

Using JavaScript

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

- **JavaScript**: The usage of JavaScript is explained on this page.
- **Action Script**: For more information about the usage of action script refer to [Using Action Script](#).
- **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.

In the Designer, you can use JavaScript operations to effectively write the function body of a (synchronous) JavaScript function.

On this Page:

- [Creating an Operation Using JavaScript](#)
 - [Via a Quick Action](#)
 - [Via the Context Menu](#)
- [Attributes of a JavaScript Operation](#)



Please note that `await` syntax is not supported in synchronous JavaScript functions.

Creating an Operation Using JavaScript

Via a Quick Action

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

A screenshot of the SAP PAS Designer interface. The left sidebar shows a tree structure with categories like Base Types, Connectors, Process, Forms, API, and Implementation. Under Implementation, there are sub-folders for Forms, Services, and Password. The 'Password' folder is currently selected. A context menu is open over the 'Password' folder icon, with the 'Add JavaScript Operation' option highlighted and circled in orange. Other options in the menu include 'Edit', 'Delete', 'Properties', and 'Script'.

Hover over the class you want to add the operation to, and click the **Add JavaScript Operation** quick action (**JS**).

Related Pages:

- [Working with the JavaScript Editor](#)
- [JavaScript Specifics](#)
- [this Context](#)
- [Using Action Script](#)
- [Modeling Data Mapping](#)
- [Modeling Activities](#)

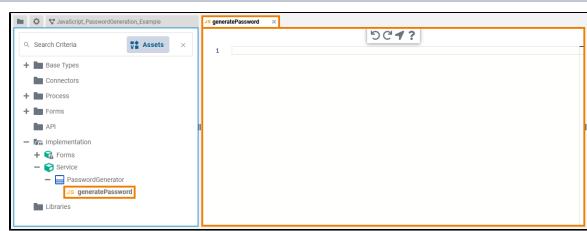
A screenshot of the 'Add JavaScript Operation' dialog box. It has a single input field containing the text 'generatePassword'. Below the input field are two buttons: 'Save' and 'Cancel'. The 'Save' button is highlighted with a yellow circle and a hand cursor icon, indicating it is the active button.

Assign a name to your new JavaScript and click **Save**.

JavaScript_PasswordGeneration_Example



Click the icon to download a simple example model that shows how you can use **JavaScript** in **SAP PAS Designer**.



The JavaScript Editor opens in a new tab.

```
const letters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
```

A screenshot of the Mendix JavaScript Editor showing a code editor window. The code shown is a single line: `const letters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';` The code editor has a blue header bar with a question mark icon.

You can now add JavaScript.

Expert Advice

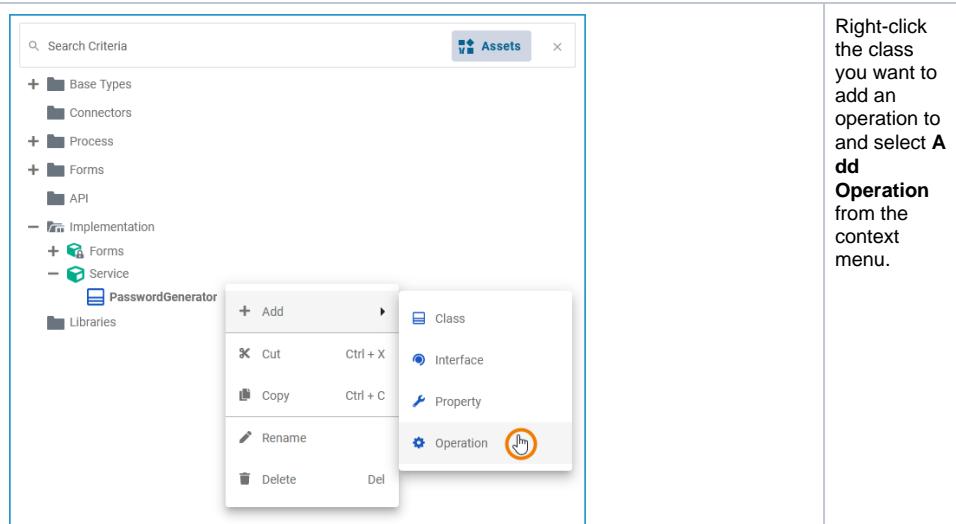
Click **Go to documentation** (?) to jump to the documentation for detailed information about the usage of JavaScript in the Designer. This documentation contains helpful basic information such as

- working with the editor or Java Script Specifics
- this Context in Java Script
- working with the editor or Java Script Specifics

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

Via the Context Menu

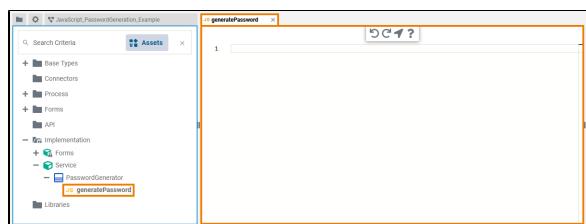
Alternatively, you can add a class operation via the context menu of a class, and create an implementation afterwards.



Right-click the class you want to add an operation to and select **Add Operation** from the context menu.

A screenshot of the 'Add Operation' dialog. It has a dropdown menu set to 'JavaScript'. Below it is a text input field containing 'generatePassword'. At the bottom are 'Save' and 'Cancel' buttons. A hand cursor icon is positioned over the 'generatePassword' input field.

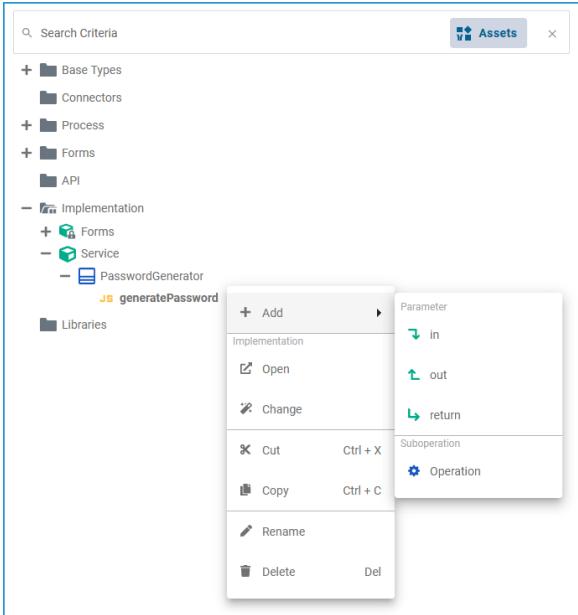
The dialog **Add Operation** opens. Select **Java Script** from the dropdown list, enter a name for the operation and click **Save**.



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



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



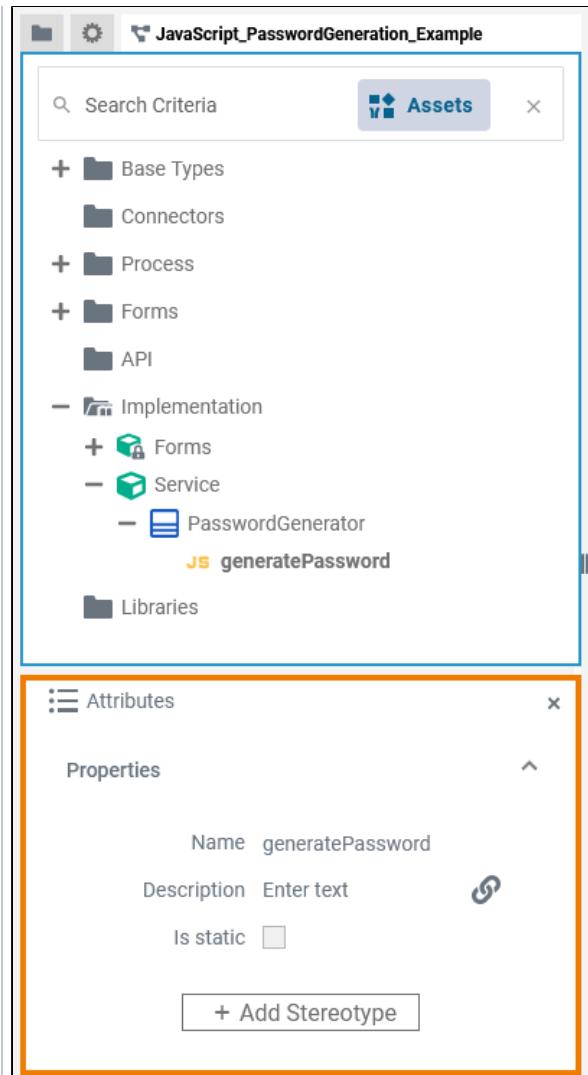
- add a parameter
 - in
 - out
 - ret
 - urn
- add a suboperation
- open the implementation of the JavaSc
- change the implementation from JavaScript to action script, activity diagram or mapping diagram
- cut the JavaScript operation
- copy the JavaScript operation
- paste the JavaScript operation (available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



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

Attributes of a JavaScript Operation

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



Attribute	Description	Possible Values / Example
Name	<p>Click here to change the Name of the related element.</p> <p>JavaScript operation names must follow certain naming rules. They</p> <ul style="list-style-type: none"> • must not contain blanks • must not start with a number • must not contain special characters 	javaScriptOperation
Description	If you want to insert or change a description for the respective JavaScript operation, click here to open a text editor where you can enter and format your text.	
Is static	<p>Specify if the operation is static (default) or not.</p> <ul style="list-style-type: none"> • Static JavaScript 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 JavaScript operation, you need to create a local instance of the related class, and call the operation on that object. This is called this context. <p>For more information, also refer to Adding Operation Calls.</p>	<p>true The JavaScript operation is static (default) and can be used outside the context of the related class.</p> <p>false The JavaScript operation is non-static and needs a self object as an input.</p>
Stereotype	Via Add Stereotype , you can add a stereotype to a JavaScript operation. By adding a stereotype, you can extend the attributes of a JavaScript operation with additional properties.	REST

Attributes

Properties

Name generatePassword

Type

When you click in the JavaScript Editor, the following attributes of the current JavaScript 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 JavaScript.	javaScript Operation
Type	Path within the implementation folder where the corresponding JavaScript operation resides.	