **Step 3:** Enter the new name into the **Rename item to** field.
- **Step 4:** Click **Rename** to take effect.

Note

If your new file name has been existing in the target folder, you will receive an error message.

Do not forget to undeploy the REST Services if they were deployed before renaming.

Change MIME Type of File

You can change the MIME type of the closed file as follows:

- **Step 1:** Click the target file in the *Workspace Panel*.
- **Step 2:** Go to **File > Rename...** from the top menu.
- **Step 3:** Select the new MIME type from **Select mime-type** combo-box, or type another non-registered MIME type in this field.
- **Step 4:** Click **Rename** to change the MIME type.

Note

You cannot open the file with the non-registered MIME type in eXo IDE.

Lock File

This feature is used to protect your files from being changed or removed by another users. You can lock or unlock the file by clicking the **Lock** or **Unlock** icon on the toolbar, or going to **Edit > Lock/Unlock File** on the top menu.

The "Lock" icon is located next to the title of the locked file in the *Workspace Panel* and at the title of such file tab in the *Content Panel*:

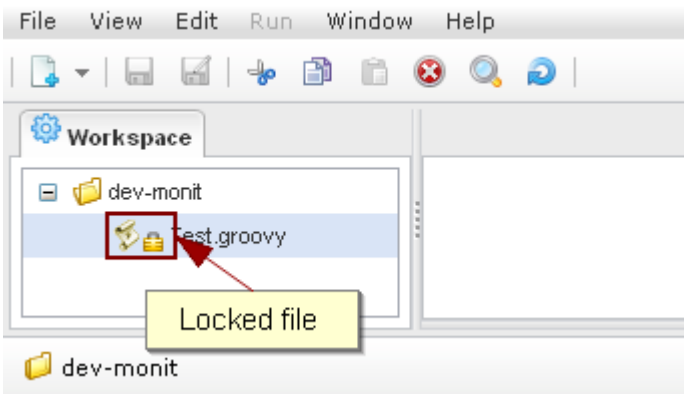


Illustration. Locked file in Workspace Panel

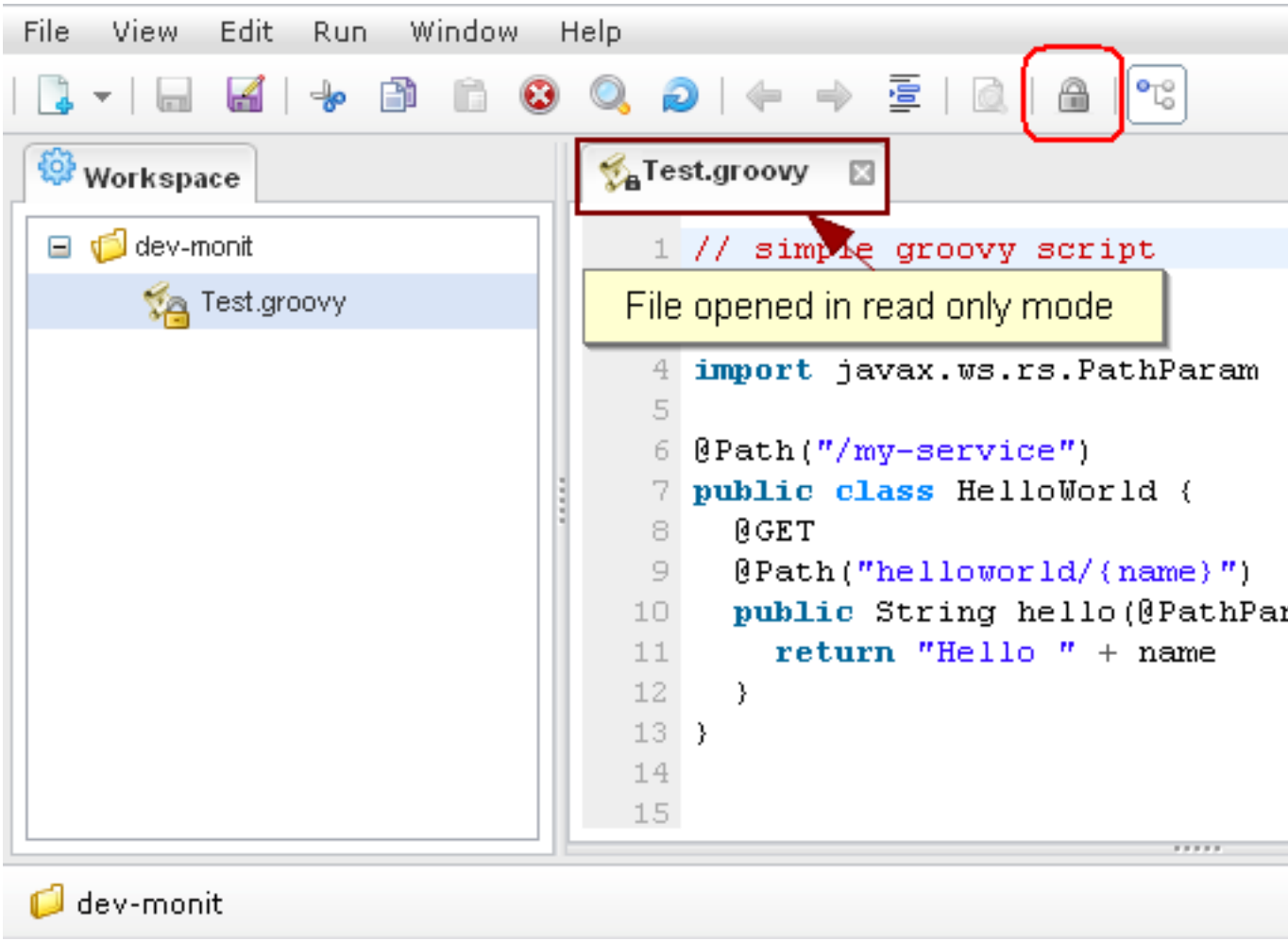


Illustration. Locked file in Content Panel

Another users can open the locked files to read only, but can not edit, save, delete, move or rename.

> If you open the locked file with the WYSIWYG Editor, you can edit its content, but the "Save" function is disabled.

Open Files with the Non-default Editor

HTML and Google gadget files can be opened in either the default *Code Editor* or *WYSIWYG Editor*. To open a file in the *WYSIWYG Editor*, do as follows:

- **Step 1:** Click the needed HTML or Google Gadget file in the *Workspace Panel*.
- **Step 2:** Go to **File > Open With...** on the top menu.

- **Step 3:** Select **WYSIWYG Editor** and then click **Open** to show the selected file in the *WYSIWYG Editor*.

If the target file has been opened in the *Content Panel* before, the confirmation dialog is shown for you to affirm reopening it:

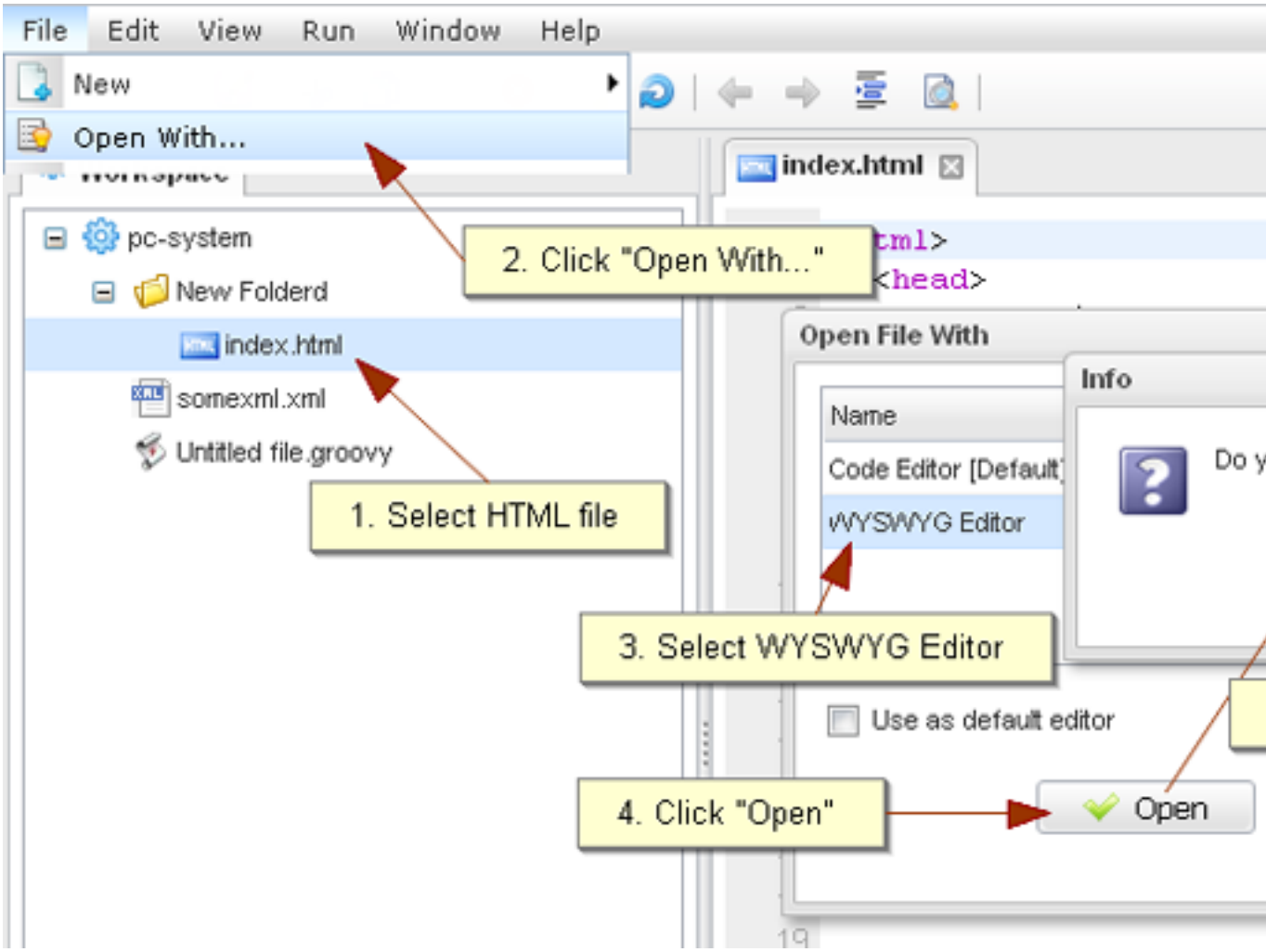


Illustration. Open the HTML file in the WYSIWYG editor

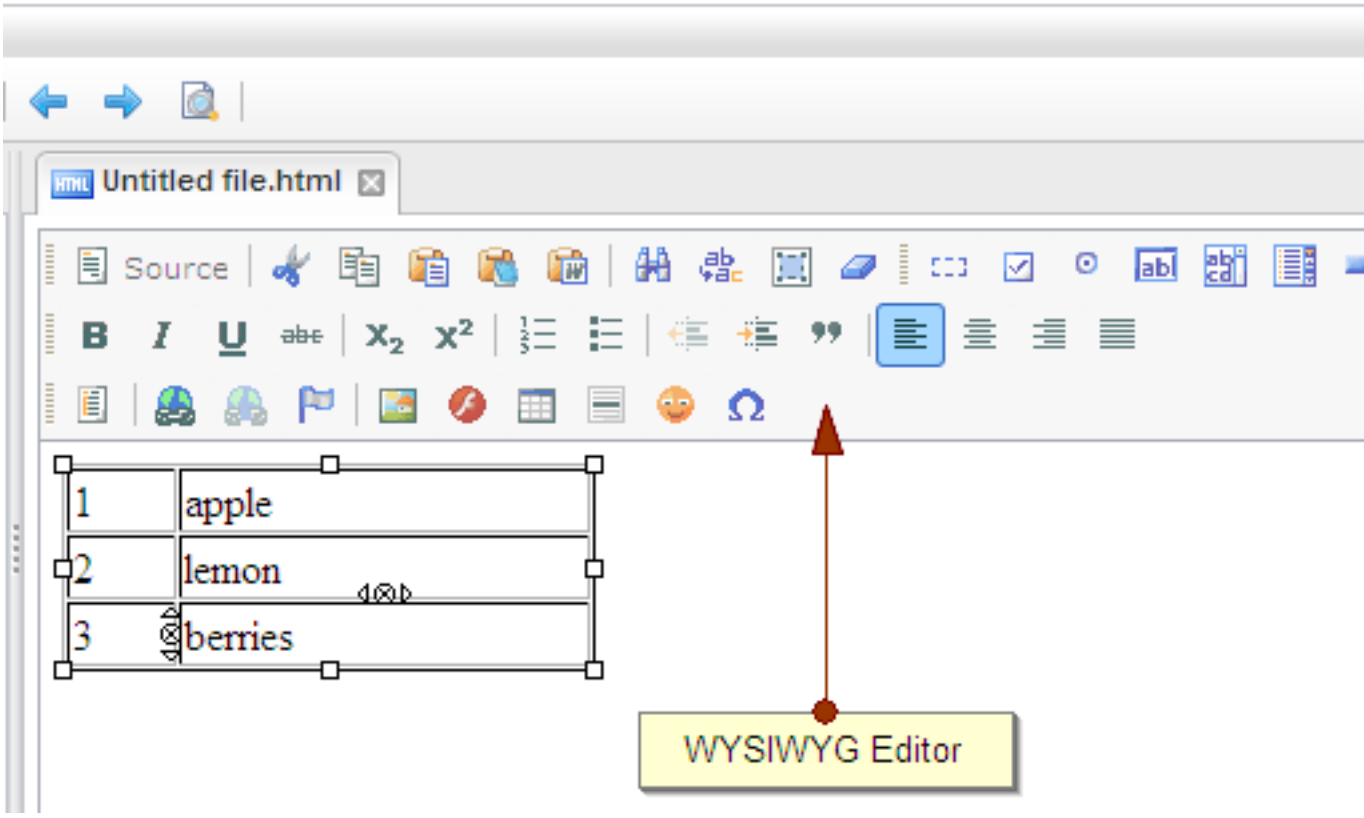


Illustration. Opened the HTML file in the WYSIWYG editor

If you want to change the default editor, simply click your desired editor and mark the **Use as default editor** checkbox in the **Open File With** dialog.

The new settings are stored in the *User Settings File* at the server that will be the default editor in next sessions. The default editor is marked by the "Default" text in the **Open File With** window.

Go to Folder

You can quickly find the opened file in the *Workspace Panel* using the **Go to Folder** function as follows:

- **Step 1:** Select the appropriate file tab in the *Content Panel*.
- **Step 2:** Go to **View > Go to Folder** on the top menu.

Your selected file will be highlighted in the *Workspace Panel*. Similarly, you can localize the found file from the *Search Tab*.

Download File from Server

If you want to get some files from the *IDE Workspace*, do as follows:

- **Step 1:** Select the target file in the *Workspace Panel.*
- **Step 2:** Go to **File > Download...** on the top menu.

The file will be sent to the client and the suggestion window will be shown for you to save the file into your local device.

Open Local File

In eXo IDE, you can edit some local files with the registered type within the *Code Editor* or *WYSIWYG Editor*. During opening, you can directly define the correct file type. To edit the local file content in eXo IDE, do as follows:

- **Step 1:** Go to **File > Open Local File...** on the top menu.
- **Step 2:** Click the **Browse...** button in the **Open file** dialog to select the required local file.
- **Step 3:** Check the file's MIME type, based on the file extension. If needed, select another MIME type from the list.
- **Step 4:** Press the **Open** button to open the selected file content in the default editor of the *Content Panel* with the local file name.

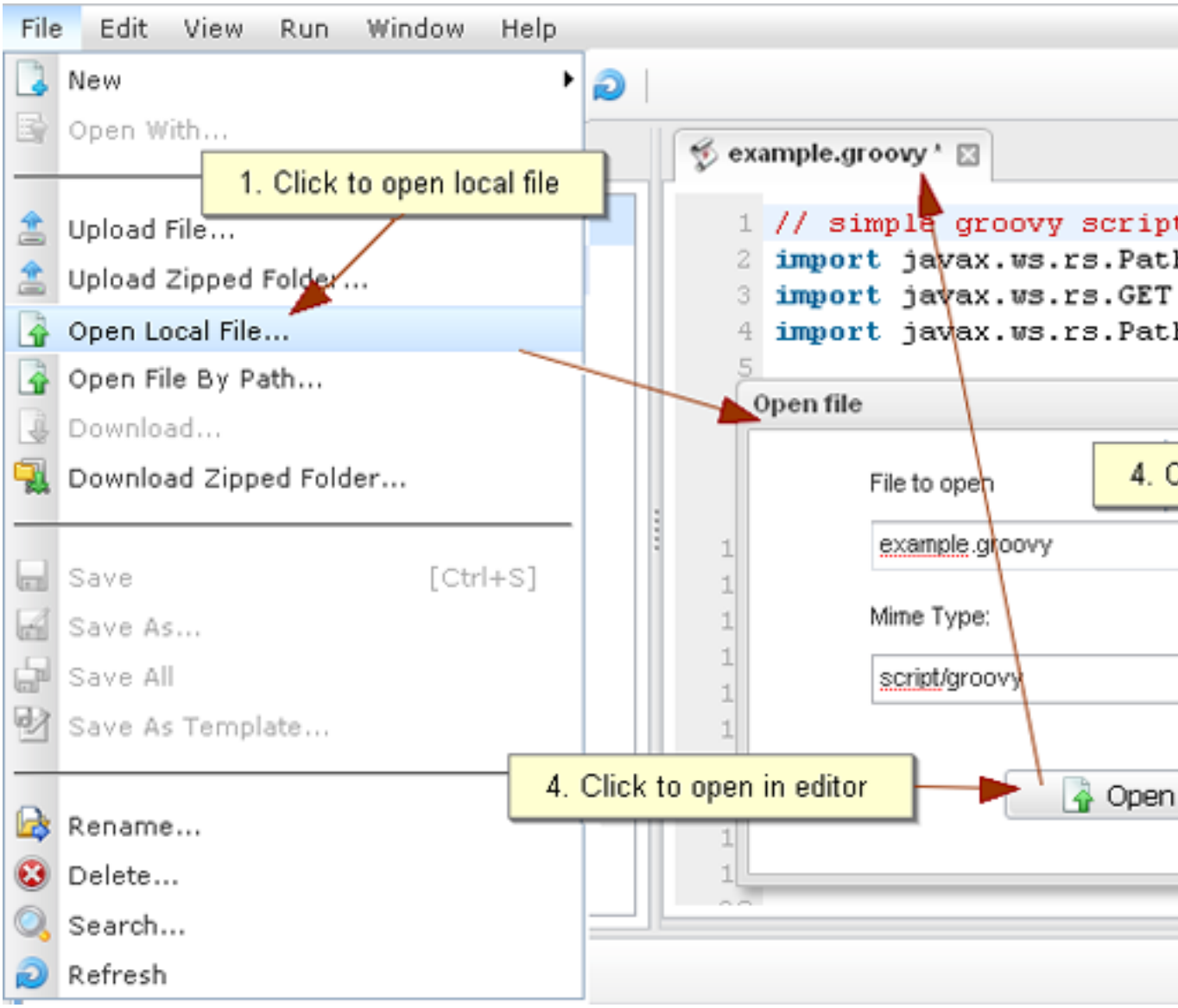


Illustration. Example of opening the local REST service

The following table is the list of MIME types registered in the eXo IDE.

Table. Registered MIME Types

Table 4.1.

File Type	MIME Type	Default File Extension	Default Editor
Text	text/plain	.txt	Code Editor
XML	text/xml; application/xml	.xml	Code Editor
REST Service	application/x-jaxrs+groovy	.grs	Code Editor

File Type	MIME Type	Default File Extension	Default Editor
POGO	application/x-groovy	.groovy	Code Editor
Template	application/x-groovy+html	.gtmpl	Code Editor
HTML	text/html	.html	Code Editor
Google Gadget	application/x-google-gadget	.xml	Code Editor
JavaScript	text/javascript; application/javascript; application/x-javascript	.js	Code Editor
CSS	text/css	.css	Code Editor
Netvibes Widget	application/x-uwa-widget	.html	Code Editor
Data Object	application/x-chromattic+groovy	.groovy	Code Editor
<unrecognized>	binary/octet-stream		

Note

To load the Google Gadget file, you should set the MIME Type as "application/x-google-gadget" in the "Open file" dialog as figured out in the below illustration:

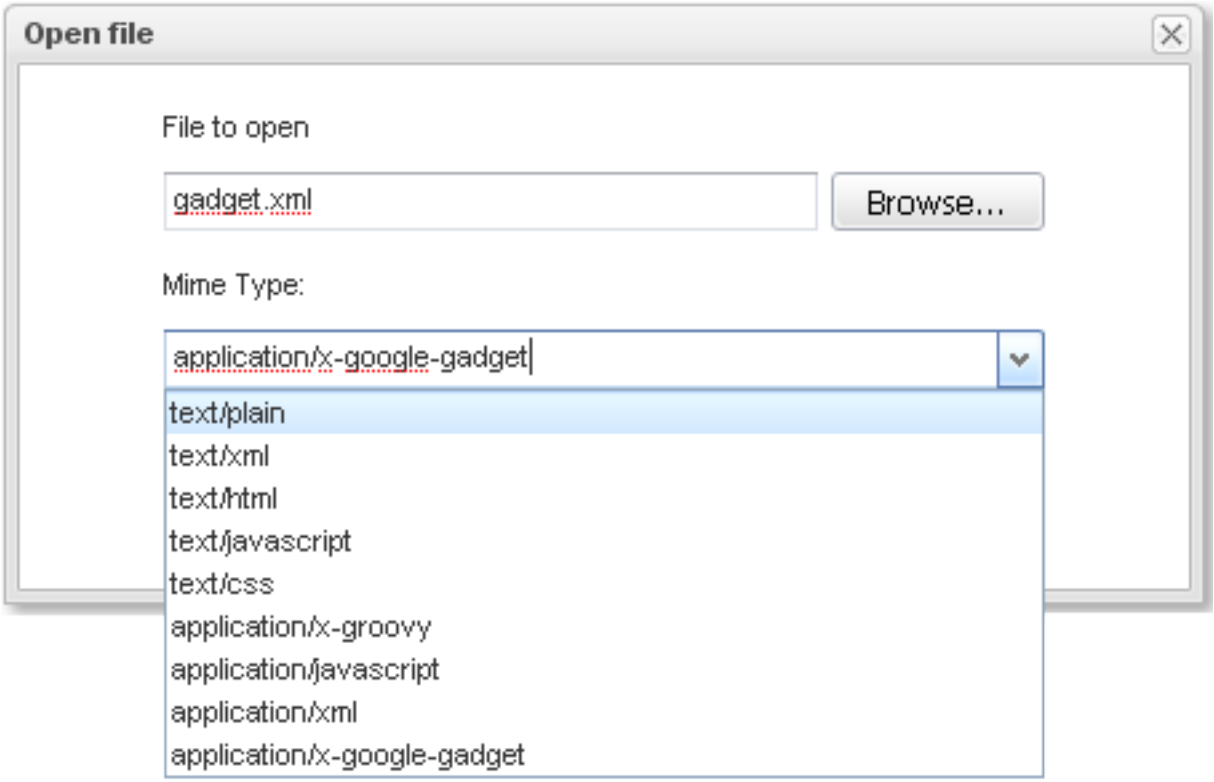


Illustration. Open the Google Gadget file

Upload File

To upload a file directly to the *IDE Workspace* on the server, go to **File > Upload File...** on the top menu. For example, to upload the **gadget.xml** file to the **Gadget** folder, do as follows:

- **Step 1:** Select the **Gadget** folder in the *Workspace Panel.*
- **Step 2:** Go to **File > Upload File...** on the top menu.
- **Step 3:** Click the **Browse...** button to select the **gadget.xml** file from the local device in the **File upload** dialog.
- **Step 4:** Select the appropriate MIME type as *application/x-google-gadget*.
- **Step 5:** Press the **Upload** button.

The **gadget.xml** gadget is uploaded to the server and placed in the **Gadget** folder.

Upload Zipped Folder

To upload a zipped folder to the *Workspace* on the server, the folder must be packaged as the .zip archive.

- **Step 1:** Select the parent folder item in the *Workspace Panel.*
- **Step 2:** Go to **File > Upload Zipped Folder...** on the top menu.
- **Step 3:** Click the **Browse...** button to select the zipped folder from the local device in the **Upload folder** dialog.
- **Step 4:** Press the **Upload** button.

The zipped folder is uploaded to the sever and unzipped in the folder selected in step 1.

Open File by Path

To open a file by its path, for example **Example.groovy**, do as follows:

- **Step 1:** Go to **File > Open File By Path...** on the top menu.
- **Step 2:** Enter the path of the target file "*http://127.0.0.1:8080/rest/private/ide-vfs-webdav/repository/production/Example.groovy*" into the **File URL** field of the **Open file by path** dialog.
- **Step 3:** Click the **Open** button to show the **Example.groovy** file in the new file tab of the *Content Panel*.

You can select **View > Get URL...** on the top menu to obtain the file path.

Note

The target file should be placed at the same domain as IDE.

Use File Template

File Template is stored in the IDE registry file, and then can be used to create another files with the same MIME type.

Save File as Template

To save the selected file as a template with the same MIME type, do as follows:

- **Step 1:** Go to **File > Save As Template...** on the top menu.
- **Step 2:** Enter the template name.
- **Step 3:** Define the brief description of the template.
- **Step 4:** Click **Save** to accept.

You can also save the uploaded file as template.

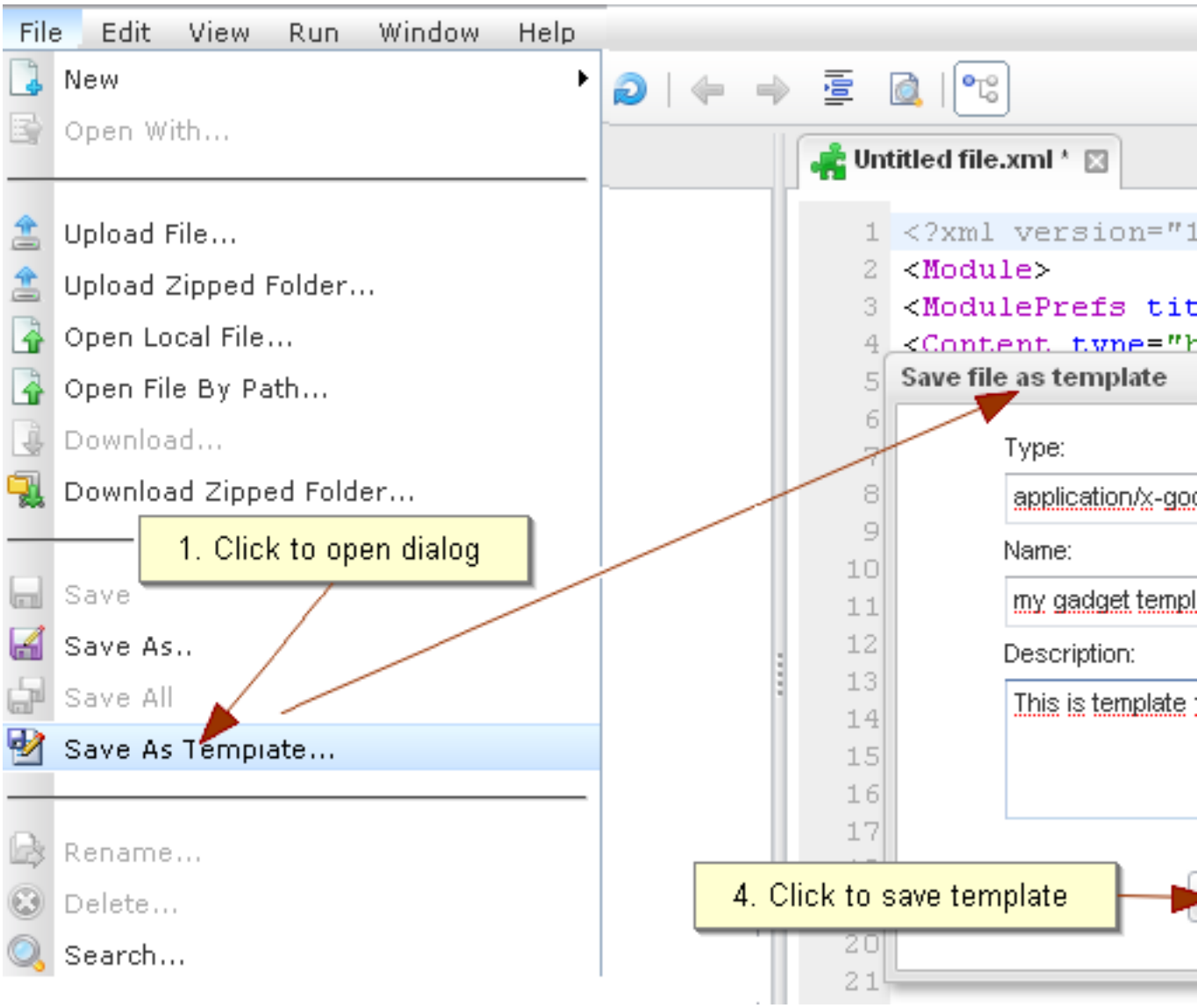


Illustration. Save file as template

Create File from Template

The **File From Template...** function enables you to create a file from the predefined template. For example, to create a simple gadget XML file, do as follows:

- **Step 1:** Click the **New** button on the toolbar and select **File From Template...** from the pop-up to open the dialog, or press **Ctrl+N** hotkeys.
- **Step 2:** Select your desired type and template for the file.
- **Step 3:** Enter the name into the **File Name** field, and click **Create** to complete adding your new file. The newly created file is opened automatically in the *Content Panel*.

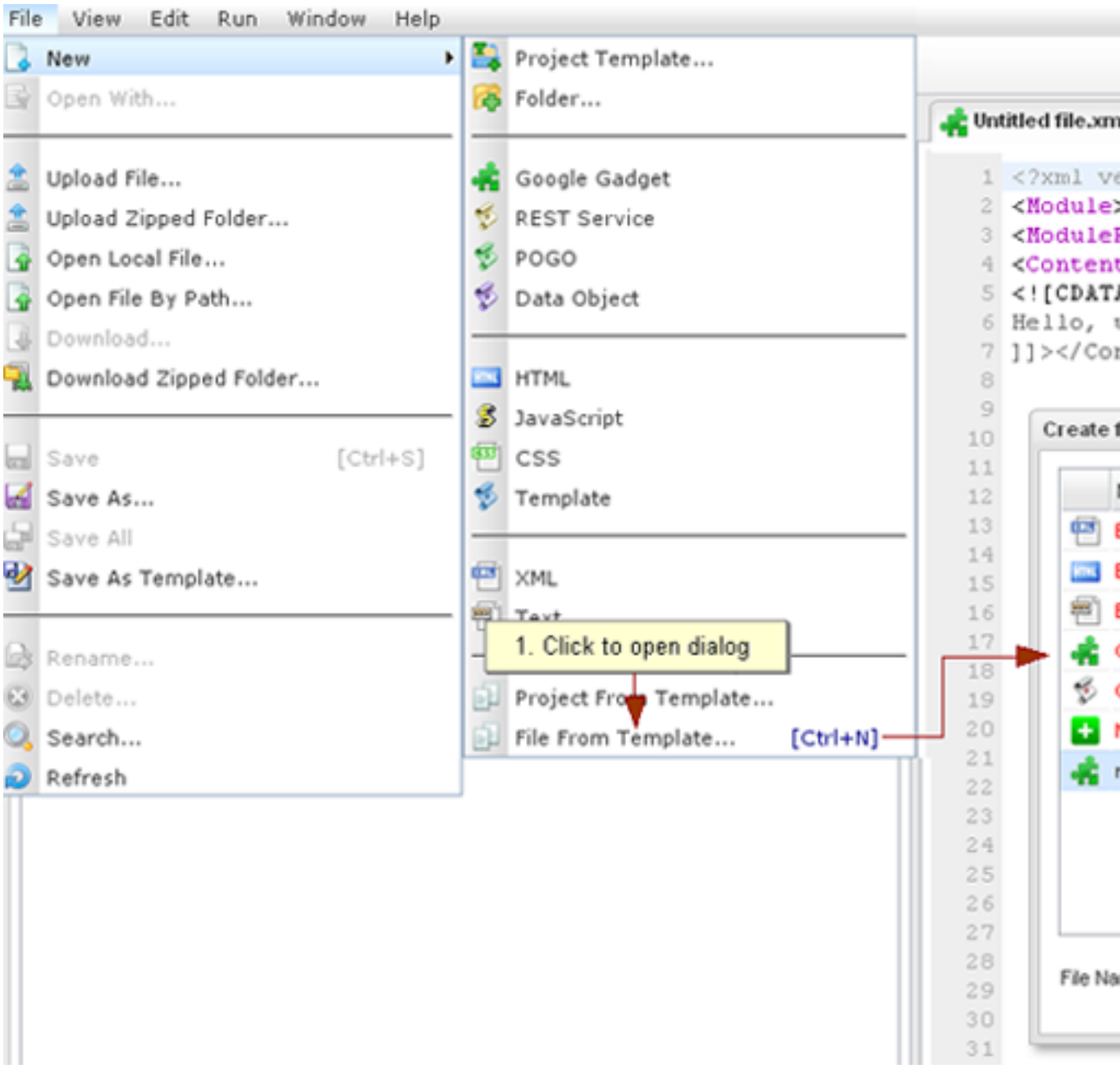


Illustration. Create the gadget from template

Remove Non-default File Template

To remove a non-default file template, do as follows:

- **Step 1:** Click the **New** button on the toolbar.
- **Step 2:** Select **File From Template...**; or press **Ctrl+N** hotkeys.
- **Step 3:** Select the needed non-default template in black and click **Delete**.

The confirmation dialog will appear for you to verify your deletion.

After deleting the file template which has been used in the project template, a new project will be created from such project template without this file.

Use Project Template

Project Template helps you create folders and files with the predefined structure and content. You can create, update or remove your project template and add new folders and files from your template.

Create Project Template

To create a project template, do as follows:

- **Step 1:** Go to **File > New > Project Template...** on the top menu; or click the **New** button on the toolbar and select the **Project Template...** command:

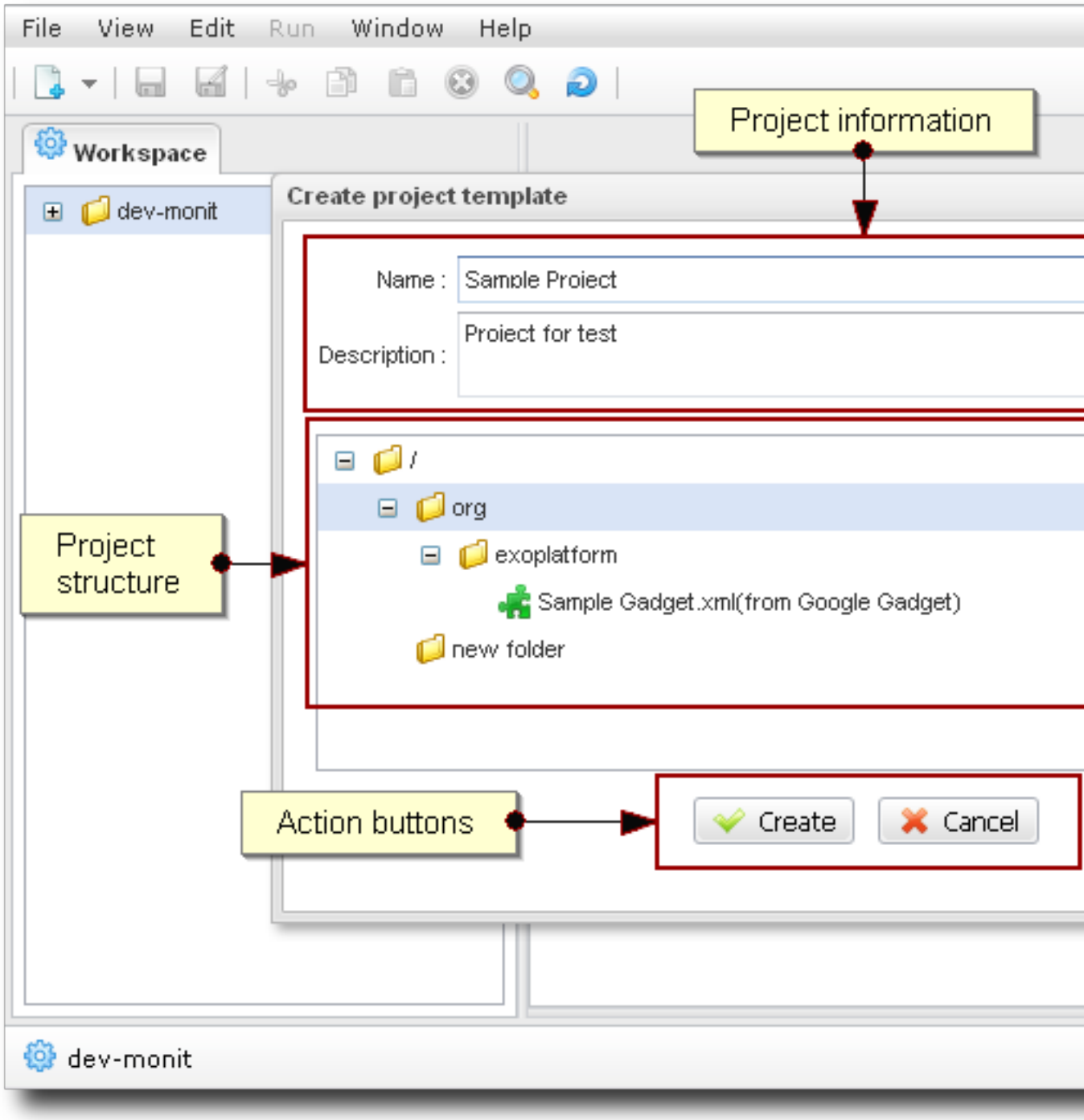


Illustration. "Create project template" dialog window.

- **Step 2:** Enter the name and brief description in the **Create project template** form, or edit its structure by clicking buttons **Add Folder**, **Add File**, **Delete** at the right part of the **Create project template** dialog.
- **Step 3:** Click **Create** button to complete your project template.

Modify the Project Template Structure

You can make changes for your project template structure by adding new folders/files or deleting some elements. To make modifications for your project template structure, do as follows:

Step 1: Open the **Create project template** window as described in Section "Create Project Template" above.

Step 2: Select the target folder in the project template tree.

Step 3: Add a new folder: 1. Click the **Add Folder** button to open the **Add folder** window.

2. Enter your desired folder name into the **Name of new folder** field and then click **Add**.

New folder then will be added to the project template.

Step 4: Add a new file:

1. Click the **Add File** button to open the **Add file** window.

2. Select your desired file template in the templates list.

3. Enter the file name into the **File Name** field, and then click the **Add** button.

Step 5: Delete some elements:

1. Select your desired item (folder/file) in the template project tree.

2. Click the **Delete** button.

Create Project From Template

To create a project from template, do as follows:

- **Step 1:** Select the parent folder in the *Workspace Panel*.
- **Step 2:** Click the **New** button on the toolbar and then select **Project From Template...** from the drop-down menu; or go to **File > New > Project From Template...** from the top menu.
- **Step 3:** Select your desired project template from the templates list.
- **Step 4:** Enter the project name into the **Project name** field.
- **Step 5:** Click the **Create** button.

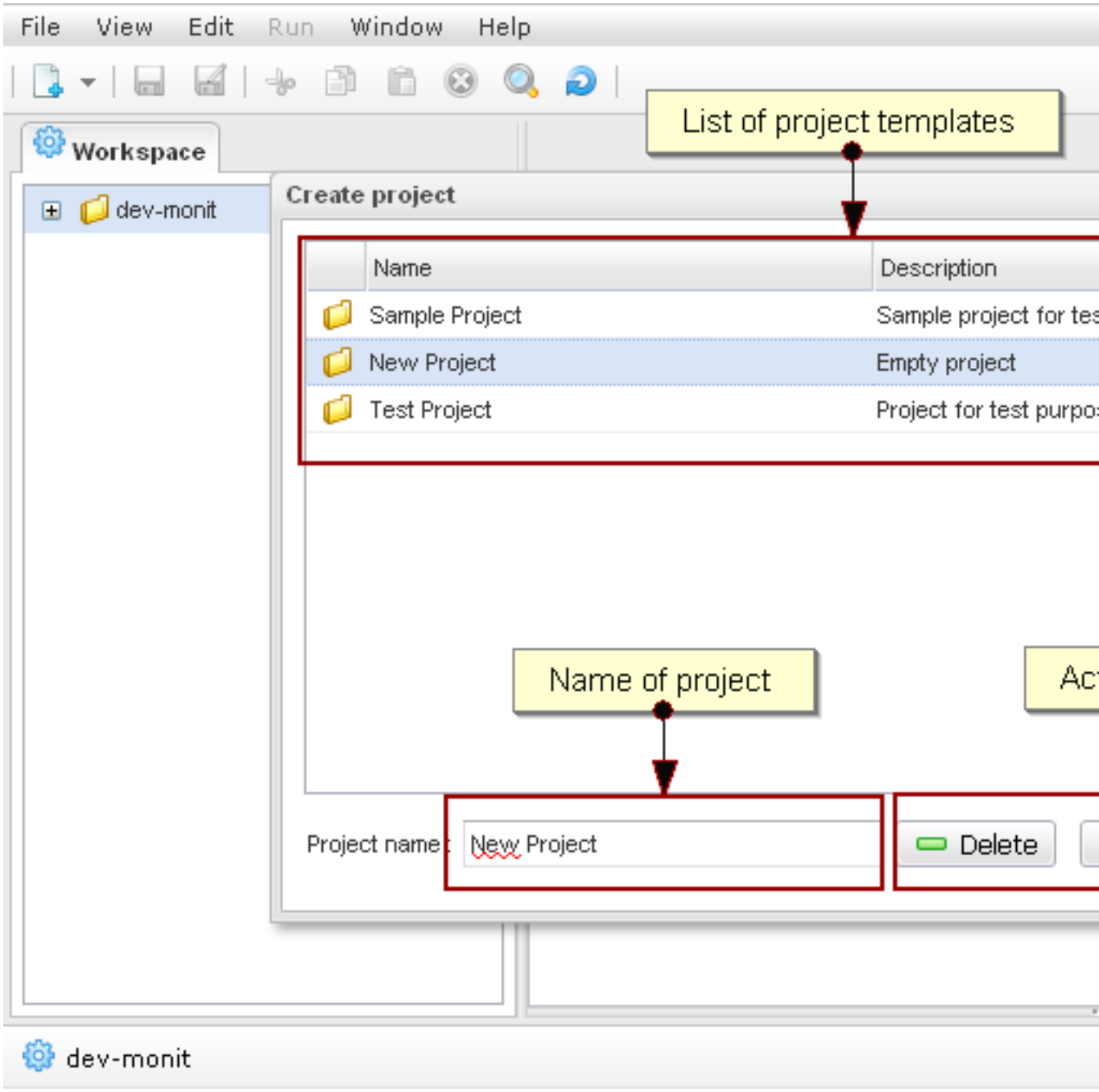


Illustration. Create project from template

Remove Project Template

To remove a project template, do as follows:

- **Step 1:** Click **New > Project From Template...** on the toolbar; or go to **File > New > Project From Template...** from the top menu.
- **Step 2:** Select your desired target project template from the list.

- **Step 3:** Click the **Delete** button.

Open/Select/Close/Browse tabs

To switch to the necessary file, click the tab title with the appropriate name. To close the file, just click the **X** button on the tab title. Also, you can use the tab browsing buttons **<**, **>** and down arrow at the top right corner of the *Content Panel* as below:

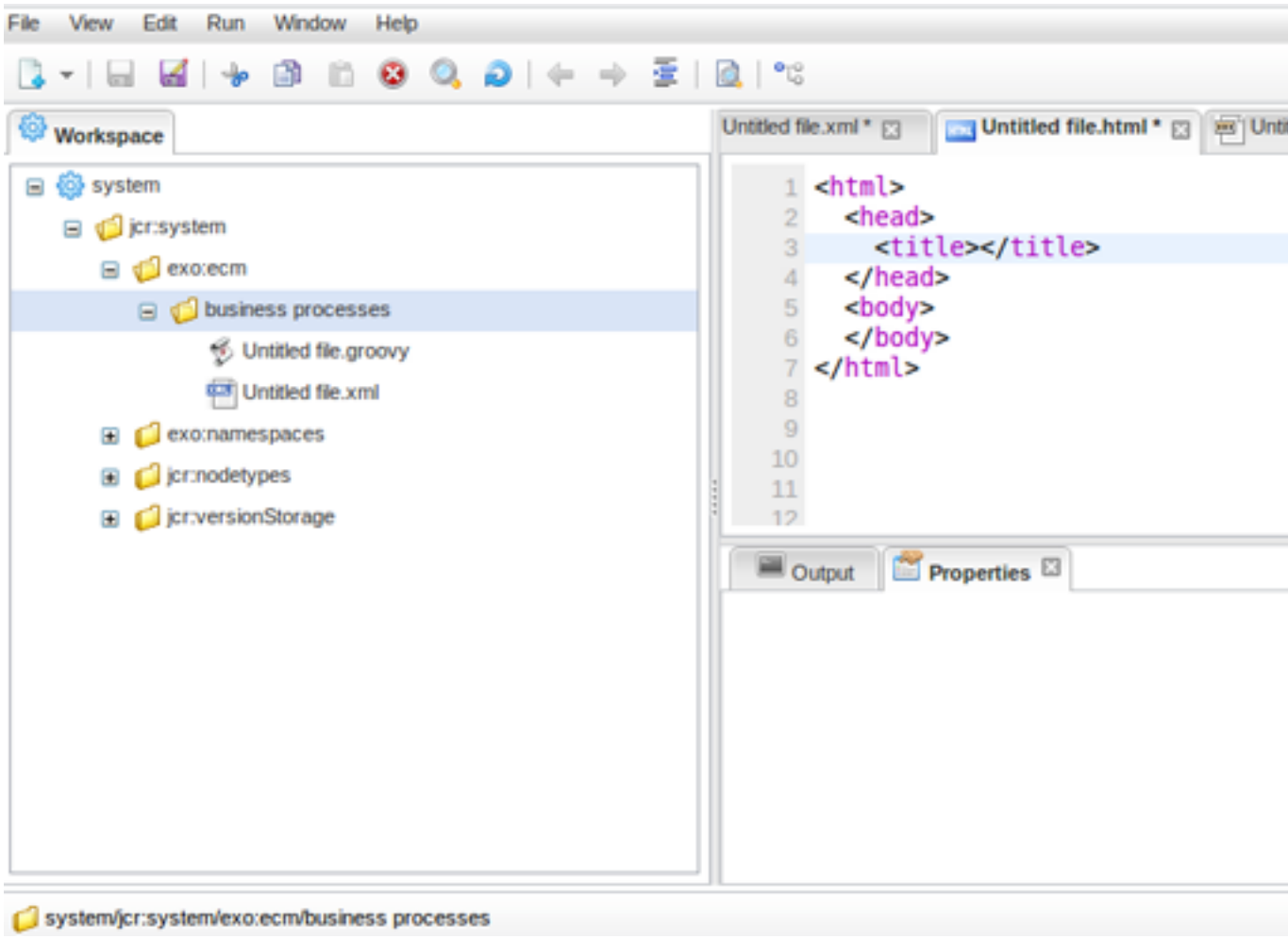


Illustration. File tab browsing buttons.

Note

You can view the full path to the opened file in the tip of the tab title.

You can see some useful commands represented with special buttons on the toolbar and in the **Edit** on the top menu: **Undo/Redo Typing**, **Format**, **Show/Hide Line Numbers**:

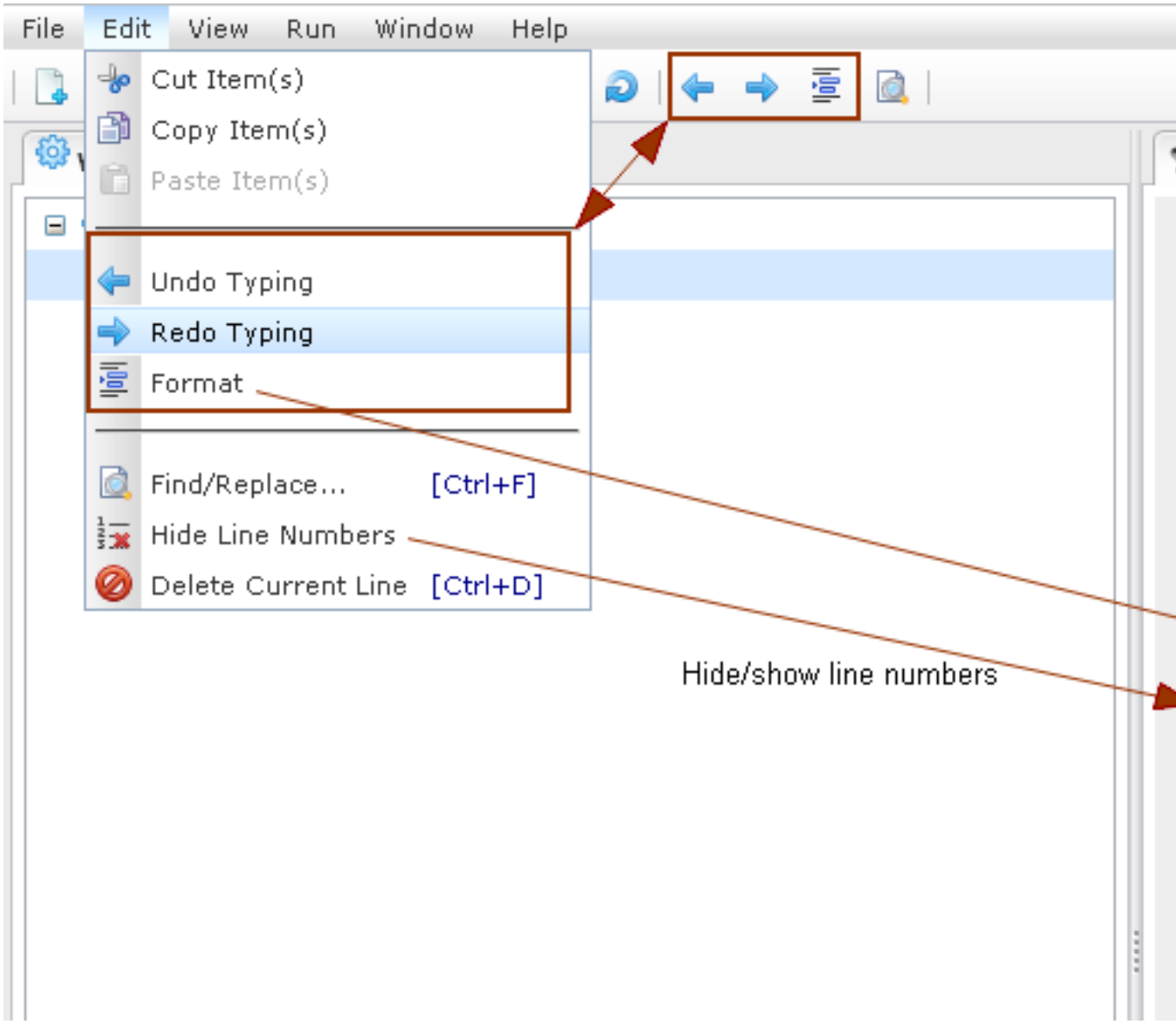


Illustration. Undo/Redo, Format and Show/Hide Line Numbers on the toolbar and on the top menu.

Undo/Redo Editing

Your actions performed with the text are added to the history as a distinct batch after each pause, saving you from fear of doing something wrong. So, you can restore your file content by using the Undo or Redo functions.

Undo

The **Undo** command in eXo IDE helps you obtain your desired changes, such as formatted text, moved blocks, deleted text by following either of two handy ways:

- Press **Ctrl+Z**.

- Click the **Undo Typing** button on the toolbar.

Note

The "Undo" function is disabled in case of no new changes in your text. To undo the Undone action, use the "Redo" function.

Redo

After you have undone your performed action on the text but unfortunately what recovered is not satisfactory, simply click the **Redo** function to set back. The **Redo** function is used in the following cases:

- Get exactly whatever the **Undo** command does. For example, if you type some text, **Undo** deletes the text, and **Redo** recovers the deleted text.

- If you use **Undo** to recover your deleted text, **Redo** will delete the text again.

The **Redo** function can be performed via one the two following ways:

- Press **Ctrl+Y**

- Click the **Redo Typing** button on the toolbar.

Note

- "Redo" is available only after "Undo" is performed.
- In the "Code Editor", the "Undo/Redo" functions are disabled in case of no changes in the "Undo/Redo" history.
- Each file opened in the editor has its own history.

Edit Files in the Code Editor

Once a file has been created/opened, its content is displayed in the separate tab on the *Content Panel* in the default *Code Editor*. eXo IDE Code Editor parses and colors the code, you can indent code, search text, locate open/close braces and brackets, see line numbers and more...

Format File

In the *Code Editor*, **Format** is to adjust row indentations of the HTML, Groovy, JavaScript, XML content with two spaces by default. To do this, select **Format** from the toolbar, or go to **Edit > Format** on the top menu.

Show/Hide Line Numbers

To show line numbers in the *Code Editor*, go to **Edit > Show Line Numbers** from the top menu. The **Show Line Numbers** is turned into **Hide Line Numbers**. You will see the left vertical grey area in all opened files.

Select "Hide Line Numbers" to hide line numbers. The last selected settings are saved in the browser cookies for next sessions.

Go to Line

To go to the exact line quickly, do as follows:

- **Step 1:** Open the **Go to Line** dialog by following one of the listed below ways:

(1) select **Edit > Go to Line...** from the top menu,

or (2) double-click the area showing the current cursor position in the *Status Bar*,

or (3) press **Ctrl+L**.

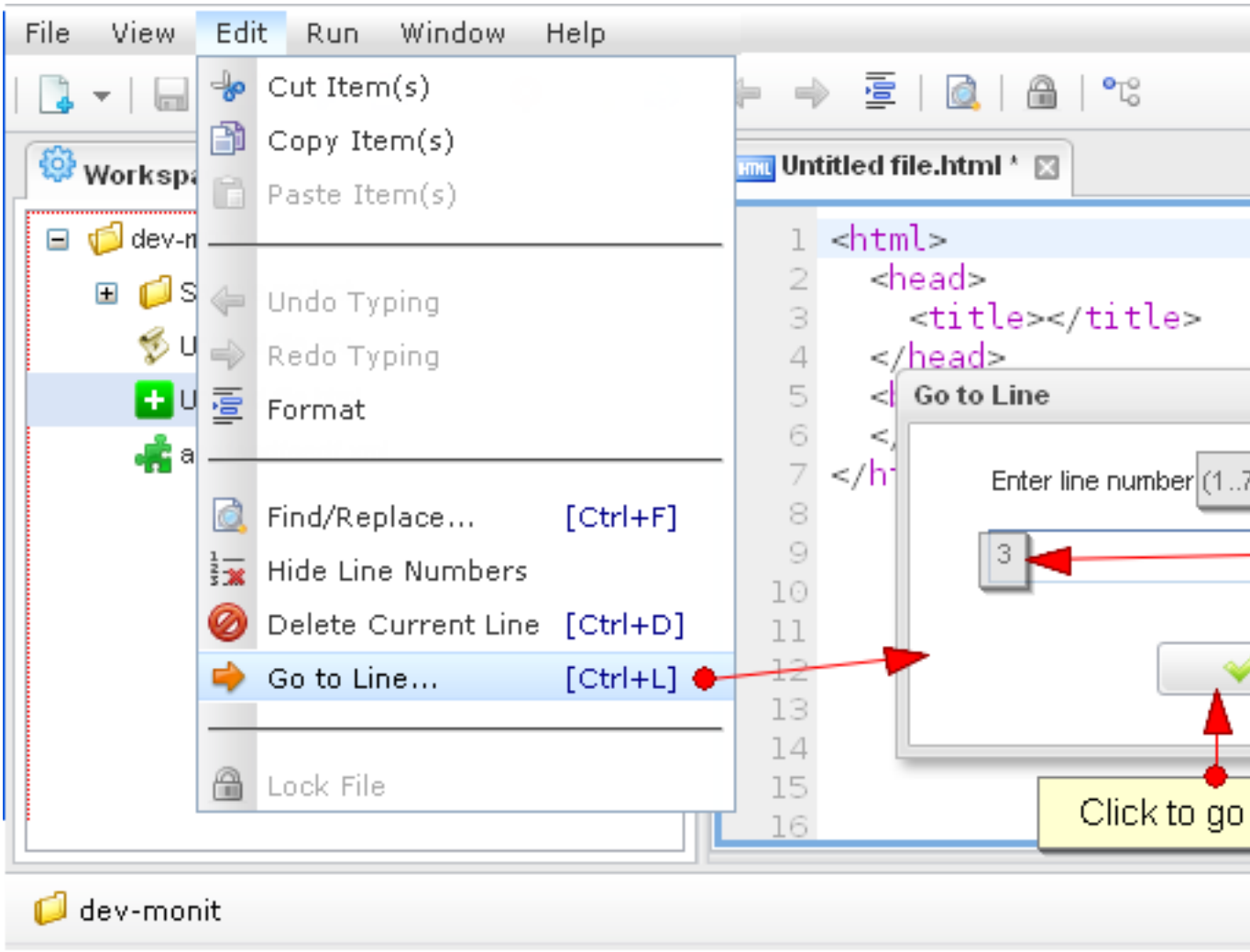


Illustration. Go to line

- **Step 2:** In Line number, type a number within the allowable range and click **Go**. If you enter the number out of range, there will be an error message.

> **Note**
>
> If "Ctrl+L" does not work, use the hot key manager to customize your hotkeys.

Watch the cursor position in the *Status Bar*

When the editor is opened, the current cursor position is shown in the *Status Bar*. The first number is the line number and the second is the column position.

Delete line

You can quickly delete the current line in the editor as follows:

- **Step 1:** Put the cursor on the needed line.
- **Step 2:** Use **Edit > Delete Current Line** command from the top menu, or press **Ctrl+D** to remove the highlighted line from the file.

Find/Replace

This function is commonly used in text files to quickly search for a specific word/phrase and to replace the text as follows:

- **Step 1:** Open your desired file.
- **Step 2:** Open the **Find/Replace** window by following one of the ways described below:

(1) select the **Find/Replace...** icon on the toolbar,

or (2) go to **Edit > Find/Replace...** from the top menu,

or (3) press **Ctrl+F**.

- **Step 3:** Enter the word/phrase for which you want to search into the **Find** field.
- **Step 4:** Click the **Find** button to do the search.

Once the text has been found, and if you want to replace the found phrase, continue doing the next steps:

- **Step 5:** Enter the replacement text into the **Replace with** field.
- **Step 6:** Click the **Replace** or **Replace/Find** button to replace the found word/phrase, or select **Replace All** to replace all.

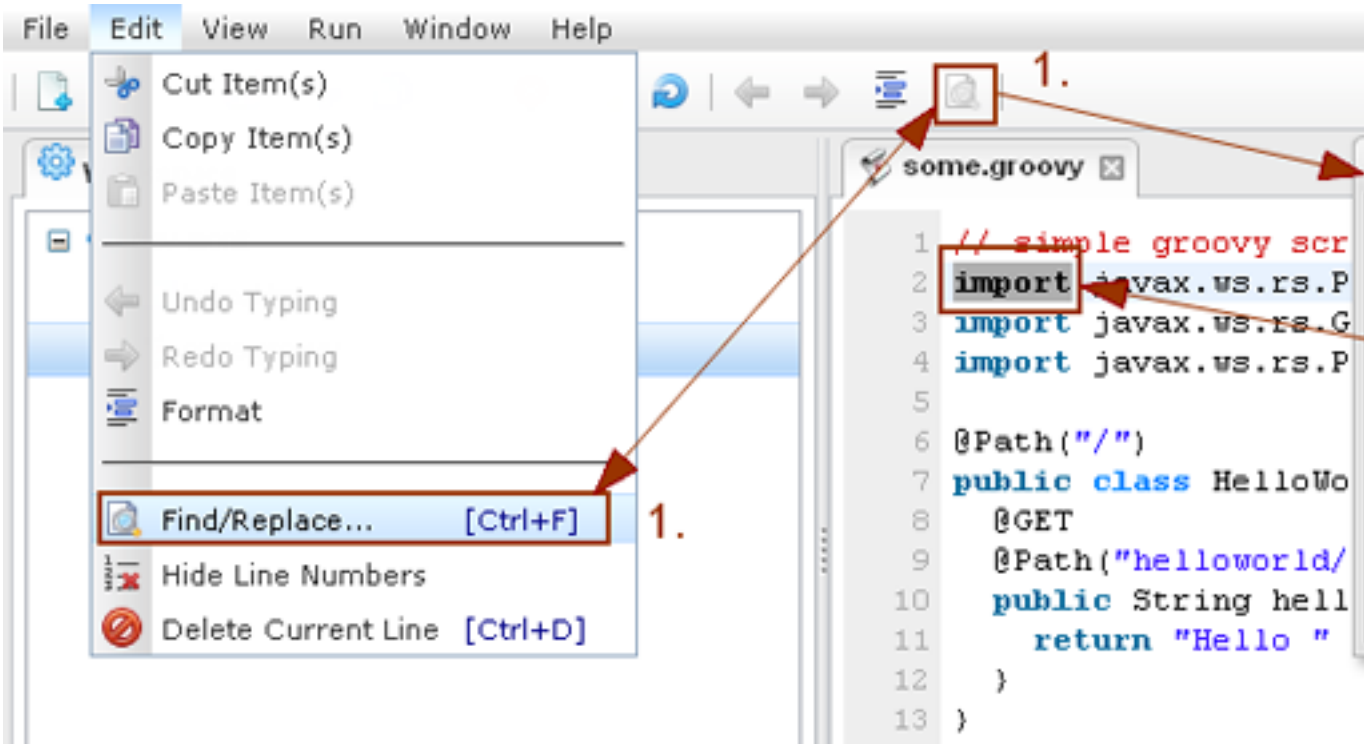


Illustration. Find and replace operations with file's content

Note

- All functions in the "Find/Replace" window are only performed as from the current position of the cursor to the end of the file.
- "Replace" and "Replace/Find" are only active if the text is found.
- Check the "Case sensitive" field to do the search based on differing the use of uppercase and lowercase letters.
- If you do the "Find/Replace" action with one file and then switch to another, this window will save the status of performed actions of the opened file in each editor.

Code Autocomplete

This feature enables you to autocomplete the word, including keywords, statements, declared variables, properties and methods, functions, objects, classes, tags, attributes, annotations in context of JavaScript, HTML, XML, Google Gadget, REST Service, Template, POGO, Netvibes Widget, Data Object files simply by clicking **Ctrl+Space** hotkeys. The **Autocomplete** form is shown as below.

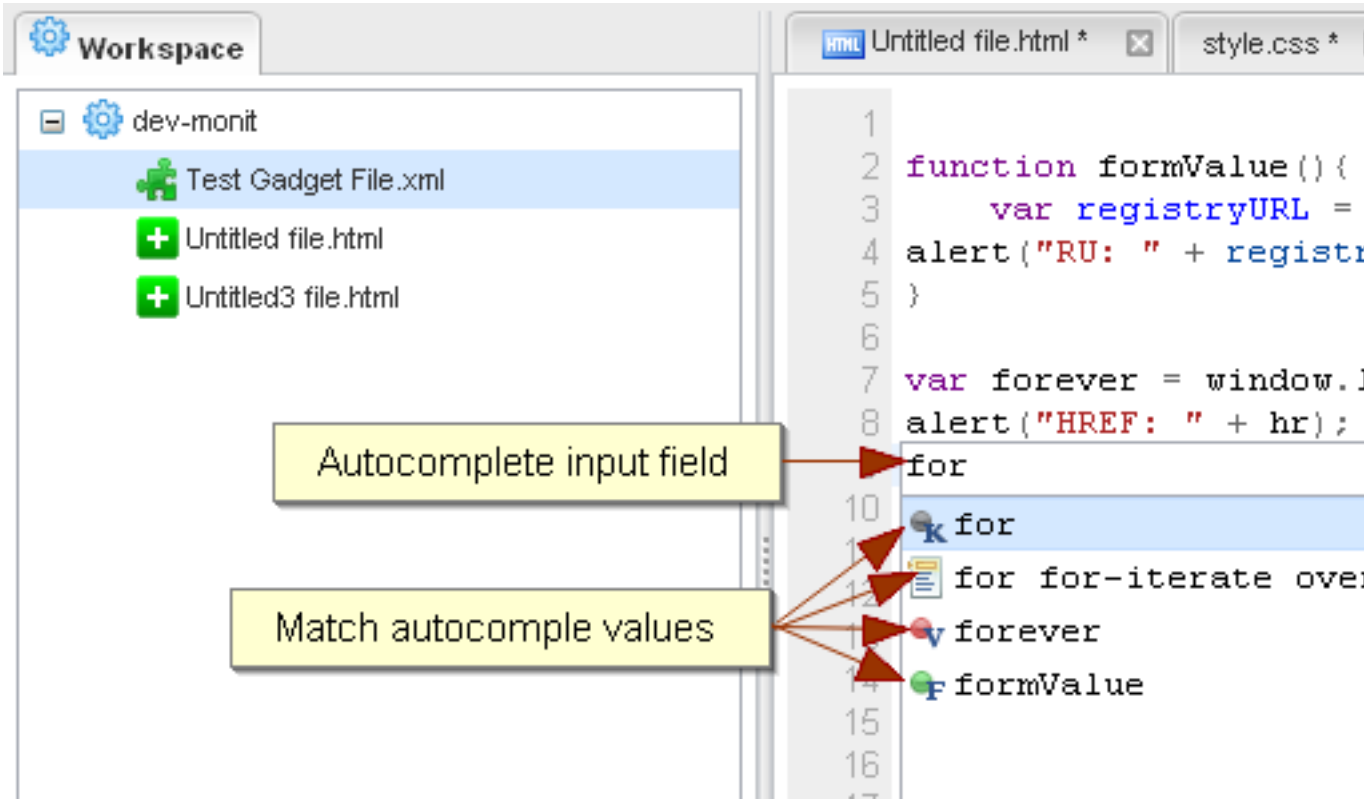


Illustration. Autocomplete form

In the files of Google Gadget, HTML and Template types, the autocomplete list displays various content for each case, depending on the cursor position within the file. If the cursor is put between <script> tags, or between <style> tags, there will be the autocomplete list for JavaScript content, or for CSS content respectively. In the other places, the autocomplete list for the HTML content is displayed.

To use the **Autocomplete** function, do as follows:

- **Step 1:** Open the target file with supported file types (JavaScript, CSS, HTML, Google Gadget, ECM Template and others) with the *Code Editor*.
- **Step 2:** Put the cursor on the needed position in the document.
- **Step 3:** Press **Ctrl+Space** to get the **Autocomplete** form.
- **Step 4:** Select the token in the list of this form; or type first letters of the token to reduce the token list and then select the appropriate token.
- **Step 5:** Press the **Enter** key or double-click the selected token to complete the word next to the cursor.

Note

- Use "Alt+Space" hotkeys instead of "Ctrl+Space" in the Mac OS.

- Click the outside area of the form to quit the "Autocomplete" form.

- Press the "Esc" key to discard the "Autocomplete" form.

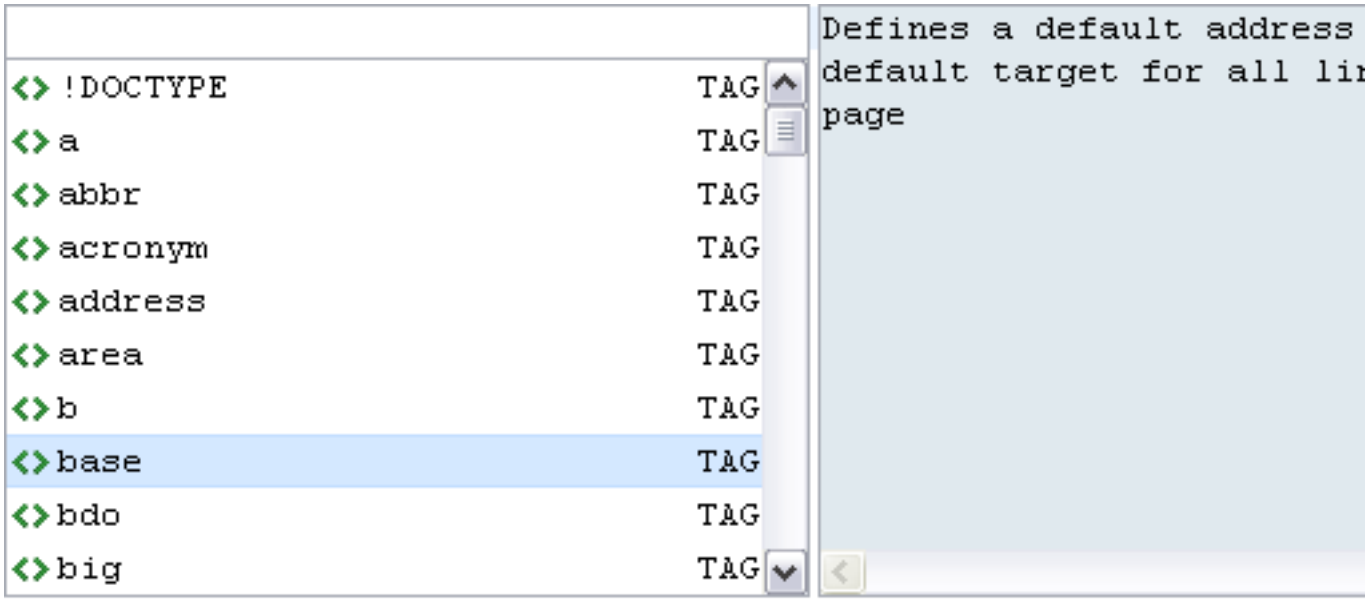


Illustration. Autocomplete form with description

JavaScript Specific Autocomplete List

There are some special template tokens in the autocomplete list to insert the code template. For example, the **if-condition** template is shown as below:

```
if (condition)
{

}
```

Template tokens have the special icon as shown in **Illustration. Autocomplete form** and tokens of different types are marked with different colors, including:

Table 4.2.

Token	Marked color
Function name	Green
Variable name	Red
Javascript key	Grey

CSS Specific Autocomplete List

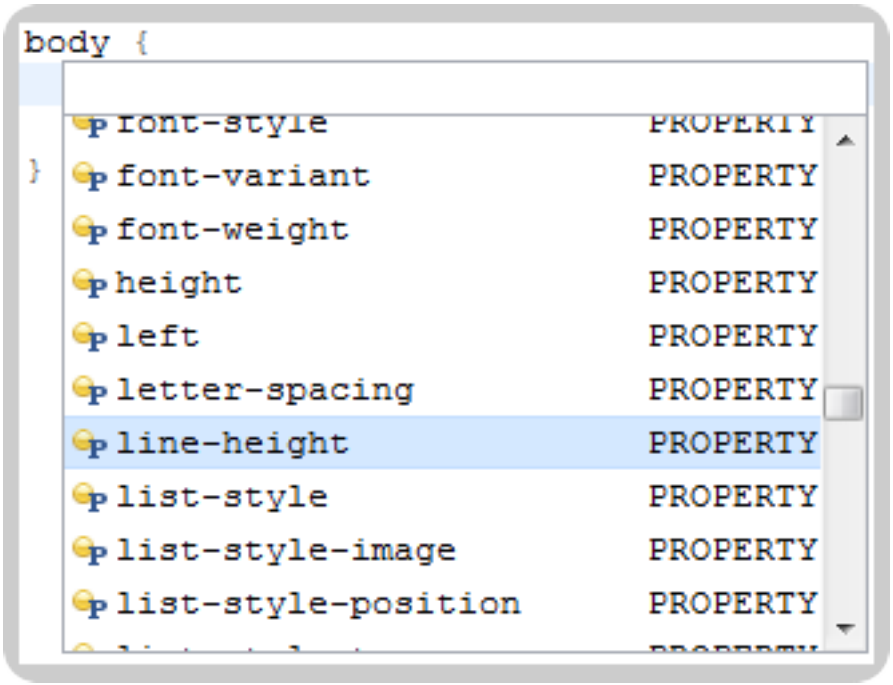


Illustration. CSS autocomplete

In the CSS file, the autocomplete list contains all the properties of Cascading Style Sheets. They are marked with the yellow token icons.

HTML Specific Autocomplete List

There are two HTML autocomplete lists: one of all HTML tags, and the other of all properties of the pointed tag.

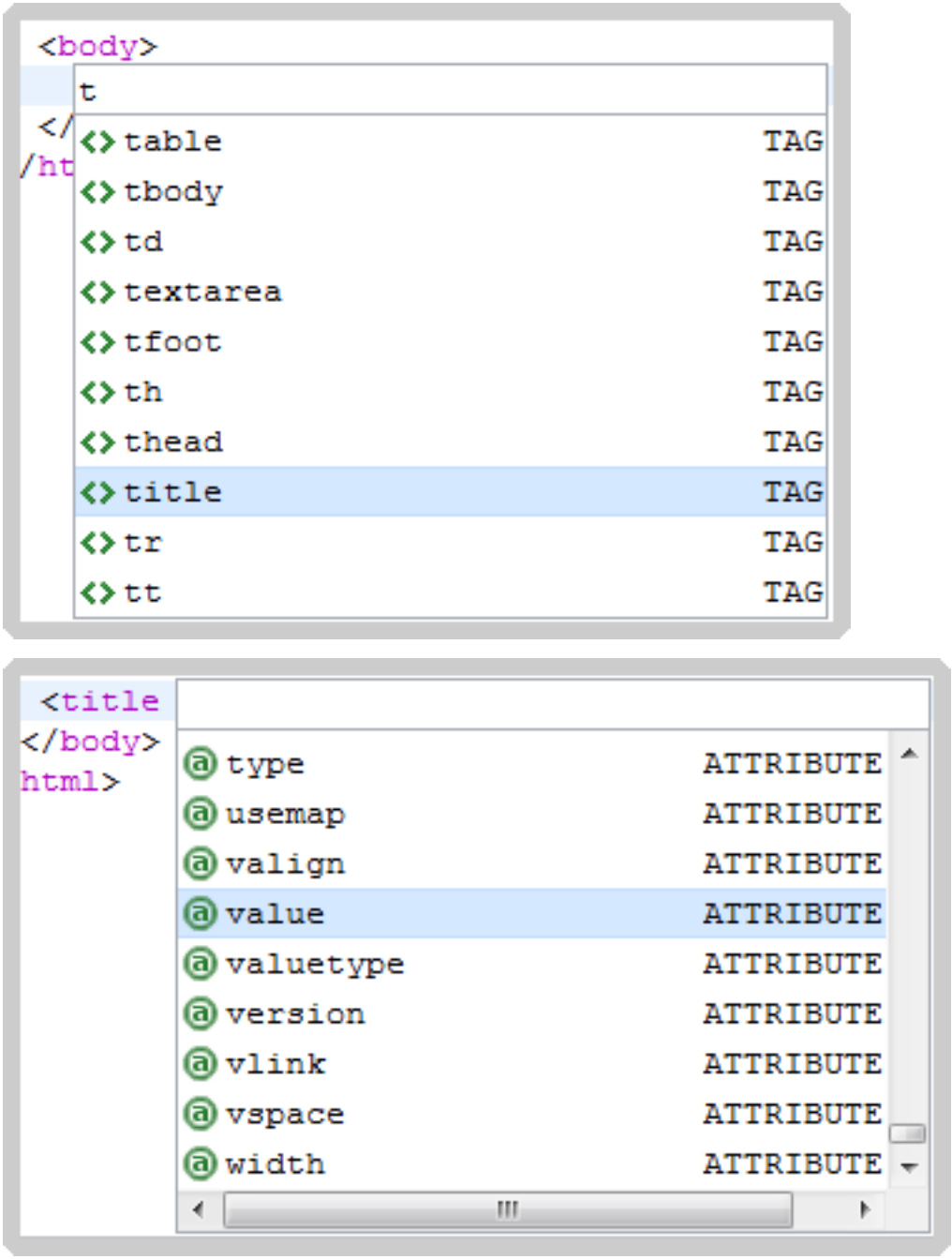


Illustration. HTML autocomplete form

Also, you can call the CSS properties and JavaScript autocomplete list if your current cursor position is inside <style> or <script> tags respectively.

XML Specific Autocomplete List

For the XML files, there is only one autocomplete list for the previously typed tag. There is no schema or namespace analysis.

ECM Template Autocomplete List

Currently, the autocomplete list of the ECM Template files is the same as that of the HTML files. Moreover, the content between Groovy tags <% %> is ignored.

Netvibes Widget Autocomplete List

The autocomplete list of the Netvibes Widget files is the same as that of the HTML files.

The autocomplete list inside javascript tags is the composition of the JavaScript autocomplete list and Netvibes snippets:

- JSON request snippet
- Flash object snippet
- Pager Control snippet
- Tabview Control snippet
- Thumbnailed List snippet

Exo IDE supports autocomolete for Netvibes Universal Widget.

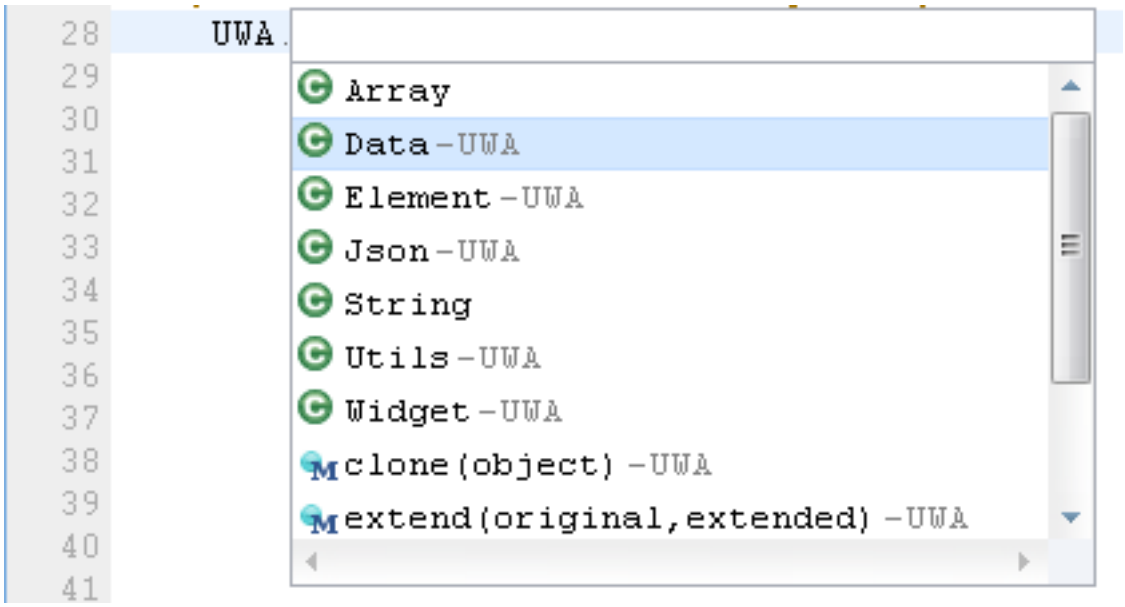


Illustration. Netvibes autocomplete form for UWA classes

It is possible to autocomplete the created Netvibes object's properties and methods.

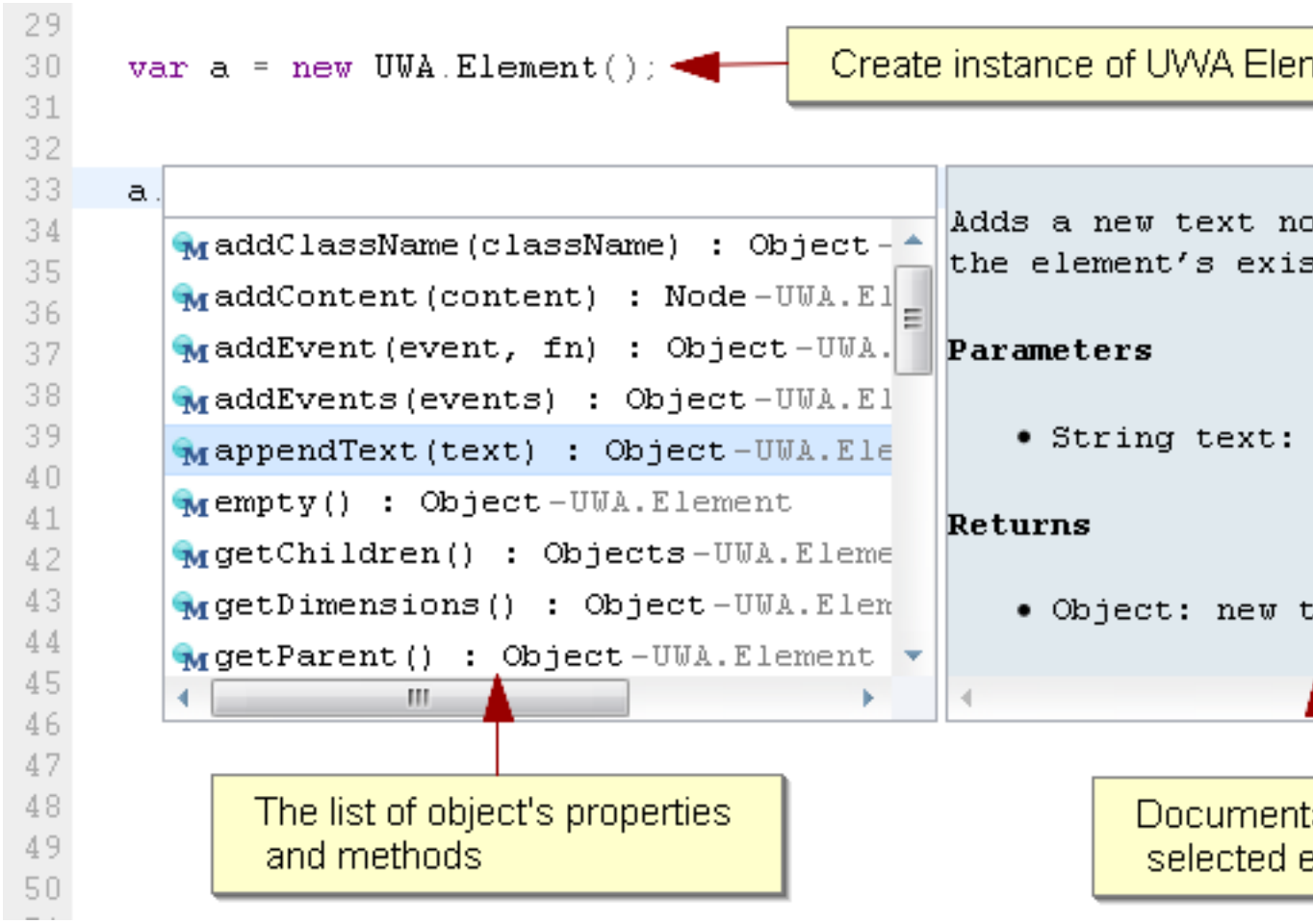


Illustration. Netvibes autocomplete form for object's properties and methods

More details about this snippets (and code, that will be inserted) you can see in hint window of autocomplete form

Groovy Code Specific Autocomplete List

eXo IDE currently supports special features for the Groovy code autocomlete list, including **REST Service** ("application/x-jaxrs+groovy"), **POJO** ("application/x-groovy") and **Data Object** ("application/x-chromattic+groovy"). The autocomplete list varies, depending on Java types and is shown in the following cases:

- <local variable name>.
- <object's field name>.
- <static field name>.
- <method's parameter name>.
- <class name>.

- @<annotation name>
- in the line after the space.

IDE compiles classes, which is are described in the project's classpath, before generation of *Autocomplete list*. If the compilation of some class is failed, the error message will be appear in *Output Panel* instead of **Autocomplete Form** as shown below:

```
[ERROR] Error (500: Internal Server Error)
startup failed, /ide-project/data/Pojo.groovy: 9: unexpected token: vpid @ line 9, column 3.
vpid fg(){
^
```

To display classes from folder, described as *folder source* in the project's classpath, in the *Autocomplete Form*, the files with such classes should fulfill conditions as follows:

1. Must be located in folder with respect to package structure.
2. The name must be the same as the class name located in a file.
3. Must have extension ".groovy" or ".grs".

Class Name

You can view class's JavaDoc and class name from the autocomplete form. eXo IDE provides appropriate import statement into the file header automatically, if needed. There is a list of default packages for the Groovy code and so are imported explicitly:

- **java.lang.**
- **java.io.**
- **java.lang.**
- **java.math.BigDecimal**
- **java.math.BigInteger**
- **java.net.**
- **java.util.**
- **groovy.lang.**

- **groovy.util.**

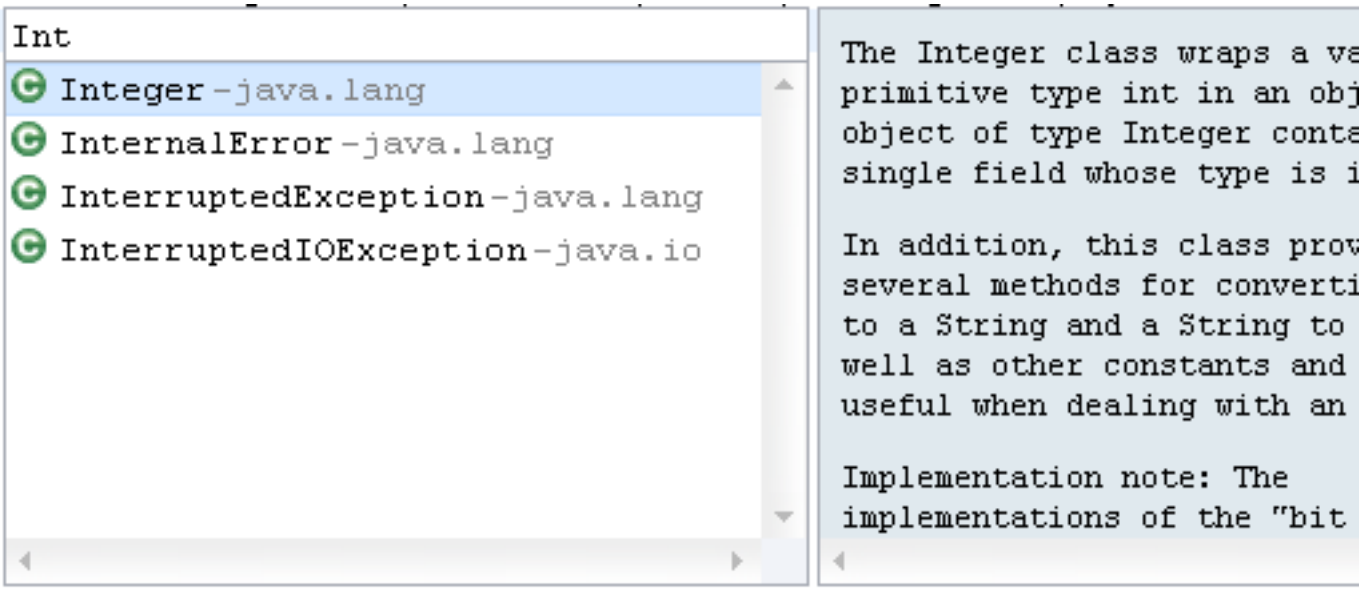


Illustration. Groovy autocomplete form for classes

Object's Fields and Methods Autocomplation

You can also view JavaDoc and autocomplete the object's fields and methods as follows:

- **Step 1:** Print the instance name with the dot (.) at the end.
- **Step 2:** Press **Ctrl+Space** to view all public methods and fields of the object. The JavaDoc for the selected field or method is shown.
- **Step 3:** Select the appropriate item in the autocomplete form and click the **Enter** key; or double-click this item.

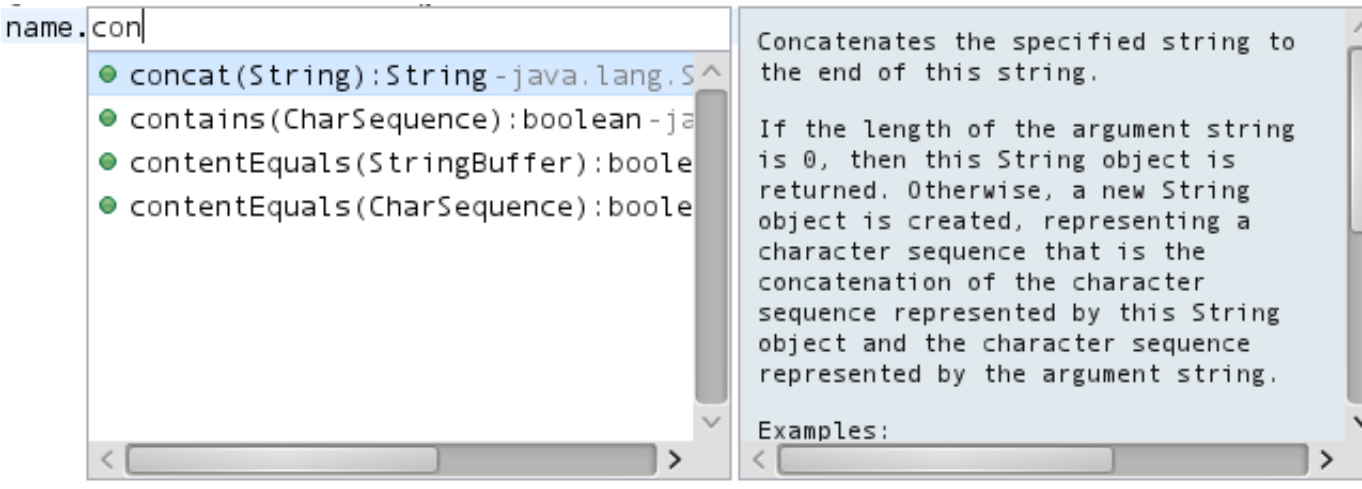


Illustration. Groovy autocomplete form for public methods

Static Fields and Methods Autocompletion

eXo IDE supports the autocompletion of static fields, methods and object's fields and methods.

Note

The autocompletion for methods and fields of the class or instance can only work, if it is called after the symbol ".".

Annotations Autocompletion

It is possible to autocomplete the annotations, followed by symbol "@". The Autocomplete form for annotation contains the list of available annotations like this:

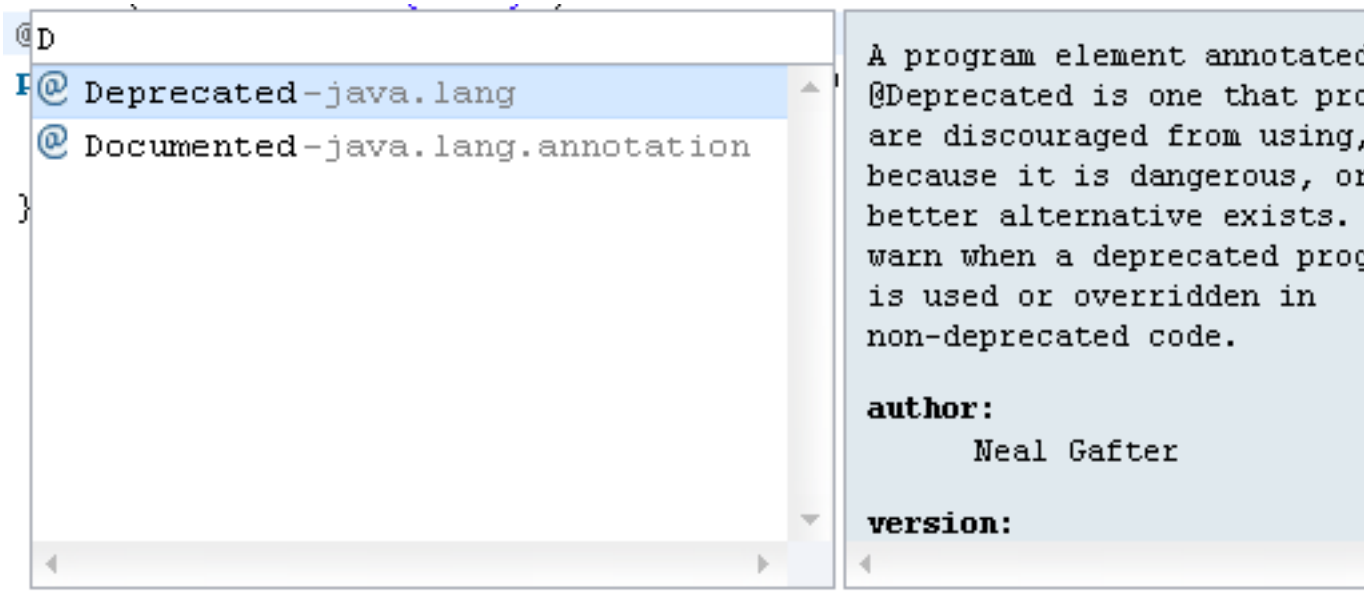


Illustration. Groovy autocomplete form for annotations

Local Variables and Parameters Autocomplation

Local variables and method's parameters are marked with the special icon in the autocomplete form -  .

eXo IDE supports the autocompletion for local variables and method's parameters.

Besides, the keywords of Groovy code are not marked with icons in the autocomplete list.

Validate Java Types and Fix Import Statements

If you forget to insert the import statement, the line with error will be marked as follows:

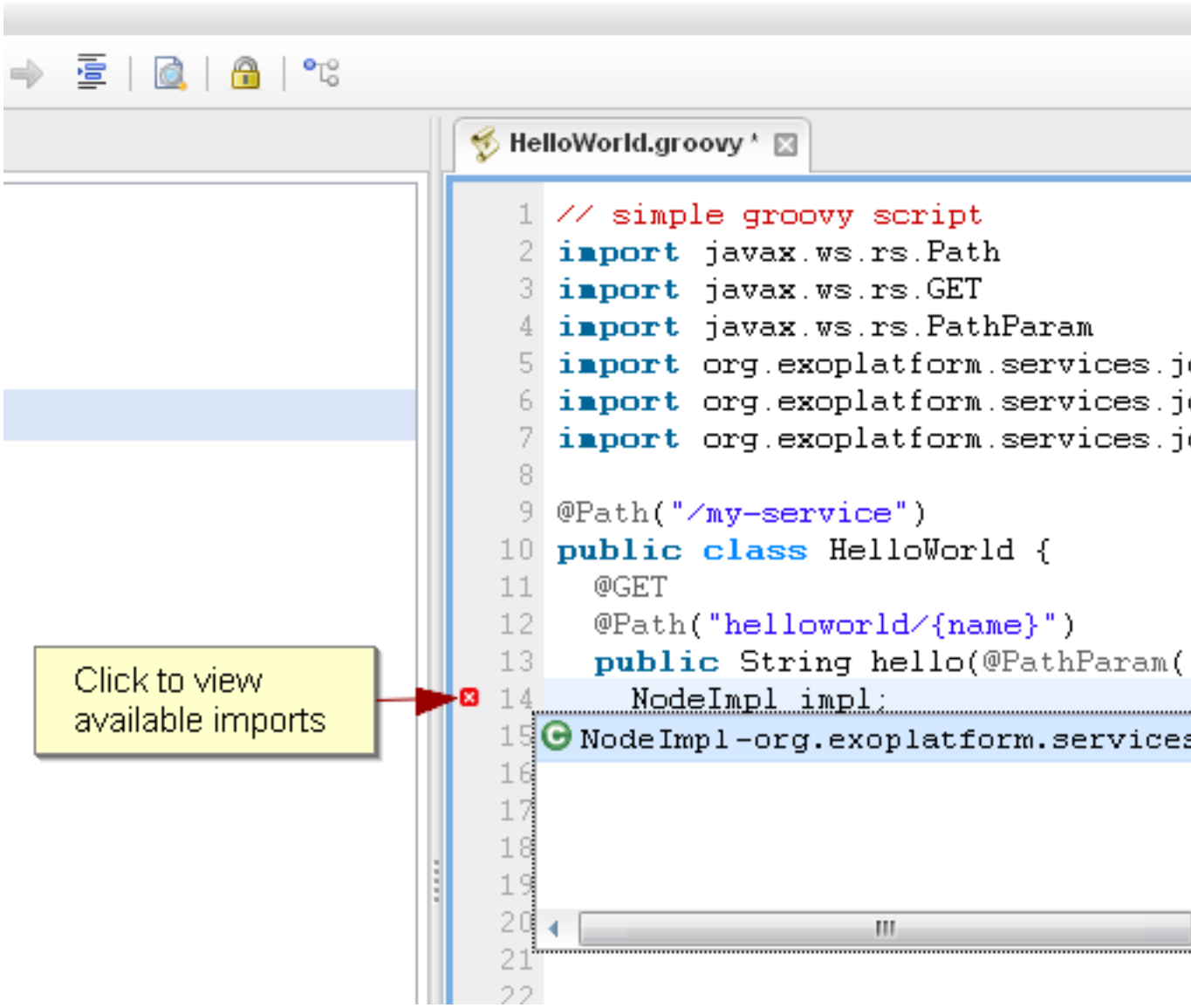


Illustration. Groovy Code Error Mark

To view and fix the Java type error, do as follows:

- **Step 1:** Hover the cursor on error marks to see the error message in tool tips.
- **Step 2:** Click the error icon in the *Line numbers* field.
- **Step 3:** Select the appropriate import statement and press the **Enter** key; or double-click to insert it into the header of the file.

Code Outline

The code outline is used to create codes productively. The outline is available for all types of files, expect for *Text* or *CSS* files.

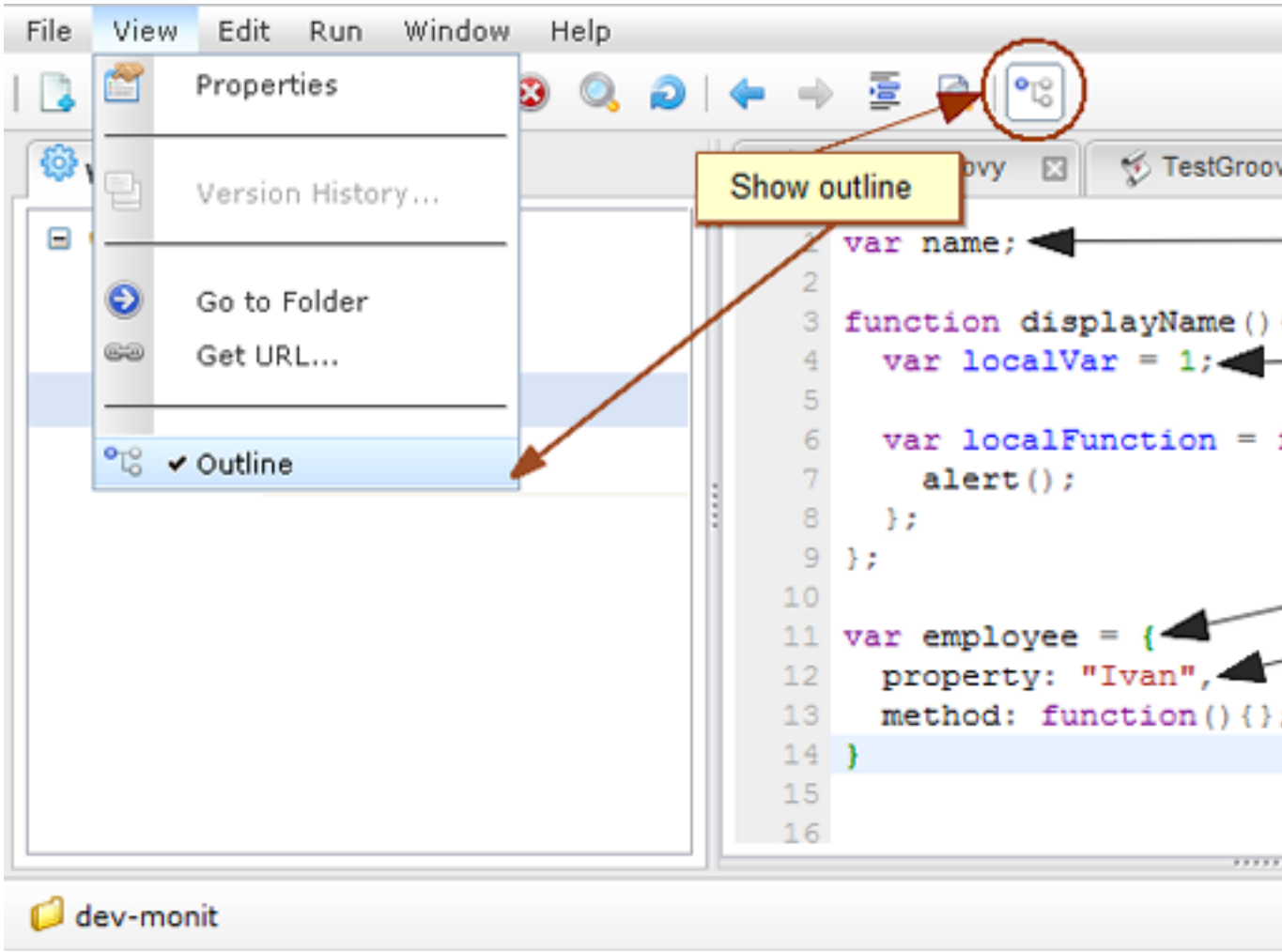


Illustration. JavaScript Code Outline

The *Outline Panel* is opened automatically once a file has been created or opened. You can close the *Outline Panel* by clicking the **Close** button on the tab header. Also, you can show/hide the *Outline Panel* by pressing the **Show/Hide Outline** button on the toolbar or select **View** on the top menu and then select/deselect **Outline** from the drop-down menu.

In the *Outline Panel*, you can see the tree of variables, functions, classes, objects, methods, fields, parameters, annotations, HTML and XML tags with icon "< >", and special code delimiters like **"<% ... %>"** in the *ECM Template* file, or **"CDATA"** in the *Google Gadget* file as the illustrations below.


Example ECM template.gtmpl *
<html>
<head>
<%
import org.exoplatform.services.security.Ident
import org.exoplatform.services.security.Conve
%>
<meta http-equiv='Content-Type' content='text/h
<link rel='selenium.base' href='' />
<title>helpAbout</title>
<script scr='test.js' />
<style>
span {
font-size: 12pt;
}
</style>
<script type='text/javascript'>
var prefs = new gadgets.Prefs();
function displayGreeting () {
// Get current time
var today = new Date();
}
</script>
</head>
<body>
<table>
<tr>
<td>1</td>
<td>2</td>
</tr>
</table>
<%
ConversationState curentState = ConversationSt
if (curentState != null)
{
Identity identity = curentState.getIdentity(
}
%>
</body>
</html>

Illustration. ECM Template Code Outline

Google Gadget example.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<Module>
  <ModulePrefs title="State Example" height="120">
    <Require feature="wave" />
  </ModulePrefs>
  <Content type="html">
    <![CDATA[
    <div id="content_div" style="height: 50px;"></div>
    <script type="text/javascript">

      var div = document.getElementById('content_div');

      function buttonClicked() {
        var value = parseInt(wave.getState().get('count',
        wave.getState().submitDelta({1: value + 1});
      }

      function stateUpdated() {
        if(!wave.getState().get('count')) {
          div.innerHTML = "The count is 0."
            }
        else {
          div.innerHTML = "The count is " + wave.getState
        }
      }

      function init() {
        if (wave && wave.isInWaveContainer()) {
          wave.setStateCallback(stateUpdated);
        }
      }
      gadgets.util.registerOnLoadHandler(init);

      // Reset value of "count" to 0
      function resetCounter(){
        wave.getState().submitDelta({1: '0'});
      }
    </script>
    <input type=button value="Click Me!" id="butCount" on
    <input type=button value="Reset" id="butReset" onClic
    ]]>
  </Content>
</Module>
```

Illustration. Google Gadget Code Outline

The outline tree is refreshed automatically every two seconds after you stop typing the code. You can click the item in the outline tree. The cursor then sets on the line where this item is defined. When you move the cursor into the file content, the appropriate node is selected in the outline tree.

Code Outline for Groovy Code

In the *Outline Panel* for the Groovy code of REST Service, ECM Template, POGO and Data Object files, you can see not only names and types of your classes, variables, methods, fields, or parameters, but also their modifiers and annotations in the view of icons and tooltips.

The following is the list of icons connected to different access modifiers.

Table 4.3.

Access modification	Icon
private	Red square
public	Green circle
protected	Yellow rhombus
without access modifier	Blue triangle

The following is the list of letters placed over the node icons of outline node and one icon connected to special modifiers.

Table 4.4.

Letter/Icon	Modifier
a	Abstract
s	Static
f	Final
t	Transient
v	Volatile
Clock	Synchronized

Class or method with annotation(s) is displayed with the @ symbol near the title in the outline tree. To see the full list of annotations, hover the cursor on @.

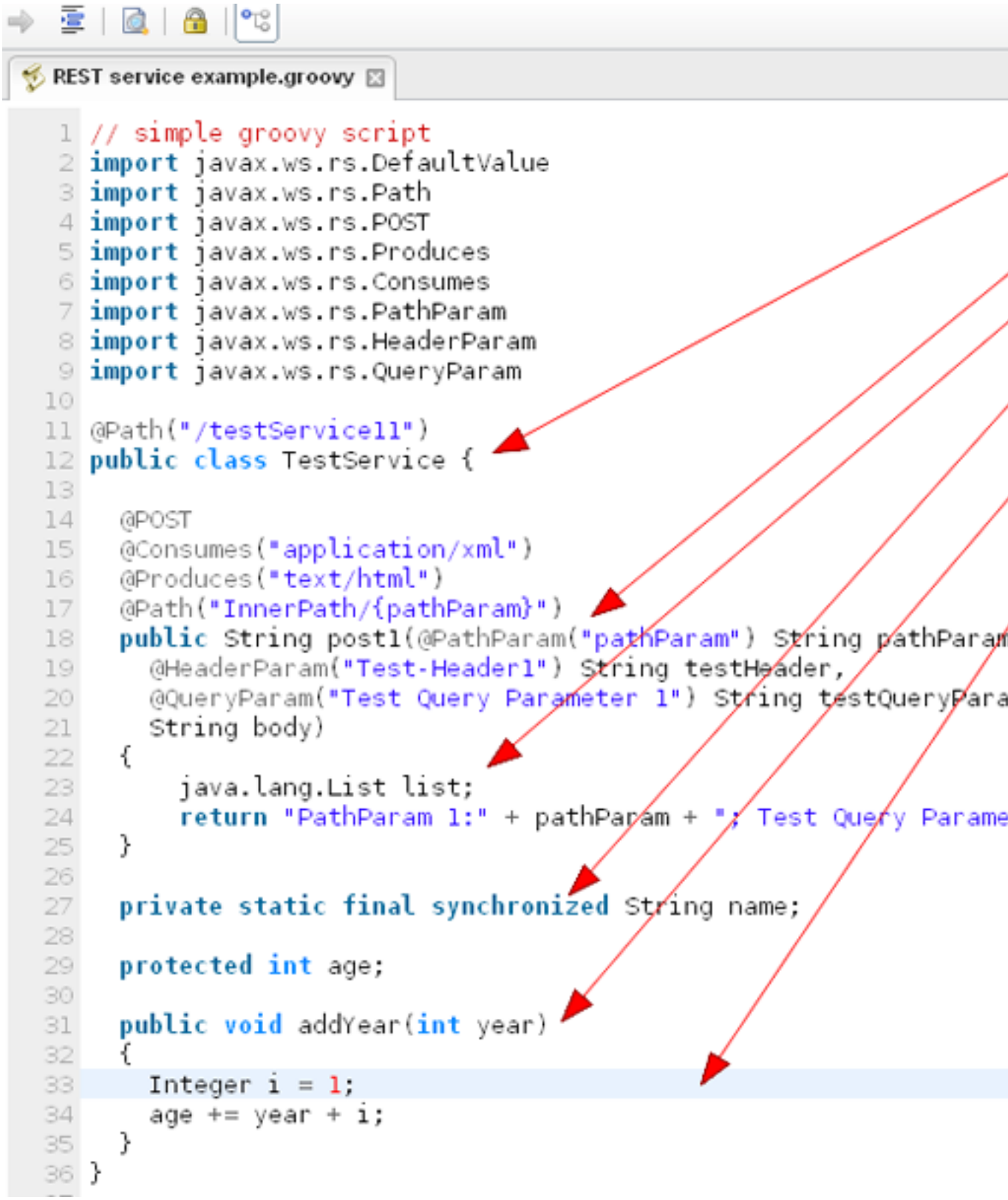


Illustration. REST Service Code Outline

Edit File in the WYSIWYG Editor

Besides the *Code Editor*, you can open and edit HTML files and *Google Gadgets* with the *WYSIWYG Editor* based on the CKEditor. You can set the *WYSIWYG Editor* for those files by default.

Currently *WYSIWYG Editor* can be used to edit the HTML, Google Gadget or Groovy Template files.

The *WYSIWYG Editor* does not support the line numbering, auto-indentation and format, code coloring, autocomplete, code outline or type error validation. However, the text edited with the *WYSIWYG Editor* looks as similar as published results. This editor has many common editing features as working in Microsoft Word, or OpenOffice.

To open a file with the *WYSIWYG Editor*, see **Open File with the Non-default Editor** above.

Preview File Properties

Connected with the JCR repository node, file properties are displayed in the *Properties Tab* in the bottom of *Content Panel*. To view file properties, do as follows:

- **Step 1:** Open the file in the *Content Panel*.
- **Step 2:** Click the **Show Properties** button at the right corner of the toolbar; or go to **View > Properties** on the top menu.

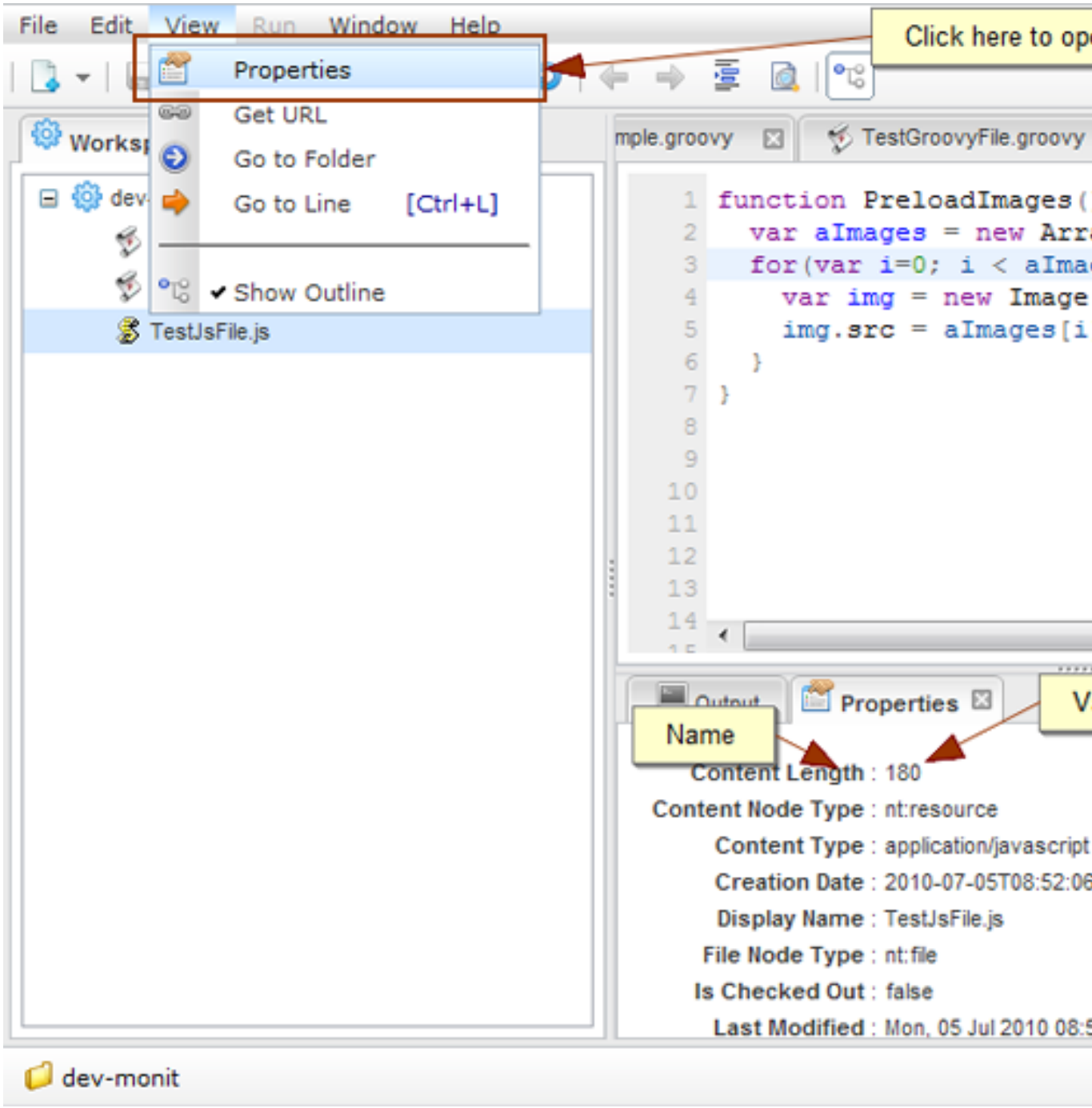


Illustration. View file's properties

Note

It is impossible to view the properties of files created but not saved yet.

Preview HTML File

To preview the HTML file stored in the JCR, simply click the **Show Preview** button on the toolbar, or go to **Run > Show Preview** on the top menu after opening the HTML file in the *Content Panel*. The content of the HTML file is opened as a webpage in the *Preview Tab*.

Note

- You cannot view files created but not saved yet.
- Unsaved changes are not displayed when you edit the HTML file.

Search Files

eXo IDE enables you to search by file content and MIME type. To find, do as follows:

- **Step 1:** Select the target folder or root folder in the *Workspace Panel*.
- **Step 2:** Click the **Search...** icon on the toolbar; or go to **File > Search...** on the top menu to open the **Search** dialog.
- **Step 3:** Set parameters for two criteria fields, including **Containing text** and **MIME type**. If those fields are empty, the gadget will display all files in the search results.
- **Step 4:** Click **Search**.

The found results are displayed in the *Search Panel* on the left part of the application:

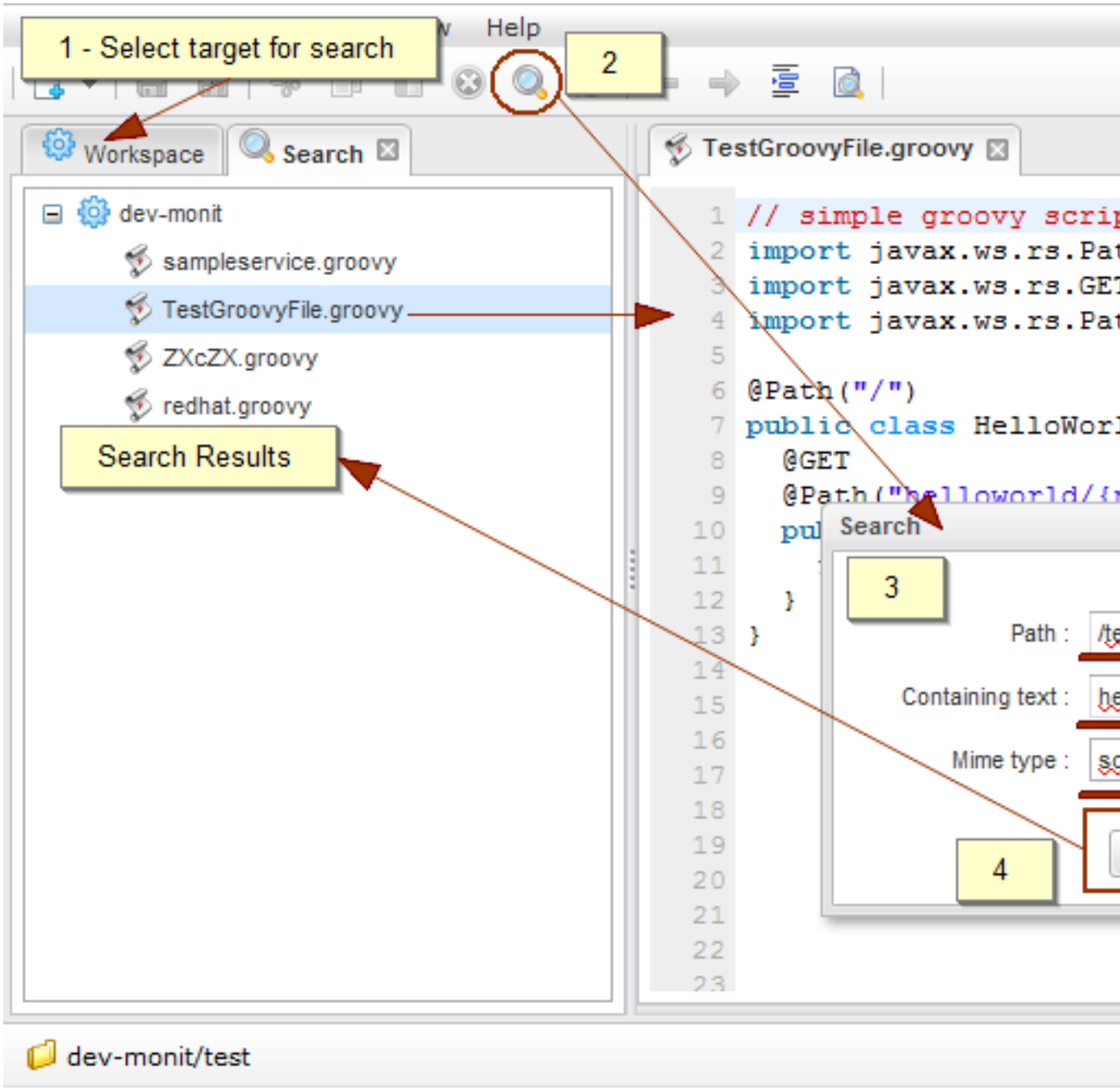


Illustration. Search files

Within the *Search Panel*, you can only open found files, edit, save, save as template, or perform some specific operations, such as deploying for services, and going to the parent folder in the *Workspace Panel*.

If you maximize the *Editor Panel*, *Properties, Output*, *Outline* or other panels, the default perspective will be restored to show the *Search Tab* when you receive search results.

REST Service Operations

The eXo IDE application provides the original approach to operate with *REST Service* to **validate**, **deploy/undeploy** and **launch** applications, which are compatible with the JSR-311 specification without restarting the server, or installing additional programs:

Table. Accessibility of REST Service Operations

Table 7.1.

Operation	User with "developer" role	User with "administrator" role
Validate	Enabled	Enabled
Deploy	Disabled	Enabled
Undeploy	Disabled	Enabled
Deploy in Sandbox	Enabled	Enabled
Undeploy from Sandbox	Enabled	Enabled
Set/Unset Autoload	Disabled	Enabled
Run in Sandbox	Enabled	Enabled
Launch REST Service	Disabled	Enabled

With eXo IDE, you can validate REST Service before saving, deploying, launching or undeploying the service. The feedback from server is displayed in the *Output Tab*:

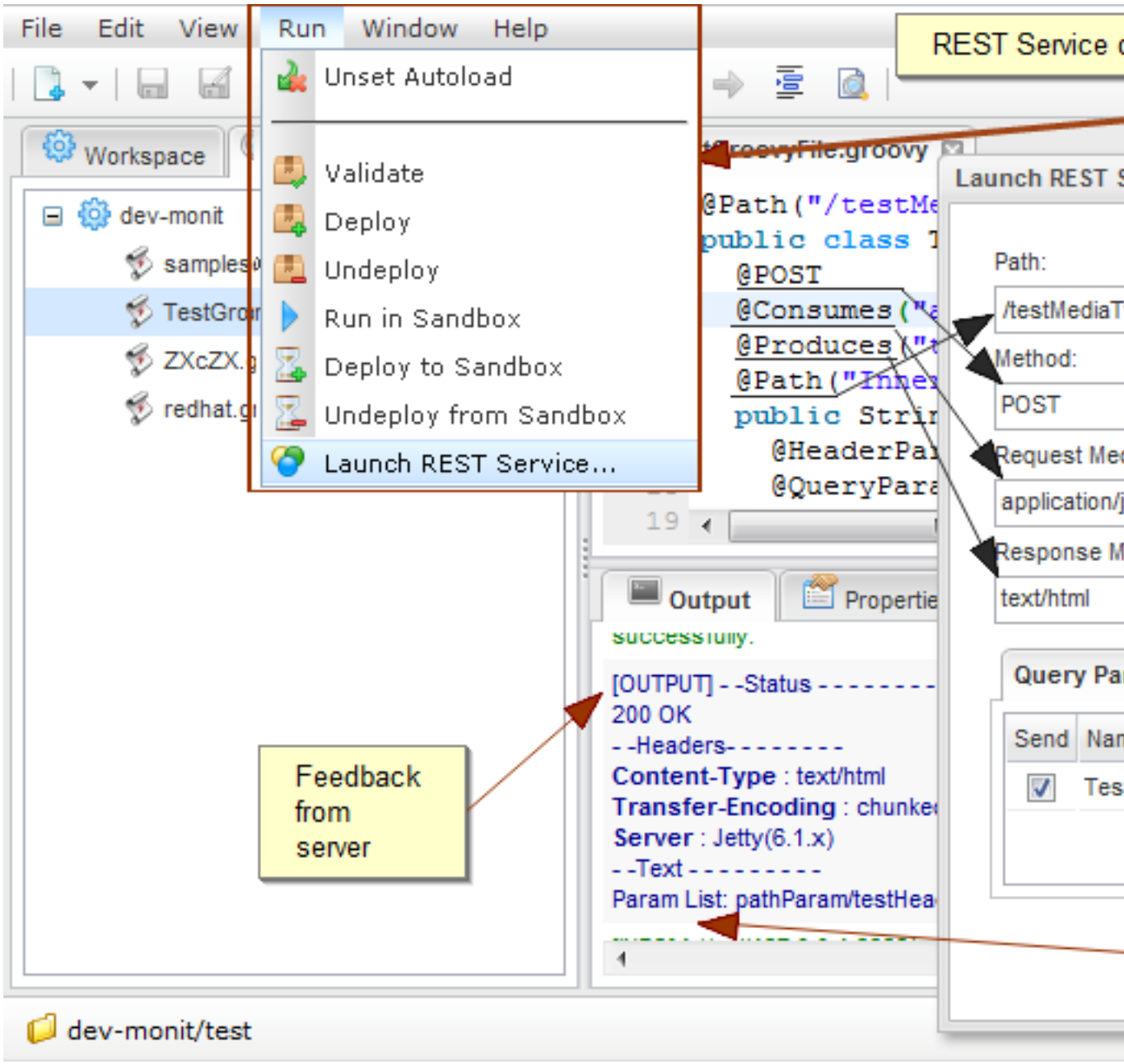


Illustration. REST Service operations

All *REST Service* commands are placed at the right part of the toolbar and in the **Run** on the top menu.

To clear the *Output Panel*, use the special button at the right top corner of this panel.

Validate REST Service

You can verify the REST Service content by using the special validation service before saving. To do that, simply click the **Validate REST Service** button, or go to **Run > Validate** on the top menu. In case of no errors in the service, there will be a message in the *Output Tab*:

```
[INFO]rennes.groovy validated successfully".
```

Otherwise, there will be an error message displayed in the *Output Tab*:

```
[ERROR] rennes.groovy validation failed. Error (400: Bad Request)
Unexpected error. Error occurs when parse stream, compiler error:
startup failed, rennes.groovy: 4: unable to resolve class javax.ws.rs.Path1
@ line 4, column 1.rennes.groovy: 8: unable to resolve classpath , unable to
 find class for annotation
@ line 8, column 1.rennes.groovy: 11: unable to resolve classpath , unable
 to find class for annotation
@ line 11, column 3.
3 errors
```

You can click the error message in the *Output Panel* to go to the line with error in the REST Service file. In case the file has been closed, eXo IDE will automatically open it. Make sure that you do not delete or rename this file after the error message has been displayed in the *Outline Panel*.

Deploy/Undeploy REST Service

This function is for users with the administrator role only.

To deploy a REST Service, do as follows:

- **Step 1:** Save the file before deploying.
- **Step 2:** Open the saved file in the *Content Panel*.
- **Step 3:** Click the **Deploy REST Service** button on the toolbar; or go to **Run > Deploy**.

The deployment request is sent to the server. In case of no errors, the message is displayed in the *Output Tab* as below:

```
[INFO]rennes.groovy deployed successfully
```

Otherwise, there will be an error message, for example:

```
[ERROR] rennes.groovy deploy failed. Error (400: Bad Request)
Unexpected error. Error occurs when parse stream, compiler error:
startup failed, rennes.groovy: 4: unable to resolve class javax.ws.rs.Path1
@ line 4, column 1.rennes.groovy: 8: unable to resolve class Path , unable
 to
find class for annotation
@ line 8, column 1.rennes.groovy: 11: unable to resolve class Path , unable
 to
find class for annotation
@ line 11, column 3.
```

```
3 errors
```

- It is impossible to deploy the service with the registered URI pattern, if the path (including name) of the new service is different from the original one. To cope with it, first undeploy the service with the registered URI pattern.
- It is necessary to validate the Groovy scripts used by the service before deploying it by selecting "Run > Validate" on the top menu.

To undeploy a Rest Service, do as follows:

- **Step 1:** Select the service deployed in the *Workspace*.
- **Step 2:** Double-click the selected service to open it in the *Content Panel*.
- **Step 3:** Click the **Undeploy REST Service** button on the toolbar.

In case of no errors, there is a success message in the *Output Tab*.

```
[INFO] /repository/collaboration/rennes.groovy undeployed successfully.
```

You can deploy multiple REST Services, and double-deploy the services. However, you cannot undeploy services which were not previously deployed. In this case, you will receive an error message from the server as below:

```
[ERROR] /repository/collaboration/rennes.groovy
undeploy failed. Error (400: Bad Request)
Can not unbind script rennes.groovy, not bound or has wrong mapping to the
resource class
```

Deleting the service will not make it undeploy. Please undeploy the service before deleting it. In other cases, the workaround is to create it again in the same place and then undeploy.

Use Autoload Property

This function is for users with the administrator role only.

REST Service is deployed automatically on the server after being saved, if the **Autoload** property is set to *True*. You can view this property in the *Property Tab* by clicking the **Show Properties** button. To manage the **Autoload** property, use **Set/Unset Autoload** commands respectively. These commands are displayed to the current status of the **Autoload** property, so you can invert

this property. The default value of the **Autoload** property is set to *False*. For more details, see the illustration below:

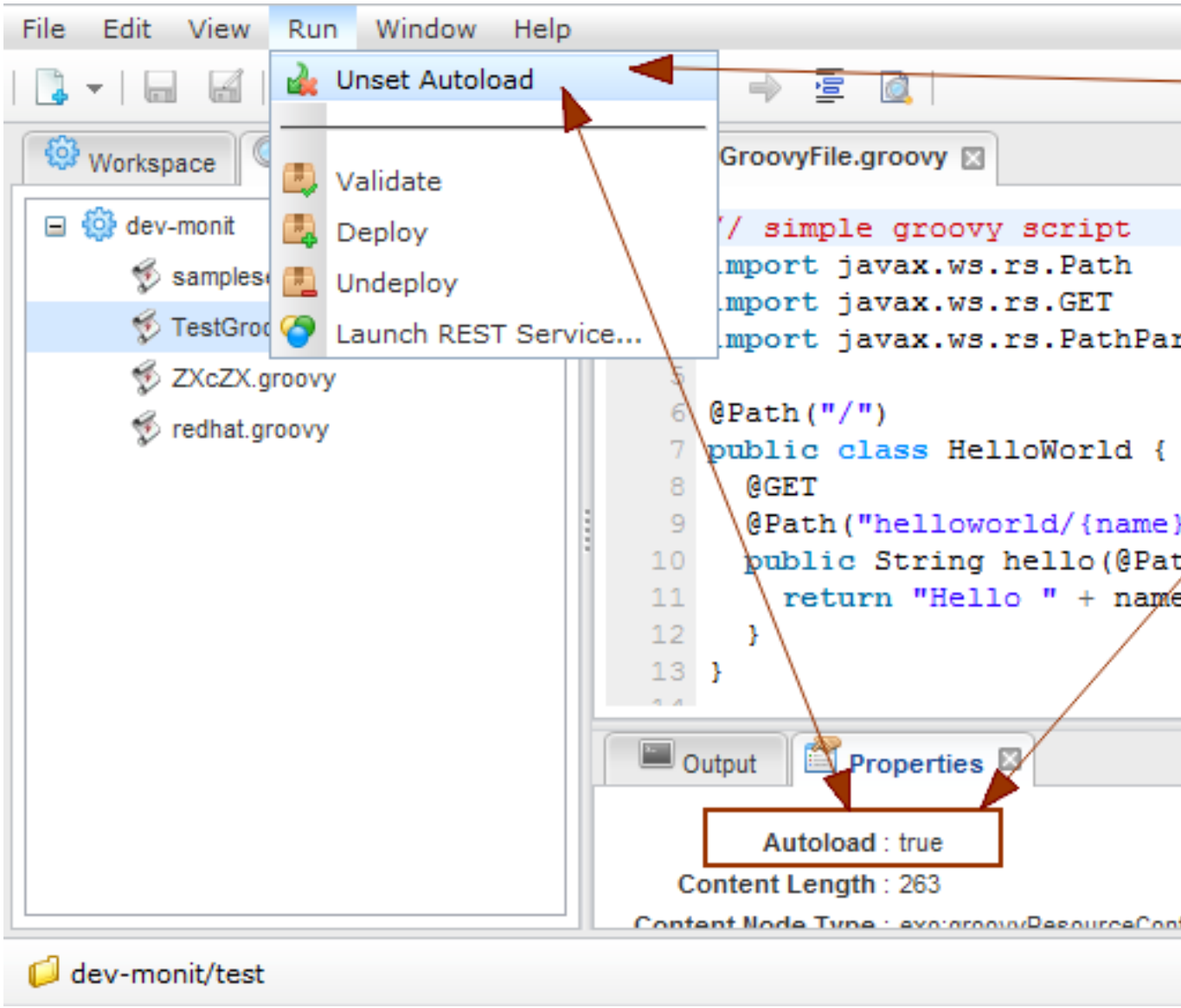


Illustration. Change Autoload service property

Launch REST Service with set parameters and Preview Server Console Messages in the *Output Tab*

With eXo IDE, you can not only write and deploy services, but launch these services and view service response in the *Output Tab*. You can view *WADL-description* of methods, make requests with your own *header*, *query*, *path parameters* and *body content* using one of the *supported methods*. To do that, simply call the **Launch REST Service...** button on the toolbar, or select **Run > Launch REST Service...** on the top menu as follows:

- **Step 1:** Select the deployed REST Service.

- **Step 2:** Click the **Launch REST Service...** button to get the REST Service form which is relied on the WADL-description of *REST Service*.
- **Step 3:** Select and type the path to the service in the **Path** field. Path parameters are enclosed in curly braces.
- **Step 4:** Select one of the supported methods from the **Method** combo-box.
- **Step 5:** Select the appropriate **Request Media Type**.
- **Step 6:** Check the **Response Media Type**.
- **Step 7:** Uncheck the redundant query parameters and set values of the rest in the table at the bottom of dialog window.
- **Step 8:** Go to the **Header Parameters** middle tab and set appropriate values.
- **Step 9:** Set request with the body content within the **Body** tab. This tab is disabled for the *GET* request.
- **Step 10:** Press the **Send** button.

You will receive a success message if the path is verified to be correct or an error message in the *Output Tab*.

To learn about launching REST Service, see the **REST Service operations** illustration above.

Table. Reproducing service annotations in the Launch REST Service **dialog**

Table 7.2.

REST Service Annotation	Element of Launch REST Service dialog
@Path	The **Path** field
@GET, @POST,...	The **Method** field.
@Consumes	The **Request Media Type** field.
@Produces	The **Response Media Type** field.
@PathParam	The path parameters figured in curly braces, for example, /service/{**param**}/{**paramList: .**}.
@HeaderParam	The **Header Parameter** tab of the bottom table.
@QueryParam	The **Query Parameter** tab of the bottom table.
@DefaultValue	The **By default** column of the bottom table.

Get URL of REST Service

From the **Send request** window, you can also get the URL of REST Service as follows:

- **Step 1:** Select the deployed REST Service.

- **Step 2:** Click the **Launch REST Service...** button to get the REST Service form.
- **Step 3:** Click the **Get URL** button to view the REST Service URL.

Review of the response headers from REST Service

You can view headers, status code and status text in the *Output Panel* after sending your requests as below:

```
[OUTPUT] - -Status - - - - - - - -
200 OK
- -Headers- - - - - - - -
Content-Type : */*
Transfer-Encoding : chunked
Server : Jetty(6.1.x)
- -Text - - - - - - - - -
Hello steve
```

Deploy to Sandbox, Undeploy from Sanbox, Run in Sandbox

This function is for users with both developer and administrator role.

In eXo IDE, developers can secure deploy untested REST Service for testing in an isolated environment named *Sandbox*. To do this, click the **Deploy REST Service to Sandbox** button on the toolbar or go to **Run > Deploy to Sandbox** on the top menu.

To undeploy from sandbox, select the **Undeploy REST Service from Sandbox** button on the toolbar, or go to **Run > Undeploy from Sandbox** on the top menu.

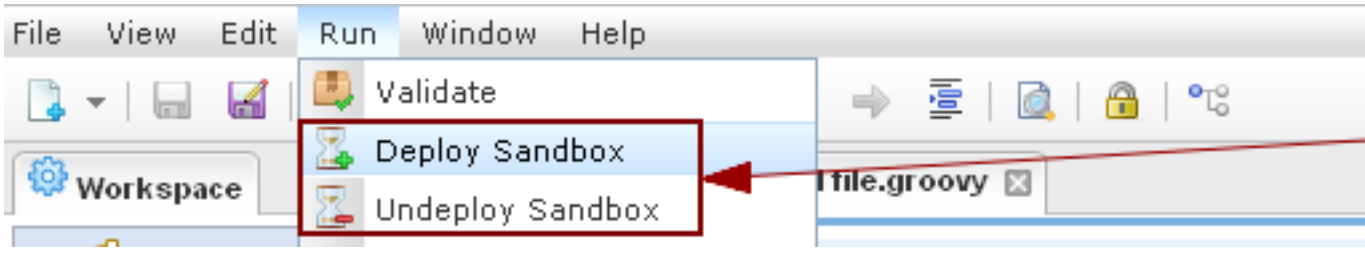


Illustration. Using REST Services Sandbox

Note

If REST Service is previously deployed by other user, you will not be able to deploy it.

To run REST Service faster, you can use the special command **Run > Run in Sandbox** which is a sequence of next operations: **Save**, **Deploy to Sandbox**, **Undeploy from Sandbox**, **Launch**

REST Service. Simply click the **Run in Sandbox** button at the right corner of the toolbar, or go to **Run > Run in Sandbox** on the top menu.

REST Service Discovery

In the eXo IDE, you can explore all REST Services which are available from the server. To open *REST Service Discovery* form, select **Help > REST Service Discovery...** from the top menu.

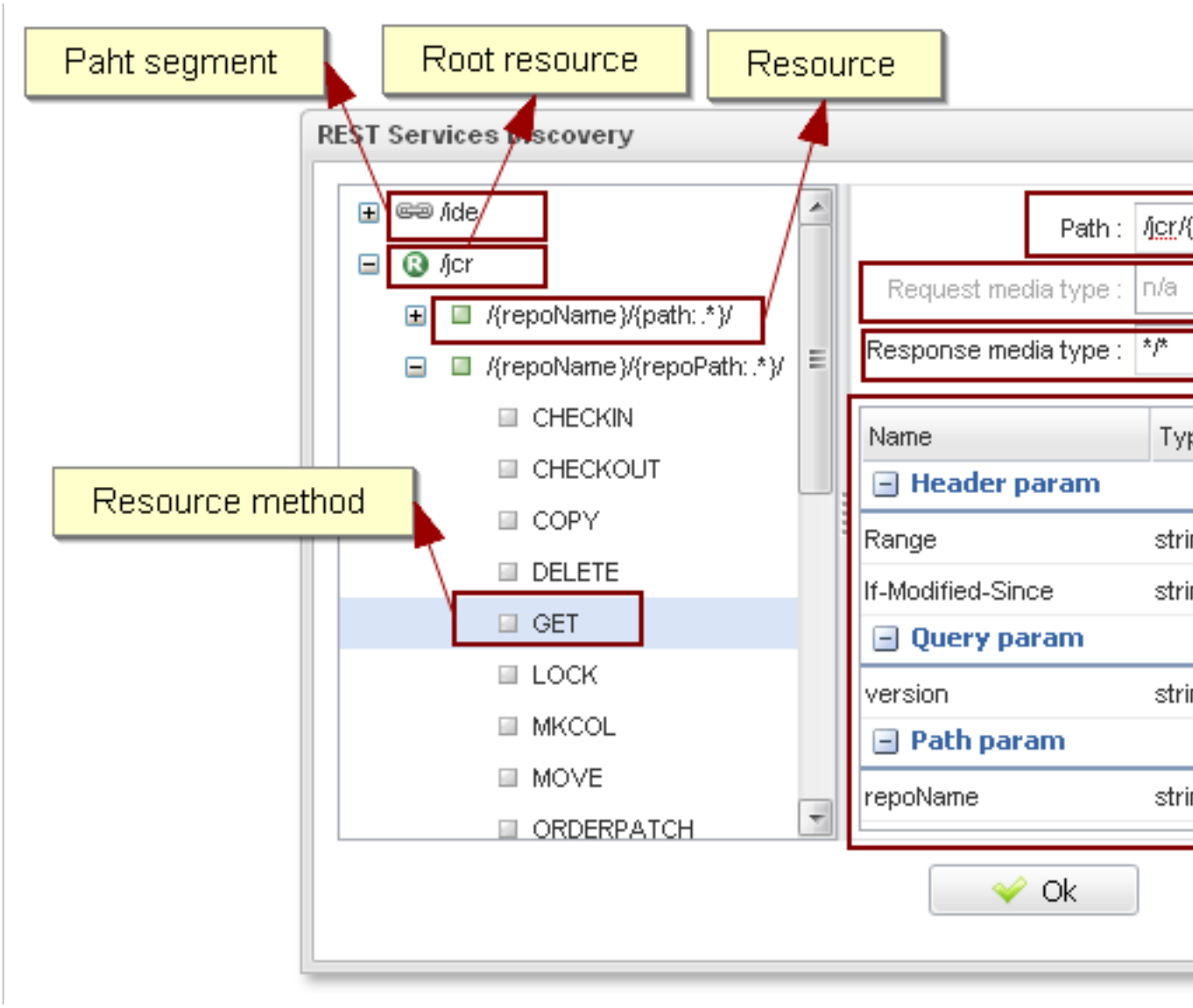


Illustration. REST Service Discovery form

Configure Classpath

The class path configuration means the setting of paths to the source files or folders. This class path will be used for project sources compilation. Classpath file is located in the root of the project's directory, but it's not visible in browser tree, because can not be directly edited by user. So, to view the sources paths of the project, do as follows:

- **Step 1.** Select the created project item in *Workspace Panel*.
- **Step 2.** Go to **File > Configure Classpath...** on the top menu to open the **Configure Classpath** window as below:

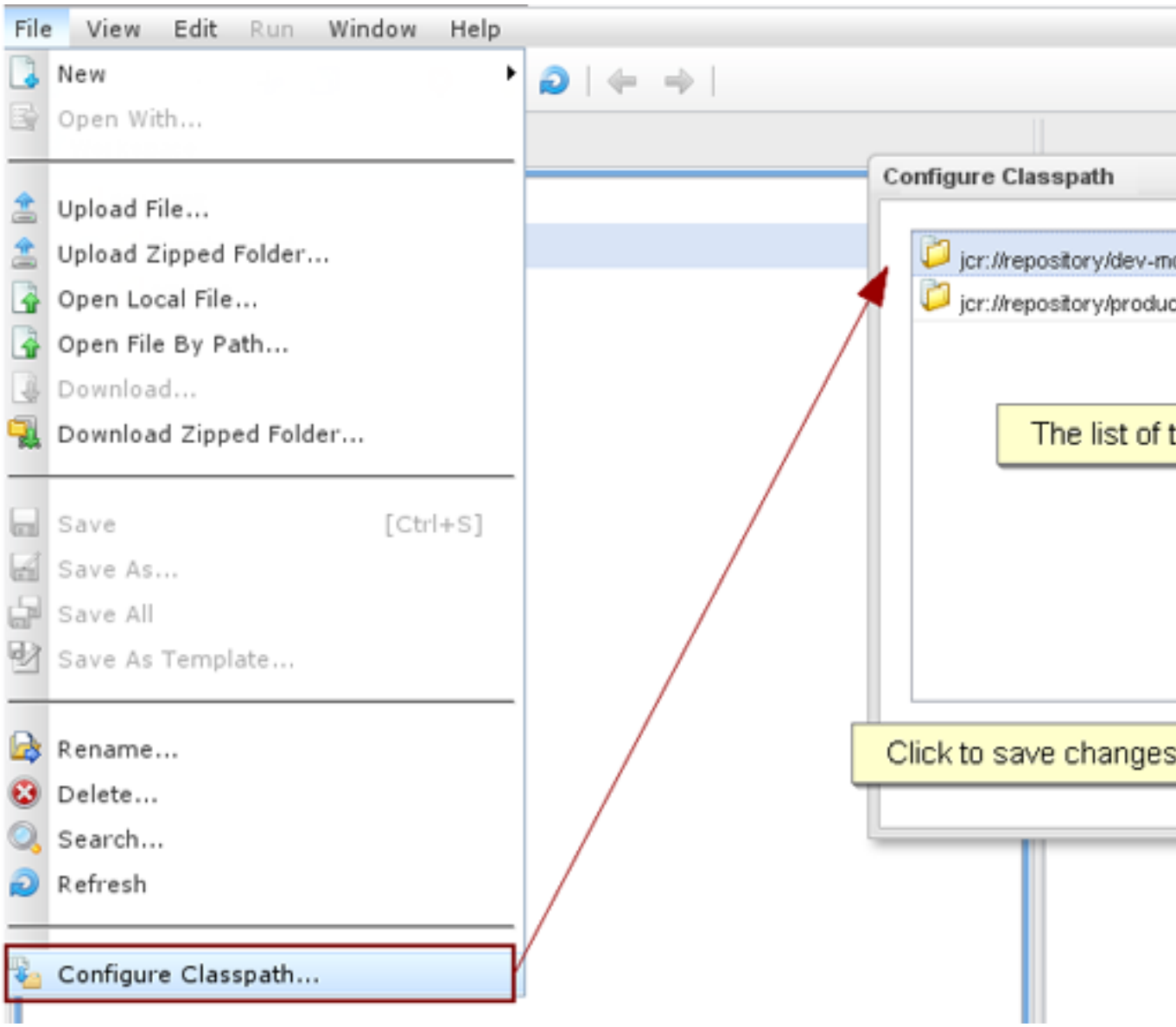


Illustration. Configure classpath form

- **Step 3.** Add or remove source in the tree.
- **Step 4.** Click the **Save** button to save changes in classpath configuration, or the **Cancel** button to discard changes in tree.

Note

If selected in "Workspace Panel" item is not a project or part of it, then you will get an error message.

Add Source to Classpath

To add source to classpath, do as follows:

- **Step 1.** Click the **Add...** button in **Configure Classpath** dialog to open "Choose source path" dialog.
- **Step 2.** Select one or more sources in the tree.
- **Step 3.** Click **Ok** button.

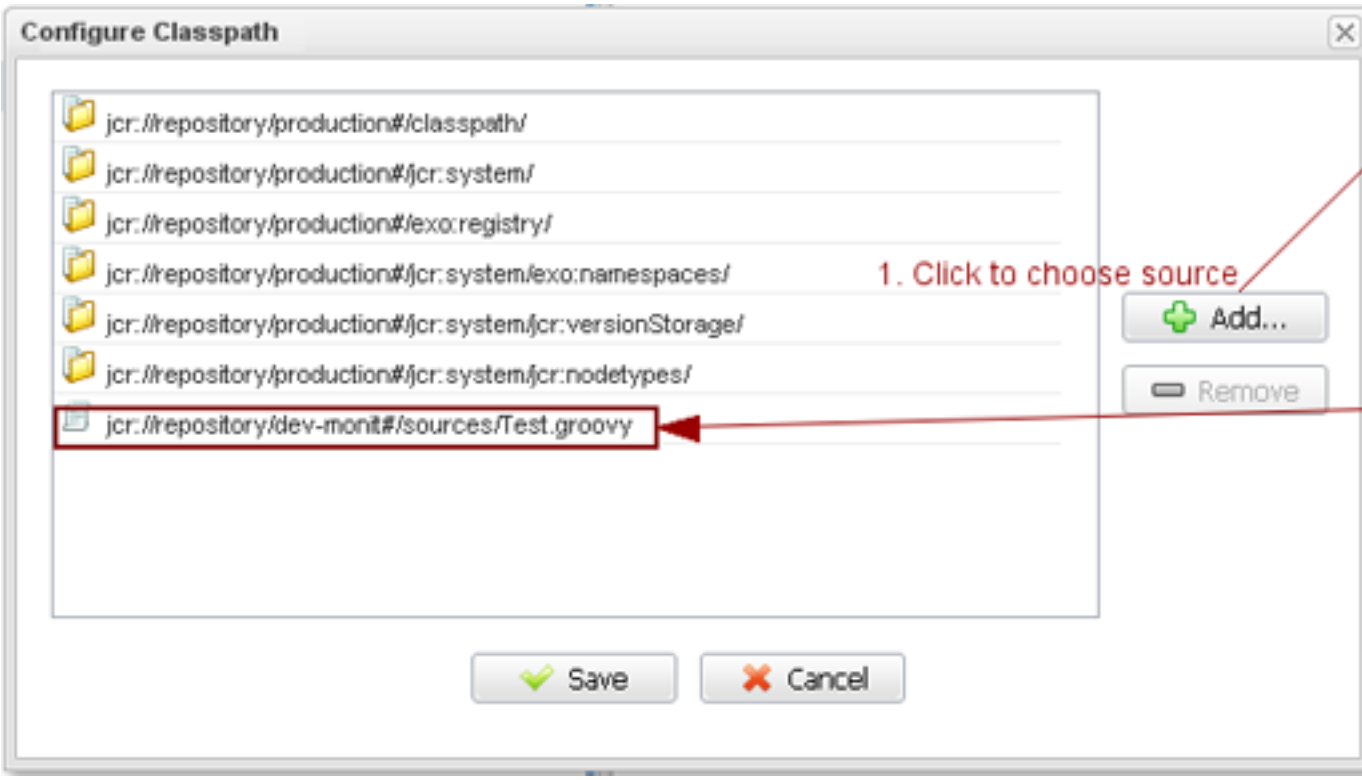


Illustration. Add source to classpath

Use "CTRL" key to select more than one item in the tree.

Note

The "Ok" button is disabled if workspace root folder is selected in the "Choose source path" dialog.

Remove Source from Classpath

To remove source(s) from project's classpath, select source in the tree of **Configure Classpath** dialog and click the **Remove** button.

Note

The "Remove" button is disabled if there is no source selected in the tree.

Example: REST service with complex media types output preview

- **Step 1:** Create, save and deploy the REST Service with next content:

```
// simple groovy script
import javax.ws.rs.Path
import javax.ws.rs.POST
import javax.ws.rs.Produces
import javax.ws.rs.Consumes
import javax.ws.rs.PathParam
import javax.ws.rs.HeaderParam
import javax.ws.rs.QueryParam

@Path("/testMediaTypes")
public class TestService {
  @POST
  @Consumes("application/xml")

  @Produces("text/html")
  @Path("InnerPath/{pathParam}")
  public String post1(@PathParam("pathParam") String pathParam,
    @HeaderParam("Test-Header1") String testHeader,
    @QueryParam("Test Query Parameter 1") String testQueryParam,
    String body) {
      return "PathParam 1:" + pathParam + "; Test Query Parameter 1: " + testQueryParam
      + "; Test-Header1: " + testHeader + "; Body: " + body;
  }

  @POST
  @Consumes("application/xml")

  @Produces("application/json")
  @Path("InnerPath/{pathParam}")
  public String post2(@PathParam("pathParam") String pathParam,
    @HeaderParam("Test-Header2") String testHeader,
    @QueryParam("Test Query Parameter 2") String testQueryParam,
    String body) {
```

```
    return "PathParam 2:" + pathParam + "; Test Query Parameter 2: " + testQueryParam
    + "; Test-Header2: " + testHeader + "; Body: " + body;
  }

}
```

- **Step 2:** Select **Run > Launch REST Service...** on the top menu, or click the **Launch REST Service...** on the right corner of the toolbar.
- **Step 3:** Select the **Path** field as "/testMediaTypes", method OPTIONS, then click the **Send** button.

The OPTIONS-request is sent. You will see the response in the *Output Tab*. For example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application xmlns="http://research.sun.com/wadl/2006/10">
  <resources base="http://192.168.0.8:8080/rest">
    <resource path="/testMediaTypes">
      <method name="OPTIONS">
        <response>
          <representation mediaType="application/vnd.sun.wadl+xml" />
        </response>
      </method>
      <resource path="InnerPath/{pathParam}">
      <param xmlns:xs="http://www.w3.org/2001/XMLSchema" type="xs:string" style="template"
    name="pathParam" />
        <method name="POST" id="post1">
          <request>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" type="xs:string" style="header"
    name="Test-Header1" />
          <param xmlns:xs="http://www.w3.org/2001/XMLSchema" type="xs:string" style="query"
              name="Test Query Parameter 1" />
            <representation mediaType="application/xml" />
          </request>
          <response>
            <representation mediaType="text/html" />
          </response>
        </method>
        <method name="POST" id="post2">
          <request>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" type="xs:string" style="header"
    name="Test-Header2" />
          <param xmlns:xs="http://www.w3.org/2001/XMLSchema" type="xs:string" style="query"
              name="Test Query Parameter 2" />
```

```
            <representation mediaType="application/xml" />
          </request>
          <response>
            <representation mediaType="application/json" />
          </response>
        </method>
      </resource>
    </resource>
  </resources>
</application>
```

- **Step 4:** Call the **Launch REST Service...** dialog again and select the another **Path** field:

```
"/testMediaTypes/InnerPath/{pathParam}"
```

- **Step 5:** Select the **Response Media Type =** "text/html" item.
- **Step 6:** Enter "/testMediaTypes/InnerPath/value1" in the **Path** field. In the *Query Parameter* tab, set "Test Query Parameter 1"="value2". In the *Header Parameter* tab, set "Test-Header1"="value3". In the *Body* tab, type "example".
- **Step 7:** Click the **Send** button.

The request is created and then sent. You will see the response in the *Output Tab*:

```
[OUTPUT] - -Status - - - - - - - -
200 OK
- -Headers- - - - - - - -
Server : Apache-Coyote/1.1
Content-Type : text/html
Transfer-Encoding : chunked
Date : Mon, 05 Jul 2010 09:06:55 GMT
- -Text - - - - - - - - -
PathParam 1:value1; Test Query Parameter 1: value2; Test-Header1: value3;
 Body: example
```

Operations With POGO File

eXo IDE supports validating and classpath configuration of POGO files that is similar to REST Service.

Operations With Google Gadget

The Google Gadget file is in the .xml format with the special MIME type "application/x-google-gadget". In eXo IDE, you can do the following actions on the Google Gadget files:

- Create **.xml** files.
- Edit **.xml** files in the *Code* or *WYSIWYG Editor*.
- Preview them in the *Preview Tab*.

Create Google Gadget

To create the Google Gadget within eXo IDE, do one of the following ways:

1. Click the **New** icon on the toolbar and then select **Google Gadget** from the drop-down menu.

2. Select **File > New > Google Gadget** on the top menu.

3. Open the existing gadget from the local system by selecting **File > Open Local File** on the top menu.

4. Upload the existing gadget directly to JCR by selecting **File > Upload File...** on the top menu.

For the third and forth ways, you must select the MIME type: *"application/x-google-gadget"*.

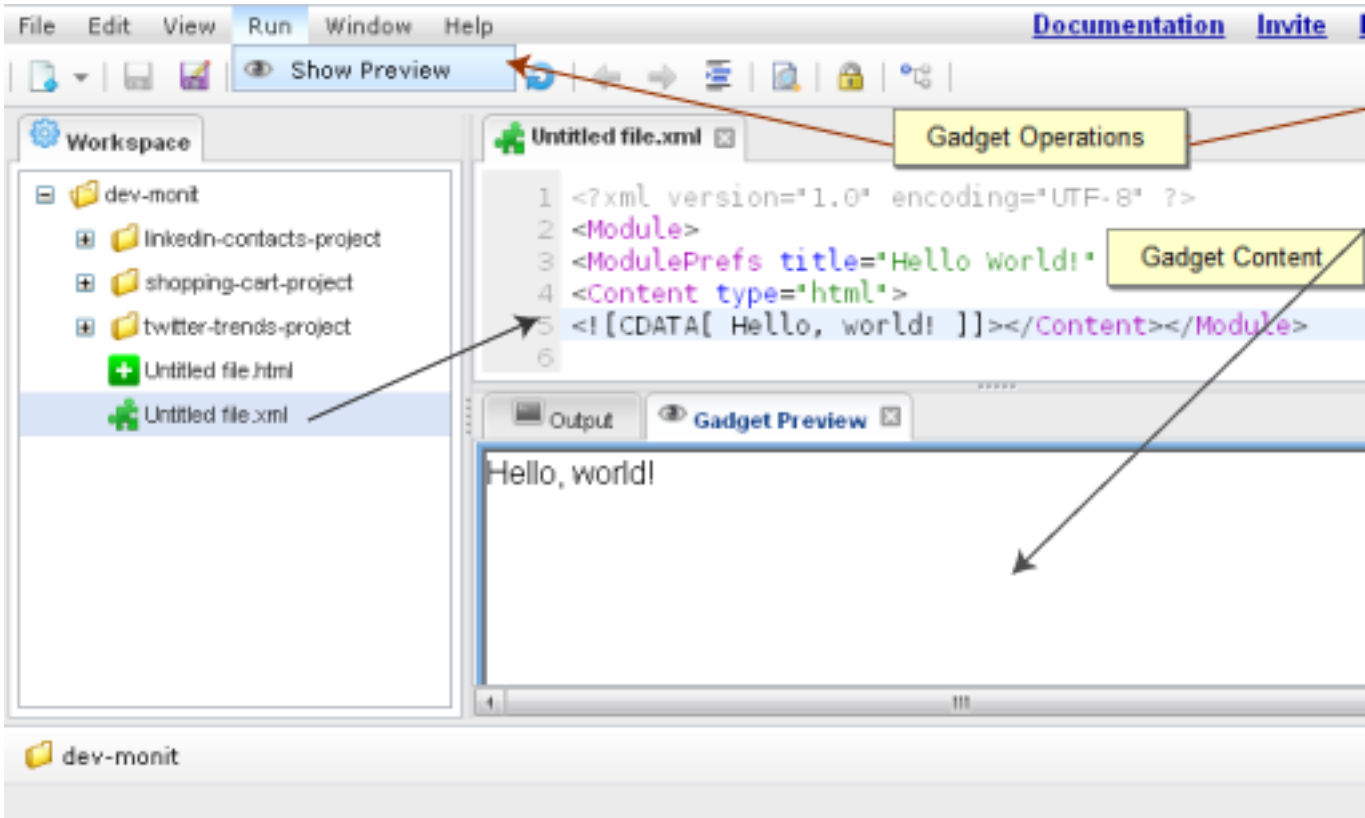


Illustration. Google Gadget Operations

Preview Google Gadget

To preview the Google Gadget, do as follows:

- **Step 1:** Open your desired Google Gadget file which was previously saved.
- **Step 2:** Click the **Show Preview** button on the toolbar; or select **Run > Show Preview** on the top menu.

The gadget is opened in the *Gadget Container* of the *Preview Tab* as described in the **Illustration. Google Gadget Operations** above.

Operations With Templates

The *Template* file is a combination of HTML code and Groovy code blocks. In eXo IDE, you can create and edit those files in editor.

Create and Edit Template

To create a new Gr template, do as follows:

- **Step 1:** Click the **New** icon on the toolbar and then select **Template** from the drop-down menu; or select **File > New > Template** on the top menu. A new file is opened in the editor.
- **Step 2:** Add content.
- **Step 3:** Save the file by selecting **File > Save As** on the top menu, or by clicking the **Save As** icon on the toolbar.

Both *Outline Panel* and Autocomplete form can be used to edit the Groovy Template.

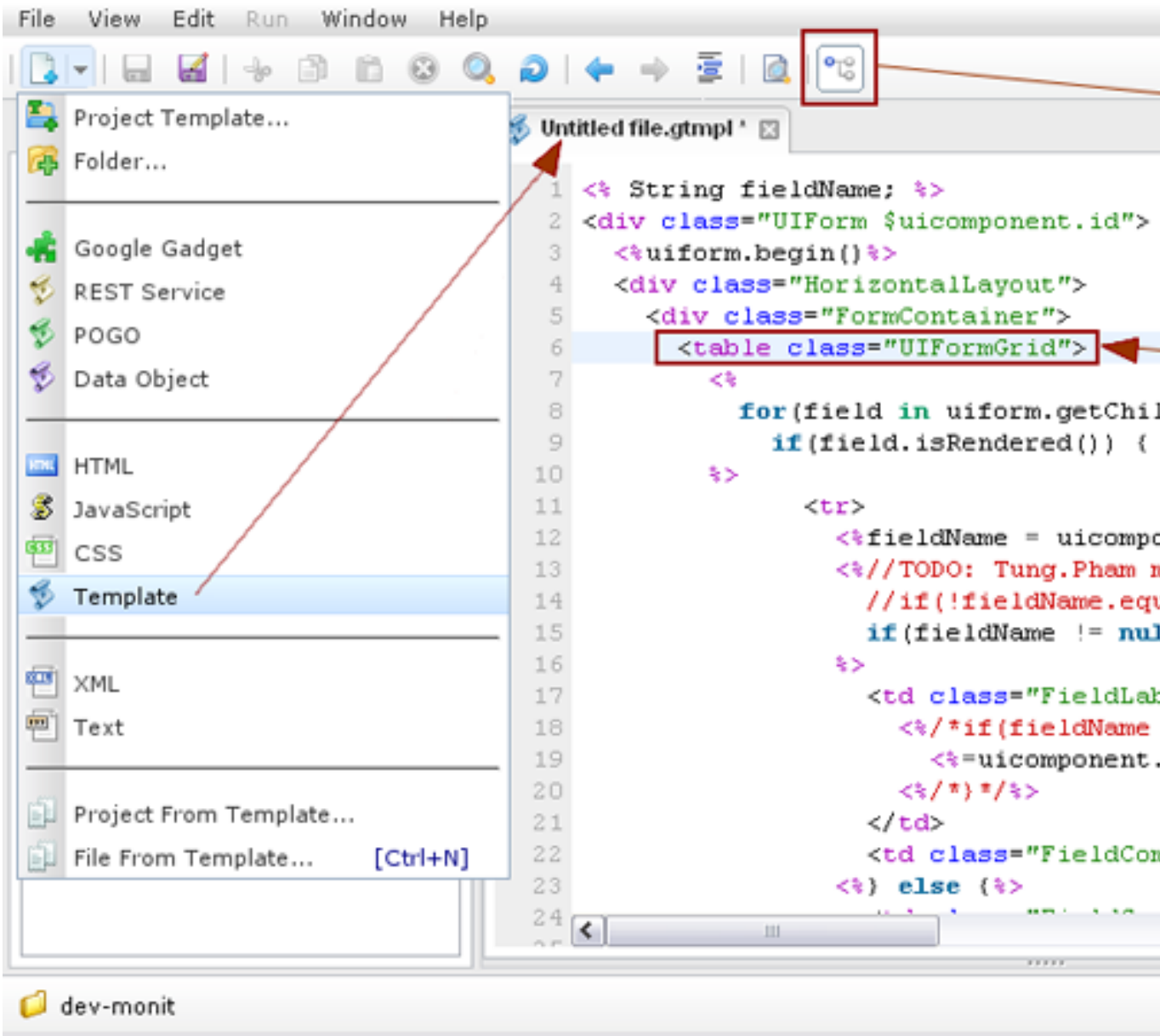


Illustration. Operations with ECM templates

View Template's Properties

To view properties of the Template, click the **Show Properties** button on the right corner of the toolbar.

Preview Template

To preview the template, do as follows:

- **Step 1:** Open your desired Groovy template in the *Content Panel*.

- **Step 2:** Click the **Show Preview** button on the toolbar; or go to **Run > Show Preview** on the top menu.

The selected template is opened in the *Gadget Container* of the *Preview Tab* as below:

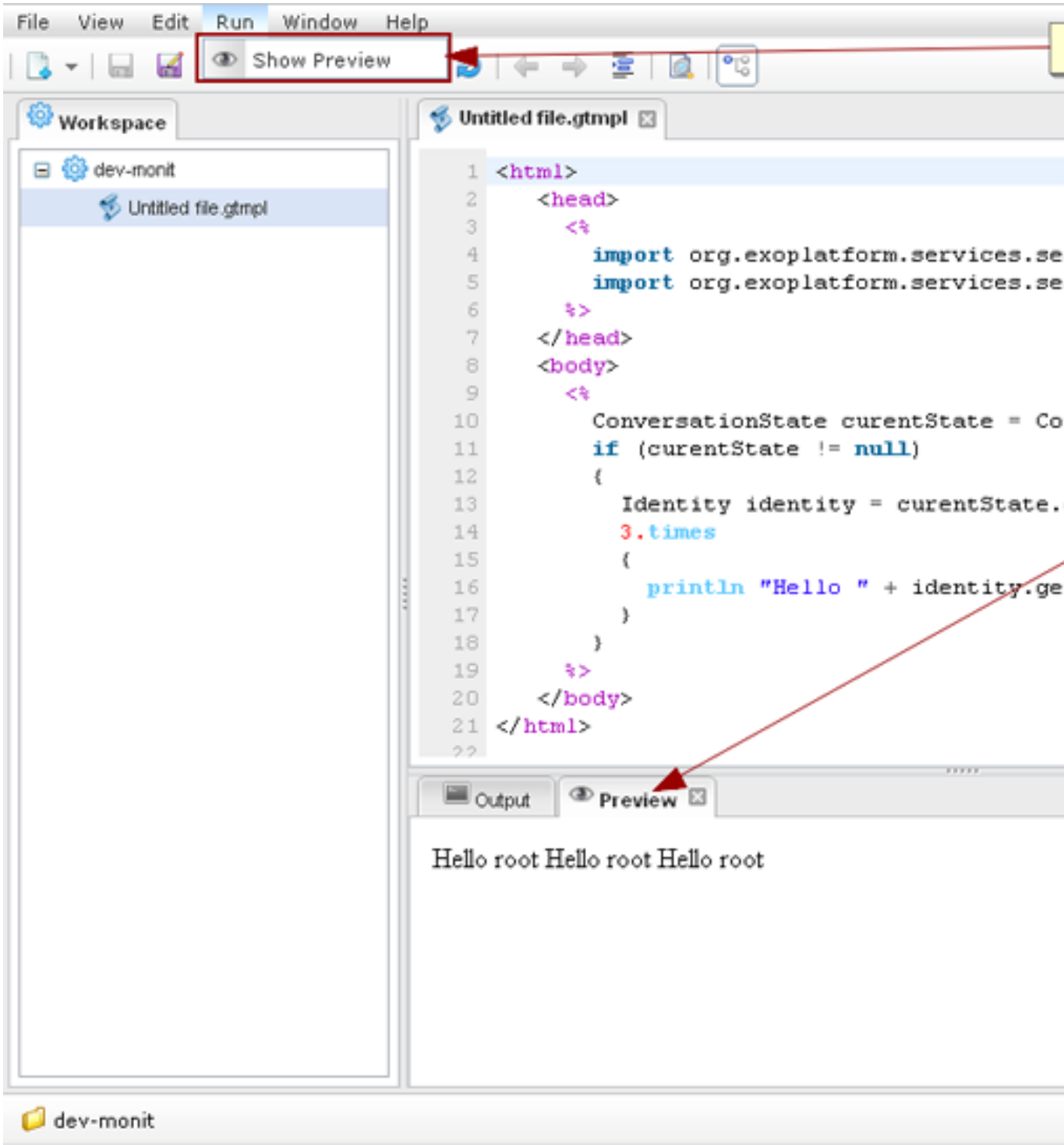


Illustration. Groovy Templates preview

Operations With Netvibes Widget

The *Netvibes widget* uses Universal Widget API (UWA) - a powerful framework for Web widgets development - for both Netvibes widgets and many various environments.

The UWA widget consists of combinations from XHTML/XML, JavaScript/Ajax, CSS code with the MIME type **application/x-uwa-widget**. In eXo IDE, you can create the Netvibes widget, preview it and deploy to *Netvibes Ecosystem* [http://eco.netvibes.com].

Note

If you want to get acquainted with writing your own widgets, refer page *http://dev.netvibes.com/doc/.*

Create Netvibes Widget

To create a Netvibes widget, click the **New** icon on the toolbar and then select **Netvibes Widget** from the drop-down menu, or go to **File > New > Netvibes Widget** on the top menu.

The new Netvibes Widget file is opened in the editor, enabling you to edit and save it.

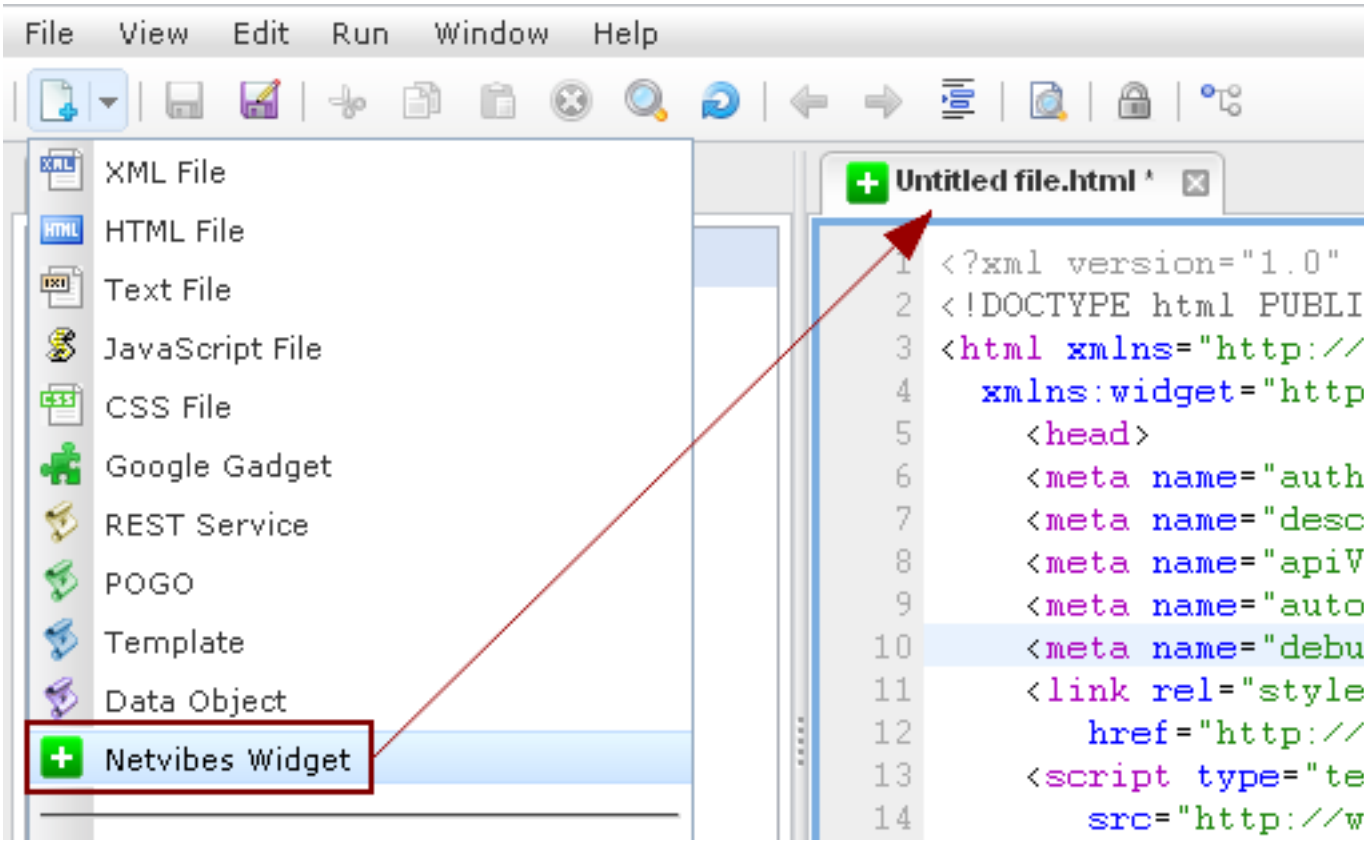


Illustration. Create new Netvibes widget

Preview Netvibes Widget

To preview the Netvibes widget, do as follows:

- **Step 1:** Open your desired widget file in the *Content Panel*.
- **Step 2:** Click the **Show Preview** button on the toolbar; or go to **Run > Show Preview** on the top menu.

The selected widget is opened in the *Preview Tab*.

Deploy Netvibes Widget To Netvibes Ecosystem

Netvibes widgets are placed in the remote system called *Netvibes Ecosystem* [http://eco.netvibes.com]. This is the location where such widgets can be found and compiled for other platforms.

There are a couple of ways to deploy a *Netvibes widget*. You can do it manually, or use the deploy mechanism. Manual deployment is easier because you don#t need to get an API key from Netvibes.

Manual deployment:

- **Step 1:** Get the url of your *Netvibes widget*. To do this, select your widget and click on the menu **View > Get URL**. Copy the url.
- **Step 2:** Go to the *netvibes ecosystem* [http://eco.netvibes.com/widgets], click on **Create or submit** and **Submit Widget**.
- **Step 3:** Paste the url of your gadget, and don#t forget to remove the #**/private**# part from it. If you do not do this, *Netvibes* will not be able to access it.
- **Step 4:** Fill-in the form and finish the process. The widget is now available in the ecosystem.

For deployment using our wizard, you need to contact Netvibes at `<business@netvibes.com>` and ask for your API key.

- **Step 1:** Get the url of your netvibes widget. To do this, select your widget and click on the menu **View > Get URL**. Copy the url..
- **Step 2:** Click on **Deploy UWA widget to Ecosystem** button on the toolbar; or go to **Run > Deploy UWA widget** on the top menu to open the **Deploy UWA widget to Ecosystem**:

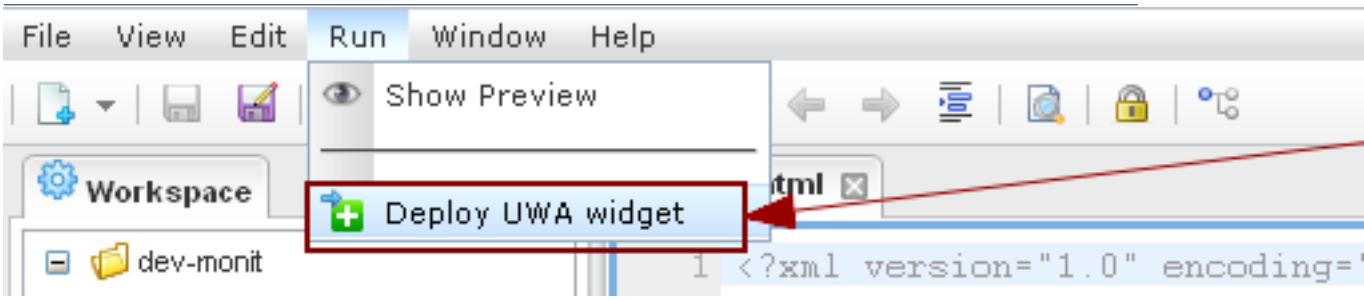


Illustration. Deploy UWA widget to Netvibes Ecosystem

- **Step 3:** Paste the url of your gadget, and don#t forget to remove the "/**private**" part from it, or netvibes will not be able to access it

- **Step 4:** Enter the information about the widget in the fields in the **Step 2** form. The fields marked with asterisks (), including "**Title**", "**Description**", "**Main language**", "**Most appropriate region**" and "**Most appropriate category**" are required.

Click **Next** after completing your details.

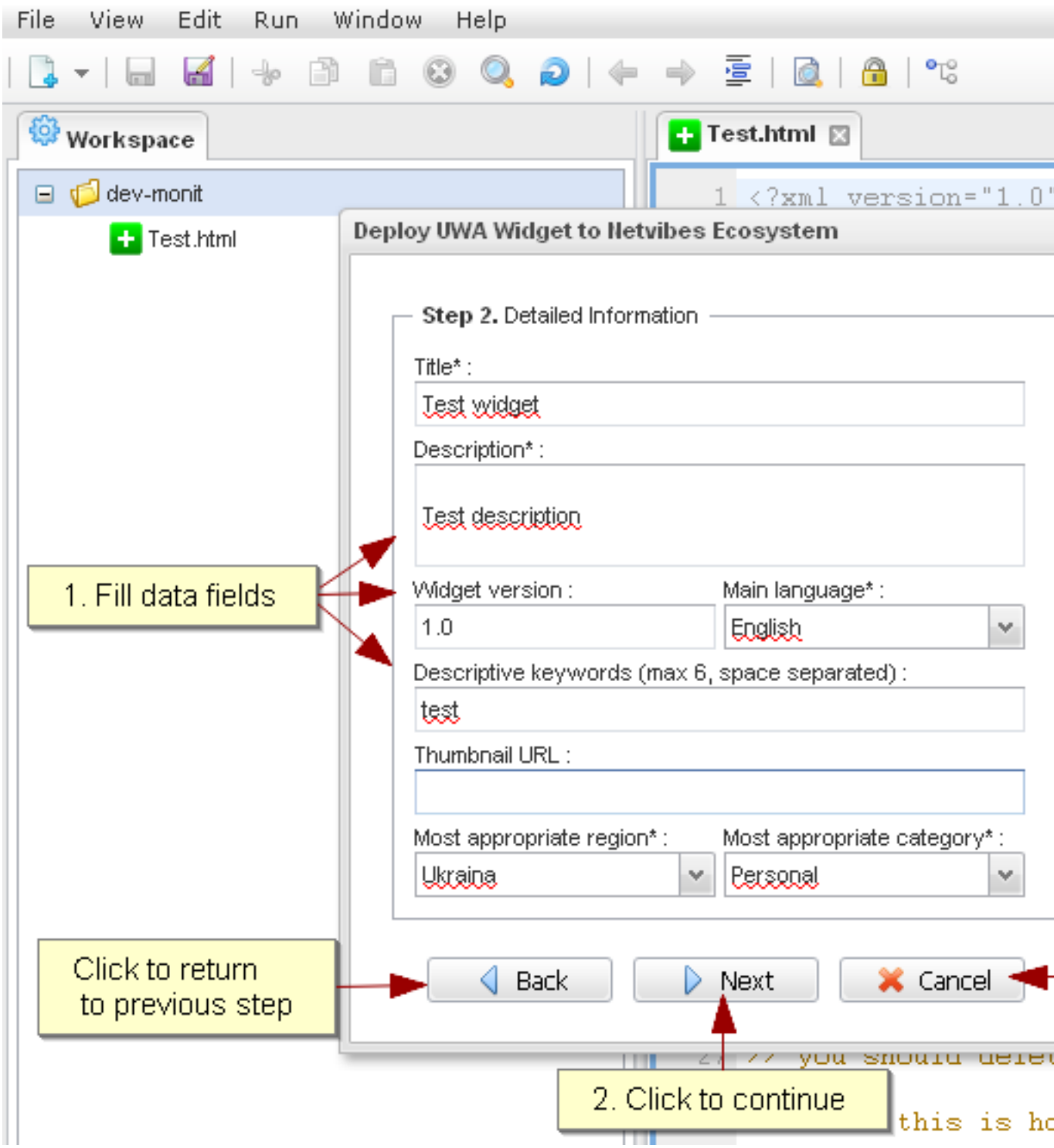


Illustration. Dialog "Step 2. Detailed Information"

- **Step 5:** Enter your netvibes credentials and the API key/secret that netvibes gave you by email. All these fields are mandatory.

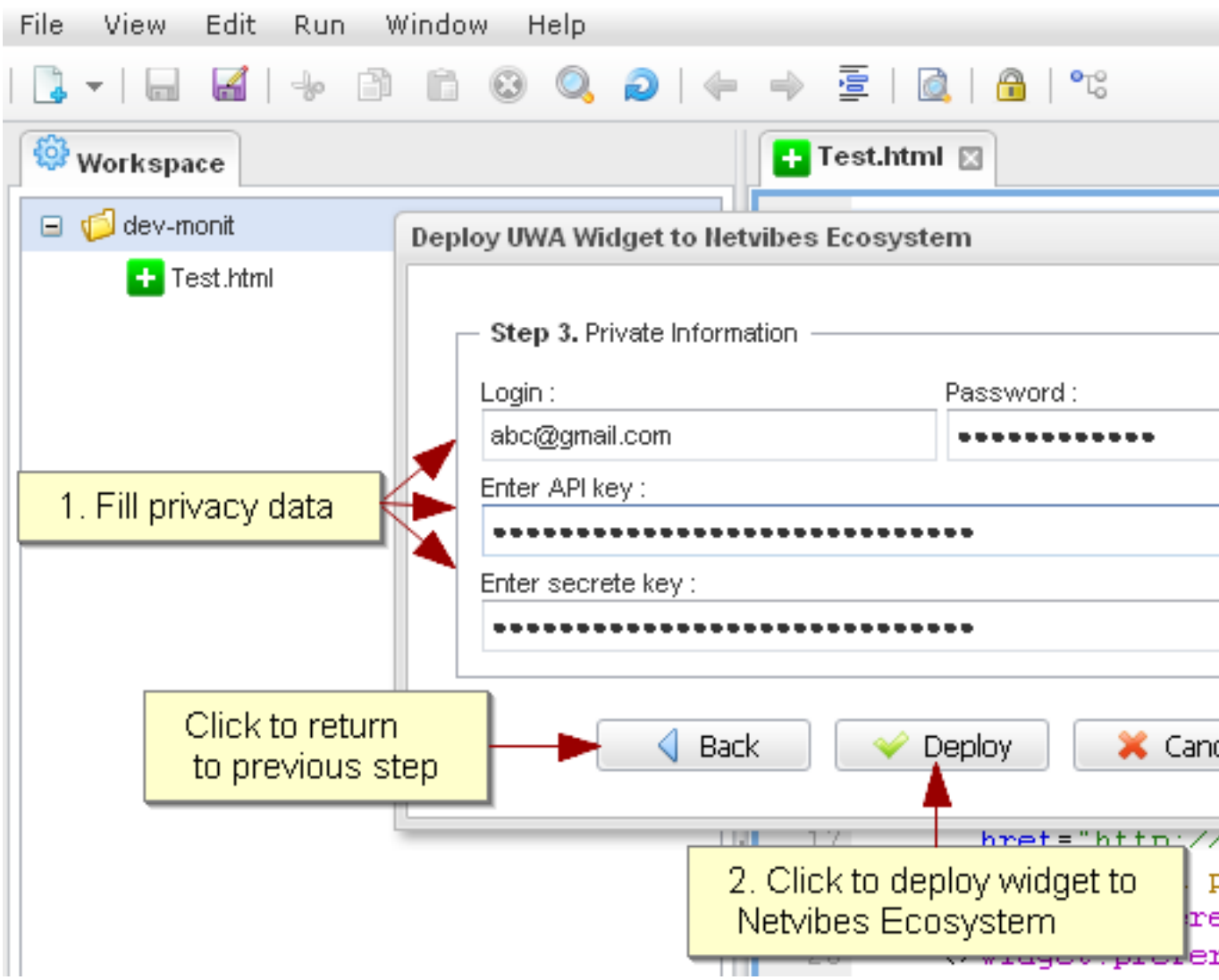


Illustration. Dialog "Step 3. Private Information"

Details:

Table 11.1.

Field	Description
Login	Your login at Netvibes Ecosystem.
Password	Your password at Netvibes Ecosystem.
Api Key	The string generated by Netvibes and used for identifying you.
Secret Key	The string generated by Netvibes and used for submitting your actions.

Click the "Back" button to return to the previous step to edit your entered information.

- **Step 6:** Click the **Deploy** button to start deploying the widget to the *Netvibes Ecosystem*.

The result is displayed in the *Output Panel*. If the deployment is successful, you will receive the message below:

```
[INFO]*[http://78.137.4.16:53081/rest/ide-vfs-webdav/repository/dev-monit/
TestWidget.html] deployed successfully.
```

If unsuccessful, you will receive the error message which varies, depending on each case.

```
[ERROR] Error: The link already exists.
```

Note

The deployed widget can be found at the Ecosystem page (*http://eco.netvibes.com/mine/widgets)* on the "My creations" tab.

View Netvibes Documentation

There is a convenient way to view the developer's documentation, while creating your widget in eXo IDE. When opening Netvibes widget in editor, click **Show Documentation** Toolbar item or select **View > Documentation**.

Top menu item and a panel with loaded documentation will be opened at the right part of the application.

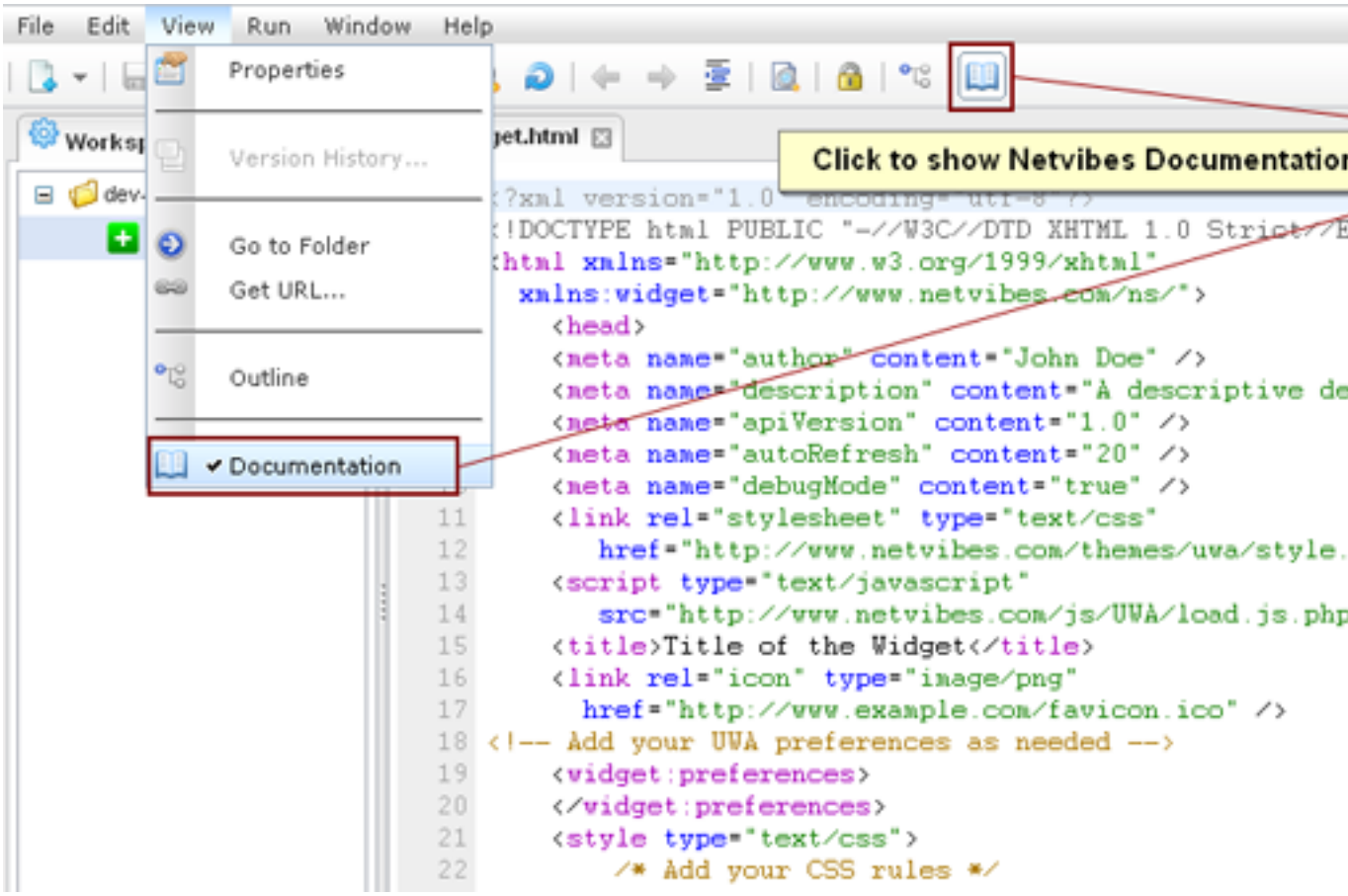


Illustration. View Netvibes Documentation

Chromattic Data Object Usage

Chromattic Data Object files has MIME type **application/x-chromattic+groovy**. With eXo IDE you can create *Chromattic Data Object* file, create and preview node type.

Create Data Object

To create new data object click the **New > Data Object** *Toolbar* button; or go to **File > New > Data Object** on the top menu. New file will be opened in editor, so you will be able to edit and save it.

Preview Node Type

To generate the node type definition and preview it, do the follows:

Step 1. Click **Preview node type** button on *Toolbar* or go to **Run > Preview node type** from the top menu.

Step 2. Select format of node type in the appeared dialog window:

- **EXO** standard XML content type to be used with XML-formatted node type streams;
- **CND** experimental content type for the compact node type definition files.

Step 3. Click **Generate** button. The generated node type definition will be displayed in the separate tab of *Operation Panel* with title **Preview node type**:

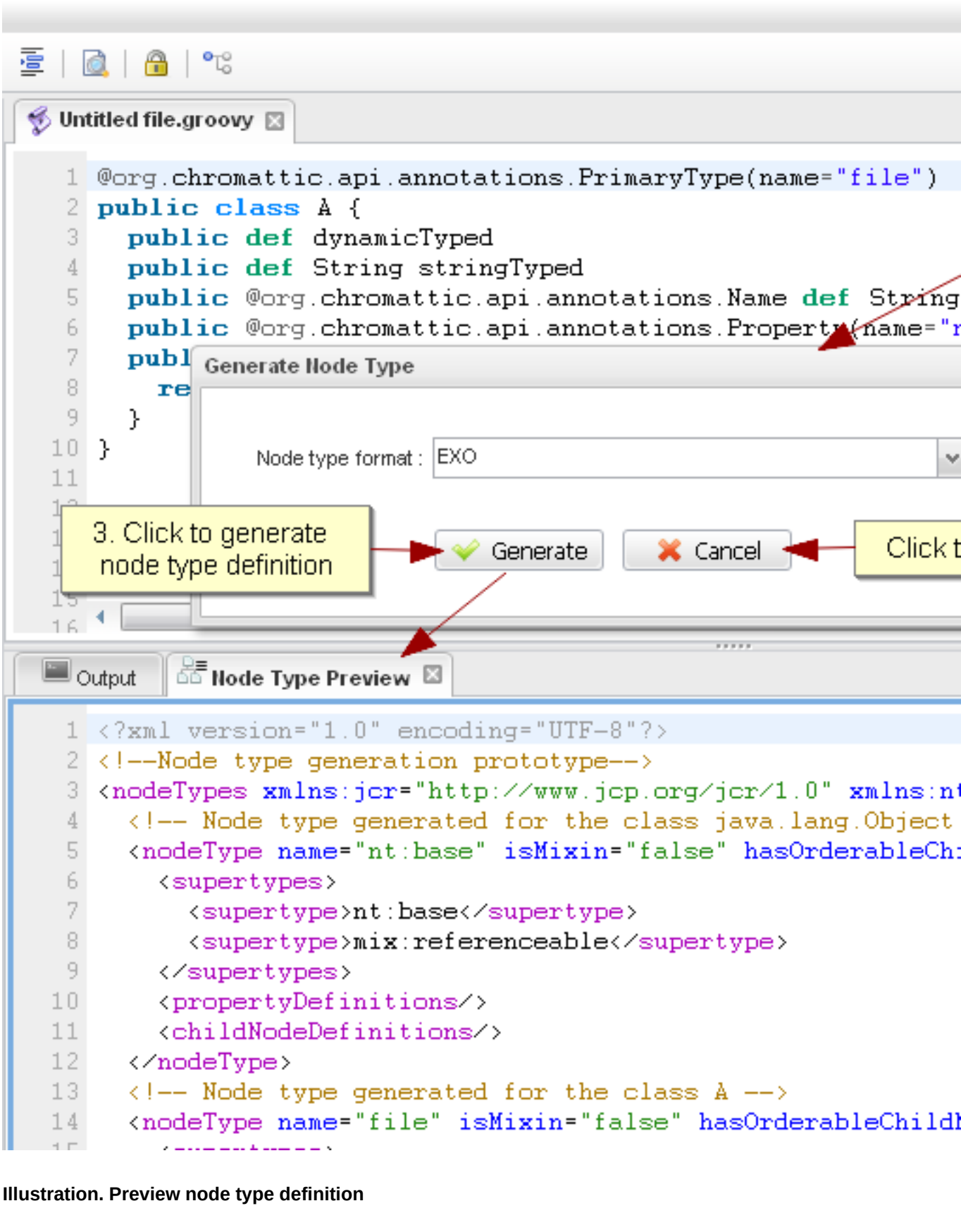


Illustration. Preview node type definition

Create Node Type

To create new node type, do as follows:

- **Step 1.** Click **Deploy node type** button on *Toolbar* or go to **Run > Deploy node type** on the top menu.
- **Step 2.** Select required format of node type in the appeared dialog window.
- **Step 3.** Choose the action in case if such node type already exists:

- **ignore if exists** if the pointed node type already exists, then ignore the creation operation. - **fail if exists** if the pointed node type already exists, then fail. - **replace if exists** if the pointed node type already exists, then replace it with new one.

- **Step 4.** Click **Deploy** button.

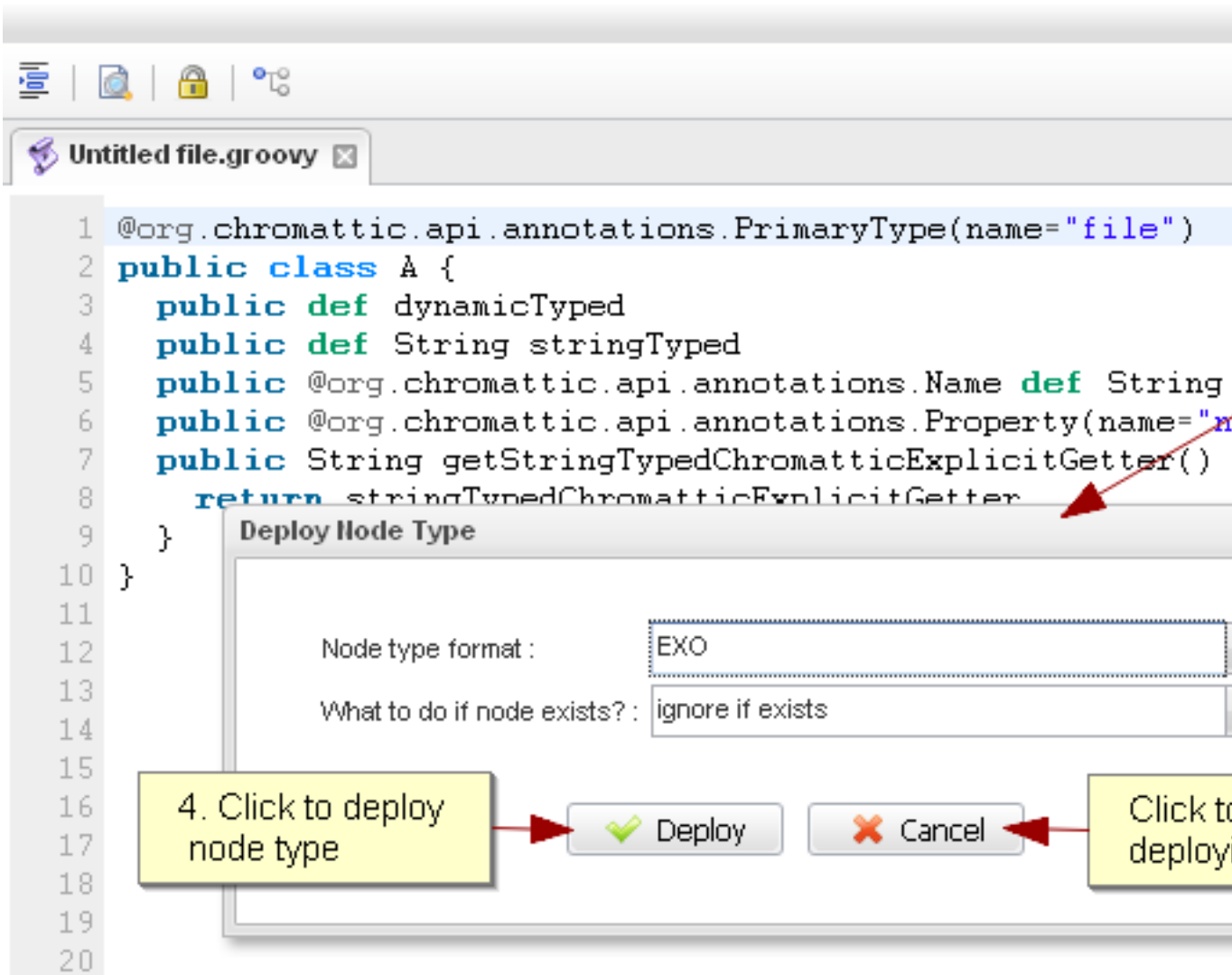


Illustration. Deploy the node type definition

If node type has been deployed successfully, you will see the message **Node type is successfully deployed**. Otherwise the error message will be shown.

Note

At present, CND format is not supported for the deploy operation.

Versioning

eXo IDE provides the versioning that enables you to realize the versions history of the file:

- view version's content;
- view the list of versions;
- restore to an accurate version.

A new version of the file is created each time after the file content has been saved by clicking the **Save** button. There are no versions in version history just after the file had been created only.

View Version History

To view the version history of the file which is being opened/selected, click the **View Item Version History** button at the right part of the toolbar; or select **View > Version History...** on the top menu. The *Version Panel* with content of the latest version are opened at the right column of the eXo IDE window:

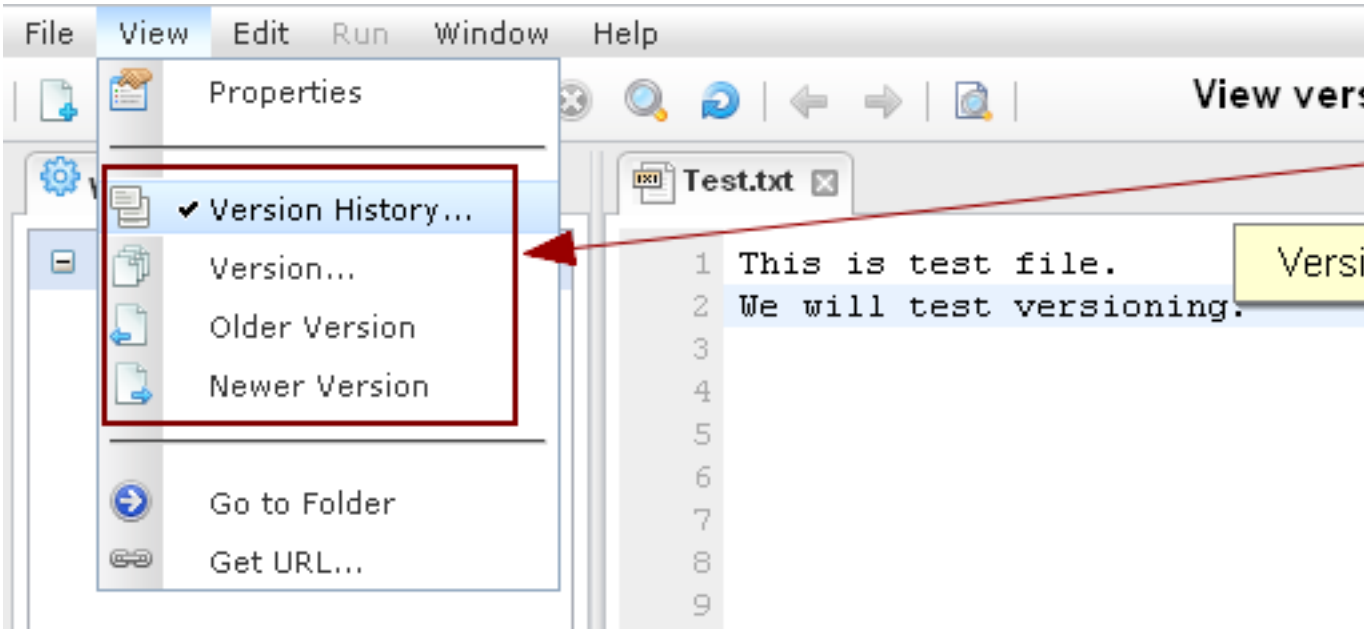


Illustration. View version history of the active file

The version name is displayed in the title of the *Version Panel*.

Note

- If the selected/opened file is not versioned yet, the "View Version History" function will be disabled.

- The file version is opened in the read-only mode. So, you can not edit it.

- If any files are opened, the "Version Panel" of the previously opened file will be closed automatically.

Version Navigation

With eXo IDE, you can navigate to any older or newer versions of the opened file.

To view the older/newer version of the opened file, click the **View Older Version/View Newer Version** button on the toolbar, or select **View > Older Version/Newer Version** on the top menu. If there is no newer/older version, the **View Newer Version/View Older Version** command will be disabled respectively.

If you want to view the another version from the versions history, do as follows:

- **Step 1:** Click the **View Item Version...** button on the toolbar; or select **View > Version...** on the top menu.
- **Step 2:** Select the target version.
- **Step 3:** Click the **Open** button or double-click the version item.

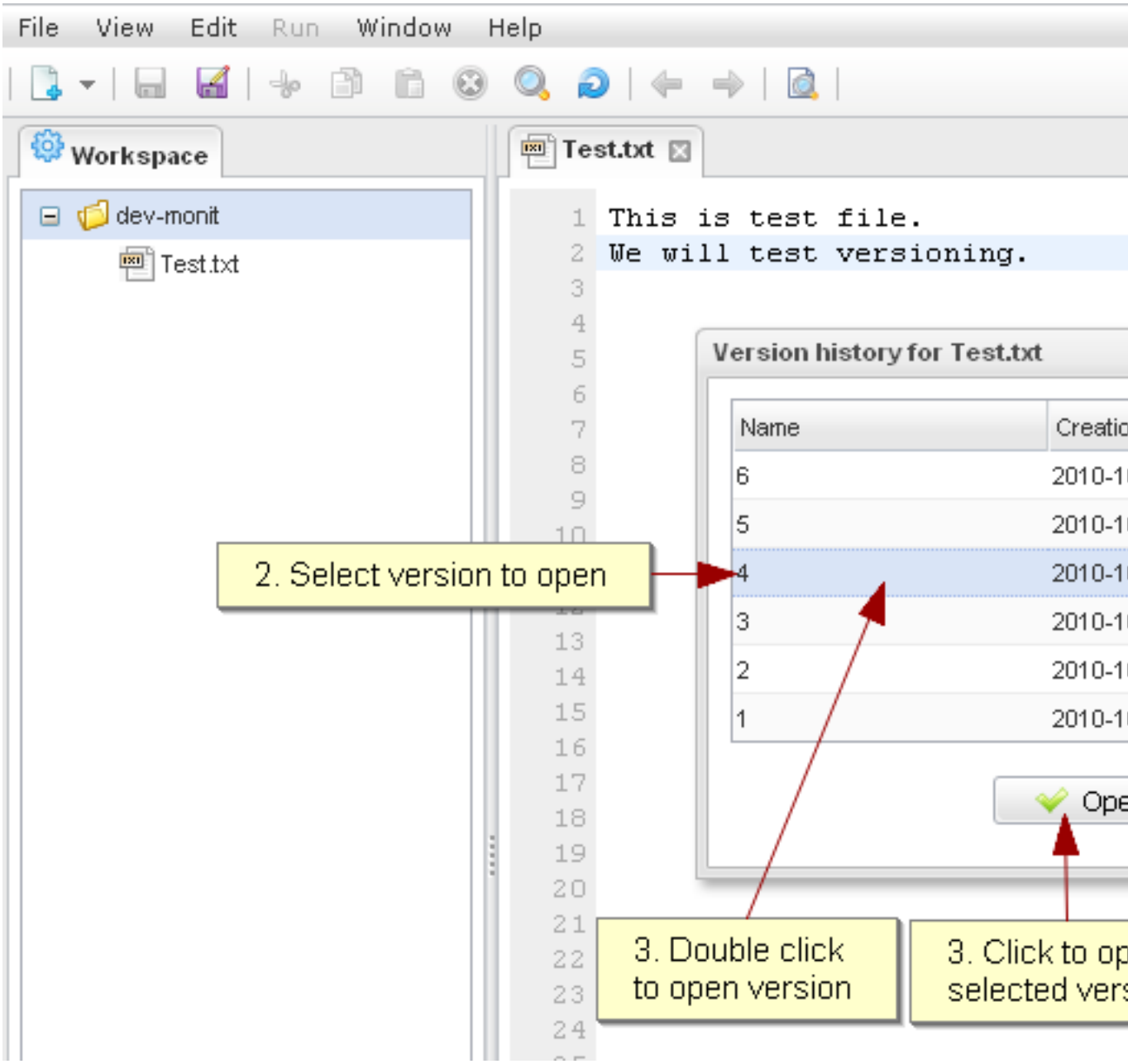


Illustration. Open version from version list

Versions are listed in the reverse order, with the latest version first.

Restore File to the Older Version

To restore a file to the older version, do as follows:

- **Step 1:** Click the **Restore To Version** button at the right corner of toolbar, or select **File > Restore To Version** on the top menu.
- **Step 2:** Click **Yes** in the confirmation dialog to restore to the version.

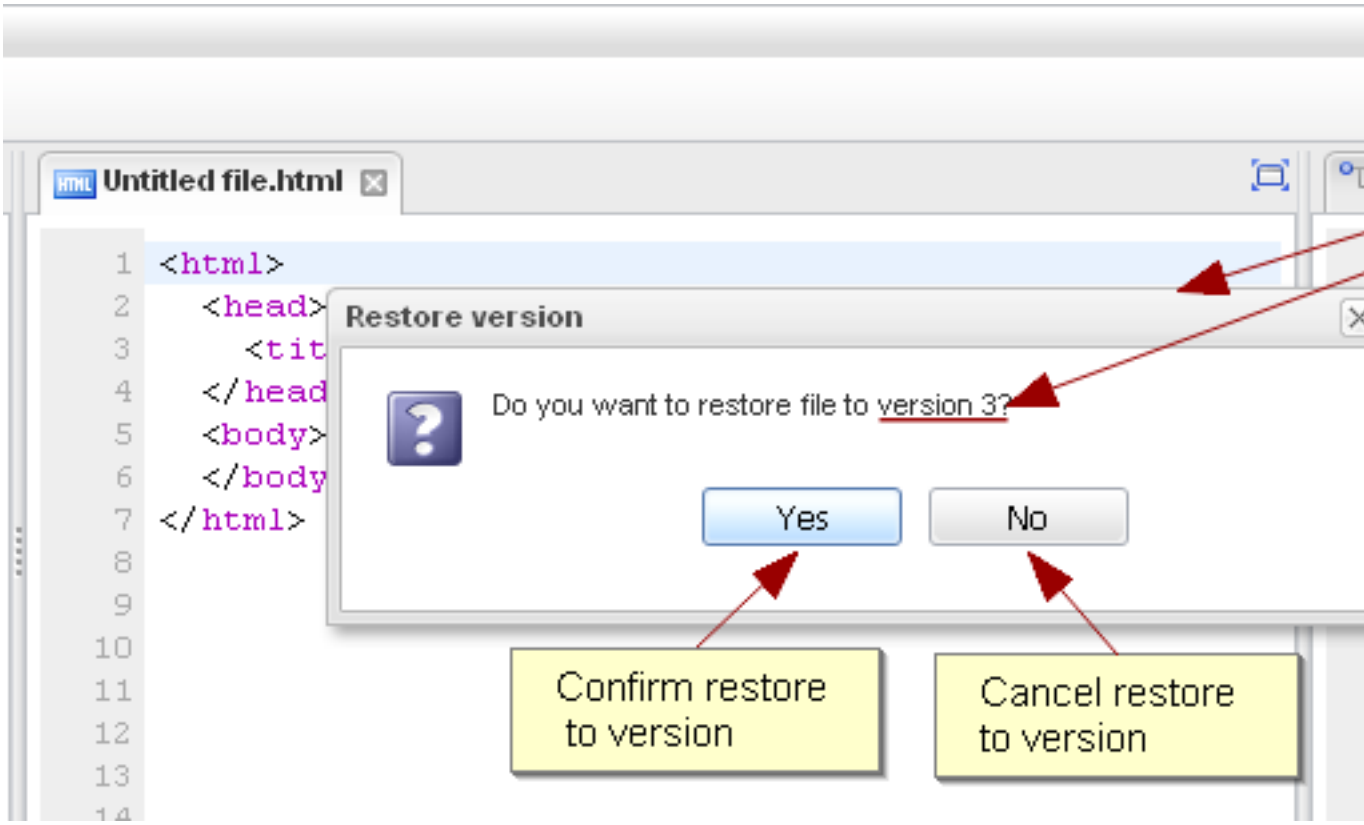


Illustration. Restore file to version

Note

The "Restore version" command is disabled, if the latest version is displayed.

Miscellaneous and Tips

View information about eXo IDE

To view information about the current version of eXo IDE, go to **Help > About** on the top menu.

Customize Toolbar

If you want to customize the toolbar, such as adding or removing some buttons, changing their positions or delimiters, select **Window > Customize Toolbar...** on the top menu. This dialog will show typical interfaces and commands set for such operations:

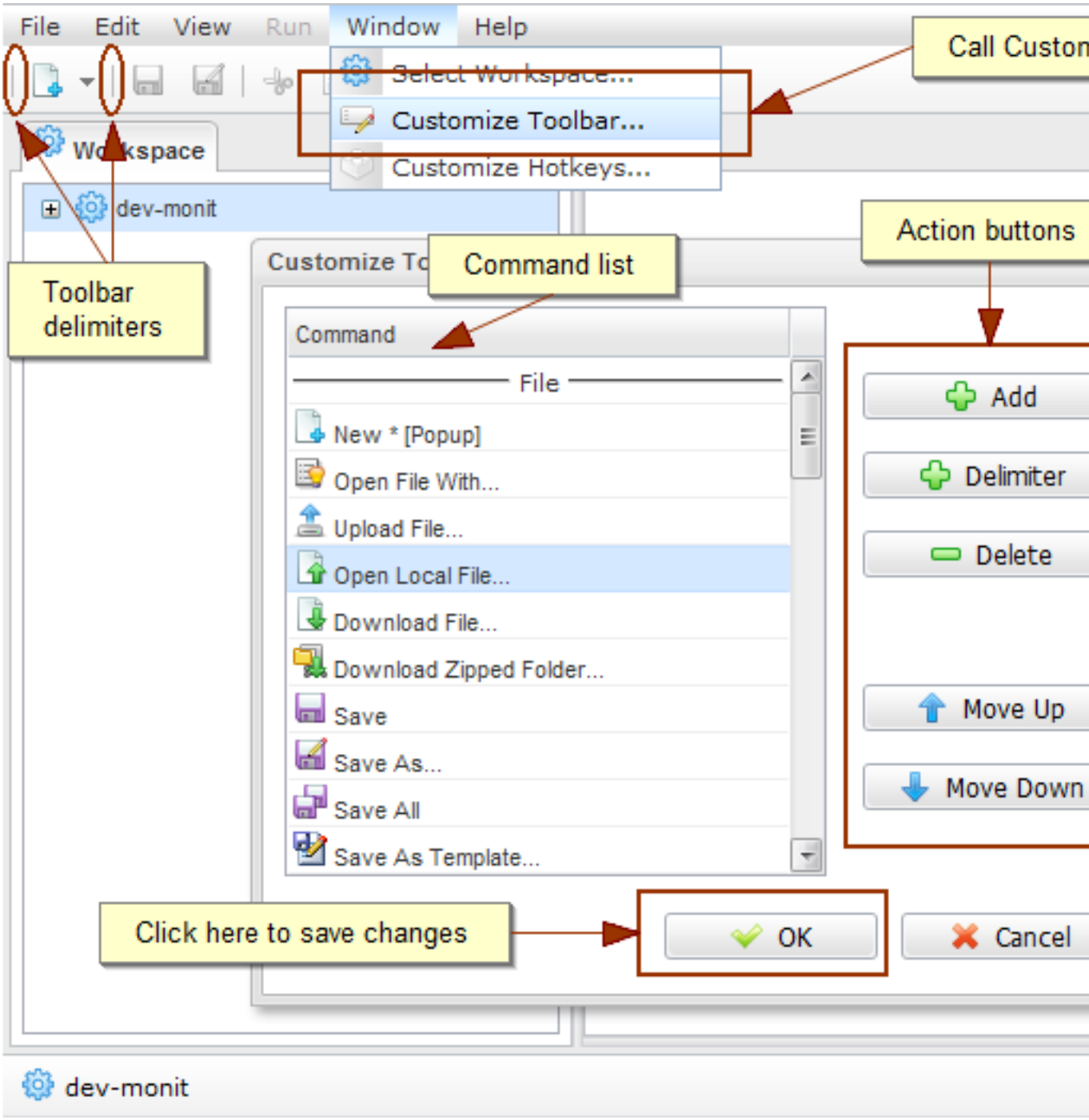


Illustration. Customize Toolbar

Customize Hotkeys

The hotkey manager enables you to customize hotkeys applied for eXo IDE. If your customized hotkeys are matching with browser hotkeys, only eXo IDE hotkeys work out.

If hotkeys of the browser and of eXo IDE are not overidden, they will work as usual.

To customize your own hotkeys, do as follows:

- **Step 1:** Go to **Window > Customize Hotkeys...** on the top menu to open the **Customize hotkeys** dialog.

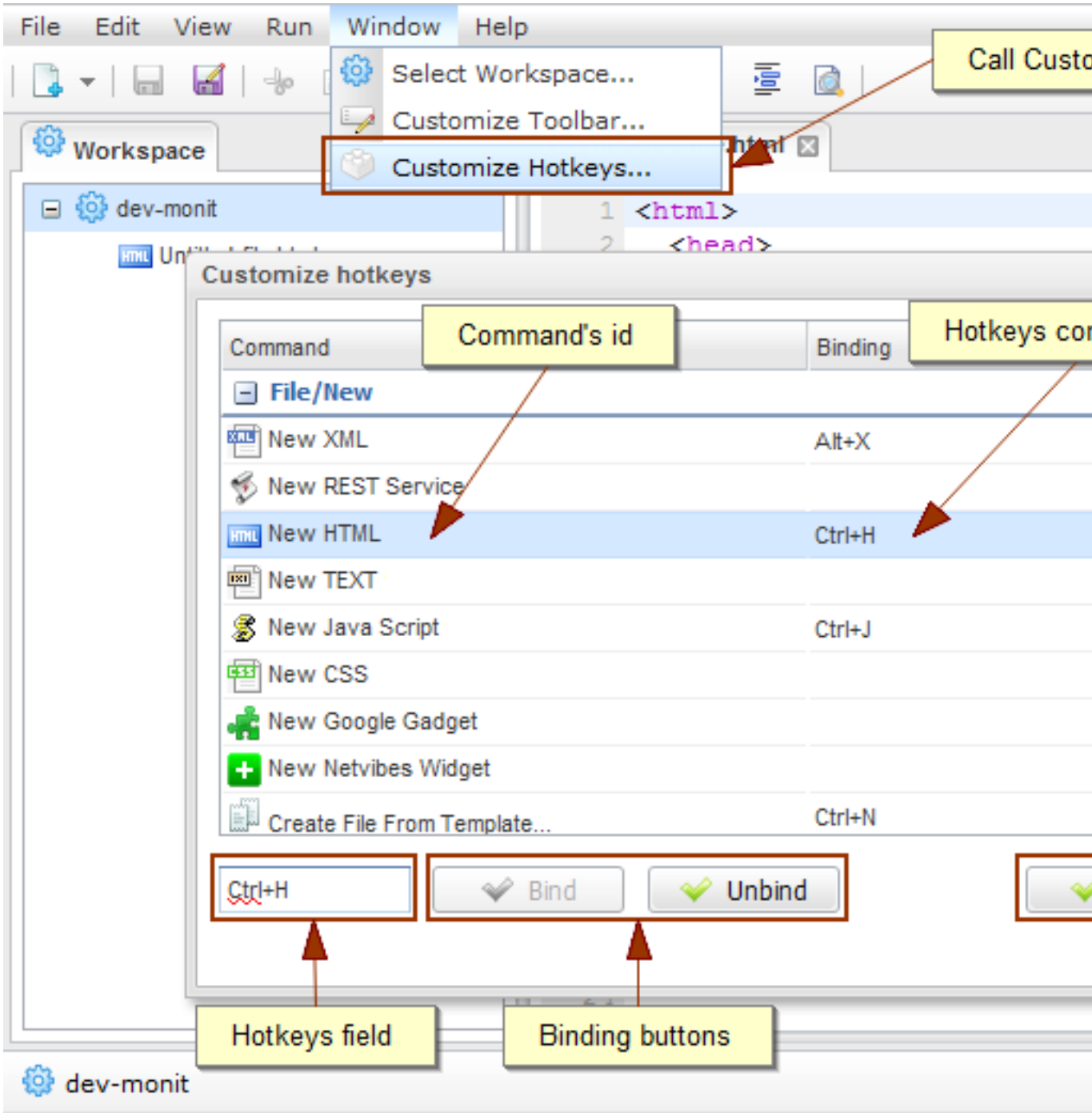


Illustration. Customize hotkeys

The **Customize hotkeys** dialog consists of the list grid with all commands divided to groups and their hotkeys.

- **Step 2:** Click your desired action in the **Customize hotkeys** dialog.

- **Step 3:** Enter the hotkeys combination in the **Hotkey** field. The first key should be "Ctrl" or "Alt".

If the hotkeys combination is not already assigned to another command, the **Bind** button will be active. Otherwise, the error message is shown under the **Hotkey** field. For example:

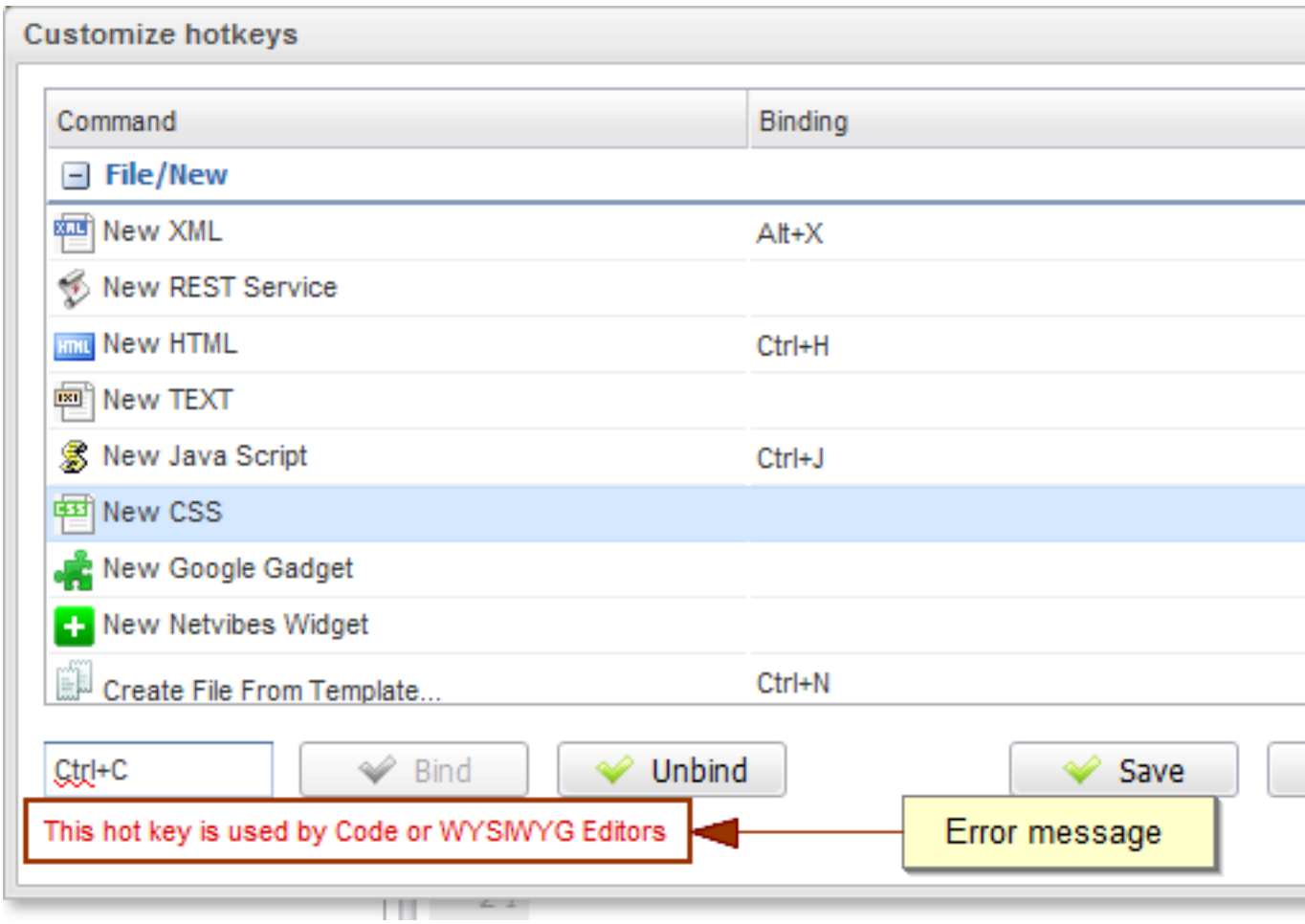


Illustration. Error message

- **Step 4:** Click the **Bind** button.
- **Step 5:** Click **Save** to accept your new settings on the server.

To unbind hotkeys for one command, do as follows:

- **Step 1:** Open the **Customize hotkeys** window.
- **Step 2:** Select your needed command in the list grid.
- **Step 3:** Click **Unbind**.
- **Step 4:** Click **Save** to accept your new settings on the server.

Note

- You can not make any changes for hotkeys of the "Editor hotkeys" group, because they are reserved and used in editors. These hotkeys also can not be reapplied for other commands. To see these hotkeys, scroll down the end of the "Customize hotkeys" window and click the plus icon next to "Editor hotkeys".

- If you still want to bind the hotkeys which are the same as the predefined hotkeys, except those of the "Editor hotkeys" group, first select and unbind the predefined hotkeys.

The following is the list of predefined hotkeys:

- Default global hotkeys

Table 14.1.

Hotkeys	Functions
Ctrl+F	Find/Replace
Ctrl+D	Delete line
Ctrl+L	Go to line
Ctrl+N	Create file from template
Ctrl+S	Save file

- Non-changeable hotkeys within editors

Table 14.2.

Hotkeys	Functions
Ctrl+Space	Autocomplete
Ctrl+B	Make text bold
Ctrl+I	Make text italic
Ctrl+U	Make text underlined
Ctrl+C	Copy selected text
Ctrl+V	Paste copied text
Ctrl+X	Cut selected text
Ctrl+Z	Undo typing
Ctrl+Y	Redo typing
Ctrl+A	Select all
Ctrl+Home	Go to the start of document

Hotkeys	Functions
Ctrl+End	Go to the end of document

Links

- *Codemirror usage* [http://codemirror.net]
- *FCKeditor user guide* [http://docs.cksource.com/FCKeditor_2.x/Users_Guide]
- *What are gadgets powered by Google* [http://www.google.com/webmasters/gadgets]
- *RESTful Web Services* [http://www.oracle.com/technetwork/articles/javase/index-137171.html]
- *Netvibes Documentation* [http://dev.netvibes.com/doc/]