

Using Action Script

When creating class operations to your own data model in the **Implementation** folder, you can select from the following different types of implementation:

- **Action Script:** The usage of action script is explained on this page.
- **JavaScript:** For more Information about the usage of java script refer to [Using JavaScript](#).
- **Mapping Diagram:** Refer to [Modeling Data Mapping](#) for more information about mapping diagrams.
- **Activity Diagram:** You can implement class operations by a UML activity diagram. Refer to [Modeling Activities](#) for more information on activity diagrams.

The xUML Action Language is a script like language that you can use in class operations to implement atomic actions like manipulating strings, arrays, and so forth. Such class operations are called action script.



Refer to [Action Script Language](#) for a comprehensive documentation of all elements of the xUML Action Language.

Creating an Operation Containing Action Script

Via a Quick Action

The fastest way to create an action script operation is via the quick actions of the related class.

The screenshot shows the 'Assets' panel in SAP Fiori. The 'Implementation' section is expanded, showing a tree structure of classes. The 'Idea' class is selected, and its properties are listed: 'approved: Boolean', 'employeeName: String', 'ideaDescription: String', 'instructionsText: String', and 'personnelNumber: String'. Below these, there are several methods: 'mapFromIdeaCheck', 'mapFromIdeaInput', 'mapFromInstructionsInput', and 'mapToInstanceList'. The 'Add Action Script Operation' button is highlighted with a red circle, and a tooltip is visible over it.

On this Page:

- Creating an Operation Containing Action Script
 - Via a Quick Action
 - Via the Context Menu
- Attributes of an Action Script Operation

Related Pages:

- Modeling Data Mapping
- Modeling Activities
- PAS Designer Developer Guide
 - Action Script Language

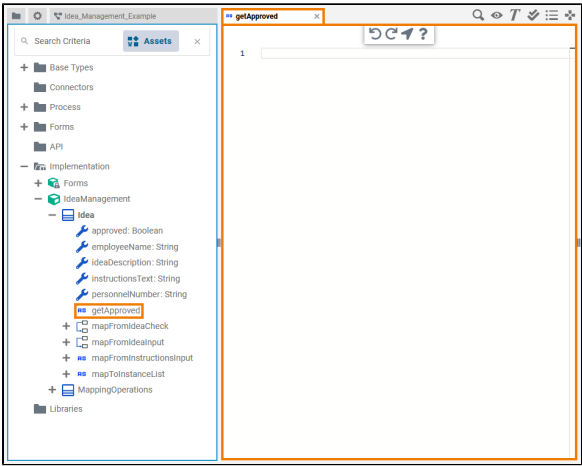
Add Action Script Operation

getApproved

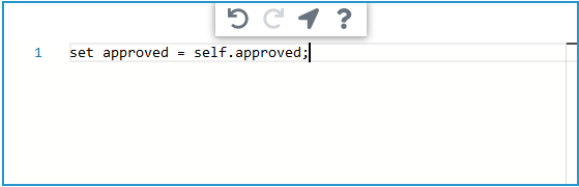
Save

Cancel

Assign a name to your new action script and click **Save**.



The Action Script Editor opens in a new tab.



You can now add action script.



Expert Advice

Click
the
help
icon (?)

to
jump
to the
docum
entatio
n of
the xU
ML
Action
Langu
age
for
detaile
d
inform
ation
about
the
usage
of
action
script.
This
docum
entatio
n
contai
ns
helpful
basic
inform
ation
such
as

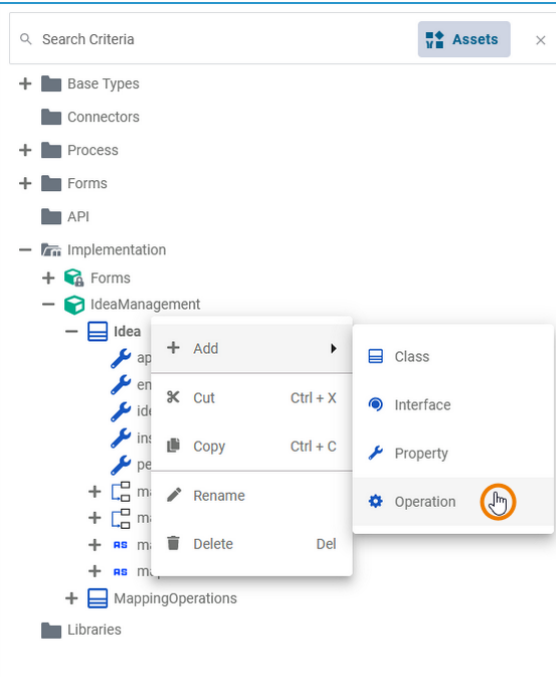
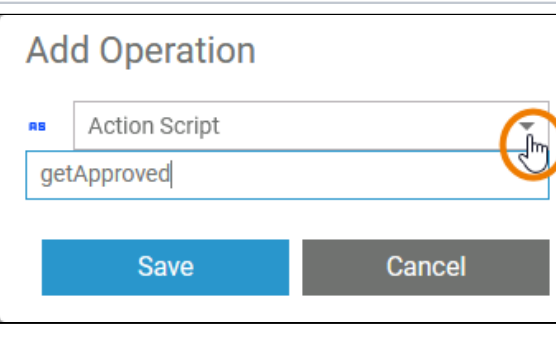
- [basic
syntax
of
the
xU
ML
Actio
n L
angu
age](#)

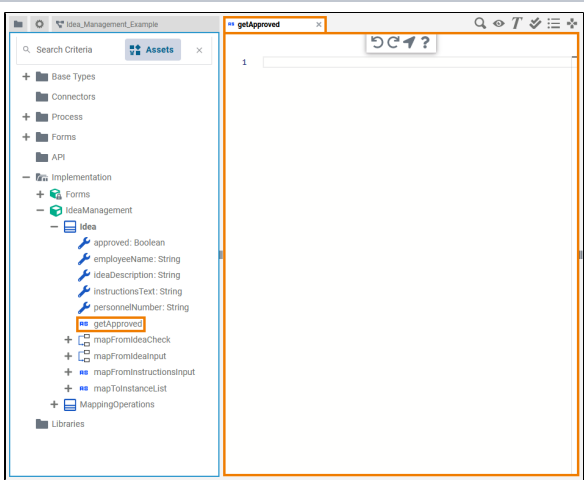
- syntax scheme of the UML Action Language
- are reference for all kinds of type operations

The script you have entered is saved and validated in the background.

Via the Context Menu

Alternatively, you can add a class operation via the context menu of a class, and create an implementation afterwards.

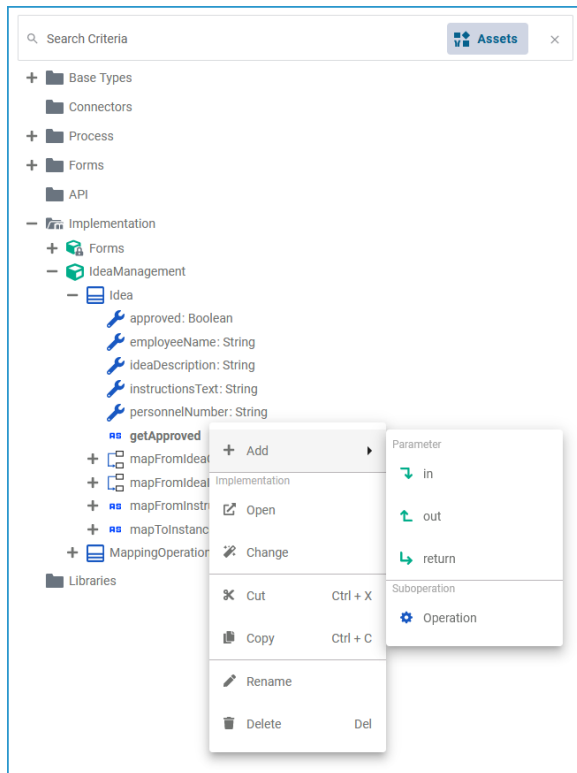
	Right-click the class you want to add an operation to and select Add Operation from the context menu.
	The dialog Add Operation opens. Select Action Script from the drop-down list, enter a name for the operation and click Save.



The new operation has been added to the class. The Action Script Editor opens automatically in a new Designer tab, and you can start entering action script (see [further above](#)).

Go to Working with the Action Script Editor for detailed information.

Once the operation has been created, you can use the quick actions and the context menu to manage it. You can:



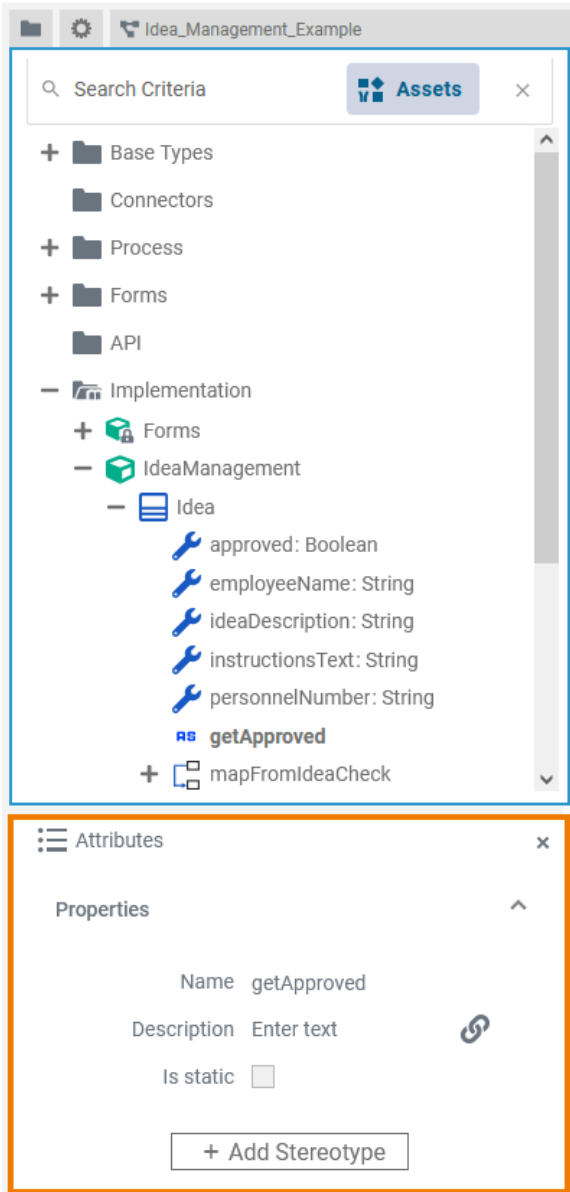
- add a parameter
 - in
 - out
 - return
- add a suboperation
- open the implementation of the action script
- change the implementation from action script to activity diagram or mapping diagram
- cut the action script operation
- copy the action script operation
- paste the action script operation (available if **Copy** or **Cut** option have been used before)
- rename the action script operation
- delete the action script operation

✓ Refer to Implementation and Modeling Data Structures for more information on your options here.

Attributes of an Action Script Operation

Select an action script operation in the **Implementation** folder of the Service panel to display its attributes in the **Attributes** panel. You can also edit them there.

Action script operations have the following attributes:



Attribute	Description	Possible Values / Example	
Name	Click here to change the Name of the related element. Action script operation names must follow certain naming rules. They • must not contain blanks • must not start with a number • must not contain special characters	getApproved	
Description	If you want to insert or change a description for the respective action script operation, click here to open a text editor where you can enter and format your text.		
Is static	Specify if the operation is static (default) or not. • Static action script operations can be called without creating an instance of the related class. They get all necessary data via their input parameters. • Wanting to call a non-static action script operation, you need to create a local instance of the related class, and call the operation on that object. This is called self context. For more information, also refer to Adding Operation Calls.	true	The action script operation is static (default) and can be used outside the context of the related class.
		false	The action script operation is non-static and needs a self object as an input.
Stereotype	Via Add Stereotype, you can add a stereotype to a action script operation. By adding a stereotype, you can extend the attributes of a action script operation with additional properties.	REST	

⋮ Attributes

×

Properties

^

Name

getApproved

Type

IdeaManagement.Id...

When you click in the Action Script Editor, the following attributes of the current action script are displayed in the **Attributes** panel. All attributes are read-only and cannot be edited there.

Attribute	Description	Example
Name	Displays the name of the current action script.	getApproved
Type	Path within the implementation folder where the corresponding action script operation resides.	IdeaManagement.Idea