IDE User Guide

Table of Contents

About eXo IDE

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 you to access and work with files online from anywhere. eXo IDE offers:

- Remote File System through the Virtual File System, including Browsing, File Locking, File Search and Versioning.
- Code Editor with syntax highlighting for many popular languages and advanced features, such as Code Auto-complete, Code Outline and WYSIWYG Editor for HTML.
- Multiple programming and descriptive languages support, such as JavaScript, HTML, XML, CSS, Groovy, and more.
- 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 that your credentials contain one of the listed roles by asking your system administrators. eXo IDE offers pretty the same functionality for them except that **developers** cannot deploy theirs REST services on a common environment and use a kind of "sandbox" to deploy them.

System requirements

Supported Web browsers:

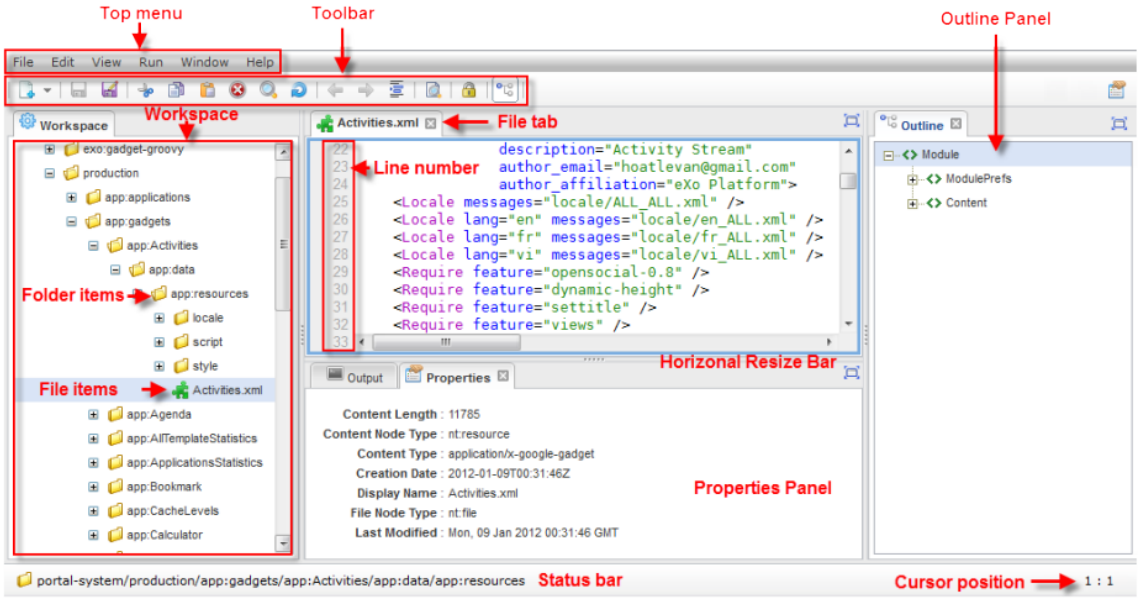
- Mozilla Firefox 3.6+
- Internet Explorer 7.0+
- Safari 5.0+
- Google Chrome 5.0+

Basic Operations

This chapter includes the following topics:

1.1. eXo IDE Interface

Before using eXo IDE, you should familiarize yourself with the application interface:



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** pane containing folders and files.

The **Content** pane consists of two horizontal sub-panes:

- The **Top** pane shows files in several file tabs with their content.
- The **Bottom Operation** pane may contain several tabs showing 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 show the file properties tab, click  on the toolbar or go to **View --> Properties**.
  - To show the file preview, go to **Run --> Show Preview** from the top menu.

These panes are divided by the **Horizontal Resize** bar. Also, both **Workspace** pane and **Content** pane are divided by the **Vertical Resize** bar and the **Vertical Resize** bar.

At the right column are the **Outline** pane and **Version** pane:

- The **Outline** pane enables you to access tags, variables, functions and other file elements quickly.
- The **Version** pane is to navigate to the file versions.

Tip

- The active pane is highlighted with a blue border.
- You can maximize or minimize one of the panes by clicking 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 via the toolbar or the top menu.

1.2. Select a workspace

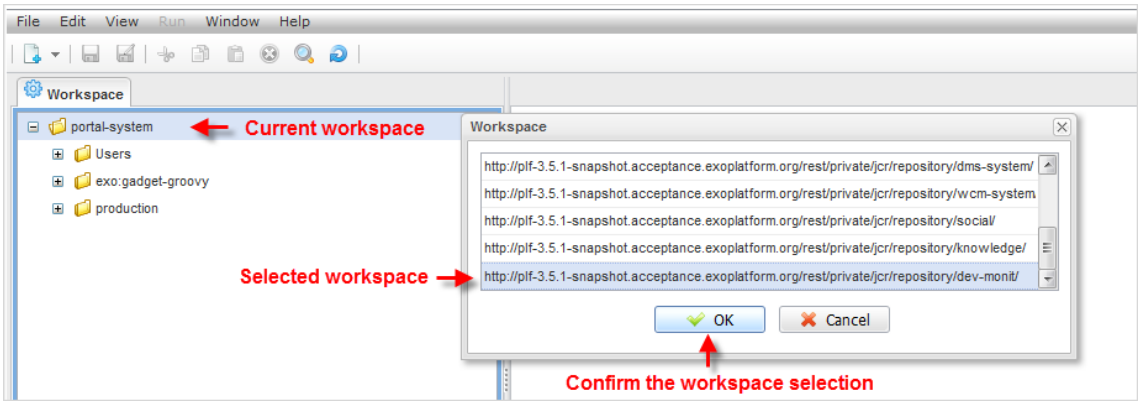
eXo IDE may provide access to one or more workspaces on the server. Each workspace has its unique URL.

Note

In case the system allows only one workspace, the **Select Workspace** option will be hidden.

Select a workspace

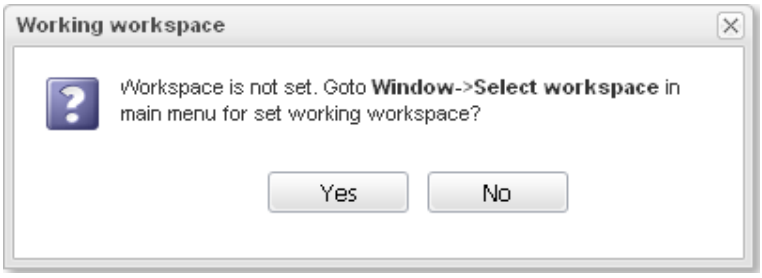
1. Go to **Window --> Select Workspace...** from the top menu.



2. Select your desired workspace and click **OK** to go to the appropriate workspace;

Or, simply double-click your desired workspace.

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



Click **Yes** to open the **Workspace** form.

1.3. Open a folder

1. Select your desired folder in the **Workspace** pane.

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.

1.4. Open a file

To open a file in the **Content** pane, double-click the file item in the **Workspace** pane.

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.

1.5. Refresh a selected folder

To refresh content of a selected folder in the **Workspace** pane, click  on the toolbar; or go to **File --> Refresh** from the top menu.

1.6. View and Copy a file's URL

1. Select the appropriate item in the **Workspace** pane.

2. Go to **View --> Get URL...** from the top menu to open the **Item URL** form.

3. Click **OK** to accept copying the target URL to the clipboard.

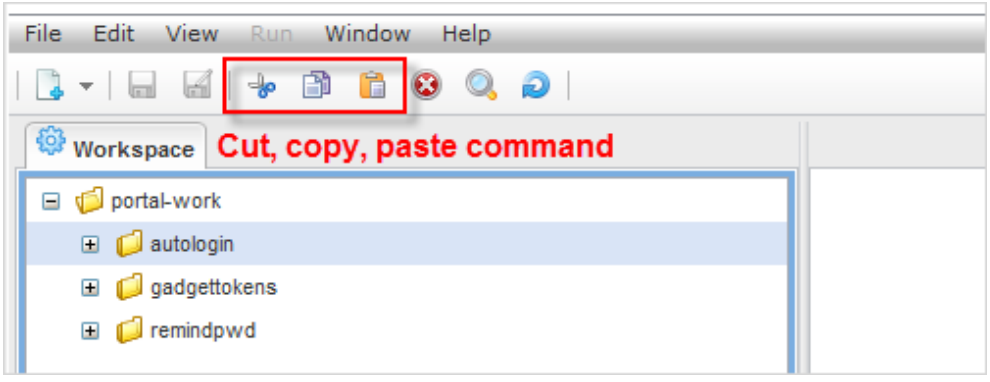
Note

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.

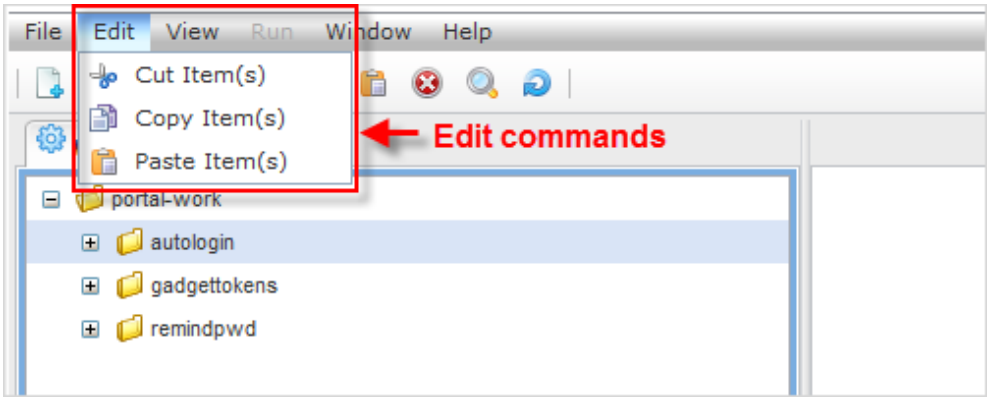
1.7. Copy/Cut/Paste files and folders

1. Click your desired files/folders in the **Workspace** pane.

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

In this chapter, you are going to learn how to work with folders in eXo IDE. The following topics will be covered:

- Create a folder
- Delete a folder
- Rename a folder
- Download a zipped folder

2.1. Create a folder

1. Click your desired target folder in the **Workspace** pane.

2. Click  on the toolbar, then select **Folder...** from the drop-down menu;

Or, go to **File --> New --> Folder...** from the top menu.

3. Enter the folder name in the **Create folder** form.

4. Click **Create** to complete creating your new folder.

Note

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

2.2. Delete a folder

1. Click your desired folder in the **Workspace** pane.

2. Click  on the toolbar;

Or, go to **File --> Delete...** from the top menu.

3. Click **Yes** to confirm your deletion in the **Delete Item(s)** form.

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

Note

You cannot remove:

- Workspace root folder.
- Multiple items from different hierarchical 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** pane.

2.3. Rename a folder

1. Click the folder you want to rename in the **Workspace** pane.

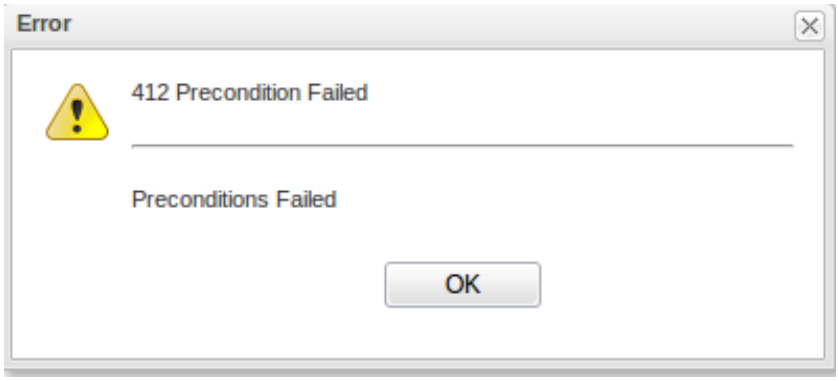
2. Go to **File --> Rename...** from the top menu to open the **Rename item** form.

3. Enter the new name into the **Rename item to** field.

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 already exists in the target folder, you will receive an error message as below.



Note

You cannot rename the workspace root folder.

Warning

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

2.4. Download a zipped folder

Download a folder from the JCR Repository

1. Click the target folder in the **Workspace** pane.

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 device.

Note

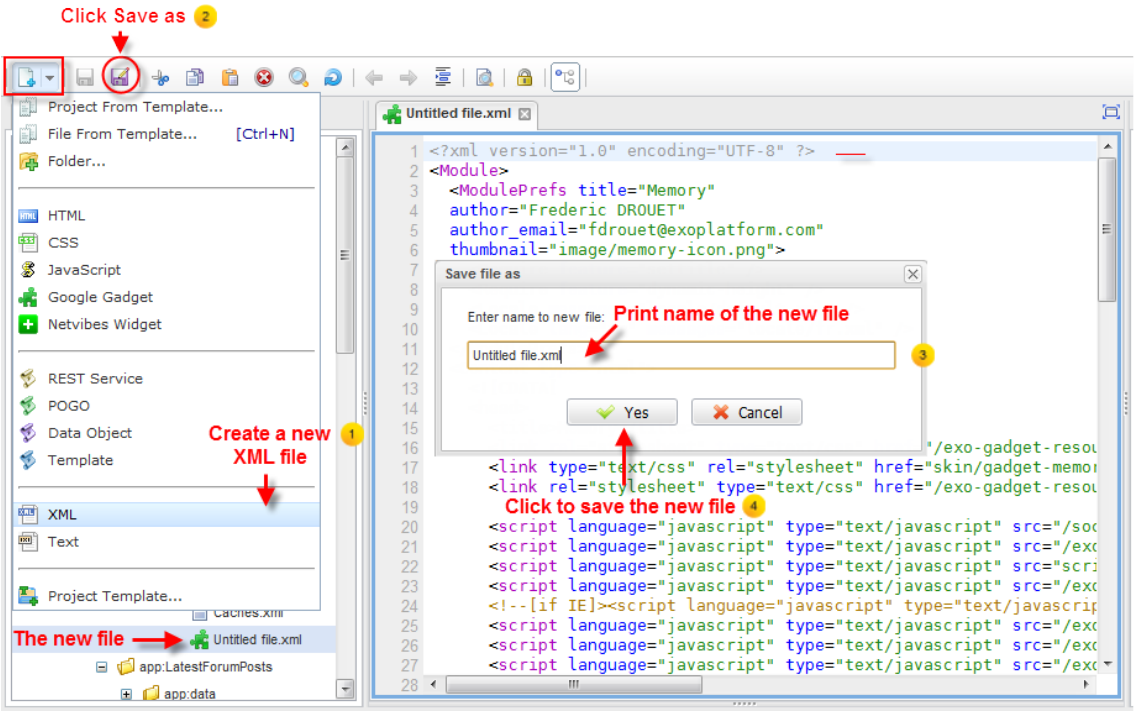
The archive uses UTF-8 encoding in entry names. Be sure that your zip extractor supports UTF-8 in file names. The Windows Explorer does not support it, you may use 7zip to unzip the archive. In Linux, the unzip (6.0 or below) command also does not support UTF-8 (depending on distributive). If you use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

Operations With File

In this chapter, you are going to learn how to work with files in eXo IDE. The following topics will be covered:

3.1. Create a file

1. Click  on the toolbar. Select a file type in drop-down menu.



2. Save the new file in some folders of the repository by selecting the target folder in the **Workspace** pane, then click on the toolbar;

Or, go to **File --> Save As...** from the top menu;

Or, press **Ctrl+S** keys.

3. Type the name of the new file in the **Save file as** form.

4. Click **Yes** to accept creating the new file.

3.2. Save/Save As... a file

Save a file into another folder or with another name

1. Select the target folder in the **Workspace** pane.

2. Click on the toolbar;

Or, go to **File --> Save As...** from the top menu;

Or, press **Ctrl+S** keys.

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

3.3. Save all files

If more than one file is created, opened and changed in the **Content** pane, 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.

3.4. Delete a file

1. Select the target file in the **Workspace** pane.

2. Click  on the toolbar;

Or, go to **File --> Delete...** from the top menu.

3. Confirm the deletion in the **Delete** form.

Files opened in the **Content** pane are closed after being removed.

Note

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

Tip

You can view the full path to the opened file when hovering your cursor over the file tab in the **Content** pane.

3.5. Rename a file

1. Select your desired file in the **Workspace** pane.

2. Go to **File --> Rename...** from the top menu.

3. Enter the new name into the **Rename item to** field.

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.

Warning

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

3.6. Change the MIME type of a file

1. Select the desired file in the **Workspace** pane.

2. Go to **File --> Rename...** from the top menu.

3. Select the new MIME type from **Select mime-type** combo-box;

Or, type another non-registered MIME type in this field.

4. Click **Rename** to change the MIME type.

Note

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

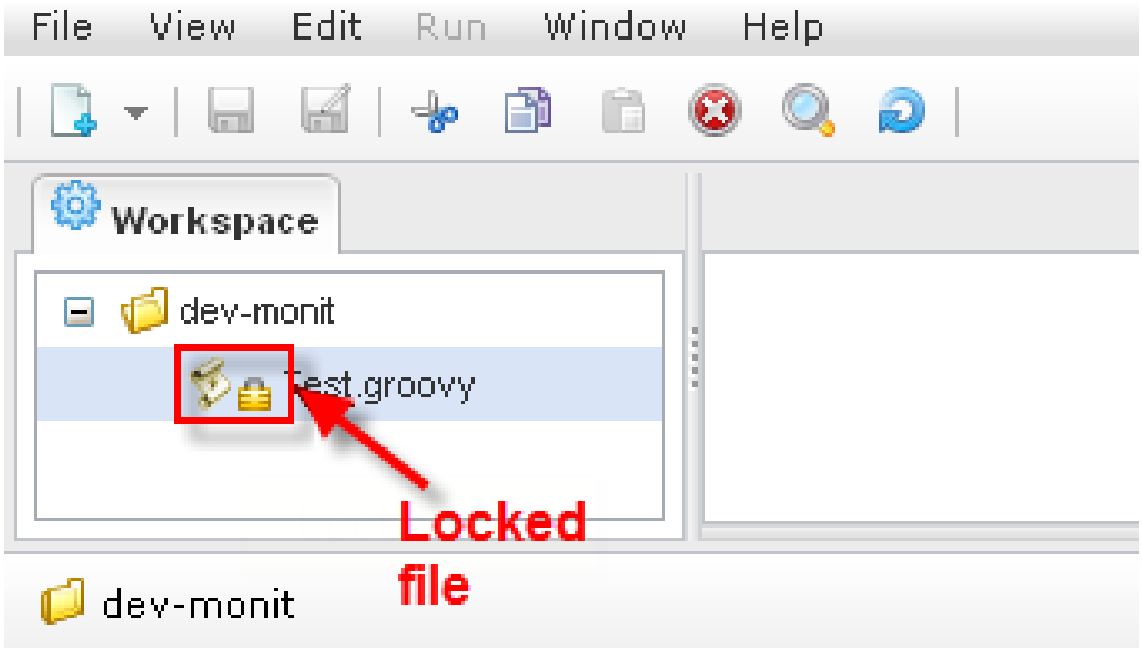
3.7. Lock a file

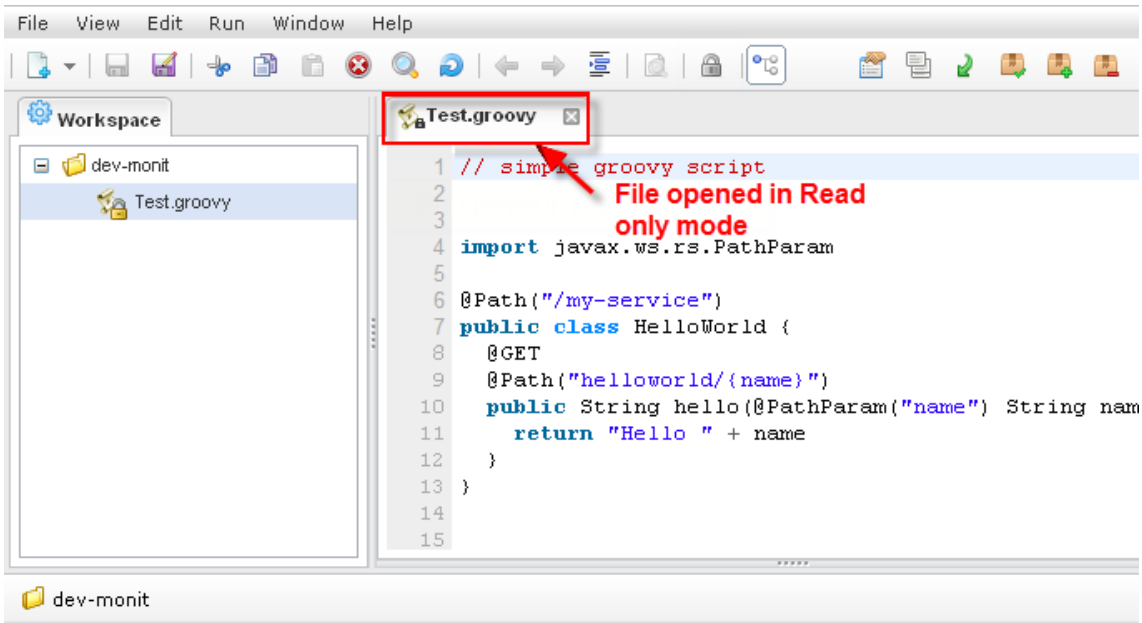
This feature is used to protect your files from being changed or removed by another users.

You can lock or unlock the file by clicking on the toolbar;

Or, going to **Edit --> Lock/Unlock File** from the top menu.

The icon is located next to the title of the locked file in the **Workspace** pane and at the title of this file tab in the **Content** pane.





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

Warning

If you open the locked file with the **WYSIWYG** editor, you can edit its content, but the **Save** function is disabled.

3.8. Open a file with the non-default editor

HTML and Google gadget files can be opened in either the default **Code** editor or **WYSIWYG** editor.

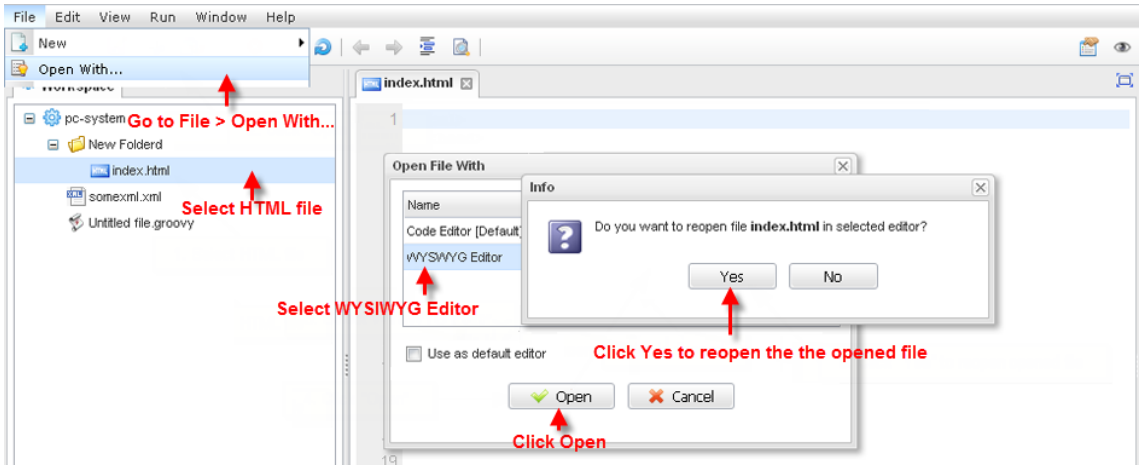
Open a file in the WYSIWYG Editor

1. Click the needed HTML or Google Gadget file in the **Workspace** pane.

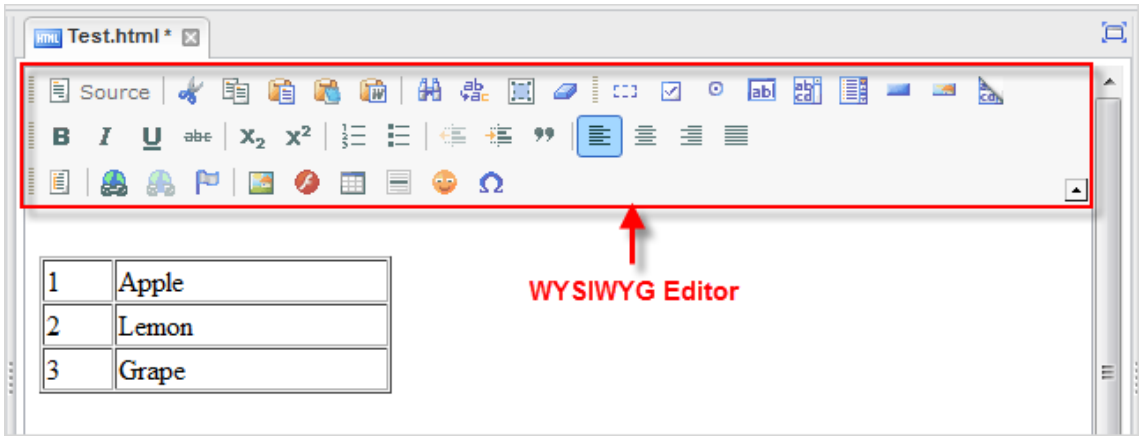
2. Go to **File --> Open With...** from the top menu.

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** pane before, the confirmation message will be shown.



Your selected file will be opened in the **WYSIWYG** editor as below.



If you want to change the default editor, simply select your desired editor and tick the **Use as default editor** checkbox in the **Open File With** form.

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

3.9. Go to a folder

This feature will help you quickly find the opened file in the **Workspace** pane.

1. Select the appropriate file tab in the **Content** pane.

2. Go to **View --> Go to Folder** from the top menu.

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

3.10. Download a file from Server

1. Select the target file in the **Workspace** pane.

2. Go to **File --> Download...** from the top menu.

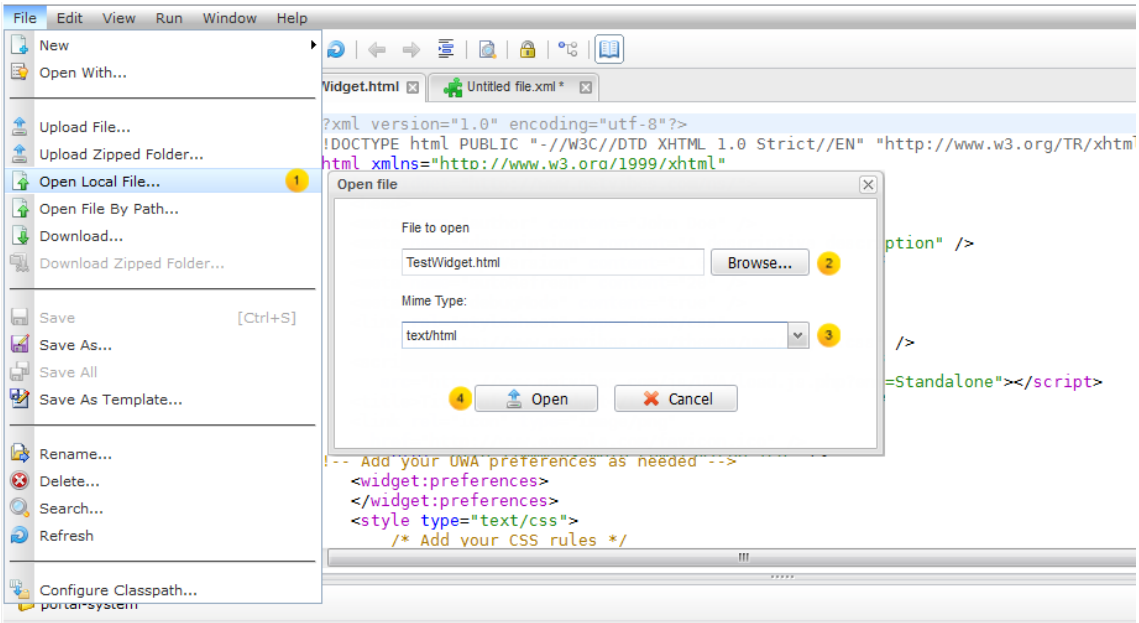
The file will be sent to the client and the suggestion window will be shown for you to save the file into your local device.

3.11. Open a 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.

Edit the local file content in eXo IDE

1. Go to **File --> Open Local File...** from the top menu.



2. Click **Browse...** in the **Open file** form to select the required local file.

3. Check the file's MIME type, based on the file extension. If needed, select another MIME type from the list.

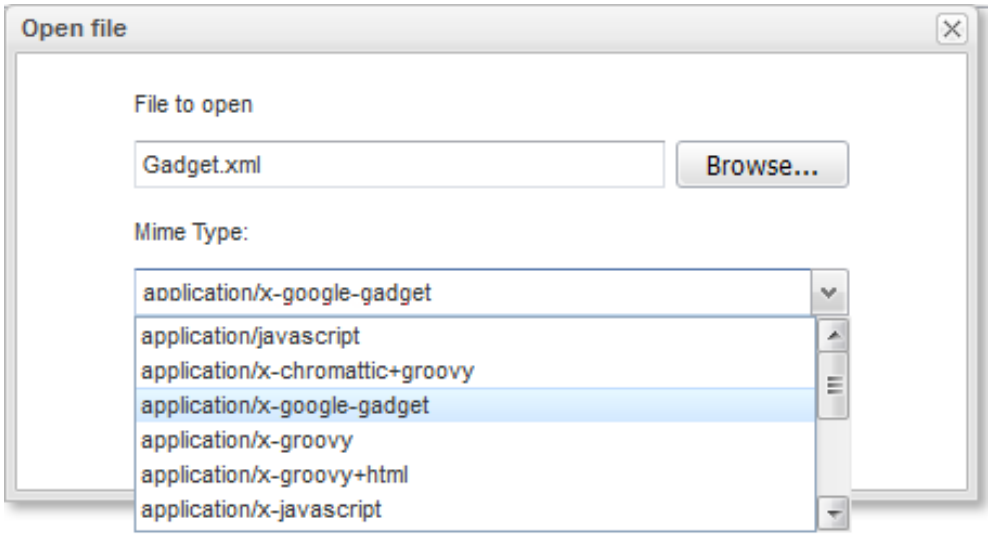
4. Press **Open** to open the selected file content in the default editor of the **Content** pane with the local file name.

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

File Type	MIME Type	Default File Extension	Default Editor
Text	text/plain	.txt	Code Editor
XML	text/xml; application/xml	.xml	Code Editor

File Type	MIME Type	Default File Extension	Default Editor
REST Service	application/x-jaxrs+groovy	.grs	Code 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		

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



3.12. Upload a file

You can upload a file from your local device to the **IDE Workspace**.

1. Select the **Gadget** folder in the **Workspace** pane.

2. Go to **File --> Upload File...** from the top menu.

3. Click **Browse...** to select your desired file from the local device.

4. Select the appropriate MIME type if needed.

5. Click **Upload**.

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

3.13. Upload a zipped folder

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

1. Select a parent folder in the **Workspace** pane.

2. Go to **File --> Upload Zipped Folder...** from the top menu.

3. Click **Browse...** in the **Upload folder** form to select a zipped folder from your local device.

4. Click **Upload**.

The zipped folder is uploaded to the server and automatically unzipped in the parent folder selected in Step 1.

3.14. Open a file by path

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

1. Go to **File --> Open File By Path...** from the top menu.

2. Enter the path of the target file into the **File URL** field.

3. Click **Open** to show the *Example.groovy* file in the new file tab of the **Content** pane.

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

Note

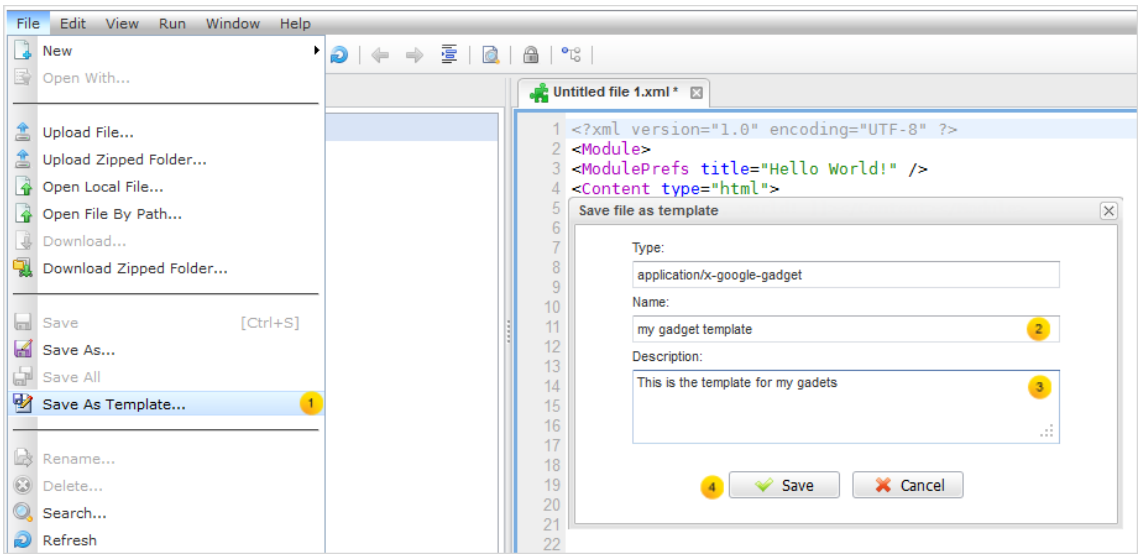
- The target file should be placed at the same domain as the IDE.

3.15. Use a 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.

3.15.1. Save a file as template

1. Go to **File --> Save As Template...** from the top menu.



2. Enter the template name.

3. Define the brief description of the template.

4. Click **Save** to accept.

You can also save the uploaded file as template.

> **Note**
>
> The template will have the same MIME type as the selected file.

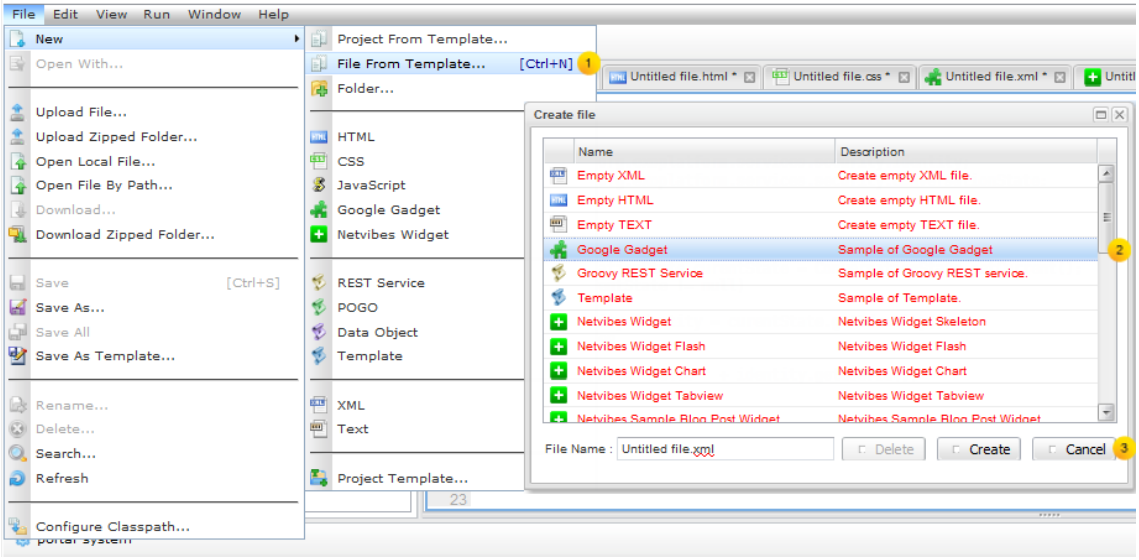
3.15.2. Create a file from a template

The **File From Template...** function enables you to create a file from the predefined template.

Create a simple gadget XML file

1. Click on the toolbar and select **File From Template...** from the pop-up to open the form;

Or, press **Ctrl+N** hotkeys.



2. Select your desired type and template for the file.

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** pane.

3.15.3. Remove a non-default file template

1. Click on the toolbar.

2. Select **File From Template...**;

Or, press **Ctrl+N** hotkeys.

3. Select the needed non-default template in black and click **Delete**.

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

Warning

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.

3.16. Use a project template

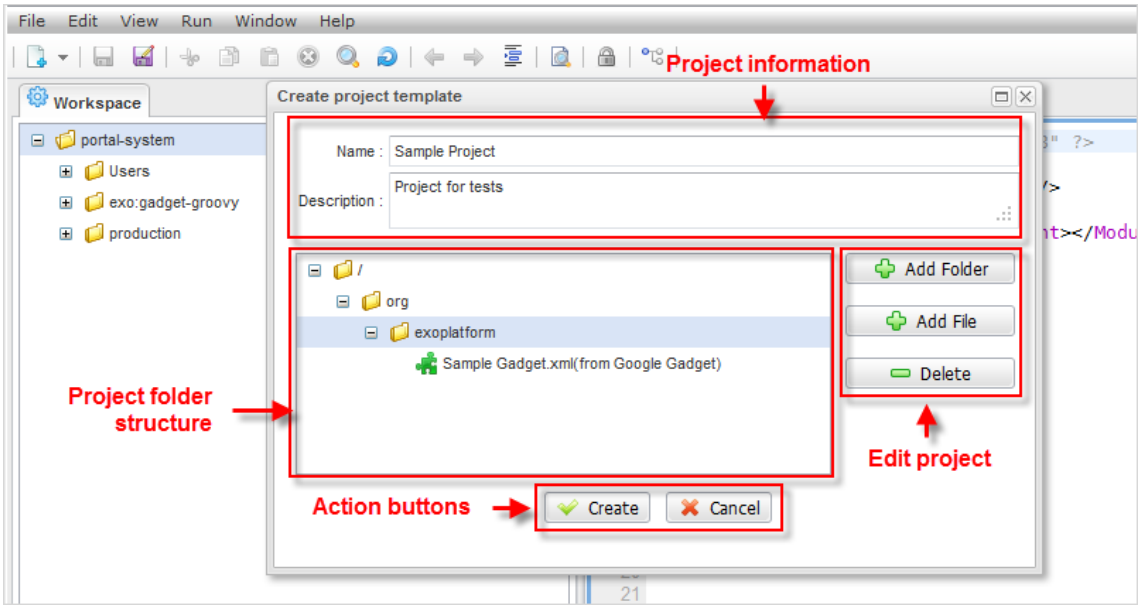
Project template helps you create folders and files with the predefined structure and content. You can create, update or remove your project template and add new folders and files from your template.

3.16.1. Create a project template

1. Go to **File --> New --> Project Template...** from the top menu;

Or, click on the toolbar and select the **Project Template...** command.

The **Create project template** form appears.



2. Enter the name and brief description in the **Create project template** form.

Optionally, you can restructure the project by clicking **Add Folder, Add File, Delete** at the right part of the **Create project template** form.

3. Click **Create** to complete your project template.

3.16.2. Modify the project template structure

You can make changes for your project template structure by adding new folders/files or deleting some elements.

Make modifications for your project template structure

1. Open the **Create project template** form as described in the **Create a project template** section above.

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

Now you can:

Add a folder

i. Click **Add Folder** to open the **Add folder** form.

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

Your newly added folder then will be added to the project template.

Add a new file

i. Click **Add File** to open the **Add file** form.

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

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

Delete some elements

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

ii. Click **Delete**.

3.16.3. Create new project from template

1. Select the parent folder in the **Workspace** pane.

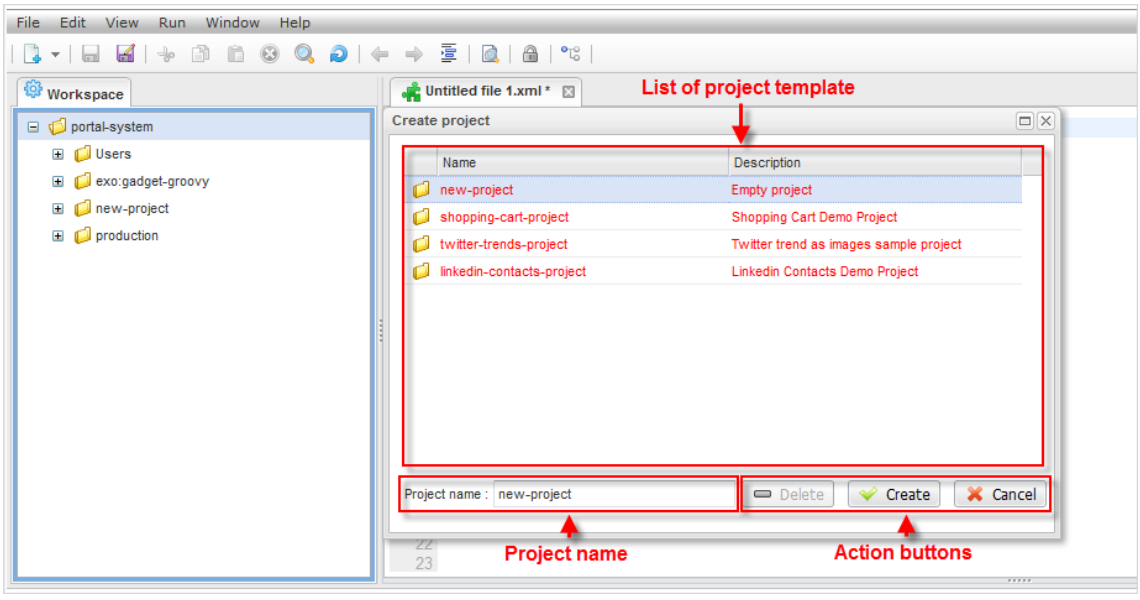
2. Click on the toolbar and then select **Project From Template...** from the drop-down menu;

Or, go to **File --> New --> Project From Template...** from the top menu.

3. Select your desired project template from the templates list.

4. Enter the project name into the **Project name** field.

5. Click **Create**.



3.16.4. Remove a project template

1. Click **New --> Project From Template...** on the toolbar;

Or, go to **File --> New --> Project From Template...** from the top menu.

2. Select your desired target project template from the list.

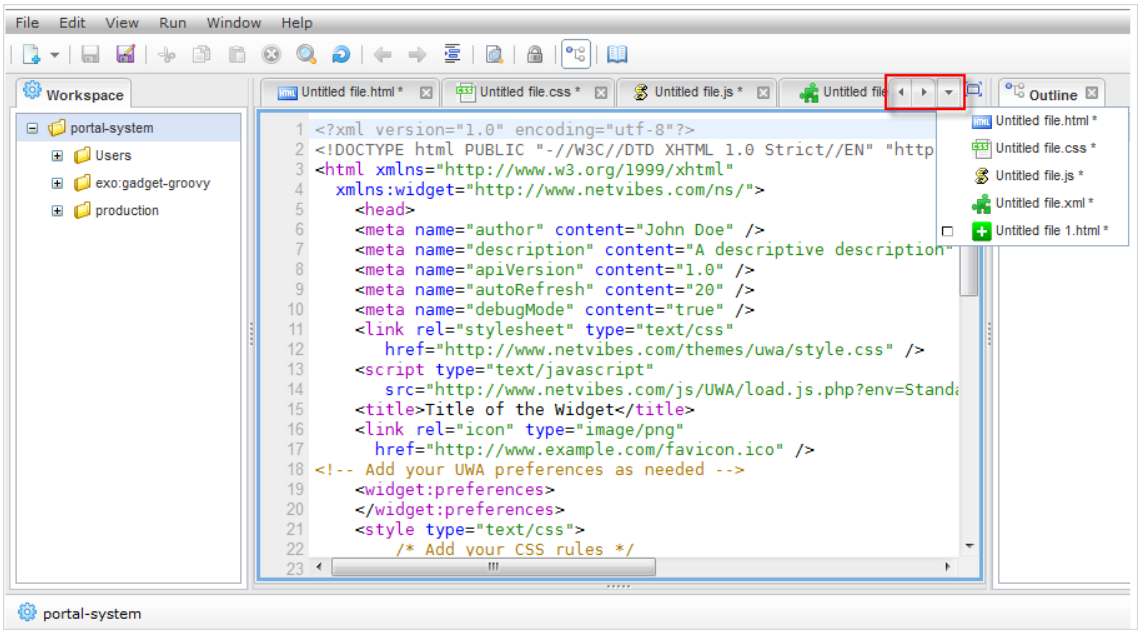
3. Click **Delete**.

3.17. 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 on the tab title.

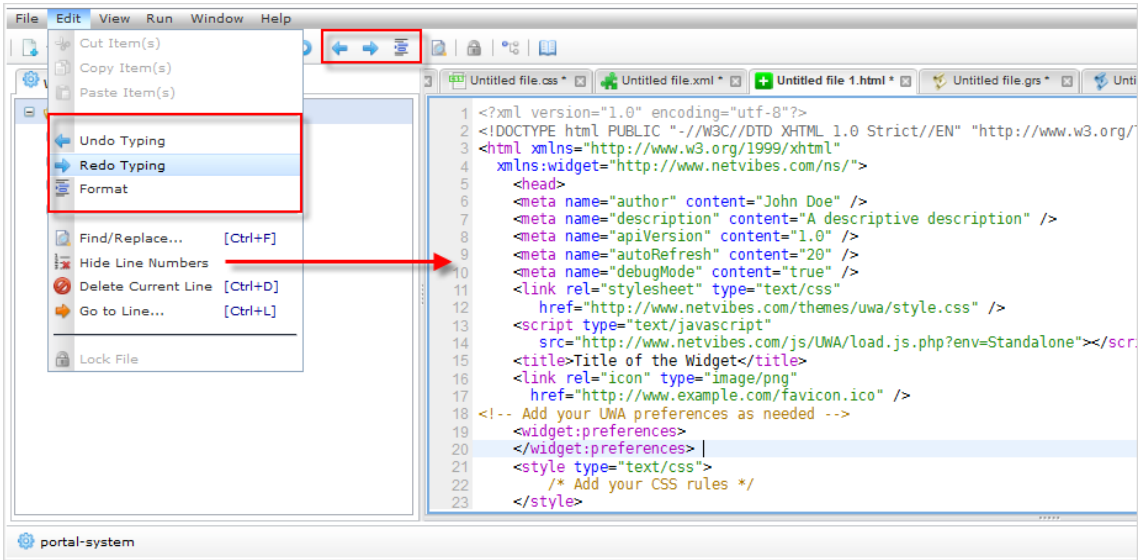
Click , , or at the top right corner of the **Content** pane to browse tabs.



Note

You can view the full path to the opened file when hovering your mouse over the tab title.

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



3.18. 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 editing

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  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 editing

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  on the toolbar.

Note

- **Redo** is available only after **Undo** was 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.

3.19. Edit files in the Code Editor

Once a file has been created/opened, its content is displayed in the separate tab on the **Content** pane 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.

3.19.1. Format a 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, click  on the toolbar, or go to **Edit --> Format** from the top menu.

3.19.2. Show/Hide Line Numbers

Show Line Numbers

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.

Hide Line Numbers

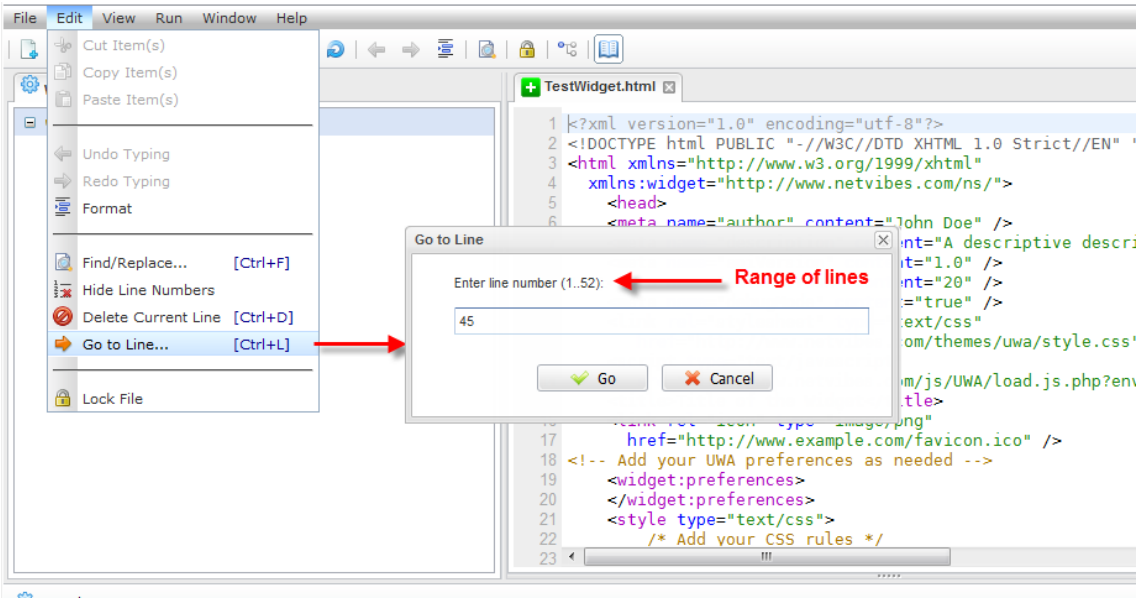
Go to **Edit --> Hide Line Numbers** from the top menu.

The last selected settings are saved in the browser cookies for next sessions.

3.19.3. Go to a line

1. Open the **Go to Line** form by following one of the listed below ways:

- Select **Edit --> Go to Line...** from the top menu.
- Double-click the area showing the current cursor position in the **Status** bar.
- Press **Ctrl+L**.



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

Note

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

3.19.4. 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.

3.19.5. Delete a line

1. Put the cursor on the desired line.

2. Use **Edit --> Delete Current Line** command from the top menu;

Or, press **Ctrl+D** to remove the highlighted line from the file.

3.19.6. 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:

1. Open your desired file.

2. Open the **Find/Replace** form by following one of the ways described below:

Select  on the toolbar;

Or, go to **Edit --> Find/Replace...** from the top menu;

Or, press **Ctrl+F**.

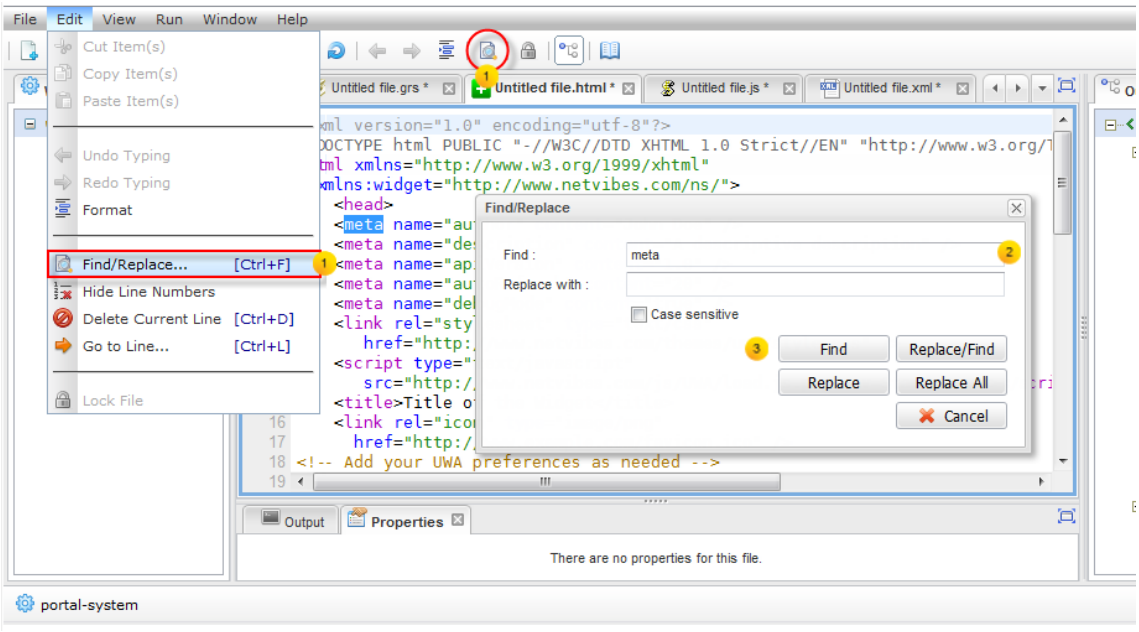
3. Enter the word/phrase for which you want to search into the **Find** field.

4. Click  to do the search.

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

5. Enter the replacement text into the **Replace with** field.

6. Click **Replace** or **Replace/Find** to replace the current selected found word/phrase, or select **Replace All** to replace all the matched phrase.

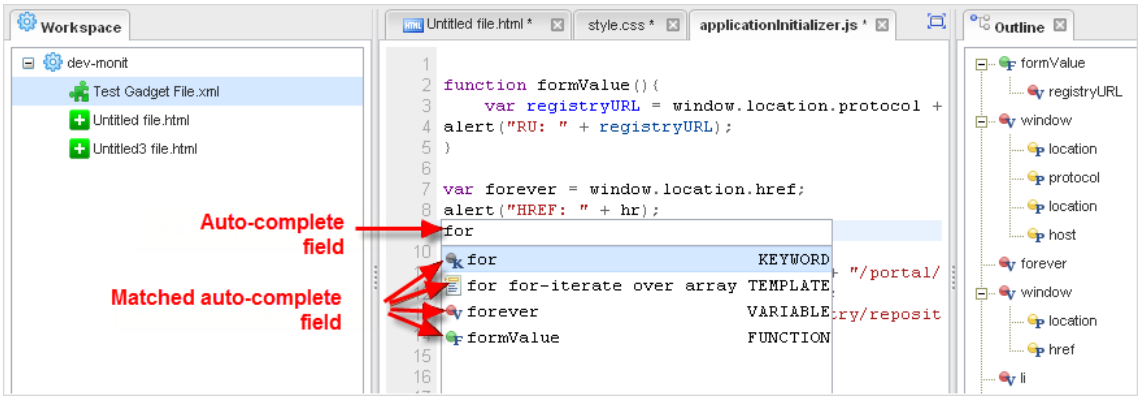


Note

- All functions in the **Find/Replace** form are only effective with the lines 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 form will save the status of performed actions of the opened file in each editor.

3.19.7. Code auto-complete

This feature enables you to auto-complete 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 **Auto-complete** form is shown as below:



In the files of Google Gadget, HTML and Template types, the auto-complete 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 auto-complete list for JavaScript content, or for CSS content respectively. In the other places, the auto-complete list for the HTML content is displayed.

Use the Auto-complete function

1. Open the target file with supported file types (JavaScript, CSS, HTML, Google Gadget, ECM Template and others) with the **Code** editor.

2. Put the cursor on the needed position in the document.

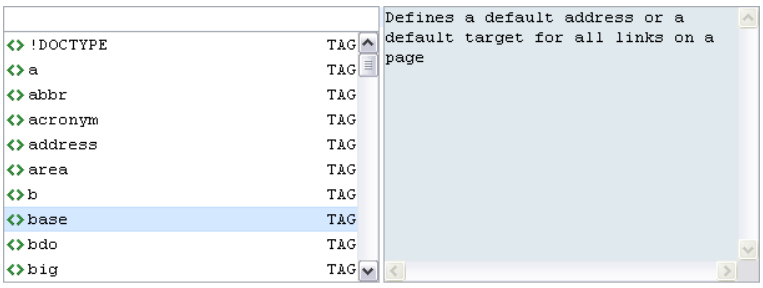
3. Press **Ctrl+Space** to get the **Auto-complete** form.

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.

5. Press **Enter** 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 **Auto-complete** form.
- Press **Esc** to discard the **Auto-complete** form.



3.19.7.1. JavaScript Specific auto-complete list

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

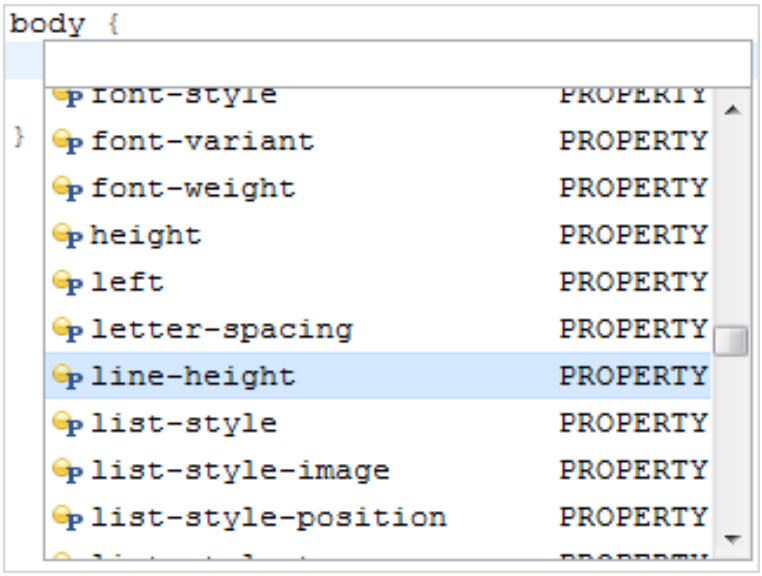
```
if (condition)
{

}
```

Template tokens have the special icon as shown in the **Auto-complete** form and tokens of different types are marked with different colors, including:

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

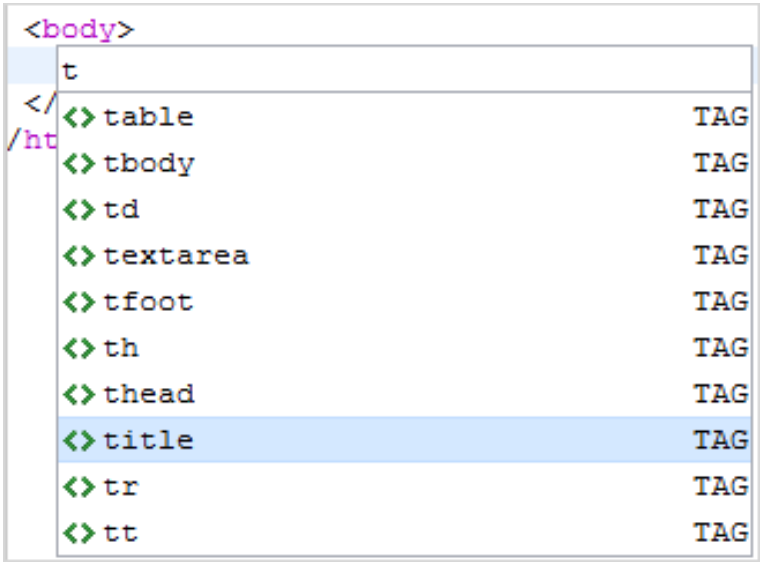
3.19.7.2. CSS specific auto-complete list

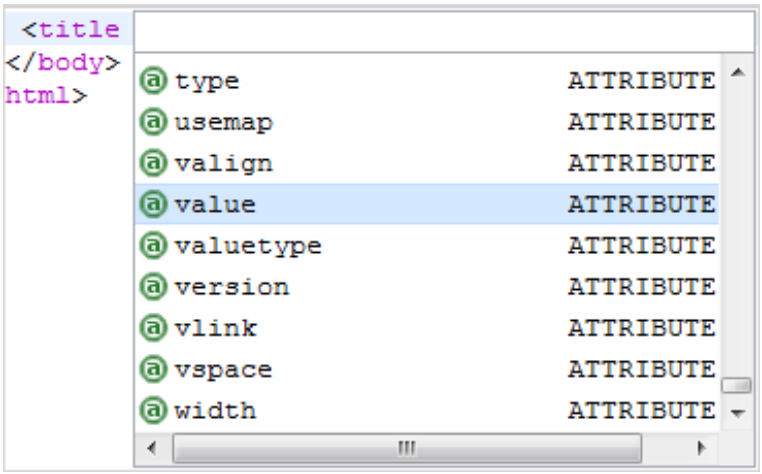


In the CSS file, the auto-complete list contains all the properties of Cascading Style Sheets. They are marked with the yellow token icons.

3.19.7.3. HTML specific auto-complete list

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





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

3.19.7.4. XML specific auto-complete list

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

3.19.7.5. ECM template auto-complete list

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

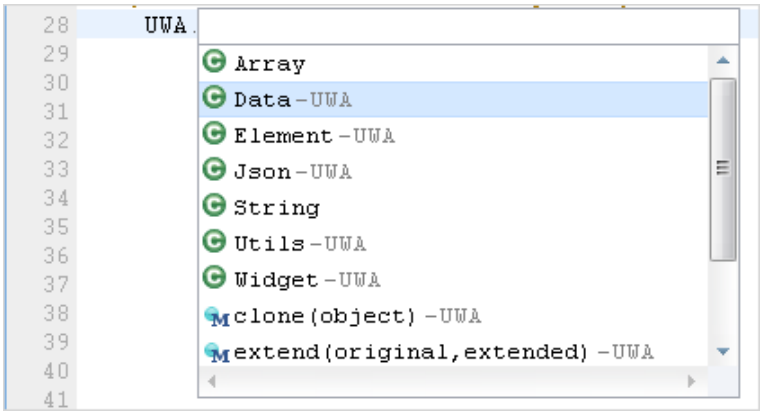
3.19.7.6. Netvibes widget auto-complete list

The auto-complete list of the Netvibes Widget files is the same as that of the HTML files.

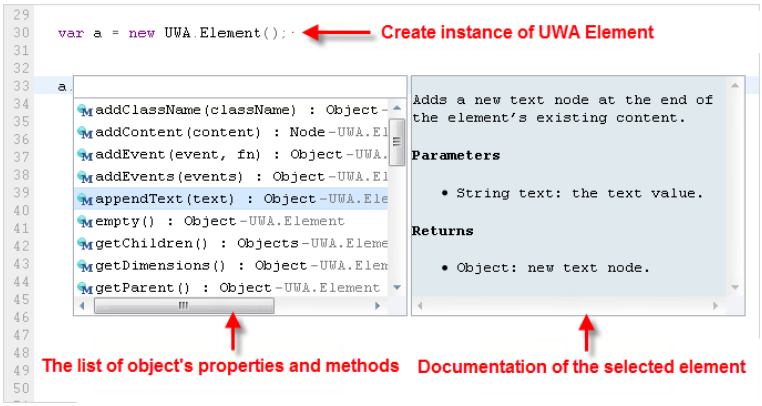
The auto-complete list inside JavaScript tags is the composition of the JavaScript auto-complete list and Netvibes snippets:

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

eXo IDE supports auto-complete for Netvibes Universal Widget.



It is possible to auto-complete the created Netvibes object's properties and methods.



You can get more details about this snippets and code that will be inserted via the hint window of the **Auto-complete** form.

3.19.7.7. Groovy code specific auto-complete list

eXo IDE currently supports special features for the Groovy code auto-complete list, including **REST Service** ("application/x-jaxrs+groovy"), **POJO** ("application/x-groovy") and **Data Object** ("application/x-chromattic+groovy"). The auto-complete 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 are described in the project's classpath, before generation of the auto-complete list. If the compilation of some class is failed, the error message will appear in the **Output** pane instead of **Auto-complete** 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 **Auto-complete** form, the files with such classes must fulfill the following conditions:

- Be located in folder with respect to package structure.
- Have the same file name as the class name located in a file.
- Have the *.groovy* or *.grs* extension.

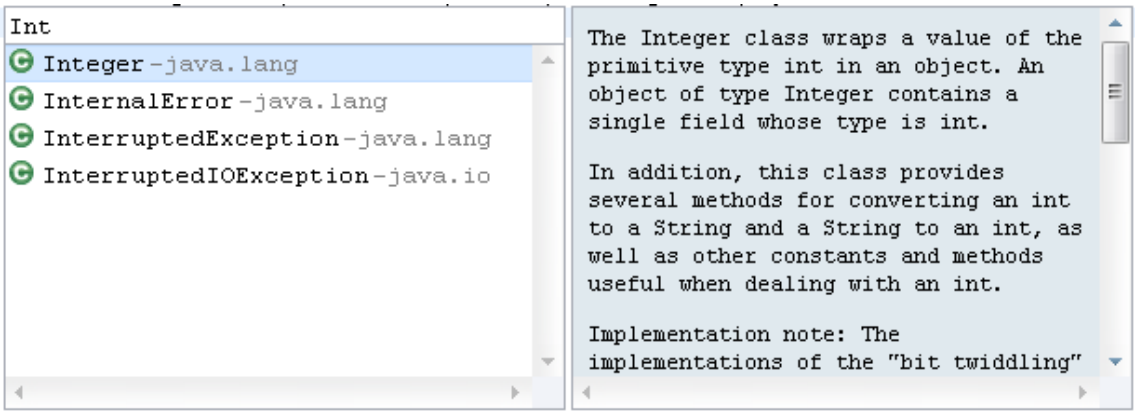
The Groovy code specific auto-complete list supports 5 cases as follows:

- Class name
- Object's fields and methods auto-completion
- Static fields and methods auto-completion
- Annotations auto-completion
- Local variables and parameters auto-completion

3.19.7.7.1. Class name

You can view class's JavaDoc and class name from the **Auto-complete** 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**



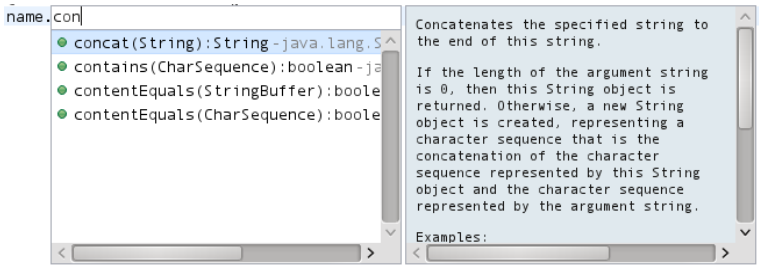
3.19.7.7.2. Object's fields and methods auto-completion

You can also view JavaDoc and auto-complete the object's fields and methods as follows:

1. Print the instance name with the dot (.) at the end.

2. Press **Ctrl+Space** to view all public methods and fields of the object. The JavaDoc for the selected field or method is shown.

3. Select the appropriate item in the **Auto-complete** form and press the **Enter** key; or double-click this item.



3.19.7.7.3. Static fields and methods auto-completion

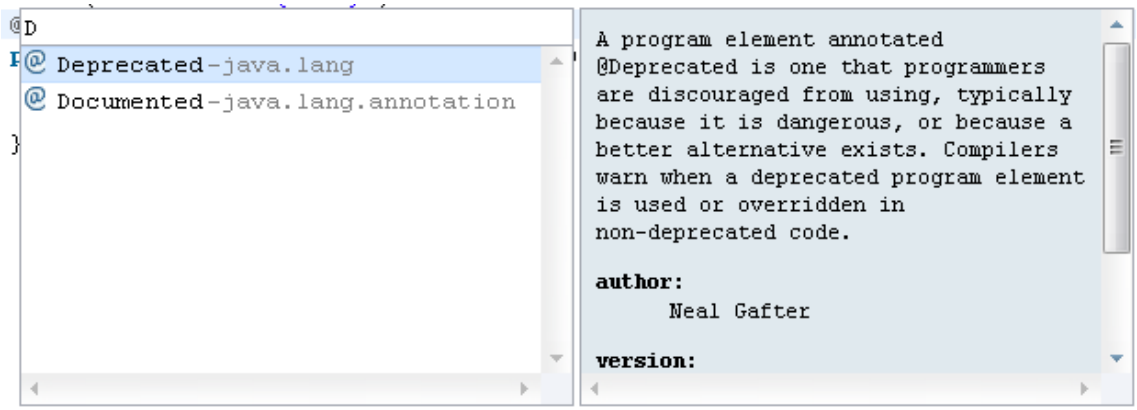
eXo IDE supports the auto-completion of static fields, methods and object's fields and methods.

Note

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

3.19.7.7.4. Annotations auto-completion

It is possible to auto-complete the annotations, followed by the symbol "@". The **Auto-complete** form for annotation contains the list of available annotations like this:



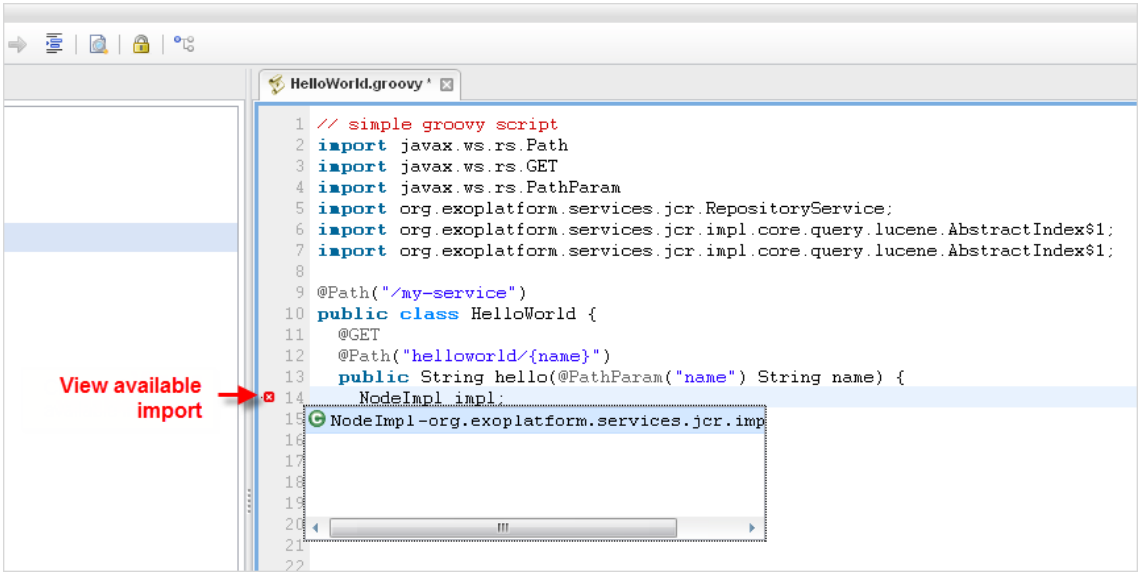
3.19.7.7.5. Local variables and parameters auto-completion

Local variables and method's parameters are marked with in the **Auto-complete** form. eXo IDE supports the auto-completion for local variables and method's parameters.

Besides, the keywords of Groovy code are not marked with icons in the auto-complete list.

3.19.7.8. Validate Java types and fix import statements

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



View and fix the Java types error

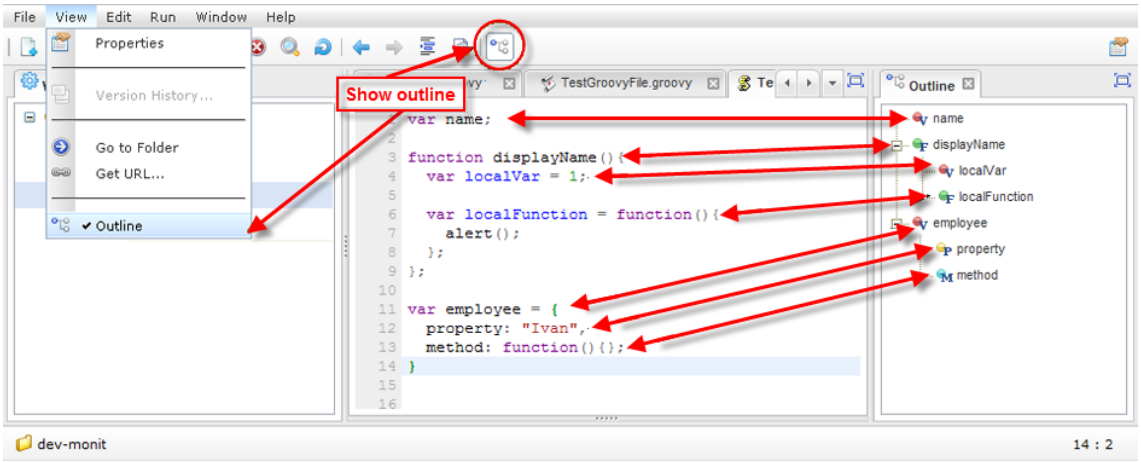
1. Hover the cursor over error marks to see the error message in tooltips.

2. Click the error icon in the **Line numbers** field.

3. Select the appropriate import statement and press the **Enter** key; or double-click to insert it into the header of the file.

3.19.8. Code outline

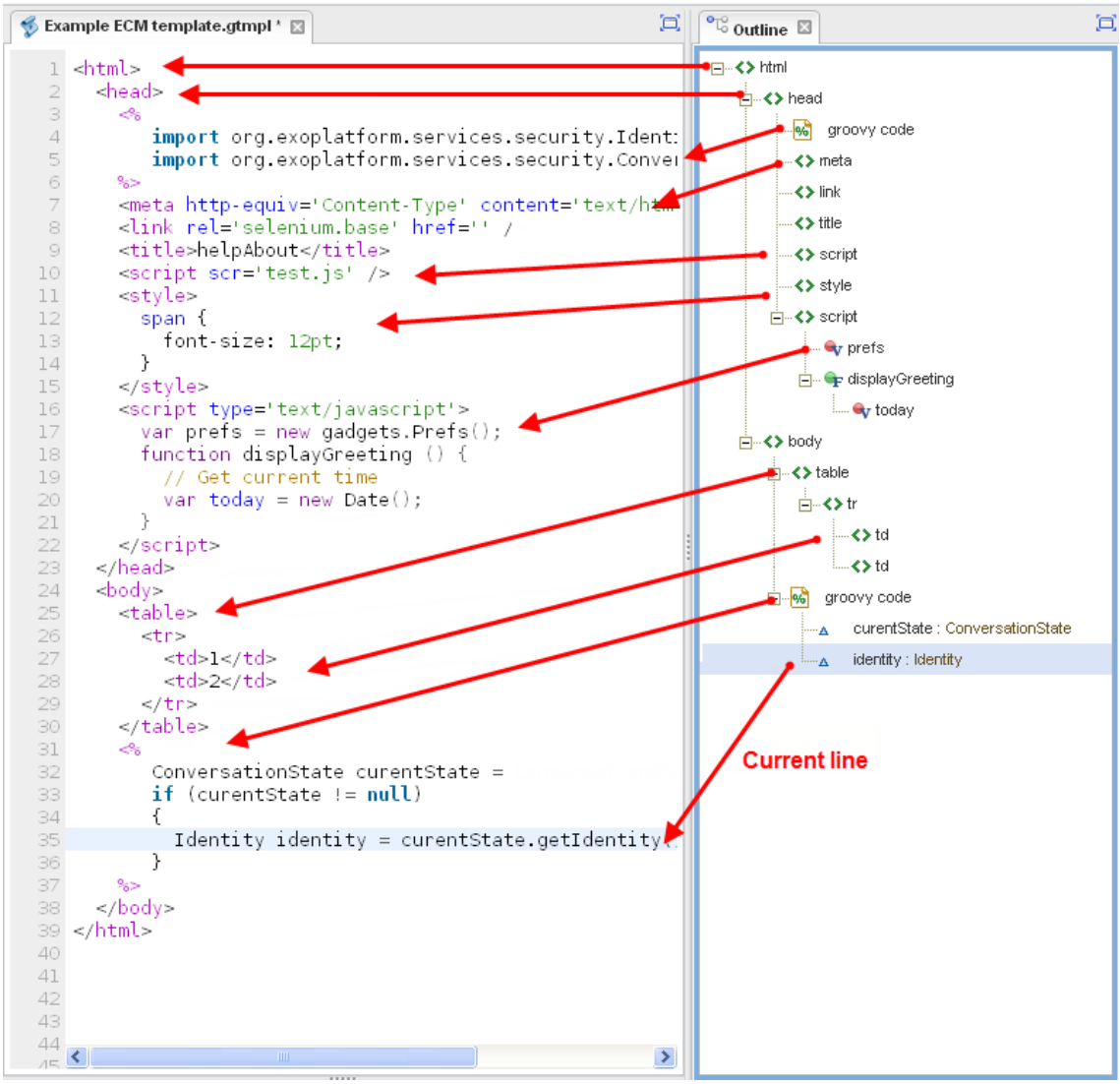
The code outline is used to create codes productively. The outline is available for all types of files, expect for Text or CSS files.

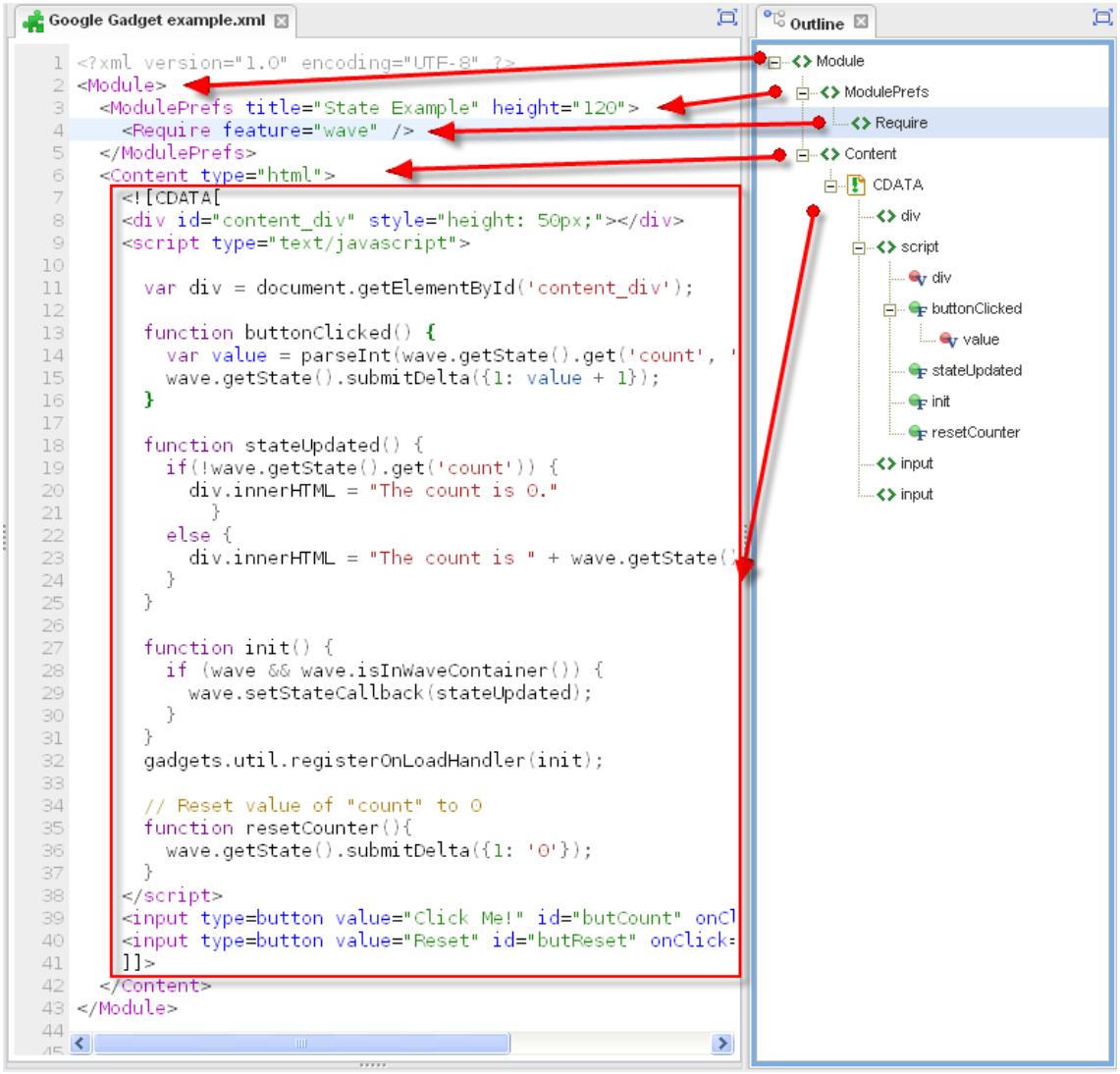


The **Outline** pane is opened automatically once a file has been created or opened. You can close the **Outline** pane by clicking 

on the tab header. Also, you can show/hide the **Outline** pane by pressing  on the toolbar or select **View** from the top menu and then select/deselect **Outline** from the drop-down menu.

In the **Outline** pane, 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 *
Outline
1 <html>
2 <head>
3 <%
4 import org.exoplatform.services.security.Ident
5 import org.exoplatform.services.security.Conve
6 %>
7 <meta http-equiv='Content-Type' content='text/h
8 <link rel='selenium.base' href='' />
9 <title>helpAbout</title>
10 <script scr='test.js' />
11 <style>
12 span {
13 font-size: 12pt;
14 }
15 </style>
16 <script type='text/javascript'>
17 var prefs = new gadgets.Prefs();
18 function displayGreeting () {
19 // Get current time
20 var today = new Date();
21 }
22 </script>
23 </head>
24 <body>
25 <table>
26 <tr>
27 <td>1</td>
28 <td>2</td>
29 </tr>
30 </table>
31 <%
32 ConversationState curentState =
33 if (curentState != null)
34 {
35 Identity identity = curentState.getIdentity(
36 }
37 %>
38 </body>
39 </html>
40
41
42
43
44
html
head
groovy code
meta
link
title
script
style
script
prefs
displayGreeting
today
body
table
tr
td
td
groovy code
curentState : ConversationState
identity : Identity
Current line



The outline tree is refreshed automatically every two seconds after you stop typing the code. When you click the item in the outline tree, the cursor will be set on the respective line where this item is defined. When you move the cursor into the file content, the respective node is selected in the outline tree.

Code outline for Groovy Code

In the **Outline** pane 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 table lists icons connected to different access modifiers.

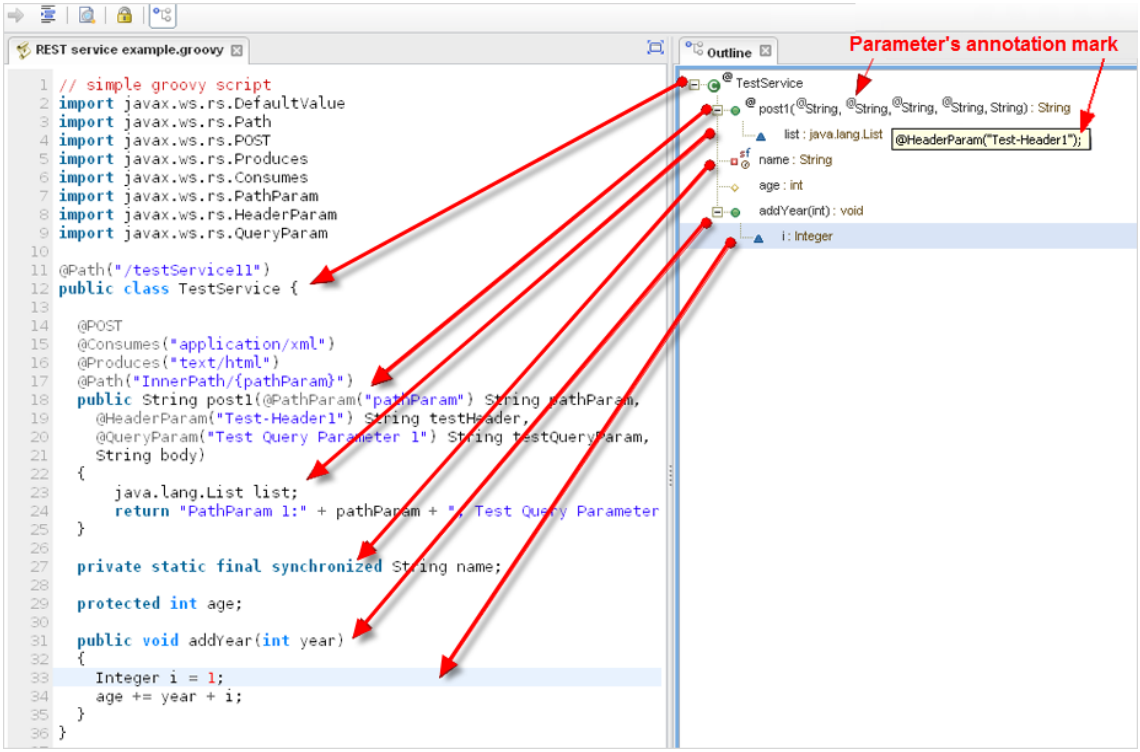
Access modification	Icon
private	Red square
public	Green circle
protected	Yellow rhombus
without access modifier	Blue triangle

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

Letter/Icon	Modifier
a	Abstract

Letter/Icon	Modifier
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 your cursor on "@".



3.20. Edit a 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, auto-complete, code outline or type error validation. However, the text edited with the **WYSIWYG** editor looks similar to 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 the **Open a file with the non-default editor** section above.

3.21. Preview file properties

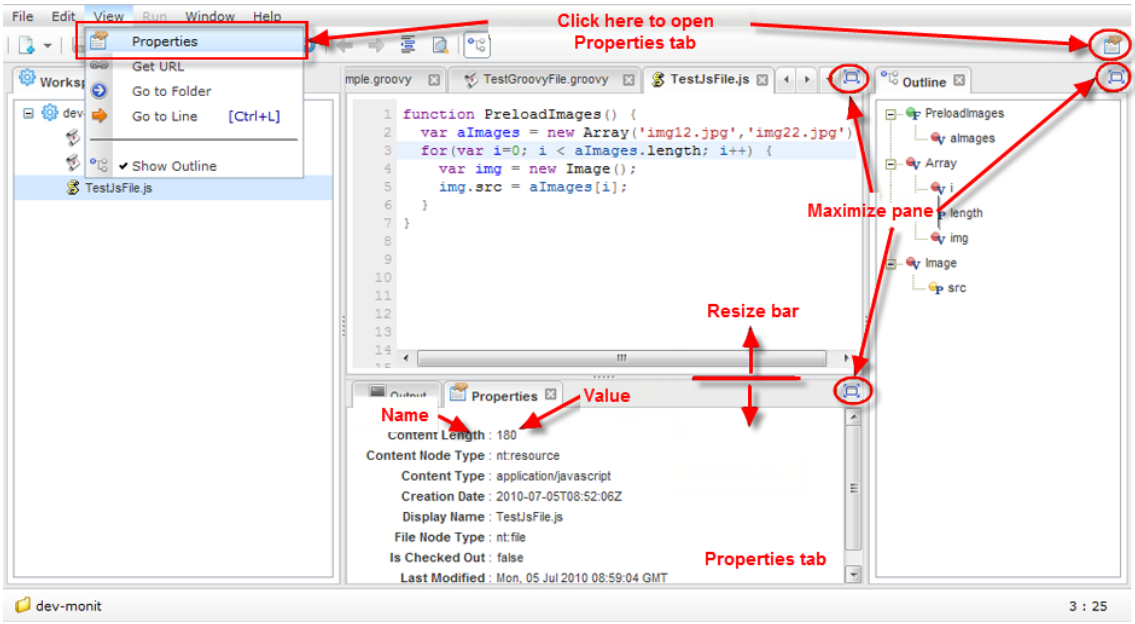
Connected with the JCR repository node, file properties are displayed in the **Properties** tab in the bottom of **Content** pane.

View file properties

1. Open the file in the **Content** pane.

2. Click  at the right corner of the toolbar;

Or, go to **View --> Properties** from the top menu.



Note

It is impossible to view the properties of files created but unsaved yet.

3.22. Preview HTML Files

To preview the HTML file stored in the JCR, simply click on the toolbar;

Or, go to **Run --> Show Preview** from the top menu after opening the HTML file in the **Content** pane. 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

1. Select the target folder or root folder in the **Workspace** pane.

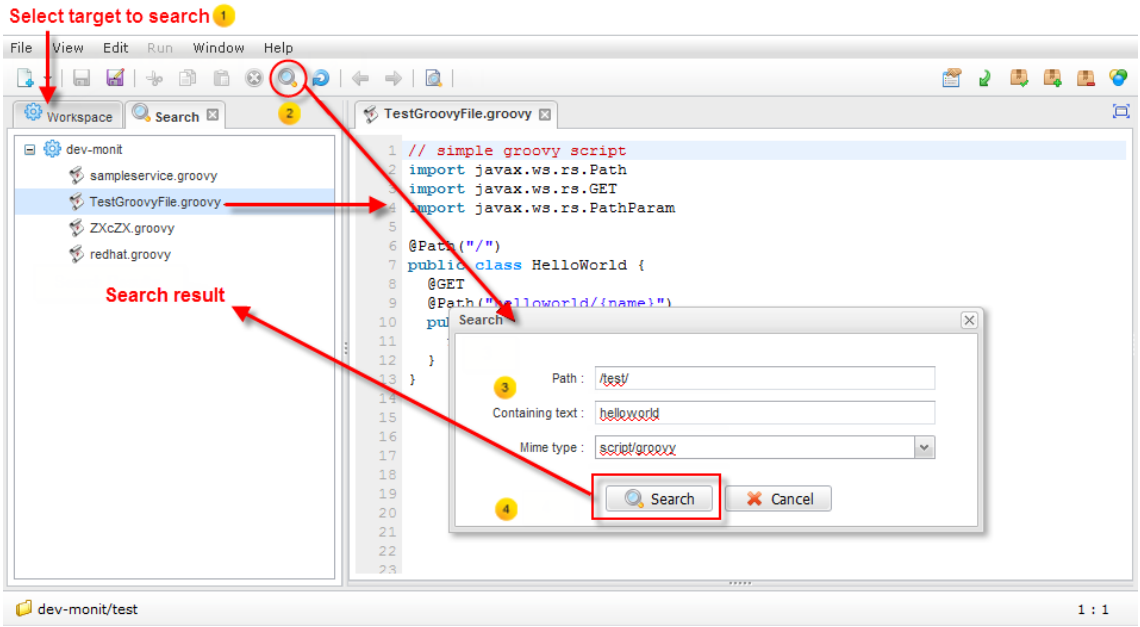
2. Click  on the toolbar;

Or, go to **File --> Search...** from the top menu to open the **Search** form.

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.

4. Click **Search**.

The found results are displayed in the **Search** pane:



Within the **Search** pane, 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** pane.

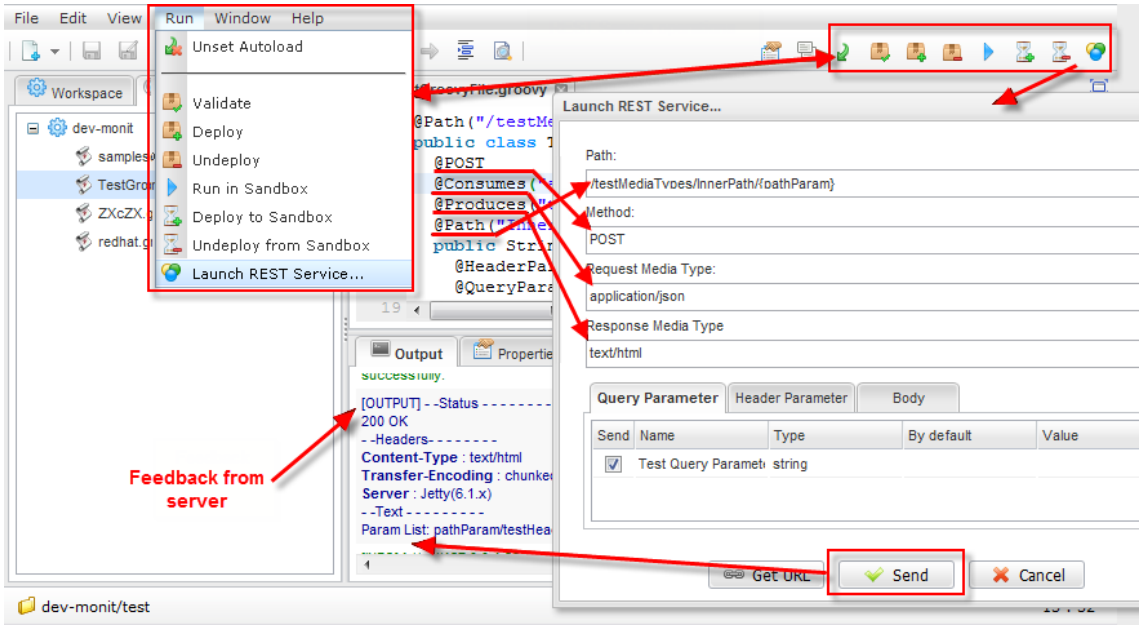
If you maximize the **Editor** pane, **Properties**, **Output**, **Outline** or other panes, the default perspective will be restored to show the **Search** tab when you receive search results.

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:

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:



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

To clear the **Output** pane, click  at the right top corner of this pane.

5.1. Validate a REST Service

You can verify the REST Service content by using the special validation service before saving.

To do that, simply click  on the toolbar;

Or, go to **Run --> Validate** from 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** pane to go to the line containing the 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** pane.

5.2. Deploy/Undeploy a REST Service

Note

This function is for users with the administrator role only.

Deploy a REST Service

1. Save the file before deploying.

2. Open the saved file in the **Content** pane.

3. Click  on the toolbar;

Or, go to **Run --> Deploy** from the top menu.

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
```

Warning

- 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, you have to undeploy the service with the registered URI pattern first.

- It is necessary to validate the Groovy scripts used by the service before deploying it by going to **Run --> Validate** from the top menu.

Undeploy a Rest Service

1. Select the service deployed in the **Workspace** pane.

2. Double-click the selected service to open it in the **Content** pane.

3. Click  on the toolbar.

In case of no errors, there will be a message informs undeploying successfully 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
```

Warning

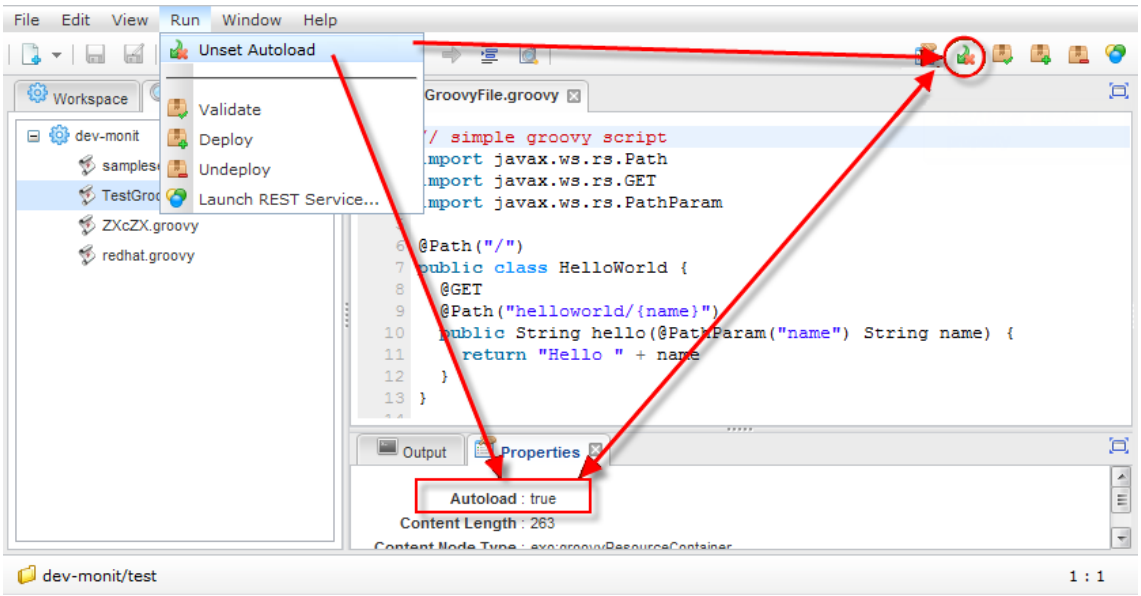
Deleting the service will not make it undeploy. You have to undeploy the service before deleting it. In other cases, the workaround is to create it again in the same place and then undeploy.

5.3. Use Autoload property

Note

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  . 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:



5.4. Launch a REST Service with set parameters and Preview Server Console Messages in the Output Tab

With eXo IDE, you cannot 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 click  on the toolbar,

Or, go to **Run --> Launch REST Service...** from the top menu as follows:

1. Select the deployed REST Service.

2. Click  on the toolbar to get the **REST Service** form which is relied on the WADL-description of REST Service.

3. Select and type the path to the service in the **Path** field. Path parameters are enclosed in curly braces.

4. Select one of the supported methods from the **Method** combo-box.

5. Select the appropriate **Request Media Type**.

6. Check the **Response Media Type**.

7. Uncheck the redundant query parameters and set values of the rest in the table at the bottom of the dialog window.

8. Go to the **Header Parameters** middle tab and set appropriate values.

9. Set request with the body content within the **Body** tab. This tab is disabled for the GET request.

10. Click **Send**.

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.

The following table shows annotations of reproducing service in the **Launch REST Service** form:

REST Service Annotation	Element of Launch REST Service form
@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.

5.5. Get URL of a REST Service

From the **Launch REST Service** form, you can also get the URL of the REST Service as follows:

1. Select the deployed REST Service.

2. Click  on the toolbar to get the **Launch REST Service** form.

3. Click **Get URL** to view the REST Service URL.

5.6. Review the response headers from REST Service

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

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

5.7. Deploy to Sandbox, Undeploy from Sandbox, Run in Sandbox

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

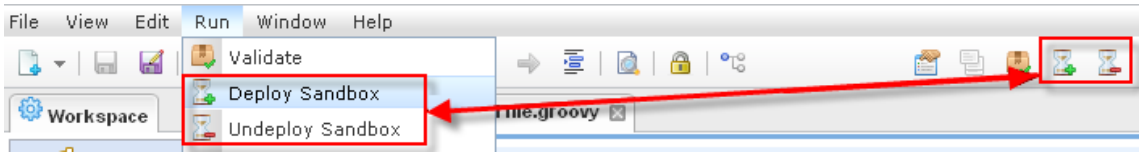
Deploy to Sandbox

In eXo IDE, developers can secure deploy untested REST Service for testing in an isolated environment named Sandbox. Do this by following either of two ways below:

- Click  on the toolbar;
- Or, go to **Run --> Deploy to Sandbox** from the top menu.

Undeploy from Sandbox

- Click  on the toolbar;
- Or, go to **Run --> Undeploy from Sandbox** from the top menu.



> **Note**
>
> You cannot deploy REST Services that were previously deployed by other users.

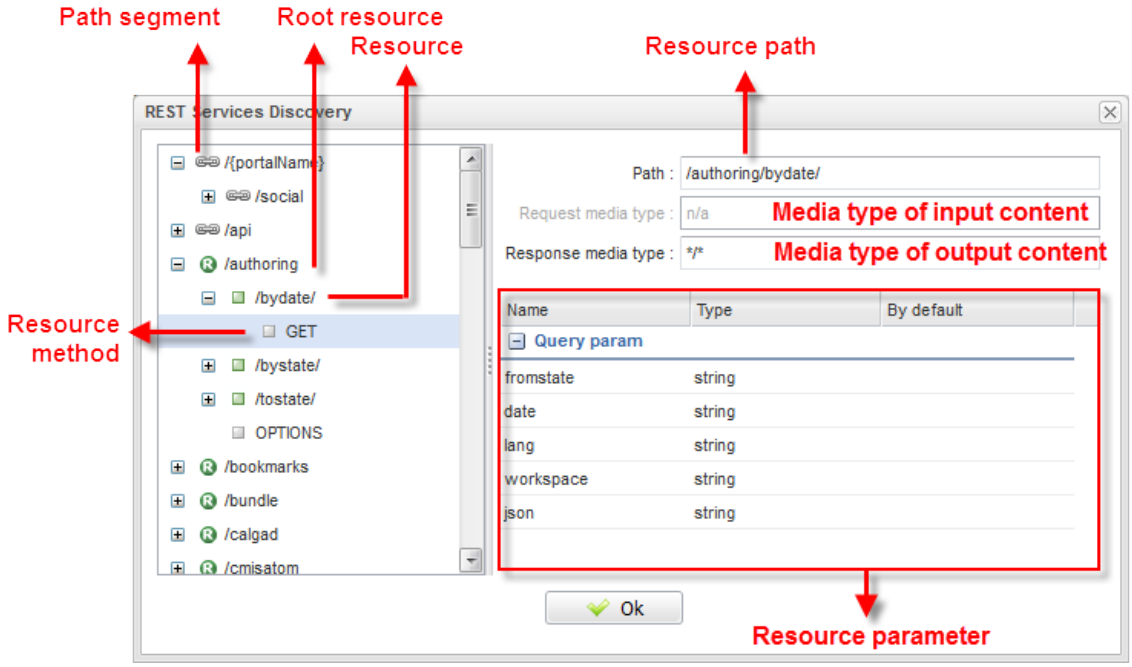
Run in Sandbox

To run the 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**. Do this by following either of two ways below:

- Click  at the right corner of the toolbar;
- Or, go to **Run --> Run in Sandbox** from the top menu.

5.8. REST Service Discovery

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



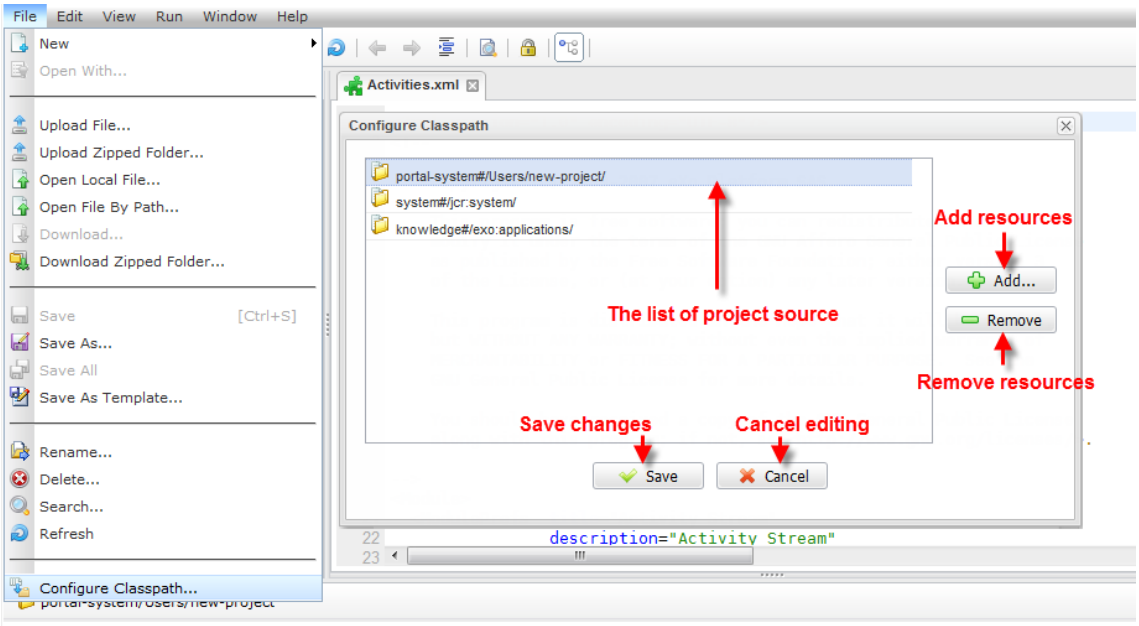
5.9. Configure classpath

The classpath 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 is not visible in the browser tree because it cannot be directly edited by user.

View the sources paths of the project

1. Select the created project item in the **Workspace** pane.

2. Go to **File --> Configure Classpath...** from the top menu to open the **Configure Classpath** form as below:



3. Add or remove source in the tree.

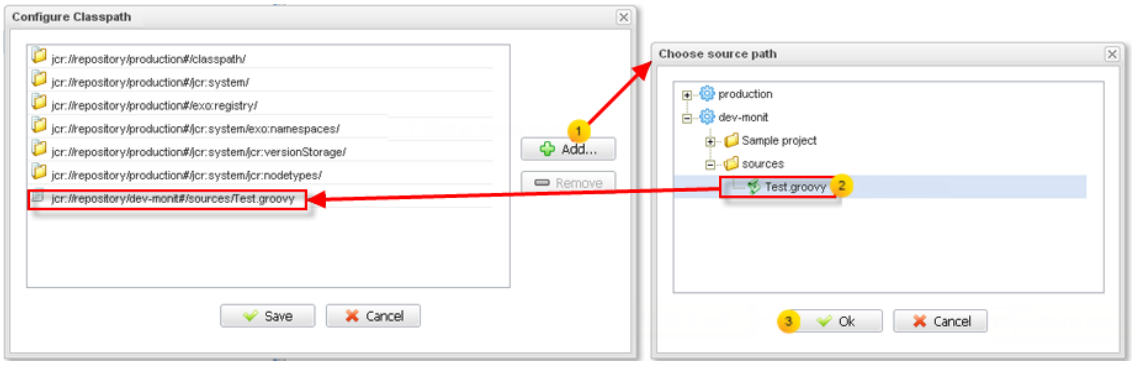
4. Click **Save** save changes in the classpath configuration, or **Cancel** to discard changes.

Note

If the selected item in the **Workspace** pane is not a project or a part of it, you will get an error message.

5.9.1. Add sources to classpath

1. Click **Add...** in the **Configure Classpath** form to open the **Choose source path** form.



2. Select one or more sources in the tree.

3. Click **OK**.

Tip

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

Note

The **OK** button is disabled if the workspace root folder is selected in the **Choose source path** form.

5.9.2. Remove sources from classpath

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

Note

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

5.10. Sample REST Service with complex media types output preview

1. Create, save and deploy the REST Service with the 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;
  }

}
```

2. Select **Run --> Launch REST Service...** from the top menu, or click  on the toolbar.

3. Select the **Path** field as "/testMediaTypes", method OPTIONS, then click **Send**.

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

```
<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" name="pathParam" style="template" type="xs:string"/>
        <method id="post1" name="POST">
          <request>
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="Test-Header1" style="header" type="xs:string"/>
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="Test Query Parameter 1" style="query" type="xs:string"/>
            <representation mediaType="application/xml"/>
          </request>
          <response>
            <representation mediaType="text/html"/>
          </response>
        </method>
        <method id="post2" name="POST">
          <request>
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="Test-Header2" style="header" type="xs:string"/>
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="Test Query Parameter 2" style="query" type="xs:string"/>
            <representation mediaType="application/xml"/>
          </request>
          <response>
            <representation mediaType="application/json"/>
          </response>
        </method>
      </resource>
    </resource>
  </resources>
</application>
```

4. Open the **Launch REST Service...** form again and select the another **Path** field:

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

5. Select the **Response Media Type =** "text/html" item.

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".

7. Click **Send**.

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:

- Create a Google Gadget
- Preview a Google Gadget

7.1. Create a Google Gadget

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

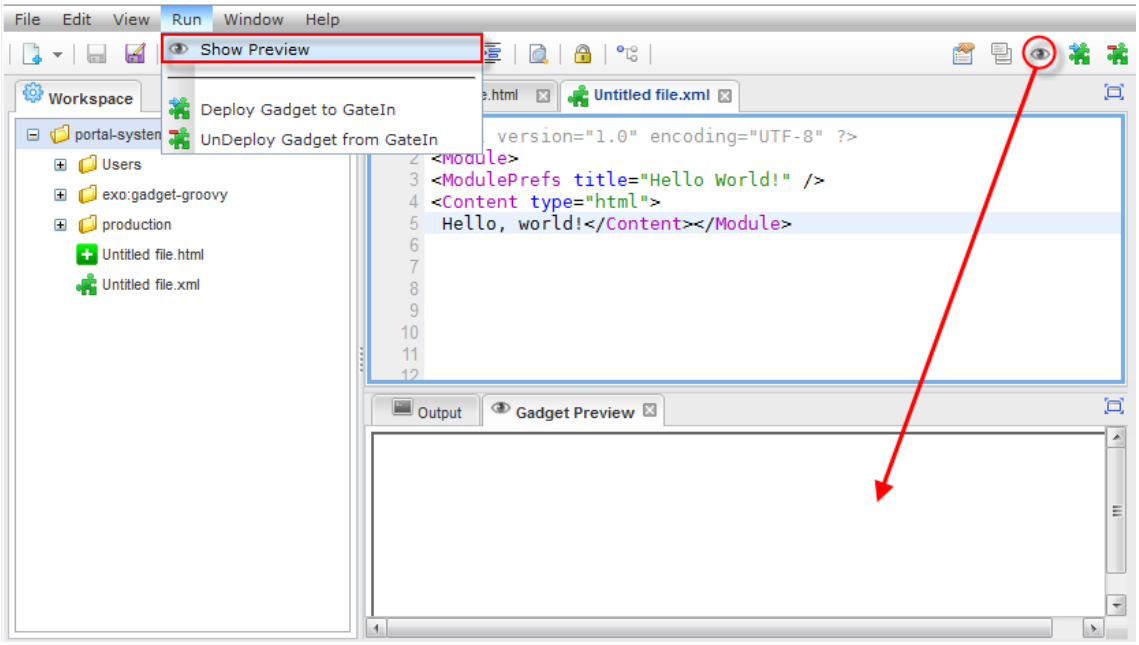
The first way: Click on the toolbar and then select **Google Gadget** from the drop-down menu.

The second way: Go to **File --> New --> Google Gadget** from the top menu.

The third way: Open the existing gadget from the local system by going to **File --> Open Local File** from the top menu.

The forth way: Upload the existing gadget directly to JCR by going to **File --> Upload File...** from the top menu.

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



7.2. Preview a Google Gadget

1. Open your desired Google Gadget file which was previously saved.

2. Click on the toolbar;

Or, go to **Run --> Show Preview** from the top menu.

The gadget is opened in the **Gadget Container** of the **Preview** tab as described in the **Google Gadget Operations** illustration 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.

The following topics are covered:

- Create and Edit a template
- View template's properties
- Preview a template

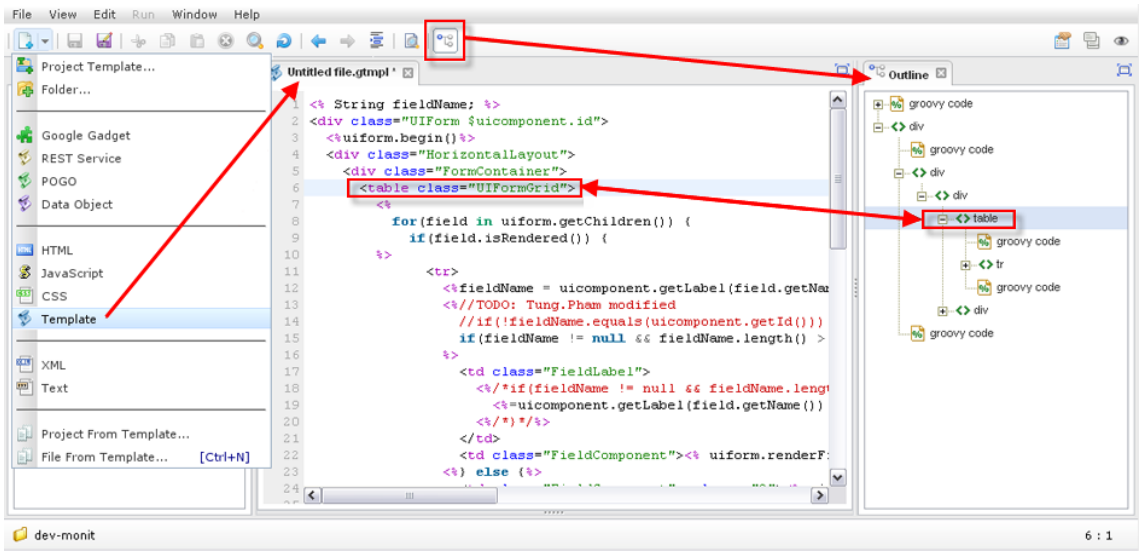
8.1. Create and Edit a template

1. Click on the toolbar and then select **Template** from the drop-down menu; or select **File --> New --> Template** from the top menu. A new file is opened in the editor.

2. Add content.

3. Save the file by selecting **File --> Save As** from the top menu, or by clicking on the toolbar.

Both **Outline** pane and **Auto-complete** form can be used to edit the Groovy Template.



8.2. View template's properties

To view properties of the template, click on the right corner of the toolbar.

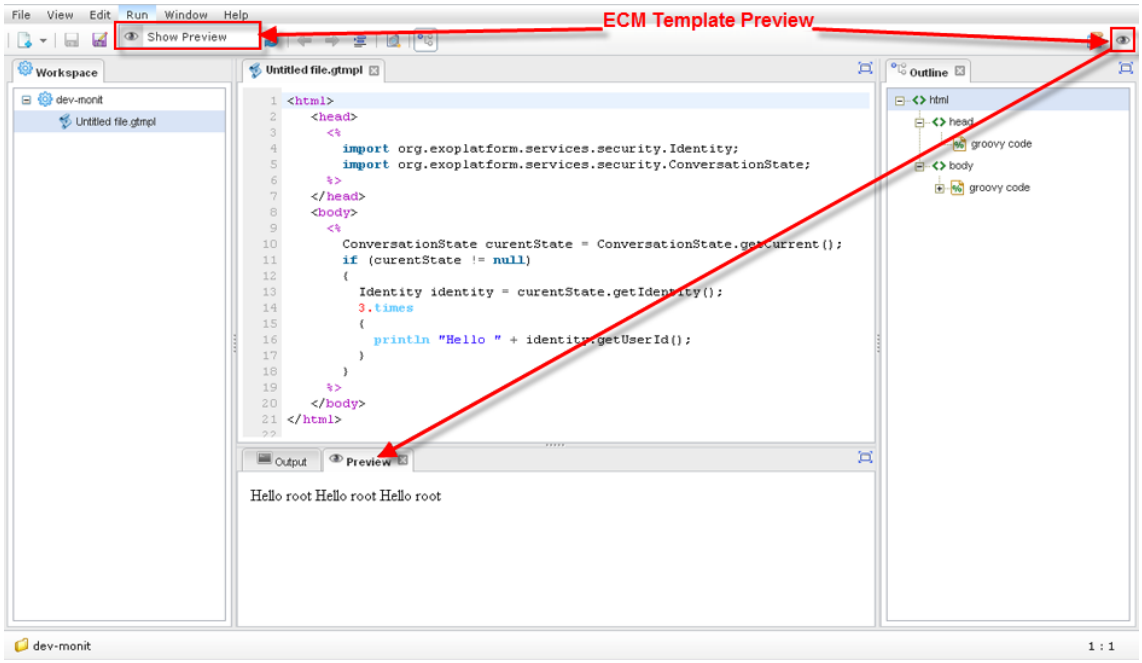
8.3. Preview a template

1. Open your desired Groovy template in the **Content** pane.

2. Click on the toolbar;

Or, go to **Run --> Show Preview** from the top menu.

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



Operations With Netvibes Widget

This chapter includes the following topics:

- Create Netvibes widget
- Preview Netvibes widget
- Deploy Netvibes widget to Netvibes Ecosystem
- View Netvibes documentation

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 the Netvibes Ecosystem.

Note

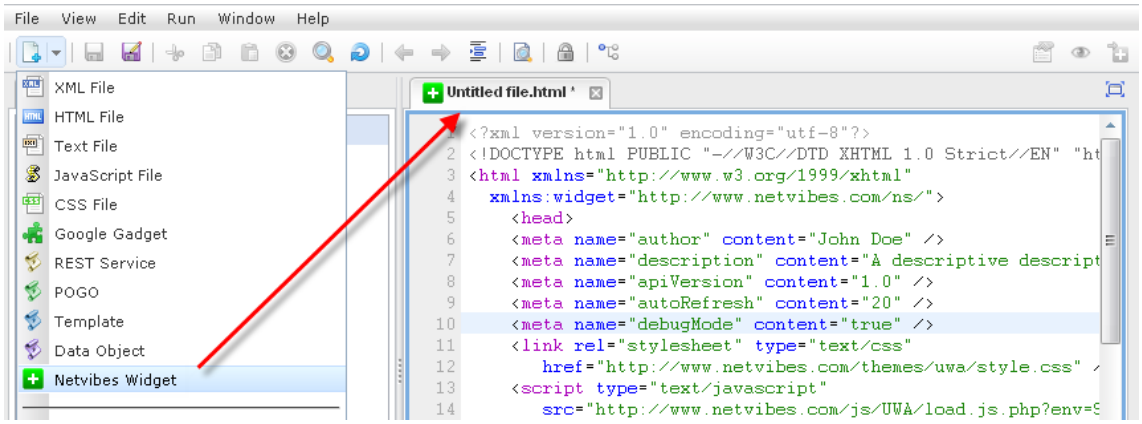
If you want to get acquainted with writing your own widgets, refer to this page.

9.1. Create Netvibes widget

Click on the toolbar and then select **Netvibes Widget** from the drop-down menu;

Or, go to **File --> New --> Netvibes Widget** from the top menu.

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



9.2. Preview Netvibes widget

1. Open your desired widget file in the **Content** pane.

2. Click on the toolbar;

Or, go to **Run --> Show Preview** from the top menu.

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

9.3. Deploy Netvibes widget to Netvibes Ecosystem

Netvibes widgets are placed in the remote system called Netvibes Ecosystem. 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 deployment mechanism. Manual deployment is easier because you do not need to get an API key from Netvibes.

Manual deployment

1. Get the URL of your Netvibes widget by selecting your widget, then clicking **View --> Get URL**. Copy the URL.

2. Go to the Netvibes Ecosystem, click **Create or submit** and **Submit Widget**.

3. Paste the URL of your gadget, and don not forget to remove the **/private** part from it. If you do not do this, Netvibes will not be able to access it.

4. Fill in the form and finish the process. The widget is now available in the ecosystem.

Note

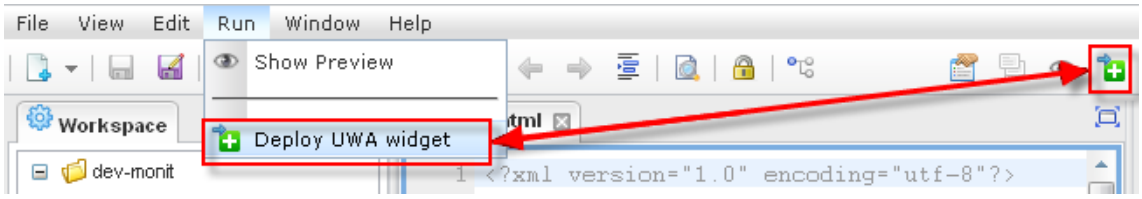
For deployment using our wizard, you need to contact Netvibes at `<business@netvibes.com>` and ask for your API key.

Deploy UWA widget:

1. Get the URL of your netvibes widget. To do this, select your widget and click **View --> Get URL** from the menu. Copy the URL.

2. Click  on the toolbar;

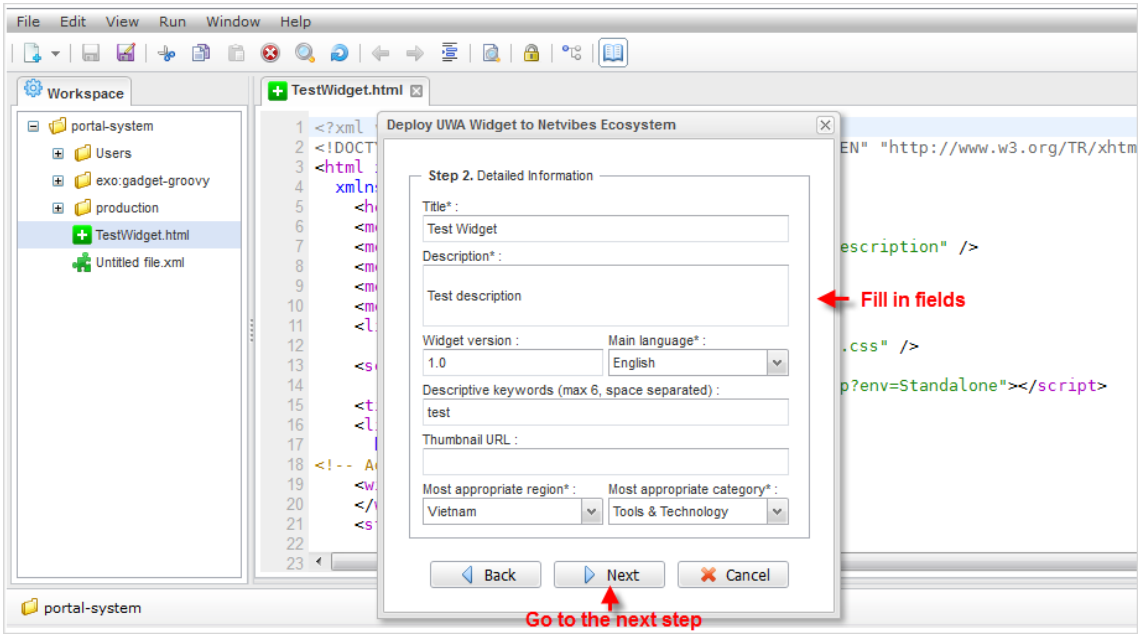
Or, go to **Run --> Deploy UWA widget** on the top menu to open the **Deploy UWA widget to Ecosystem** form:



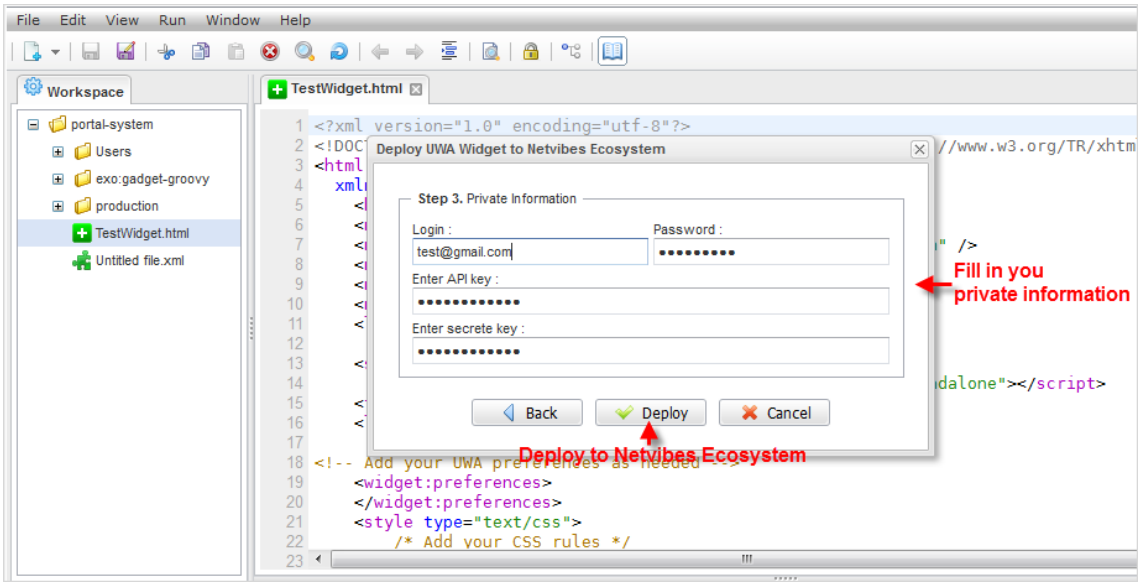
3. Paste the URL of your gadget, and do not forget to remove the "/**private**" part from it, or netvibes will not be able to access it.

4. Enter the information about the widget in the fields in the **Step 2** form. All fields marked with asterisk (*) are required..

Click **Next** after completing your details.



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



Details:

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.

Tip

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

6. Click  to start deploying the widget to the Netvibes Ecosyste.

The result is displayed in the **Output** pane. 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.
```

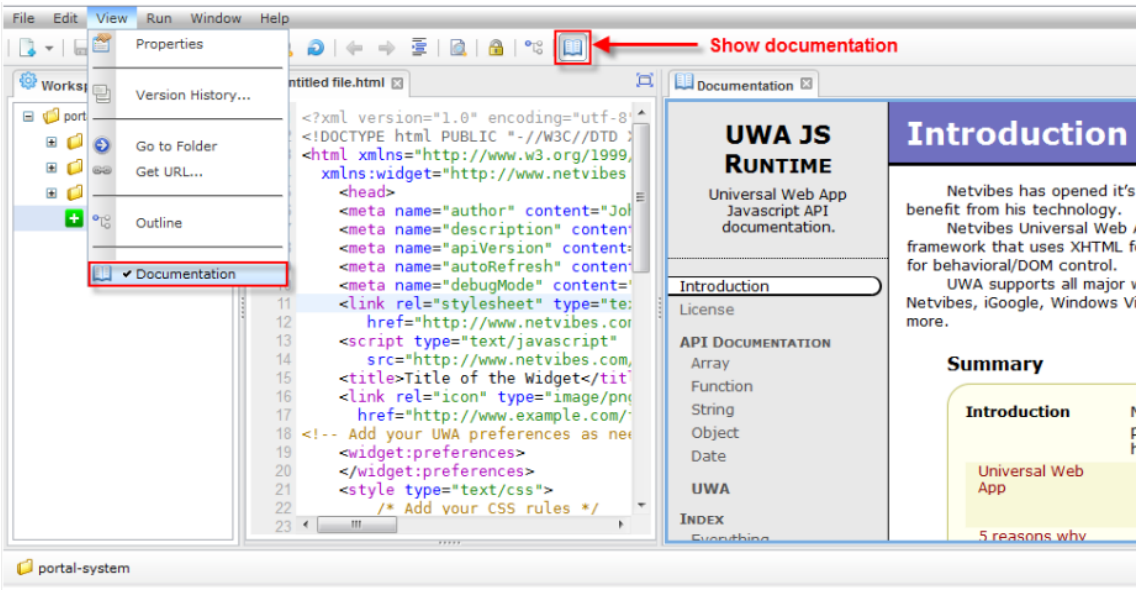
Note

The deployed widget can be found at the Ecosystem page on the **My creation** tab.

9.4. 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** on the toolbar or go to **View --> Documentation**.

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



Chromattic Data Object Usage

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

The following topics are covered:

- Create a Data Object
- Preview a Node Type
- Create a Node Type

10.1. Create a Data Object

Click  **--> Data Object**;

Or, go to **File --> New --> Data Object** from the top menu.

Your newly created file will be opened in editor, so you will be able to edit and save it.

10.2. Preview a Node Type

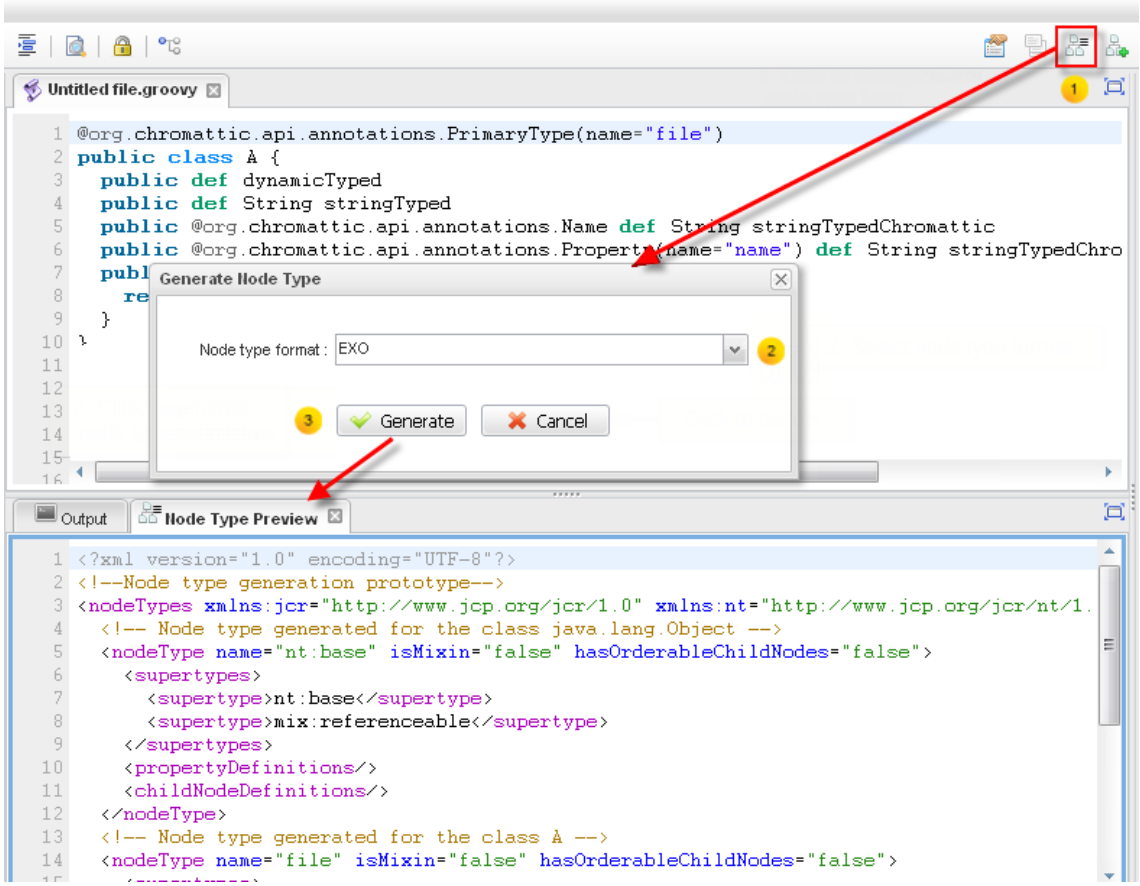
1. Click  on the **Toolbar**;

Or, go to **Run --> Preview node type** from the top menu.

2. Select a node type format in the appeared form:

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

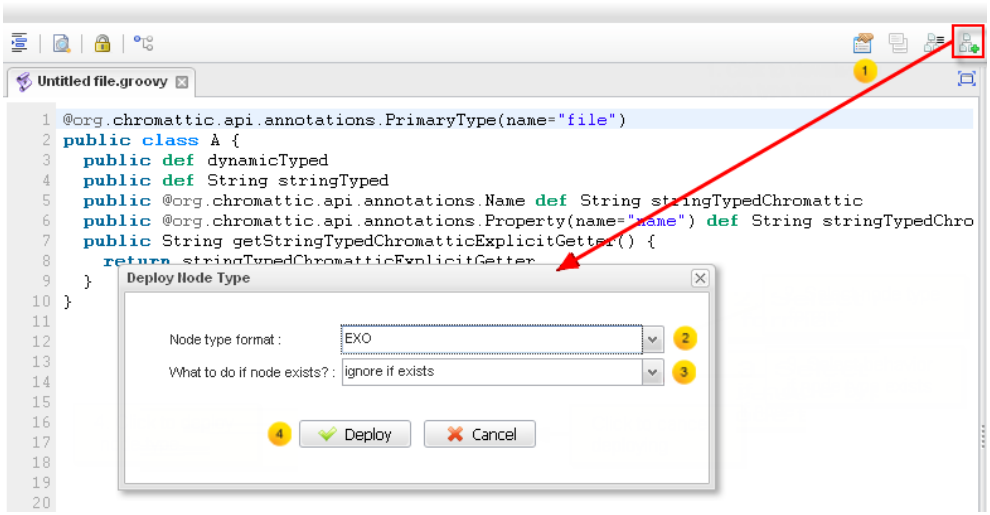
3. Click **Generate**. The generated node type definition will be displayed in the separate tab of **Operation** pane with title **Preview node type**:



10.3. Create a Node Type

1. Click  on the toolbar;

Or, go to **Run --> Deploy node type** from the top menu.



2. Select a required node type format in the **Deploy Node Type** form.

3. Select the action in case such node type already exists:

- **ignore if exists**: if the pointed node type already exists, the creation operation is ignored.

- **fail if exists**: if the pointed node type already exists, the creation operation fails.
- **replace if exists**: if the pointed node type already exists, the existing one will be replaced.

4. Click **Deploy**.

If the 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, the 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. With the versioning feature, you can:

- View version history
- Version navigation
- Restore a file to the older version

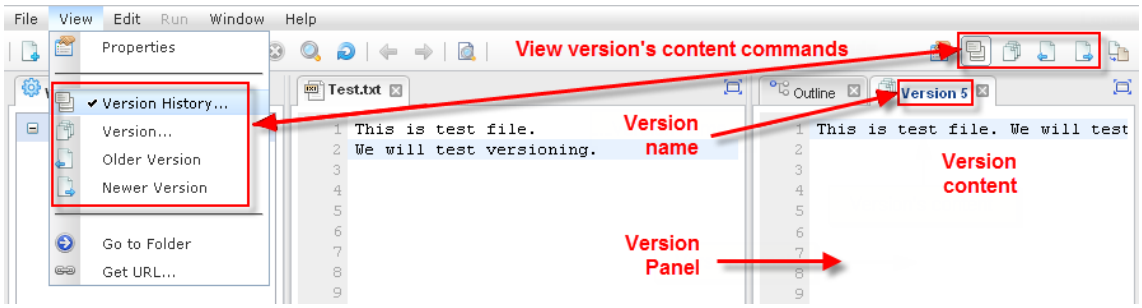
A new version of the file is created each time after the file content has been saved by clicking **Save**. There is no versionF in the version history after the file had been created only.

11.1. View version history

To view the version history of the file which is being opened/selected, click  at the right part of the toolbar;

Or, select **View --> Version History...** from the top menu.

The **Version** pane with content of the latest version are opened at the right column of the eXo IDE window.



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

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 cannot edit it.
- If any files are opened, the **Version** pane of the previously opened file will be closed automatically.

11.2. 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  or  on the toolbar, or select **View --> Older Version/Newer Version** from the top menu. If there is no newer/older version, the **View Newer Version/View Older Version** command will be disabled respectively.

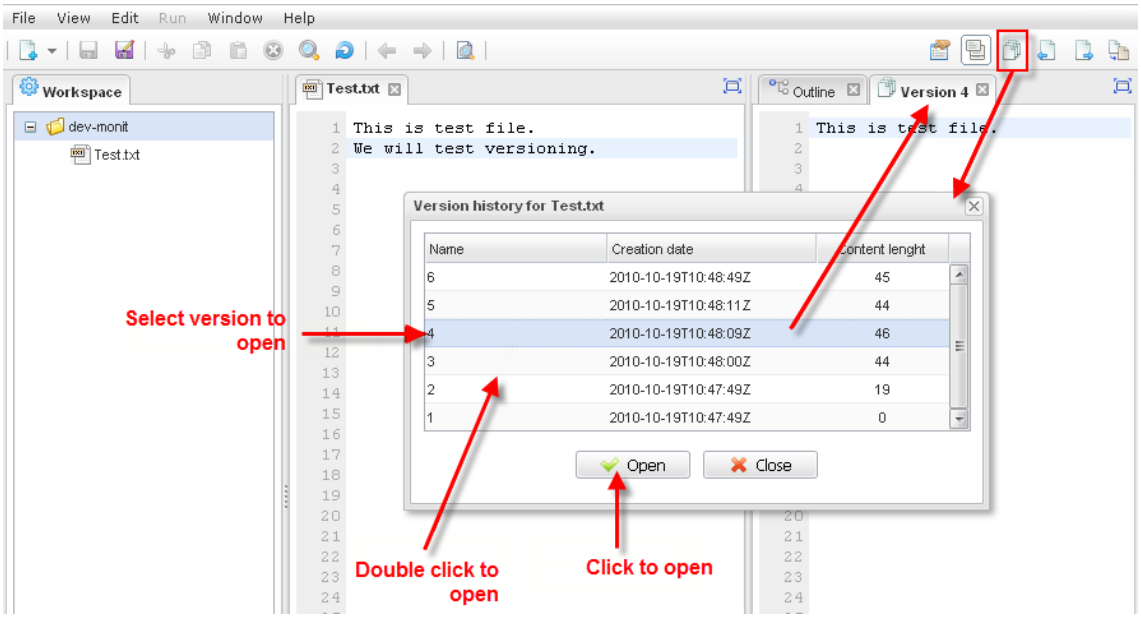
View another version from the versions history

1. Click  on the toolbar;

Or, select **View --> Version...** from the top menu.

2. Select the target version.

3. Click **Open** or double-click the version item.

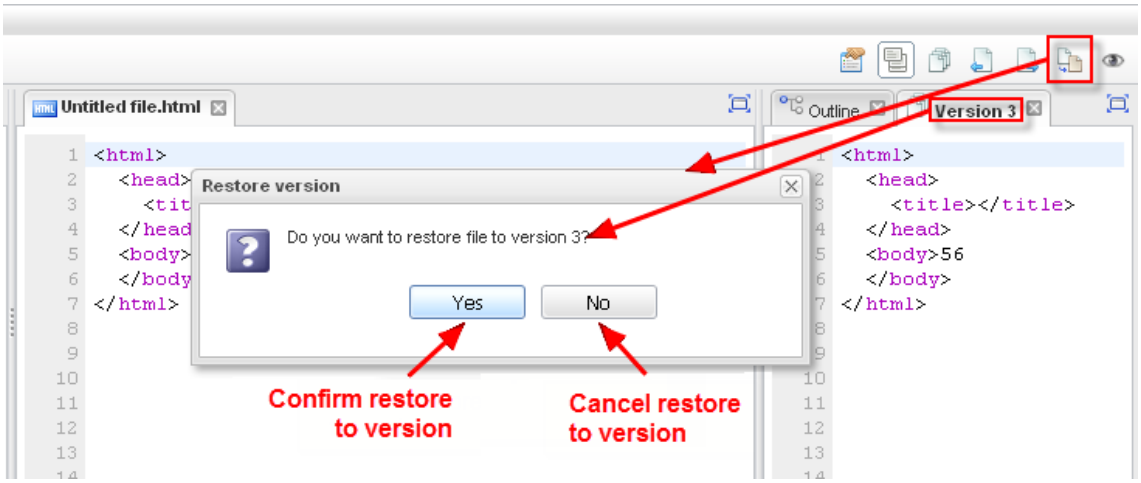


Versions are listed in the reverse order, with the latest version first.

11.3. Restore a file to the older version

1. Click  at the right corner of toolbar, or select **File --> Restore To Version** from the top menu.

2. Click **Yes** in the confirmation message to restore to the version.



Note

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

Miscellaneous and Tips

This chapter will show you all tips that may help you customize the eXo IDE to your preferences. The following topics will be covered:

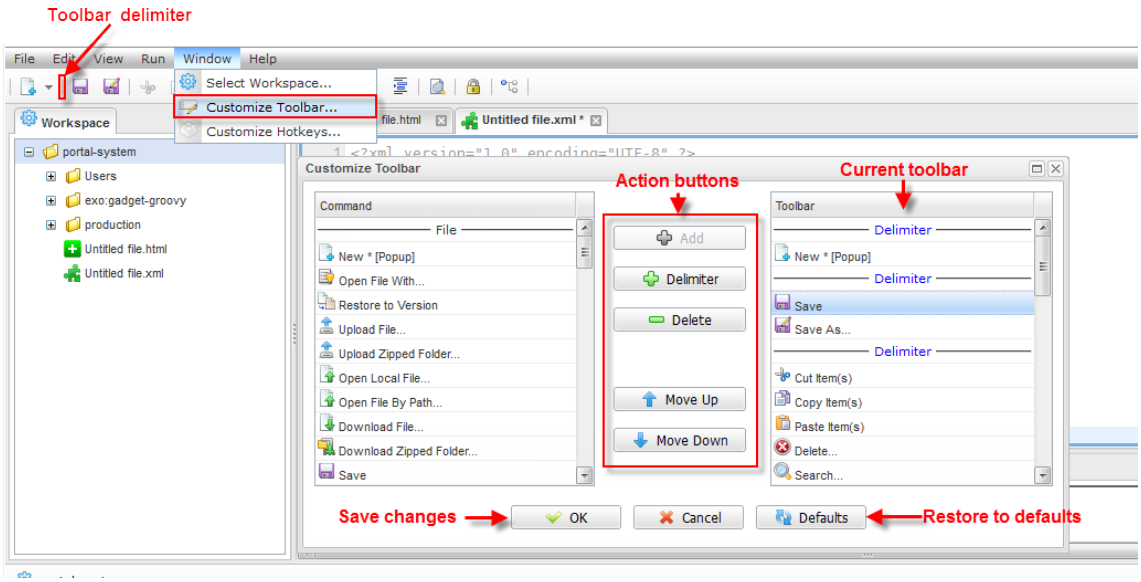
- View information about eXo IDE
- Customize the toolbar
- Customize hotkeys

12.1. View information about eXo IDE

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

12.2. Customize the toolbar

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



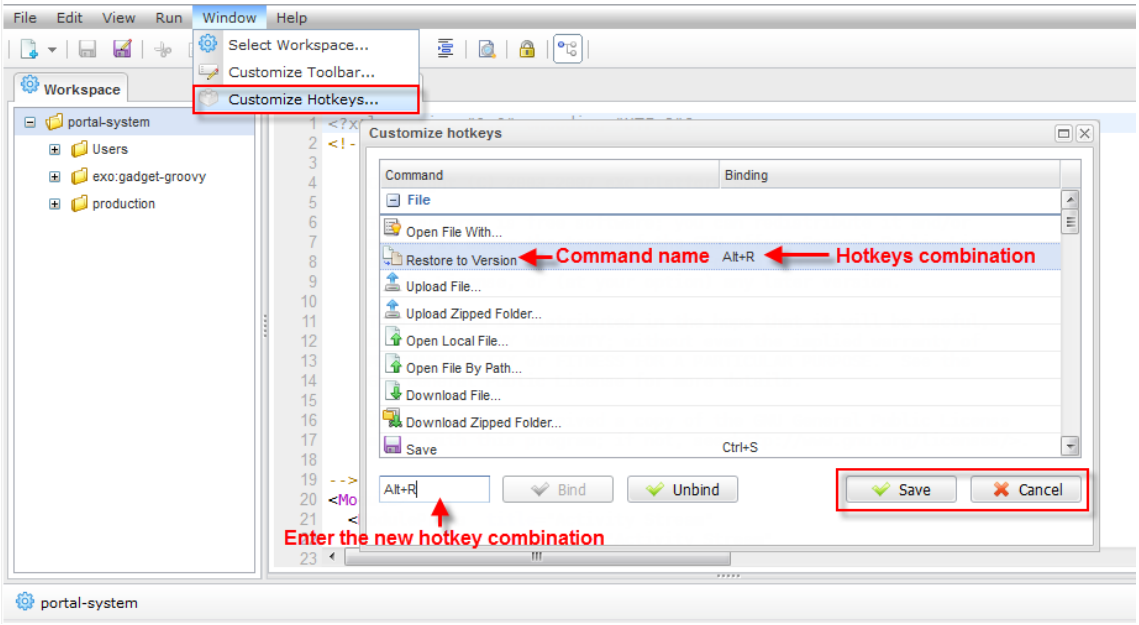
12.3. Customize hotkeys

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 overridden, they will work as usual.

Customize your own hotkeys

1. Go to **Window --> Customize Hotkeys...** from the top menu to open the **Customize hotkeys** form.

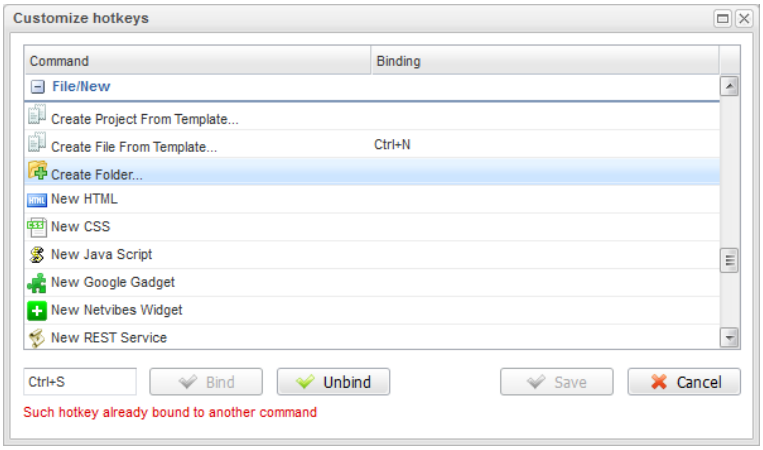


The **Customize hotkeys** form lists all commands with their respective binding hotkeys.

2. Click your desired action in the **Customize hotkeys** form.

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:



4. Click **Bind**.

5. Click **Save** to accept your new settings on the server.

Unbind hotkeys for a command

1. Open the **Customize hotkeys** window.

2. Select your needed command in the list grid.

3. Click **Unbind**.

4. Click **Save** to accept your new settings on the server.

Note

- You cannot make any changes for hotkeys of the **Editor hotkeys** group, because they are reserved and used in editors. These hotkeys also cannot be re-applied for other commands. To see these hotkeys, scroll down the end of the **Customize hotkeys** form and click the plus icon next to the **Editor hotkeys**.
- To bind the hotkeys which are the same as the predefined hotkeys, except those of the "Editor hotkeys" group, you have to select and unbind the predefined hotkeys first.

The following is the list of predefined hotkeys:

- Default global hotkeys

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

- Non-changeable hotkeys within editors

Hotkeys	Functions
Ctrl+Space	Auto-complete
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 beginning of document
Ctrl+End	Go to the end of document

Related References

- Codemirror usage
- FCKeditor user guide
- What are gadgets powered by Google
- RESTful Web Services
- Netvibes Documentation