Simplifier Developer

Documentation & Community

PDF generated January 28, 2020

Table Of Contents

Access Business Object via Script 4
Access Control for Web Applications 5
Action 9
Active Directory 10
Add a new Library 12
Add a PDF Template 15
Additional Requirements for Oracle Databases as Backend 16
Administrate Templates 17
Android Client 19
Anonymous Profile for Plugins 24
App Links 25
Applications 26
Assets 29
Assign Roles 31
Asynchronous Connector Request Json Examples 32
Authentication 33
Authentication for Web Applications 34
Automated Testing 35
Backups 38
Basic Concept / Technology 39
Basic Protection of Internet Access 41
BROWSE Call - OPC/UA Connector 42
Build a PDF Template 46
Business Objects 48
Change your Password 52
Checklist - Simplifier onPremise Installation 54
Checklist SAP SSO over SOAP 55
Client-Side - Access Connectors 58
Client-Side - Access other Business Objects 59
Client-Side - Access Plugins 60
Client-Side Business Object API 61
Code Designer 63
Collaboration 64
Collection Type 66
Conferencing Plugin for WebRTC Calls 70
Connector Access via Script 89
Connector Call Specific Parameters 90
Connector Call via Script 95
Connectors 96
Content Files 98
Content Repository 113
Controlled Integration of Data and Content into Web Applications 122
Copy Connector Calls 123
Copy Data Types 125
Create a Transport 127
Create an OpenUI5 Widget 131
Create and manage connector calls 137
Create and Manage Connectors 143
Create and Manage Functions 151
Create client-side Business Object 156
Create server-side Business Objects 161

Create your first Application . . . 166
CSS Editor . . . 170
CSV Connector . . . 173
CSV Connector Calls . . . 175
Current Release & Archive . . . 180
Data Centers of Simplifier Cloud . . . 183
Data Object . . . 184
Data Types . . . 185
Data Workbench . . . 188
Database Setup . . . 193
Delete a PDF Template . . . 194
Deploy and Preview . . . 195
Deployment & Installation Instructions . . . 198
Device Condition . . . 200
Docker Hub . . . 202
Docker Installation . . . 203
Docker on Mac . . . 205
Docker on Ubuntu / Debian . . . 208
Docker on Windows 10 . . . 210
Domain Type . . . 216
DQP System . . . 220
Edit a PDF Template . . . 223
Email Connector . . . 225
Email Connector Call . . . 226
Enumeration in Widget Properties . . . 229
Events . . . 233
Example Apps . . . 239
Example of using OData . . . 240
Execution Log . . . 258
Features and supported operating systems . . . 261
Fetch a PDF Template . . . 263
Filter . . . 264
FQDN . . . 267
General Instructions . . . 268
General Requirements for On-Premise-Installations . . . 269
Getting Started . . . 271
Glossar . . . 274
Group Overview . . . 275
Handling & Updating an On-Premise Installation . . . 279
Implementation of Web Application Firewalls . . . 281
Import Manual Transport . . . 286
Installation PDF Plugin . . . 291
Installing an On-Premise Image . . . 293
Integration of external Libraries . . . 295
Integration of Libraries - addAfterInitHandler . . . 298
Integration of Libraries - addBeforeInitHandler . . . 299
Integration of Libraries - addScript . . . 300
Integration of Libraries - addStyle . . . 301
Interfaces . . . 302

Access Business Object via Script

https://developer.simplifier.io/documentation/applications/process-dashboard-and-designer/logic/business-object-via-script/

```
this.callBusinessObject(businessObjectName, method, payload, callback, showBusyIndicator, failOnError, failCallback, parametrized)
```

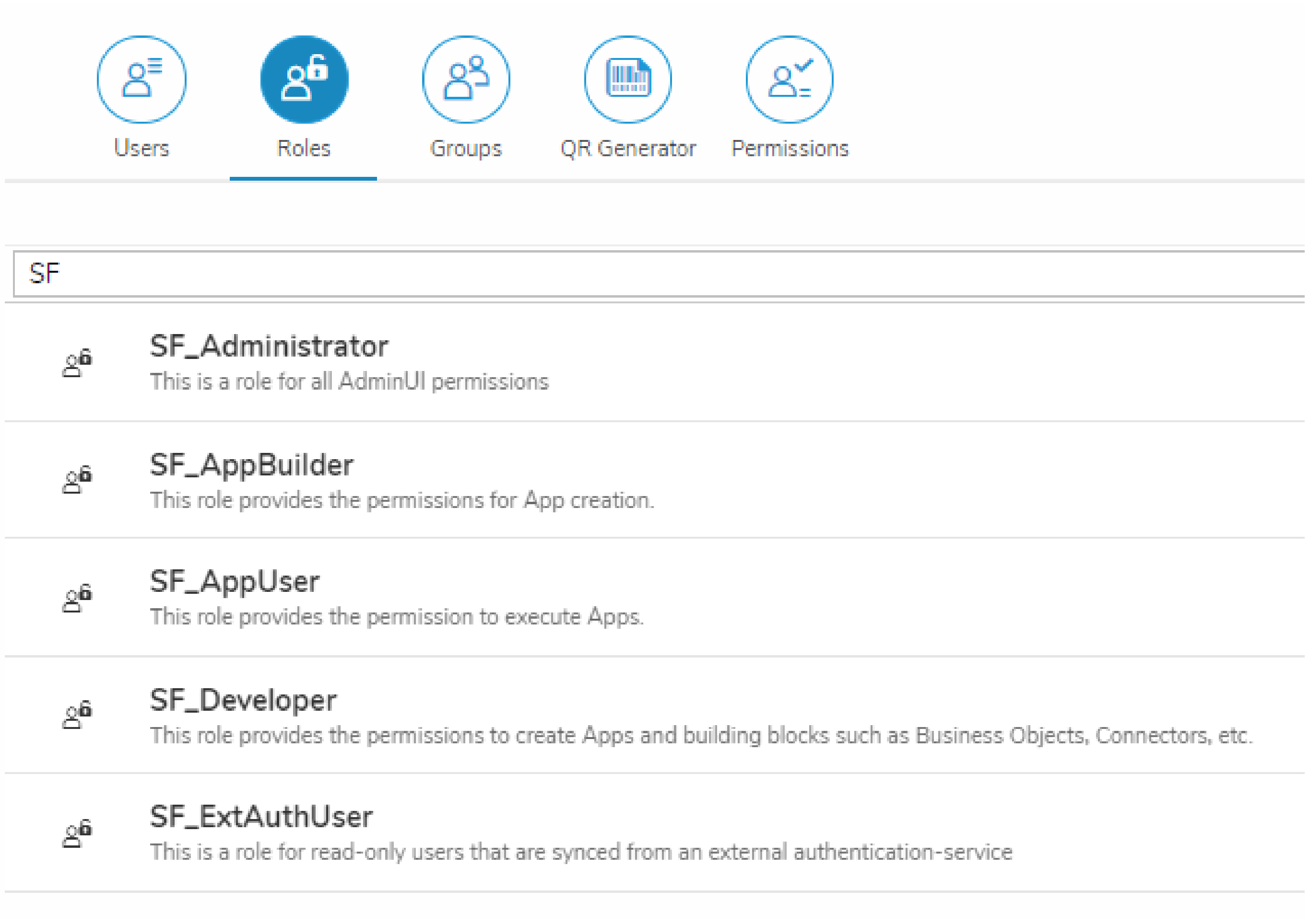
businessObjectName	the name of the business object
method	name of the script template to be called
payload	JSON object with parameters as required by the called script
callback	function, which is called after the successful execution of the connector
showBusyIndicator	boolean value that indicates whether the screen has to be blocked by a loading bar (true) or not (false)
failOnError	boolean value that indicates whether the connector should be called in case of an error of the function passed via "failCallback" (false) or not (true)
failCallback	function, which is called in case of an error in the connector, if false "failOnError" is passed
parametrized	boolean value that indicates whether the called parameters in the payload according to the rules in the script template are to be verified (true) or not (false)

Access Control for Web Applications

https://developer.simplifier.io/documentation/security-guidelines/access-control-for-web-applications/

Access control is very important because it prevents unauthorized persons from having access.

Use the standard roles for access control



SF

SF_Administrator
This is a role for all AdminUI permissions

SF_AppBuilder
This role provides the permissions for App creation.

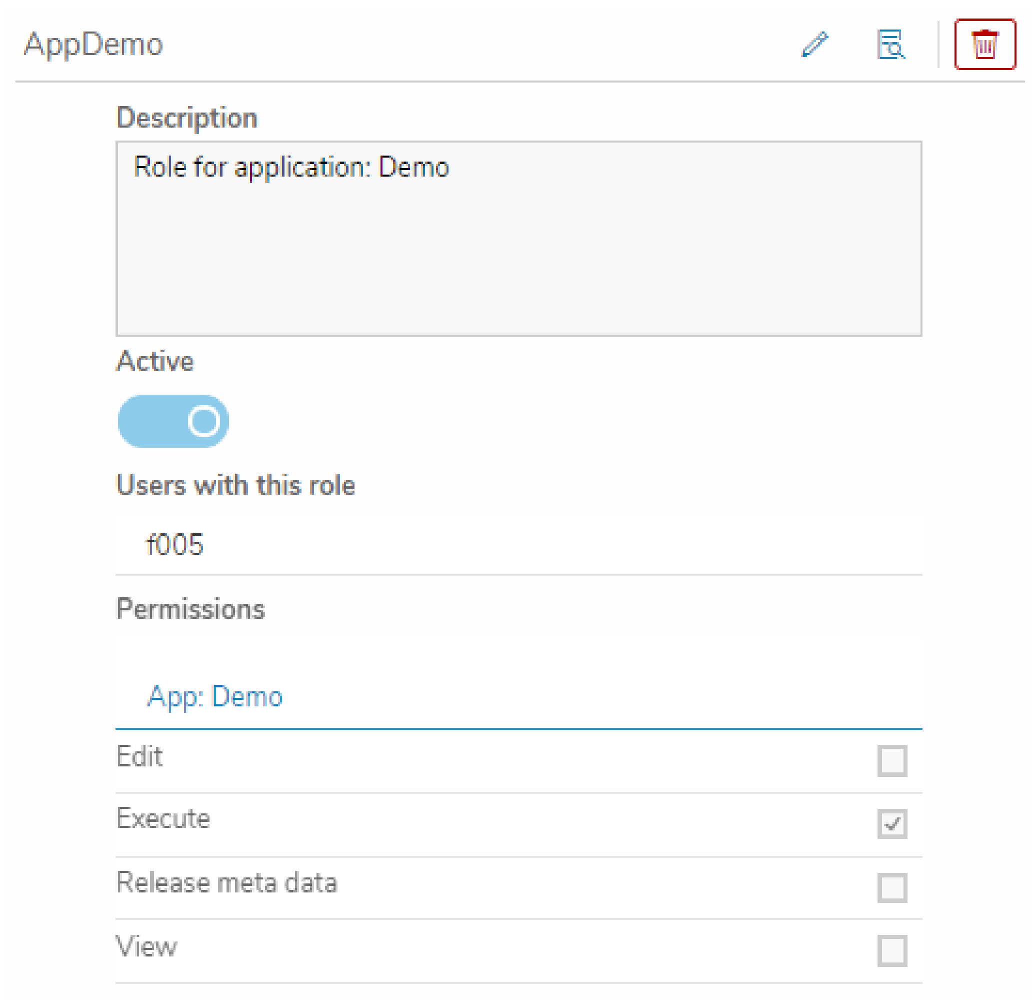
SF_AppUser
This role provides the permission to execute Apps.

SF_Developer
This role provides the permissions to create Apps and building blocks such as Business Objects, Connectors, etc.

SF_ExtAuthUser
This is a role for read-only users that are synced from an external authentication-service

SF_Administrator	This is a role for all AdminUI permissions
SF_AppBuilder	This role provides the permissions for App creation
SF_AppUser	This role provides the permission to execute Apps
SF_Developer	This role provides the permissions to create Apps and building blocks such as Business Objects, Connectors, etc.
SF_ExtAuthUser	This is a role for read-only users that are synced from an external authentication-service

In general, you should only assign permissions with the Characteristic **Execute** to end-users.


AppDemo

Description

Role for application: Demo

Active

Users with this role

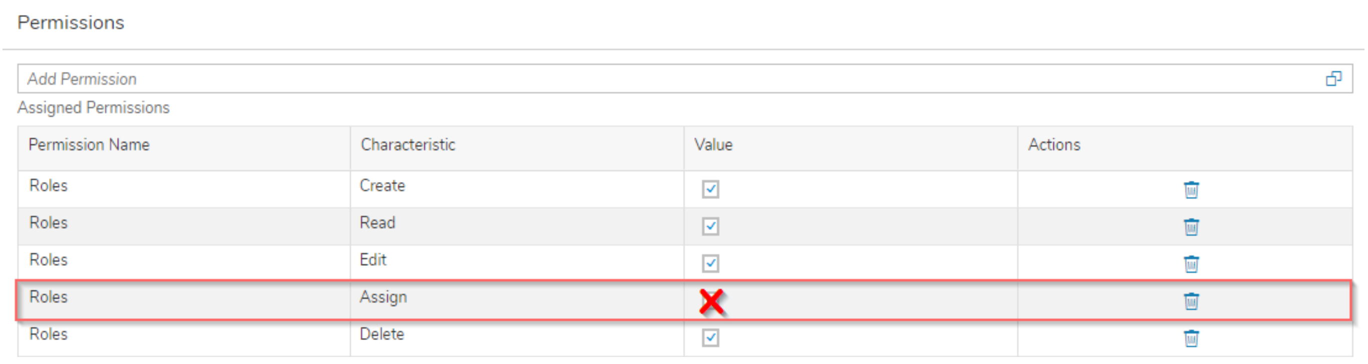
f005

Permissions

App: Demo

Edit	☐
Execute	✓
Release meta data	☐
View	☐

Never assign the Permission **Roles** with the Characteristic **Assign** to external users!


Permissions

Add Permission

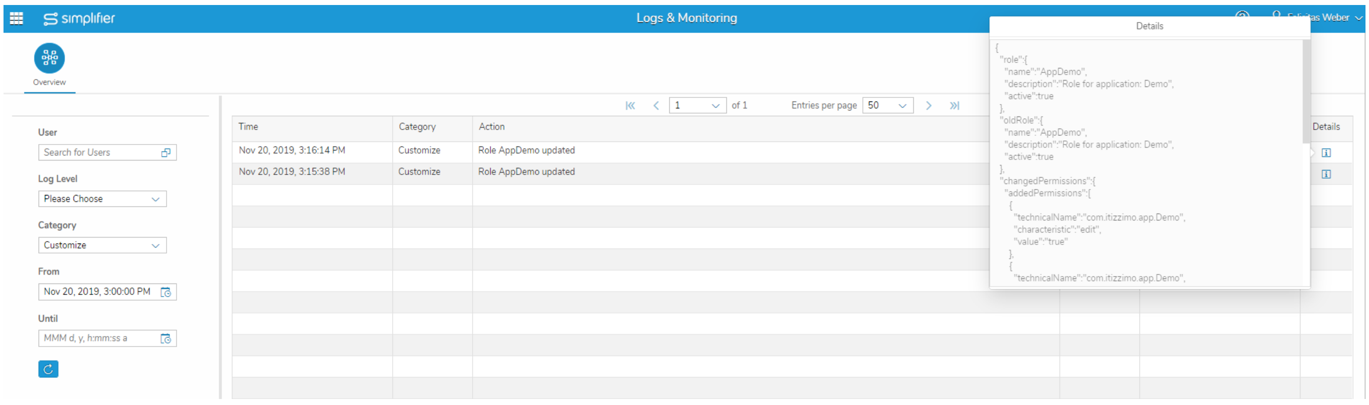
Assigned Permissions

Permission Name	Characteristic	Value	Actions
Roles	Create	✓	
Roles	Read	✓	
Roles	Edit	✓	
Roles	Assign	✗	
Roles	Delete	✓	

Never select this checkbox for external or non-administrative users!

Monitor role changes in the system centrally

All changes are written to the system log. This enables you to monitor role changes centrally, as well as role and permission assignments.



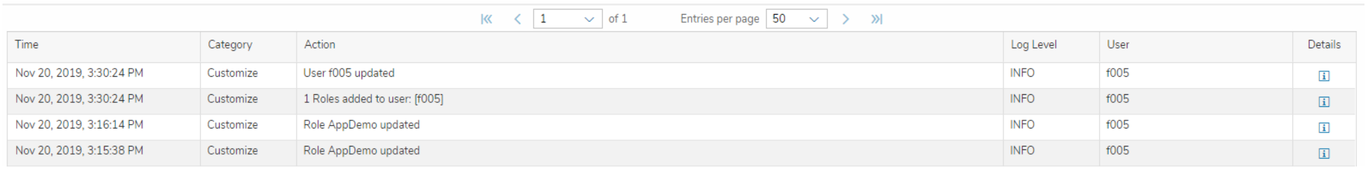
User
Search for Users
Log Level
Please Choose
Category
Customize
From
Nov 20, 2019, 3:00:00 PM
Until
MMM d, y, h:mm:ss a

1 of 1 Entries per page 50

Time	Category	Action	Details
Nov 20, 2019, 3:16:14 PM	Customize	Role AppDemo updated	ℹ
Nov 20, 2019, 3:15:38 PM	Customize	Role AppDemo updated	ℹ

Details

```
{
 "role":{
  "name":"AppDemo",
  "description":"Role for application: Demo",
  "active":true
 },
 "oldRole":{
  "name":"AppDemo",
  "description":"Role for application: Demo",
  "active":true
 },
 "changedPermissions":{
  "addedPermissions":[
   {
    "technicalName":"com.itizzimo.app.Demo",
    "characteristic":"edit",
    "value":"true"
   },
   {
    "technicalName":"com.itizzimo.app.Demo",
```



1 of 1 Entries per page 50

Time	Category	Action	Log Level	User	Details
Nov 20, 2019, 3:30:24 PM	Customize	User f005 updated	INFO	f005	ℹ
Nov 20, 2019, 3:30:24 PM	Customize	1 Roles added to user: [f005]	INFO	f005	ℹ
Nov 20, 2019, 3:16:14 PM	Customize	Role AppDemo updated	INFO	f005	ℹ
Nov 20, 2019, 3:15:38 PM	Customize	Role AppDemo updated	INFO	f005	ℹ

Action

https://developer.simplifier.io/documentation/applications/process-dashboard-and-designer/action/

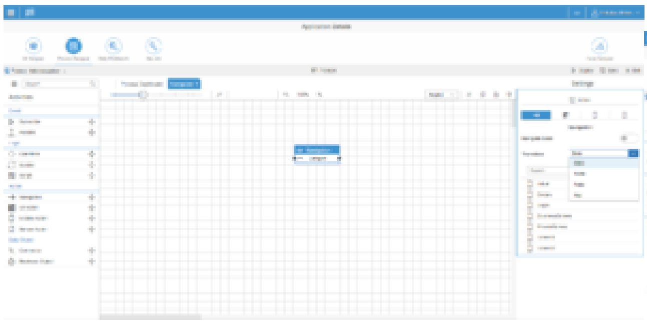
Navigation | UI Action | Mobile Action | Server Action

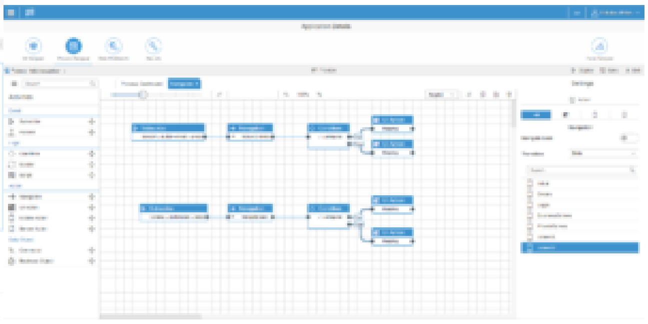
Navigation

Vimeo Video

The Navigation element is used to trigger a navigation from one screen to another. By dragging it to the main screen, a list of screens to navigate to will appear in the right pane.

You can set that the navigation should navigate back, or choose between several transitions: Slide, Fade, Flip or None.





UI Action

With the UI Action element, you can map different widgets, variables and auto fields to another.

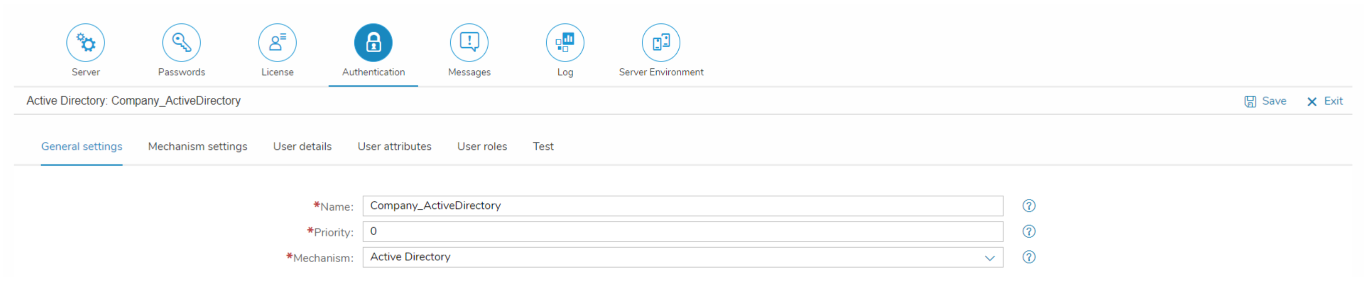
Let's say there is a very simple login screen with an input field for the name and a button to submit.
The button should not be responsive, as long as the login field is empty. This can be achieved with the UI Action in the Process Designer.

Active Directory

https://developer.simplifier.io/documentation/admin-settings/authentication-settings/active-directory/

Simplifier is able to sync users of Active Directories, like users from other LDAP sources.

General Settings

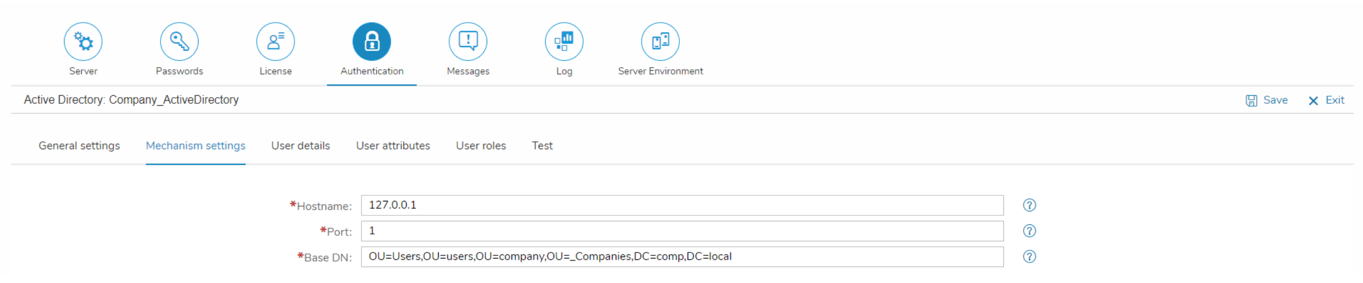


Active Directory: Company_ActiveDirectory

*Name: Company_ActiveDirectory
*Priority: 0
*Mechanism: Active Directory

Name	Name under which this authentication mechanism settings is saved
Priority	The position of the execution of the respective authentication mechanism – the higher the number, the earlier the respective authentication mechanism is used. If same numbers are available, the sequence is determined lexicographically ascending
Mechanism	The authentication mechanism

Mechanism Settings



Active Directory: Company_ActiveDirectory

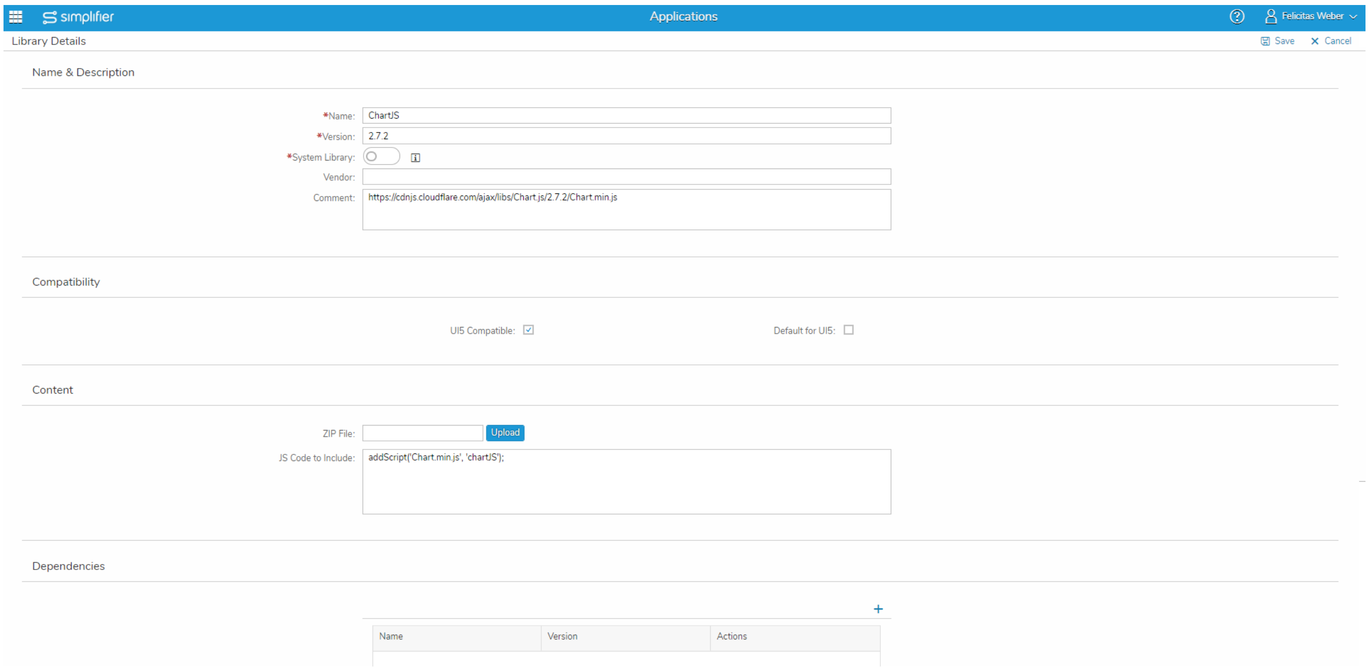
*Hostname: 127.0.0.1
*Port: 1
*Base DN: OU=Users,OU=users,OU=company,OU=_Companies,DC=comp,DC=local

Hostname	The hostname of the server may be an IPv4 address or a fully-qualified hostname (FQHN)
Port	The port of the server
Base DN	The entry point for the directories.

Add a new Library

https://developer.simplifier.io/documentation/applications/including-libraries/add-new-library/

To add a new library click on the "+" in the right corner.



Name & Description

*Name:	ChartJS
*Version:	2.7.2
*System Library:	
Vendor:	
Comment:	https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.2/Chart.min.js

Compatibility

UI5 Compatible: ✓ Default for UI5: ☐

Content

ZIP File: Upload

JS Code to Include: addScript('Chart.min.js', 'chartJS');

Dependencies

Name	Version	Actions

Now you can fill in the following parameter:

	Parameter	**Description**
Name & Description	Name	The name of the library **NOTE:** The combination of name and version number must be unique!
	Version	The version of the library
	Vendor	The vendor of the library
	Comment	A description of the library, or e.g. license information
Compatability	UI5 compatible	Controls the assignment to UI5 Apps
	Default for UI5	Assigns the library automatically when creating UI5 Apps
Content	ZIP file	The ZIP file, that contains the library
	JS code to include	Code snippet to integrate the library into Apps
Dependencies	Dependencies	Dependencies to other libraries can be added via the plus icon

Add a PDF Template

https://developer.simplifier.io/documentation/plugins/pdf-plugin/technical-call-pdf-plugin/add-pdf-template/

Add Template

To add a template, you need the following parameter:

URL	/client/1.0/PLUGIN/pdfPlugin/adminTemplateAdd	
Input-Parameter	**Name**	Template name
	Data	Template content (Base64-coded)
	Stylesheet	Content of the LESS Stylesheets (Base64-coded, optional)
	PreviewJson	Content of the sample data in JSON format (Base64-coded, optional)
Output-Parameter	None	

Example for a call:

```
{
    "name": "templatename",
    "data": "SGFsbG8gV2VsdA==\",
    "stylesheet: "SGFsbG8gV2VsdA==\",
    "previewJson": "SGFsbG8gV2VsdA==\"
}
```

Output example:

```
{
    "success": true
}
```

Additional Requirements for Oracle Databases as Backend

https://developer.simplifier.io/documentation/installation-instructions/general-instructions/additional-requirements-oracle-databases-backend-premise-installation/

On-Premise Installation

Oracle as a DB backend for the Simplifier requires some additional server settings, which are listed below. The Simplifier is currently running with MySQL 5.7 and Oracle 11g.

Database Settings within the Oracle Database:

Parameter	Recommended Value
OPEN_CURSORS	3000

Supported Oracle version:

Oracle Database 11g Release 11.2 - 64bit

Desired/recommended instance names (Productive and Test):

simplifierp and simplified

Required tablespaces:

simplifier 5G, Temp 1G, Undo 512 MB, Users 5MB

Oracle user and required roles and permissions:

simplifier, simplifier_np (in Prod and Test) permissions to run DDL

Database Characterset:

AL32UTF8

National Characterset:

UTF8

Default language:

German, Germany

Processes and Sessions:

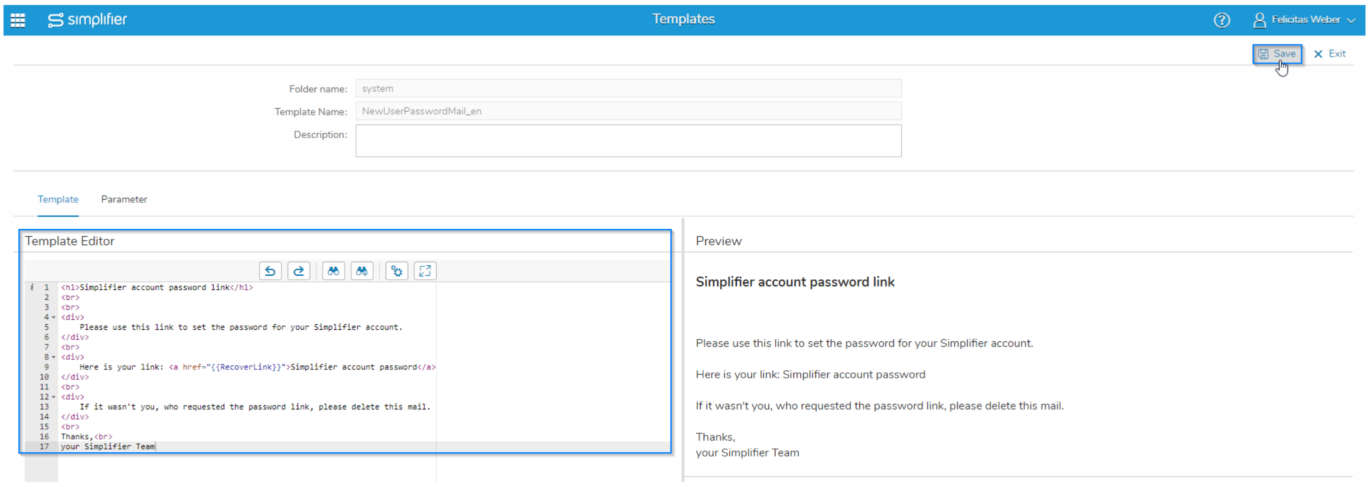
Value to 1000

Administrate Templates

https://developer.simplifier.io/documentation/templates/administrate-templates/

Templates are HTML templates that allow you to create and consistently use patterns.

To create a new template, click on the plus icon in the template overview. Select the folder name and enter a template name, optionally a description. Now enter the HTML template content. On the right side, you see a preview of the template. Once you have created the template content, click Save.



Folder name: system
Template Name: NewUserPasswordMail_en
Description:

```
<h1>Simplifier account password link</h1>
<br>
<br>
<div>
    Please use this link to set the password for your Simplifier account.
</div>
<br>
<div>
    Here is your link: <a href="{{RecoverLink}}">Simplifier account password</a>
</div>
<br>
<div>
    If it wasn't you, who requested the password link, please delete this mail.
</div>
<br>
Thanks,<br>
your Simplifier Team
```

Simplifier account password link

Please use this link to set the password for your Simplifier account.

Here is your link: Simplifier account password

If it wasn't you, who requested the password link, please delete this mail.

Thanks,
your Simplifier Team

In the Template Editor, you have several options in the toolbar:

- Undo
- Redo
- Search
- Search and Replace
- Settings
- Fullscreen

In addition, it is possible to parameterize templates. To do this, switch to the tab ‘Parameters’. Via the plus icon, you can add new input parameter.

Android Client

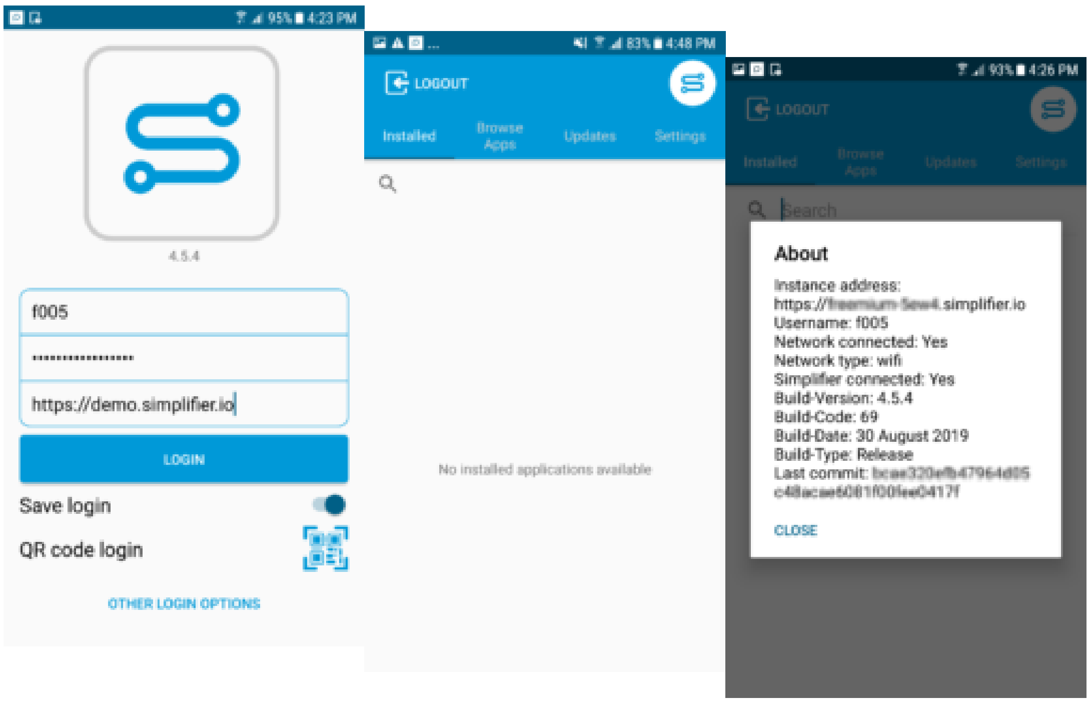
https://developer.simplifier.io/documentation/getting-started/simplifier-mobile-client/android-client/

Below is a description of the **Simplifier Mobile Client for Android**. After you have downloaded the Simplifier Mobile Client from the Play Store, start it on your mobile device.

First, you have to authenticate yourself on the login screen with your Simplifier **username** and **password**. Enter the **instance** you want to access. If the device has Touch ID, you can choose to restore your password with it. You can **save your login** so you don't have to re-enter it every time.

Tip: Use the **QR code login** that fills in your username, password and instance URL. Read here how to create a corresponding QR code in Simplifier.

Once you have been successfully authenticated, you are in the overview of installed applications. In the beginning, this overview is empty. At any time, you can log out by clicking on the logout button in the top left corner. At the top right, on the Simplifier icon, various information will be displayed.



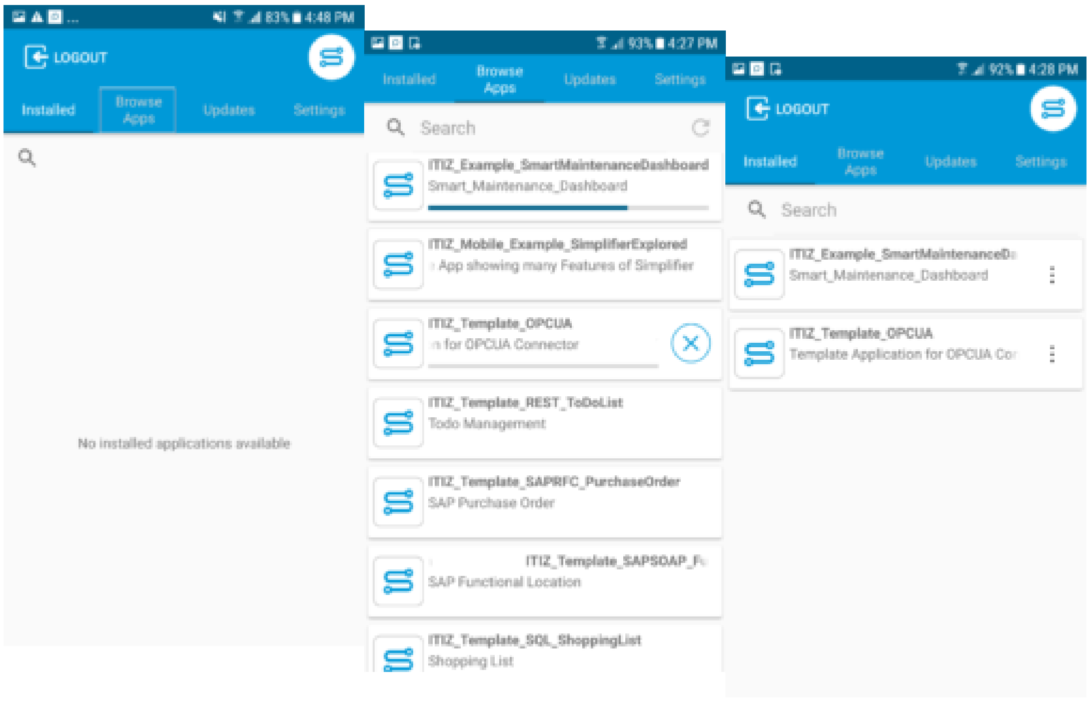
About

Instance address:
https://[redacted].simplifier.io
Username: f005
Network connected: Yes
Network type: wifi
Simplifier connected: Yes
Build-Version: 4.5.4
Build-Code: 69
Build-Date: 30 August 2019
Build-Type: Release
Last commit: [redacted]

CLOSE

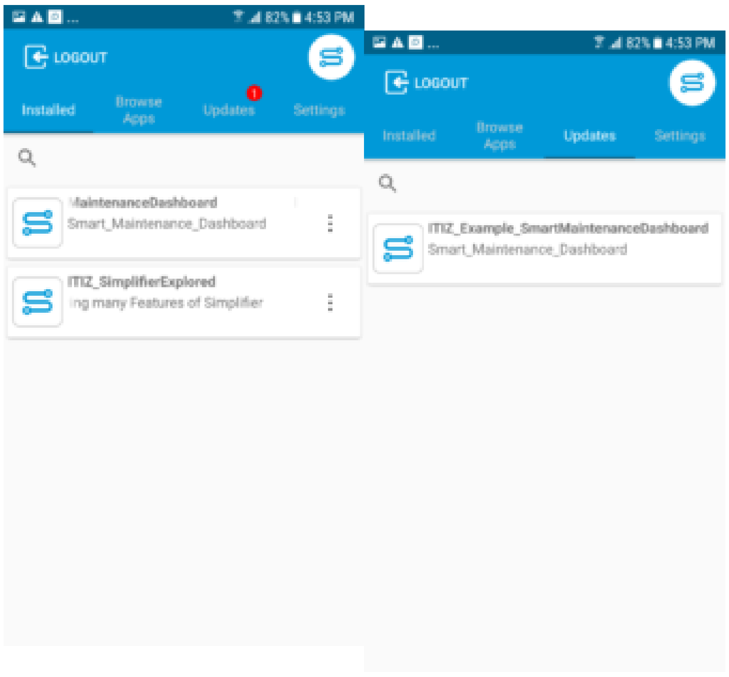
Browse Apps

To use apps on your mobile device, switch to the screen **Browse Apps**. You see an overview of all applications that are on the specified instance. To install apps, simply click on the entries. When the apps are downloaded, **Installed** will display a notification with the number of newly installed apps. You can delete the installed application by swiping to the left.


LOGOUT
Installed
Browse Apps
Updates
Settings
No installed applications available
Search
ITIZ_Example_SmartMaintenanceDashboard
Smart_Maintenance_Dashboard
ITIZ_Mobile_Example_SimplifierExplored
App showing many Features of Simplifier
ITIZ_Template_OPCUA
for OPCUA Connector
ITIZ_Template_REST_ToDoList
Todo Management
ITIZ_Template_SAPRFC_PurchaseOrder
SAP Purchase Order
ITIZ_Template_SAPSOAP_F
SAP Functional Location
ITIZ_Template_SQL_ShoppingList
Shopping List
Template Application for OPCUA Co

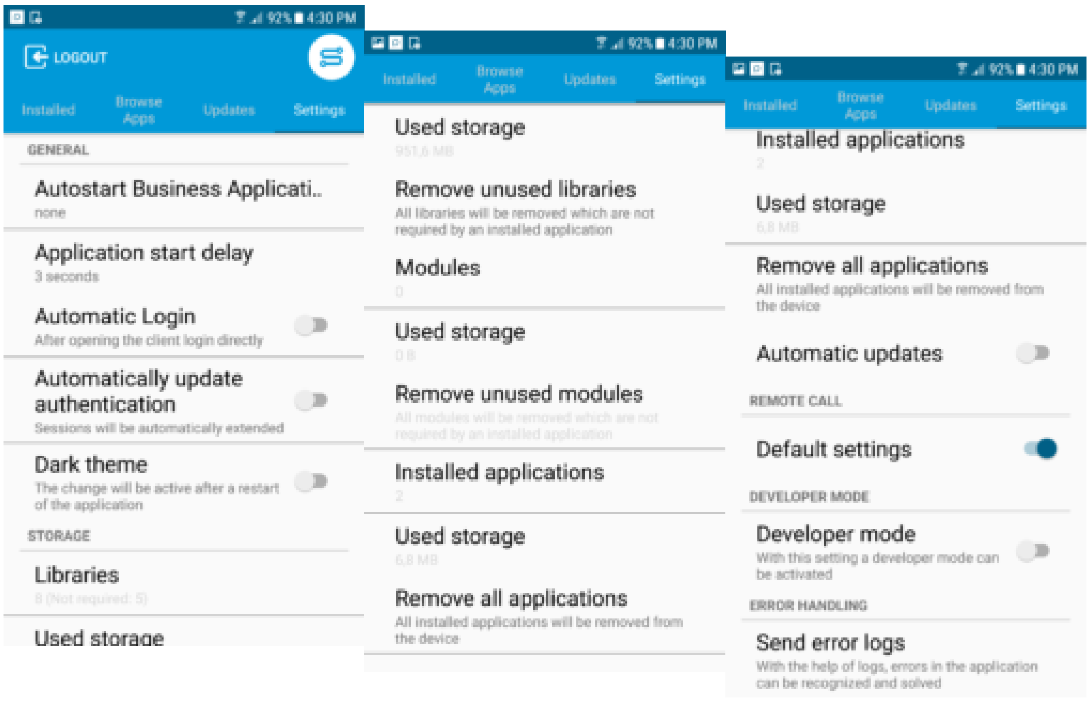
Application Updates

If an app, that you have already installed, has been newly deployed on the instance, you will be informed about updates of the application.


LOGOUT
Installed
Browse Apps
Updates
Settings
MaintenanceDashboard
Smart_Maintenance_Dashboard
ITIZ_SimplifierExplored
ing many Features of Simplifier
ITIZ_Example_SmartMaintenanceDashboard
Smart_Maintenance_Dashboard

Settings

To define the settings, click on the right tab **Settings**.



GENERAL

Autostart Business Applicati...
none

Application start delay
3 seconds

Automatic Login
After opening the client login directly

Automatically update authentication
Sessions will be automatically extended

Dark theme
The change will be active after a restart of the application

STORAGE

Libraries
8 (Not required: 5)

Used storage
951,6 MB

Remove unused libraries
All libraries will be removed which are not required by an installed application

Modules
0

Used storage
0 B

Remove unused modules
All modules will be removed which are not required by an installed application

Installed applications
2

Used storage
6,8 MB

Remove all applications
All installed applications will be removed from the device

Automatic updates

REMOTE CALL

Default settings

DEVELOPER MODE

Developer mode
With this setting a developer mode can be activated

ERROR HANDLING

Send error logs
With the help of logs, errors in the application can be recognized and solved

Section	**Setting**	**Description**
General	Autostart Business Application	Choose an application to start automatically after login (default none).
	Application start delay	Delay the autostart for a number of seconds (default 3 seconds).
	Automatic Login	After opening the client login directly (default off).
	Automatically update authentication	Sessions will be automatically extended (default off).
	Dark theme	Enable the dark theme (default off).
Storage	Libraries	Number of downloaded libraries.
	Used storage	Displays the used storage of libraries.
	Remove unused libraries	Removes unused libraries (it's a button).
	Modules	Number of downloaded modules.
	Used storage	Displays the used storage of modules.
	Remove unused modules	Removes unused modules (it's a button).
	Installed applications	Displays the number of installed applications.
	Used storage	Displays the used storage of applications.
	Remove all applications	Removes all business applications from the device.
	Automatic updated	Updates business application before launch (default off).
Remote Call	Default settings	Uses default remote call settings (default on). Custom settings will be visible if default is off.

	Show camera dialog	Show a camera choose dialog when starting a call (default off).
	Prefer front camera	With this setting remote calls use initially the front camera. Switching between cameras is still possible. (Default off)
	Resolution	Select a maximum target resolution supported by a camera (640x480px).
	Frames per second	Set the frames per seconds to be sent (default 30).
	Maximum video bandwith	Set the maximum video bandwith in kb/s (default 5000 kb/s).
	Maximum audio bandwidth	Set the maximum audio bandwidth in kb/s (default 200 kb/s).
	Video codec	Choose the video codec for the streams (default H264).
	Audio codec	Choose the audio codec for the streams (default OPUS).
Developer Mode	Developer mode	With this setting, the developer mode can be activated (default off).
	Show JavaScript logs	Show a separate view with JavaScript logs (default off).
	Send JavaScript logs	The application will send all JavaScript to the Simplifier instance (default off).

	Transmission interval	The logs are being buffered until the specified time has passed (default 15 min).
	Log level	Specify the minimum level of logs to be sent (default Error).
	Enable Webview debugging	Debug the webview with Chrome Dev Tools (requires adb on your pc)(default depends on activated or deactivated Developer mode).
Error Handling	Send error logs	

Anonymous Profile for Plugins

https://developer.simplifier.io/documentation/plugins/anonymous-profile-plugins/

If you want to access plugins, you can work with anonymous users. Therefore the PluginAPI works with AnonymousAppProfile.

So only the assigned role to your Application needs the permission to use the Plugin.

Read more about roles.

App Links

https://developer.simplifier.io/documentation/getting-started/simplifier-mobile-client/app-links/

Through App Links you can launch specific Simplifier business applications via Deep Links / URL. Use this feature to crosslink different business applications on your mobile device.
App Links can be used with the following URL scheme:

Scheme:

```
simplifierclient:///<action>/<value>?<param>=<value>[&<paramN>=<valueN>]
```

Explanation:

simplifierclient:// - The url type, on that the simplifier client is registered. All uris with this link opens the client by default. If parameters or path components are missing, at least the client is always started.

/<action> - The action to take. For now only "**appDirect**" is available.

/<value> - The value for the action.

?<param>=<value> - The URL arguments are beeing passed to webview so business app can access them. So on the client the local href would be something like
file:///some_very_long_ios_path/www/businessapps/Simplifier_Explored?foo=1&bar=2

Example:

```
simplifierclient:///appDirect/Simplifier_Explored?foo=1&bar=2
```

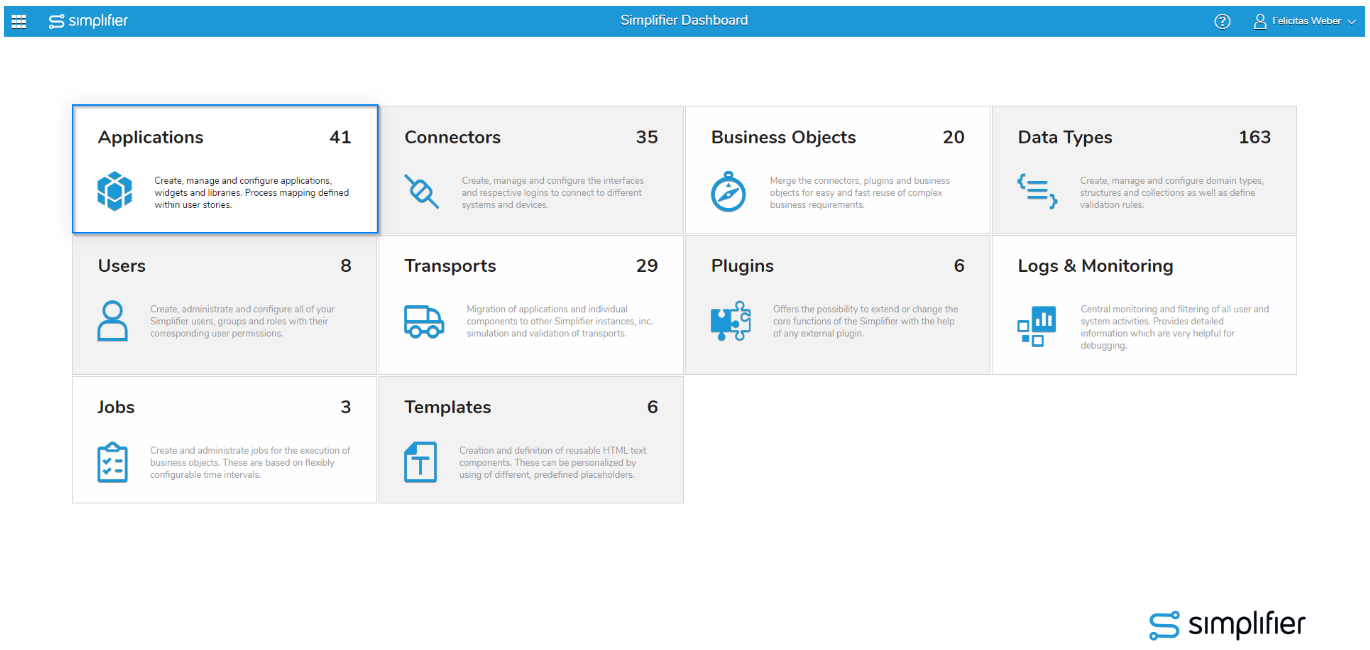
The example above launches the simplifier client if installed and runs the app "Simplifier_Explored".

Restrictions:

- Simplifier Client needs to be installed
- if client is not running, client will be startet and user has to login
- shows popup with countdown when a link was clicked
- url-launch is higher prioritized than automatic app-launch
- if client is already running with a business app, nothing will happen to prevent misbehavoir in app lifecyle
- shows warning if desired app is not installed
- if autoupdate before launch is enabled, the business app will be updated before launch
- broken or non valid links are not beeing processed

Applications

https://developer.simplifier.io/documentation/applications/



Applications 41
Create, manage and configure applications, widgets and libraries. Process mapping defined within user stories.

Connectors 35
Create, manage and configure the interfaces and respective logins to connect to different systems and devices.

Business Objects 20
Merge the connectors, plugins and business objects for easy and fast reuse of complex business requirements.

Data Types 163
Create, manage and configure domain types, structures and collections as well as define validation rules.

Users 8
Create, administrate and configure all of your Simplifier users, groups and roles with their corresponding user permissions.

Transports 29
Migration of applications and individual components to other Simplifier instances, inc. simulation and validation of transports.

Plugins 6
Offers the possibility to extend or change the core functions of the Simplifier with the help of any external plugin.

Logs & Monitoring
Central monitoring and filtering of all user and system activities. Provides detailed information which are very helpful for debugging.

Jobs 3
Create and administrate jobs for the execution of business objects. These are based on flexibly configurable time intervals.

Templates 6
Creation and definition of reusable HTML text components. These can be personalized by using of different, predefined placeholders.

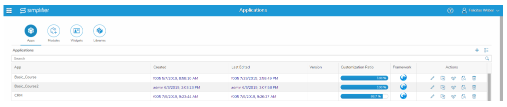
Simplifier transforms your business process into a configured business application for

- Web Portals for Desktop-Browsers
- Mobile Phones and Tablets
- Wearables Devices like Smartwatches and Glasses

Applications run on any device because it is generated on common open standard technology.

Overview

By clicking on the Applications tile, you will be lead to the overview. At default, there you will see a table with all the created applications. Within this table, you'll get information like the name of the app, created by, last edited by, version, customization ratio, framework, and several actions.



App	Created	Last Edited	Version	Customization Ratio	Framework	Actions
Basic_Course	f005 5/7/2019, 8:58:10 AM	f005 7/29/2019, 2:58:49 PM		100 %		
Basic_Course2	admin 6/3/2019, 2:03:23 PM	admin 6/5/2019, 3:07:58 PM		100 %		
CRM	f005 7/9/2019, 9:23:44 AM	f005 7/9/2019, 9:26:27 AM		88.7 %		

If you click on an application in the overview, further information and actions are displayed on the right side. On the one hand, you can edit the app name, switch directly to the application preview, or delete the application. On the other hand, you can customize the description, the app icon, look at the customization ratio and the version. Below this information, you then have various actions, in this case, Copy Application and Show Releases.

App	It's the name of the application.
Created	The name of the person who has created the application (with date and time).
Last Edited	The name of the person who last edited the application (with date and time).
Version	It's the version number of the releases.
Customization Ratio	

Assets

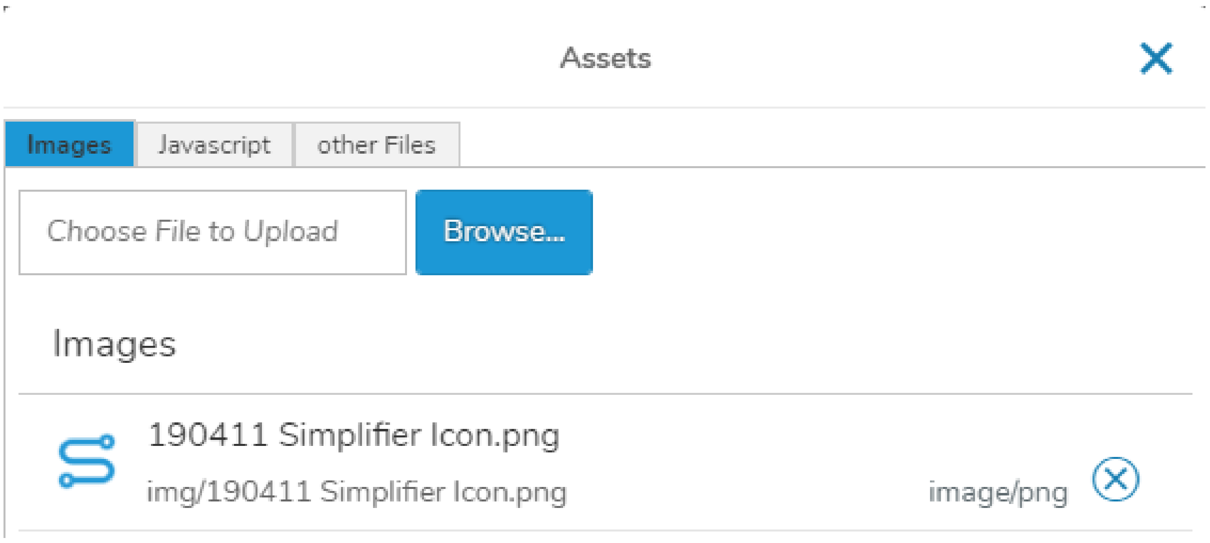
https://developer.simplifier.io/documentation/applications/ui-designer/assets/

To upload files like documents, images, videos, 3D models or office documents to your application, click on the assets button.

You can choose between three different options:

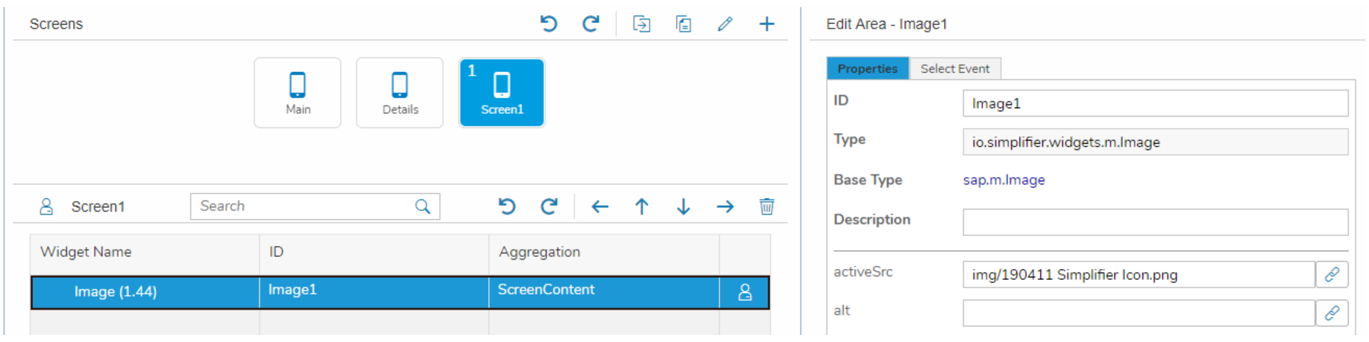
- Images like .PNG, .TIFF, .JPEG or .BMP Files
- Javascript for extending your application with other libraries
- Other files like .PDF Documents, 3D-Models or Media-Files (Audio, Video, etc)

To upload an image, choose it from your client via the upload button - a preview will be generated after uploading and also the path for referencing it later into an parameter of an image widget. In our screenshot the path is *img/Wine.jpg*.
By clicking on the red cross on the right side, you can delete the asset file.



190411 Simplifier Icon.png
img/190411 Simplifier Icon.png
image/png

To insert the assets into your user interface add, an image widget and write the path in the source field (src) in the Edit Area on the right.



Widget Name	ID	Aggregation
Image (1.44)	Image1	ScreenContent

ID	Image1
Type	io.simplifier.widgets.m.Image
Base Type	sap.m.Image
Description	
activeSrc	img/190411 Simplifier Icon.png
alt	

Assign Roles

https://developer.simplifier.io/documentation/connectors/push-connector/assign-roles/

Roles control who receives the messages. Any user who has the corresponding role and uses the Simplifier client at the time of sending a message will receive the message.

Enter the required roles in JSON notation as follows:

```
{
"roles": [
"<Role-1>",
"<Role-2>",
...
"<Role-n>"
]
}
```

Asynchronous Connector Request Json Examples

https://developer.simplifier.io/documentation/connectors/connector-via-script/websocket-communication-with-connectors/asynchronous-connector-request-json-examples/

This section contains the required request data Jsons for different connectors and the description of each individual field.

The following Connectors are described with an example:

- OPC/UA Connector (Monitoring Requests)

Authentication

https://developer.simplifier.io/documentation/admin-settings/authentication-settings/

The **Authentication settings** allow you to establish a connection to external Identity Providers in order to sync external user to the Simplifier.
The following Providers are supported:

LDAP	Active Directory (AD)	SAML 2.0	oAuth 2.0	SAP Single-Sign-On (SSO)

Note: if all authentication systems that are set have been run through and no result has been obtained, a login is executed against the Simplifier User database.

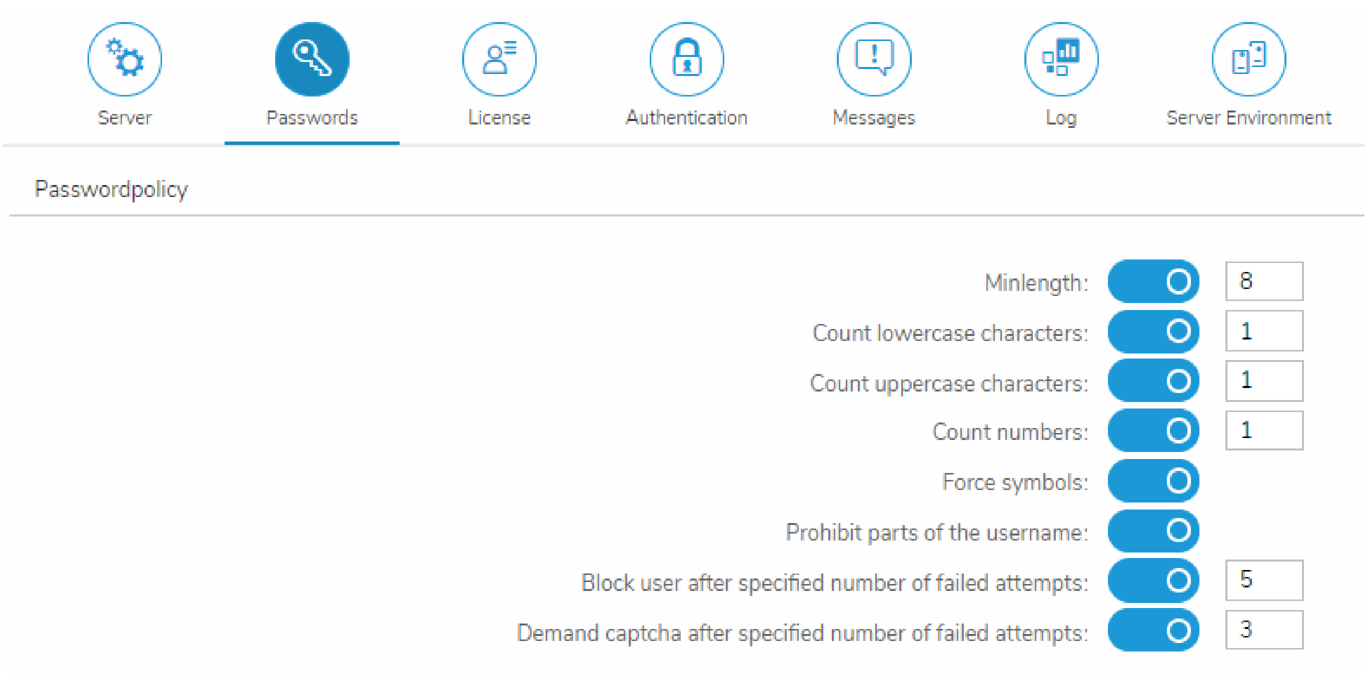
Authentication for Web Applications

https://developer.simplifier.io/documentation/security-guidelines/authentication-for-web-applications/

When authenticating for web applications, you should differentiate between internal and external employees.

Internal employees should only authenticate via single sign-on and internal IDP.

For external employees, you should define password policies and set up a logon configuration.



Passwordpolicy

Minlength:	8
Count lowercase characters:	1
Count uppercase characters:	1
Count numbers:	1
Force symbols:	
Prohibit parts of the username:	
Block user after specified number of failed attempts:	5
Demand captcha after specified number of failed attempts:	3

Automated Testing

https://developer.simplifier.io/documentation/applications/automated-testing/

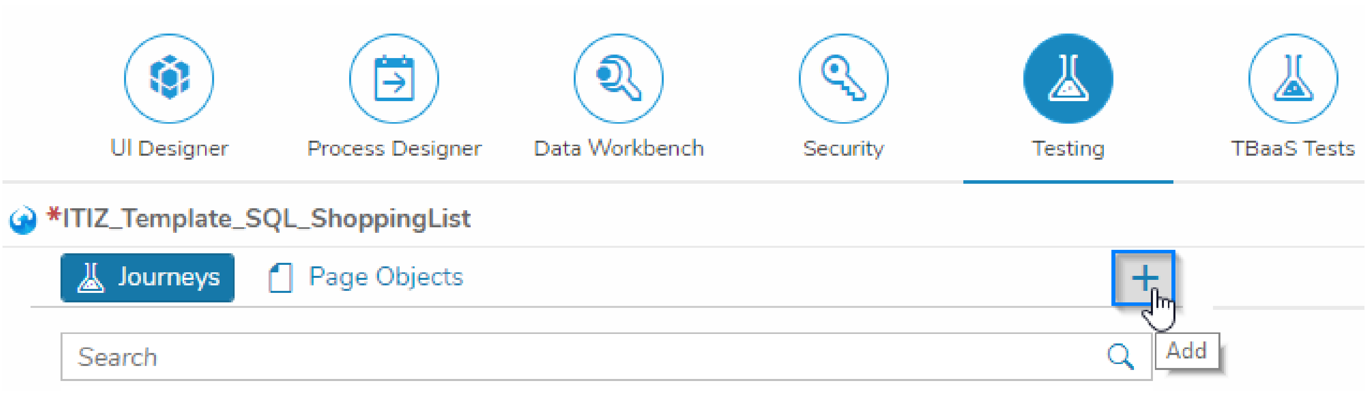
Vimeo Video

On application deployment, the app generator provides a basic self-test for the business application. The automated tests are based on the SAP OPA5 test framework.

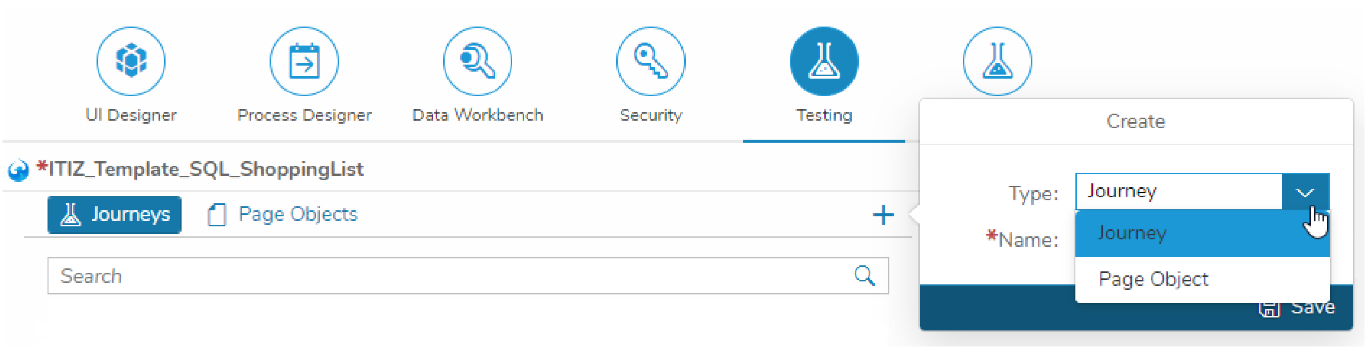
The URL of the test page is relative to the deployed business application used under the subpath /test/integration/opaTests.qunit.html and can be opened with a browser.

As an admin, you can perform an automated test. Make sure you are already in the UI Designer for the testing application. Switch to the tab **Testing**.

To create a new test case, click on the plus icon.

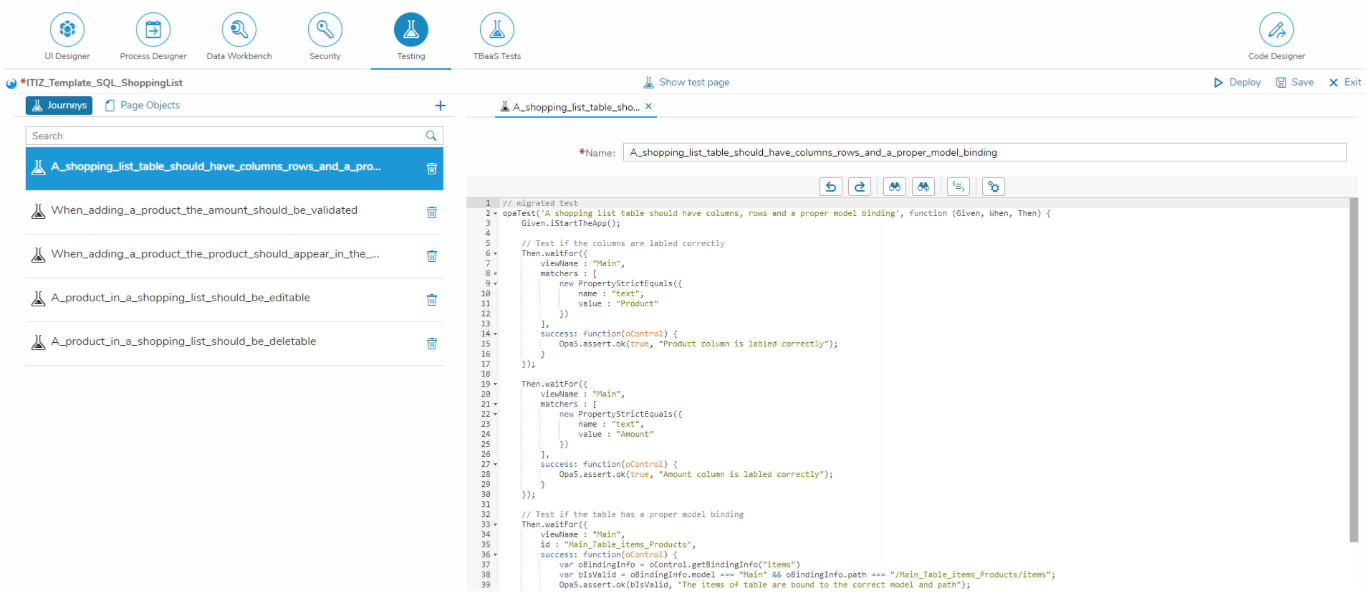


Select the type in the opened pop-up and enter a test case name.



Click on Save.

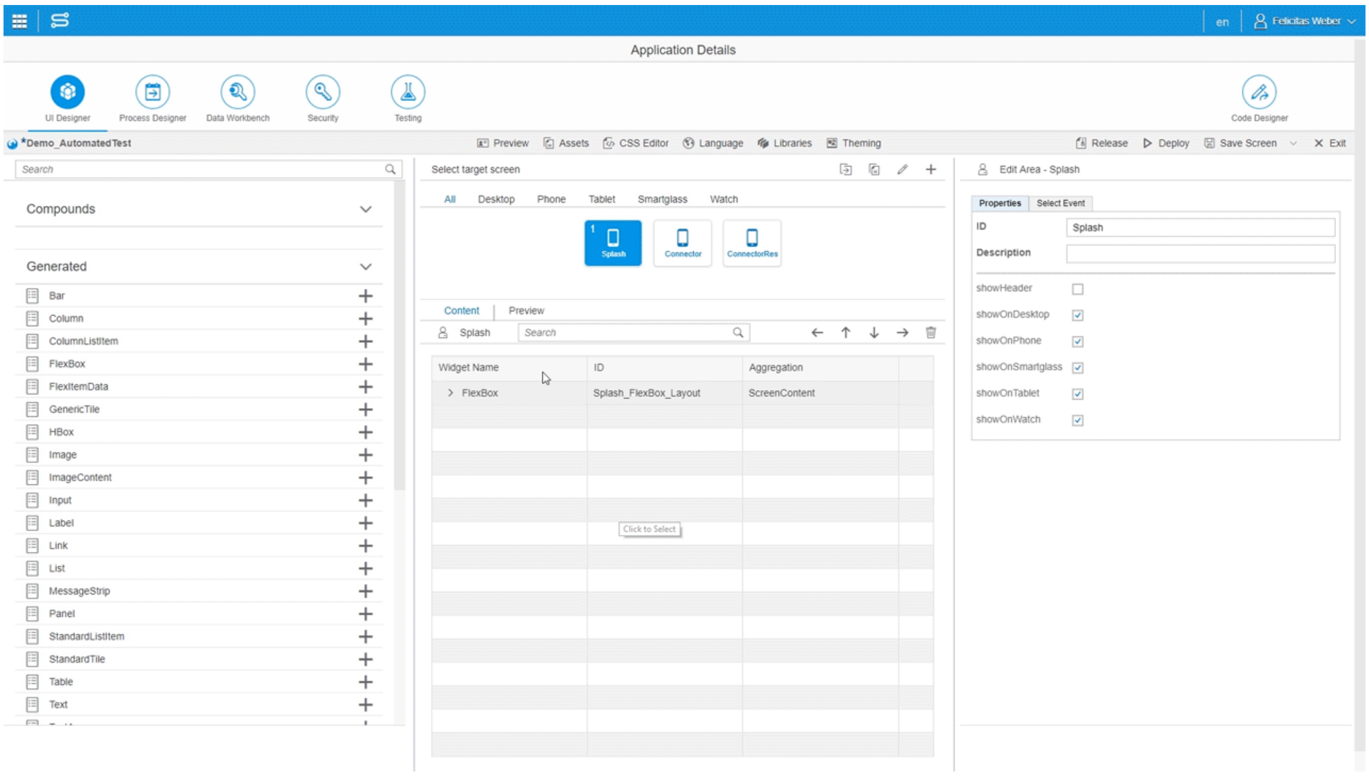
Select the new Journey or Page Object on the left and add the testing code.


UI Designer
Process Designer
Data Workbench
Security
Testing
TBaaS Tests
Code Designer
*ITIZ_Template_SQL_ShoppingList
Show test page
Deploy
Save
Exit
Journeys
Page Objects
Search
A_shopping_list_table_should_have_columns_rows_and_a_pro...
When_adding_a_product_the_amount_should_be_validated
When_adding_a_product_the_product_should_appear_in_the_...
A_product_in_a_shopping_list_should_be_editable
A_product_in_a_shopping_list_should_be_deletable
A_shopping_list_table_sho...

*Name: A_shopping_list_table_should_have_columns_rows_and_a_proper_model_binding

```
// migrated test
opaTest('A shopping list table should have columns, rows and a proper model binding', function (Given, When, Then) {
    Given.iStartTheApp();

    // Test if the columns are labled correctly
    Then.waitFor({
        viewName : "Main",
        matchers : [
            new PropertyStrictEquals({
                name : "text",
                value : "Product"
            })
        ],
        success: function(oControl) {
            Opa5.assert.ok(true, "Product column is labled correctly");
        }
    });

    Then.waitFor({
        viewName : "Main",
        matchers : [
            new PropertyStrictEquals({
                name : "text",
                value : "Amount"
            })
        ],
        success: function(oControl) {
            Opa5.assert.ok(true, "Amount column is labled correctly");
        }
    });

    // Test if the table has a proper model binding
    Then.waitFor({
        viewName : "Main",
        id : "Main_Table_items_Products",
        success: function(oControl) {
            var oBindingInfo = oControl.getBindingInfo("items")
            var bIsValid = oBindingInfo.model === "Main" && oBindingInfo.path === "/Main_Table_items_Products/items";
            Opa5.assert.ok(bIsValid, "The items of table are bound to the correct model and path");
```

Deploy the Journeys or Page Objects and click on **Show test page**.



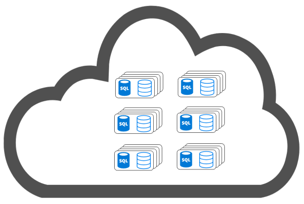
Widget Name	ID	Aggregation
FlexBox	Splash_FlexBox_Layout	ScreenContent

ID	Splash
Description	
showHeader	☐
showOnDesktop	✓
showOnPhone	✓
showOnSmartglass	✓
showOnTablet	✓
showOnWatch	✓

Backups

https://developer.simplifier.io/documentation/installation-instructions/simplifier-cloud/backups/

Every instance is backed up daily in the Simplifier Cloud.


Both files and a logical database backup (dump) are stored directly on the machine. These are held locally for 4 weeks. This is very useful when restoring a single Simplifier instance.
Furthermore, the Simplifier Cloud is image-based backed up every day. These backups are held for 14 days.
Should the system fail completely, we can initiate a complete restore at any time.

Type of backup	Backup interval	How long are the backups stored?
tarball of files and a logical database backup	daily	4 weeks
image-based backup of the whole Simplifier Cloud	daily	2 weeks

Basic Concept / Technology

https://developer.simplifier.io/documentation/applications/basic-concept-technology/

All Simplifier applications are based on OpenUI5.

SAPUI5 and its open-source variant OpenUI5 help you build enterprise-ready Web apps that are responsive to all devices. The JavaScript UI library and development toolkit contains many feature-rich controls and implements the award-winning SAP Fiori user experience. It helps developers ease and speeds up the development of full-blown HTML5 Web applications.

The Simplifier App Generator generates OpenUI5 Applications based on OpenUI5 Controls. Within Simplifier OpenUI5 Controls are represented by Widgets.

User Interface

To create the user interface of Simplifier applications visually the UI Designer is used. In general, OpenUI5 uses pages to represent views within a single-page-app. Within Simplifier UI Designer, pages are represented by screens. Simplifier applications consist of one or more screens and every screen can be populated with widgets.

Application Logic

Within Simplifier application logic is separated by user stories within the Process Dashboard. Every user story contains an isolated part of the overall application logic and can be edited with the Process Designer. The Process Designer is a visual scripting environment to create application logic based on configuration elements. To find out how to use configuration elements see chapter Process Designer.

How OpenUI5 concepts are handled within Simplifier

OpenUI5	Simplifier	Description
Pages (Views)	Screens	OpenUI5 pages are represented by screens within Simplifier. A screen collects several widgets in a specific order to represent the user interface.
Controls	Widgets	OpenUI5 controls are represented by widgets within Simplifier. A widget represents a specific element in the user interface like buttons, checkboxes, tables and input fields.
View Controller	Screen Controller/User Story	OpenUI5 view controllers are represented by screen controllers/user stories. There is an n:n relation between

		user stories and controllers.
Models	Screen Models and variableHolder Model	Within Simplifier there is a global model named variableHolder and each screen has its own model named by the screenId.

Basic Protection of Internet Access

https://developer.simplifier.io/documentation/security-guidelines/basic-protection-of-internet-access/

To ensure the security of the Simplifier instance, you should get familiar with the paths in the table below.

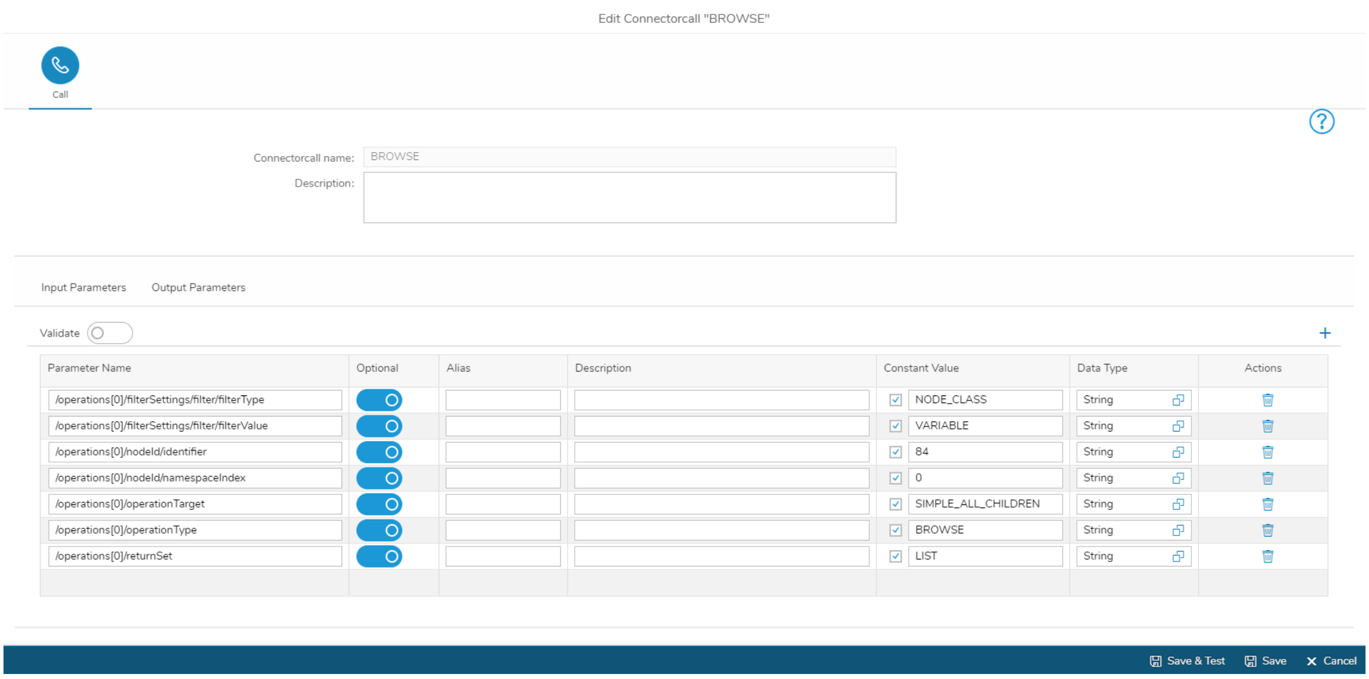
These paths, with the exception of /UserInterface/, should be accessible to the **application user**.

Location / Path	**Description**
"^/genToken/$"	The Simplifier Authentification Service based on Tokens
"^/assets/(.*)$"	The static assets like images, pdf files, etc for an Application
"^/client/(.*)$"	The Client REST API to access business objects, connector or plugins
"^/library-managed/(.*)$"	Third-Party Javascript Libraries that need for the HTML5 Applications
"^/library-static/(.*)$"	Third-Party Javascript Libraries that need for the HTML5 Applications
"^/appDirect/(.*)$"	Hosting Path for the created HTML5 Applications
"^/UserInterface/(.*)$"	

BROWSE Call - OPC/UA Connector

https://developer.simplifier.io/documentation/connectors/opcua-connector-details/opc-ua-connector-calls/browse-call/

Call for BROWSE operations (the name TIA_BROWSE_ALL_VARIABLES is the arbitrarily chosen name for this call)


Edit Connectorcall "BROWSE"

Connectorcall name: BROWSE
Description:

Validate

Parameter Name	Optional	Alias	Description	Constant Value	Data Type	Actions
/operations[0]/filterSettings/filter/filterType	●			✓ NODE_CLASS	String	
/operations[0]/filterSettings/filter/filterValue	●			✓ VARIABLE	String	
/operations[0]/nodeId/identifier	●			✓ 84	String	
/operations[0]/nodeId/namespaceIndex	●			✓ 0	String	
/operations[0]/operationTarget	●			✓ SIMPLE_ALL_CHILDREN	String	
/operations[0]/operationType	●			✓ BROWSE	String	
/operations[0]/returnSet	●			✓ LIST	String	

Input Parameter

For the Browse connector call, you need to configure the "**operationType** " and the "**nodeId**" (consisting of 2 parameter: **identifier** and **namespaceIndex**). Furthermore, you need to define the **operationTarget**,
a **returnSet** and **filterSettings** (optional).

operationType: Defines which operation you want to execute, in this case, "BROWSE".
Parameter Name: operations/arrayItem[0]/operationType
Constant Value: BROWSE
Data Type: String

---

nodeID: Defines the identification of the OPC/UA node. It is split in 2 parameter:

- **Identifier**:
  Parameter Name: operations[0]/nodeId/identifier
  Data Type: String or Numeric
- **NamespaceIndex**:
  Parameter Name: operations[0]/nodeId/namespaceIndex
  Data Type: String
  In every namespace, each ID must be unique (it is possible to use the String "7617" and the Numeric 7167 together in one namespace)
- **identifierType (optional)**: Searches for the Identifier with a fixes Data Type.
  Parameter Name: operations[0]/identifierType
  Constant Value: Numeric, UUID, String, Byte String

---

operationTarget: You can browse references forward, backward or in both directions. Choose between the basic attributes (simple) or further ones, depending on the class (extended).
Parameter Name: operations/arrayItem[0]/operationTarget
Data Type: String
Constant Value: Choose between

- SIMPLE_ALL_CHILDREN
- SIMPLE_ALL_PARENTS
- SIMPLE_BOTH
- EXTENDED_ALL_CHILDREN
- EXTENDED_ALL_PARENTS
- EXTENDED_BOTH

returnSet:
Parameter Name: operations[0]/returnSet
Data Type: String
Constant Value: LIST

filterSettings (optional):

- **Type**:
  Parameter Name: operations[0]/filterSettings/filter/filterType
  Data Type: String
  Constant Value: NODE_CLASS
- **Value**:
  Parameter Name: operations[0]/filterSettings/filter/filterValue
  Data Type: String
  Constant Value: Choose between

    - DATA_TYPE
    - METHOD
    - OBJECT
    - OBJECT_TYPE
    - REFERENCE_TYPE
    - VARIABLE
    - VARIABLE_TYPE
    - VIEW
    - UNSPECIFIED

NOTE: The specific commands are NOT defined here!

Output parameters

You can return all output parameter like this:

Parameter Name: /
Data Type: String

If you want to get only selected output parameter, use the following syntax:

Parameter Name: operationsResult/[0]/browseResult/children/nodes/
Data Type: depends on the parameter you want to be returned.

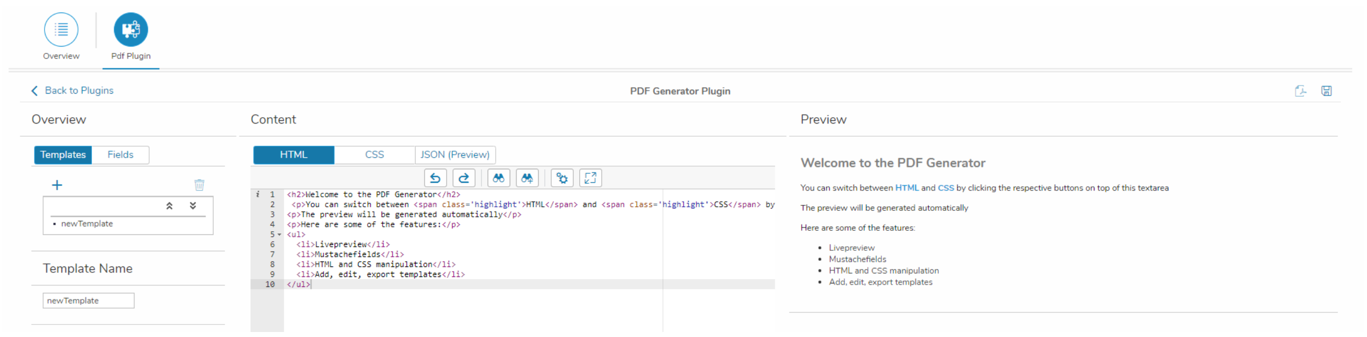
For now only the complete unformatted JSON will be returned.

Build a PDF Template

https://developer.simplifier.io/documentation/plugins/pdf-plugin/built-pdf-template/

Administration

To generate a PDF and manage templates, the role "pdfPlugin" has to be assigned to your user.



```
<h2>Welcome to the PDF Generator</h2>
 <p>You can switch between <span class='highlight'>HTML</span> and <span class='highlight'>CSS</span> by
<p>The preview will be generated automatically</p>
<p>Here are some of the features:</p>
<ul>
  <li>Livepreview</li>
  <li>Mustachefields</li>
  <li>HTML and CSS manipulation</li>
  <li>Add, edit, export templates</li>
</ul>
```

Welcome to the PDF Generator

You can switch between HTML and CSS by clicking the respective buttons on top of this textarea

The preview will be generated automatically

Here are some of the features:

- Livepreview
- Mustachefields
- HTML and CSS manipulation
- Add, edit, export templates

Templates

You can build a PDF Template by using HTML, CSS and JSON. A live preview is provided on the right, so you can see changes in real-time.

The rendering is executed with wkhtmltopdf, therefore every HTML format and feature that supports the QT Webkit render engine is working.

With every template, a stylesheet in LESS format is generated and will be embedded automatically. You can maintain this stylesheet via the same interface as the HTML template.

The inclusion of graphics (**<img src="...">**) and additional stylesheets (**<link rel="stylesheet" href="...">**) is also supported. These external asserts are retrieved via the "assets" slot of the AppServer (they should be uploaded there in advance). You can refer to them in the template with a relative filename (no "http://" prefix, no path, etc.!).
Example: **<img src="image.jpg">** (if the file was uploaded as "image.jpg")

Furthermore, you can add expressions in mustache format. These "variables" are later replaced by values from the update file to a session.

The dynamic data is retrieved as a JSON string in the key-value-store with the key: "**sessiondata/$session**".
($session = the session ID that is specified for the generation)

Merging

You can combine your PDF document with other PDFs or images from the key-value store.

For this purpose, you can call the list of all the resources you want to merge with the key "**merge/$session**" in the key-value store. The list should correspond to a JSON-Array, where the entries of the JSON-Array are the keys of the resources to be merged. For example: **["document1.pdf", "document2.pdf", "image.jpg"]**.

The binary data of the corresponding documents should be filed in the key-value store under the keys "document2.pdf", "document2.pdf" and "image.jpg".

If the list of merge resources is not found for a session or if the list is empty, the merge is skipped.

Saving the generated PDF

After a PDF has been successfully generated, the binary data is stored in the Key-Value Store under the key "**pdf/$jobid.pdf**".
($jobid = the job ID, that will be returned after the generation has started)

If the generation can not be executed successfully due to an error, a fault reporting is stored under the key "pdf/$jobid.log" in the key-value store.

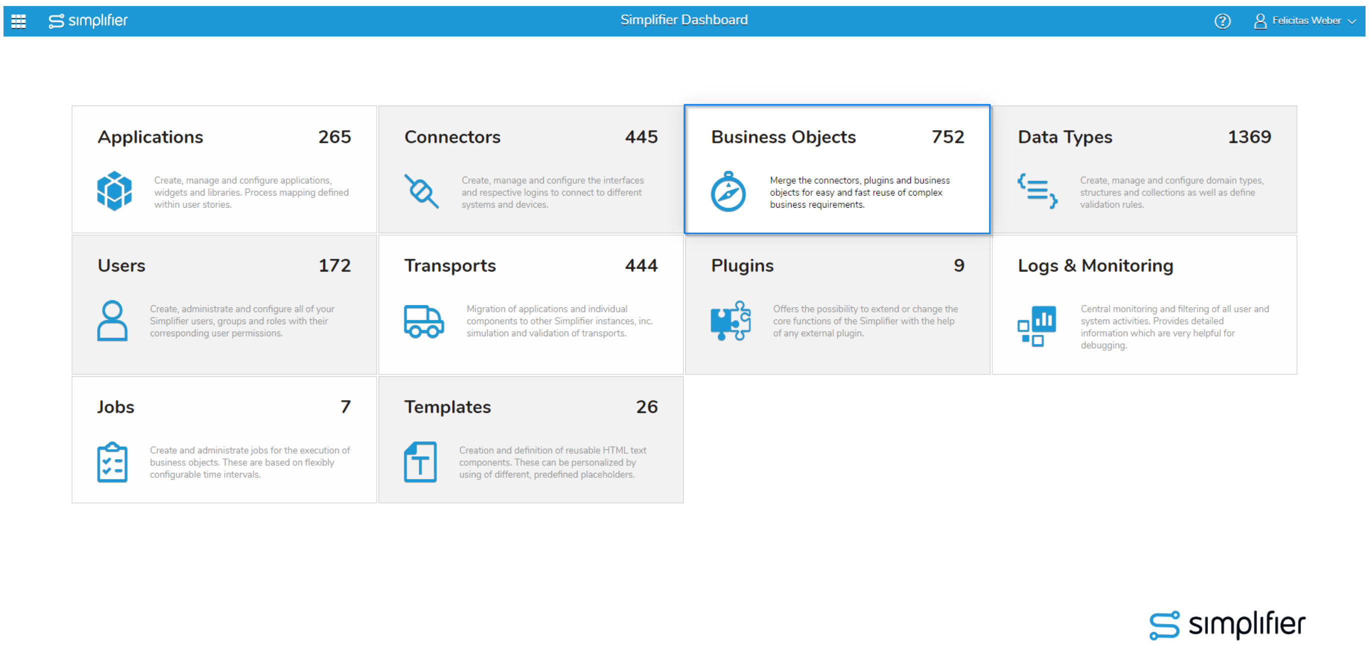
Business Objects

https://developer.simplifier.io/documentation/business-objects/

Vimeo Video

Simplifier allows you to create complex integrated applications up to a high degree solely through configuration. Nevertheless, at some point in time, advanced business logic might be required, which can’t be implemented merely by configuration. This is when Business Objects come into play.

Business objects are implemented via JavaScript. This way they integrate seamlessly into Simplifier applications. They allow you to write arbitrary business logic and interact with other Simplifier artifacts like connectors, plugins or other business objects. They can also be used among different applications.



Applications 265
Create, manage and configure applications, widgets and libraries. Process mapping defined within user stories.

Connectors 445
Create, manage and configure the interfaces and respective logins to connect to different systems and devices.

Business Objects 752
Merge the connectors, plugins and business objects for easy and fast reuse of complex business requirements.

Data Types 1369
Create, manage and configure domain types, structures and collections as well as define validation rules.

Users 172
Create, administrate and configure all of your Simplifier users, groups and roles with their corresponding user permissions.

Transports 444
Migration of applications and individual components to other Simplifier instances, inc. simulation and validation of transports.

Plugins 9
Offers the possibility to extend or change the core functions of the Simplifier with the help of any external plugin.

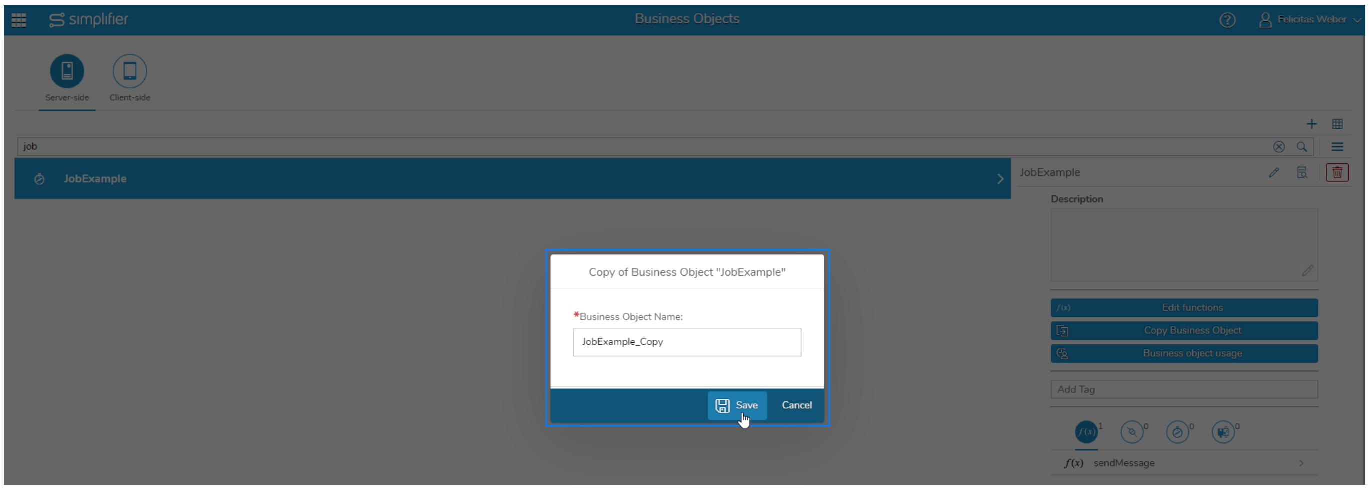
Logs & Monitoring
Central monitoring and filtering of all user and system activities. Provides detailed information which are very helpful for debugging.

Jobs 7
Create and administrate jobs for the execution of business objects. These are based on flexibly configurable time intervals.

Templates 26
Creation and definition of reusable HTML text components. These can be personalized by using of different, predefined placeholders.

To copy a business object, just click in the overview of business objects on the 'Copy Business Object' button on the right of the selected entry.

After you have clicked on it, a pop-up appears in which you can specify the new name of the business object. Then click on 'Save'.



Copy of Business Object "JobExample"

*Business Object Name: JobExample_Copy

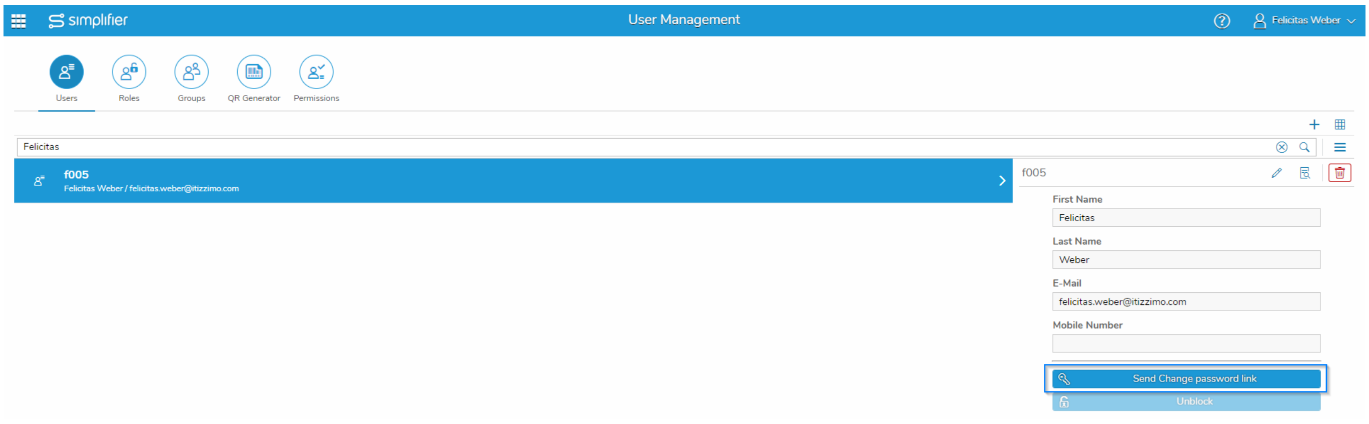
Now the business object has been copied. All included connectors, plugins and other business objects, as well as the script templates, are available in the copy.

Change your Password

https://developer.simplifier.io/documentation/user-management/change-your-password/

For security reasons, it is always a good idea to update your password regularly.

In order to change your password, you have to switch to the '**Users**' tile in the Simplifier dashboard. After that, search for the user whose password you want to change and click on **Send Change password link** on the right side.



Felicitas

f005
Felicitas Weber / felicitas.weber@itizzimo.com

f005

First Name
Felicitas

Last Name
Weber

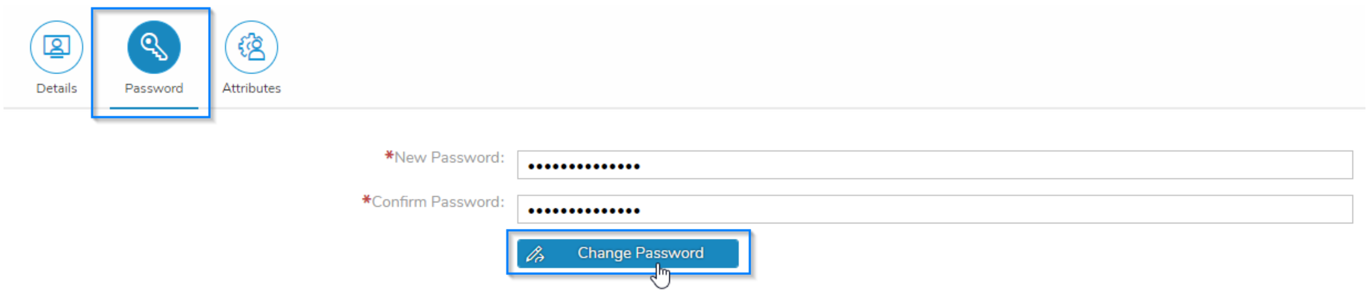
E-Mail
felicitas.weber@itizzimo.com

Mobile Number

You will receive an email with a link to change your password.

If you need help, please contact an admin.

If you're an admin and want to change someones password, click on the ‚**Password**‘ tab in the upper left corner. Now all you have to do is enter the new password, confirm it and finally click on ‚**Change Password**‘.


Details
Password
Attributes
*New Password:
*Confirm Password:
Change Password

Checklist - Simplifier onPremise Installation

https://developer.simplifier.io/documentation/installation-instructions/on-premise/checklist-simplifier-onpremise-installation/

Here you will find a checklist for all On Premise installations. You can check off the points when you've finished them.
During this time, please do not reload this page.

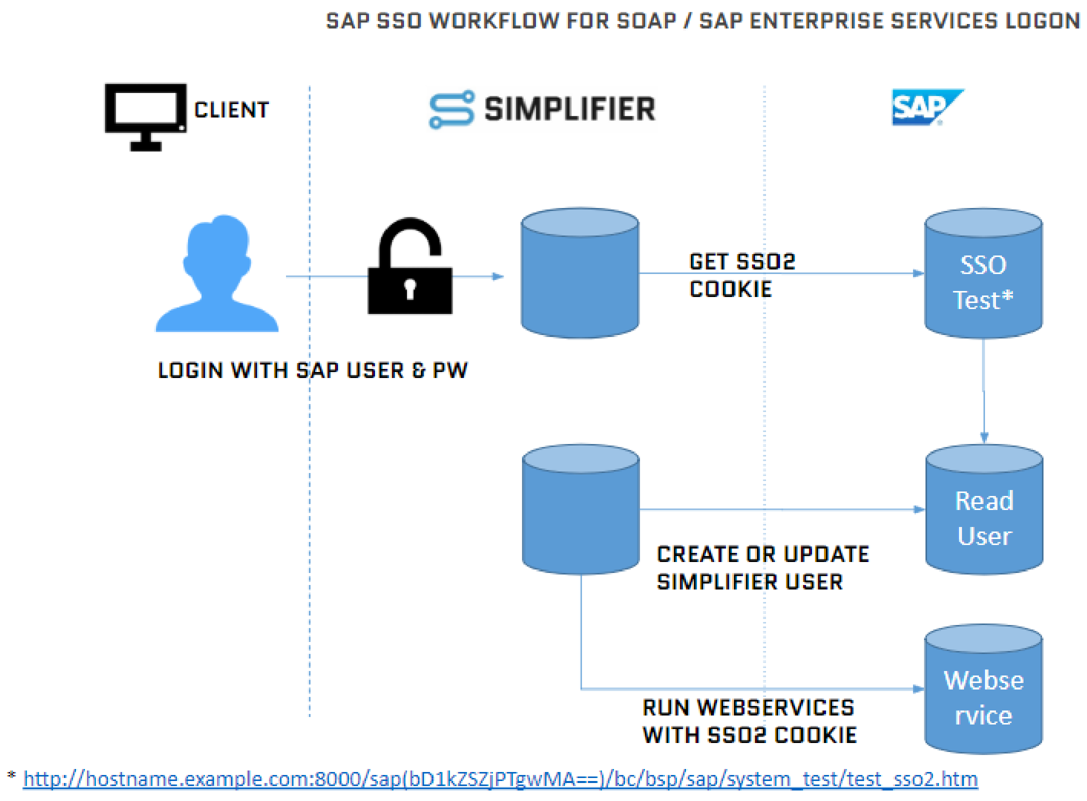
Have a FQDN (Fully-Qualified Domain Name) for each instance of the D (Development) Q (QA /Test System) P (Productive) System
Set the 3 DNS entries for the 3 FQDN

Make sure that the Simplifier server reaches the Internet and make sure that the clients have access to the required ports of the Simplifier: 80 (TCP), 443 (TCP), 8090 (TCP)
Install the latest version of docker engine

Create or specify the Simplifier workspace, set the environment variables, and ensure that enough space is available. Also for future updates. We recommend at least 60 GB Storage.
Use the image from the Docker Hub “simplifierag/onpremise:latest” or use the image provided by our Infrastructure Team
Provide the SSL Certificate and Intermediate Certificate. Best practice is a globally valid certificate issued by a trusted certification authority.
If you do not want to use the database already provided in the on-premise Docker, set up an external database for the Simplifier Core platform

Checklist SAP SSO over SOAP

https://developer.simplifier.io/documentation/admin-settings/authentication-settings/sap-sso/checklist-sap-sso-over-soap/


SAP SSO WORKFLOW FOR SOAP / SAP ENTERPRISE SERVICES LOGON
CLIENT
SIMPLIFIER
SAP
LOGIN WITH SAP USER & PW
GET SSO2 COOKIE
SSO Test*
Read User
CREATE OR UPDATE SIMPLIFIER USER
Webservice
RUN WEBSERVICES WITH SSO2 COOKIE
* http://hostname.example.com:8000/sap(bD1kZSZjPTgwMA==)/bc/bsp/sap/system_test/test_sso2.htm

Check 1: SSO2 Check

1. Start Transaction SE80
2. Choose Type BSP Application
3. Choose SYSTEM_TEST/test_sso2.htm
4. Test/Run (F8)
   http://hostname.example.com:8000/sap(bD1kZSZjPTgwMA==)/bc/bsp/sap/system_test/test_sso2.htm
5. Check if Cookie ‚MYSAPSSO2=…‘ available

Check 2: SSO Parameter

1. Run transaction code RZ11(temporary) RZ10 (permanent)
2. Check if the following parameter has been set

login/accept_sso2_ticket	1
login/create_sso2_ticket	2 (without certificate)

Check 3: SSO Login

1. Open transaction SA38
2. Choose report SEC_TRACE_ANALYZER

Check 4: Permissions

Every user needs the following permission object:

S_SERVICE

Attributes	**Values**
SRV_NAME	Name of Webservice
SRV_TYPE	Type of Webservice (HS)

Troubleshooting / Common Errors & Solutions

The following section documents the most common errors with possible solutions.

Q: What should I do when HTTPS/SSL is not available?

A: If you have problems with the connection set it from SSL to None

Q: What if the WSDL Consumer has problems parsing the WSDL?

A: Manually replace the string ws_policy in the WSDL with standard

Q: How can I monitor the error log of SAP Web services?

A: The error log can be viewed with transaction "srt_util".

Q: How can I change the SAP web service login language?

A: The standard login language is also via SAP Webservices in English. Thus, all data determinations according to e.g.: Status texts, material text ect. always return in English language.

To be able to change it to German, the following prefix must be appended to the **SOAP Webservice operation URL**: "?sap-language=DE"
This does NOT mean the WSDL URL!

Q: How can I call the web service from another SAP client?

A: The **web service operation call** must be done with the parameter?sap-client=[client] so that the system can recognize the client.

Q: What if the Simplifier does not have access to the SAP system?

A: Check the following points:

Please make sure that there is a physical connection between the Simplifer (host) instance and the system. Firewall/Ports may need to be enabled to allow communication in both directions.
The Simplifier Docker or host system must be maintained with the correct network settings for on premise installations. This includes, for example, the setting for DNS servers.

Client-Side - Access Connectors

https://developer.simplifier.io/documentation/business-objects/create-client-side-business-object/client-side-access-connectors/

To access a connector using your business object, you must first add the connector to it.

```
var lfx_success = function(data) {
console.log(data)
};
var lfx_error = function(data) {
console.log(data)
};
var lb_showBusyIndicator = true;
var lb_failOnError = true;
Simplifier.Connector.GIS.getGisDivision({}, lfx_success, lb_showBusyIndicator, lb_failOnError, lfx_error)
```

Client-Side - Access other Business Objects

https://developer.simplifier.io/documentation/business-objects/create-client-side-business-object/client-side-access-other-business-objects/

To access other business objects using your business object, you must first add them to your current business object.

Access Server-Side Business Object

Simplifier.BusinessObject.ContentRepository.contentFolderEdit({}, lfx_success, lb_showBusyIndicator, lb_failOnError, lfx_error

Access Client-Side Business Object

Simplifier.ClientsideBusinessObject.ClientSideBO.getData({}, lfx_success, lb_showBusyIndicator, lb_failOnError, lfx_error)

Client-Side - Access Plugins

https://developer.simplifier.io/documentation/business-objects/create-client-side-business-object/client-side-access-plugins/

To access a plugin using your business object, you must first add the plugin to it.

```
var lfx_success = function(data) {
console.log(data)
};
var lfx_error = function(data) {
console.log(data)
};
var lb_showBusyIndicator = true;
var lb_failOnError = true;
Simplifier.Plugin.contentRepoPlugin.contentFileEdit({}, lfx_success, lb_showBusyIndicator, lb_failOnError, lfx_error)
```

Client-Side Business Object API

https://developer.simplifier.io/documentation/business-objects/create-client-side-business-object/client-side-business-object-api/

You can access any methods of the Simplifier by using the Simplifier Object.

Connectors

```
Simplifier.Connector.<ConnectorName>(payload: object, successCallback: function, busy
Flag?: boolean, failOnError?: boolean, errorCallback?: function): void
Simplifier.Connector.<ConnectorName>.<CallName>(payload: object, successCallback: fun
ction, busyFlag?: boolean, failOnError?: boolean, errorCallback?: function): void
```

Example:

```
var payload = {bindingName: "Binding", operationName: "MyOp", soap: {foo: "bar"}};
function onSuccess (data) { resolve(data); };
Simplifier.Connector.MySoap(payload, onSuccess, true, true);
Simplifier.Connector.MySoap.myCall(payload, onSuccess, true, false, function () { con
sole.log("something went wrong"); });
```

Business Objects

```
 Simplifier.BusinessObject.<BOName>.<MethodName>(payload: object, successCallback: fu
nction, busyFlag?: boolean, failOnError?: boolean, errorCallback?: function, parametr
ized?: boolean = true): void
```

Example:

```
var payload = {leftOperand: 3, operation: "add", rightOperand: 4};
function onSuccess (data) { resolve(data); };
Simplifier.BusinessObject.OtherBO.someMethod(payload, onSuccess, true, false, functio
n () { console.log("something went wrong"); }, true);
```

Client-side Business Objects

```
Simplifier.ClientsideBusinessObject.<CSBOName>.<FunctionName>(payload: object, succes
sCallback: function, busyFlag?: boolean, failOnError?: boolean, errorCallback?: funct
ion): void
```

```
Simplifier.CurrentClientsideBusinessObject.<FunctionName>(payload: object, successCal
lback: function, busyFlag?: boolean, failOnError?: boolean, errorCallback?: function)
: void
```

Example:

```
var payload = {leftOperand: 3, operation: "add", rightOperand: 4};
function onSuccess (data) { resolve(data); };
Simplifier.ClientsideBusinessObject.OtherBO.someMethod(payload, onSuccess, true, fals
e, function () { console.log("something went wrong"); });
Simplifier.CurrentClientsideBusinessObject.someMethod(null, onSuccess, true, false, f
unction () { console.log("something went wrong"); });
```

Plugins

```
Simplifier.Plugin.<PluginName>.<SlotName>(payload: object, successCallback: function,
 busyFlag?: boolean, failOnError?: boolean, errorCallback?: function): void
```

Example:

```
var payload = {name: ""};
function onSuccess (data) { resolve(data); };
Simplifier.Plugin.contentRepoPlugin.listRepos(null, onSuccess);
Simplifier.Plugin.contentRepoPlugin.createRepo(payload, onSuccess);
```

CryptoJS

```
var sMySecretKey = "secret";
var oCrypted = CryptoJS.AES.encrypt("dontStealMyData", sMySecretKey);
output.result = CryptoJS.AES.decrypt(oCrypted, sMySecretKey).toString(CryptoJS.enc.Ut
f8)
```

Take also a look at crypto-js.

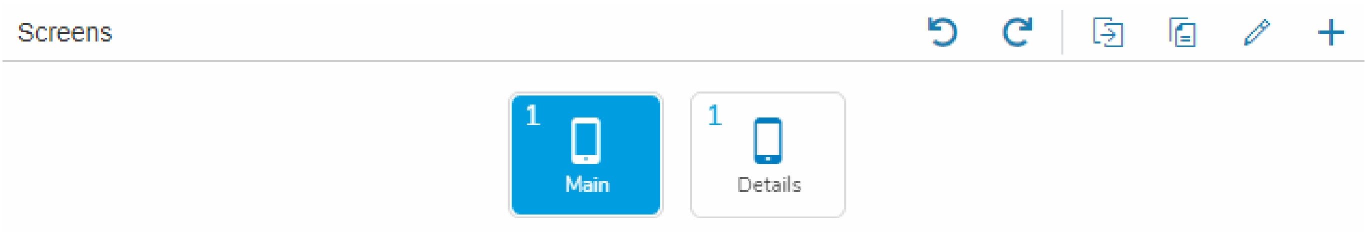
Code Designer

https://developer.simplifier.io/documentation/applications/code-designer/

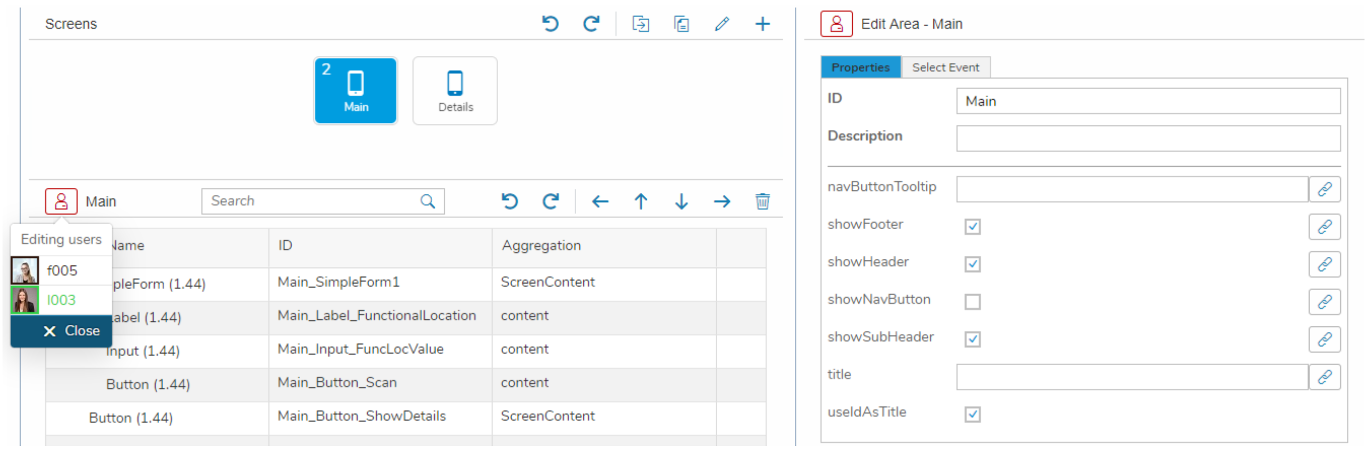
Collaboration

https://developer.simplifier.io/documentation/applications/ui-designer/collaboration/

It is possible to see if another user is editing the screens within the application. The number on the top left of the screen tile represents the number of editors on each screen.



As soon as another user starts editing the same screen, the color of the button (in the upper right as well as in the content area) changes to red. To show all editors, click on the button to open a popover with the editing users.

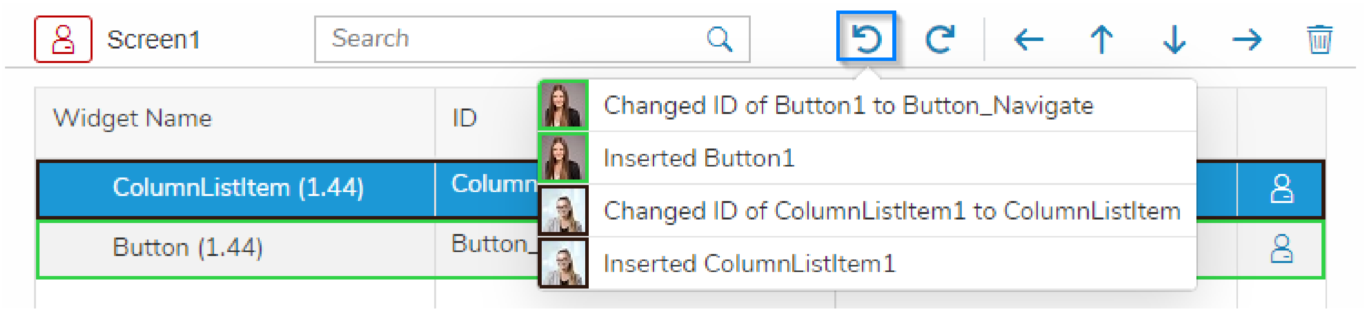


Name	ID	Aggregation
pleForm (1.44)	Main_SimpleForm1	ScreenContent
abel (1.44)	Main_Label_FunctionalLocation	content
Input (1.44)	Main_Input_FuncLocValue	content
Button (1.44)	Main_Button_Scan	content
Button (1.44)	Main_Button_ShowDetails	ScreenContent

ID	Main
Description	
navButtonTooltip	
showFooter	✓
showHeader	✓
showNavButton	☐
showSubHeader	✓
title	
useIdAsTitle	✓

The list of editors is always up to date, so you get an immediate response if someone else starts editing the screen.

Due to the autosave function, there are possibilities to make changes undo or redo. That means, that user actions within the screen content and properties can be undone by clicking on the undo-icon.



Changed ID of Button1 to Button_Navigate
Inserted Button1
Changed ID of ColumnListItem1 to ColumnListItem
Inserted ColumnListItem1

The undo-list offers the last actions on the current screen, starting with the last one. If you select an entry from the list, it will be undone including its subsequent actions.

An avatar screen of the corresponding user is displayed within the list. In the case of collaboration, it becomes obvious that the operation will eventually undo the work of another user.

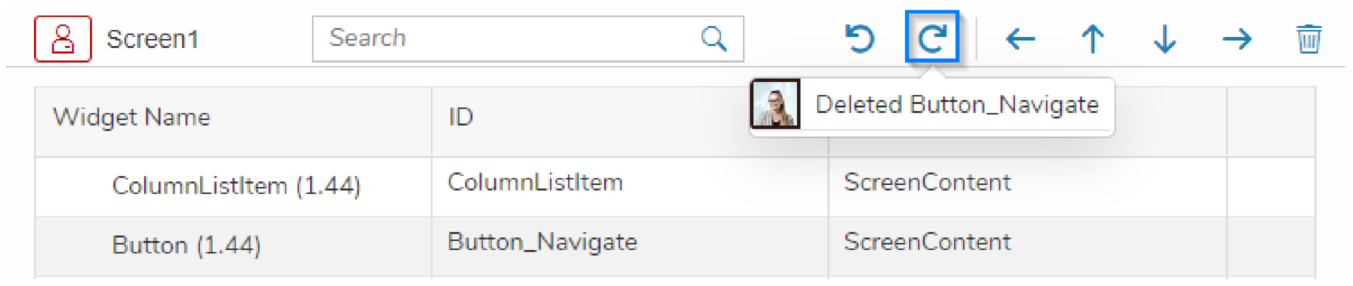
The following actions are listed:

- edit the screen properties
- add and remove widgets (screen elements)
- move widgets in the element tree
- edit widget properties, including ID, data aggregation, validations and events

Please note:

The undo lists at the application level are only retained as long as a user is active in the UI Designer. When the last user leaves the application, the lists and all deleted screens are permanently deleted.

The redo-icon provides recently undone actions to redo. The list is cleared when regular editing takes place.



Deleted Button_Navigate

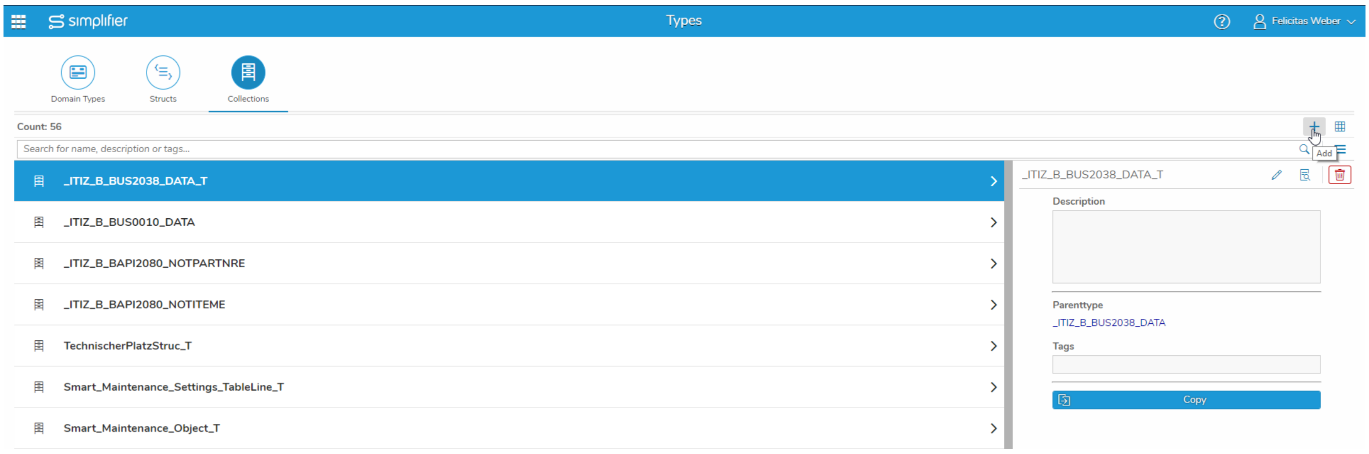
Widget Name	ID		
ColumnListItem (1.44)	ColumnListItem	ScreenContent	
Button (1.44)	Button_Navigate	ScreenContent	

Collection Type

https://developer.simplifier.io/documentation/data-types/create-edit-a-collection-type/

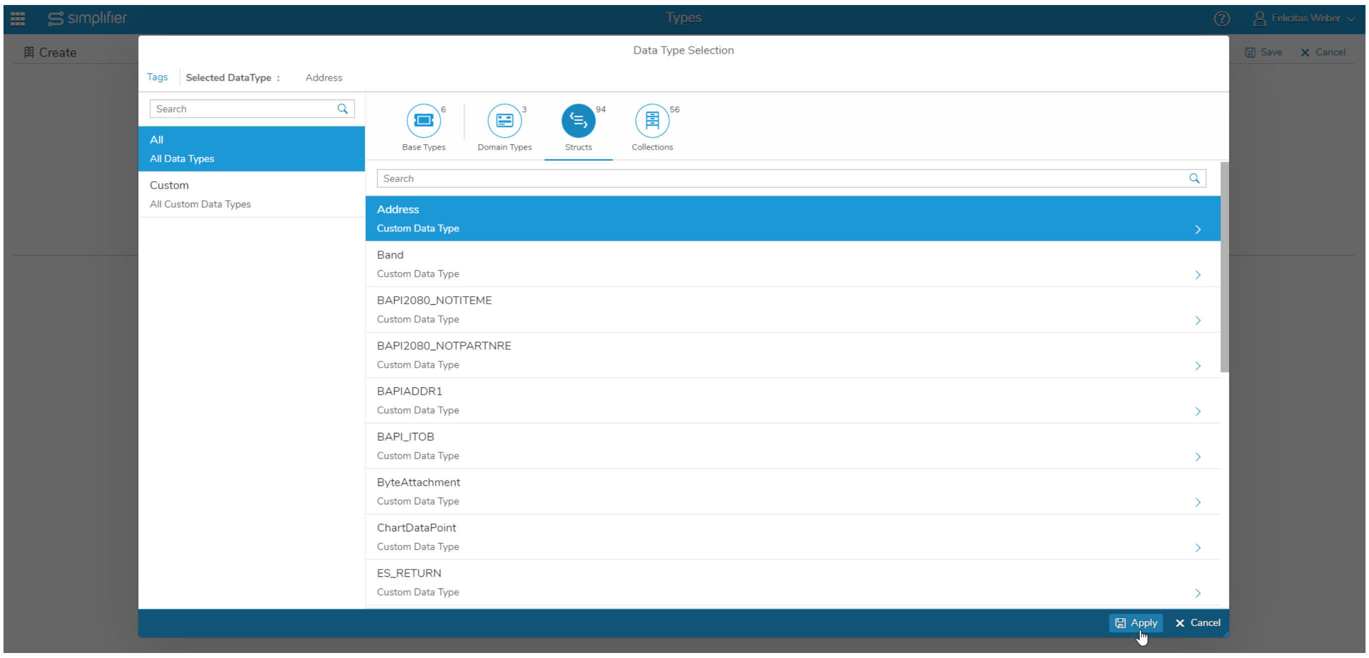
Collections represent multiple results of structs. For example, a database request may deliver a list of addresses from numerous people.

To create a new collection type click on the "+" button.

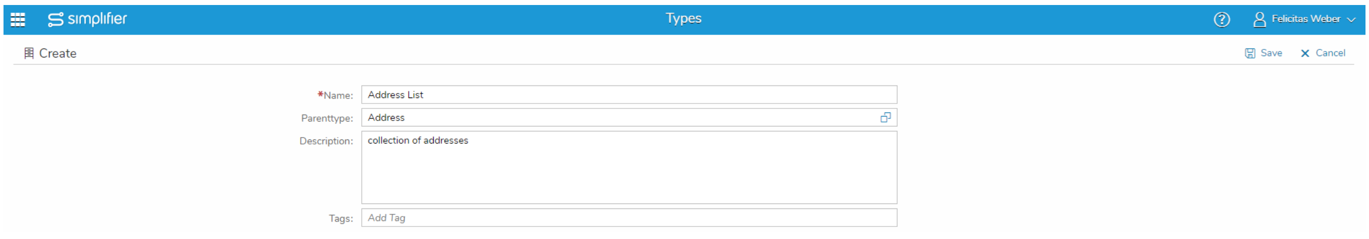


Define a unique collection **name** and a **description**.

By clicking on **Parenttype** a new pop up opens, where you can choose the parent type from.


simplifier
Types
Felicitas Weber
Create
Save
Cancel
Data Type Selection
Tags
Selected DataType : Address
Search
All
All Data Types
Custom
All Custom Data Types
Base Types 6
Domain Types 3
Structs 94
Collections 56
Search
Address
Custom Data Type
Band
Custom Data Type
BAPI2080_NOTITEME
Custom Data Type
BAPI2080_NOTPARTNRE
Custom Data Type
BAPIADDR1
Custom Data Type
BAPI_ITOB
Custom Data Type
ByteAttachment
Custom Data Type
ChartDataPoint
Custom Data Type
ES_RETURN
Custom Data Type
Apply
Cancel

After you have clicked on **Apply**, the parent type will be used.



*Name:	Address List
Parenttype:	Address
Description:	collection of addresses
Tags:	Add Tag

Conferencing Plugin for WebRTC Calls

https://developer.simplifier.io/documentation/getting-started/simplifier-mobile-client/conferencing-plugin-for-webrtc-calls/

The Cordova Conferencing provides WebRTC Conferencing functionality via Intel's WebRTC SDK.

Intel WebRTC SDK Version: 4.1

Platforms:

- Android (5.1+ - x86, armV7, arm64-v8a)
- iOS (11+)

Cordova: 8+

General information

When the local user connects to a room, this plugin overlays the cordova webview to show local and/or remote media streams.

Possible Actions

Active Actions

Actions triggered by local user

- init - Take stun/turn-, userconfig and initializes connection
- joinRoom - Join a room
- leaveRoom - Leave a room
- getParticipant - Get participants of a room
- editParticipant - Update participant properties
- kickParticipant - Kick a participant
- createRoom - Create a room
- getRoom - Get a room
- editRoom - Update room options
- deleteRoom - Delete a room
- subscribeConferencing - Subscribe the mixstream of a room
- subscribePeerToPeer - Subscribe the remoteStream from a participant
- unsubscribe - Unsubscribe a remote stream
- publish - Publish a local stream to a room
- unpublish - Unpublish a local stream
- startScreenSharing - Start to sharing/streaming the screen content
- stopScreenSharing - Stop screen sharing
- sendMessage - Send message to selected participantIds
- getStream - Get a stream of selected room
- editStream - Update stream properties
- deleteStream - Delete a stream
- startRecording - Start recording a stream
- editRecording - Update recording options
- stopRecording - Stop recording a stream
- getRecording - Get a recording
- startStreamIn - Add an external stream to room
- stopStreamIn - Remove the external stream

Passive Actions

Actions triggered by server/remote user/network

- onReceivedChatMessage - User receive a chat message
- onChangedParticipantStatus - Participant leave room
- onChangedConnectionStatus - Server/room get disconnected

Usage

Configuration description

Object	Field	Type	Possible Values	Default	Description
connectionConfig (required)	server	String	URL-Schema		signaling server url
	untrustedCertificate	boolean			Trust all certificates
mediaConfig (optional)	maxWidth	Integer	a resolution that makes sense	640	set maximum width of transmitted video to next (smaller) possible video resolution supported by device camera
	maxHeight	Integer	a resolution that makes sense	480	set maximum height of transmitted video to next (smaller) possible video resolution supported by device camera
	preferFrontCamera	boolean	true/false	false	if camera dialog is turned off, front camera is prefered to be opened if device has such
	maxFps	Integer		30	max transmitted frames
	maxVideoBandwidth	Integer	min 200 @ low resolution	1500	limits bandwidth of video channel to given value in mBit/s
	maxAudioBandwidth	Integer	min ~30	100	limits bandwidth of video channel to given value in mBit/s
	videoCodec	String	VP8, VP9, H264, H265	H264	switches used hardware decoder. if h264 is not present on device, VP8 is used a fallback
	audioCodec	String	opus, pcma, pcmu	opus	switches used hardware decoder. if h264 is not present on device, VP8 is used a fallback
iceConfig (Array - optional)	url	String	URL		Turn/Stun-URL
	username	String			username for Stun/Turn authentication
	password	String			password - can be empty for anonymous access

Cordova API

init

```
/**
 * Initializes native views and config properties.
 *
 * @param config     JSONObject - The config data (see example)
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.init(config, success, error);
```

joinRoom

```
/**
 * Create a conferencing token and join the selected room
 *
 * @param data       JSONObject - Username, roomId and role
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.joinRoom(data, token, success, error);
```

leaveRoom

```
/**
 * Leave the current room and unpublish and unsubscribe all streams
 *
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.leaveRoom(success, error);
```

getParticipant

```
/**
 * Get all participants or the selected participant of a room
 *
 * @param data       JSONObject - RoomId and userId
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.getParticipant(data, token, success, error);
```

editParticipant

```
/**
 * Update the participant properties
 *
 * @param data       JSONObject - RoomId, userId and userProperties
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.editParticipant(data, token, success, error);
```

kickParticipant

```
/**
 * Kick a participant
 *
 * @param data       JSONObject - RoomId and userId
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.kickParticipant(data, token, success, error);
```

createRoom

```
/**
 * Create a room
 *
 * @param data       JSONObject - Name and roomOptions
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.createRoom(data, token, success, error);
```

getRoom

```
/**
 * Get all rooms or the selected room
 *
 * @param data       JSONObject - RoomId
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.getRoom(data, token, success, error);
```

editRoom

```
/**
 * Update the room options
 *
 * @param data       JSONObject - RoomId and roomOptions
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.editRoom(data, token, success, error);
```

deleteRoom

```
/**
 * Delete a room
 *
 * @param data       JSONObject - RoomId
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.deleteRoom(data, token, success, error);
```

subscribeConferencing

```
/**
 * Subscribe the common stream of a room
 *
 * @param div        HTML Object - The HTML remote stream view
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.subscribeConferencing(div, success, error);
```

subscribePeerToPeer

```
/**
 * Subscribe the remoteStream from a participant
 * Note: Max. 2 participants
 *
 * @param div        HTML Object - The HTML remote stream view
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.subscribePeerToPeer(div, success, error);
```

unsubscribe

```
/**
 * Unsubscribe a remote stream
 *
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.unsubscribe(success, error);
```

publish

```
/**
 * Publish a local stream in a room
 *
 * @param div        HTML Object - The HTML local stream view
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.publish(div, success, error);
```

unpublish

```
/**
 * Unpublish a local stream
 *
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.unpublish(success, error);
```

startScreenSharing

```
/**
 * Start screen sharing
 *
 * @param div        HTML Object - The HTML local stream view
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.startScreenSharing(div, success, error);
```

stopScreenSharing

```
/**
 * Stop screen sharing
 *
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.stopScreenSharing(success, error);
```

sendMessage

```
/**
 * Send messages to the selected participants in a room
 *
 * @param message    String - messagetext
 * @param data       JSONArray - Selected participants (objects)
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.sendMessage(message, participants, success, error);
```

onReceivedChatMessage

```
/**
 * Action triggered by server when received a chat message
 *
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.onReceivedChatMessage(success, error);
```

onChangedParticipantStatus

```
/**
 * Action triggered by server when participant changed his status (participant leave
room)
 *
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.onChangedParticipantStatus(success, error);
```

onChangedConnectionStatus

```
/**
 * Action triggered by server when server/room changed his connections status
 *
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.onChangedConnectionStatus(success, error);
```

getStream

```
/**
 * Get all streams or the selected stream of a room
 *
 * @param data       JSONObject - RoomId and streamId
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.getStream(data, token, success, error);
```

deleteStream

```
/**
 * Delete a stream
 *
 * @param data       JSONObject - RoomId and streamId
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.deleteStream(data, token, success, error);
```

editStream

```
/**
 * Update the stream properties
 *
 * @param data       JSONObject - RoomId, streamId and streamProperties
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.editStream(data, token, success, error);
```

startRecording

```
/**
 * Start recording a stream of a room
 *
 * @param data       JSONObject - RoomId, container (mp4), media
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.startRecording(data, token, success, error);
```

editRecording

```
/**
 * Update the recording options
 *
 * @param data       JSONObject - RoomId, recordId, recordingOptions
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
```

```
ConferencingPlugin.editRecording(data, token, success, error);
```

stopRecording

```
/**
 * Stop recording of a stream
 *
 * @param data       JSONObject - RoomId and recordId
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.stopRecording(data, token, success, error);
```

getRecording

```
/**
 *  Get all or the selected recording of a room
 *
 * @param data       JSONObject - RoomId and recordId
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.getRecording(data, token, success, error);
```

startStreamIn

```
/**
 * Add an external stream to a room
 *
 * @param data       JSONObject - RoomId, url, transport (tcp), media (audio or video
)
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.startStreamIn(data, token, success, error);
```

stopStreamIn

```
/**
 * Remove the external stream
 *
 * @param data       JSONObject - RoomId and streamId
 * @param token      String - The simplifier token
 * @param success    The success callback is triggered when process is successful
 * @param error      The error callback is triggered when process failed
 */
ConferencingPlugin.stopStreamIn(data, token, success, error);
```

General event object description

```
{
  "action": "<String>", //see "Possible Actions"
  "result": "<String>", //some results depending on action
  "errorCode": "<Int>" //error code when action failed
}
```

Error Codes

Code	Description
1	SERVER_ERR
2	JSON_PARSE_ERR
3	PLUGIN_INIT_ERR
4	EDIT_COMMON_STREAM_ERR
5	JOIN_ROOM_ERR
6	CONFERENCE_CLIENT_CONFIGURATION_ERR
7	NO_ROOM_SELECTED_ERR
8	SUBSCRIPTION_ALREADY_EXIST_ERR
9	PUBLICATION_ALREADY_EXIST_ERR
10	SCREEN_PUBLICATION_ALREADY_EXIST_ERR
11	NO_SUBSCRIPTION_EXIST_ERR
12	NO_PUBLICATION_EXIST_ERR
13	NO_SCREEN_PUBLICATION_EXIST_ERR
14	PEER_TO_PEER_ERR
15	PUBLISH_STREAM_ERR
16	SCREEN_SHARING_ERR
17	SUBSCRIBE_STREAM_ERR
18	SEND_MESSAGE_ERR
19	LEAVE_ROOM_FAILED_ERR

Examples

Predefined callbacks and variables for the examples

```
let successCallback = function(event){
    console.log(event);
}

let errorCallback = function(event){
    console.error(event);
}

// Your SimplifierToken
let token = "dk4dfanew30239naa12dk2323r90asdf=="
```

- init

```
let config = {
        "connectionConfig": {
            "username": "TestUser",
            "server": "serverUrl_simplifier",
            "untrustedCertificate": true
        },
        "iceConfig": [{
                "url": "stun:turn.itizzimo.com:3478"
            },
            {
                "url": "stun:turn.itizzimo.com:3479"
            },
            {
                "url": "stun:turn.itizzimo.com:5349"
            },
            {
                "url": "stun:turn.itizzimo.com:5350"
            },
            {
                "url": "turn:turn.itizzimo.com:3478?transport=tcp",
                "username": "admin",
                "password": "admin"
            },
            {
                "url": "turn:turn.itizzimo.com:3478?transport=udp",
                "username": "admin",
                "password": "admin"
            },
            {
                "url": "turn:turn.itizzimo.com:3479?transport=tcp",
                "username": "admin",
                "password": "admin"
            },
            {
                "url": "turn:turn.itizzimo.com:3479?transport=udp",
                "username": "admin",
                "password": "admin"
            },
            {
                "url": "turn:turn.itizzimo.com:5349?transport=tcp",
                "username": "admin",
                "password": "admin"
            },
            {
                "url": "turn:turn.itizzimo.com:5349?transport=udp",
                "username": "admin",
                "password": "admin"
            },
            {
                "url": "turn:turn.itizzimo.com:5350?transport=tcp",
                "username": "admin",
                "password": "admin"
            },
            {
                "url": "turn:turn.itizzimo.com:5350?transport=udp",
                "username": "admin",
                "password": "admin"
            }
        ],
        "viewConfig": {
            "scalingFactor": 0.3,
            "gravity": "bottom_right"
        },
        "mediaConfig": {
            "maxWidth": 1280,
            "maxHeight": 720,
            "cameraDialog": true,
            "preferFrontCamera": true,
            "maxFps": 30,
            "maxVideoBandwidth": 5000,
            "maxAudioBandwidth": 200,
            "videoCodec": "H264",
            "audioCodec": "opus"
        },
        "debug": {
            "local": {
                "fps": false,
                "bitrate": false,
                "audiolevels": false
            },
            "remote": {
                "fps": false,
                "bitrate": false,
                "audiolevels": false
            }
        }
    };

ConferencingPlugin.init(config, successCallback, errorCallback);
```

- joinRoom

```
let data = {
    "role": "presenter",
    "username": "specialUser28",
    "room": "3023dak23dka1"  //roomId
};

ConferencingPlugin.joinRoom(data, token, successCallback, errorCallback);
```

- getParticipant

```
let data = {
    "room": "ksdfkfsdew3232",  //roomId
    "userId": "9SN_jqHVelwHksjaAACf"  //optional
};

ConferencingPlugin.getParticipant(data, token, successCallback, errorCallback);
```

- editParticipant

```
let data = {
    "room": "5c1b966d9869270cb9134328",
    "participantId": "HfwkX_3D1HYHjq-4AAEC",
    "items": [{
        "op": "replace",
        "path": "/permission/publish",
        "value": {
            "audio": false,
            "video": false
        }
    }]
};

ConferencingPlugin.editParticipant(data, token, successCallback, errorCallback);
```

- kickParticipant

```
let data = {
    "room": "5c1b966d9869270cb9134328",
    "participantId": "wRgmC31dz3hJHJzjAAEB"
};

ConferencingPlugin.kickParticipant(data, token, successCallback, errorCallback);
```

- createRoom

```
let data = {
    "name": "Testroom",
    "options": {
        "views": [{
            "video": {
                "parameters": {
                    "resolution": {
                        "height": 1080,
                        "width": 1920
                    },
                    "framerate": 60
                },
                "format": {
                    "codec": "h264",
                    "profile": "CB"   //For "h264" output only, "CB", "B", "M", "H"
                }
            }
        }],
        "participantLimit": 10,
        "inputLimit": -1
    }
};
ConferencingPlugin.createRoom(data, token, successCallback, errorCallback);
```

- getRoom

```
let data = {
    "room": "5c1b966d9869270cb9134328"  //optional
};

ConferencingPlugin.getRoom(data, token, successCallback, errorCallback);
```

- editRoom

```
let data = {
    "room": "5c1b966d9869270cb9134328",
    "options": {
        "name": "Testroom2",   //required
        "views": [{
            "video": {
                "parameters": {
                    "resolution": {
                        "height": 480,
                        "width": 640
                    },
                    "framerate": 30
                },
                "format": {
                    "codec": "h264",
                    "profile": "CB"
                }
            }
        }],
        "participantLimit": 24
    }
};

ConferencingPlugin.editRoom(data, token, successCallback, errorCallback);
```

- deleteRoom

```
let data = {
    "room": "5c1b966d9869270cb9134328"
};

ConferencingPlugin.deleteRoom(data, token, successCallback, errorCallback);
```

- getStream

```
let data = {
    "room": "5c3c632f2686480cbf83e7e1",
    "streamId": "519038650981614300"    //optional
};

ConferencingPlugin.getStream(data, token, successCallback, errorCallback);
```

- deleteStream

```
let data = {
    "room": "5c3c632f2686480cbf83e7e1",
    "streamId": "519038650981614300"
};

ConferencingPlugin.deleteStream(data, token, successCallback, errorCallback);
```

- editStream

```
let data = {
    "room": "5c3c632f2686480cbf83e7e1",
    "streamId": "140827592383601540",
    "items": [{
        "op": "replace",
        "path": "/media/video/status",
        "value": "inactive"
    }]
};

ConferencingPlugin.editStream(data, token, successCallback, errorCallback);
```

- startRecording

```
let data = {
    "room": "5bbb5a846ee20b02aa9cf430",
    "container": "mp4",
    "media": {
        "audio": {
            "from": "764484888390779500",
            "format": {
                "codec": "aac",
                "sampleRate": 48000,
                "channelNum": 2
            }
        },
        "video": {
            "from": "764484888390779500",
            "parameters": {
                "keyFrameInterval": 2
            },
            "format": {
                "codec": "h264",
                "profile": "CB"
            }
        }
    }
};

ConferencingPlugin.startRecording(data, token, successCallback, errorCallback);
```

- getRecording

```
let data = {
    "room": "5c3c632f2686480cbf83e7e1",
    "recordId": "287103235454593920"  //<optional>
};

ConferencingPlugin.getRecording(data, token, successCallback, errorCallback);
```

- editRecording

```
let data = {
    "room": "5c3c632f2686480cbf83e7e1",
    "recordId": "287103235454593920",
    "items": [
        {
                    "op": "replace",
                    "path": "/media/video/parameters/framerate",
                    "value": 60
        }
    ]
};

ConferencingPlugin.editRecording(data, token, successCallback, errorCallback);
```

- stopRecording

```
let data = {
    "room": "5c3c632f2686480cbf83e7e1",
    "recordId": "915058137572230700"
};

ConferencingPlugin.stopRecording(data, token, successCallback, errorCallback);
```

- startStreamIn

```
let data = {
    "room": "5c3c632f2686480cbf83e7e1",
    "url": "<video url>",
    "transport": "tcp",
    "media": {
        "audio": true,
        "video": true
    }
};

ConferencingPlugin.startStreamIn(data, token, successCallback, errorCallback);
```

- stopStreamIn

```
let data = {
    "room": "5c3c632f2686480cbf83e7e1",
    "streamId": "5c3c6313752307bf83e7d121"
};

ConferencingPlugin.stopStreamIn(data, token, successCallback, errorCallback);
```

Create a customized video layout for a room

A layout example for two participants in a room.

```
let data = {
    "name": "TestLayout",
    "options": {
        "views": [{
            "video": {
                "parameters": {
                    "resolution": {
                        "height": 1080,
                        "width": 1920
                    },
                    "framerate": 60
                },
                "format": {
                    "codec": "h264",
                    "profile": "CB"
                },
                "layout": {
                    "fitPolicy": "letterbox",   //letterbox or crop
                    "templates": {
                        "base": "fluid",   //fluid, lecture, void
                        "custom": [{
                            "region": [
                                {
                                "id": "1",
                                "shape": "rectangle",
                                "area": {
                                    "left": "0",
                                    "top": "0",
                                    "width": "1/2",
                                    "height": "1"
                                }
                                },
                                {
                                "id": "2",
                                "shape": "rectangle",
                                "area": {
                                    "left": "1/2",
                                    "top": "0",
                                    "width": "1/2",
                                    "height": "1"
                                }
                                }]
                            }]
                    }
                }
            }
        }],
        "participantLimit": 10,
        "inputLimit": -1
    }
}

ConferencingPlugin.createRoom(data, token, successCallback, errorCallback);
```

Testing

If you want to test the conferencing plugin, you have to follow the these steps.

1. Create a new cordova project

```
cordova create TestConferencingApp com.example.test TestConferencingApp
```

2. Add the following plugins

```
cordova plugin add cordova-plugin-test-framework

cordova plugin add cordova-plugin-conferencing

cordova plugin add cordova-plugin-conferencing-tests
```

3. Change the start page in config.xml from

```
<content src="index.html" /> to <content src="cdvtests/index.html" />
```

4. Add the android platform

```
cordova platform add android
```

5. Run the app

```
cordova run android
```

- **Auto Tests**
  - By clicking on the button **Auto Tests**you can run the automatic tests.
  - **Note:**If there are problems with token/authentication, you can change the credentials for the server with the button **getSimplifierToken** in the Manual Tests section. After the change, you can run the auto tests again.
- **Manual Tests**
  - By clicking on the button **Manual Tests**you can run the manual tests.
  - **Note:**For the conferencing functions you need a token and a conferencing room.

Sample Eventlog

```
//initialization success
Success: {"action":"init","result":"App initialized"}
//get all rooms - Result: Name, id, options
```

```
Success: {"action":"getRoom","result": JSONArray}
//join room
Success: {"action":"createToken","result":"asdj0dDK3kf239332=="}
//get participant - Result: id, user, role, permissions
Success: {"action":"getParticipant","result": JSONArray}
//publish local stream to room
Success: {"action":"publish","result":"Published stream"}
//subscribe common stream of the selected room
Success: {"action":"subscribeConferencing","result":"Subscribed stream"}
//unsubscribe remotestream
Success: {"action":"unsubscribe","result":"Unsubscribed stream"}
//create room - Result: Name, id, options
Success: {"action":"createRoom","result": JSONObject}
//delete room
Success: {"action":"deleteRoom","result":"Room deleted"}
//start screen sharing
Success: {"action":"startScreenSharing","result":"ScreenSharing started"}
//stop screen sharing
Success: {"action":"stopScreenSharing","result":"ScreenSharing stopped"}
//start stream recording - Result: id, storagePath, mediaOptions
Success: {"action":"startRecording","result": JSONObject}
//stop stream recording
Success: {"action":"stopRecording","result": ""}
//add external stream to room - Result: id, info, options
Success: {"action":"startStreamIn","result": JSONObject}
//remove external stream
Success: {"action":"stopStreamIn","result":""}
//leave room
Success: {"action":"leaveRoom","result":"Room left"}
```

More Information

Intel WebRTC SDK: https://software.intel.com/en-us/webrtc-sdk

Connector Access via Script

https://developer.simplifier.io/documentation/connectors/connector-via-script/

```
this.callConnector(connectorName, payload, callback, showBusyIndicator, failOnError,
failCallback)
```

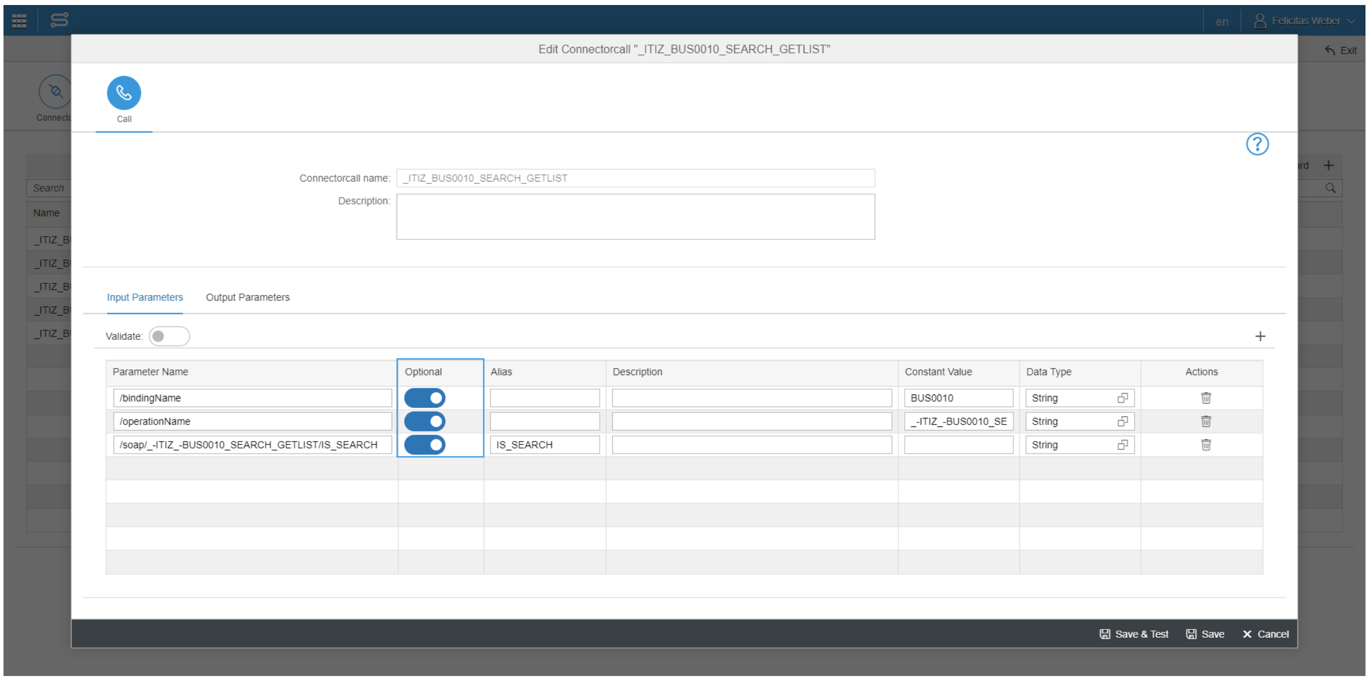
connectorName	the name of the connector
payload	a JSON object with the required parameters of the call
callback	function, which is called after the successful execution of the connector
showBusyIndicator	boolean value that indicates whether the screen has to be blocked by a loading bar during the call (true) or not (false)
failOnError	boolean value that indicates whether the connector should be called in case of an error of the function passed via "failCallback" (false) or not (true)
failCallback	function, which is called in case of an error in the connector, if false "failOnError" is passed

Connector Call Specific Parameters

https://developer.simplifier.io/documentation/connectors/create-and-manage-connector-calls/connector-call-specific-parameters/

The user interface for configuring a connector call is generic, thus it looks the same for all kinds of underlying types of connectors. Having the same interface for all kinds of connector calls is very convenient. But one drawback of this approach is, that some connectors require fixed parameters to be set, in order to work properly. This section tells you more about these details.

You can declare parameters of Connectors as optional.



Edit Connectorcall "_ITIZ_BUS0010_SEARCH_GETLIST"

Connectorcall name: _ITIZ_BUS0010_SEARCH_GETLIST

Description:

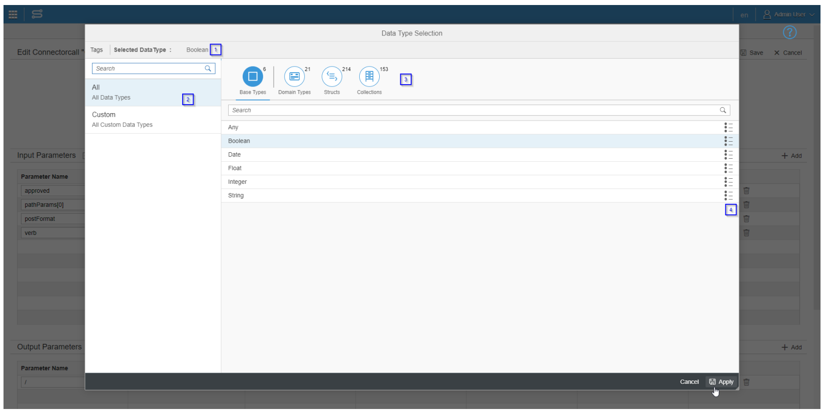
Input Parameters | Output Parameters

Validate:

Parameter Name	Optional	Alias	Description	Constant Value	Data Type	Actions
/bindingName	✓			BUS0010	String	
/operationName	✓			_-ITIZ_-BUS0010_SE	String	
/soap/_-ITIZ_-BUS0010_SEARCH_GETLIST/IS_SEARCH	✓	IS_SEARCH			String	

When declaring a parameter as non optional, the validation of the call will fail if the parameter is not provided.

Data Type Selector



If you click on the data type field, a selector opens.

Number	**Description**
1	The currently selected data type.
2	In this filtering list of all data types, you can find manually and automatically built data types. Custom data types are only manual data types. When the dialog opens in an automatically generated connector call, the data types of the connector can also be selected.
3	Prefilter of base and domain types, structs and collections.
4	You can always step deeper in the structure to select a data type.

Connector Call via Script

https://developer.simplifier.io/documentation/connectors/connector-via-script/connector-call-via-script/

In order to execute a connector call please use this code snippet:

```
this.callConnectorCall(connectorName, connectorCallName, payload, callback, showBusyI
ndicator, failOnError, failCallback)
```

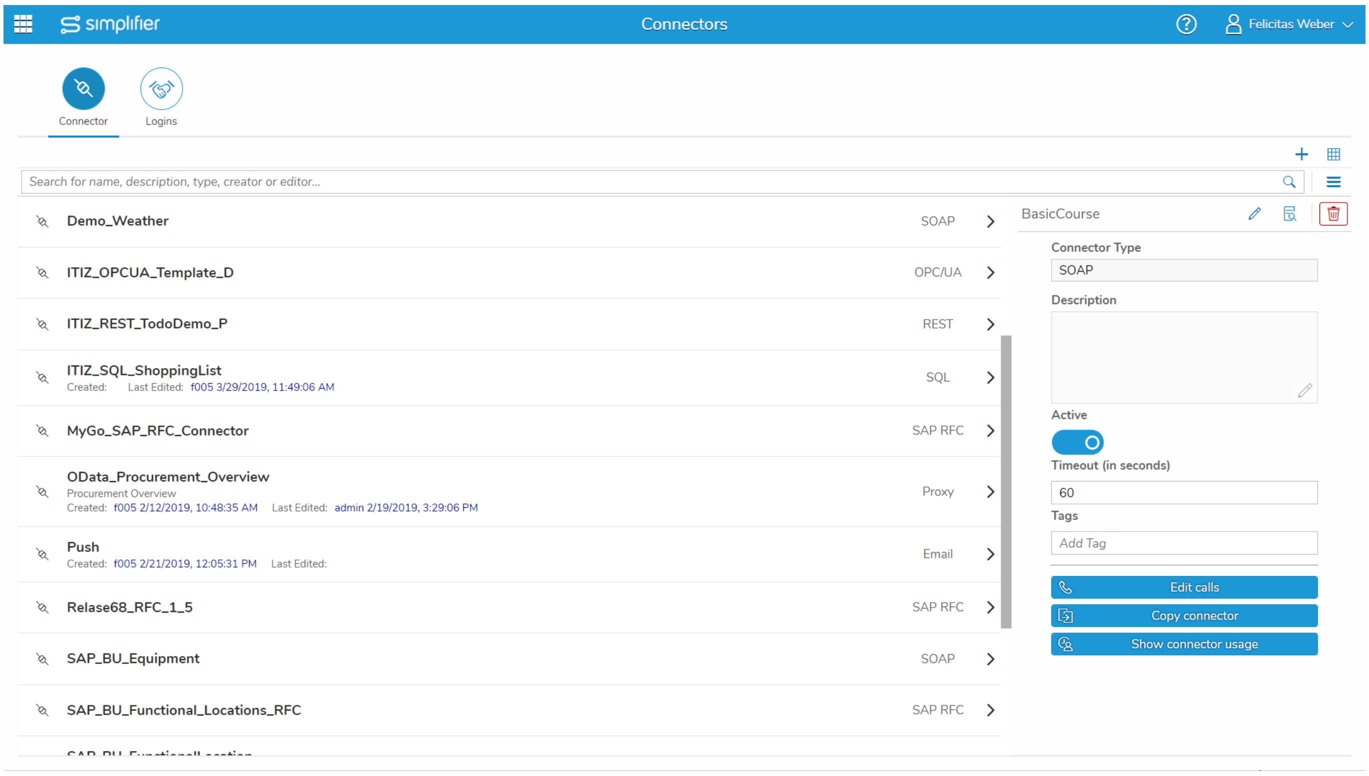
connectorName	the name of the connector
connectorCallName	the name of the connector call name
payload	a JSON object with the required parameters for the call
callback	function, which is called after the successful execution of the connector call
showBusyIndicator	boolean value that indicates whether the screen has to be blocked by a loading bar during the call (true) or not (false)
failOnError	boolean value that indicates whether the connector call should be called in case of an error of the function passed via "failCallback" (false) or not (true)
failCallback	function, which is called in case of an error in the connector call, if false for "failOnError" is passed

Connectors

https://developer.simplifier.io/documentation/connectors/

Connectors are the interface between a backend system and Simplifier to communicate with each other. They consist of at least one connector call.

Connector	Addresses a specific backend system (like SAP, or Database, etc.)
Connector Call	Leads a connector into action and contains input and output parameters



Demo_Weather	SOAP
ITIZ_OPCUA_Template_D	OPC/UA
ITIZ_REST_TodoDemo_P	REST
ITIZ_SQL_ShoppingList Created: Last Edited: f005 3/29/2019, 11:49:06 AM	SQL
MyGo_SAP_RFC_Connector	SAP RFC
OData_Procurement_Overview Procurement Overview Created: f005 2/12/2019, 10:48:35 AM Last Edited: admin 2/19/2019, 3:29:06 PM	Proxy
Push Created: f005 2/21/2019, 12:05:31 PM Last Edited:	Email
Relase68_RFC_1_5	SAP RFC
SAP_BU_Equipment	SOAP
SAP_BU_Functional_Locations_RFC	SAP RFC

BasicCourse

Connector Type: SOAP

Description:

Active

Timeout (in seconds): 60

Tags: Add Tag

Standardized Connectors in Simplifier

Connector Type	Description
SOAP	Use the SOAP connector to access a **S**imple **O**bject **A**ccess **P**rotocol based on HTTP and XML Format.
REST	The REST (**RE**presentational **S**tate **T**ransfer) connector is used for HTTP REST Services. The architecture uses standardized operations (GET, PUT, POST, DELETE) on web services. REST API is an alternative to other interfaces like SOAP. However, REST itself is neither protocol nor standard.
SQL	With the SQL (**S**tructured **Q**uery **L**anguage) connector, SQL statements are executed in a database schema, to request or edit based databases.
OPC/UA	The OPC/UA (**O**pen **P**latform **C**ommunications **U**nified **A**rchitecture) connector accesses to an OPC-UA server and performs READ/WRITE/SUBSCRIBE operations.
SAP RFC	The SAP RFC (**SAP R**emote **F**unction **C**all) connector is based on standard JCo SAP RFC to call functions

	in remote systems.
MQTT	MQTT (**M**essage **Q**ueuing **T**elemetry **T**ransport) is an open message protocol for machine-to-machine communication (M2M) that allows telemetry data to be transmitted as messages between devices, despite high delays or limited networks. This connector acts as a client and can publish or subscribe messages from an MQTT server (broker).
Push	The Push connector sends push notifications over WebSockets directly to Simplifier Clients or Simplifier Browser Apps without using Google or Apple's Cloud Services to support data protection and privacy.
CSV	Use the CSV (**C**omma-**s**eparated **v**alues) connector to read and/or write comma-separated files on a local file store.
OData V2	OData (**O**pen **Data** Protocol) is an open protocol based on HTTP for data access to enable CRUD operations. It enables the creation of REST-based data services to be published and edited by Web clients using simple HTTP messages.
Email	Use the Email connector to send emails over SMTP (Simple Mail Transfer Protocol) with or without SSL Encryption.
Logging	The Logging connector transfers the Simplifier application logs to a central monitoring tool/logwatch.
Proxy	The Proxy Connector allows the usage of any HTTP services that are not based on specific protocol architectures such as REST, SOAP or OData.

Content Files

https://developer.simplifier.io/documentation/plugins/content-repository/content-files/

Add

Slot	Description
contentFileAdd	This function adds a new content file

FileSystem:

Input parameters

Key	Type	Description
folderId	Integer	ID of the parent folder
name	String	File name (also used to determine the MimeType)
description	String (optional)	Description of the file
securitySchemeID	String	'public': file is public, 'private': file is not public
permissionObjectType	String	Must be specified as 'Session'
permissionObjectID	String	The ID of the Object Type can be freely selected
data	String (optional)	Base64 encoded content of the file
uploadSession	String (optional)	Session of an AppServer Html5 Upload
copyFrom	Integer (optional)	ID of the copied file

Note:

The content of the file can be transferred in three different ways. Exactly one of the following parameters must be passed:

- **data**: The content is passed directly with the JSON request as a Base64 encoded byte array.
- **uploadSession**: The content is first transferred to the AppServer via chunked HTML5 upload and the returned session is used as source. The Content Repository plugin downloads the file from the app server and uses it as content of the file
- **copyFrom**: The content is copied from another existing content file (copyFrom contains the ID of the file to be copied). Attention: The calling user must have the appropriate permissions to read the content file. Only the content is copied, not other properties (such as names, access rights, etc.). Any content file can be used as a source file, even in a different repository.

```
{
    "folderId" : 5,
    "name" : "test.txt",
    "description" : "My file description",
    "securitySchemeID" : "public",
    "permissionObjectType" : "Session",
    "permissionObjectID" : "abc",
    "data" : "dGVzdA=="
}
```

Output parameters

Key	Type	Description
id	Integer	ID of the created content file

name	String	Name of the created content file

```
{
    "id": 15,
    "name": "test.txt"
}
```

ClearFileSystem:

Input parameters

Key	**Type**	**Description**
contentId	Integer	ID of the content repository
fileName	String	Name of the file
folderPath	String	Path under which the file is to be stored
data	String (optional)	Base64 encoded content of the file
uploadSession	String (optional)	Session of an AppServer Html5 Upload
copyFrom	String (optional)	ID of the copied file
forceOverwrite	Boolean (optional)	If the flag has the value 'true', any existing file with the same name will be overwritten; If not set or 'false', the creation leads to an error if a file with the same name already exists

Note:

The content of the file can be transferred in three different ways. Exactly one of the following parameters must be passed:

- **data**: The content is passed directly with the JSON request as a Base64 encoded byte array.
- **uploadSession**: The content is first transferred to the AppServer via chunked HTML5 upload and the returned session is used as the source. The Content-Repo plugin downloads the file from the AppServer and uses it as the content of the file
- **copyFrom**: The content is copied from another existing content file (copyFrom contains the ID of the file to be copied). Attention: The calling user must have the appropriate permissions to read the content file. Only the content is copied, not other properties (such as names, access rights, etc.). Any content file can be used as a source file, even in a different repository.

```
{
    "contentId" : 5,
    "fileName" : "test.txt",
    "folderPath" : "MyParentFolder/MyChildFolder"
    "data" : "dGVzdA=="
}
```

Find

Slot	**Description**
contentFileFind	This function lists the searched content file

FileSystem:

Input parameters

Key	Type	Description
folderId	Integer	ID of the content folder in which the content is listed
name	String	Name of the searched file

```
{
    "folderId": 3,
    "name": "test.txt"
}
```

Output parameters

Key	Type	Description
files	Array	Array of all files (max. 1 element)
id	Integer	ID of the file
name	String	Name of the file
description	String	Description of the file
permissionObjectType	String	Must be specified as 'Session'
permissionObjectID	String	The ID of the Object Type can be freely selected
securitySchemeID	String	Security scheme ('public'/'private')
statusSchemeID	String	Status scheme (not implemented yet; always 'default')
statusID	String	Status scheme (not implemented yet; always 'default')
mimeType	Object	MimeType information
mimeType/extension	String	The file extension
mimeType/mimeType	String	The mimeType stored in the MimeMapping for the file extension
url	String	The download URL of the file

```
{
    "files": [
        {
            "id": 3,
            "name": "test.txt",
            "description": "My file description 1",
            "statusSchemeID": "Default",
            "statusID": "Default",
            "securitySchemeID": "public",
            "permissionObjectType": "Session",
            "permissionObjectID": "abc",
            "mimeType": {
                "extension": "jpg",
                "mimeType": "image"
            },
            "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/Re
poName/ParentFolder/ChildFolder/file.jpg/",
        }
    ]
}
```

ClearFileSystem:

Input parameters

Key	Type	Description
contentId	Integer	ID of the content repository in which you want to search
fileName	String	Name of the searched file
folderPath	String (optional)	Path of the folder to search in

```
{
    "contentId": 3,
    "filename": "test.txt",
    "folderPath: "MyParentFolder/MyChildfolder"
}
```

Output parameters

Key	Type	Description
files	Array	Array of all files
filePath	String	Path of the file
mimeType	Object	MimeType information
mimeType/extension	String	The file extension
mimeType/mimeType	String	The mimeType stored in the MimeMapping for the file extension
url	String	The download URL of the file

```
{
    "files": [
        {
            "filePath": "MyParentFolder/MyChildFolder/test.txt",
            "mimeType": {
                "extension": "txt",
                "mimeType": "text"
            },
            "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/Re
poName/MyParentFolder/MyChildFolder/test.txt/"
        },
        {
            "filePath": "MyParentFolder/MyChildFolder/MyFolder/test.txt",
            "mimeType": {
                "extension": "txt",
                "mimeType": "text"
            },
            "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/Re
poName/MyParentFolder/MyChildFolder/MyFolder/test.txt/"
        }
    ]
}
```

List

Slot	Description
contentFileList	This function lists a file

FileSystem:

Input parameters

Key	Type	Description
folderId	Integer	ID of the listed content folder

```
{
    "folderId": 3
}
```

Output parameters

Key	Type	Description
files	Array	Array of all files
id	Integer	ID of the file
name	String	Name of the file
description	String	Description of the file
permissionObjectType	String	Must be specified as 'Session'
permissionObjectID	String	The ID of the Object Type can be freely selected
securitySchemeID	String	Security scheme ('public'/'private')
statusSchemeID	String	Status scheme (not implemented yet; always 'default')
statusID	String	Status scheme (not implemented yet; always 'default')
mimeType	Object	MimeType information
mimeType/extension	String	The file extension
mimeType/mimeType	String	The mimeType stored in the MimeMapping for the file extension
url	String	The download URL of the file

```
{
    "files": [
        {
            "id": 3,
            "name": "test.txt",
            "description": "My file description 1",
            "statusSchemeID": "Default",
            "statusID": "Default",
            "securitySchemeID": "public",
            "permissionObjectType": "Session",
            "permissionObjectID": "abc",
            "mimeType": {
                "extension": "txt",
                "mimeType": "text"
            },
            "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/Re
poName/MyParentFolder/MyChildFolder/test.txt/"
        },
        {
            "id": 4,
            "name": "test2.txt",
            "description": "My file description 2",
            "statusSchemeID": "Default",
            "statusID": "Default",
            "securitySchemeID": "public",
            "permissionObjectType": "Session",
            "permissionObjectID": "abc",
            "mimeType": {
                "extension": "txt",
                "mimeType": "text"
            },
            "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/Re
poName/MyParentFolder/MyChildFolder/test2.txt/"
        }
    ]
}
```

ClearFileSystem:

Input parameters

Key	**Type**	**Description**
contentId	Integer	ID of the listed content repository
folderPath	String (optional)	Path of the folder of the listed files

```
{
    "contentId": 3,
    "folderPath": "MyFolder"
}
```

Output parameters

Key	**Type**	**Description**
files	Array	Array of all files
fileName	String	Name of the file
mimeType	Object	MimeType information
mimeType/extension	String	The file extension
mimeType/mimeType	String	The mimeType stored in the MimeMapping for the file extension
url	String	The download URL of the file

```
{
    "files": [
        {
            "fileName": "test.txt",
            "mimeType": {
                "extension": "txt",
                "mimeType": "text"
            },
            "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/Re
poName/MyParentFolder/MyChildFolder/test.txt/"
        },
        {
            "name": "test2.txt",
            "mimeType": {
                "extension": "txt",
                "mimeType": "text"
            },
            "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/Re
poName/MyParentFolder/MyChildFolder/test2.txt/"
        }
    ]
}
```

Get

Slot	Description
contentFileGet	This function queries

FileSystem:

Input parameters

Key	Type	Description
id	Integer	Primary key

```
{
    "id": 3
}
```

Output parameters

Key	Type	Description
id	Integer	ID of the file
folderId	Integer	ID of the parent folder
name	String	Name of the file
description	String	Description of the file
permissionObjectType	String	Must be specified as 'Session'
permissionObjectID	String	The ID of the Object Type can be freely selected
securitySchemeID	String	Security scheme ('public'/'private')
statusSchemeID	String	Status scheme (not implemented yet; always 'default')
statusID	String	Status scheme (not implemented yet; always 'default')
data	String	Base64 encoded content of the file
mimeType	Object	MimeType information
mimeType/extension	String	The file extension
mimeType/mimeType	String	The mimeType stored in the MimeMapping for the file extension
url	String	The download URL of the file

```
{
    "id": 3,
    "folderId": 5,
    "name": "test.txt",
    "description": "My file description",
    "statusSchemeID": "Default",
    "statusID": "Default",
    "securitySchemeID": "public",
    "permissionObjectType": "Session",
    "permissionObjectID": "abc",
    "data": "dGVzdA==",
    "mimeType": {
        "extension": "txt",
        "mimeType": "text"
    },
    "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/RepoName/M
yParentFolder/MyChildFolder/test.txt/"
}
```

ClearFileSystem:

Input parameters

Key	Type	Description
contentId	Integer	ID of the listed content repository
filePath	String	Path of the file

```
{
    "contentId": 3,
    "filePath": "MyFolder/test.txt"
}
```

Output parameters

Key	Type	Description
filePath	String	File name
data	String	Base64 encoded content of the file
length	Integer	Length of the file in bytes
mimeType	Object	MimeType information
mimeType/extension	String	The file extension
mimeType/mimeType	String	The mimeType stored in the MimeMapping for the file extension
url	String	The download URL of the file

```
{
    "filePath": "MyFolder/test.txt",
    "data": "dGVzdA==",
    "length": 59570,
    "mimeType": {
        "extension": "txt",
        "mimeType": "text"
    },
    "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/RepoName/M
yFolder/test.txt/"
}
```

Get Metadata

Slot	Description
contentFileGetMetadata	This function queries the metadata

FileSystem:

Input parameters

Key	Type	Description
id	Integer	Primary key

```
{
    "id": 3
}
```

Output parameters

Key	Type	Description
id	Integer	ID of the file
folderId	Integer	ID of the parent folder
name	String	File name
description	String	Description of the file
permissionObjectType	String	Must be specified as 'Session'
permissionObjectID	String	The ID of the Object Type can be freely selected
securitySchemeID	String	Security scheme ('public'/'private')
statusSchemeID	String	Status scheme (not implemented yet; always 'default')
statusID	String	Status scheme (not implemented yet; always 'default')
mimeType	Object	MimeType information
mimeType/extension	String	The file extension
mimeType/mimeType	String	The mimeType stored in the MimeMapping for the file extension
url	String	The download URL of the file

```
{
    "id": 3,
    "folderId": 5,
    "name": "test.txt",
    "description": "My file description",
    "statusSchemeID": "Default",
    "statusID": "Default",
    "securitySchemeID": "public",
    "permissionObjectType": "Session",
    "permissionObjectID": "abc",
    "mimeType": {
        "extension": "txt",
        "mimeType": "text"
    },
    "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/RepoName/MyParentFolder/MyChildFolder/test.txt"
}
```

ClearFileSystem:

Input parameters

Key	Type	Description
contentId	Integer	ID of the listed content repository
filePath	String	Path of the file

```
{
    "contentId": 3,
    "filePath": "MyFolder/test.txt"
}
```

Output parameters

Key	Type	Description
filePath	String	File name
mimeType	Object	MimeType information
mimeType/extension	String	The file extension
mimeType/mimeType	String	The mimeType stored in the MimeMapping for the file extension
url	String	The download URL of the file

```
{
    "filePath": "MyFolder/test.txt",
    "mimeType": {
        "extension": "txt",
        "mimeType": "text"
    },
    "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/RepoName/MyFolder/test.txt"
}
```

Get Metadata batched

Slot	Description
contentFileGetMetadataBatched	This function queries the metadata batched

FileSystem:

Input parameters

Key	Type	Description
contentId	Integer	ID of the repository in which the files are stored
files	Array[Object]	A list of file objects
files/id	Integer	ID of the file

```
{
    "contentId": 1,
    "files": [{
        "id": 1
    },
    {
        "id": 2
    }]
}
```

Output parameters

Key	Type	Description
fileMetadata	Array[Object]	A list of metadata objects
fileMetadata/id	Integer	ID of the file
fileMetadata/folderId	Integer	ID of the parent folder
fileMetadata/name	String	Name of the file
fileMetadata/description	String	Description of the file
fileMetadata/permissionObjectType	String	Must be specified as 'Session'
fileMetadata/permissionObjectID	String	The ID of the Object Type can be freely selected
fileMetadata/securitySchemeID	String	Security scheme ('public'/'private')
fileMetadata/statusSchemeID	String	Status scheme (not implemented yet; always 'default')
fileMetadata/statusID	String	Status scheme (not implemented yet; always 'default')
fileMetadata/mimeType	Object	MimeType information
fileMetadata/mimeType/extension	String	The file extension
fileMetadata/mimeType/mimeType	String	The mimeType stored in the MimeMapping for the file extension
fileMetadata/url	String	The download URL of the file

```
{
    fileMetadata: [{
        "id": 1,
        "folderId": 5,
        "name": "test.txt",
        "description": "My file description",
        "statusSchemeID": "Default",
        "statusID": "Default",
        "securitySchemeID": "public",
        "permissionObjectType": "Session",
        "permissionObjectID": "abc",
        "mimeType": {
            "extension": "txt",
            "mimeType": "text"
        },
        "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/RepoNa
me/MyParentFolder/MyChildFolder/test.txt"
    },
    {
        "id": 2,
        "folderId": 3,
        "name": "picture.jpg",
        "description": "My file description",
        "statusSchemeID": "Default",
        "statusID": "Default",
        "securitySchemeID": "public",
        "permissionObjectType": "Session",
        "permissionObjectID": "abc",
        "mimeType": {
            "extension": "jpg",
            "mimeType": "picture"
        },
        "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/RepoNa
me/MyParentFolder/MyChildFolder2/picture.jpg"
    }]
}
```

ClearFileSystem:

Input parameters

Key	Type	Description
contentId	Integer	ID of the listed content repository
files	Array[Object]	A list of file objects
files/filePath	String	Path of the file

```
{
    "contentId": 6,
    "files": [{
        "filePath": "Folder/picture.jpg"
    },
    {
        "filePath": "Folder2/text.txt"
    }]
}
```

Output parameters

Key	Type	Description
fileMetadata	Array[Object]	A list of metadata object
fileMetadata/filePath	String	File name
fileMetadata/mimeType	Object	MimeType information
fileMetadata/mimeType/extension	String	The file extension
fileMetadata/mimeType/mimeType	String	The mimeType stored in the

		MimeMapping for the file extension
fileMetadata/url	String	The download URL of the file

```
{
    fileMetadata: [{
        "filePath": "Folder/picutre.jpg",
        "mimeType": {
            "extension": "jpg",
            "mimeType": "picture"
        },
        "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/RepoNa
me/Folder/picture.jpg"
    },
    {
        "filePath": "Folder2/test.txt",
        "mimeType": {
            "extension": "txt",
            "mimeType": "text"
        },
        "url": "http://localhost:8080/client/2.0/plugin/contentRepoPlugin/file/RepoNa
me/Folder2/test.txt"
    }]
}
```

Edit

Slot	**Description**
contentFileEdit	This function edits a content file

FileSystem:

Input parameters

Key	**Type**	**Description**
id	Integer	ID of the data to be processed
name	String	File name (also used to determine the MimeType)
description	String (Optional)	Description of the file
securitySchemeID	String	'public': file is public, 'private': file is not public
permissionObjectType	String	Must be specified as 'Session'
permissionObjectID	String	The ID of the Object Type can be freely selected
data	String	Base64 encoded content of the file

```
{
    "id" : 5,
    "name" : "test.txt",
    "description": "My new file description",
    "securitySchemeID" : "public",
    "permissionObjectType" : "Session",
    "permissionObjectID" : "abc",
    "data" : "dGVzdA=="
}
```

ClearFileSystem:

Input parameters

Key	Type	Description
contentId	Integer	ID of the content repository in which the file is stored
sourceFilePath	String	Path of the file to be edited
destFilePath	String	Path incl. new name under which the file is to be stored
forceOverwrite	Boolean (optional)	If the flag has the value 'true', any existing file with the same name will be overwritten; If not set or 'false', the creation leads to an error if a file with the same name already exists

```
{
    "contentId" : 5,
    "sourceFilePath" : "MyParentFolder/test.txt",
    "destFilePath": "MyParentFolder/MyChildFolder/myRenamedMovedFile.txt"
}
```

Delete

Slot	Description
contentFileDelete	This function deletes a content file

FileSystem:

Input parameters

Key	Type	Description
id	Integer	Primary key

```
{
    "id": 15
}
```

ClearFileSystem:

Input parameters

Key	Type	Description
contentId	Integer	ID of the content repository in which the file is stored
filePath	String	Path of the file to be deleted

```
{
    "contentId": 10,
    "filePath" : "MyFolder/myFile.txt"
}
```

Content Repository

https://developer.simplifier.io/documentation/plugins/content-repository/

The Content Repository Plugin is used to implement a persistence layer for data so that you can store images and videos using this plugin. It contains a repository, folders and files, so you create a repository (parent folder) in which subfolders can be stored in any hierarchy.

Example call of a Content Repository Plugin function via a server-side Business Object:

```
var payloadClearFileSystem = {
    slot: "contentRepositoryAdd",
                payload: {
                               "provider": "ClearFileSystem",
                               "name": input.name,
                               "description": input.description
                }
};

var result = JSON.parse(PLUGIN_contentRepoPlugin.run(JSON.stringify(payloadClearFileS
ystem)))
```

The payload configuration depends on the required slot.

Difference between File System and Clear File System:

The file system stores the received content repository data in a database.
The clear file system stores this data in an actual file system (compare Windows Explorer).

Content Repositories

Add

Slot	Description
contentRepositoryAdd	This function adds a new content repository

FileSystem:

Input parameters

Key	Type	Description
name	String	Name of the repository
description	String (optional)	Description of the repository
provider	String	Content provider (must be specified as 'FileSystem')
permissionObjectType	String	Must be specified as 'App'
permissionObjectID	String	The ID of the Object Type can be freely selected

```
{
    "permissionObjectType" : "App",
    "permissionObjectID": "DummyApp",
    "provider" : "FileSystem",
    "name": "MyTestRepo",
    "description": "MyTestRepoDescription"
}
```

Output parameter

Key	Type	Description
id	String	The ID of the created repository

```
{
    "id": 15
}
```

ClearFileSystem:

Input parameters

Key	Type	Description
name	String	Name of the repository
description	String (optional)	Description of the repository
provider	String	Content provider (must be specified as 'ClearFileSystem')

```
{
    "name": "MyTestRepo",
    "provider" : "ClearFileSystem",
    "description": "MyTestRepoDescription"
}
```

Output parameters

Key	Type	Description
id	Integer	ID of the created ContentRepository
description	String	Description of the repository

```
{
    "id": 15,
    "description": "MyTestRepoDescription"
}
```

Find

Slot	**Description**
contentRepositoryFind	This function lists only repositories for which the user has authorizations

FileSystem:

Input parameter

Key	**Type**	**Description**
name	String	Name of the searched repository

```
{
    "name": "MyRepo"
}
```

Output parameters

Key	**Type**	**Description**
repositories	Array	Array of all repositories (max. 1 element)
id	Integer	ID of the repository
name	String	Name of the repository
description	String	Description of the repository
permissionObjectType	String	Must be specified as 'App'
permissionObjectID	String	The ID of the Object Type can be freely selected
provider	String	Content provider (must be specified as 'FileSystem')

```
{
    "repositories": [
        {
            "id": 3,
            "name": "MyRepo",
            "description": "My repo description",
            "permissionObjectType": "App",
            "permissionObjectID": "DummyApp",
            "provider": "FileSystem",
        }
    ]
}
```

ClearFileSystem:

Input parameter

Key	Type	Description
name	String	Name of the searched repository

```
{
    "name": "MyRepo"
}
```

Output parameters

Key	Type	Description
repositories	Array	Array of all repositories (max. 1 element)
id	Integer	ID of the repository
name	String	Name of the repository
description	String	Description of the repository
provider	String	Content provider (must be specified as 'ClearFileSystem')

```
{
    "repositories": [
        {
            "id": 3,
            "name": "MyRepo",
            "description": "My repo description",
            "provider": "ClearFileSystem"
        }
    ]
}
```

List

Slot	Description
contentRepositoryList	This function finds only repositories for which the user has authorizations

FileSystem:

Input parameter

Key	Type	Description
provider	String (optional)	Content provider (must be specified as 'FileSystem')

		If no provider is specified, all repositories are returned

```
{
    "provider": "FileSystem"
}
```

Output parameters

Key	Type	Description
repositories	Array	Array of all repositories
id	Integer	ID of the repository
name	String	Name of the repository
description	String	Description of the repository
permissionObjectType	String	Must be specified as 'App'
permissionObjectID	String	The ID of the Object Type can be freely selected
provider	String	Content provider

```
{
    "repositories": [
        {
            "id": 3,
            "name": "MyRepo",
            "description": "My repo description",
            "permissionObjectType": "App",
            "permissionObjectID": "DummyApp",
            "provider": "FileSystem",
        },
        {
            "id": 4,
            "name": "MyRepo2",
            "description": "My repo description 2",
            "permissionObjectType": "Session",
            "permissionObjectID": "abc",
            "provider": "FileSystem",
        }
    ]
}
```

ClearFileSystem:

Input parameter

Key	Type	Description
provider	String (optional)	Content provider (must be specified as 'ClearFileSystem') If no provider is specified, all repositories are returned

```
{
    "provider": "ClearFileSystem"
}
```

Output parameters

Key	Type	Description
repositories	Array	Array of all repositories
id	Integer	ID of the repository
name	String	Name of the repository
description	String	Description of the repository
provider	String	Content provider

```
{
    "repositories": [
        {
            "id": 5,
            "name": "MyRepo5",
            "description": "My repo description 5",
            "provider": "ClearFileSystem"
        },
        {
            "id": 6,
            "name": "MyRepo6",
            "description": "My repo description 6",
            "provider": "ClearFileSystem"
        }
    ]
}
```

Get

Slot
contentRepositoryGet

FileSystem:

Input parameter

Key	Type	Description
id	Integer	Primary key

```
{
    "id": 3
}
```

Output parameters

Key	Type	Description
id	Integer	ID of the repository
name	String	Name of the repository
description	String	Description of the repository
permissionObjectType	String	Must be specified as 'App'
permissionObjectID	String	The ID of the Object Type can be freely selected
provider	String	Content provider (must be specified as 'FileSystem')

```
{
    "id": 3,
    "name": "MyRepo",
    "description": "My repo description",
    "permissionObjectType": "App",
    "permissionObjectID": "DummyApp",
    "provider": "FileSystem",
}
```

ClearFileSystem:

Input parameter

Key	Type	Description
id	Integer	Primary key

```
{
    "id": 3
}
```

Output parameters

Key	Type	Description
id	Integer	ID of the repository
name	String	Name of the repository
description	String	Description of the repository
provider	String	Content provider (must be specified as 'ClearFileSystem')

```
{
    "id": 3,
    "name": "MyRepo",
    "description": "My repo description",
    "provider": "ClearFileSystem",
}
```

Edit

Slot	Description
contentRepositoryEdit	This function edits a content repository

FileSystem:

Input parameters

Key	Type	Description
id	Integer	Primary key (ID of the repository)
name	String	Name of the repository
description	String (optional)	Description of the repository
permissionObjectType	String	Must be specified as 'App'
permissionObjectID	String	The ID of the Object Type can be freely selected

```
{
    "id" : 15,
    "permissionObjectType" : "App",
    "permissionObjectID": "DummyApp",
    "name": "MyTestRepo",
    "description": "My new description",
}
```

ClearFileSystem:

Input parameters

Key	Type	Description
id	Integer	Primary key (ID of the repository)
name	String	Name of the repository
description	String (optional)	Description of the repository

```
{
    "id" : 15,
    "name": "MyTestRepo",
    "description": "My new description"
}
```

Delete

A repository can only be deleted if it does not contain any content folders.

Slot	Description
contentRepositoryDelete	This function deletes a content repository

Input parameter

Key	**Type**	**Description**
id	Integer	Primary key

```
{
    "id": 15
}
```

Controlled Integration of Data and Content into Web Applications

https://developer.simplifier.io/documentation/security-guidelines/controlled-integration-of-data-and-content-into-web-applications/

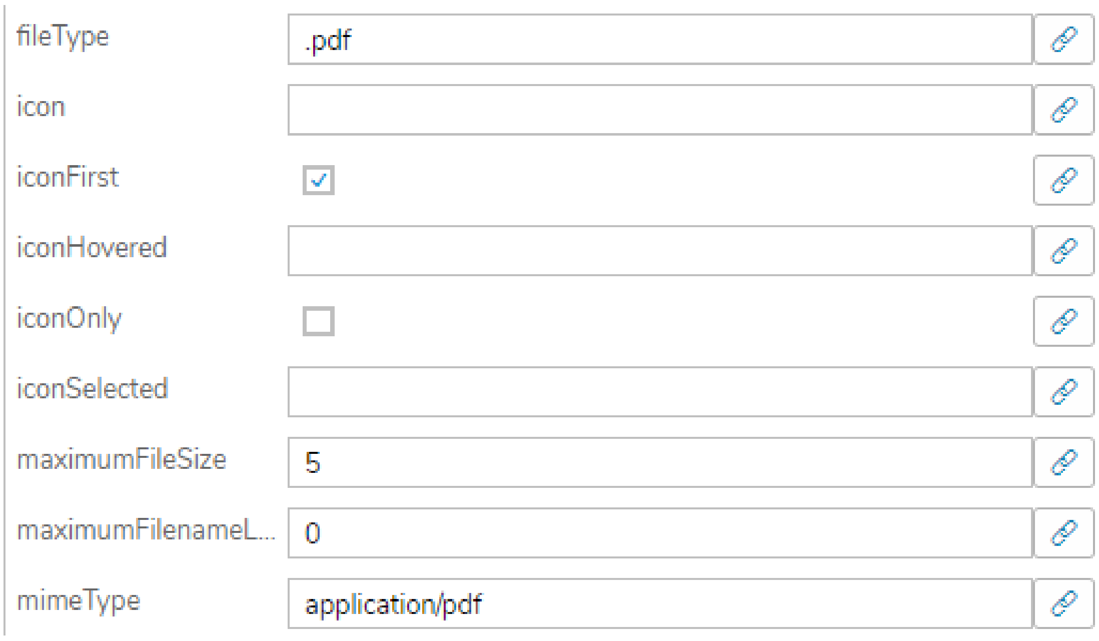
Recommendation: In general, uploads to Simplifier should be checked by a Web Application Firewall (WAF) virus scanner or by connecting an external virus scanner via ICAP interface in the configuration of the reverse proxy.

If a virus is found, the WAF or ICAP connected virus scanner should respond with an HTTP header status code to 409 (Conflict).

The body of the response (JSON format) should look like this:

```
{
  success: false,
  msg: "A virus was found in the file. The file cannot be uploaded."
}
```

The widget "FileUploader" is configured to process a status code 409 as a virus discovery.

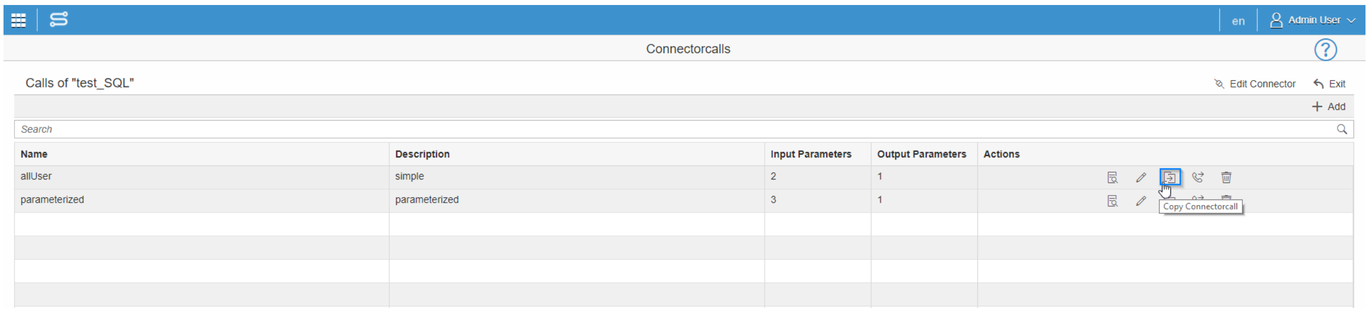


fileType	.pdf
icon	
iconFirst	✓
iconHovered	
iconOnly	☐
iconSelected	
maximumFileSize	5
maximumFilenameL...	0
mimeType	application/pdf

Copy Connector Calls

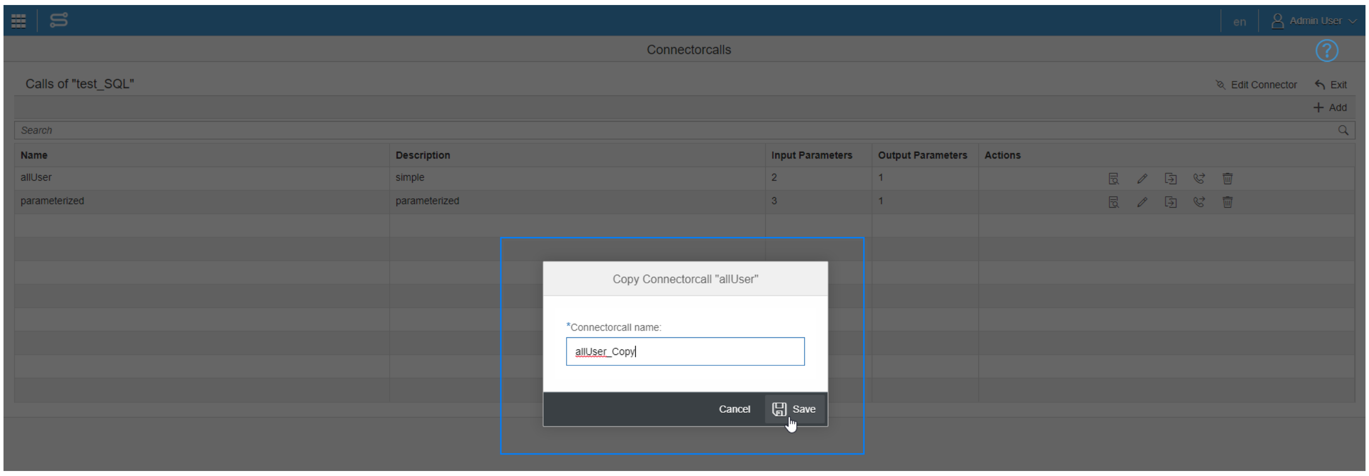
https://developer.simplifier.io/documentation/connectors/create-and-manage-connector-calls/copy-connectorcalls/

You can copy a connector call within a connector in the connector call overview by clicking the appropriate copy button.



Name	Description	Input Parameters	Output Parameters	Actions
allUser	simple	2	1	
parameterized	parameterized	3	1	

By clicking the button a new pop up opens in which you can specify the name of the copied connector call. The default value is the name of the copied connector call added _copy.



Name	Description	Input Parameters	Output Parameters	Actions
allUser	simple	2	1	
parameterized	parameterized	3	1	

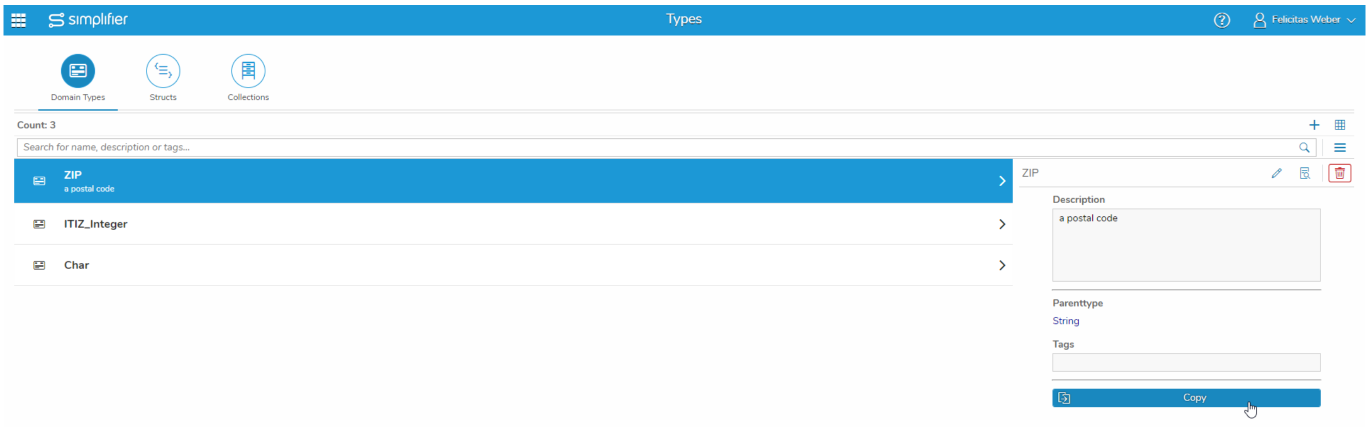
*Connectorcall name: allUser_Copy

Once you have assigned a name, click on the save button. Your connector call has been copied with all input and output parameters.

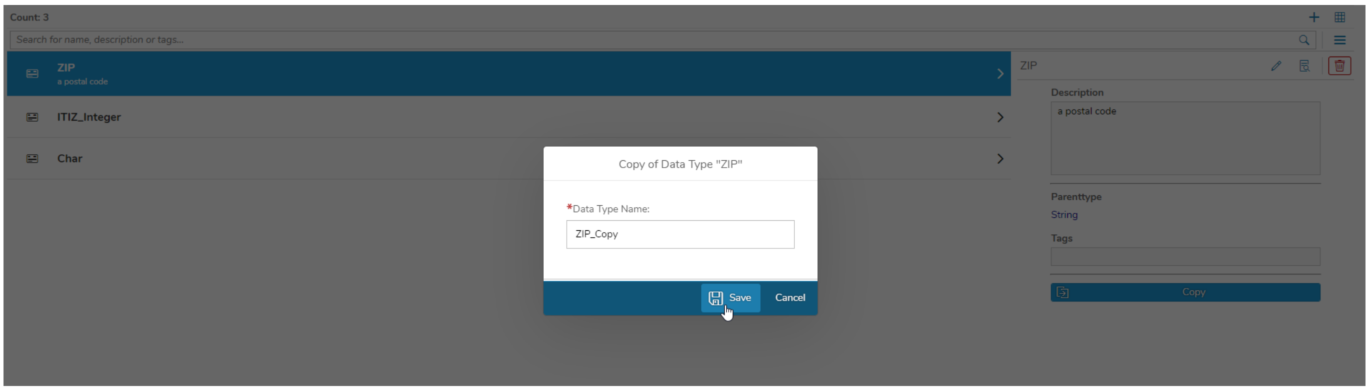
Copy Data Types

https://developer.simplifier.io/documentation/data-types/copy-data-type/

You can copy any Data Type of Simplifier. The copy will have all attributes/fields and any tags are given to the copied template.



Click on the copy icon and a new pop-up will appear.



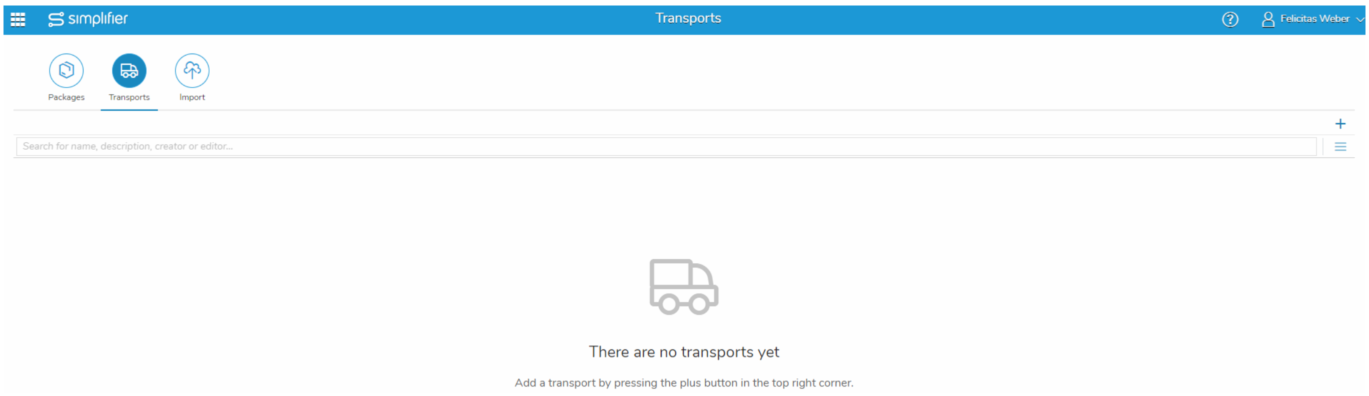
By default, "_Copy" is added to the current Data Type Name. However, you can also assign a new name. Click 'Save' and the copy has been created.

Create a Transport

https://developer.simplifier.io/documentation/transports/create-a-transport/

Vimeo Video

If you switch to the Transports tab, the overview of transports appears. It allows you to define transport requests that group one or more packages.

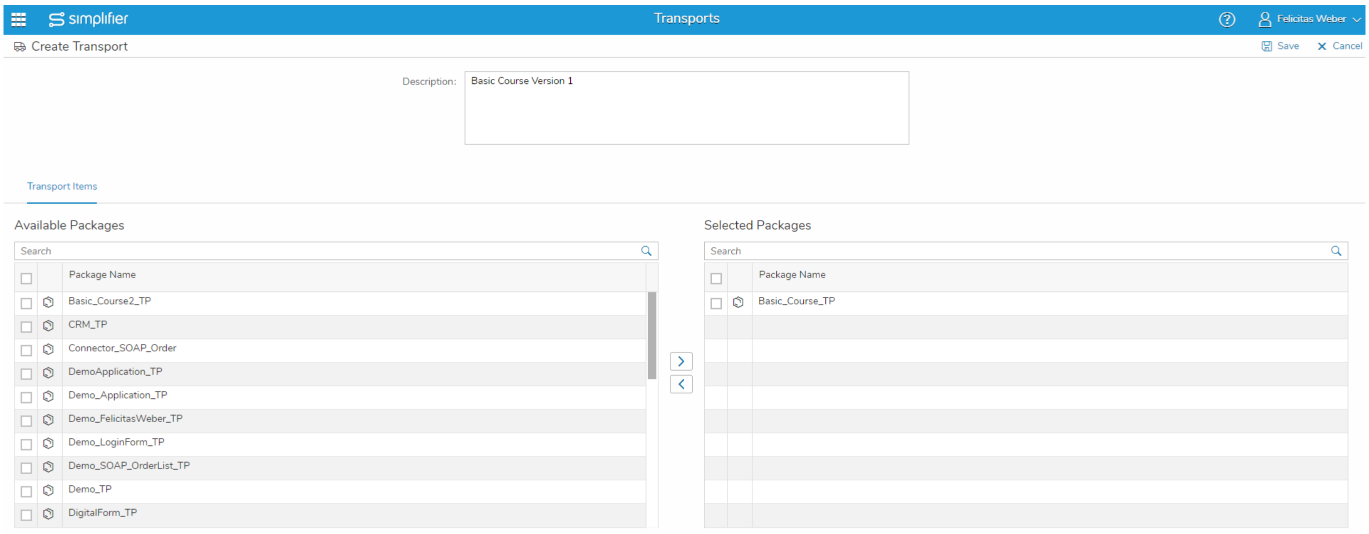


There are no transports yet

Add a transport by pressing the plus button in the top right corner.

Click on the plus icon to create a new transport. The name of a new transport is generated automatically and is unique within the server environment. It always consists of the instance name and a 10-digit number.

You can add a description, that will be displayed in the overview, and below you can add the packages to your transport with the > button and remove them again with the < button.



Description: Basic Course Version 1

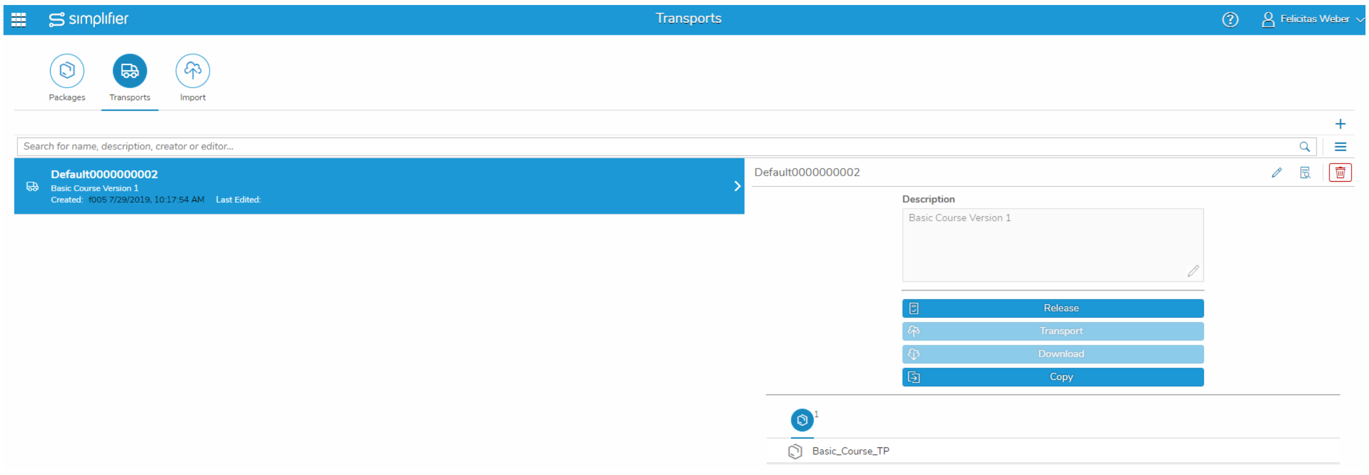
Available Packages

Package Name
Basic_Course2_TP
CRM_TP
Connector_SOAP_Order
DemoApplication_TP
Demo_Application_TP
Demo_FelicitasWeber_TP
Demo_LoginForm_TP
Demo_SOAP_OrderList_TP
Demo_TP
DigitalForm_TP

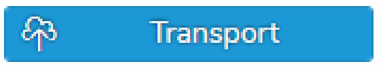
Selected Packages

Package Name
Basic_Course_TP

After you have saved your changes, you return to the overview of transports.



On the right side you have the following possibilities:



Create an OpenUI5 Widget

https://developer.simplifier.io/documentation/applications/widget-customizer/create-openui5-widget/

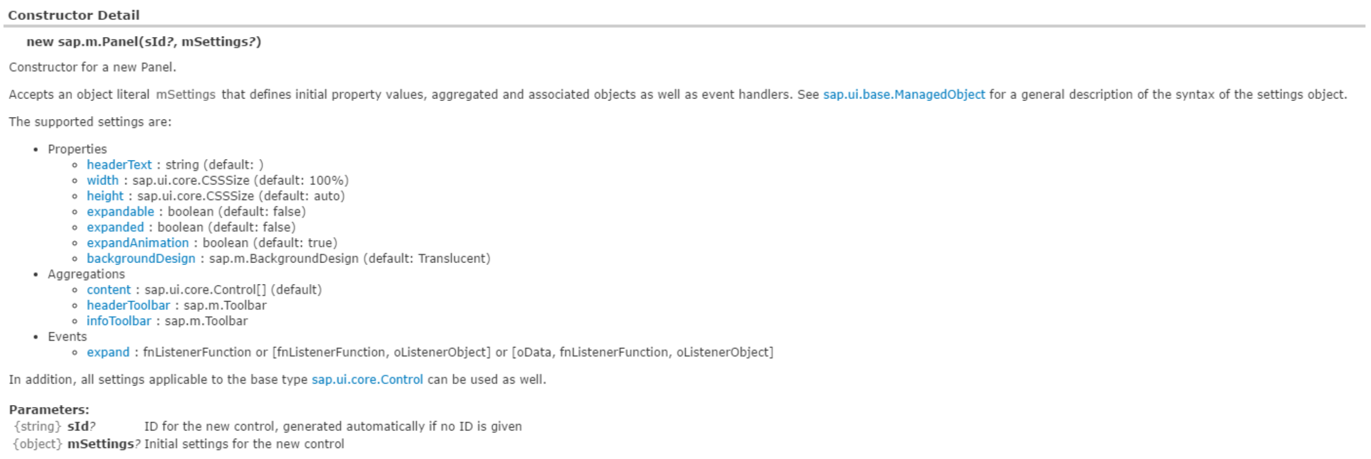
Vimeo Video

If you choose to create a new OpenUi5 Widget, you have to take a look at the constructor details in its API reference.

Here you can find the OpenUi5 API of all Widgets and a description when to use them.

Start

Let's create a mobile version of a Panel. For starters search new sap.m.Panel in the API reference.
You will see the supported settings, in this case: Properties, Aggregations and Events.



Constructor Detail

new sap.m.Panel(sId?, mSettings?)

Constructor for a new Panel.

Accepts an object literal mSettings that defines initial property values, aggregated and associated objects as well as event handlers. See sap.ui.base.ManagedObject for a general description of the syntax of the settings object.

The supported settings are:

- Properties
  - headerText : string (default:)
  - width : sap.ui.core.CSSSize (default: 100%)
  - height : sap.ui.core.CSSSize (default: auto)
  - expandable : boolean (default: false)
  - expanded : boolean (default: false)
  - expandAnimation : boolean (default: true)
  - backgroundDesign : sap.m.BackgroundDesign (default: Translucent)
- Aggregations
  - content : sap.ui.core.Control[] (default)
  - headerToolbar : sap.m.Toolbar
  - infoToolbar : sap.m.Toolbar
- Events
  - expand : fnListenerFunction or [fnListenerFunction, oListenerObject] or [oData, fnListenerFunction, oListenerObject]

In addition, all settings applicable to the base type sap.ui.core.Control can be used as well.

Parameters:

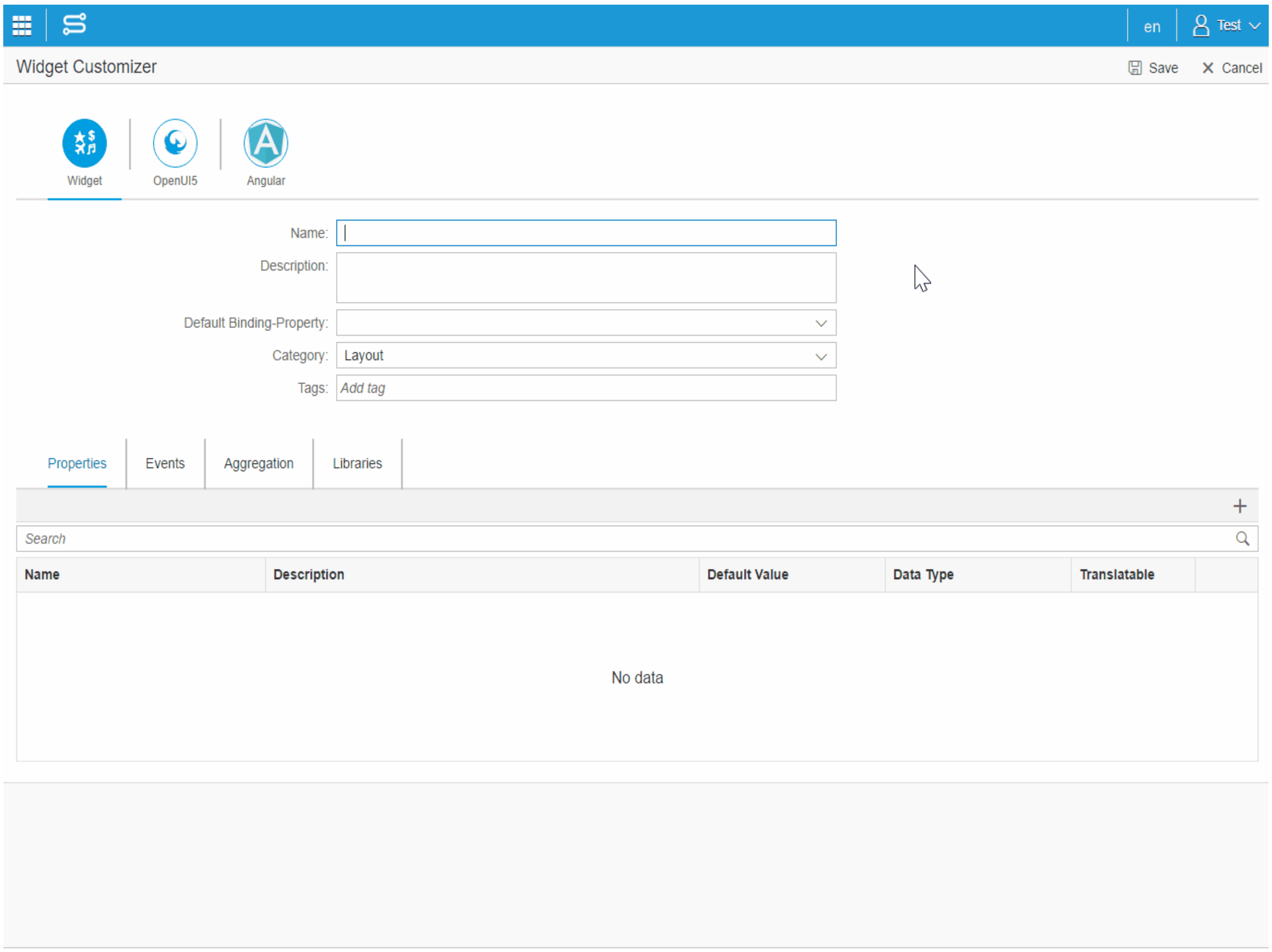
{string}	**sId**?	ID for the new control, generated automatically if no ID is given
{object}	**mSettings**?	Initial settings for the new control

- Properties describe the different attributes of an element (e.g. width or height).

- Aggregations describe which other elements the Widget could contain (e.g. a panel consists of a header & info toolbar and content).
  Depending on the Control that is displayed in the API, you can use every or just specific Controls (e.g. sap.ui.core.Control vs sap.m.Toolbar)
- Events describe the possible direct interactions for the user (e.g. expand the panel).

Step 1

For the first step name your Widget, write a short description and choose a category in the Widget tab.



Name:	
Description:	
Default Binding-Property:	
Category:	Layout
Tags:	Add tag

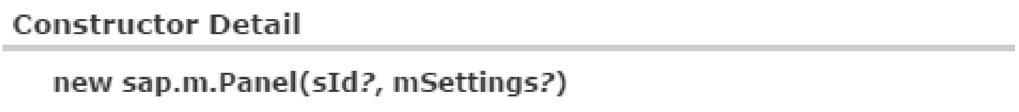
Name	Description	Default Value	Data Type	Translatable

No data

In addition you've got now the possibility to add custom tags to the widget. You can search and filter for widget tags in the search field of the widget overview list and in the widget search field of the UI designer.

Step 2

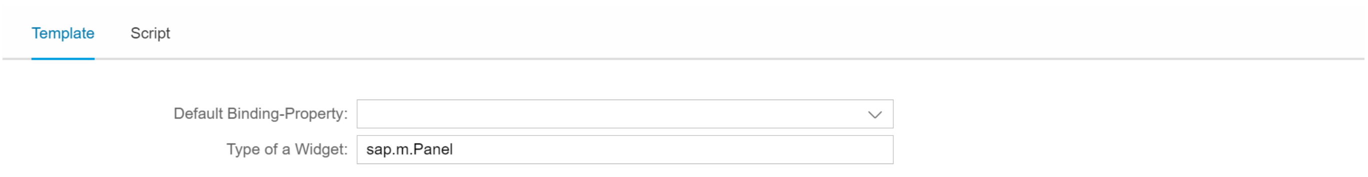
Click on the OpenUI5 tab to fill out the specific parameters.
The Widget Type has to be the same as the UI5 control name. In this case: *sap.m.Panel*



Constructor Detail

```
new sap.m.Panel(sId?, mSettings?)
```

API reference



Default Binding-Property:
Type of a Widget: sap.m.Panel

Simplifier Widget Type

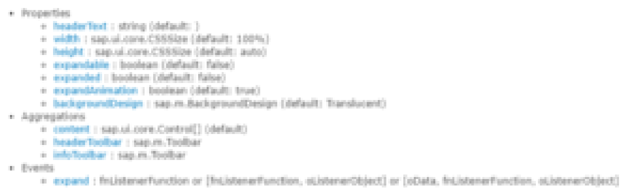
Step 3

Fill in the template for the Widget. It represents a blueprint of the Widget which can be used by the Simplifier. The OpenUI5 templates are written in JSON with Mustache placeholder syntax.

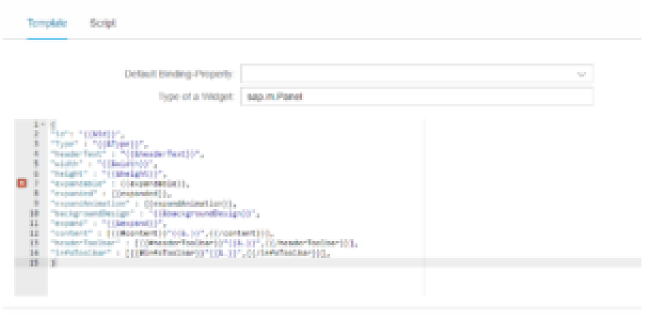
The Simplifier supports three different types:

- String: "{{&placeholderName}}"
- Boolean: {{placeholderName}}
- Aggregation: [{{#placeholderName}}"{{&}}",{{/placeholderName}}]

The placeholderName will be used to declare properties, events, etc.. It is advisable to use the OpenUI5 names. For SAP Controls (e.g. sap.m.BackgroundDesign) you can use the „String“ Data Type.



- Properties
  - headerText : string (default:)
  - width : sap.ui.core.CSSSize (default: 100%)
  - height : sap.ui.core.CSSSize (default: auto)
  - expandable : boolean (default: false)
  - expanded : boolean (default: false)
  - expandAnimation : boolean (default: true)
  - backgroundDesign : sap.m.BackgroundDesign (default: Translucent)
- Aggregations
  - content : sap.ui.core.Control[] (default)
  - headerToolbar : sap.m.Toolbar
  - infoToolbar : sap.m.Toolbar
- Events
  - expand : fnListenerFunction or [fnListenerFunction, oListenerObject] or [oData, fnListenerFunction, oListenerObject]



Default Binding-Property:
Type of a Widget: sap.m.Panel

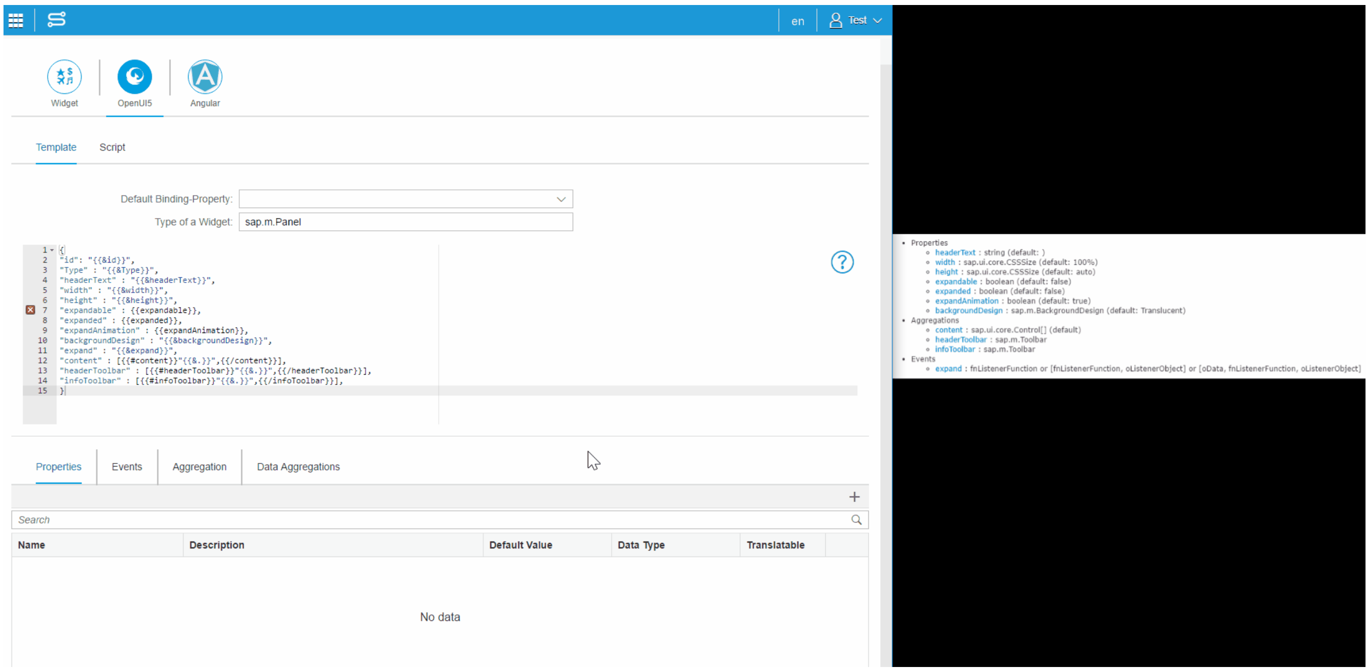
```
{
"id": "{{&id}}",
"type" : "{{&Type}}",
"headerText" : "{{&headerText}}",
"width" : "{{&width}}",
"height" : "{{&height}}",
"expandable" : {{expandable}},
"expanded" : {{expanded}},
"expandAnimation" : {{expandAnimation}},
"backgroundDesign" : "{{&backgroundDesign}}",
"expand" : "{{&expand}}",
"content" : [{{#content}}"{{&.}}",{{/content}}],
"headerToolbar" : [{{#headerToolbar}}"{{&.}}",{{/headerToolbar}}],
"infoToolbar" : [{{#infoToolbar}}"{{&.}}",{{/infoToolbar}}],
}
```

Step 4

All attributes that are declared with a mustache value in the template, have to be declared in the properties / events / etc. area below as well, so you can work with them in the UI Designer later in the process. You can use constant values instead of mustache e.g. if you don‘t want a property to be editable.

- Properties:
  Fill in the name (your template placeholderName), a description (optional), the default value and the data type (as written in the API reference). If a property should be translatable, you have to check it here.
- Events:
  Simply put the name (again the template placeholderName) on the list.

- Aggregation:
  Transfer your template placeholderName and the content type (API reference).
  If your aggregation shall be able to contain more than just one control, check the "Multiple" checkbox.



Default Binding-Property:
Type of a Widget: sap.m.Panel

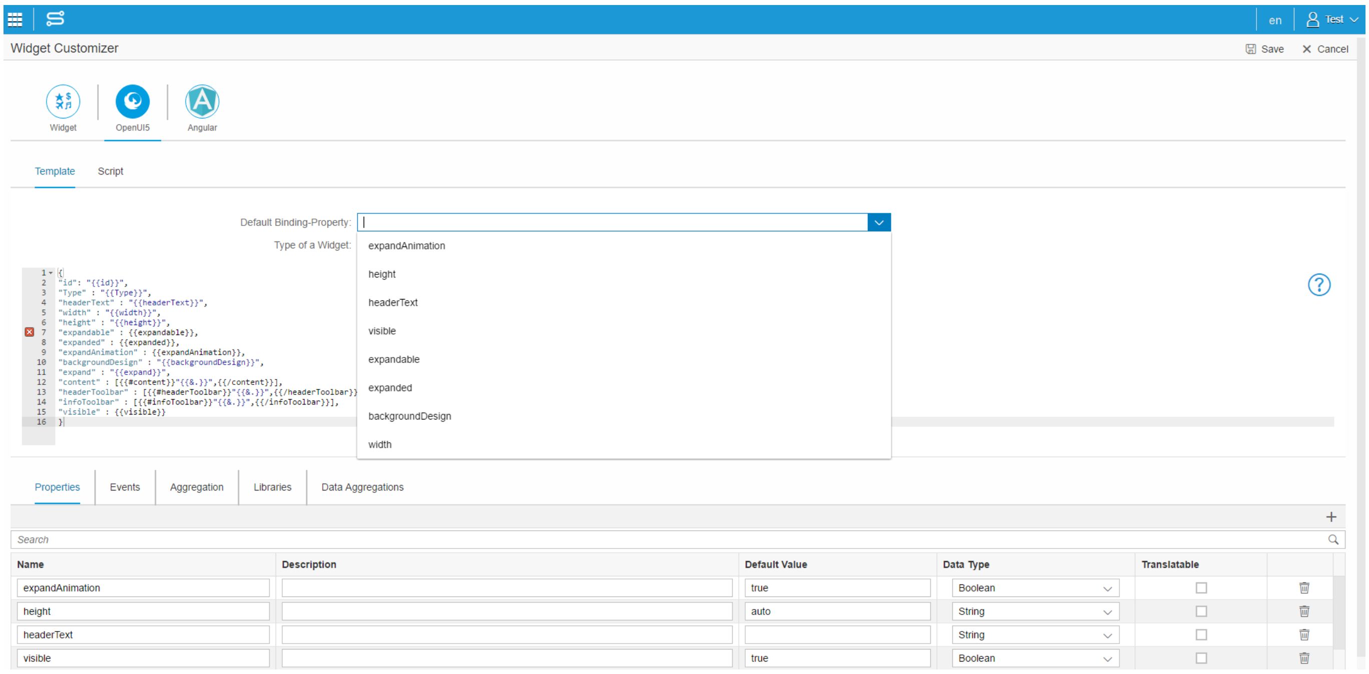
```
{
"id": "{{&id}}",
"Type" : "{{&Type}}",
"headerText" : "{{&headerText}}",
"width" : "{{&width}}",
"height" : "{{&height}}",
"expandable" : {{expandable}},
"expanded" : {{expanded}},
"expandAnimation" : {{expandAnimation}},
"backgroundDesign" : "{{&backgroundDesign}}",
"expand" : "{{&expand}}",
"content" : [{{#content}}"{{&.}}",{{/content}}],
"headerToolbar" : [{{#headerToolbar}}"{{&.}}",{{/headerToolbar}}],
"infoToolbar" : [{{#infoToolbar}}"{{&.}}",{{/infoToolbar}}],
}
```

Name	Description	Default Value	Data Type	Translatable
No data				

- Properties
  - headerText : string (default:)
  - width : sap.ui.core.CSSSize (default: 100%)
  - height : sap.ui.core.CSSSize (default: auto)
  - expandable : boolean (default: false)
  - expanded : boolean (default: false)
  - expandAnimation : boolean (default: true)
  - backgroundDesign : sap.m.BackgroundDesign (default: Translucent)
- Aggregations
  - content : sap.ui.core.Control[] (default)
  - headerToolbar : sap.m.Toolbar
  - infoToolbar : sap.m.Toolbar
- Events
  - expand : fnListenerFunction or [fnListenerFunction, oListenerObject] or [oData, fnListenerFunction, oListenerObject]

Step 5

If all properties are listed, you can set the Default Binding-Property which is the prioritized widget property used in the edit mode of a user story (Process Designer).



```
{
"id": "{{id}}",
"Type" : "{{Type}}",
"headerText" : "{{headerText}}",
"width" : "{{width}}",
"height" : "{{height}}",
"expandable" : {{expandable}},
"expanded" : {{expanded}},
"expandAnimation" : {{expandAnimation}},
"backgroundDesign" : "{{backgroundDesign}}",
"expand" : "{{expand}}",
"content" : [{{#content}}"{{&.}}",{{/content}}],
"headerToolbar" : [{{#headerToolbar}}"{{&.}}",{{/headerToolbar}}
"infoToolbar" : [{{#infoToolbar}}"{{&.}}",{{/infoToolbar}}],
"visible" : {{visible}}
}
```

Name	Description	Default Value	Data Type	Translatable	
expandAnimation		true	Boolean	☐	
height		auto	String	☐	
headerText			String	☐	
visible		true	Boolean	☐	

End

After hitting the "Save" button, you've successfully created a Panel Widget for your application.

Create and manage connector calls

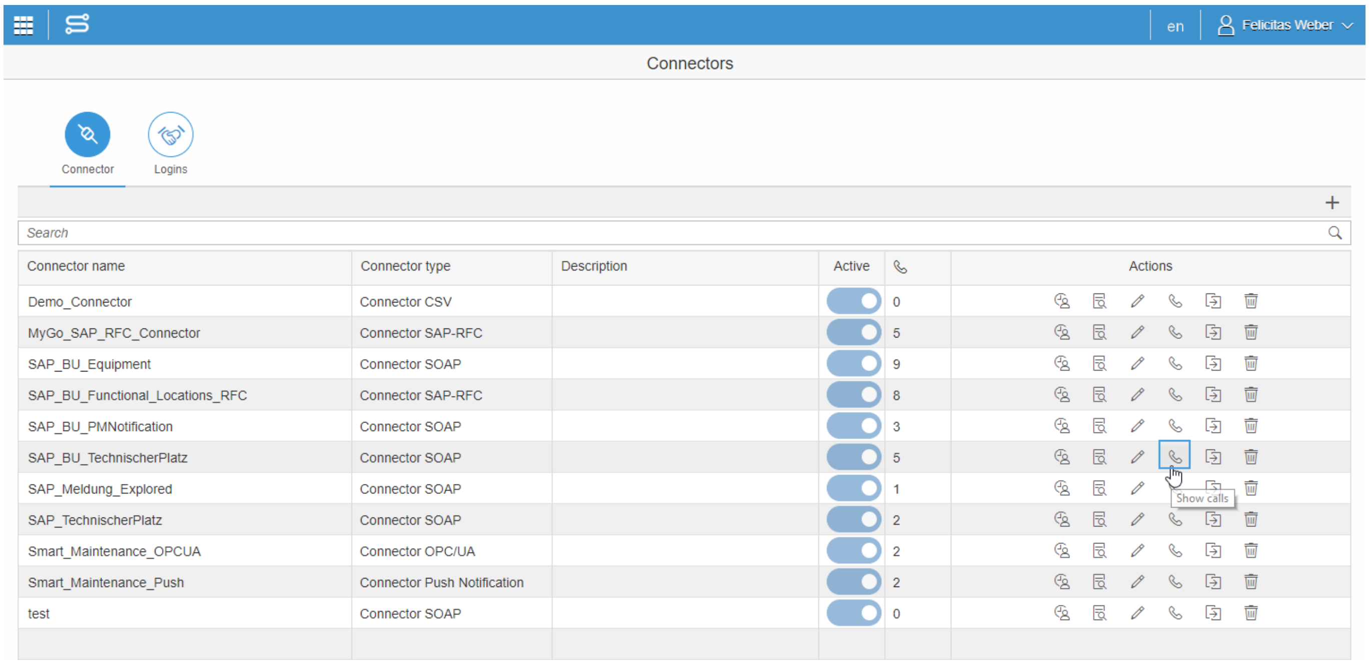
https://developer.simplifier.io/documentation/connectors/create-and-manage-connector-calls/

Vimeo Video

Connectors define the connection (entry point) to an external system. Given such a connection, you might send several different requests to the connected system. We call one such concrete pair of request / response a “connector call”. In order to use a connector call in the edit mode of a user story (Process Designer), you must create at least one connector call for each connector.

Step 1

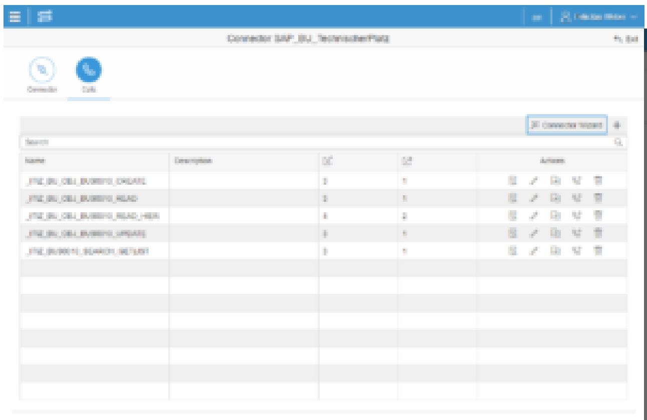
Choose a connector from the overview and click the call icon. In the call overview add a new call for each connector operation.

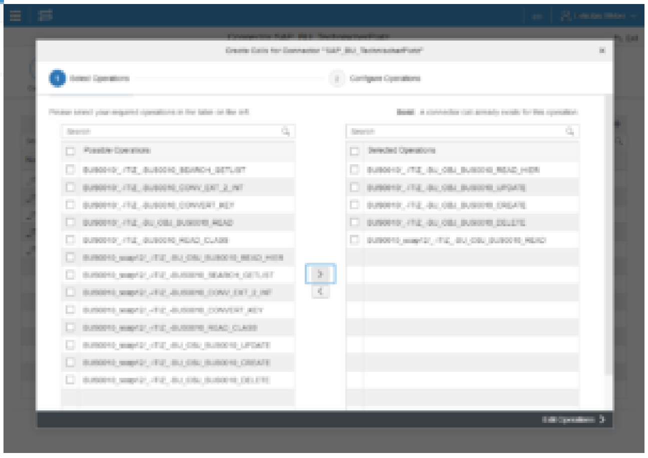


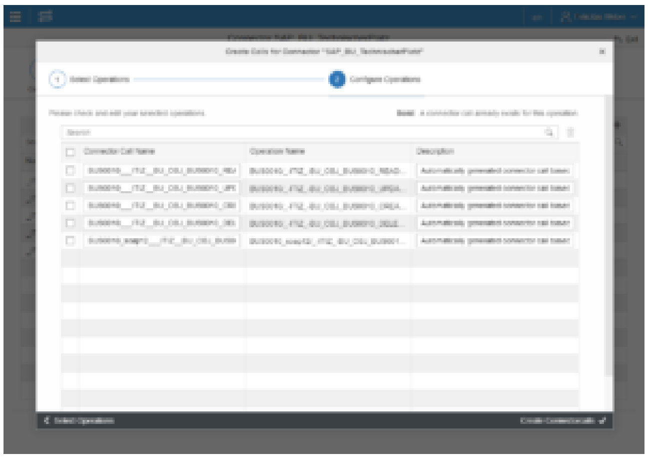
Connector name	Connector type	Description	Active	📞	Actions
Demo_Connector	Connector CSV			0	
MyGo_SAP_RFC_Connector	Connector SAP-RFC			5	
SAP_BU_Equipment	Connector SOAP			9	
SAP_BU_Functional_Locations_RFC	Connector SAP-RFC			8	
SAP_BU_PMNotification	Connector SOAP			3	
SAP_BU_TechnischerPlatz	Connector SOAP			5	
SAP_Meldung_Explored	Connector SOAP			1	
SAP_TechnischerPlatz	Connector SOAP			2	
Smart_Maintenance_OPCUA	Connector OPC/UA			2	
Smart_Maintenance_Push	Connector Push Notification			2	
test	Connector SOAP			0	

Step 2

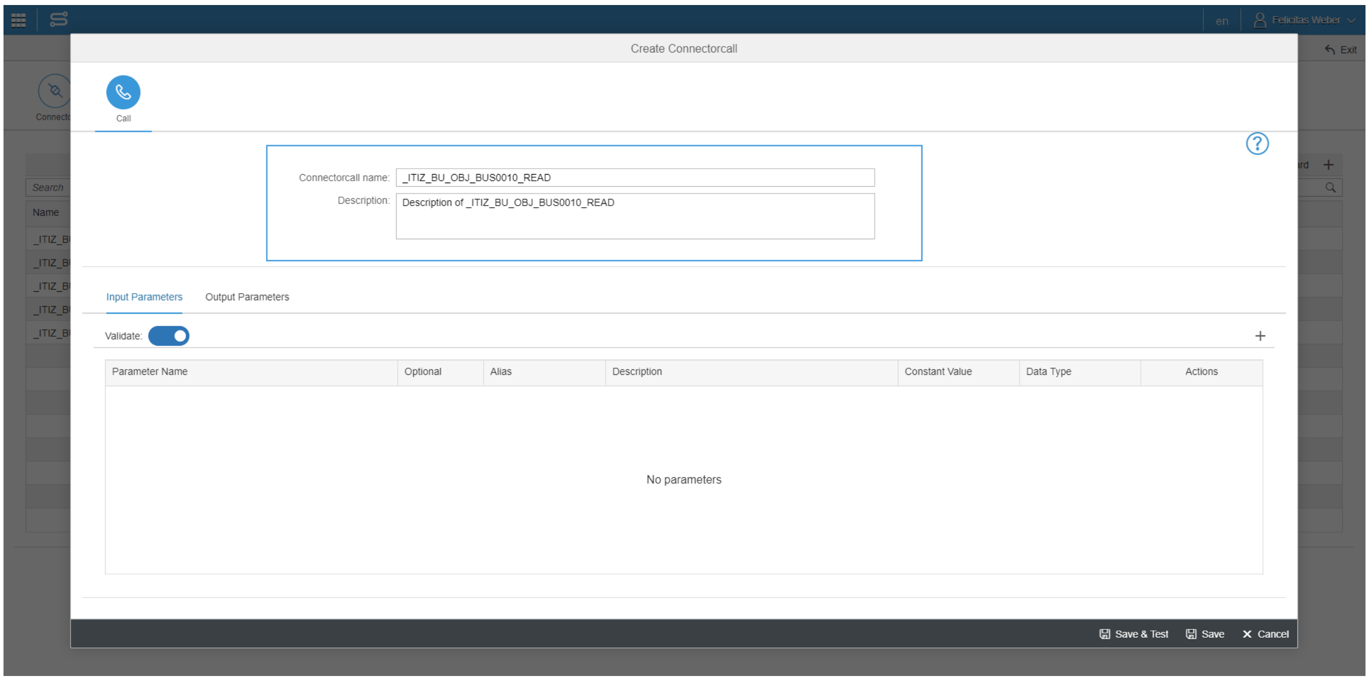
For SOAP and SQL connectors, you have the possibility to use the **Connector Wizard**. It helps you to create your connector calls much easier and faster. If you click on it, you can choose the ones that you need.







Otherwise click on the plus icon in the upper right and enter a unique call name that describes the operation (e.g. read, write, update, delete, search, ...).



Connectorcall name: _ITIZ_BU_OBJ_BUS0010_READ
Description: Description of _ITIZ_BU_OBJ_BUS0010_READ

Validate:

Parameter Name	Optional	Alias	Description	Constant Value	Data Type	Actions

No parameters

Connector Call name

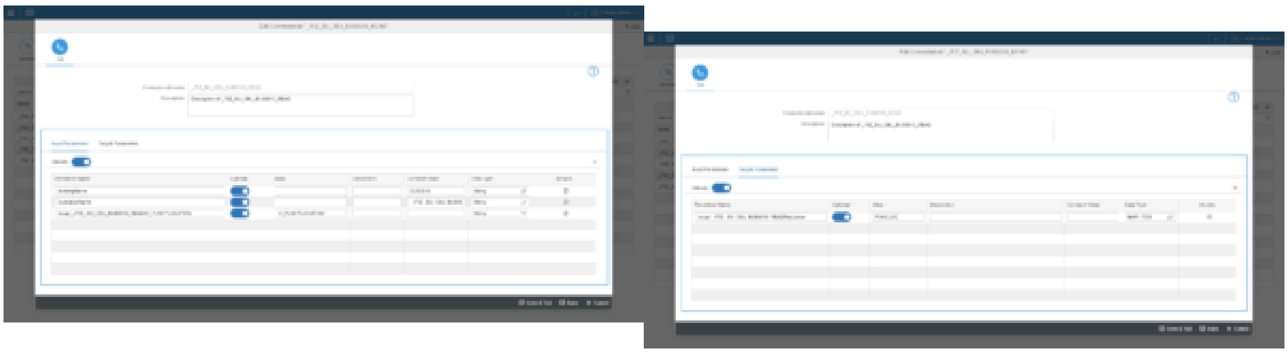
Unique name without spaces to describe the operation.

Description

Description of the operation.

Step 3

For configuring a connector call, you have to specify input and output parameters in the following tables:
Each connector call has its own specific parameters.



Validate

You can validate the Input and Output parameter in the backend. It validates:

- Base type against type security
- Domain type against security and restrictions
- Structures against type security and underlying property types
- Collections against type security and the underlying types / property categories

If the validation is **not** successful, the client is notified of all failed validations and it's written to the Connector log or System log at the same time.

For every new Connector Call, this flag is set by default. Already existing Connector Calls **do not** have this checkbox flagged to guarantee the compatibility.

Parametername

The technical path or name within a rest api definition or web service description language or csv header column.

Alias

A meaningful non-technical description for the technical parameter. This wording is used in the edit mode for a user story (Process Designer) for mapping data with ui elements.

Description

Optional description of the parameter.

Constant Value

A constant value like SAP Client or company code that can’t be overwritten by any business apps. The value will be validated, so that it's not possible to use a constant value with a wrong base type in Connector Calls and Buisness Objects.

Data Type

Assigned Simplifier data type for validating data before it gets back or from a backend system.

Step 4

After finishing the parameters, you can save the connector call settings.

Create and Manage Connectors

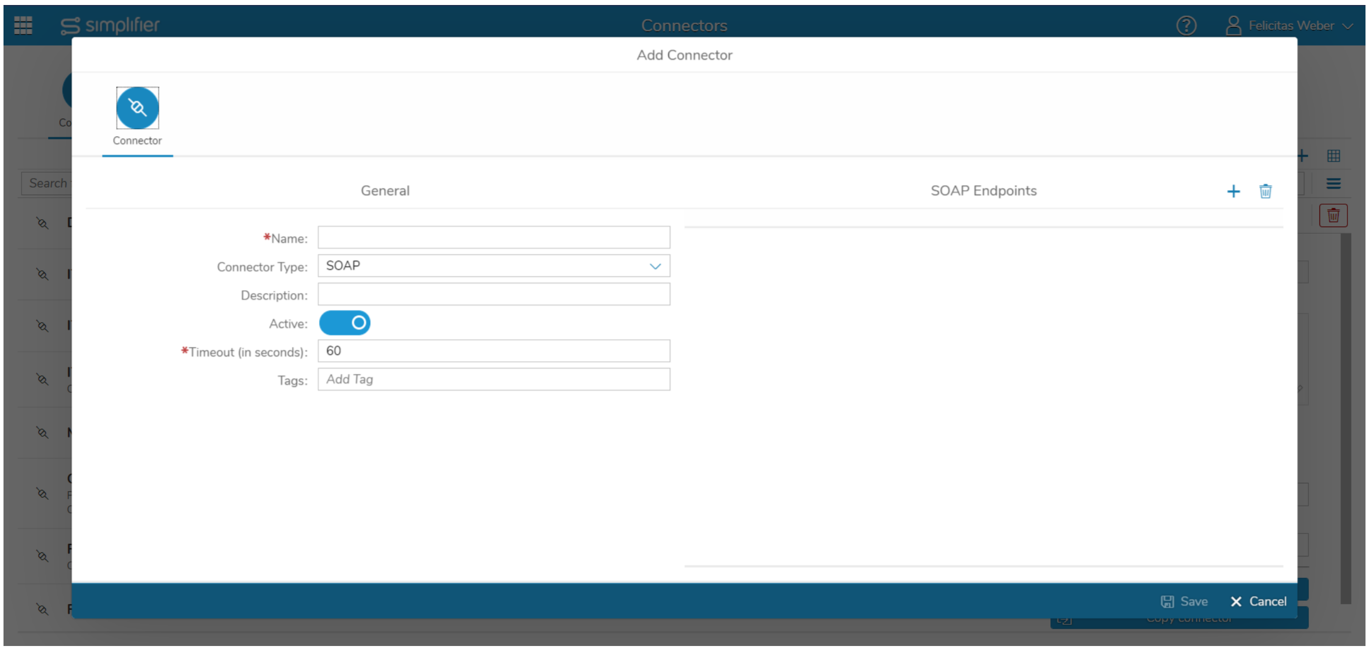
https://developer.simplifier.io/documentation/connectors/create-and-manage-connectors/

Connector Type | Login Method | Connector Details | Copy a Connector | Usage of Connector

Vimeo Video

To create a new connector, click on the plus icon on the upper right corner within the connector overview. It opens a new pop up where you can select the connector type and enter the required and optional information.

Connector Type

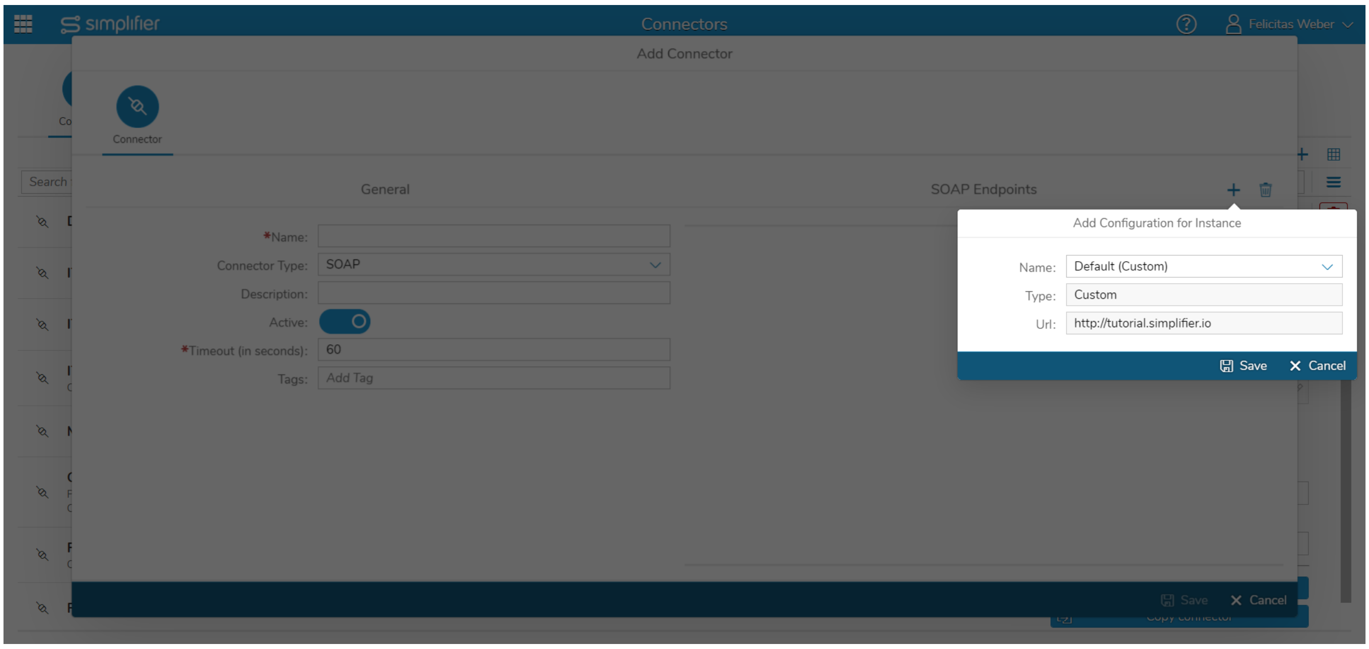


*Name:	
Connector Type:	SOAP
Description:	
Active:	●
*Timeout (in seconds):	60
Tags:	Add Tag

Name	The connector needs a unique name
Connector Type	Set the technical protocol of the interface
Description	Add a description
Active	Set the connector active. You can see within the overview which connector is active
Timeout time (in seconds)	Set the time in seconds until the connector request will run. After the set timeout, the request will be discontinued
Tags	You can add tags to your connector (e.g. the name of a project)

Endpoints

After you have created the connector information, add your endpoints by clicking the plus icon.

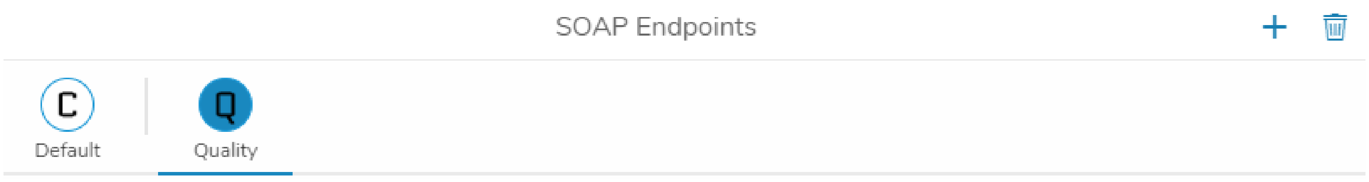


*Name:
Connector Type: SOAP
Description:
Active:
*Timeout (in seconds): 60
Tags: Add Tag

Add Configuration for Instance

Name: Default (Custom)
Type: Custom
Url: http://tutorial.simplifier.io

You can set several, but at least one is required.

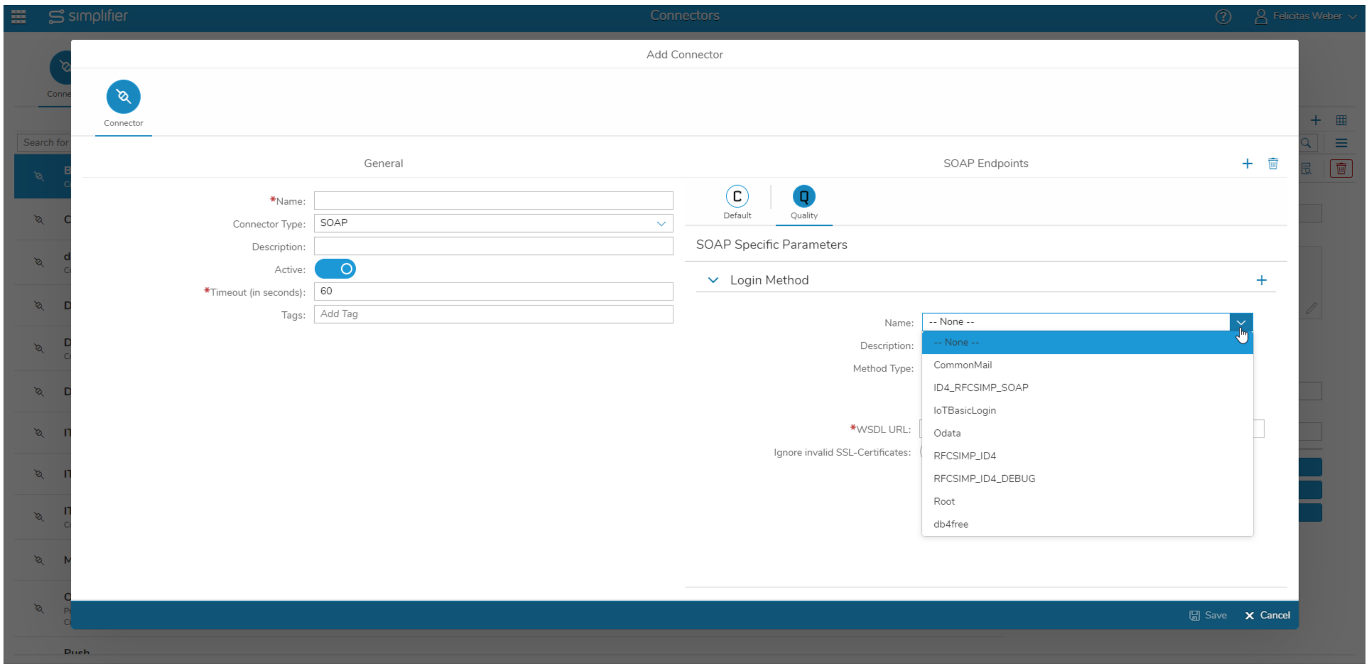


You can switch between endpoints and customize the Specific Parameters. Each connector has specific parameters that depend on the properties of the communication protocol. Read more on the following pages.

Login Method

You can add or select the login method for the specific backend systems. To select an existing login method, click on the corresponding field. It opens a drop-down where you can select it.

If you want to create a new one, you can choose between using Username/Password, Single-Sign-On, OAuth2 and SAML2.0.


Add Connector

General

*Name:
Connector Type: SOAP
Description:
Active:
*Timeout (in seconds): 60
Tags: Add Tag

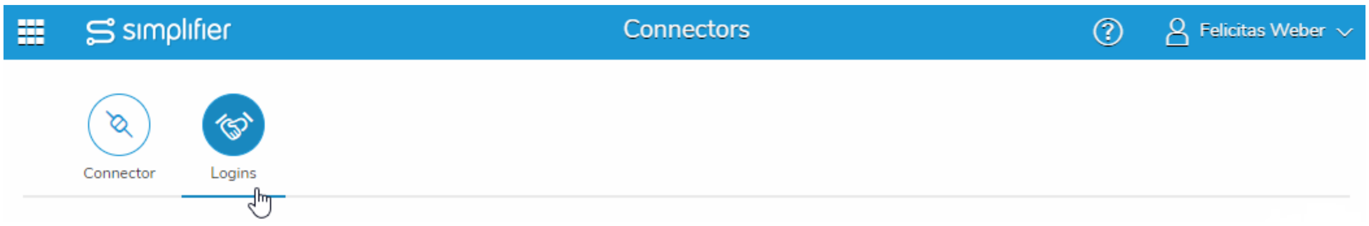
SOAP Endpoints

SOAP Specific Parameters

Login Method

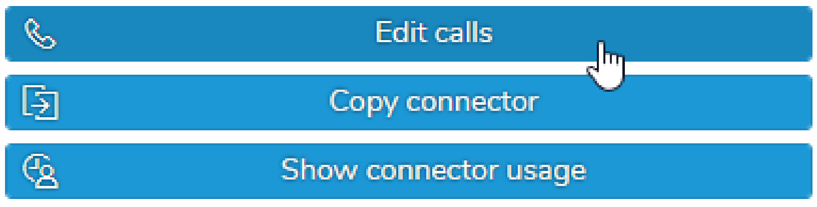
Name: -- None --
Description:
Method Type:
*WSDL URL:
Ignore invalid SSL-Certificates:

If you'd like to get an overview of your existing login methods and manage them, click on the "Logins" tab in the connector overview.



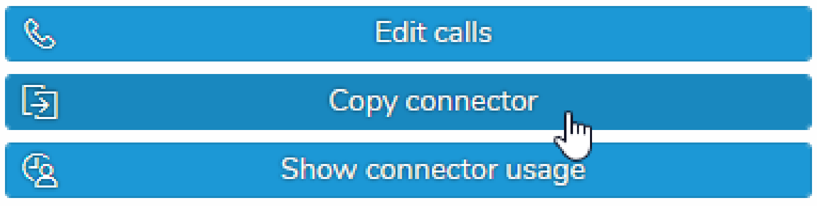
Edit Calls

By clicking on **Edit calls**, you directly jump to the overview of connector calls.

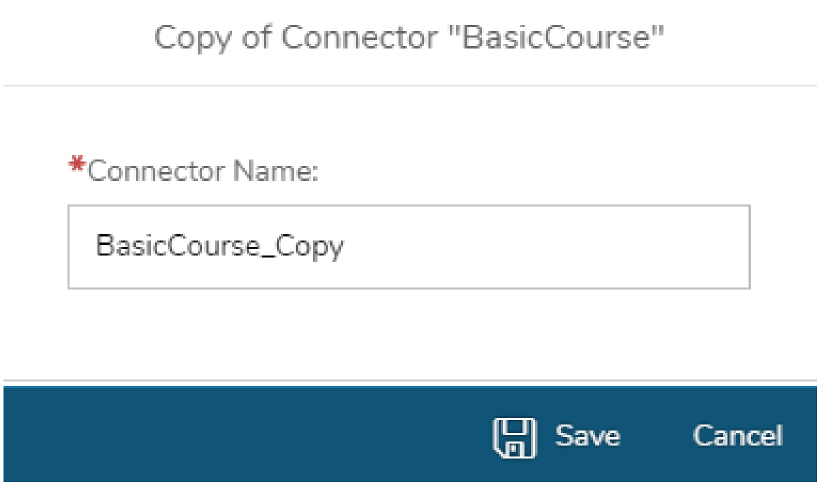


Copy a Connector

You can copy an existing connector by clicking on **Copy connector**.



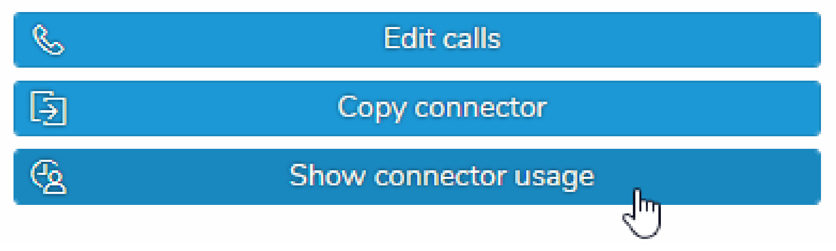
It opens the following dialog. Set a new name for the copied connector.



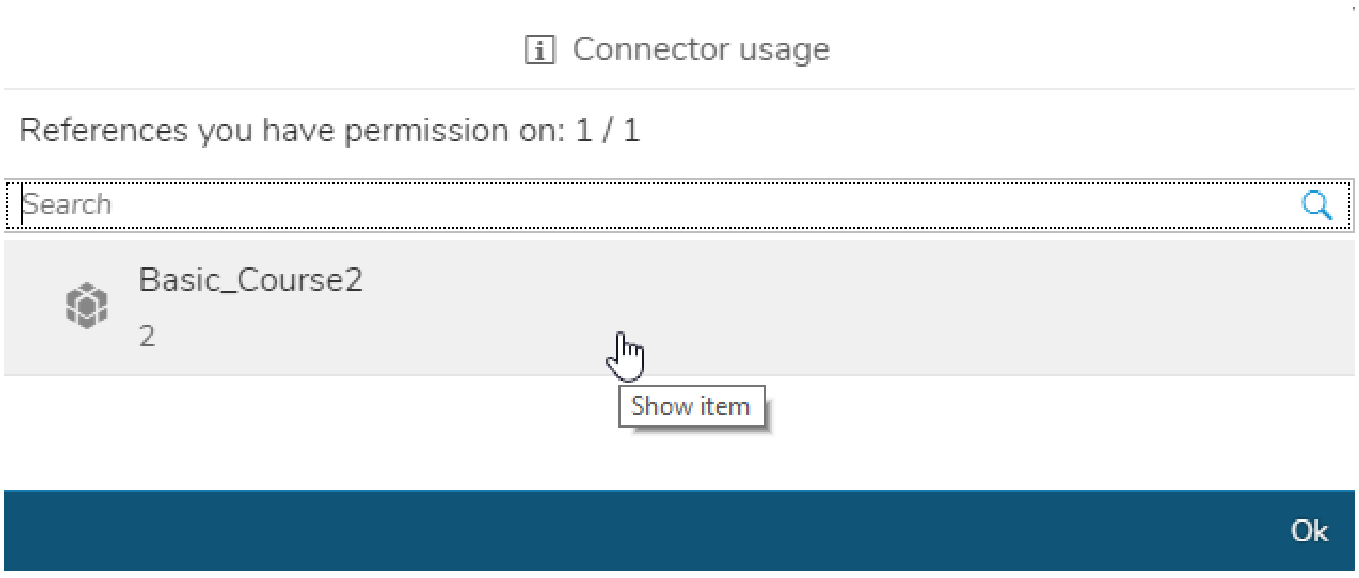
The complete configuration of the connector, including its connector calls, is copied and created with the duplicate.

Usage of Connector

You can see which artifacts are using the connector. For that, click on the appropriate icon within the connector overview underneath 'Actions'.



It opens a pop-up that displays all artifacts that use the connector:

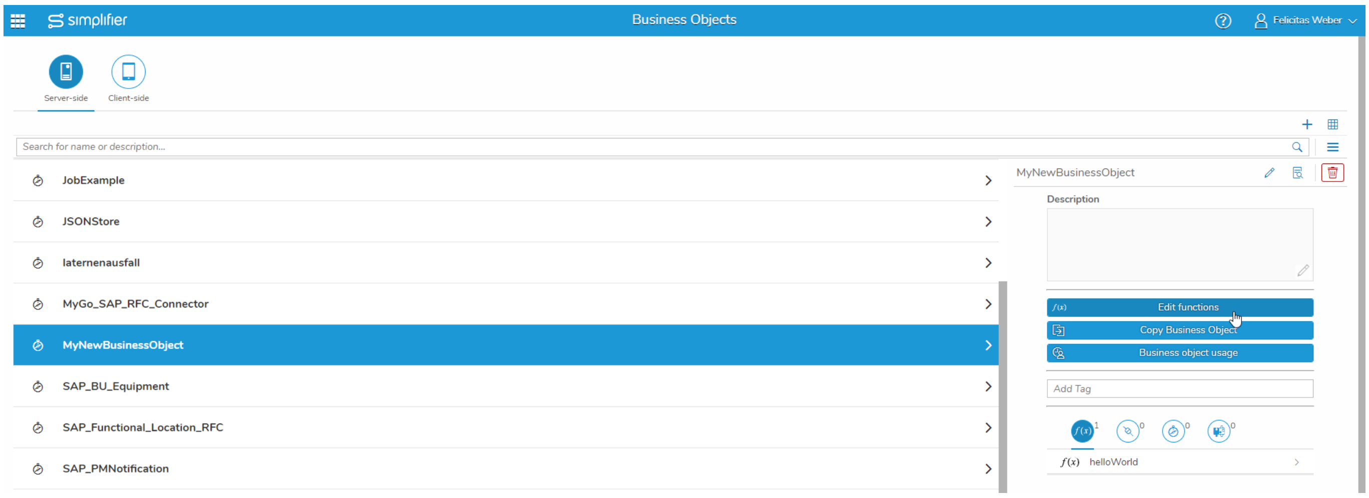


By clicking on an entry, you jump directly into the item.

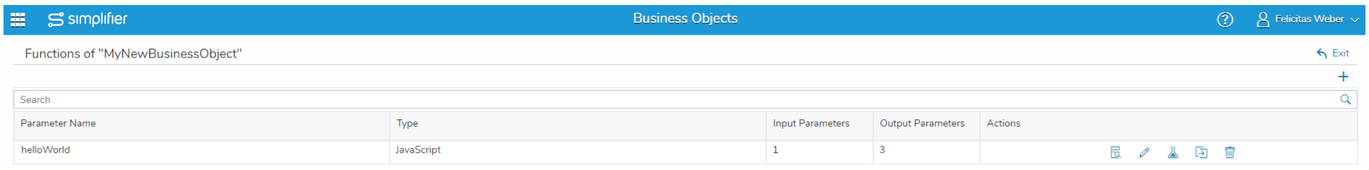
Create and Manage Functions

https://developer.simplifier.io/documentation/business-objects/create-and-manage-functions/

The logic of a business object is implemented via script functions. Each business object can hold as many functions as wanted. Click on the 'Edit functions' icon.



You are forwarded to the overview of the functions of the business object. There you can go to the function details, edit a function, test it, copy or delete it.



Functions of "MyNewBusinessObject"

Parameter Name	Type	Input Parameters	Output Parameters	Actions
helloWorld	JavaScript	1	3	

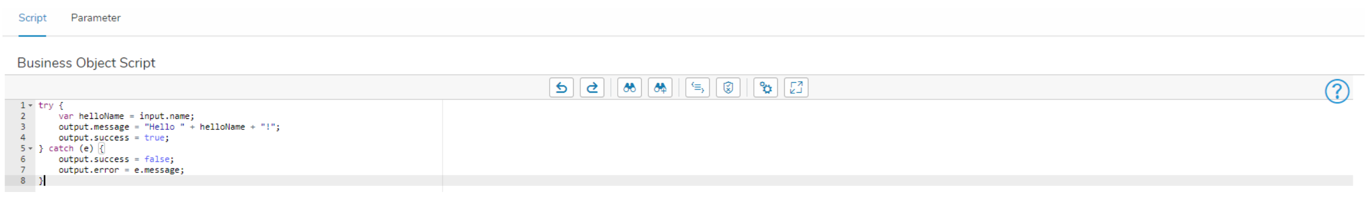
To add a new function, click on the '+'.

Creating a function involves writing a method via JavaScript and editing parameters. In the script editor, you can code [logic that can be called in the Process Designer](#).

The content corresponds to the inner body of a JavaScript function. In other words: your main code block must not be wrapped into a separate function definition but rests on the top-level context. It's nevertheless possible to define sub-functions on top of your main code block.

It's best practice to wrap your main code block in a try-catch-block to handle possible errors.

Let's have a typical hello world example:



```
try {
    var helloName = input.name;
    output.message = "Hello " + helloName + "!";
    output.success = true;
} catch (e) {
    output.success = false;
    output.error = e.message;
}
```

In the toolbar above you have several possibilities:

- undo (ctrl+z)
- redo (ctrl+y)
- search (ctrl+f)
- search and replace
- format code
- validate code
- settings (ctrl+,)
- fullscreen

This example reads a name that is provided to the business object as an input parameter, compiles it into a greeting message and writes the result to the output.

Take a look at the try-catch-block surrounding the main code section:
If no error occurs, the upper part of the code inside the “try”-section will execute and return the greeting message and output.success = true. However, if any error occurs, the function will jump down into the “catch” section and return output.success = false and assign any details of the failure to attribute output.error.

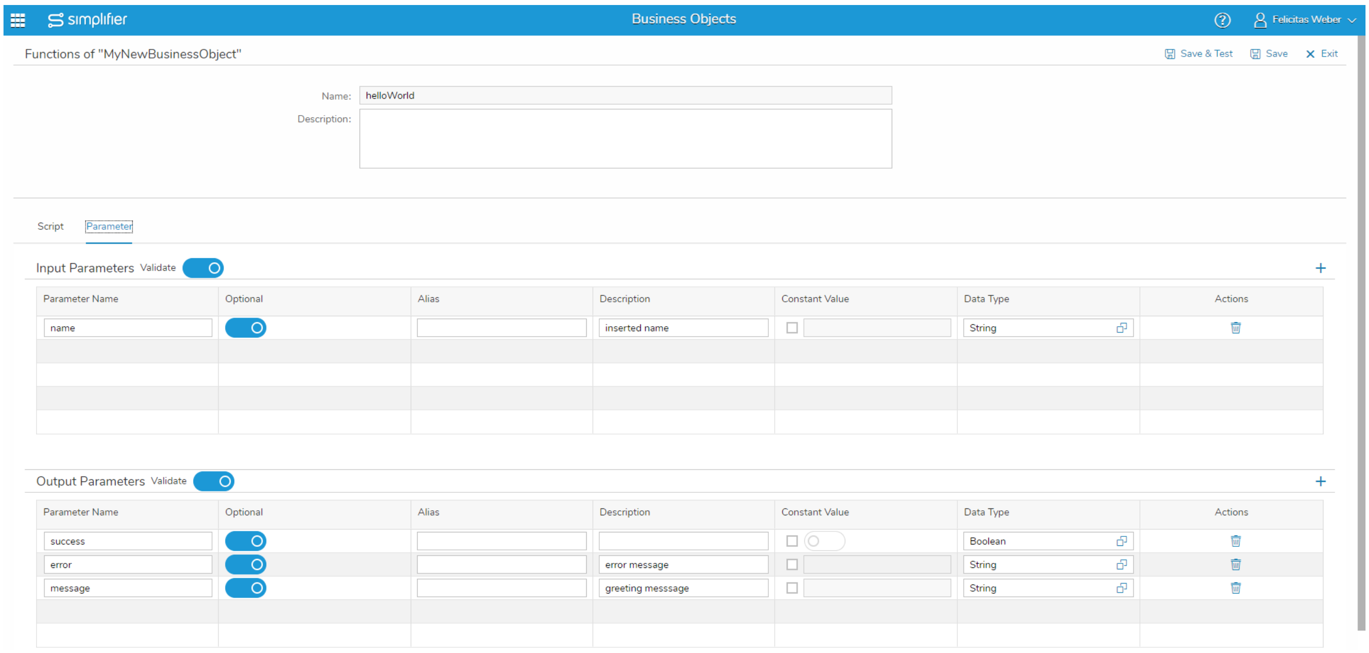
In order to use the script function correctly, you have to add the same input and output parameters you used in the payload. Those parameters will be shown in the mapping dialogs in the Process Designer. Please note the corresponding handling of in- and output through the (JSON) objects “input” and “output". Both of them may carry arbitrary attributes.

In this example the object "input" carries the attribute:

- *input.name* to read the inserted name.

The "output" object carries the attributes:

- *output.message* to send a greeting message.
- *output.success (true/false)* to indicate whether the script template executed successfully.
- *output.error* to hold the root cause of an error in case of failure.



Functions of "MyNewBusinessObject"

Name: helloWorld

Description:

Input Parameters Validate

Parameter Name	Optional	Alias	Description	Constant Value	Data Type	Actions
name	●		inserted name	☐	String	🗑

Output Parameters Validate

Parameter Name	Optional	Alias	Description	Constant Value	Data Type	Actions
success	●			☐	Boolean	🗑
error	●		error message	☐	String	🗑
message	●		greeting messsage	☐	String	🗑

Validate

You can validate the input and output parameter in the backend. It validates:

- Base type against type security
- Domain type against security and restrictions
- Structures against type security and underlying property types
- Collections against type security and the underlying types/property categories

If the validation is **not** successful, the client is notified of all failed validations and it’s written to the business object log or system log at the same time.

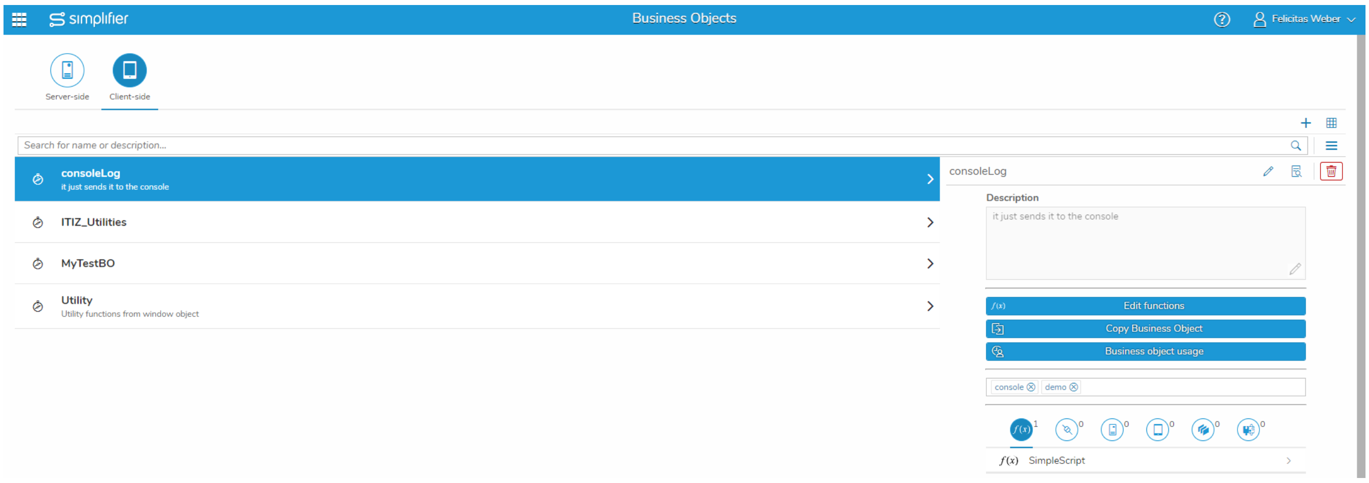
For every new business object, this flag is set by default. Already existing business objects **do not** have this checkbox flagged to guarantee the compatibility.

You have the possibility to declare parameters as optional. When declaring a parameter as non-optional, the validation will fail if the parameter is not provided.

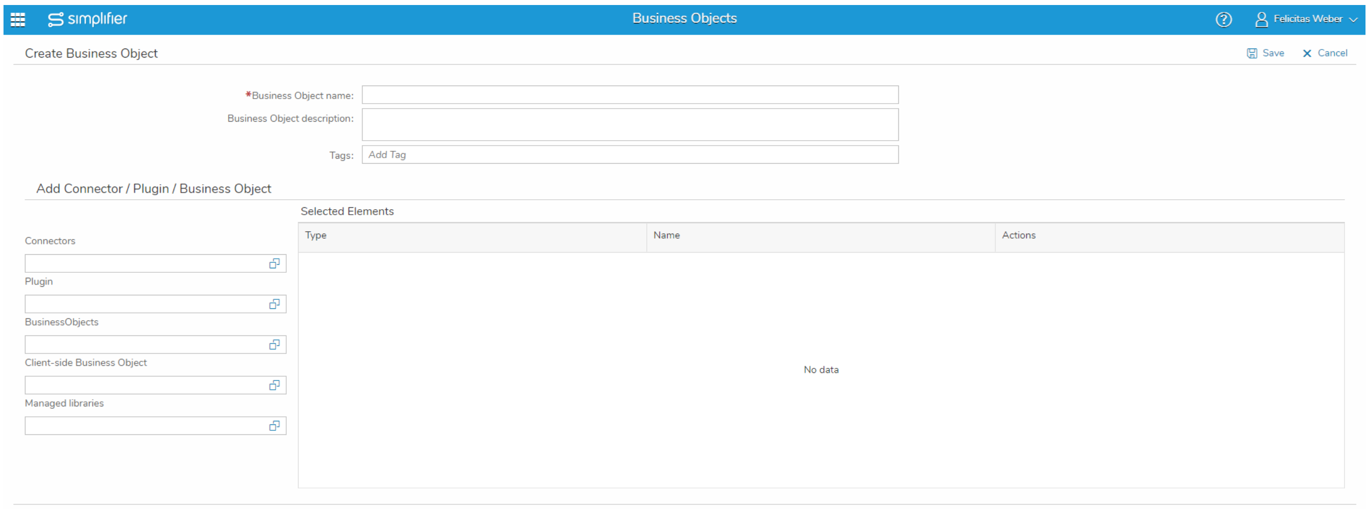
Create client-side Business Object

https://developer.simplifier.io/documentation/business-objects/create-client-side-business-object/

Business objects are managed under the module 'Business Objects'. The main screen lists all existing business objects in table form. On the top left, you can switch between server-side and client-side business objects.

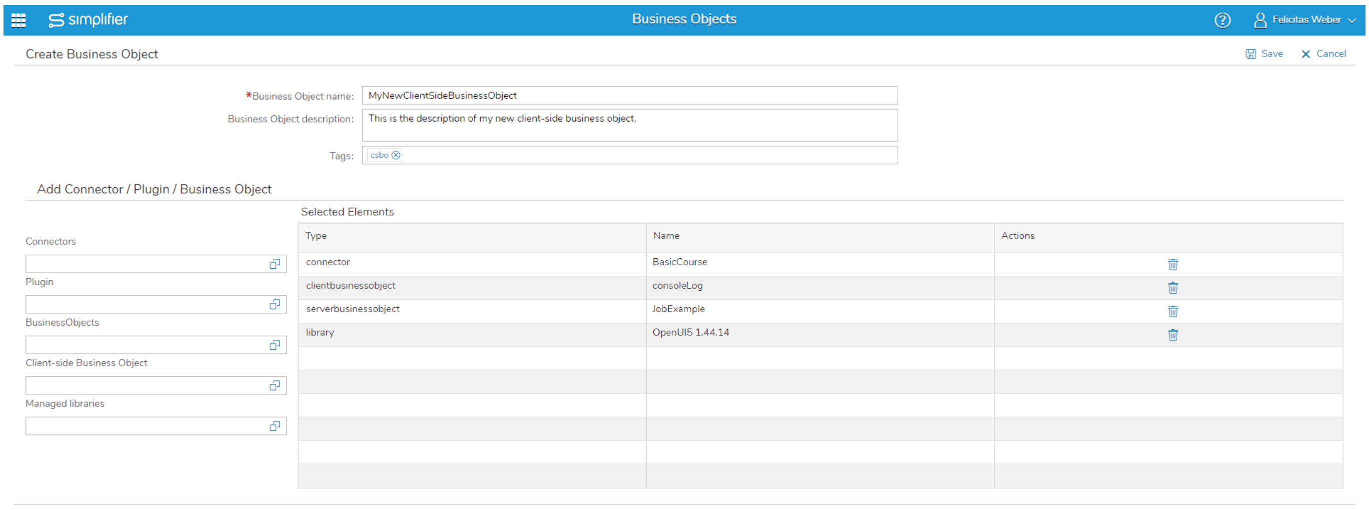


Press '+' in the upper right corner to create a new one from scratch. This fires up the following screen:



First, choose a name for your client-side business object and define a description (optional). Add some tags, so you can search in the overviews and the UI Designer by the tags.

You may then select any connector, plugin, server-side business object, client-side business object or managed libraries on the left side. It opens a dialog where you can select it. Each selected item appears in the list below, from where you might also remove it again by clicking the delete icon underneath 'Actions'.



Create Business Object

*Business Object name:	MyNewClientSideBusinessObject
Business Object description:	This is the description of my new client-side business object.
Tags:	csbo

Add Connector / Plugin / Business Object

Connectors

Plugin

BusinessObjects

Client-side Business Object

Managed libraries

Selected Elements

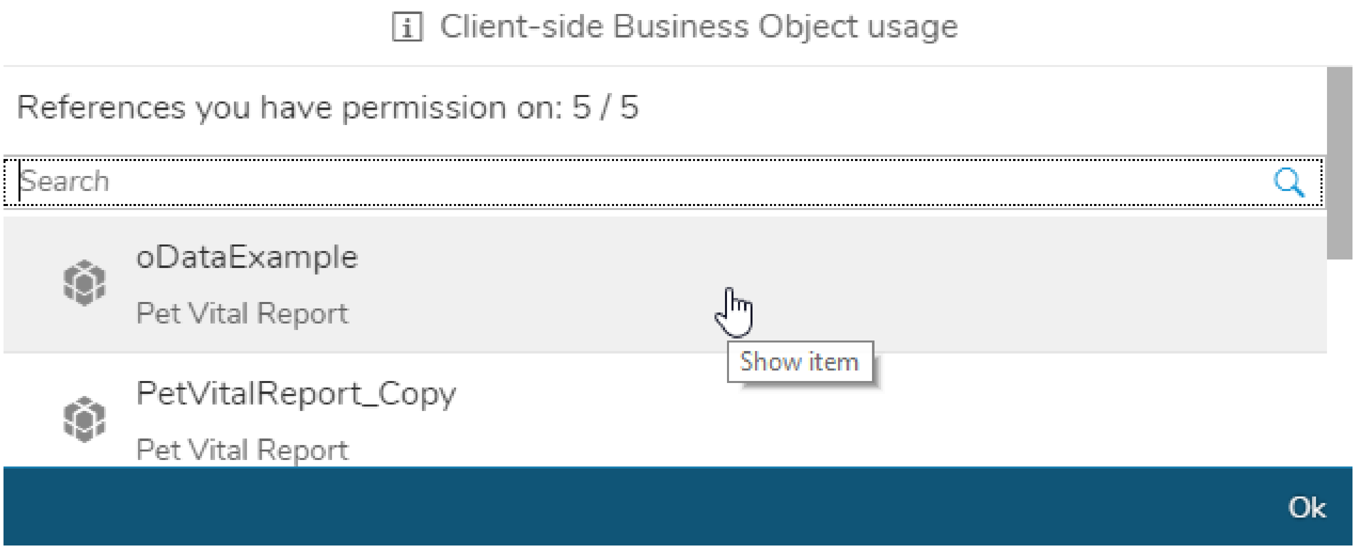
Type	Name	Actions
connector	BasicCourse	🗑
clientbusinessobject	consoleLog	🗑
serverbusinessobject	JobExample	🗑
library	OpenUI5 1.44.14	🗑

When you’re done, leave the screen by hitting the 'Save' button and return to the overview page. Your new business object appears in the table.

Usage of client-side Business Objects

You can see which applications or interfaces are using the business object. For that, click on the appropriate icon within the business object overview underneath ‘Actions’.

It opens a popup that displays all used applications:



By clicking on an entry, the application opens so that you can edit it directly.

Accessing input and output parameters of client-side Business Objects

You can access your input parameters via oPayload.<myInputParameter>.

To use the output parameters you have to return an object that has your parameters as properties. E.g.

```
return {
      myOutputParameter : myOutputValue
}
```

As client-side Business Objects and their contents are called asynchronously, it may happen when a connector call is called that it is not yet finished and is returned undefined or null.

To avoid this, you must call fnSuccess instead.

```
fnSuccess ({
      myOutputParameter : myOutputValue
})
```

It must be called in your last callback/function of your client-side Business Object and returns the data.

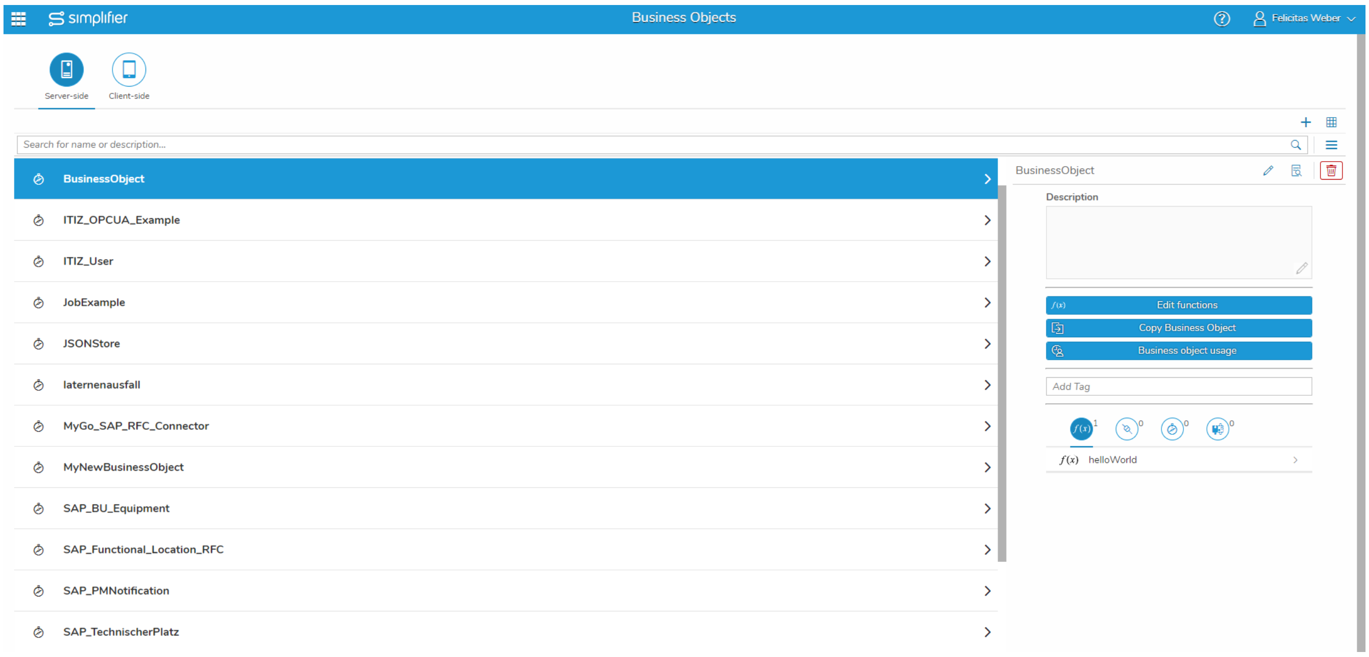
In the case of an error, the following can be specified:

```
fnError ({
      myErrorMessage : myErrorMessageValue
})
```

Create server-side Business Objects

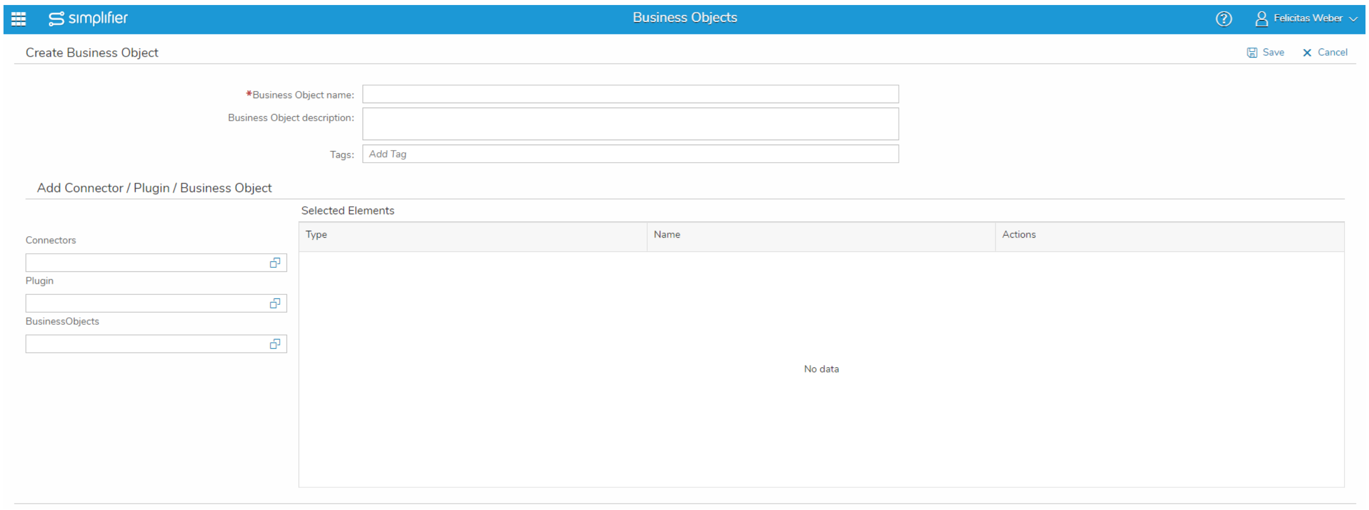
https://developer.simplifier.io/documentation/business-objects/create-business-objects/

Business objects are managed in the tile 'Business Objects'. The main screen lists all existing business objects in table form. On the top left, you can switch between server-side and client-side business objects.

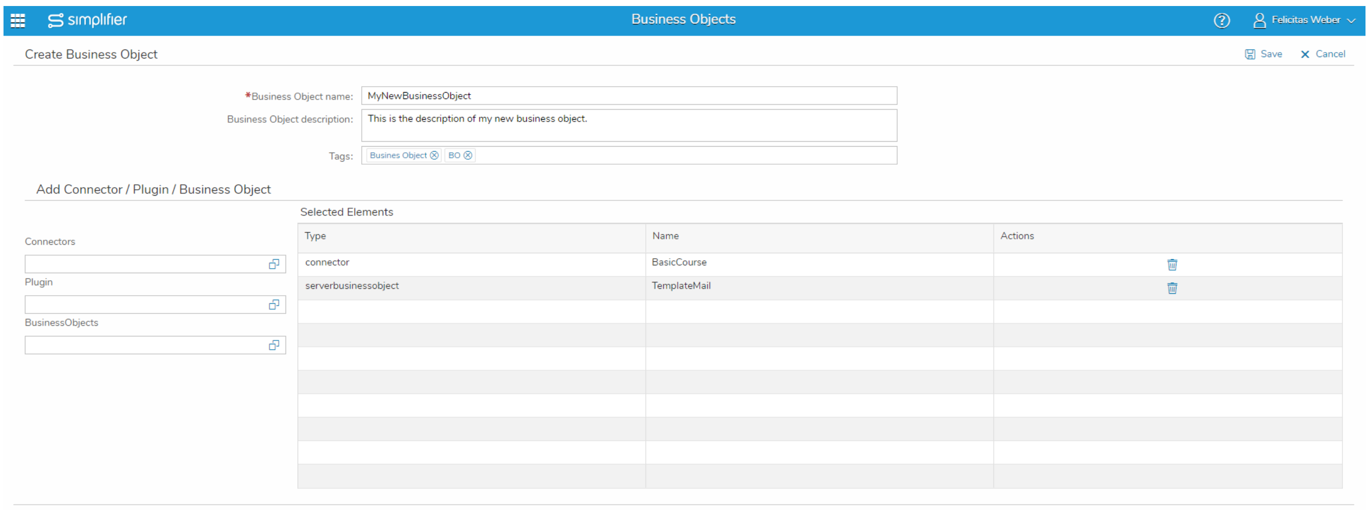


simplifier
Business Objects
Felicitas Weber
Server-side
Client-side
Search for name or description...
BusinessObject
ITIZ_OPCUA_Example
ITIZ_User
JobExample
JSONStore
laternenausfall
MyGo_SAP_RFC_Connector
MyNewBusinessObject
SAP_BU_Equipment
SAP_Functional_Location_RFC
SAP_PMNotification
SAP_TechnischerPlatz
BusinessObject
Description
Edit functions
Copy Business Object
Business object usage
Add Tag
helloWorld

Press '+' in the upper right corner to create a new one from scratch. This fires up the following screen:



First, choose a name for your business object and define a description (optional). Add some tags, so you can search in the overviews and the UI Designer by the tags.
You may then select any connector, plugin or other business objects you want to refer on the left side. It opens a dialog where you can select it. Each selected item appears in the list below, from where you might also remove it again by clicking the delete icon underneath ‚Actions‘.



*Business Object name:	MyNewBusinessObject
Business Object description:	This is the description of my new business object.
Tags:	Busines Object, BO

Add Connector / Plugin / Business Object

Connectors

Plugin

BusinessObjects

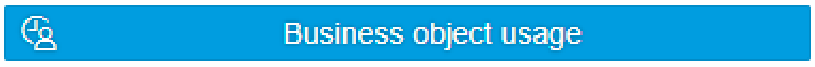
Selected Elements

Type	Name	Actions
connector	BasicCourse	
serverbusinessobject	TemplateMail	

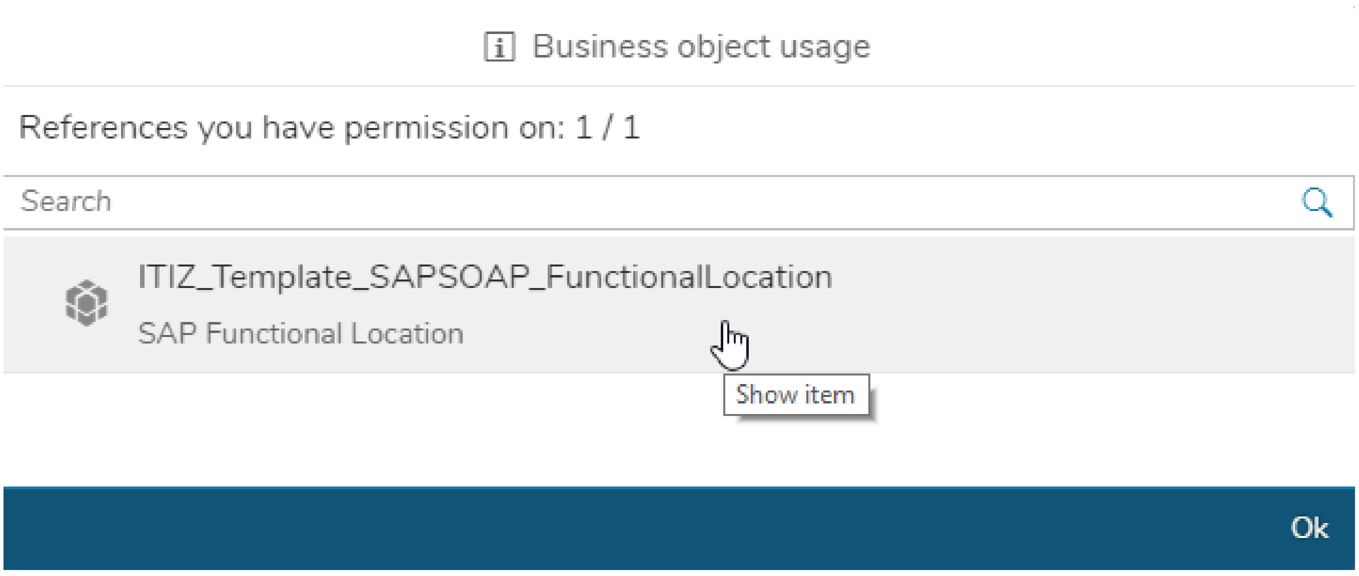
When you’re done, leave the screen by hitting the 'Save' button and return to the overview page. Your new business object appears in the table.

Usage of server-side Business Objects

You can see which applications or interfaces are using the business object. For that, click on the appropriate icon on the right.



It opens a popup that displays all used applications:


Business object usage
References you have permission on: 1 / 1
Search
ITIZ_Template_SAPSOAP_FunctionalLocation
SAP Functional Location
Show item
Ok

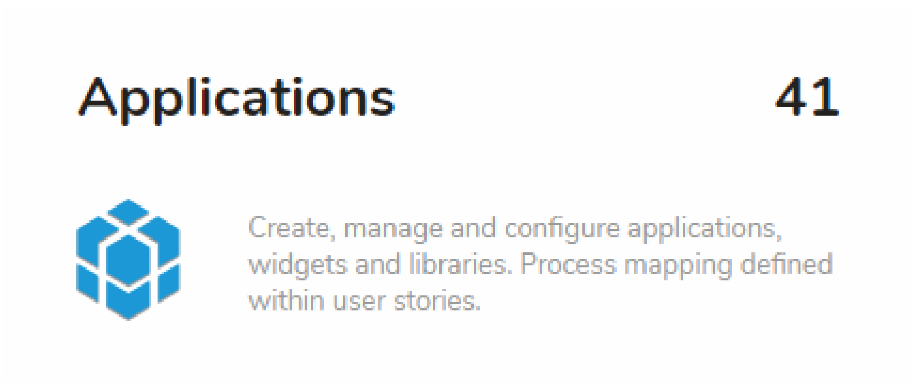
By clicking on an entry, the application opens so that you can edit it directly.

Create your first Application

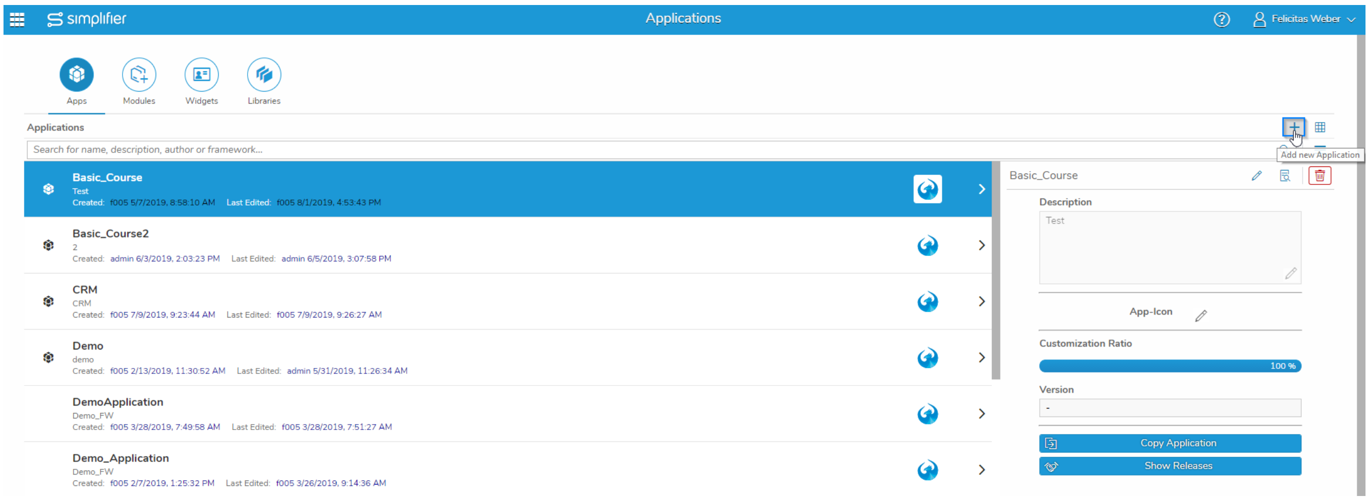
https://developer.simplifier.io/documentation/getting-started/create-your-first-application/

Vimeo Video

To create a new application, click on the Applications tile.


Applications
41
Create, manage and configure applications, widgets and libraries. Process mapping defined within user stories.

Then click on the "+" button on the top right to create a new business app:



Applications

Search for name, description, author or framework...

Basic_Course
Test
Created: f005 5/7/2019, 8:58:10 AM Last Edited: f005 8/1/2019, 4:53:43 PM

Basic_Course2
2
Created: admin 6/3/2019, 2:03:23 PM Last Edited: admin 6/5/2019, 3:07:58 PM

CRM
CRM
Created: f005 7/9/2019, 9:23:44 AM Last Edited: f005 7/9/2019, 9:26:27 AM

Demo
demo
Created: f005 2/13/2019, 11:30:52 AM Last Edited: admin 5/31/2019, 11:26:34 AM

DemoApplication
Demo_FW
Created: f005 3/28/2019, 7:49:58 AM Last Edited: f005 3/28/2019, 7:51:27 AM

Demo_Application
Demo_FW
Created: f005 2/7/2019, 1:25:32 PM Last Edited: f005 3/26/2019, 9:14:36 AM

Basic_Course

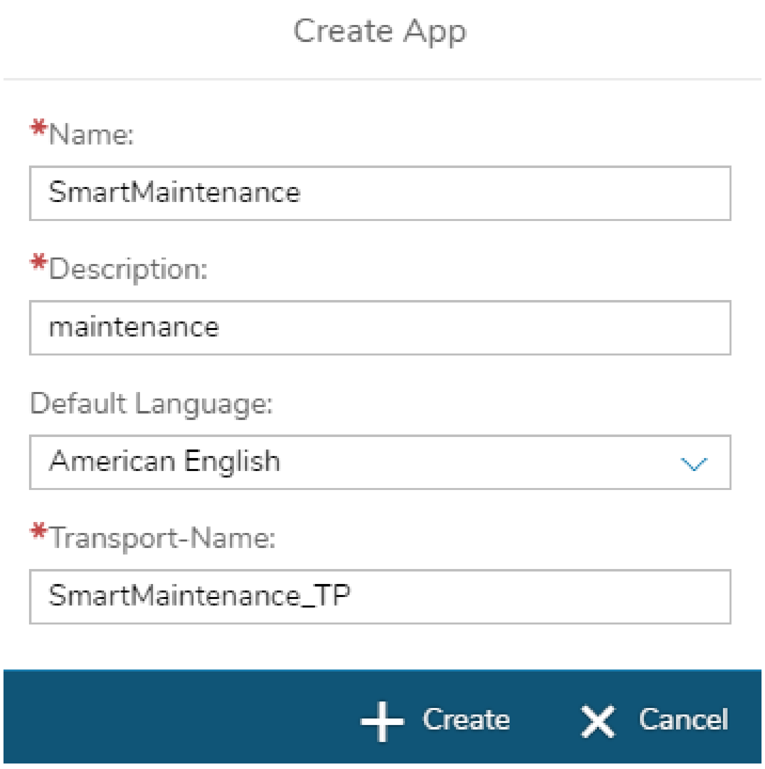
Description
Test

App-Icon

Customization Ratio
100 %

Version
-

The following dialog will appear:


Create App

*Name:
SmartMaintenance

*Description:
maintenance

Default Language:
American English

*Transport-Name:
SmartMaintenance_TP

Fill out the necessary fields analog to the table below:

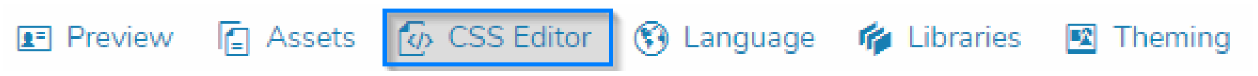
App-Name	Unique app name like “SmartMaintenance”
App-Description	A short description of your app
Default Language	Your default language for configuration - you can translate this language later via the language translation feature
App-Transport-Name	The name of the transport for transferring your configuration to another Simplifier instance like a quality assurance system

After the app creation you can configure it with the following tabs:

	UI Designer	Configure the user interface and mark all necessary events to design your workflow.
	Process Dashboard	Define your workflow with simple visual events and actions.

CSS Editor

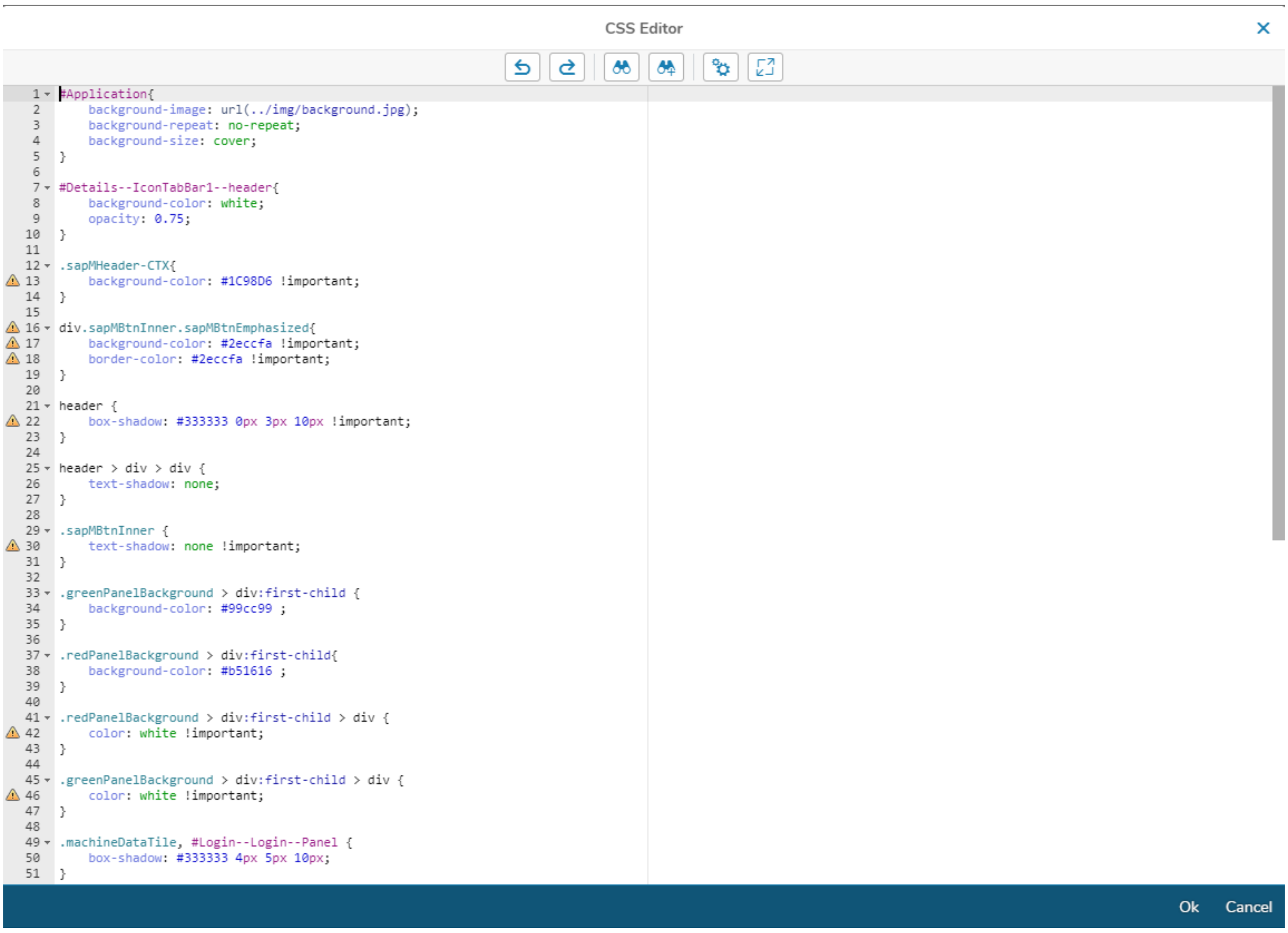
https://developer.simplifier.io/documentation/applications/ui-designer/css-editor/



To design your application, you can use CSS for styling with standard web cascading style sheets.

UI5 and CSS: Please be aware to use CSS directives mainly for design purposes, not for basic layouting to preserve the responsiveness of UI5. For design purposes, we recommend using the theming functionality of UI5. Therefore you can use the Theme Designer.

Warning: The syntax for accessing widgets by ID with CSS is: #<screenId>--<widgetId>



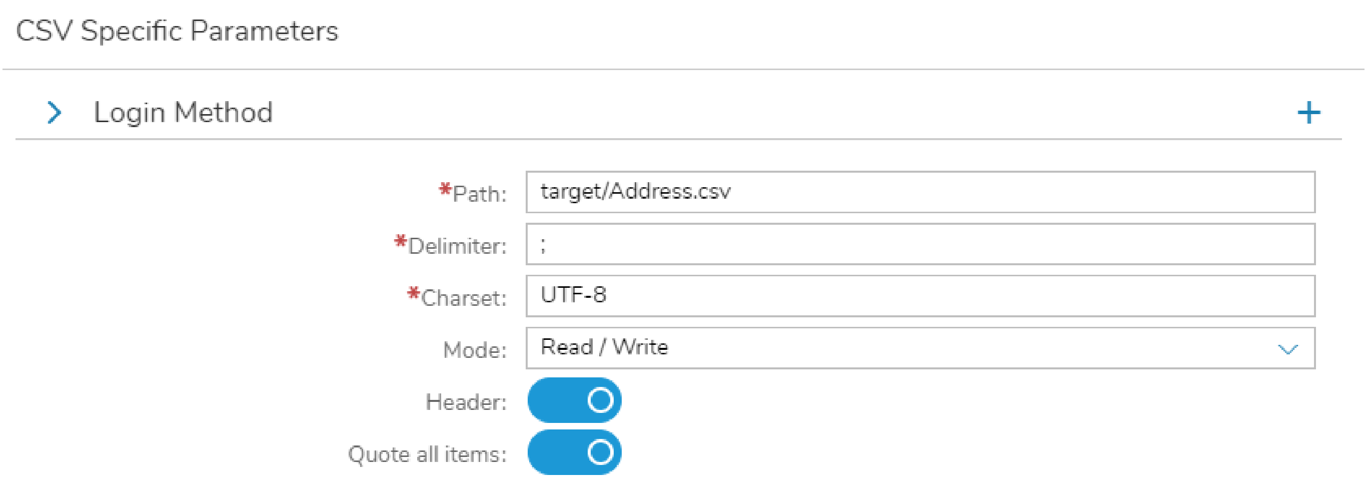
```css
#Application{
    background-image: url(../img/background.jpg);
    background-repeat: no-repeat;
    background-size: cover;
}

#Details--IconTabBar1--header{
    background-color: white;
    opacity: 0.75;
}

.sapMHeader-CTX{
    background-color: #1C98D6 !important;
}

div.sapMBtnInner.sapMBtnEmphasized{
    background-color: #2eccfa !important;
    border-color: #2eccfa !important;
}

header {
    box-shadow: #333333 0px 3px 10px !important;
}

header > div > div {
    text-shadow: none;
}

.sapMBtnInner {
    text-shadow: none !important;
}

.greenPanelBackground > div:first-child {
    background-color: #99cc99 ;
}

.redPanelBackground > div:first-child{
    background-color: #b51616 ;
}

.redPanelBackground > div:first-child > div {
    color: white !important;
}

.greenPanelBackground > div:first-child > div {
    color: white !important;
}

.machineDataTile, #Login--Login--Panel {
    box-shadow: #333333 4px 5px 10px;
}
```

After changing CSS properties you have to deploy your application.

If you would like to learn more about CSS, take a look at the tutorials of the w3 schools or Codecademy.

CSV Connector

https://developer.simplifier.io/documentation/connectors/csv-connector-details/


CSV Specific Parameters

Login Method

*Path:	target/Address.csv
*Delimiter:	;
*Charset:	UTF-8
Mode:	Read / Write
Header:	●
Quote all items:	●

Path
Filepath and Filename to local .CSV File that should be written, relative to the current working directory of the application server. It is recommended to give an absolute path, so it doesn't matter which directory is set as "Current Working Directory" from the app server start script.

If you want to provide the files by the Html5 uploader you have to specify the path to the uploads directory. By default the path to the upload directory is

/opt/simplifier/data/storage/uploads.

Please consider: If you have defined a different path in your Simplifier configuration file (settings.conf) or in your docker environment, then change the path accordingly.

Delimiter

Delimiter of the columns that separate the values like comma or semicolon. This must be exactly one character, more than one character is not supported by the library.
In order to use the tabulator character, the expression '\t' can be used in the Admin UI. If more than one character is specified, all but the first character will be discarded.

Charset

The character encoding used to read/write the file. If a charset is used that is unknown to the application server JVM, all read/write operations will fail.

Mode

Operation Mode of the Connector, either READ, WRITE or READ/WRITE - the CSV Connector can currently only read the referenced CSV file.

Header

Activate the checkbox if the CSV File has a header in the first row.

Quote all Items

Activate the checkbox if all items should be quoted in terms of strings ("). Otherwise only values that contain the delimiter are put in quotes. This setting is ignored when reading.

Go to CSV Connector Calls to configure the corresponding Calls.

CSV Connector Calls

https://developer.simplifier.io/documentation/connectors/csv-connector-details/csv-connector-calls/

Go to CSV Connector Details for more information about the CSV Connector.

A CSV Connector can be configured in 3 different modes:

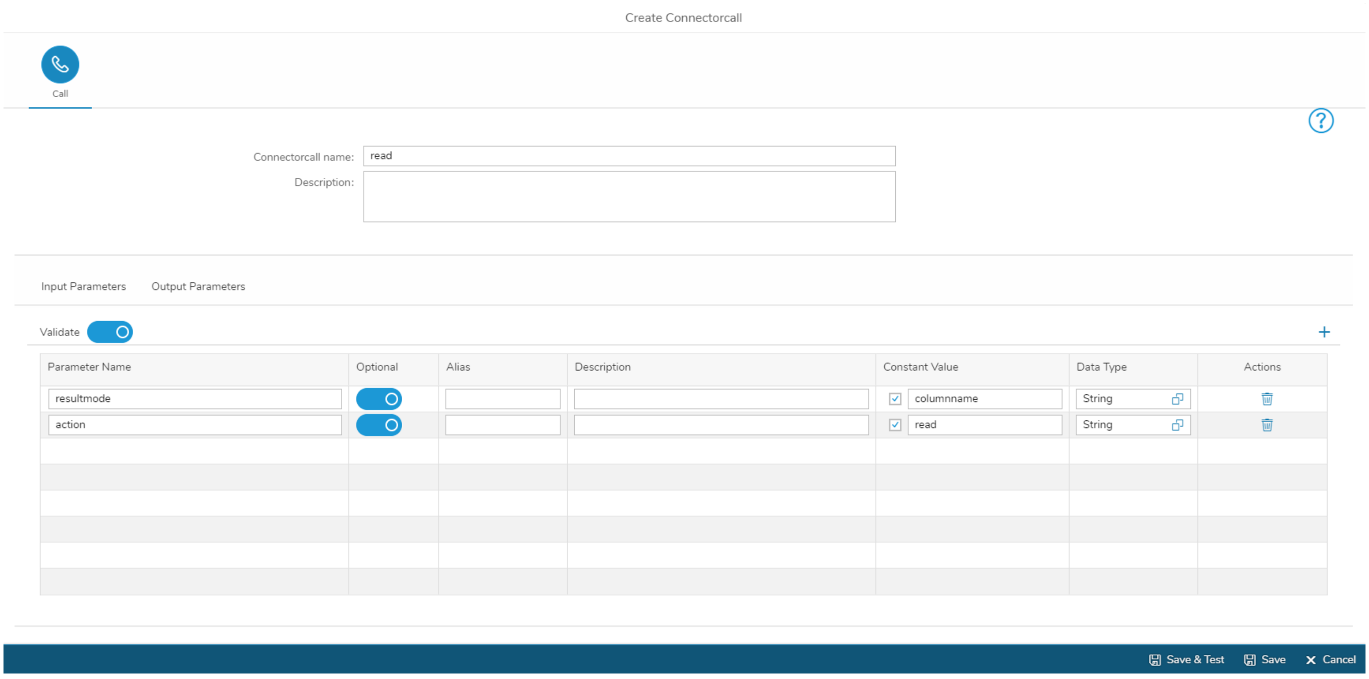
***** *READ*: The connector can only read from the specified CSV file *path*, no write operations are permitted.

***** *WRITE*: The connector can only write to the CSV file, but not read from it.

***** *READ/WRITE:* The connector can read from the file and also write to it.

READ

The CSV Connector Call for a READ operation requires 2 Input parameters: "*action*" and "*resultmode*".



Create Connectorcall

Connectorcall name: read
Description:

Parameter Name	Optional	Alias	Description	Constant Value	Data Type	Actions
resultmode	●			✓ columnname	String	
action	●			✓ read	String	

To execute a read operation, call the Connector with the parameter ***"action"*** and the constant value ***"read"***.

Reading Connectors get the result as JSON array of arrays by default. There can be definied two "***resultmode***"parameter:

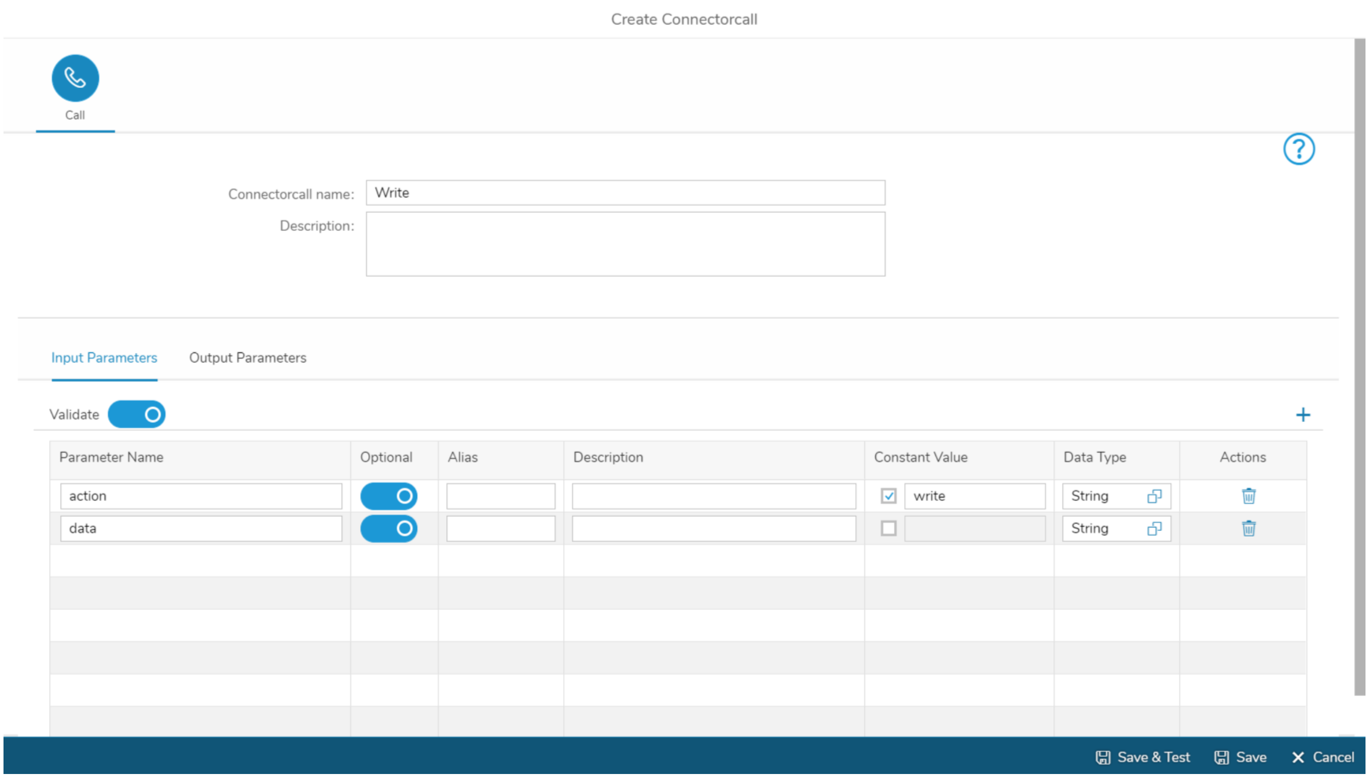
- "***columnnumber***" returns an array of JSON objects, where the key is "col0", "col1", ... "colX" for the column.
- "***columnname***" returns an array of JSON objects where the key is the String taken from the header row (only available if *`headerInFirstLine`* is true - see CSV Connector Details).
- **"array"** returns a two-dimensional array (array of arrays), where the first array contains the row and the second array the column. This mode is the stablest, as the data type conversion must be done by the user. This is also the standard mode if you do not provide a resultmode.

WRITE

The CSV Connector Call for a WRITE operation requires also 2 Input parameters: "*action*" and "*data*".

The data parameter must be a two-dimensional array consisting of only Strings. You can specify the array in the call itself by adding the respective indices after the data parameter name e.g.

data[0][0], data[0][1]. In this case you can provide multiple fields of the parameter data. You have to be sure, that the indices are unique.



Connectorcall name: Write
Description:

Validate: on

Parameter Name	Optional	Alias	Description	Constant Value	Data Type	Actions
action	on			✓ write	String	
data	on				String	

The Connector returns everything if you use "/" as Output parameter.

Current Release & Archive

https://developer.simplifier.io/documentation/current-release/

The Documentation refers to the latest version of Simplifier (4.5). If you have an On-Premise installation and need help with an older version, please contact us via contact@simplifier.io, we are glad to support you.

New Features Release 4.5

Modules

The new module feature enables the logical separation of user interfaces and application logic. It divides your application into smaller components, increasing reusability, improving runtime performance through dynamic loading, and improving maintenance. Interface descriptions are used for communication between the application and the modules to exchange data bidirectionally.

Instance Administration

The new instance administration declares your Simplifier environment into a logical development, quality assurance and productive system and provides the foundation for customizing multiple backend systems within a connector and deploying transport request within the reworked transport system.

Multiple Endpoints for each Connector

By means of the new instance administration, it is possible to connect your Simplifier instance in a connector with the appropriate backend systems according to the purpose (development, quality assurance or productive system). In this way, you can easily connect a Simplifier test environment with your suitable test backend systems - Simplifier makes sure that the right backend system is addressed in the right environment.

OData v2 Connector

With the new release, it is now possible to configure OData backend services automatically using a wizard. This allows OData services to be used as data objects in the same way as other connector types, including visual mapping.

Proxy Connector

The new Proxy Connector allows the usage of any HTTP services that are not based on specific protocol architectures such as REST, SOAP or OData.

Updated Features

Transport System

Our transport system has been reworked completely. The release of a transport request now freezes all content and enables a rollback to an earlier version. Furthermore, the content can now be transferred from one Simplifier instance to another via remote HTTP connection without time-consuming manual download and upload.

The import and export history ensures full transparency at all times and shows you when a transport was imported to which instance and where conflicts may have arisen.

The automatic assignment of a unique transport number ensures documentation in terms of Application Change Management.

Automated Application Testing

Our automated testing feature published in Release 4.0 has been revised. We introduced Journey and Page Objects analog to the OPA5 standard. Now you are able to group and run several test cases within one Journey. For this you can access reusable test runs from the Page Objects.

Mobile Action Audio/Video Call

Our audio/video call feature has been migrated from P2P technology to client-server technology. This allows audio/video conferences with multiple participants (P2P allows only 1:1). This requires a separate conferencing plugin, which must be installed on Simplifier.

New Features within Process Designer

The search in the Process Designer now allows you to search for variables, auto fields and parameters in the mapping dialogs as well as for their unique ids. Furthermore, descriptions, as well as function names of business objects or connector call names are searchable.

A new condition *isSet* or *isNotSet* now checks whether a variable is defined or not.

Errors in business objects or connectors can be handled further by using the *ErrorMessage* parameter in the mapping dialog.

A syntax check via ESLint via the ECMAScript 5 standard in script blocks or screen events is now optionally possible for older browsers.

New Features within UI Designer

Thanks to new screen events such as *onBeforeFirstShow*, *onBeforeHide* and *afterShow*, your app is now even more configurable and you achieve better runtime behavior.

Additionally, it is now possible to define a binding path to a data model directly to each widget property in the UI Designer.

Update to UI5 1.60

The Simplifier administration interface now uses the latest UI5 version 1.60. Furthermore, we have also adapted all widgets to the new version. Of course, all Simplifier apps can benefit from it.

Enhanced SOAP Connector

The SOAP Connector now also enables file transfers in the form of attachments. Besides that, you can design the header variables arbitrarily and add custom HTTP headers.

Enhanced SQL Connector

A larger input mask including syntax highlighting is now available in the SQL connector for entering SQL statements.

Furthermore, we offer a new transaction mode, in which all statements are executed bundled as one database transaction.

For MySQL, SQLLite and MSSQL databases, also the contents of the primary keys are given as output parameter *generatedKeys*.

Reworked SAP RFC Connector

The reworked SAP RFC Connector now also supports the ABAP TABLES parameters in function modules.

Redesigned Admin UI

The administration interface has been updated to UI5 version 1.60. Its appearance has been adapted to the new Simplifier logo and corporate design. Moreover, there is a convenient way to report bugs, that can be called from the menu right upper side.

Refactoring of Business Objects

By revising the business objects, we achieve up to 500% faster execution speed of server-side program logic. On top of that, we have optimized the usability and enable a jump to functions directly from the list view. The functions of the library numeral.js are now also available to format or calculate numbers.

For older browsers, a syntax check via ESLint via the ECMAScript 5 standard is now optionally possible.

Performance Monitoring

Logs & Monitoring now allows you to measure the execution time of a connector.

Performance Optimization

The execution speed of Simplifier has been improved in 3 points:

- up to 500% faster execution time of server-side business objects
- up to 300% faster loading time of Simplifier application due to an adapted preload method
- faster loading time of the UI Designer as well as business object maintenance

Reworked SAML 2.0 Authentication

We have revised the SAML 2.0 authentication in the context of ADFS (Active Directory Federation Services).

Reworked App Permissions

The UI Designer authorization object has now been extended to *Create*, *View*, *Edit* and *Delete*.

Reworked Widgets

A widget can now be deployed for multiple UI5 versions. The widget can be directly linked to the UI5 versions and will be generated accordingly. Via a filter in the UI Designer, only the compatible widgets matching the stored UI5 version of the application are displayed.

Data Centers of Simplifier Cloud

https://developer.simplifier.io/documentation/installation-instructions/simplifier-cloud/data-centers-of-simplifier-cloud/

[vc_row row_type="row" stretch_row_type="no"][vc_column][vc_column_text]The Simplifier Cloud is hosted in data centers of T-Systems Germany.

The locations of the twin-core data centers are:

Lübecker Str. 2 39124 Magdeburg Germany	Am Schiens 10-11 39221 Bördeland/Biere Germany

[/vc_column_text][/vc_column][/vc_row][vc_row row_type="row" stretch_row_type="no"][vc_column][vc_gmaps link="#E-8_JTNDaWZyYW1lJTIwc3JjJTNEJTIyaHR0cHMlM0ElMkYlMkZ3d3cuZ29vZ2xlLmNvbSUyRm1hcHMlMkZkJTJGdSUyRjAlMkZlbWJlZCUzRm1pZCUzRDF2YXJBRGVLbkpzYkQ0UW9ZX19FTWl5OUpLT0ZhcE5JdyUyMiUyMHdpZHRoJTNEJTIyNjQwJTIyJTIwaGVpZ2h0JTNEJTIyNDgwJTIyJTNFJTNDJTJGaWZyYW1lJTNF" title="T-Systems data center"][/vc_column][/vc_row]

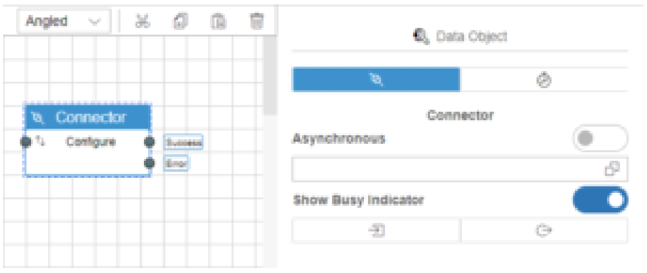
Data Object

https://developer.simplifier.io/documentation/applications/process-dashboard-and-designer/data-objects/

Connector | Asynchronous Connector | Business Object | Mapping Collections | Mapping Structs

Vimeo Video

Data Objects represent data sources and destinations, which can be triggered for execution. You can choose either a predefined connector (you can activate it to asynchronous) or a business object.



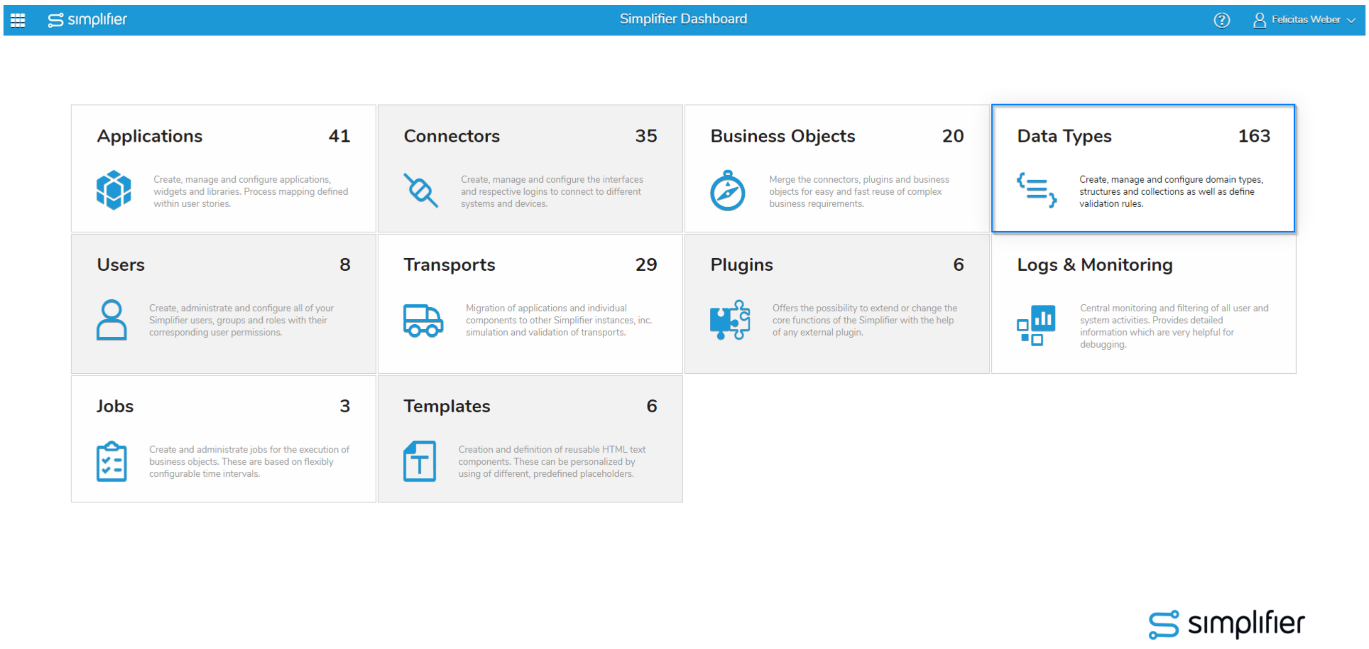
Function	**Description**
Asynchronous	If you select this option, the value helper assistant only offers asynchronous connectors.
Value Helper	If you open the value helper, an assistant opens that guide you to your connector.
Show Busy Indicator	You have the possibility to configure if the UI is blocked by the busy indicator, or can configure which element on your screen should be blocked by it.
Input Mapping	You can map variables, auto fields, widget properties and constants to the input parameter of your connector.
Output Mapping	You can map the output parameter of your connector to variables and properties.

Connector

Data Types

https://developer.simplifier.io/documentation/data-types/

The tile “Data Types” is the central way to define different types of data, structures and collection of Data Types and their validation rules.



Applications 41
Create, manage and configure applications, widgets and libraries. Process mapping defined within user stories.

Connectors 35
Create, manage and configure the interfaces and respective logins to connect to different systems and devices.

Business Objects 20
Merge the connectors, plugins and business objects for easy and fast reuse of complex business requirements.

Data Types 163
Create, manage and configure domain types, structures and collections as well as define validation rules.

Users 8
Create, administrate and configure all of your Simplifier users, groups and roles with their corresponding user permissions.

Transports 29
Migration of applications and individual components to other Simplifier instances, inc. simulation and validation of transports.

Plugins 6
Offers the possibility to extend or change the core functions of the Simplifier with the help of any external plugin.

Logs & Monitoring
Central monitoring and filtering of all user and system activities. Provides detailed information which are very helpful for debugging.

Jobs 3
Create and administrate jobs for the execution of business objects. These are based on flexibly configurable time intervals.

Templates 6
Creation and definition of reusable HTML text components. These can be personalized by using of different, predefined placeholders.

Data Types are a way to ensure data are sent and received in the right type format to and from the backend systems.
With this feature, you can define data definitions to validate your data with client and server-side validation to prevent security issues and backend saving problems due to wrong data formats or hacker attacks.

Data Types can be assigned to widgets and connector calls to validate input and output data.

There are six **Base Data Types** defined in the Simplifier:

Date	Date Format
String	Characters, numbers and any other symbols from the Unicode Character Set
Boolean	True or False
Integer	Positive and negative numbers like -2, -1, 0, 1, 2 ...
Float	Numbers with precisions like 2,503
Any	Accept all kind of Data Types even heterogeneous Arrays.

With the "Data Types" tile, you are able to enhance the Base Types and define your own logic.
The new Data Types are split into three different types:

- Domain types
- Structs
- Collections

You can assign tags to Data Types to find them easily in the search bar.

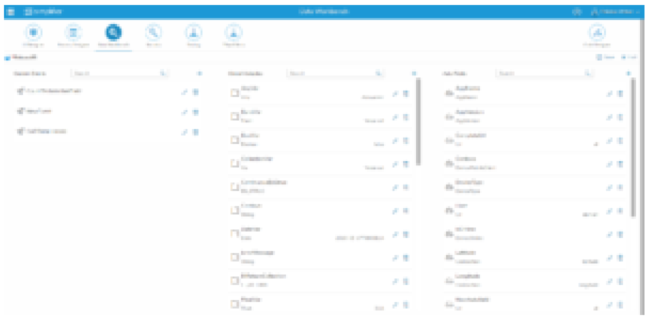
Data Workbench

https://developer.simplifier.io/documentation/applications/data-workbench/

Custom Events | Global Variables | Global Auto Fields

Vimeo Video

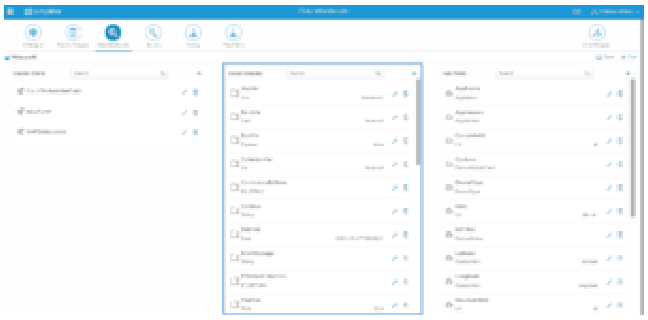
Within the Data Workbench, it is possible to administer custom events, global variables and auto fields that you want to use cross-functional in the user stories.



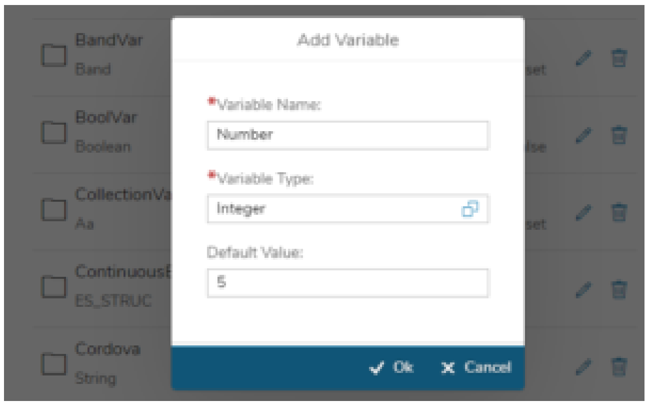
Custom Events

Global Variables

Use global variables as a container to buffer data, e.g. if a connector returns a lot of data and you would like to use some of it later in your work process, you can save the parameter as variables and map them later.



To create a new one, click on the plus icon. Enter a variable name, variable type and default value (optional).



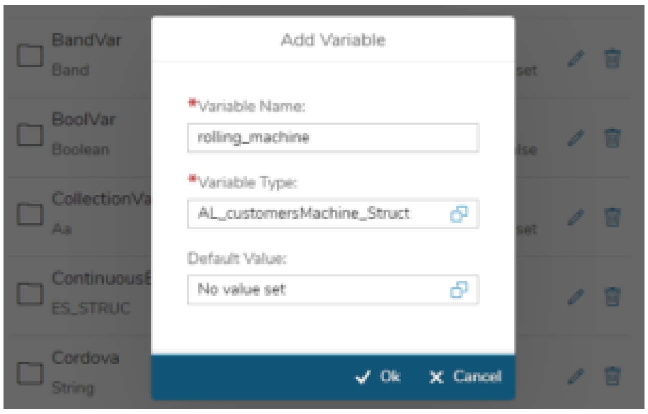
Add Variable

*Variable Name: Number

*Variable Type: Integer

Default Value: 5

You can also select, e.g. a struct as the variable type.



*Variable Name: rolling_machine
*Variable Type: AL_customersMachine_Struct
Default Value: No value set

You get the information about the fields on the right panel to configure the default values.

In general, the fields have four different appearances depending on their own data type:

Data type	**Appearance**	**Behavior/Usage**
String, Integer, Float	Input field with validation (depending on data type)	Values are written in the input field and saved on live change.
Date	Date time picker	The date-time picker dialog opens and the user is able to select the date and time.
Boolean	Switch	The switch can be set to true or false.
Collection, Struct	Link	By clicking on the link, you will be navigated to a complex data type in the left tree, that will also be selected automatically.

Example of a link press for clarification:

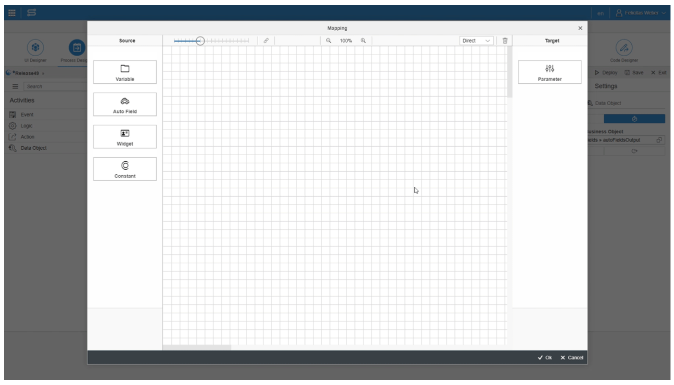
By pressing the 'lubricant' link from the dialog above, there are also different appearances of the tree items depending on their datatype:

Data type	Icon	Is Expandable	Plus Button	Remove Button
Base (String, Integer, Float, Date, Boolean)	☐	No	No	No
Domain	🪪	No	No	No
Struct	{≡}	Yes	No	No
Collection	☰	Depending on collection items	Yes	No (only if it is a collection in a collection)

Collection exception:

By adding a collection object, the item is inserted into the structure below the collection. The collection object can then be clicked like an ordinary tree element. The only difference is that a collection item can be removed using the delete button.

You can reference variables in data objects as in- or output parameter. To do so, drag a variable (that you've created previously in the Data Workbench) from the toolbar in the mapping dialog.



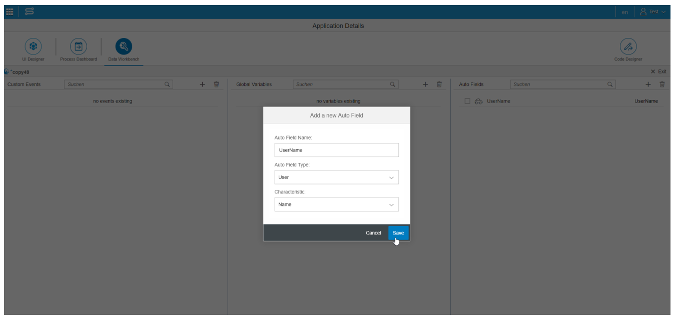
Global Auto Fields

Auto fields are automatically computed/filled fields. You can use them e.g. if you want to greet the user who is logged in with his actual name or load e.g. the version number.

To create a new auto field, click on the plus icon in the Data Workbench. Enter a name for the auto field, the type, and the characteristic.

It is possible to declare auto fields from five types:

- Application
- User
- URL
- Geolocation
- Device



Add a new Auto Field

Auto Field Name: UserName

Auto Field Type: User

Characteristic: Name

Category	Characteristic	Description
Application	Name	The name of the application.
	Version	The current version of the application. If the application is not yet released, it is stated as "n/a".
User	Name	The currently logged in user name.
URL	Complete URL	

Database Setup

https://developer.simplifier.io/documentation/installation-instructions/on-premise/database-setup/

In the following article, you find a description of **how to set up an external database for the Simplifier Core platform**.

1. Switch to configuration persistence path, e.g. /opt/simplifier/data/conf/

2. Create new include.conf file with the following format:

```
database {

  dbms: "oracle"

  user: ""

  pass: ""

  host: ""

  port: 1521

  database: ""

  table_prefix: ""

}
```

Database Credentials

dbms	mysql or oracle
user	Username of the database
pass	Password of the database user
host	IP address of the database server
port	Port of the database server
database	Name of the database / database scheme
table_prefix	Prefix of the database

Delete a PDF Template

https://developer.simplifier.io/documentation/plugins/pdf-plugin/technical-call-pdf-plugin/delete-pdf-template/

Delete Template

To delete a template, you need the following parameter:

URL			/client/1.0/PLUGIN/pdfPlugin/adminTemplateDelete
Input-Parameter	**Name**	Template name	
Output-Parameter	None		

Example input::

```
{
    "name": "templatename"
}
```

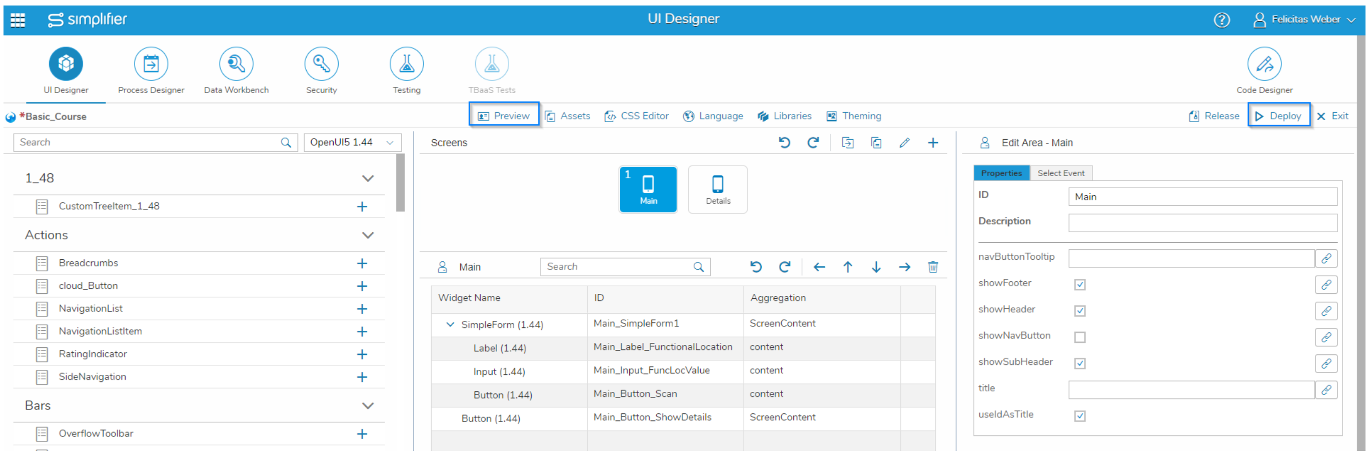
Example output:

```
{
    "success": true
}
```

Deploy and Preview

https://developer.simplifier.io/documentation/applications/ui-designer/deploy-and-preview/

Applications can be deployed rapidly and previewed within a standard web browser and on the mobile device using the Simplifier Mobile Client.



Main

Widget Name	ID	Aggregation
SimpleForm (1.44)	Main_SimpleForm1	ScreenContent
Label (1.44)	Main_Label_FunctionalLocation	content
Input (1.44)	Main_Input_FuncLocValue	content
Button (1.44)	Main_Button_Scan	content
Button (1.44)	Main_Button_ShowDetails	ScreenContent

ID	Main
Description	
navButtonTooltip	
showFooter	✓
showHeader	✓
showNavButton	☐
showSubHeader	✓
title	
useIdAsTitle	✓

If you click on preview, it opens a new browser tab. Every time you will deploy your application, the browser tab will be reloaded automatically.

Warning: Make sure that pop-ups are not blocked in your browser for your Simplifier instance.

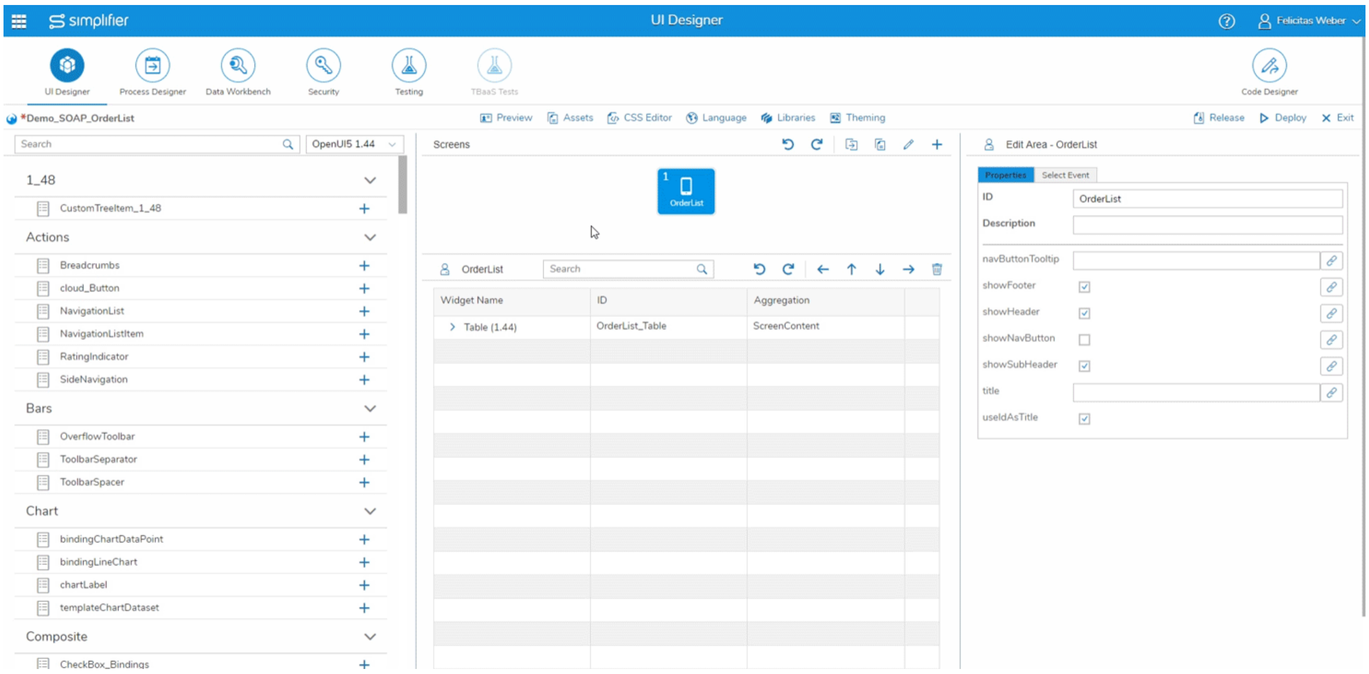
You can simulate your preview for different mobile devices with the Chrome Developer Tools or use it for debugging.

To access the DevTools, open your app preview in Google Chrome and press **F12**.

Alternatives:

- Select the **Chrome menu** ☰ at the top-right of your browser window, then select **Tools** > **Developer Tools**.
- Or use Ctrl+Shift+I (or Cmd+Opt+I on Mac).

Via the toggle device toolbar, you can simulate different devices like Galaxy S5, iPhone 6 or iPad to preview your application. More information on Google Chrome DevTools Device Mode.



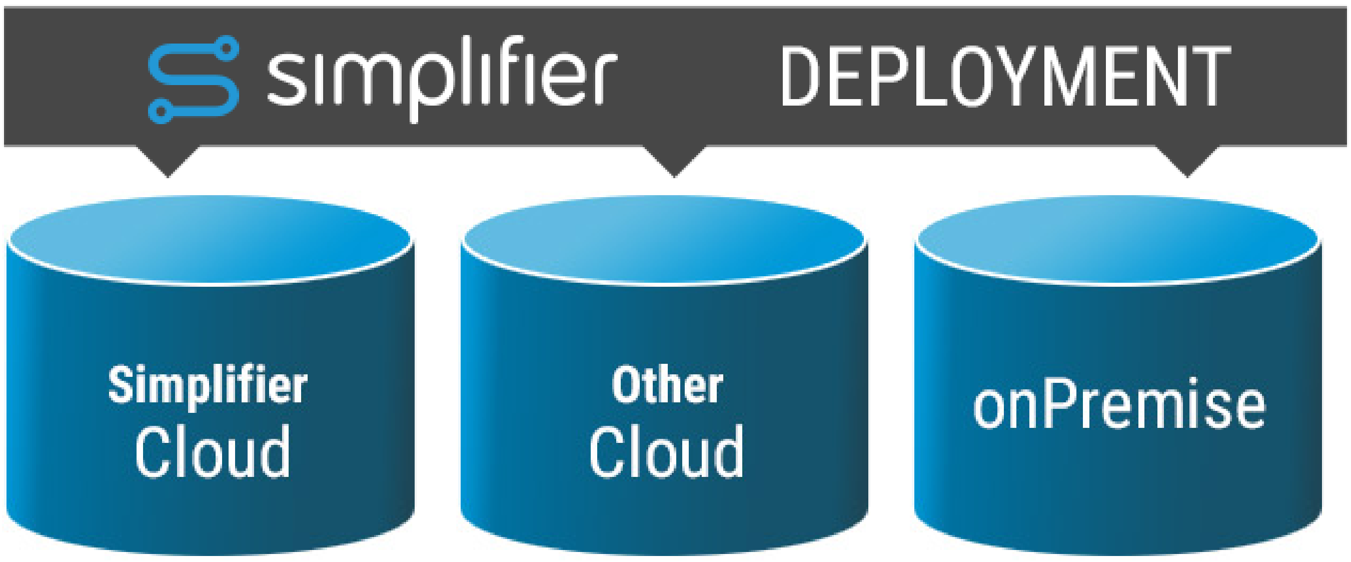
Widget Name	ID	Aggregation
Table (1.44)	OrderList_Table	ScreenContent

ID	OrderList
Description	
navButtonTooltip	
showFooter	✓
showHeader	✓
showNavButton	☐
showSubHeader	✓
title	
useIdAsTitle	✓

Deployment & Installation Instructions

https://developer.simplifier.io/documentation/installation-instructions/

A Simplifier application can be deployed in different ways. You can deploy to your local machine for development and testing, you can deploy to the Simplifier cloud, Cloud Foundry-based platforms, Azure, AWS, SAP Cloud, or a server you configured yourself.



Cloud installations are hosted and maintained by Simplifier AG. Each instance is reachable via a unique DNS name:

https://<instance-name>.simplifier.io

On-premise installations are hosted by our customers, on their own infrastructure. This scheme is especially useful if the Simplifier shall be integrated into a closed network infrastructure.

Checklist for Installation

For Installation of the Simplifier, the following persons and things are required:

- IT Security Officer
- System owner regarding Backend Interfacing
- Firewall Administrators
- Reverse Proxy Administrator
- IT Administrators
- SSL Certificates for HTTPS
- The initial login credentials are admin / admin

Use the following checkpoints for a successful installation

Checkpoint	Description
✓	System-Requirements are clear
✓	Domain-Name for Development, QA and Productive Systems are clear
✓	SSL Certifcates for all 3 Instances is available

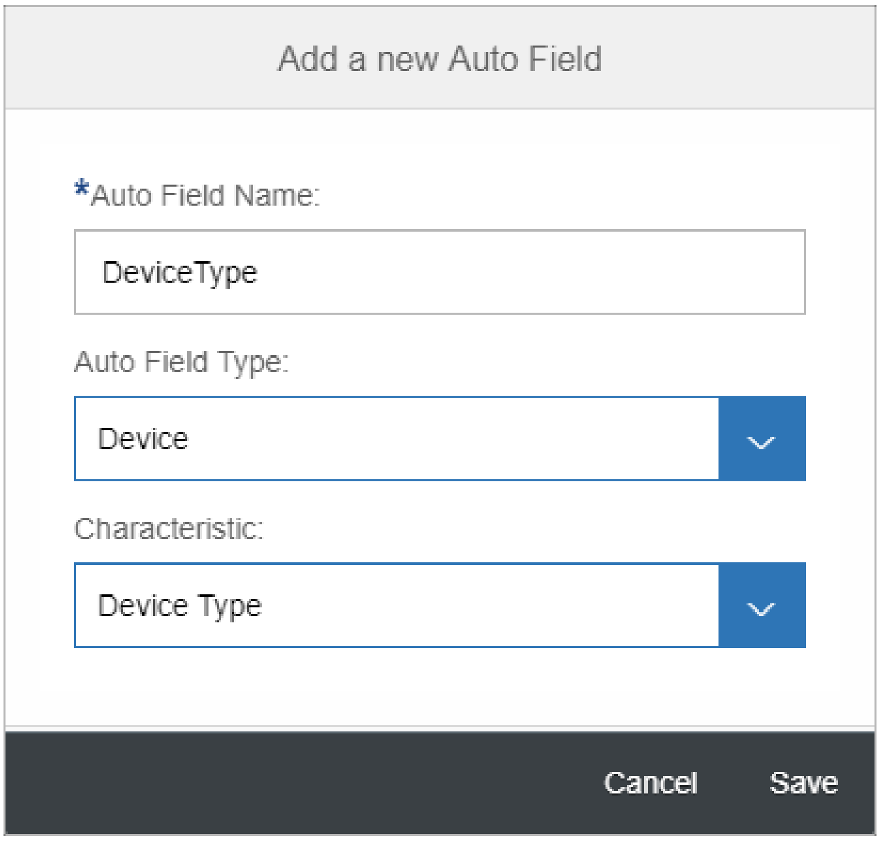
✓	Firewallports 443 and 587 are open
✓	Backendsystem is reachable via Supported Protocols

Device Condition

https://developer.simplifier.io/documentation/applications/process-dashboard-and-designer/logic/device-condition/

The Simplifier is able to recognize the device type that is used by the application, so you can assign different functionalities to it, e.g. different designed login screens for mobile or wearable devices.

At first, you need to create a new auto field type with the auto field type "Device" and the characteristic "Device Type". How to create auto fields, you can see here.



Add a new Auto Field

*Auto Field Name: DeviceType

Auto Field Type: Device

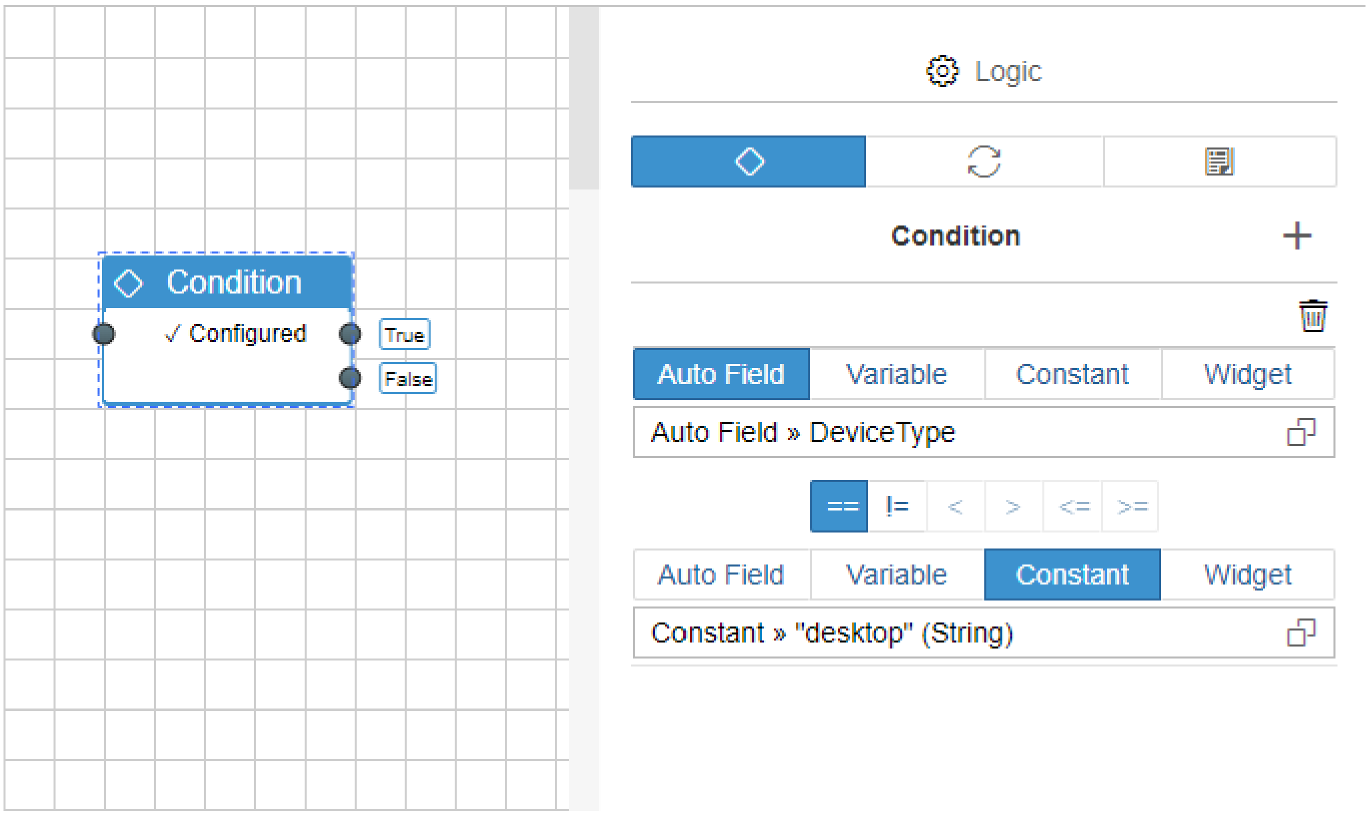
Characteristic: Device Type

Within the Process Designer, you can refer to this auto field using a condition. Select the corresponding auto field and assign it to a constant (String).

Choose between:

- desktop
- phone
- smartglass
- tablet
- watch

Please pay attention to the lower case!



Auto Field » DeviceType

Constant » "desktop" (String)

Docker Hub

https://developer.simplifier.io/documentation/installation-instructions/general-instructions/docker-hub/

The Simplifier is also available on Docker Hub.

Short Instructions

Create the directory which will host all external user-specific data:

```
$ mkdir -p /home/simplifier/data
$ export SIMPLIFIER_DIR="/home/simplifier/data"
```

Install SSL certificates:

```
$ mkdir -p $SIMPLIFIER_DIR/certs
$ cp <certificate.pem> SIMPLIFIER_DIR/certs/default.crt
$ cp <keyfile.pem> SIMPLIFIER_DIR/certs/default.key
```

Run docker:

Alternative 1: with SSL/Certificates

```
$ docker run -d -v $SIMPLIFIER_DIR:/opt/simplifier/data \
-p 80:80 -p 443:443 -p 8090:8090  \
--name=simplifier simplifierag/onpremise:latest
```

Alternative 2: without SSL/Certificates

```
$ docker run -d -v $SIMPLIFIER_DIR:/opt/simplifier/data \
-p 80:8080 -p 8090:8091 \
--name=simplifier simplifierag/onpremise:latest
```

Docker Installation

https://developer.simplifier.io/documentation/installation-instructions/general-instructions/docker-installation/

Get Docker CE

Referenced to the official Docker instructions.

Note: This installation instructions is based on the example of the operating system Ubuntu 16.04 LTS.

SET UP THE REPOSITORY

Update the apt package index:

```
$ sudo apt-get update
```

Install packages to allow apt to use a repository over HTTPS:

```
$ sudo apt-get install \

    apt-transport-https \

    ca-certificates \

    curl \

    software-properties-common
```

Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Verify that you now have the key with the fingerprint 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88, by searching for the last 8 characters of the fingerprint.

```
$ sudo apt-key fingerprint 0EBFCD88

pub   4096R/0EBFCD88 2017-02-22

      Key fingerprint = 9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88

uid                  Docker Release (CE deb) <docker@docker.com>

sub   4096R/F273FCD8 2017-02-22
```

Use the following command to set up the stable repository. You always need the stable repository, even if you want to install builds from the edge or test repositories as well. To add the edge or test repository, add the word edge or test (or both) after the word stable in the commands below.

Note: The lsb_release -cs sub-command below returns the name of your Ubuntu distribution, such as xenial. Sometimes, in a distribution like Linux Mint, you might have to change $(lsb_release -cs) to your parent Ubuntu distribution. For example, if you are using Linux Mint Rafaela, you could use trusty.

amd64:

```
$ sudo add-apt-repository \

   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \

   $(lsb_release -cs) \

   stable"
```

INSTALL DOCKER CE

Update the apt package index:

```
$ sudo apt-get update
```

Install the latest version of Docker CE:

```
$ sudo apt-get install docker-ce
```

Docker on Mac

https://developer.simplifier.io/documentation/installation-instructions/locally/docker-on-mac/

Install Docker for Mac

Docker for Mac is the Community Edition (CE) of Docker for MacOS. To download Docker for Mac, head to Docker Store.

Download from Docker Store

What to know before you install

README FIRST for Docker Toolbox and Docker Machine users

If you are already running Docker on your machine, first read Docker for Mac vs. Docker Toolbox to understand the impact of this installation on your existing setup, how to set your environment for Docker for Mac, and how the two products can coexist.

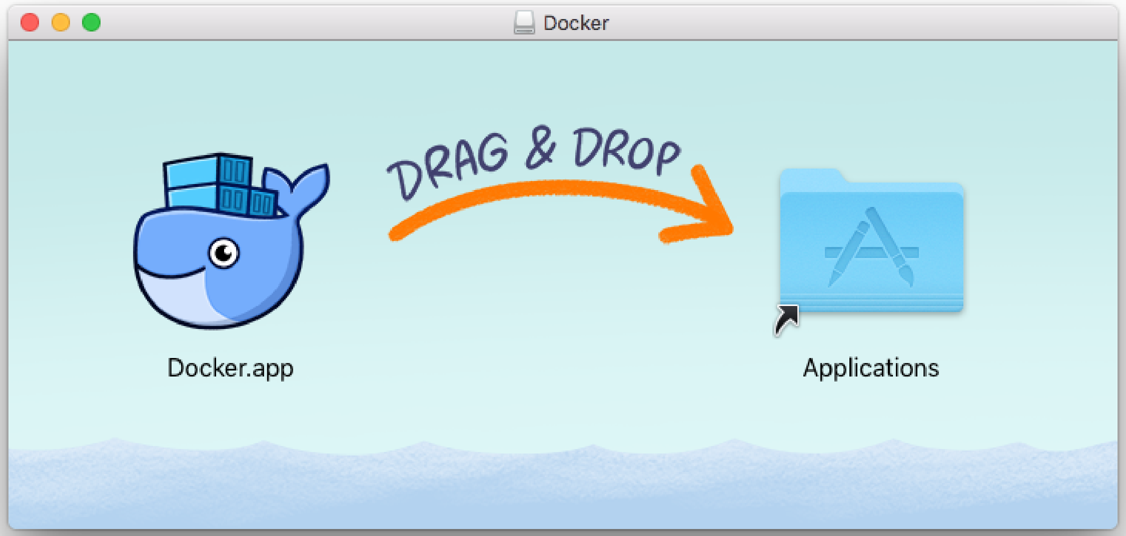
- **Relationship to Docker Machine**: Installing Docker for Mac does not affect machines you created with Docker Machine. You have the option to copy containers and images from your local default machine (if one exists) to the new Docker for Mac HyperKitVM. When you are running Docker for Mac, you do not need Docker Machine nodes running at all locally (or anywhere else). With Docker for Mac, you have a new, native virtualization system running (HyperKit) which takes the place of the VirtualBox system. To learn more, see Docker for Mac vs. Docker Toolbox.
    - Mac hardware must be a 2010 or newer model, with Intel's hardware support for memory management unit (MMU) virtualization, including Extended Page Tables (EPT) and Unrestricted Mode. You can check to see if your machine has this support by running the following command in a terminal: sysctl kern.hv_support
    - macOS El Capitan 10.11 and newer macOS releases are supported. We recommend upgrading to the latest version of macOS.
    - At least 4GB of RAM
    - VirtualBox prior to version 4.3.30 must NOT be installed (it is incompatible with Docker for Mac). If you have a newer version of VirtualBox installed, it's fine.**System Requirements**: Docker for Mac launches only if all of these requirements are met.

Note: If your system does not satisfy these requirements, you can install Docker Toolbox, which uses Oracle VirtualBox instead of HyperKit.

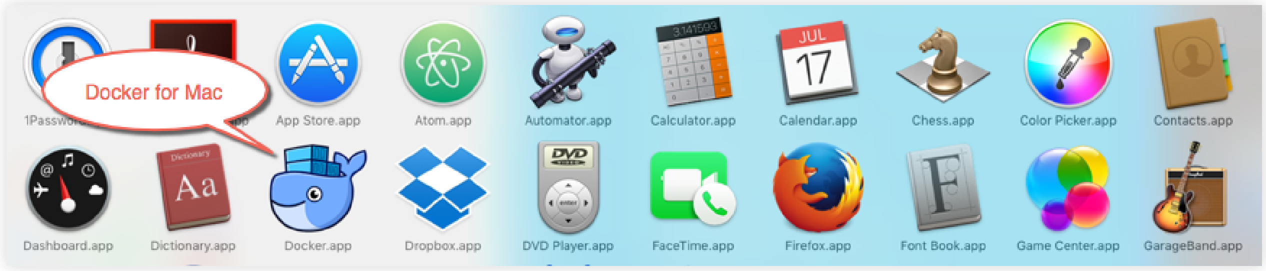
- **What the install includes**: The installation provides Docker Engine, Docker CLI client, Docker Compose, Docker Machine, and Kitematic.

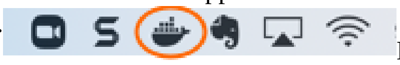
Install and run Docker for Mac

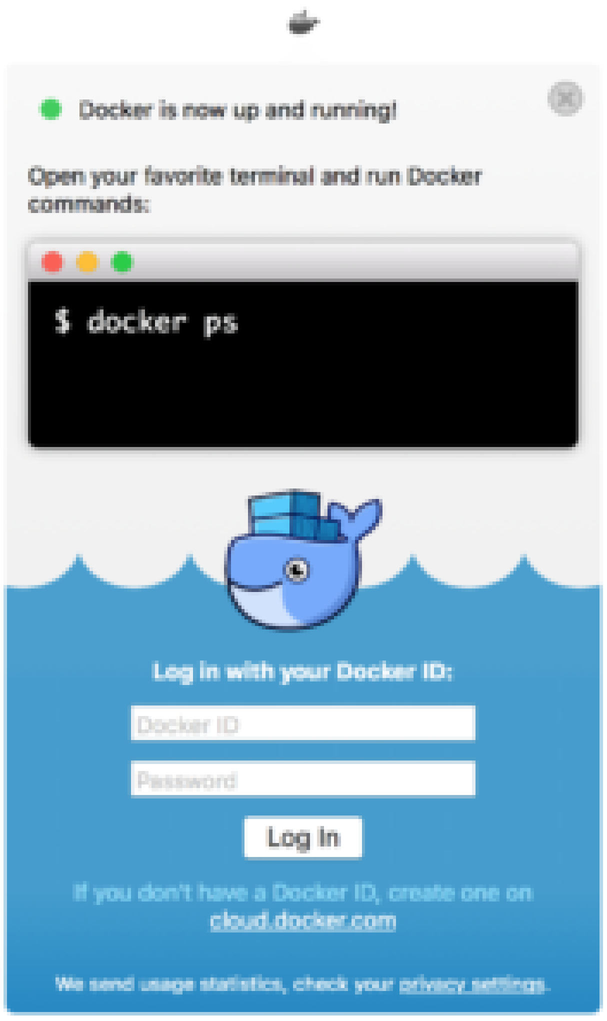
1. Double-click Docker.dmg to open the installer, then drag Moby the whale to the Applications folder.



2. Double-click Docker.app in the Applications folder to start Docker. (In the example below, the Applications folder is in "grid" view mode.)



You are prompted to authorize Docker.app with your system password after you launch it. Privileged access is needed to install networking components and links to the Docker apps.The whale in the top status bar indicates that Docker is running, and accessible from a terminal.  If you just installed the app, you also get a success message with suggested next steps and a link to this documentation. Click the whale ( ) in the status bar to dismiss this popup.


Docker is now up and running!

Open your favorite terminal and run Docker commands:

```
$ docker ps
```

Log in with your Docker ID:

If you don't have a Docker ID, create one on cloud.docker.com

We send usage statistics, check your privacy settings.

3. Click the whale ( ) to get Preferences and other options.
4. Select **About Docker** to verify that you have the latest version.

Congratulations! You are up and running with Docker for Mac.

Docker on Ubuntu / Debian

https://developer.simplifier.io/documentation/installation-instructions/locally/docker-on-ubuntu-debian/

Get Docker CE

Referenced to the official Docker instructions.

Note: This installation instructions is based on the example of the operating system Ubuntu 16.04 LTS.

SET UP THE REPOSITORY

Update the apt package index:

```
$ sudo apt-get update
```

Install packages to allow apt to use a repository over HTTPS:

```
$ sudo apt-get install \

    apt-transport-https \

    ca-certificates \

    curl \

    software-properties-common
```

Add Docker’s official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Verify that you now have the key with the fingerprint 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88, by searching for the last 8 characters of the fingerprint.

```
$ sudo apt-key fingerprint 0EBFCD88

pub   4096R/0EBFCD88 2017-02-22

      Key fingerprint = 9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88

uid                  Docker Release (CE deb) <docker@docker.com>

sub   4096R/F273FCD8 2017-02-22
```

Use the following command to set up the stable repository. You always need the stable repository, even if you want to install builds from the edge or test repositories as well. To add the edge or test repository, add the word edge or test (or both) after the word stable in the commands below.

Note: The lsb_release -cs sub-command below returns the name of your Ubuntu distribution, such as xenial. Sometimes, in a distribution like Linux Mint, you might have to change $(lsb_release -cs) to your parent Ubuntu distribution. For example, if you are using Linux Mint Rafaela, you could use trusty.

amd64:

```
$ sudo add-apt-repository \

   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \

   $(lsb_release -cs) \

   stable"
```

INSTALL DOCKER CE

Update the apt package index:

```
$ sudo apt-get update
```

Install the latest version of Docker CE:

```
$ sudo apt-get install docker-ce
```

Docker on Windows 10

https://developer.simplifier.io/documentation/installation-instructions/locally/docker-on-windows-10/

Install Docker for Windows

Docker for Windows is the Community Edition (CE) of Docker for Microsoft Windows. To download Docker for Windows, head to Docker Store.

Download from Docker Store

What to know before you install

If your system does not meet the requirements to run Docker for Windows, you can install Docker Toolbox, which uses Oracle Virtual Box instead of Hyper-V.

- **README FIRST for Docker Toolbox and Docker Machine users**: Docker for Windows requires Microsoft Hyper-V to run. The Docker for Windows installer enables Hyper-V for you, if needed, and restart your machine. After Hyper-V is enabled, VirtualBox no longer works, but any VirtualBox VM images remain. VirtualBox VMs created with docker-machine (including the defaultone typically created during Toolbox install) no longer start. These VMs cannot be used side-by-side with Docker for Windows. However, you can still use docker-machine to manage remote VMs.
- Virtualization must be enabled in BIOS and CPU SLAT-capable. Typically, virtualization is enabled by default. This is different from having Hyper-V enabled. For more detail see Virtualization must be enabled in Troubleshooting.
- The current version of Docker for Windows runs on 64bit Windows 10 Pro, Enterprise and Education (1607 Anniversary Update, Build 14393 or later).
- Containers and images created with Docker for Windows are shared between all user accounts on machines where it is installed. This is because all Windows accounts use the same VM to build and run containers.
- Nested virtualization scenarios, such as running Docker for Windows on a VMWare or Parallels instance, might work, but come with no guarantees. For more information, see Running Docker for Windows in nested virtualization scenarios
- **What the Docker for Windows install includes**: The installation provides Docker Engine, Docker CLI client, Docker Compose, Docker Machine, and Kitematic.

About Windows containers

Looking for information on using Windows containers?

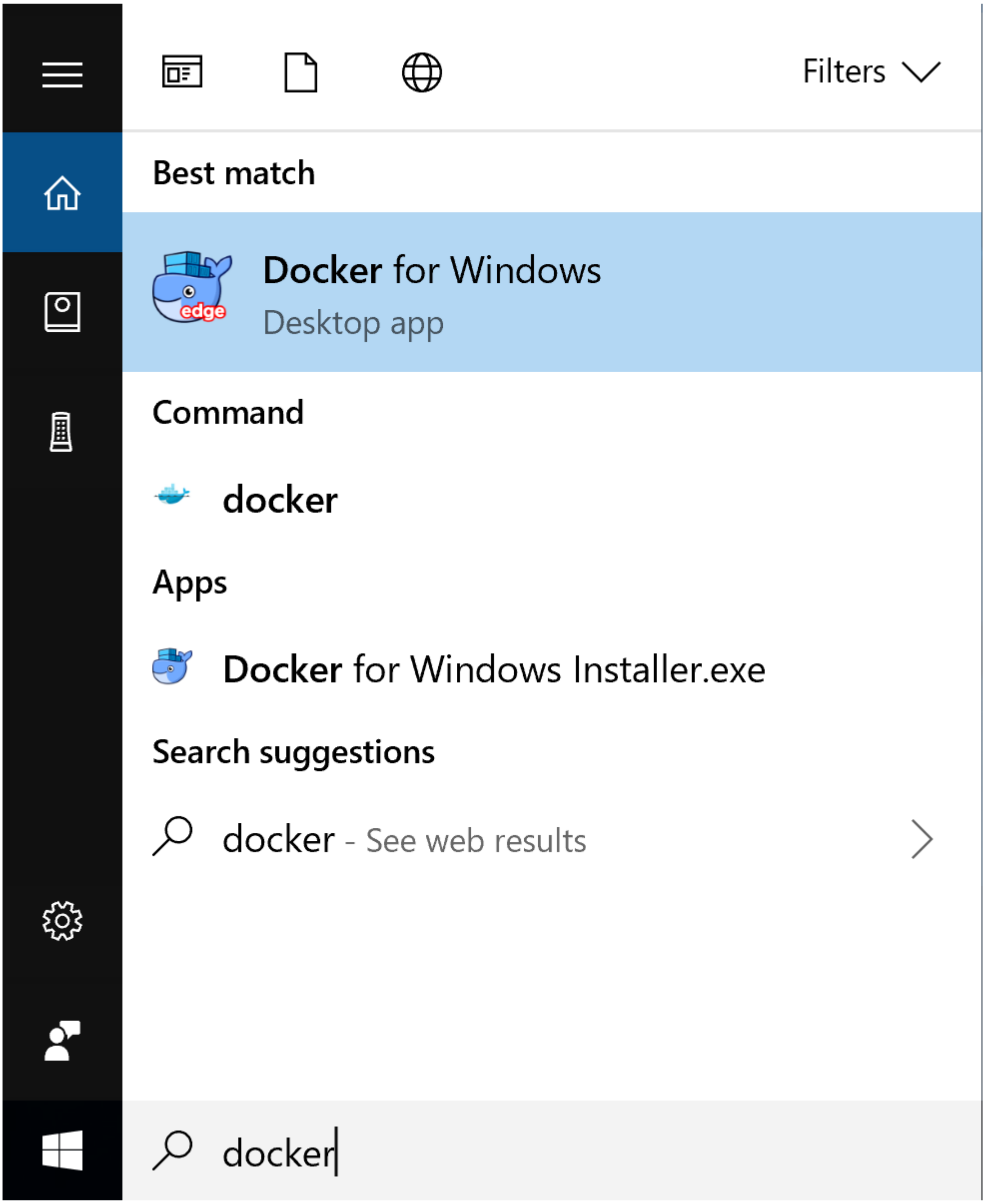
- Switch between Windows and Linux containers describes the Linux / Windows containers toggle in Docker for Windows and points you to the tutorial mentioned above.
- Getting Started with Windows Containers (Lab) provides a tutorial on how to set up and run Windows containers on Windows 10 or with Windows Server 2016. It shows you how to use a MusicStore application with Windows containers.
- Docker Container Platform for Windows Server 2016 articles and blog posts on the Docker website

Install Docker for Windows desktop app

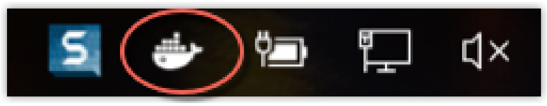
1. Double-click **Docker for Windows Installer.exe** to run the installer.If you haven't already downloaded the installer (Docker for Windows Installer.exe), you can get it from **download.docker.com**. It typically downloads to your Downloads folder, or you can run it from the recent downloads bar at the bottom of your web browser.
2. Follow the install wizard to accept the license, authorize the installer, and proceed with the install.You are asked to authorize Docker.app with your system password during the install process. Privileged access is needed to install networking components, links to the Docker apps, and manage the Hyper-V VMs.
3. Click **Finish** on the setup complete dialog to launch Docker.

Start Docker for Windows

Docker does not start automatically after installation. To start it, search for Docker, select **Docker for Windows** in the search results, and click it (or hit Enter).

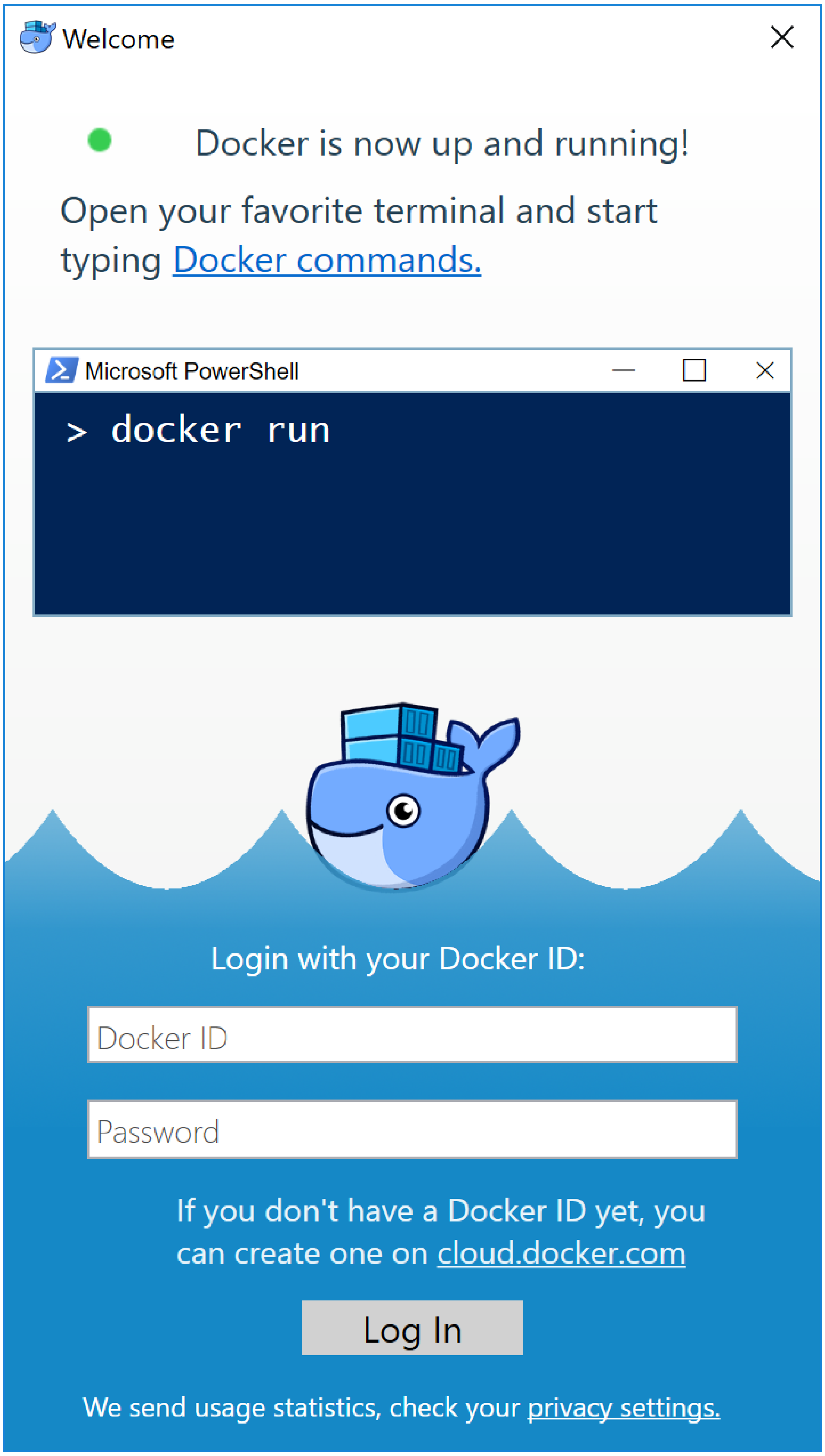

Filters
Best match
Docker for Windows
Desktop app
Command
docker
Apps
Docker for Windows Installer.exe
Search suggestions
docker - See web results
docker

When the whale in the status bar stays steady, Docker is up-and-running, and accessible from any terminal window.



If the whale is hidden in the Notifications area, click the up arrow on the taskbar to show it. To learn more, see Docker Settings.

If you just installed the app, you also get a popup success message with suggested next steps, and a link to this documentation.



Welcome

Docker is now up and running!

Open your favorite terminal and start typing Docker commands.

Microsoft PowerShell

```
> docker run
```

Login with your Docker ID:

Docker ID

Password

If you don't have a Docker ID yet, you can create one on cloud.docker.com

Log In

We send usage statistics, check your privacy settings.

When initialization is complete, select **About Docker** from the notification area icon to verify that you have the latest version.

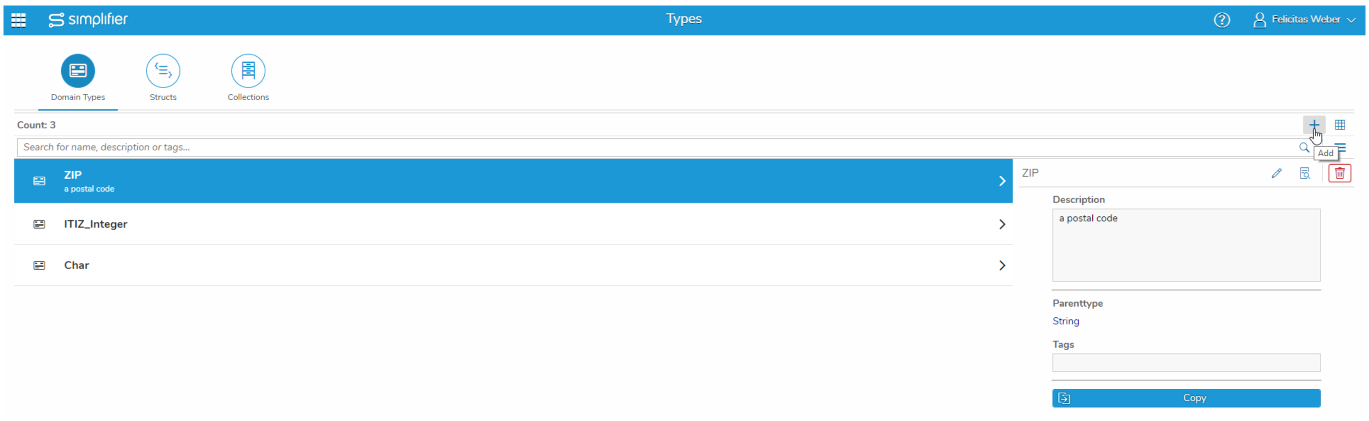
Congratulations! You are up and running with Docker for Windows.

Domain Type

https://developer.simplifier.io/documentation/data-types/create-edit-a-domain-type/

A Domain type represents a single Data Type that inherits from a Base Type like String, Integer, Float, Date, etc. and can include different properties.
For example, a ZIP Code is an inherited type of string with the property of a maximum of 5 chars length.

To create a new Domain Type click on the "+" button.



Count: 3

Search for name, description or tags...

ZIP
a postal code

ITIZ_Integer

Char

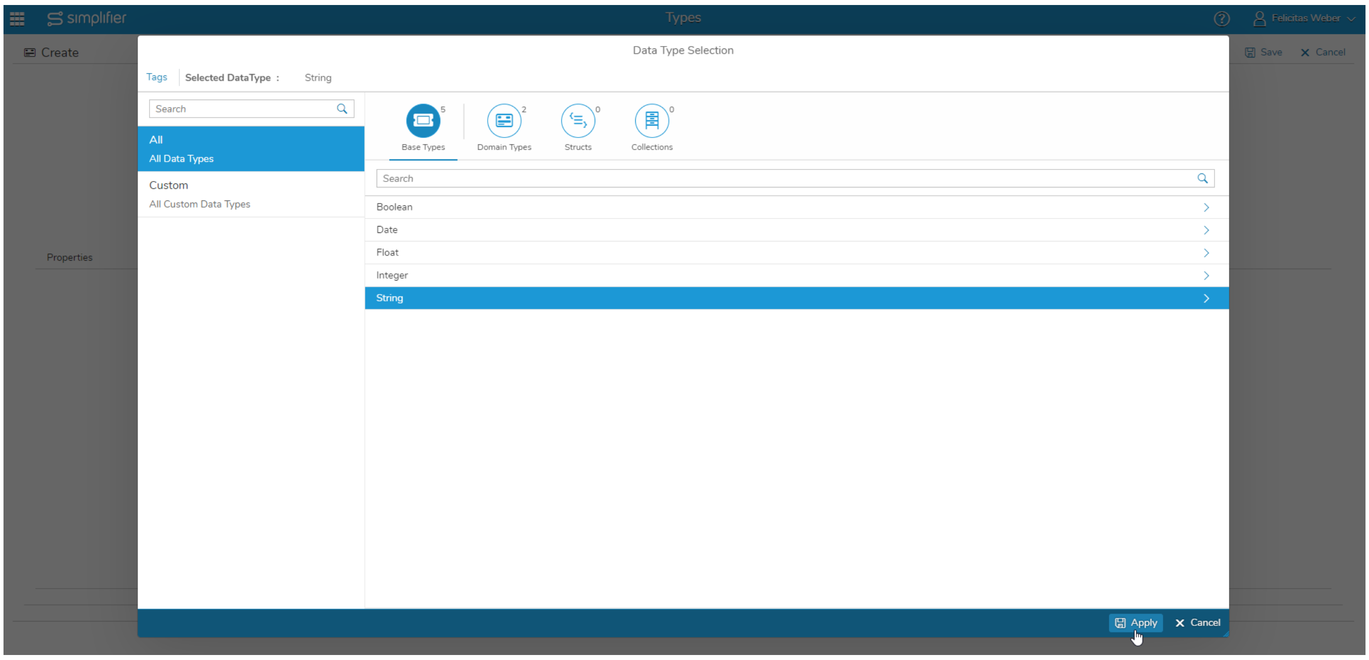
ZIP

Description
a postal code

Parenttype
String

Tags

Enter a unique Name for the Domain Type and an optional Description.
Also, select a Parenttype to inherit from by clicking on the appropriate field.



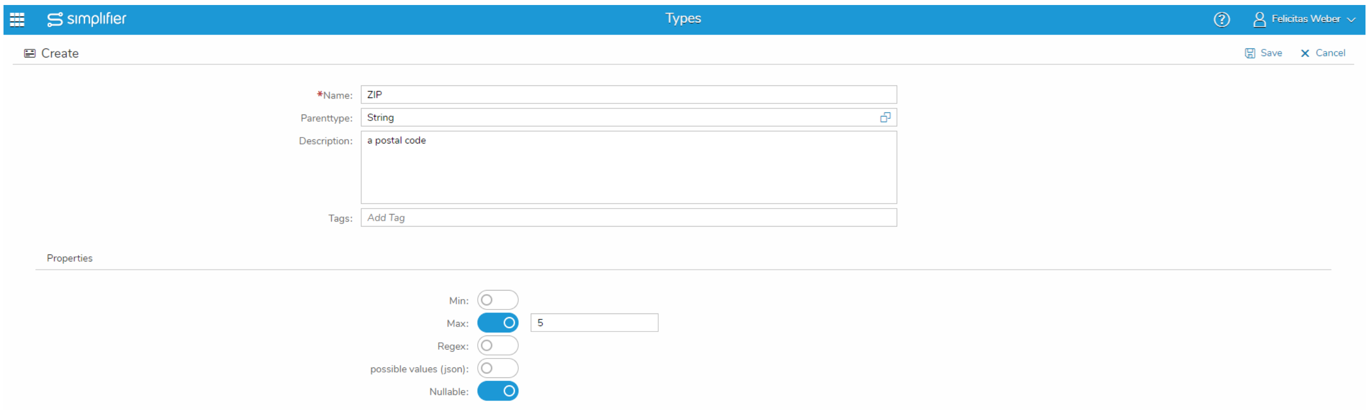
Selected DataType : String

All
All Data Types

Custom
All Custom Data Types

Base Types 5 | Domain Types 2 | Structs 0 | Collections 0

Boolean
Date
Float
Integer
String



*Name: ZIP
Parenttype: String
Description: a postal code
Tags: Add Tag

Min: off
Max: on 5
Regex: off
possible values (json): off
Nullable: on

In the area below you can set the properties of your Data Type:

Min

Minimum length of a String, minimum number of a range, earliest date of a date range.

Max

Maximum length of a String, maximum number range, latest date of a date range.

RegEx

Regular expression to validate this Domain Type - click here for more information.

Possible Values

Simple JSON Array of Strings repesenting literals of chosen Base Type.

Nullable

If activated, this value can be empty (null).

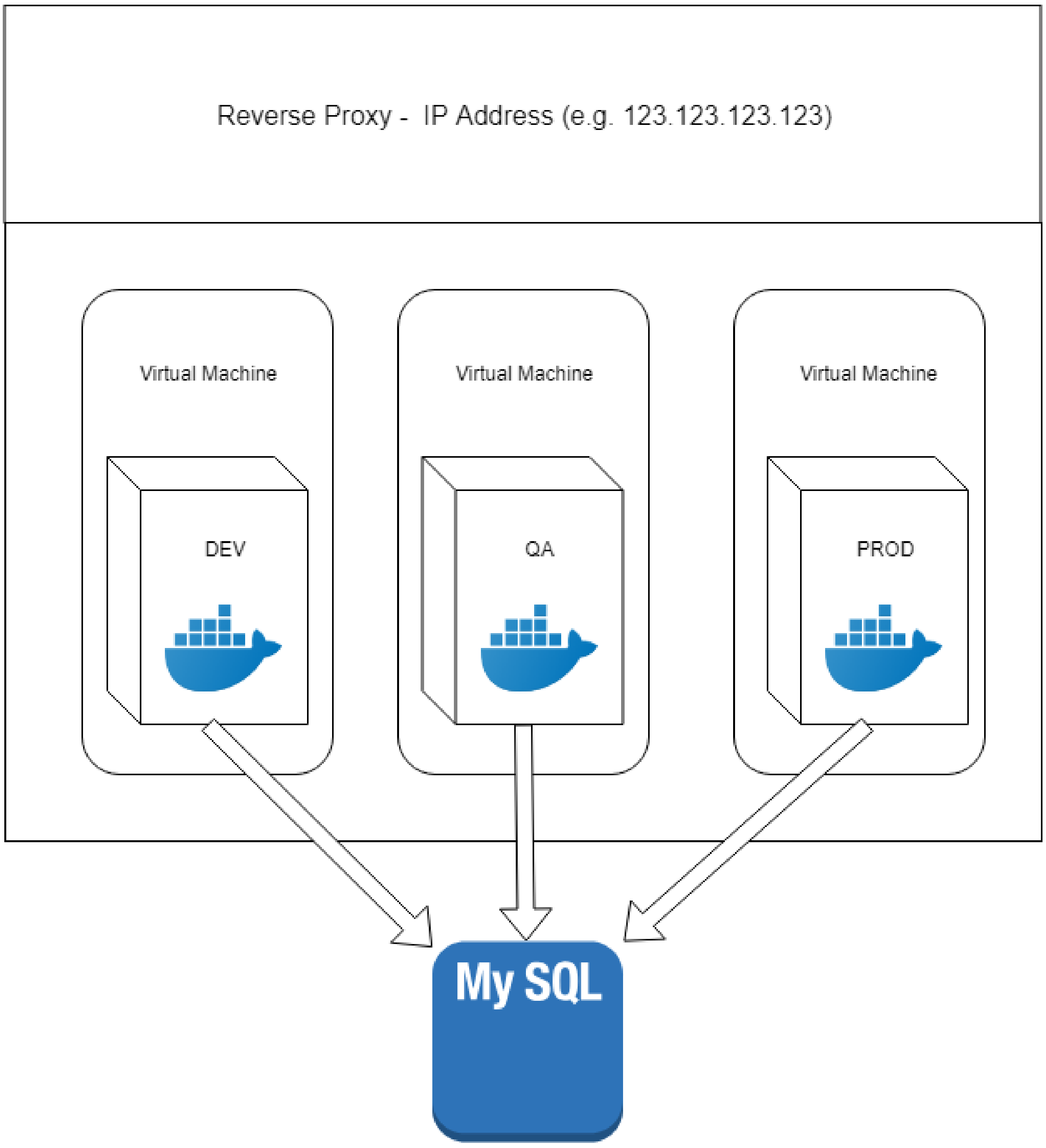
DQP System

https://developer.simplifier.io/documentation/glossar/dqp-system/

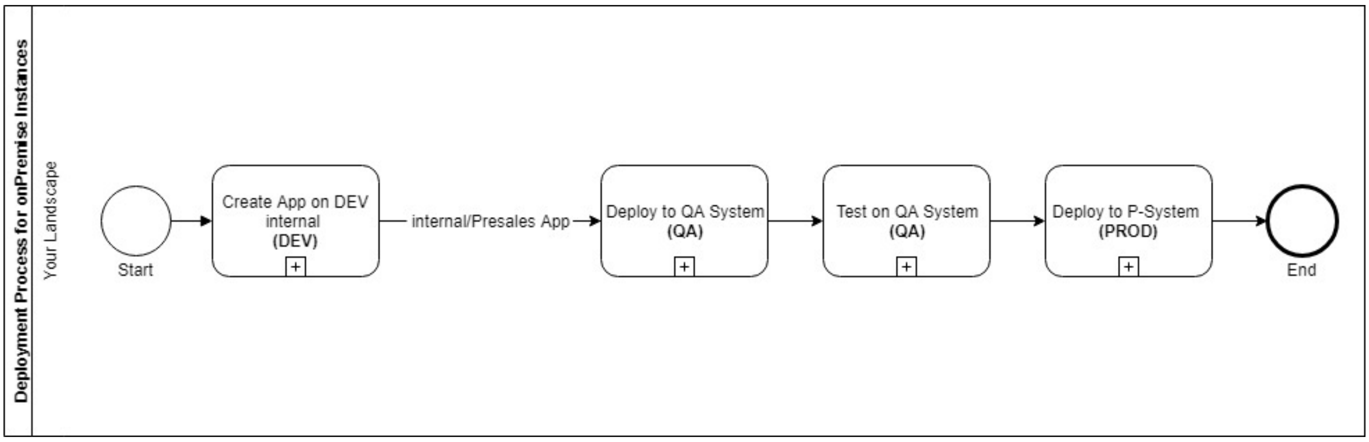
Simplifier usually consists of 3 environments:

- Development
- Quality Assurance
- Productive

An environment can be configured within global Simplifier Settings- Server Environment and is used to transport applications within the defined Simplifier Instances (D,Q,P). Following is an example for deploying a DQP system on three virtual machines.



Deployment and Integration Workflow D-Q-P


Deployment Process for onPremise Instances
Your Landscape
Start
Create App on DEV internal (DEV)
internal/Presales App
Deploy to QA System (QA)
Test on QA System (QA)
Deploy to P-System (PROD)
End

- Development of an app on the DEV instance
- Transport to the QA instance
- Testing the app on the QA instance
- Transport to the Productive instance

Please see also Transports and Remote Transports for detailed information on how to transport Apps within a Simplifier Server Environment

Edit a PDF Template

https://developer.simplifier.io/documentation/plugins/pdf-plugin/technical-call-pdf-plugin/edit-pdf-template/

Edit Template

To edit a PDF template, you need the following parameter:

URL

Input-Parameter

Data

Stylesheet

PreviewJson

Output-Parameter

Example for a call:

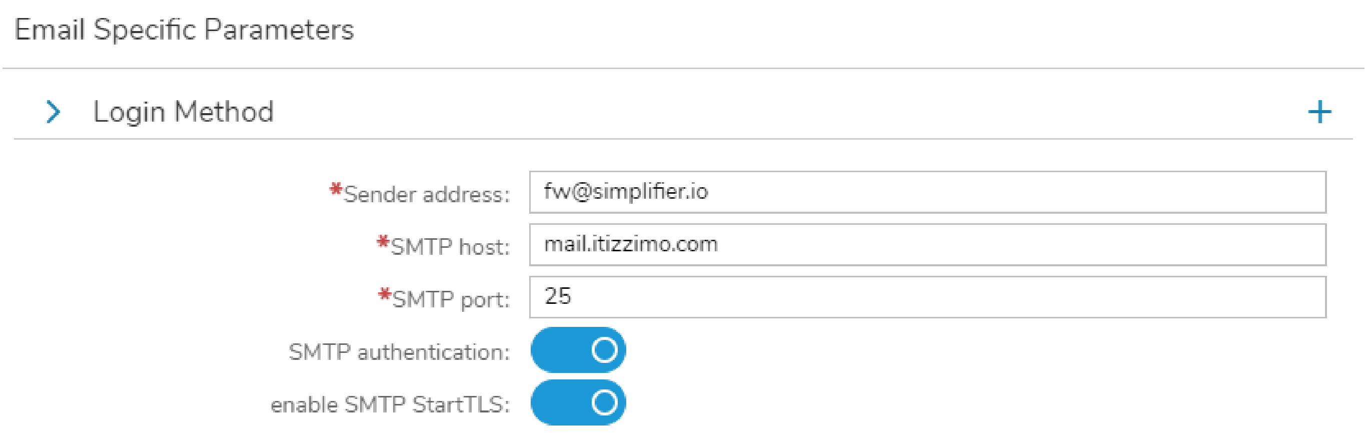
```
{
    "name": "templatename",
    "data": "SGFsbG8gV2VsdA==\",
    "stylesheet: "SGFsbG8gV2VsdA==\",
    "previewJson": "SGFsbG8gV2VsdA==\"
}
```

Output example:

```
{
    "success": true
}
```

Email Connector

https://developer.simplifier.io/documentation/connectors/email-connector-details/


Email Specific Parameters

Login Method

*Sender address:	fw@simplifier.io
*SMTP host:	mail.itizzimo.com
*SMTP port:	25
SMTP authentication:	●
enable SMTP StartTLS:	●

SMTP host

Hostname of SMTP Server

SMPT port

Port of SMTP server

SMTP authentication

If enabled, the client attempt to authenticate the user using the AUTH command. Default to false.

SMTP StartTLS

If activated, it enables the use of the STARTTLS command (if supported by the server) to switch the connection to a TLS-protected connection before issuing any login commands. Note that an appropriate trust store must be configured, so that the client will trust the server's certificate. Default to false.

Take a look at Email Connector Call.

Email Connector Call

https://developer.simplifier.io/documentation/connectors/email-connector-details/email-connector-call/

The email connector call requires 3 input parameters to be defined:

receiver	The email address of the receiver. Several email addresses can be specified separated by a comma	String
subject	The subject of your message	String
msg	The message itself	String

All data types should be set as String.

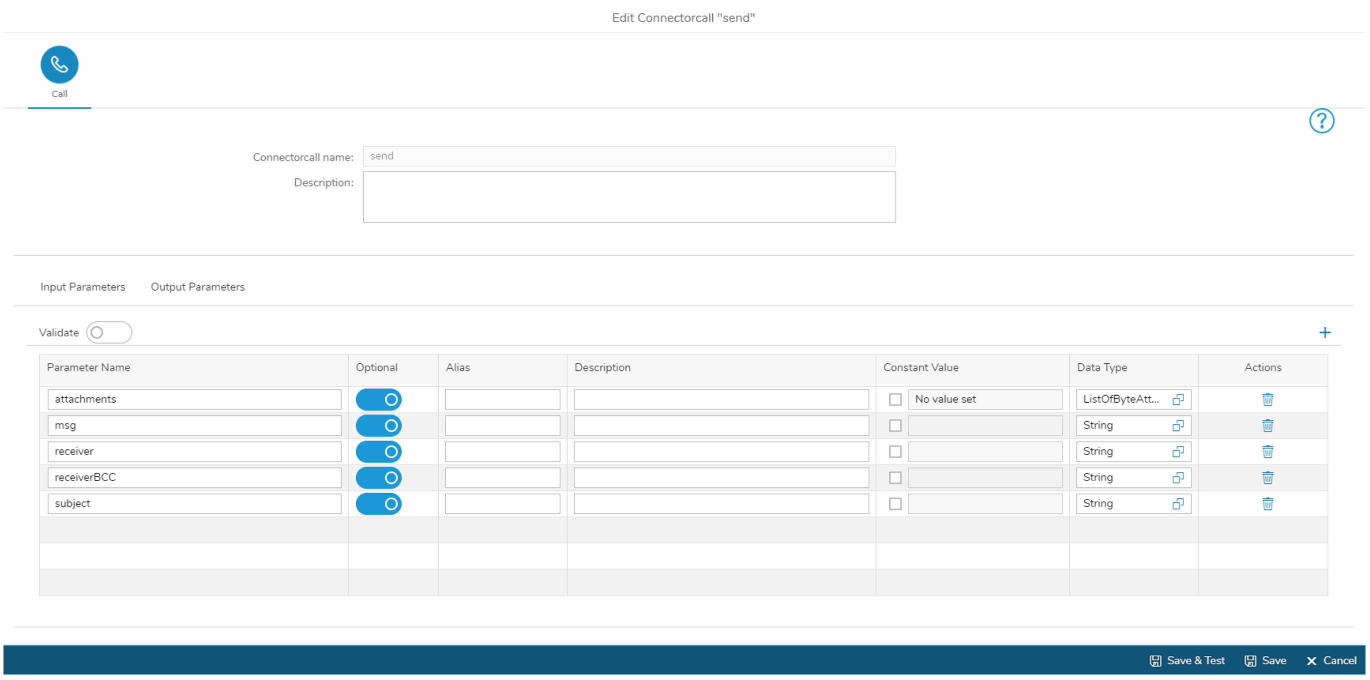
There are 3 optional parameters that can be configured as well:

receiverCC	The email address of the copy receiver. Several email addresses can be specified separated by a comma	String
receiverBCC	The email address of the blind copy receiver. Several email addresses can be specified separated by a comma	String
sender	The email address of the sender	String
attachments	A list of all attachments	List[ByteAttachment]

ByteAttachment

```
{
    "session": String,
    "fileName": String,
    "attachmentMimeType": String
}
```

Example:


Edit Connectorcall "send"

Connectorcall name: send
Description:

Parameter Name	Optional	Alias	Description	Constant Value	Data Type	Actions
attachments	●			☐ No value set	ListOfByteAtt...	🗑
msg	●			☐	String	🗑
receiver	●			☐	String	🗑
receiverBCC	●			☐	String	🗑
subject	●			☐	String	🗑

Enumeration in Widget Properties

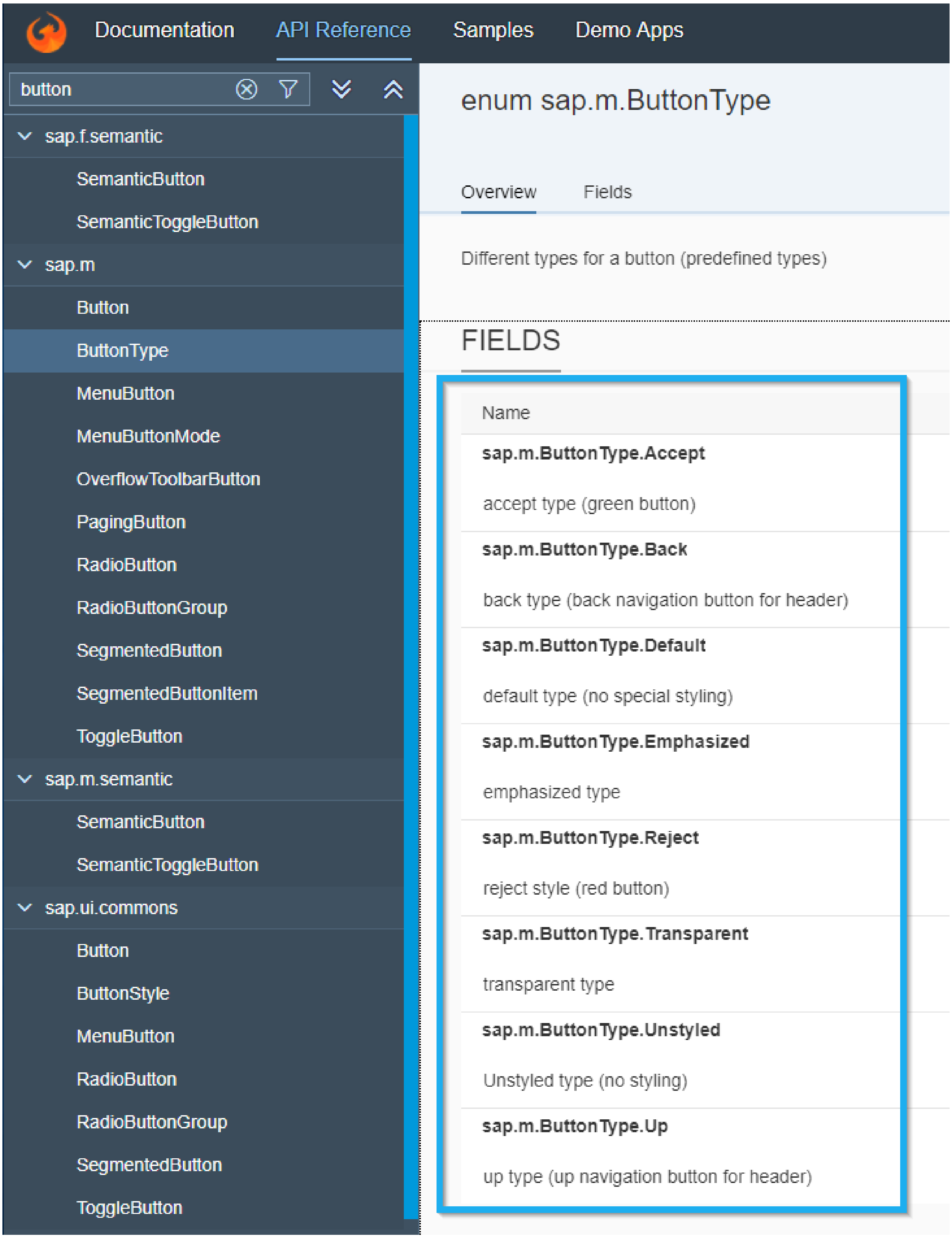
https://developer.simplifier.io/documentation/applications/widget-customizer/enumeration-widget-properties/

You can maintain enumeration for widget properties. Enumerated properties can only hold the defined values and will be displayed as a selection in the UI Designer.

In order to define the enumerations, you have to add them in an array notation to the default value of the property. It is not necessary to set the values in quotes.

Example

There are different predefined types of buttons in OpenUI5. The properties can be maintained as a list in the widget mask. In our example, we look at all different button types in the OpenUi5 API Reference and transfer them into the widget edit mask with the appropriate syntax: [typ1, typ2].

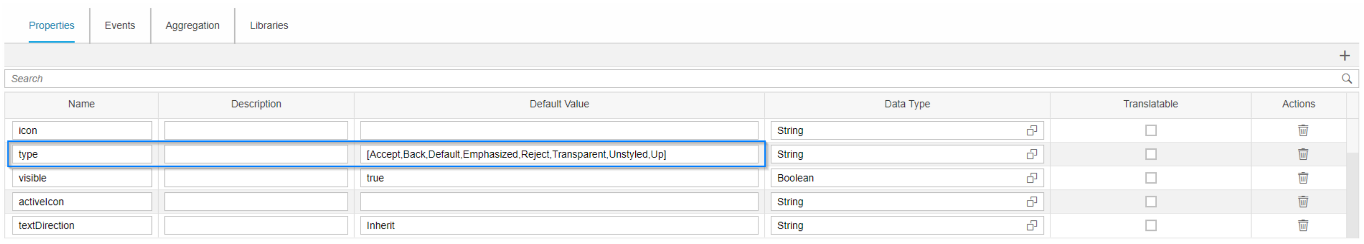


enum sap.m.ButtonType

Different types for a button (predefined types)

FIELDS

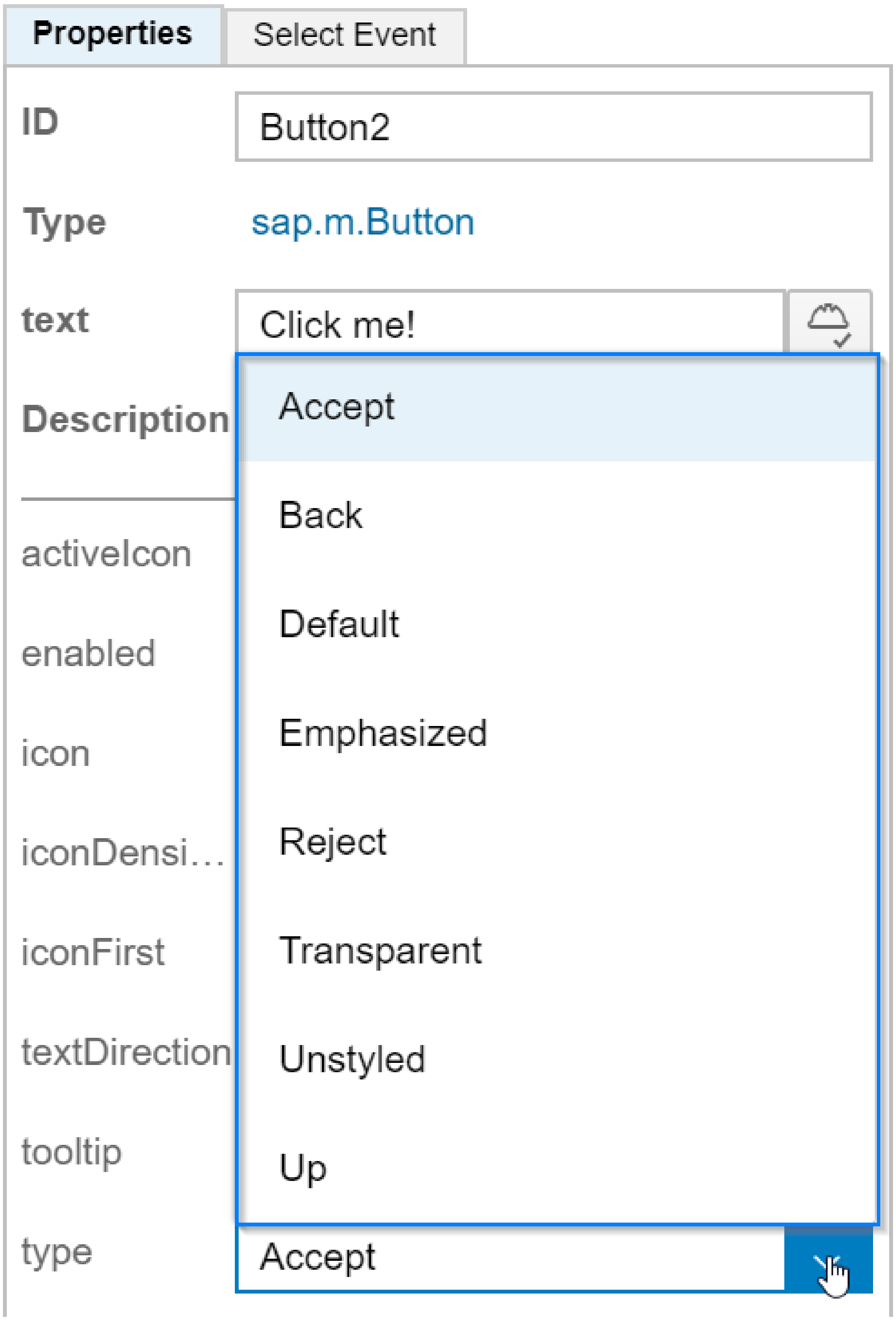
Name
sap.m.ButtonType.Accept accept type (green button)
sap.m.ButtonType.Back back type (back navigation button for header)
sap.m.ButtonType.Default default type (no special styling)
sap.m.ButtonType.Emphasized emphasized type
sap.m.ButtonType.Reject reject style (red button)
sap.m.ButtonType.Transparent transparent type
sap.m.ButtonType.Unstyled Unstyled type (no styling)
sap.m.ButtonType.Up up type (up navigation button for header)



Properties | Events | Aggregation | Libraries

Name	Description	Default Value	Data Type	Translatable	Actions
icon			String	☐	
type		[Accept,Back,Default,Emphasized,Reject,Transparent,Unstyled,Up]	String	☐	
visible		true	Boolean	☐	
activeIcon			String	☐	
textDirection		Inherit	String	☐	

Result in the UI Designer



ID	Button2
Type	sap.m.Button
text	Click me!
Description	
activeIcon	
enabled	
icon	
iconDensi…	
iconFirst	
textDirection	
tooltip	
type	Accept

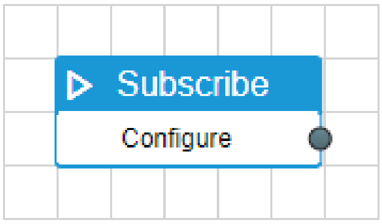
Events

https://developer.simplifier.io/documentation/applications/process-dashboard-and-designer/events/

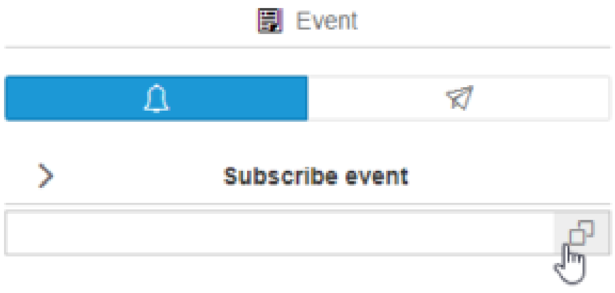
Publish Custom Events | Subscribe to Custom Events

Each workflow in the Process Designer starts with an event.

Click on the Event element under “Activities” on the left pane and drag & drop it into the drawing area.

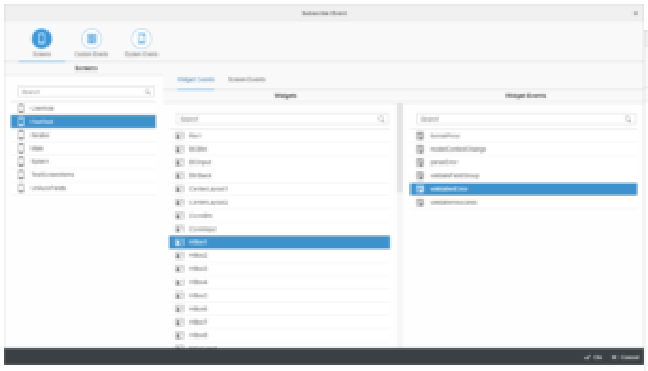


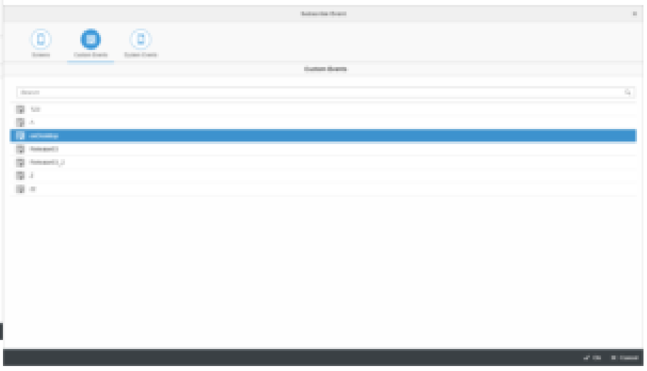
If you double click on the shape of the activity, the event selection assistant opens. You can also open it by clicking right underneath “Subscribe event” / “Publish event”.

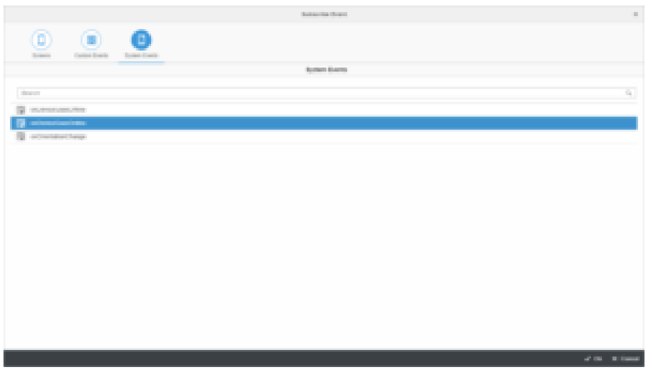


It guides you to your required event. When subscribing to an Event, you can select events from these categories: Widget Events, Screen Events, Custom Events and System Events. Each category is sorted and searchable.

If you want to use the Event in the User Story you are currently working in, select one of the Events under the tab “Subscribe Event”.



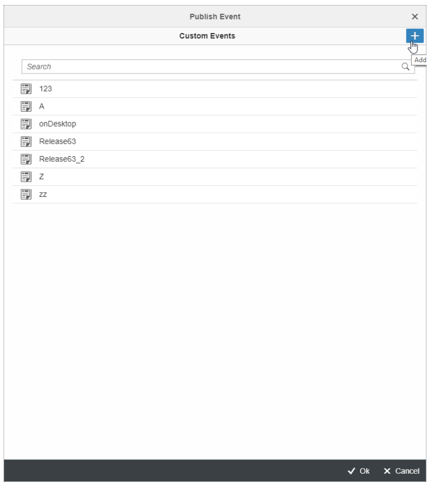




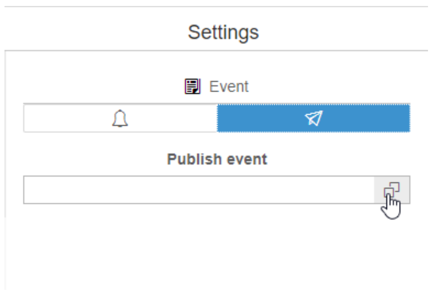
Event Type	**Description**
Widget Events	Events that are available for certain widgets, e.g. button press event.
Screen Events	Events related to screens, e.g. onInit and onAfterShow.
Custom Events	Individual events within an application that can be published and subscribed at any time.
System Events	Events related to devices, e.g. onDeviceGoesOffline and onOrientationChange.

Publish Custom Events

If you want to make an event usable in other user stories, create a new Custom Event under the tab "Publish Event".


Publish Event
Custom Events
Search
123
A
onDesktop
Release63
Release63_2
Z
zz
Ok
Cancel

A shortcut for creating Custom Events can be found in the top right-hand corner.



Subscribe to Custom Events

You can subscribe to a custom event in another user story to connect the logic between different user stories.

Example:

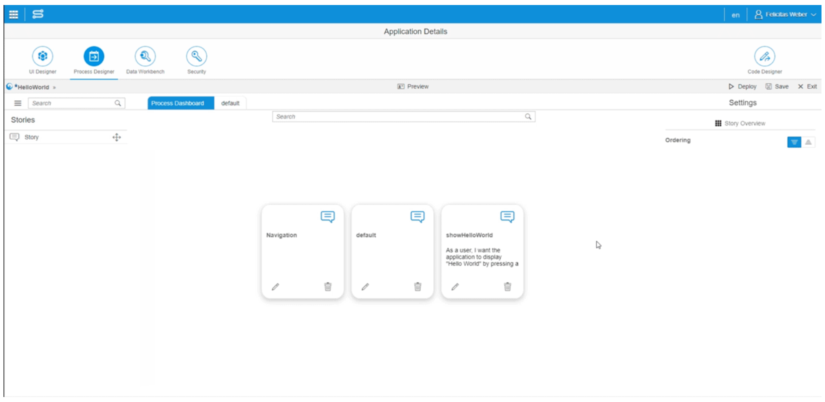
You have a process in User Story 1 that contains a condition to check if an input field is filled out after clicking on a “Login” button. Afterwards, the user should be navigated forward to the next screen.

Imagine you have the other user story exclusively for the whole navigation of your application. So you want the end of the event from User Story 1 (the navigation) subscribed to User Story 2.

Therefore we published the new Custom Event “LoginButton” in User Story 1 and subscribed to it in User Story 2.

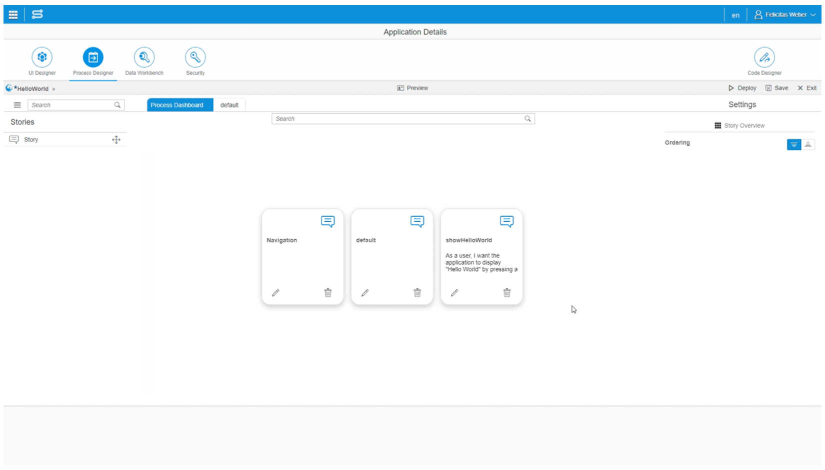
?

User Story 1



As a user, I want the application to display "Hello World" by pressing a

User Story 2



As a user, I want the application to display "Hello World" by pressing a

Custom Events can be maintained in the Data Workbench.

Example Apps

https://developer.simplifier.io/documentation/getting-started/example-apps/

To demonstrate various features of Simplifier the following Example Apps are available with every Simplifier Freemium Instance and can be downloaded here

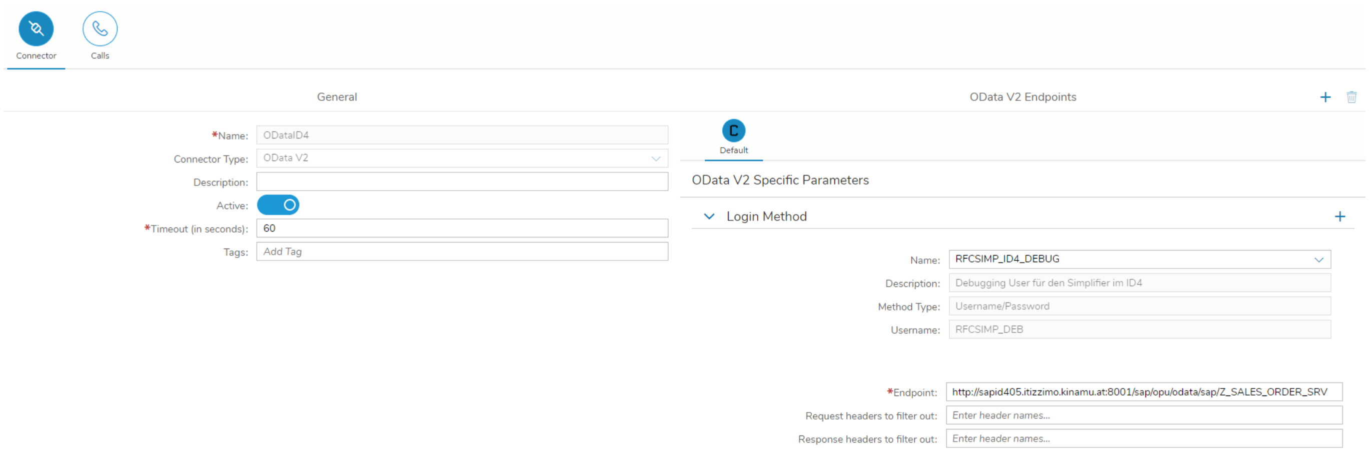
AppName and Description

Example of using OData

https://developer.simplifier.io/documentation/connectors/odata-v2-connector/example-of-using-odata/

In this example, the data of an OData service is displayed in a table by pressing a button.

The connector is configured as follows:



*Name:	ODataID4
Connector Type:	OData V2
Description:	
Active:	
*Timeout (in seconds):	60
Tags:	Add Tag

Name:	RFCSIMP_ID4_DEBUG
Description:	Debugging User für den Simplifier im ID4
Method Type:	Username/Password
Username:	RFCSIMP_DEB

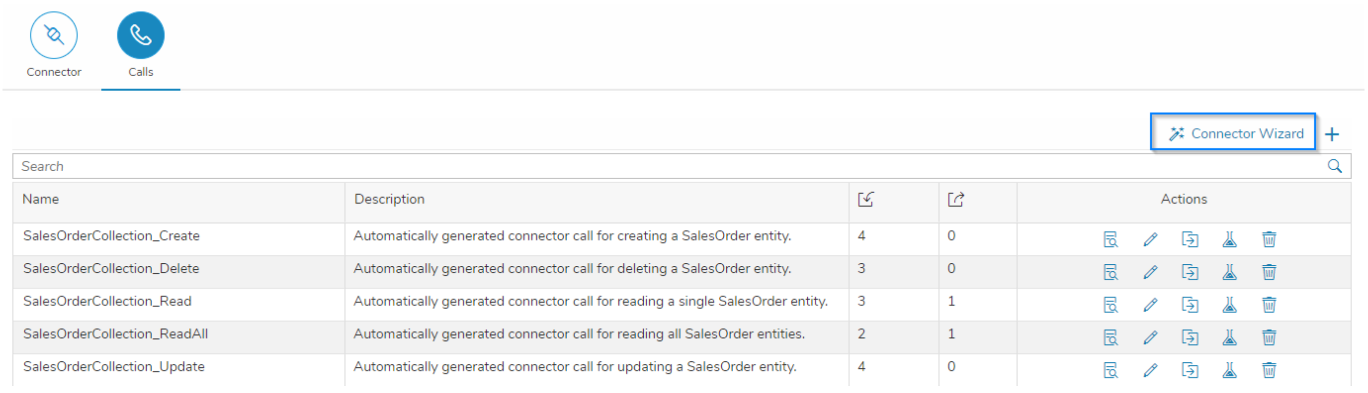
*Endpoint:	http://sapid405.itizzimo.kinamu.at:8001/sap/opu/odata/sap/Z_SALES_ORDER_SRV
Request headers to filter out:	Enter header names...
Response headers to filter out:	Enter header names...

Endpoint:

```
http://sapid405.itizzimo.kinamu.at:8001/sap/opu/odata/sap/Z_SALES_ORDER_SRV
```

Important: Do not skip the login method.

The connector calls were generated automatically using the **Connector Wizard**:



Name	Description			Actions
SalesOrderCollection_Create	Automatically generated connector call for creating a SalesOrder entity.	4	0	
SalesOrderCollection_Delete	Automatically generated connector call for deleting a SalesOrder entity.	3	0	
SalesOrderCollection_Read	Automatically generated connector call for reading a single SalesOrder entity.	3	1	
SalesOrderCollection_ReadAll	Automatically generated connector call for reading all SalesOrder entities.	2	1	
SalesOrderCollection_Update	Automatically generated connector call for updating a SalesOrder entity.	4	0	

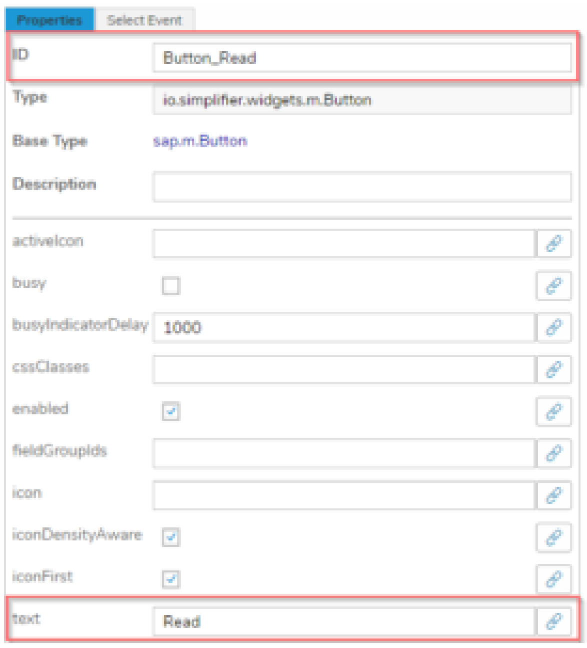
Aside from the automated generated connector calls, you always have the possibility to add other connector calls via the plus icon.

*In this example, we will only go in detail on **SalesOrderCollection_ReadAll**.*

Create the UI

Remember, we want to display the data of the Odata service in a table. The data should be loaded by pressing a button.

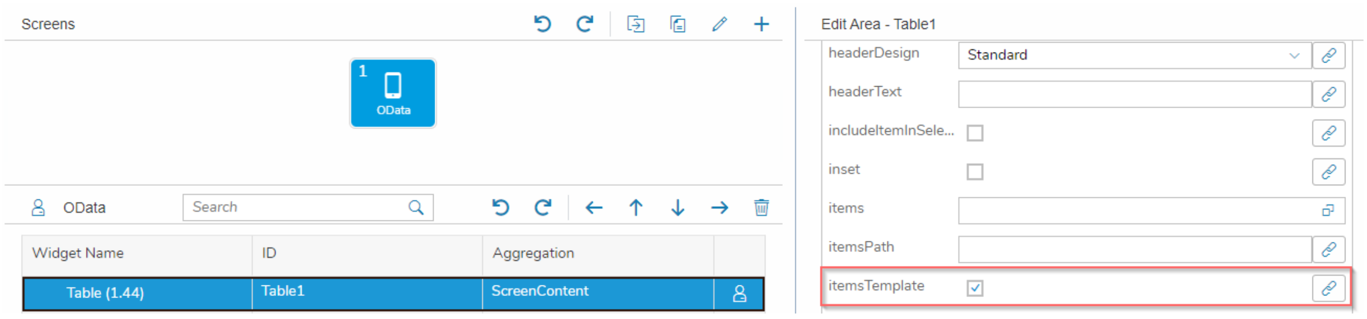
So at first, add the widget **Button** to the screen. Change the ID of the button accordingly (e.g. Button_Read) and add the appropriate **text** (e.g. Read) to the *Properties*.



ID	Button_Read
Type	io.simplifier.widgets.m.Button
Base Type	sap.m.Button
Description	
activeIcon	
busy	☐
busyIndicatorDelay	1000
cssClasses	
enabled	✓
fieldGroupIds	
icon	
iconDensityAware	✓
iconFirst	✓
text	Read

For the UI you also need the Widget **Table**, so add it to the screen. Here you have to customize the properties as follows:

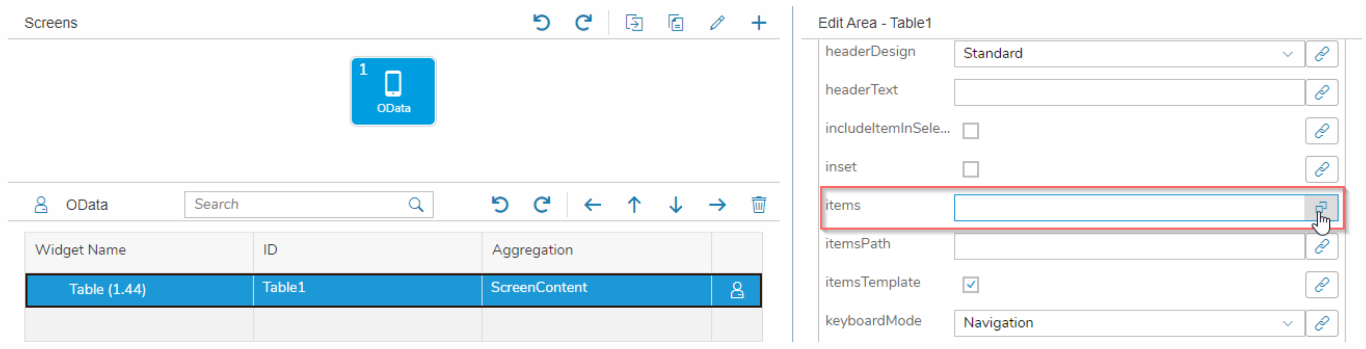
1. Tick the checkbox besides **itemsTemplate**.



Widget Name	ID	Aggregation
Table (1.44)	Table1	ScreenContent

headerDesign	Standard
headerText	
includeItemInSele...	☐
inset	☐
items	
itemsPath	
itemsTemplate	✓

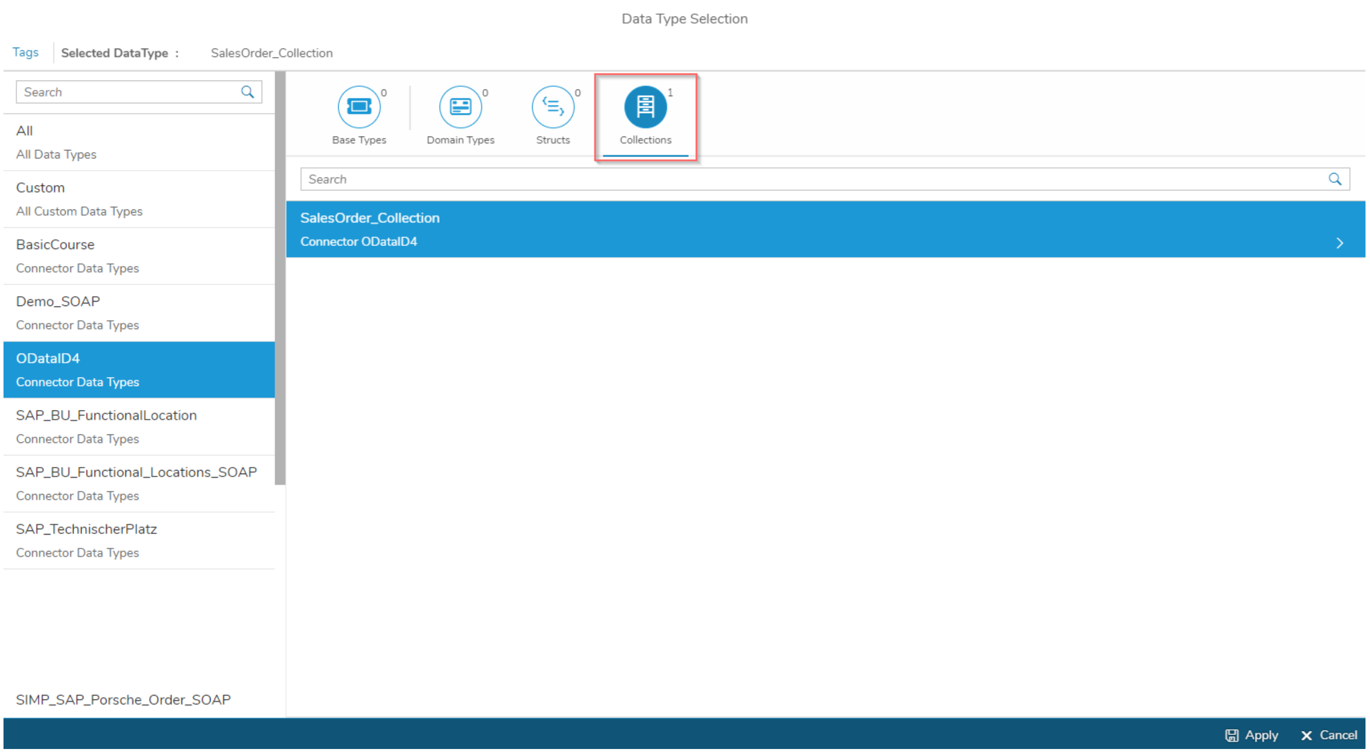
2. Click on the Selection Helper at **items**.



Widget Name	ID	Aggregation
Table (1.44)	Table1	ScreenContent

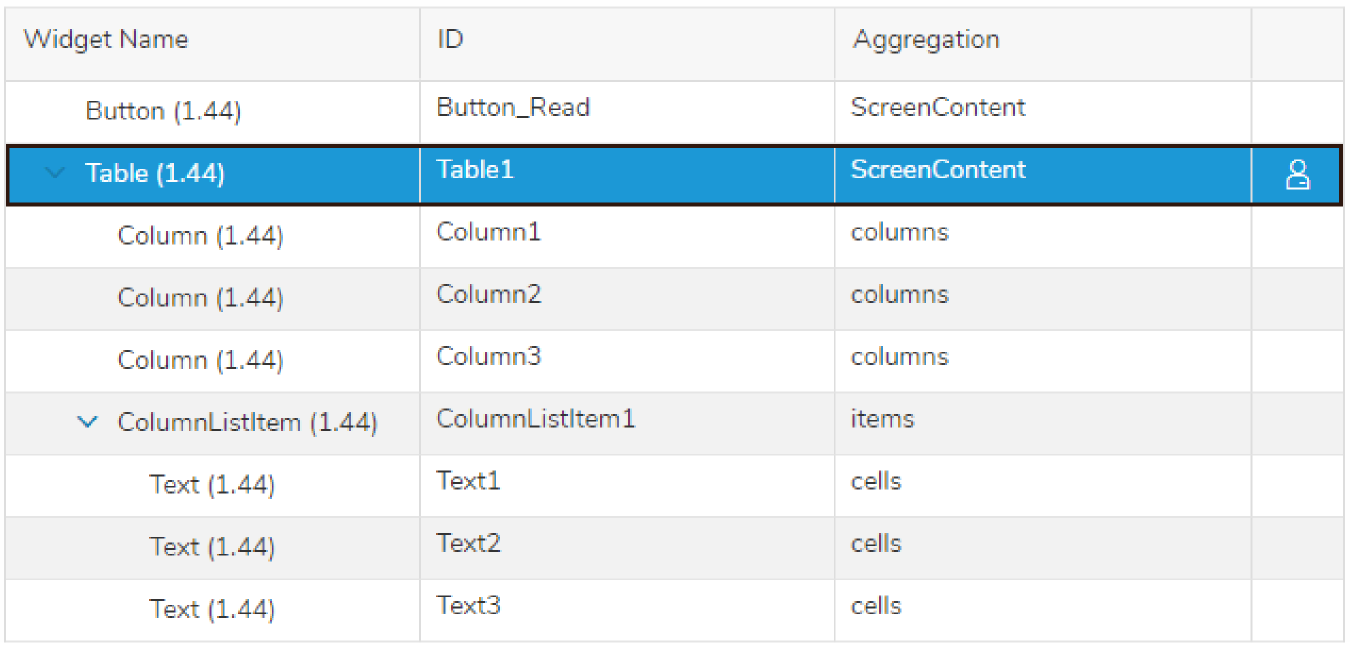
headerDesign	Standard
headerText	
includeItemInSele...	☐
inset	☐
items	
itemsPath	
itemsTemplate	✓
keyboardMode	Navigation

A dialog opens in which you have to select the connector on the left and then the ***Collection*** *SalesOrder_Collection*.



Since we want to display three values of the OData service, we still need three **Columns** in the table and a **ColumnListItem** that contains three **Text** widgets.

Your screen content should look like this:

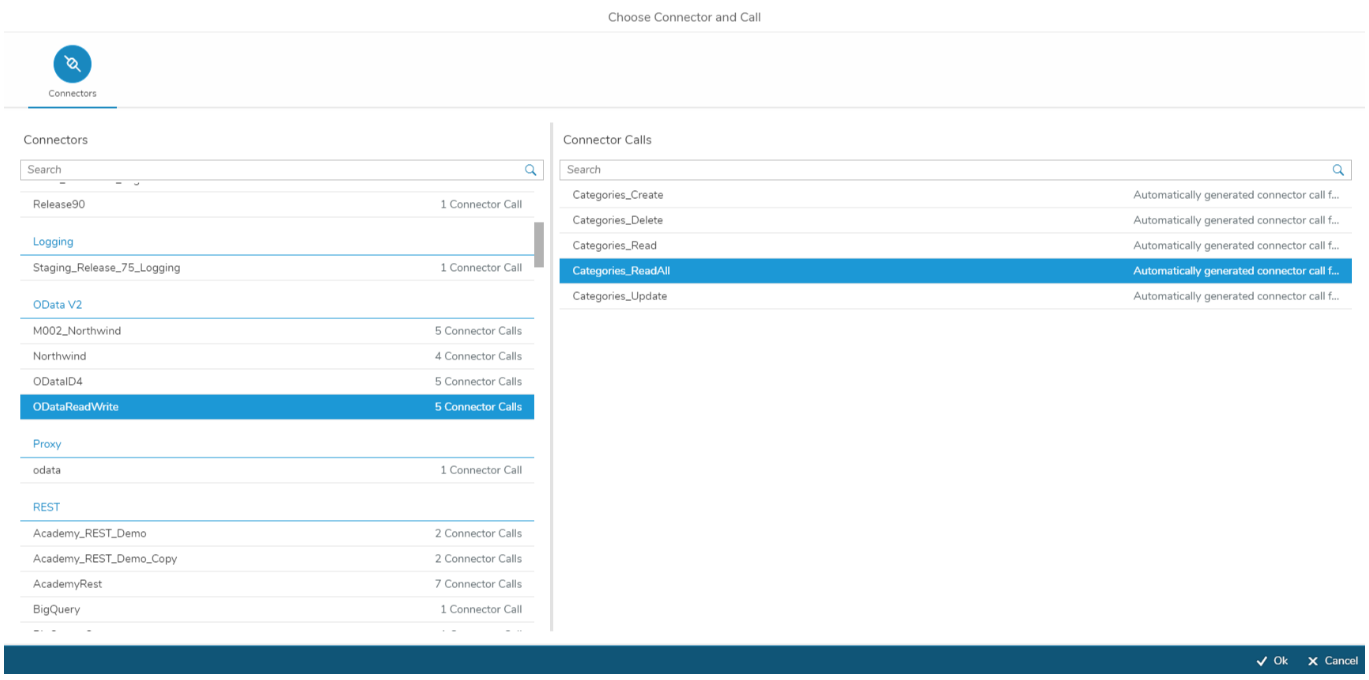


Widget Name	ID	Aggregation	
Button (1.44)	Button_Read	ScreenContent	
Table (1.44)	Table1	ScreenContent	
Column (1.44)	Column1	columns	
Column (1.44)	Column2	columns	
Column (1.44)	Column3	columns	
ColumnListItem (1.44)	ColumnListItem1	items	
Text (1.44)	Text1	cells	
Text (1.44)	Text2	cells	
Text (1.44)	Text3	cells	

Switch to the Process Designer and create the user story for the process logic.

Configure the Process Logic

Start the process with a Subscribe event of the read button (Widget Event: press). Then add the *Data Object* Connector and select the *Connector* and the corresponding *Connector Call* **Categories_ReadAll**.



Choose Connector and Call

Connectors

Release90	1 Connector Call
Logging	
Staging_Release_75_Logging	1 Connector Call
OData V2	
M002_Northwind	5 Connector Calls
Northwind	4 Connector Calls
ODataID4	5 Connector Calls
ODataReadWrite	5 Connector Calls
Proxy	
odata	1 Connector Call
REST	
Academy_REST_Demo	2 Connector Calls
Academy_REST_Demo_Copy	2 Connector Calls
AcademyRest	7 Connector Calls
BigQuery	1 Connector Call

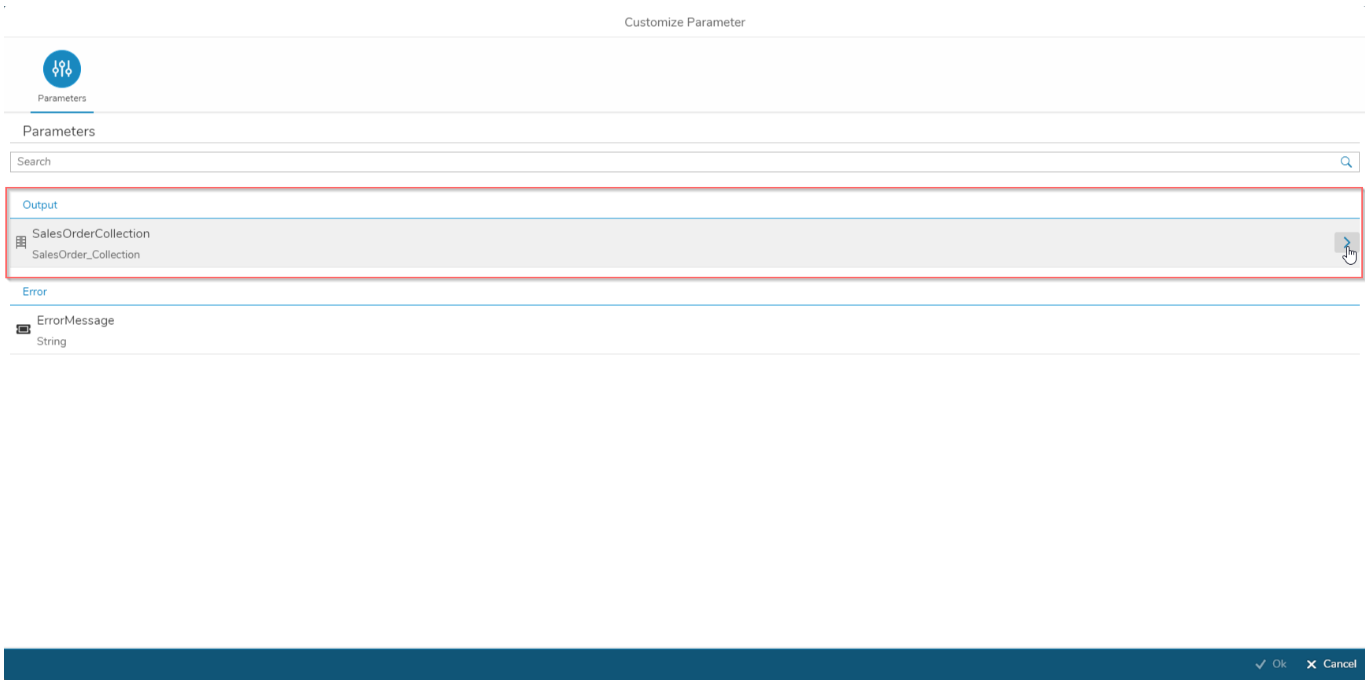
Connector Calls

Categories_Create	Automatically generated connector call f...
Categories_Delete	Automatically generated connector call f...
Categories_Read	Automatically generated connector call f...
Categories_ReadAll	Automatically generated connector call f...
Categories_Update	Automatically generated connector call f...

For the Read Connector Call you don't need an Input Mapping, because you don't have to pass a value to read all data.

Only the output mapping has to be defined.

Open the **Output Mapping** and drag the **Parameter** from the left. Go deeper into *Output* – **SalesOrderCollection**.


Customize Parameter

Parameters

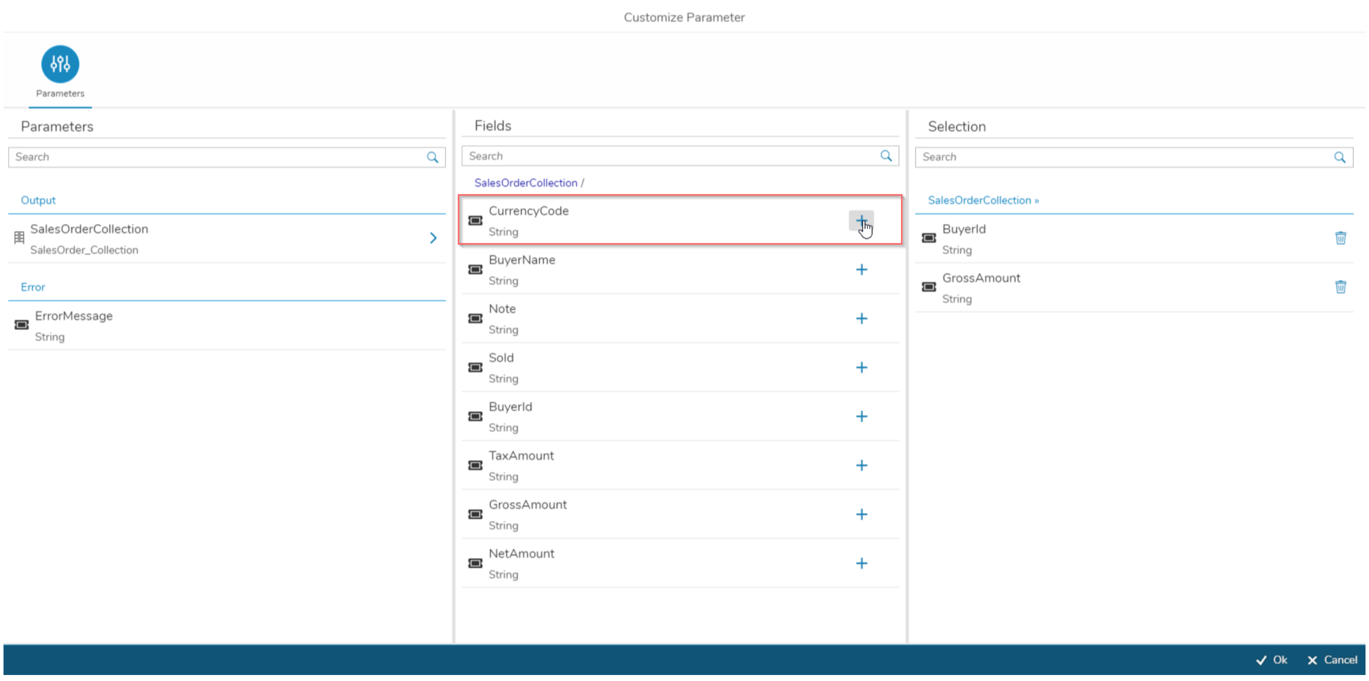
Output

SalesOrderCollection
SalesOrder_Collection

Error

ErrorMessage
String

Select the fields you need. For example, select **BuyerId**, **GrossAmount** and **CurrencyCode** by clicking on the plus.

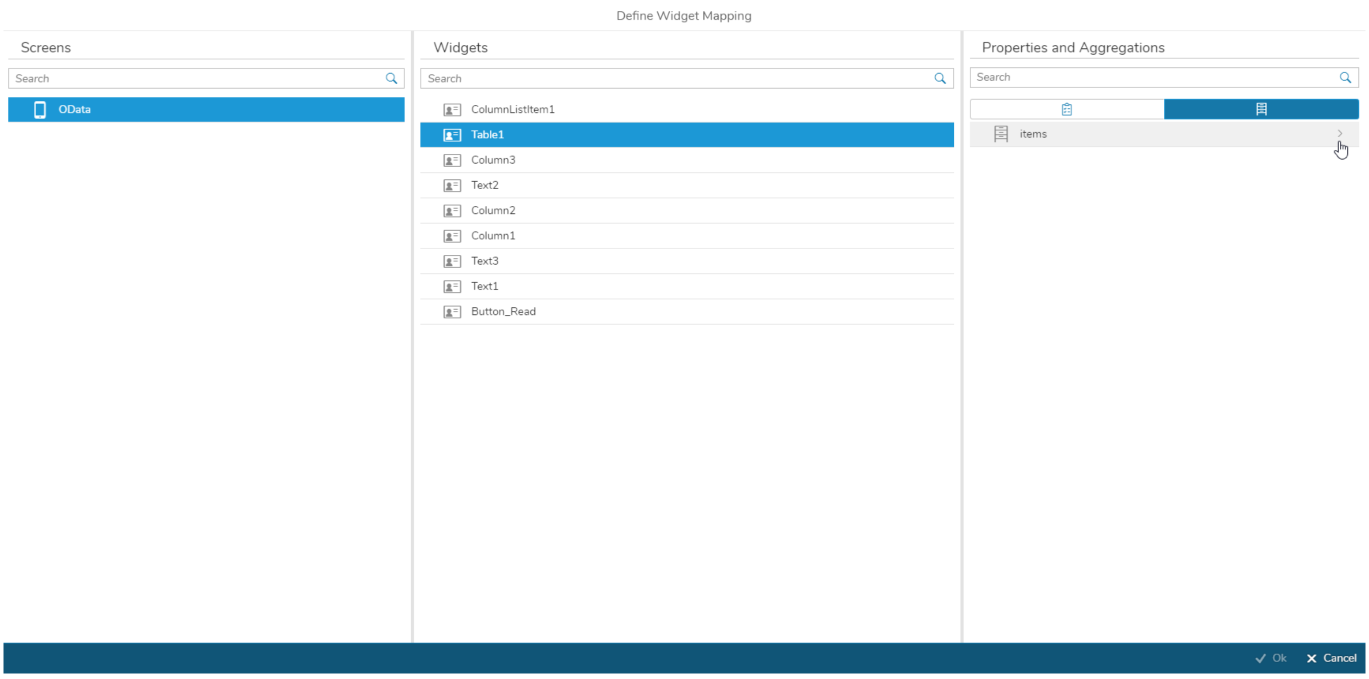

Customize Parameter
Parameters
Parameters
Search
Output
SalesOrderCollection
SalesOrder_Collection
Error
ErrorMessage
String
Fields
Search
SalesOrderCollection /
CurrencyCode
String
BuyerName
String
Note
String
SoId
String
BuyerId
String
TaxAmount
String
GrossAmount
String
NetAmount
String
Selection
Search
SalesOrderCollection »
BuyerId
String
GrossAmount
String
Ok
Cancel

After you have added the fields, click **Ok**.

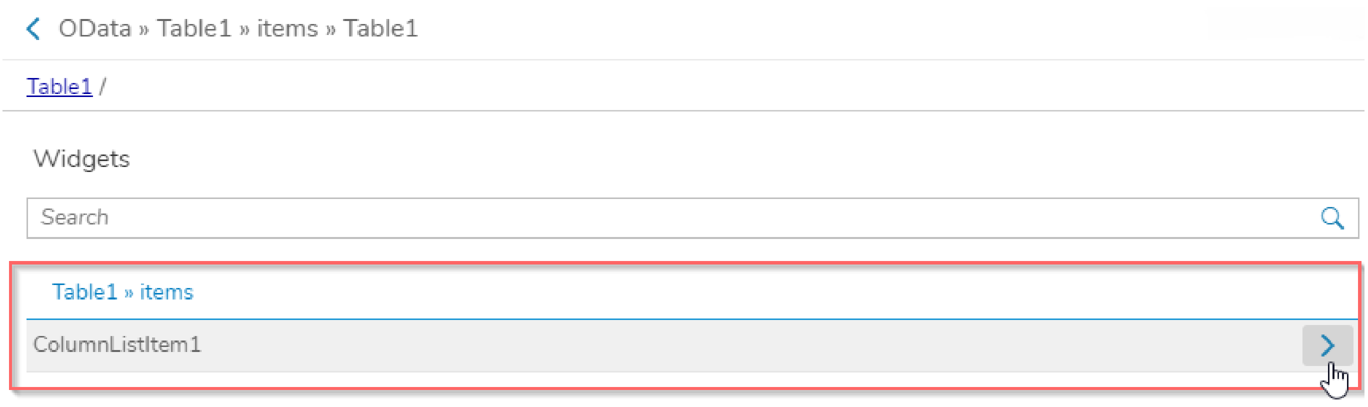
Now you have to define the **widgets** (drag it from the right) in which the selected parameters should be mapped.

Select the screen, the widget **Table** and switch to **Data Aggregation** within the section *Properties and Aggregations*.

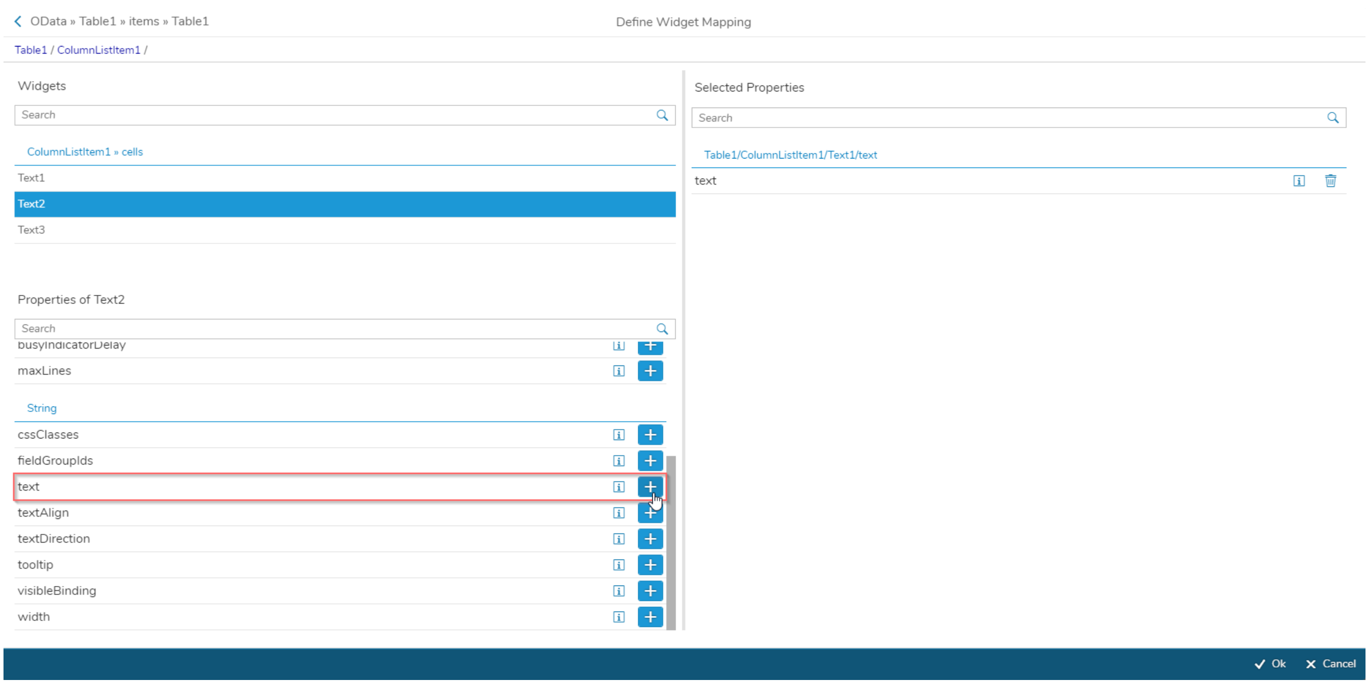
Go deeper into **items**.


Define Widget Mapping
Screens
Search
OData
Widgets
Search
ColumnListItem1
Table1
Column3
Text2
Column2
Column1
Text3
Text1
Button_Read
Properties and Aggregations
Search
items
Ok
Cancel

Now go deeper into the items of the Table - **ColumnListItem**.

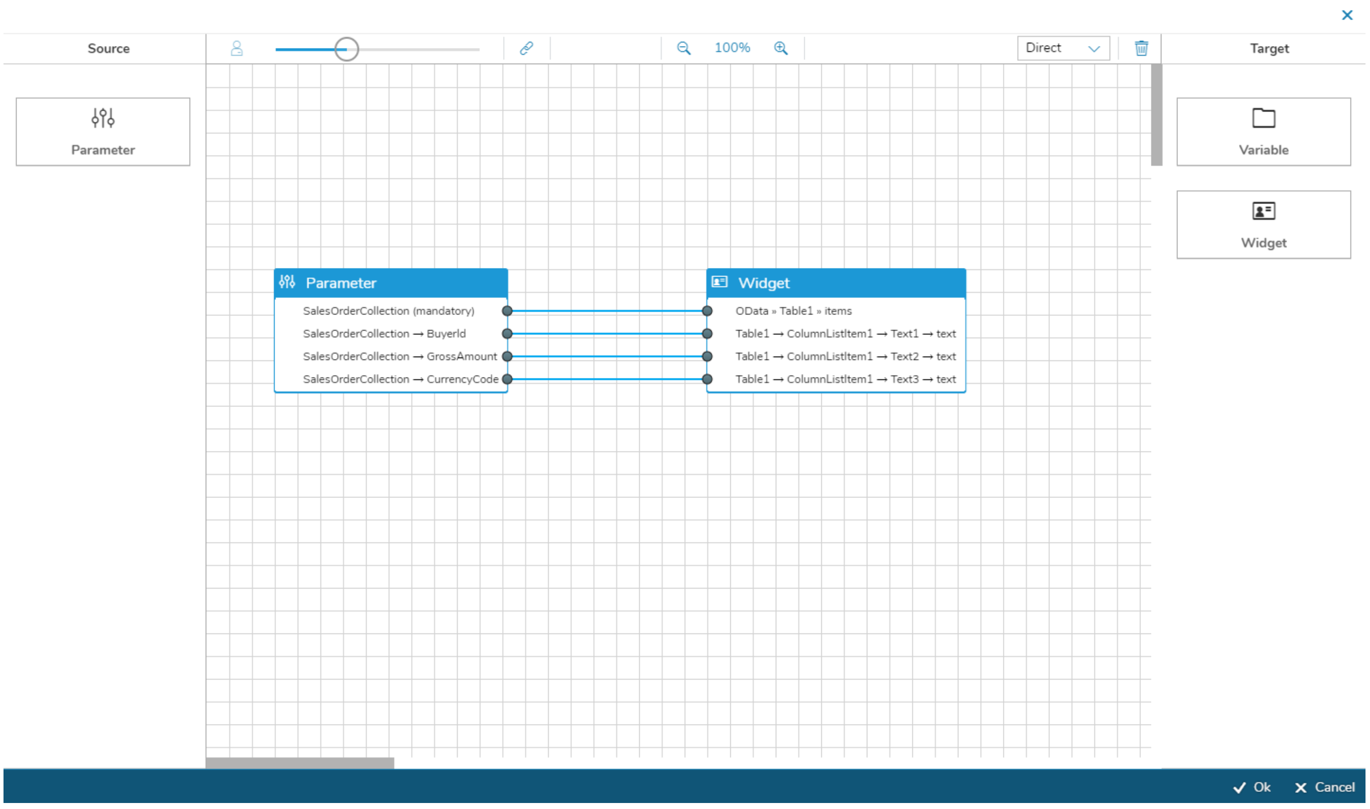


You will now see the three **Text** widgets as cells of the list item. For each T**ext** you have to select String **text** as property. It then appears on the right under *Selected Properties*.

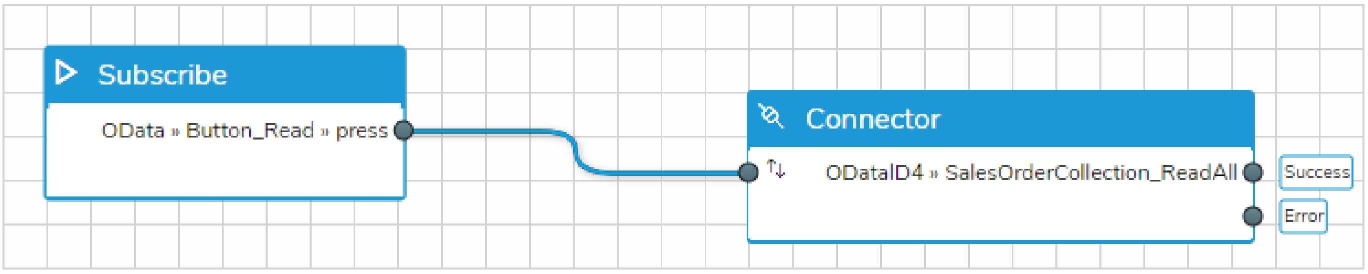

OData » Table1 » items » Table1
Define Widget Mapping
Table1 / ColumnListItem1 /
Widgets
Search
ColumnListItem1 » cells
Text1
Text2
Text3
Properties of Text2
Search
busyIndicatorDelay
maxLines
String
cssClasses
fieldGroupIds
text
textAlign
textDirection
tooltip
visibleBinding
width
Selected Properties
Search
Table1/ColumnListItem1/Text1/text
text
Ok
Cancel

When you've done this for all three of them, click **Ok**.

Now map the parameters into the table. The **Output Mapping** should look like this:

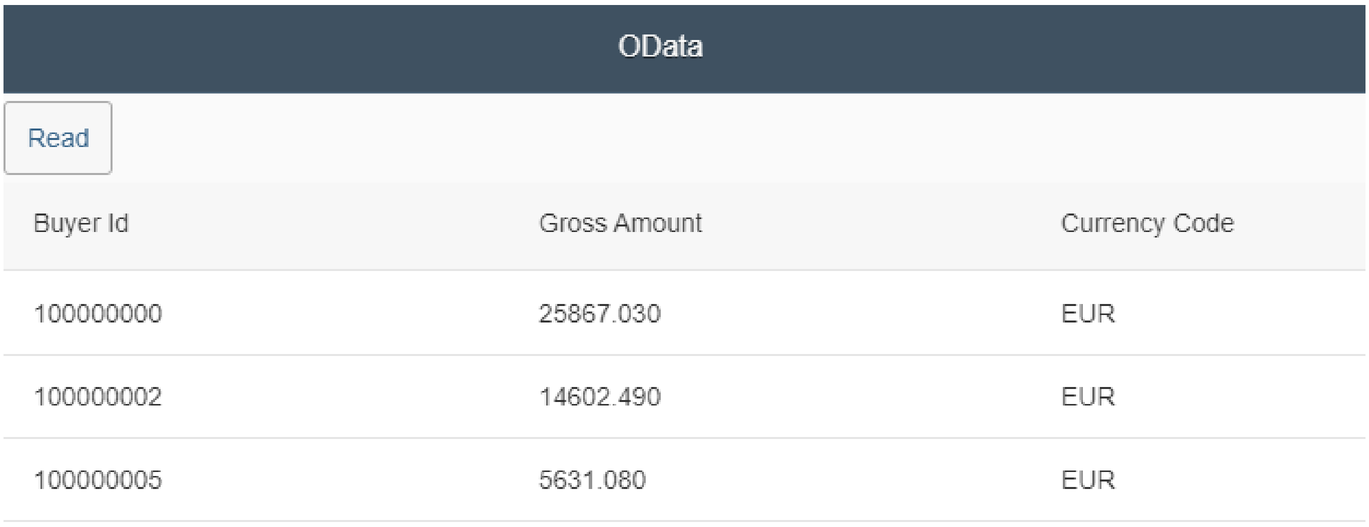

Source
100%
Direct
Target
Parameter
Variable
Widget
Parameter
SalesOrderCollection (mandatory)
SalesOrderCollection → BuyerId
SalesOrderCollection → GrossAmount
SalesOrderCollection → CurrencyCode
Widget
OData » Table1 » items
Table1 → ColumnListItem1 → Text1 → text
Table1 → ColumnListItem1 → Text2 → text
Table1 → ColumnListItem1 → Text3 → text
Ok
Cancel

Connect the two shapes with each other:


Subscribe
OData » Button_Read » press
Connector
ODataID4 » SalesOrderCollection_ReadAll
Success
Error

[Make sure that the business application has the appropriate permissions to execute the connector.]

After successful **deployment** the data will be read and displayed by clicking the button.



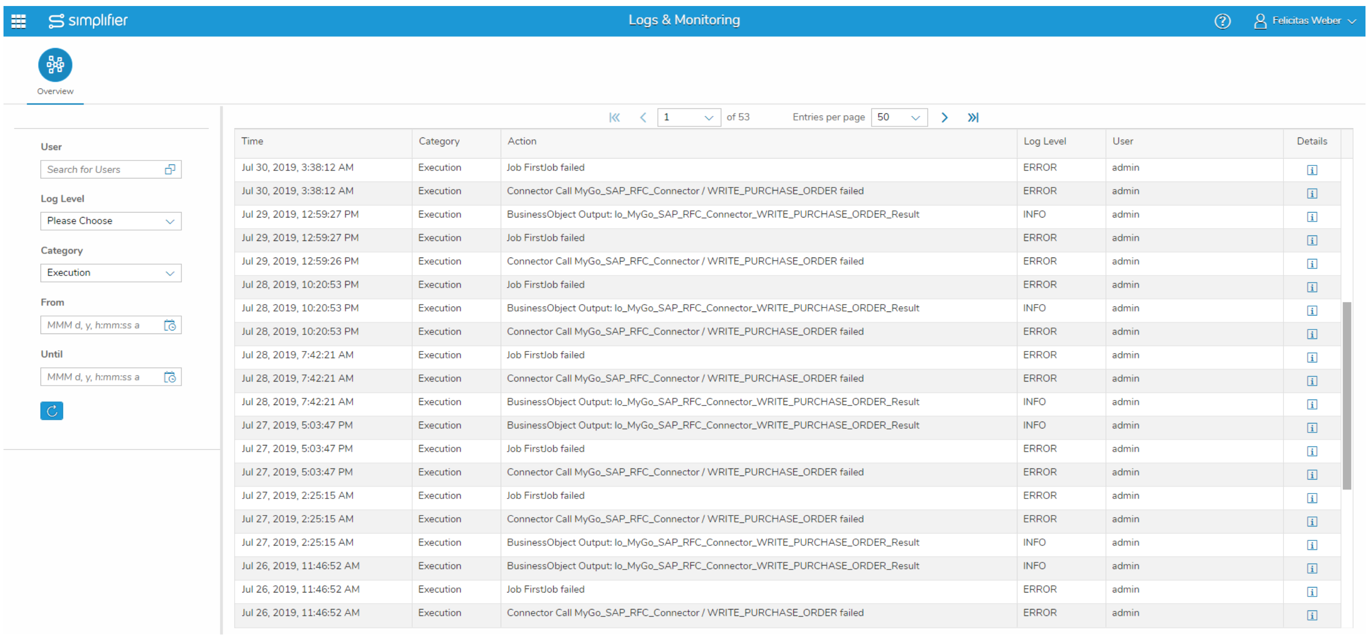
Buyer Id	Gross Amount	Currency Code
100000000	25867.030	EUR
100000002	14602.490	EUR
100000005	5631.080	EUR

For this screenshot, in the UI Designer a label has been added to each column for the Buyer Id, Gross Amount and Currency Code.

Execution Log

https://developer.simplifier.io/documentation/logging-monitoring/execution-log/

You can use the execution log to trace the execution of e.g. connectors.



User
Search for Users
Log Level
Please Choose
Category
Execution
From
MMM d, y, h:mm:ss a
Until
MMM d, y, h:mm:ss a

1 of 53 Entries per page 50

Time	Category	Action	Log Level	User	Details
Jul 30, 2019, 3:38:12 AM	Execution	Job FirstJob failed	ERROR	admin	ⓘ
Jul 30, 2019, 3:38:12 AM	Execution	Connector Call MyGo_SAP_RFC_Connector / WRITE_PURCHASE_ORDER failed	ERROR	admin	ⓘ
Jul 29, 2019, 12:59:27 PM	Execution	BusinessObject Output: lo_MyGo_SAP_RFC_Connector_WRITE_PURCHASE_ORDER_Result	INFO	admin	ⓘ
Jul 29, 2019, 12:59:27 PM	Execution	Job FirstJob failed	ERROR	admin	ⓘ
Jul 29, 2019, 12:59:26 PM	Execution	Connector Call MyGo_SAP_RFC_Connector / WRITE_PURCHASE_ORDER failed	ERROR	admin	ⓘ
Jul 28, 2019, 10:20:53 PM	Execution	Job FirstJob failed	ERROR	admin	ⓘ
Jul 28, 2019, 10:20:53 PM	Execution	BusinessObject Output: lo_MyGo_SAP_RFC_Connector_WRITE_PURCHASE_ORDER_Result	INFO	admin	ⓘ
Jul 28, 2019, 10:20:53 PM	Execution	Connector Call MyGo_SAP_RFC_Connector / WRITE_PURCHASE_ORDER failed	ERROR	admin	ⓘ
Jul 28, 2019, 7:42:21 AM	Execution	Job FirstJob failed	ERROR	admin	ⓘ
Jul 28, 2019, 7:42:21 AM	Execution	Connector Call MyGo_SAP_RFC_Connector / WRITE_PURCHASE_ORDER failed	ERROR	admin	ⓘ
Jul 28, 2019, 7:42:21 AM	Execution	BusinessObject Output: lo_MyGo_SAP_RFC_Connector_WRITE_PURCHASE_ORDER_Result	INFO	admin	ⓘ
Jul 27, 2019, 5:03:47 PM	Execution	BusinessObject Output: lo_MyGo_SAP_RFC_Connector_WRITE_PURCHASE_ORDER_Result	INFO	admin	ⓘ
Jul 27, 2019, 5:03:47 PM	Execution	Job FirstJob failed	ERROR	admin	ⓘ
Jul 27, 2019, 5:03:47 PM	Execution	Connector Call MyGo_SAP_RFC_Connector / WRITE_PURCHASE_ORDER failed	ERROR	admin	ⓘ
Jul 27, 2019, 2:25:15 AM	Execution	Job FirstJob failed	ERROR	admin	ⓘ
Jul 27, 2019, 2:25:15 AM	Execution	Connector Call MyGo_SAP_RFC_Connector / WRITE_PURCHASE_ORDER failed	ERROR	admin	ⓘ
Jul 27, 2019, 2:25:15 AM	Execution	BusinessObject Output: lo_MyGo_SAP_RFC_Connector_WRITE_PURCHASE_ORDER_Result	INFO	admin	ⓘ
Jul 26, 2019, 11:46:52 AM	Execution	BusinessObject Output: lo_MyGo_SAP_RFC_Connector_WRITE_PURCHASE_ORDER_Result	INFO	admin	ⓘ
Jul 26, 2019, 11:46:52 AM	Execution	Job FirstJob failed	ERROR	admin	ⓘ
Jul 26, 2019, 11:46:52 AM	Execution	Connector Call MyGo_SAP_RFC_Connector / WRITE_PURCHASE_ORDER failed	ERROR	admin	ⓘ

The following type of entries are logged:

Type	**Description**
Open App	When the direct path to an app is opened (appDirect)
Download App	When downloading the app to the client (user context is provided)
Connector Execution	When using a connector directly, the execution and payload will be logged
Connector Call Execution	When a connector call is invoked. All parameters, even the constant parameters are logged
Business Object Execution	When using a business object, the payload and parameters are logged
Plugins	Plugins which are called by the old Akka interface
Asynchronous Connectors	When subscribing and unsubscribing a connector
Job Execution	Every execution of the job
Any above	Any exception by executing an artifact above

Features and supported operating systems

https://developer.simplifier.io/documentation/getting-started/simplifier-mobile-client/features-and-supported-operating-systems/

Auto-Login Functionality
OAuth Login
Developer Mode
App Autostart

See the full-featured list of supported operating systems:

Plugin	Description	Android > 5.1+	iOS > 10.x+	Windows 10
cordova-plugin-background-mode	For the Cordova framework to perform infinite background execution	?	?	
cordova-plugin-badge	Displays a badge number beside the Simplifier icon	?	?	
cordova-plugin-barcodescanner	Scans many different kinds of barcodes	?	?	?
cordova-plugin-ble-central	Enables communication betweena phone and Bluetooth Low Energy (BLE) peripherals	?	?	?
cordova-plugin-device	Defines a global device object, which describes the device's hardware	?	?	?
cordova-plugin-device-motion	Provides access to the device's accelerometer	?	?	?
cordova-plugin-device-orientation	Provides information about the physical orientation and motion of the device	?	?	?
cordova-plugin-file	Implements a File API allowing read/write access to files residing on the device	?	?	?
cordova-plugin-device-file-transfer	Allows you to upload and download files	?	?	?
cordova-plugin-dialogs	Provides access to some native dialog UI elements	?	?	?
cordova-plugin-embedded-pdf-viewer	Views PDF files	?	?	?
cordova-plugin-fullscreen	Interactive fullscreen mode	?		?
cordova-plugin-geolocation	Provides information about the device's location, such as latitude and longitude	?	?	?
cordova-plugin-ibeacon	Provides the functionality to use beacons with the iBeacon protocol	?	?	
cordova-plugin-inappbrowser	Provides an in-app browser	?	?	
cordova-plugin-conferencing	Plugin to provide WebRTC Conferencing functionality via Open WebRTC Toolkit Media Server	?	?	
cordova-plugin-keyboard	Controls your soft keyboard	?	?	
cordova-plugin-local-notifications	Allows to schedule multiple notifications at once	?	?	?
cordova-plugin-media	Provides the ability to record and play back audio files on a device	?	?	
cordova-plugin-media-	Provides access to the device's audio, image, and video	?	?	?

capture	capture capabilities			
cordova-plugin-network-information	Provides an implementation of an old version of the Network Information API	?	?	?
cordova-plugin-photo-library	Provides access to your photo libraries	?	?	
cordova-plugin-screen-orientation	Sets/locks the screen orientation in a common way	?	?	
cordova-plugin-speechrecognition	Speech Recognition functionality	?	?	
cordova-plugin-statusbar	Enables the user to make changes to the status bar of a mobile device	?	?	
cordova-plugin-streaming-media	Allows you to stream audio and video in a fullscreen, native player on iOS and Android	?	?	
cordova-plugin-tts	Text to Speech functionality	?	?	?
cordova-plugin-vibration	Provides a way to vibrate the device	?	?	?
cordova-sqlite-storage	A Cordova/PhoneGap plugin to open and use sqlite databases	?	?	?
Flashlight-PhoneGap-Plugin	Allows you to switch the flashlight / torch of the device on and off	?	?	
Insomnia-PhoneGap-Plugin	Prevents the screen of the mobile device from falling asleep	?	?	
phonegap-nfc	Allows you to read and write NFC tags	?	?	?
phonegap-plugin-battery-status	Provides an implementation based on the W3C Battery Status Events API	?	?	?
SocialSharing-PhoneGap-Plugin	Allows you to use the native sharing window of your mobile device	?	?	?
sockets-for-cordova	Provides JavaScript API, that allows you to communicate with server through TCP protocol	?	?	
wikitude-cordova-plugin	Provides augmented reality functionality by Wikitude	?	?	

Fetch a PDF Template

https://developer.simplifier.io/documentation/plugins/pdf-plugin/technical-call-pdf-plugin/fetch-pdf-template/

Fetch Template

To fetch a PDF template, you need the following parameter:

URL	/client/1.0/PLUGIN/pdfPlugin/adminTemplateFetch		
Input-Parameter	**Name**		Template name
Output-Parameter	**Value**	**Template**	HTML Template Content (Base64-coded)
		Stylesheet	Content of the LESS Stylesheets ((Base64-coded, optional)
		PreviewJson	Content of the sample data in JSON format (Base64-coded, optional)

Example for a call:

```
{
    "name": "templatename"
}
```

Output example:

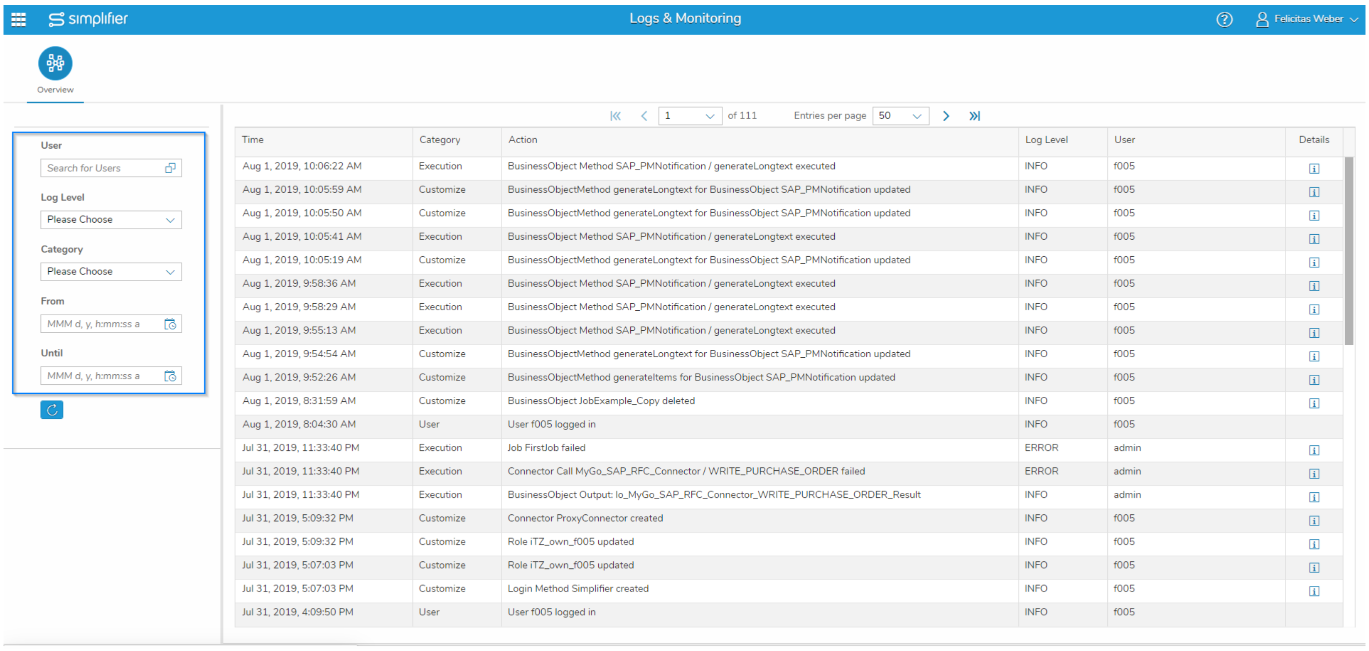
```
{
    "success": true,
    "value":  {
        "template": "SGFsbG8gV2VsdA==\",
        "stylesheet: "SGFsbG8gV2VsdA==\"
        "previewJson": "SGFsbG8gV2VsdA==\"
     }
}
```

Filter

https://developer.simplifier.io/documentation/logging-monitoring/filter/

The Logs & Monitoring tile uses all search features of the backend (i.e. pagination or filtering).

On the left-hand side, you can set filters.



User
Search for Users
Log Level
Please Choose
Category
Please Choose
From
MMM d, y, h:mm:ss a
Until
MMM d, y, h:mm:ss a

1 of 111 Entries per page 50

Time	Category	Action	Log Level	User	Details
Aug 1, 2019, 10:06:22 AM	Execution	BusinessObject Method SAP_PMNotification / generateLongtext executed	INFO	f005	ⓘ
Aug 1, 2019, 10:05:59 AM	Customize	BusinessObjectMethod generateLongtext for BusinessObject SAP_PMNotification updated	INFO	f005	ⓘ
Aug 1, 2019, 10:05:50 AM	Customize	BusinessObjectMethod generateLongtext for BusinessObject SAP_PMNotification updated	INFO	f005	ⓘ
Aug 1, 2019, 10:05:41 AM	Execution	BusinessObject Method SAP_PMNotification / generateLongtext executed	INFO	f005	ⓘ
Aug 1, 2019, 10:05:19 AM	Customize	BusinessObjectMethod generateLongtext for BusinessObject SAP_PMNotification updated	INFO	f005	ⓘ
Aug 1, 2019, 9:58:36 AM	Execution	BusinessObject Method SAP_PMNotification / generateLongtext executed	INFO	f005	ⓘ
Aug 1, 2019, 9:58:29 AM	Execution	BusinessObject Method SAP_PMNotification / generateLongtext executed	INFO	f005	ⓘ
Aug 1, 2019, 9:55:13 AM	Execution	BusinessObject Method SAP_PMNotification / generateLongtext executed	INFO	f005	ⓘ
Aug 1, 2019, 9:54:54 AM	Customize	BusinessObjectMethod generateLongtext for BusinessObject SAP_PMNotification updated	INFO	f005	ⓘ
Aug 1, 2019, 9:52:26 AM	Customize	BusinessObjectMethod generateItems for BusinessObject SAP_PMNotification updated	INFO	f005	ⓘ
Aug 1, 2019, 8:31:59 AM	Customize	BusinessObject JobExample_Copy deleted	INFO	f005	ⓘ
Aug 1, 2019, 8:04:30 AM	User	User f005 logged in	INFO	f005	
Jul 31, 2019, 11:33:40 PM	Execution	Job FirstJob failed	ERROR	admin	ⓘ
Jul 31, 2019, 11:33:40 PM	Execution	Connector Call MyGo_SAP_RFC_Connector / WRITE_PURCHASE_ORDER failed	ERROR	admin	ⓘ
Jul 31, 2019, 11:33:40 PM	Execution	BusinessObject Output: lo_MyGo_SAP_RFC_Connector_WRITE_PURCHASE_ORDER_Result	INFO	admin	ⓘ
Jul 31, 2019, 5:09:32 PM	Customize	Connector ProxyConnector created	INFO	f005	ⓘ
Jul 31, 2019, 5:09:32 PM	Customize	Role iTZ_own_f005 updated	INFO	f005	ⓘ
Jul 31, 2019, 5:07:03 PM	Customize	Role iTZ_own_f005 updated	INFO	f005	ⓘ
Jul 31, 2019, 5:07:03 PM	Customize	Login Method Simplifier created	INFO	f005	ⓘ
Jul 31, 2019, 4:09:50 PM	User	User f005 logged in	INFO	f005	

You can choose between the following filters.

Filter	**Function**
User	Filter for specific user actions
Log Level	

FQDN

https://developer.simplifier.io/documentation/glossar/fqdn/

A fully qualified domain name (FQDN) is sometimes also referred to as an absolute domain name.

Example on our Simplifier cloud:

Development	dev-yourcompany.simplifier.io
Quality Assurance	qa-yourcompany.simplifier.io
Productive	yourcompany.simplifier.io

Example for onpremise installation:

Development	dev-simplifier.yourcompany.com
Quality Assurance	qa-simplifier.yourcompany.com
Productive	simplifier.yourcompany.com

General Instructions

https://developer.simplifier.io/documentation/installation-instructions/general-instructions/

Here you will find general instructions about Simplifier deployment:

- Docker Installation
- Reverse Proxy Requirements
- Additional Requirements for Oracle Databases as Backend
- Docker Hub

General Requirements for On-Premise-Installations

https://developer.simplifier.io/documentation/installation-instructions/on-premise/general-requirements-premise-installations/

We support you with on-premise installations of Simplifier. To do that, we deliver a prepared Docker image to you. The image comes pre-configured and contains all the required components, including a Simplifier server in its most recent version.

The target instance must fulfill the following requirements:

- At least 12 GB RAM minimum, 16 GB recommended
- x64 CPU with minimum 2 cores, 4 cores recommended and at least 2 GHz per core
- At least 40 GB of free hard disk space
- Opened incoming ports: http/80 (TCP), https/443 (TCP), https/8090 (TCP)
- Optional: Opened outgoing ports for
  - preconfigured SMTP-Server (StartTLS Port 587) by Simplifier
  - your Backend Systems to configure and reach the Data Sources successfully
  - SSL Certificate for encrypted https traffic in frontend access
- Operating system:
  - Linux (recommended)
    - In general, the Docker engine can run on all Linux versions with kernel version >= 3.10, but for the versions below, there are "official" releases. If you are uncertain about the compatibility go to the Docker website.
      Tested Distributions:
      - Ubuntu: 64-Bit Versions of Ubuntu 18.04 (Bionic Beaver), 16.04 (Xenial) or 14.04 (Trusty)
      - CentOS 7.3: 64-Bit
      - Debian: Debian Stretch (Testing), Jessie (8.0), Wheezy (7.7, with Kernel-Update to Version 3.10)
      - Fedora: Versions 24 & 25
      - RHEL (Redhat Enterprise Linux) and SUSE Enterprise are officially supported only by paid docker variants (EE), Installations from CentOS Repository respectively OpenSUSE Repository are possible to use
  - Windows
    Install Docker for Windows

    Windows 10 Professional
    The runtime is given, but not as a Windows Service. The Docker Containers only stays up and runs in a locked user session.

    The Simplifier Windows Deployment is not recommend for production use, because of the limited support for container

  - Mac
    Install Docker for Mac
    Note: Our Docker containers, respectively the database server, require a file system which can be case-sensitive under MacOS. Therefore, it may be necessary to create a separate volume for the user data which is configured with the option "-v" when the container is started.

    The Simplifier MacOS Deployment is not recommend for production use, because of the limited support for container

"D-Q-P"-landscape

To ensure high availability and qualified operations, it is necessary to build a three-stage system landscape (= DQP-landscape: development, quality, production). Please note that with a DQP-landscape the system requirements are tripled.

Getting Started

https://developer.simplifier.io/documentation/getting-started/

Vimeo Video

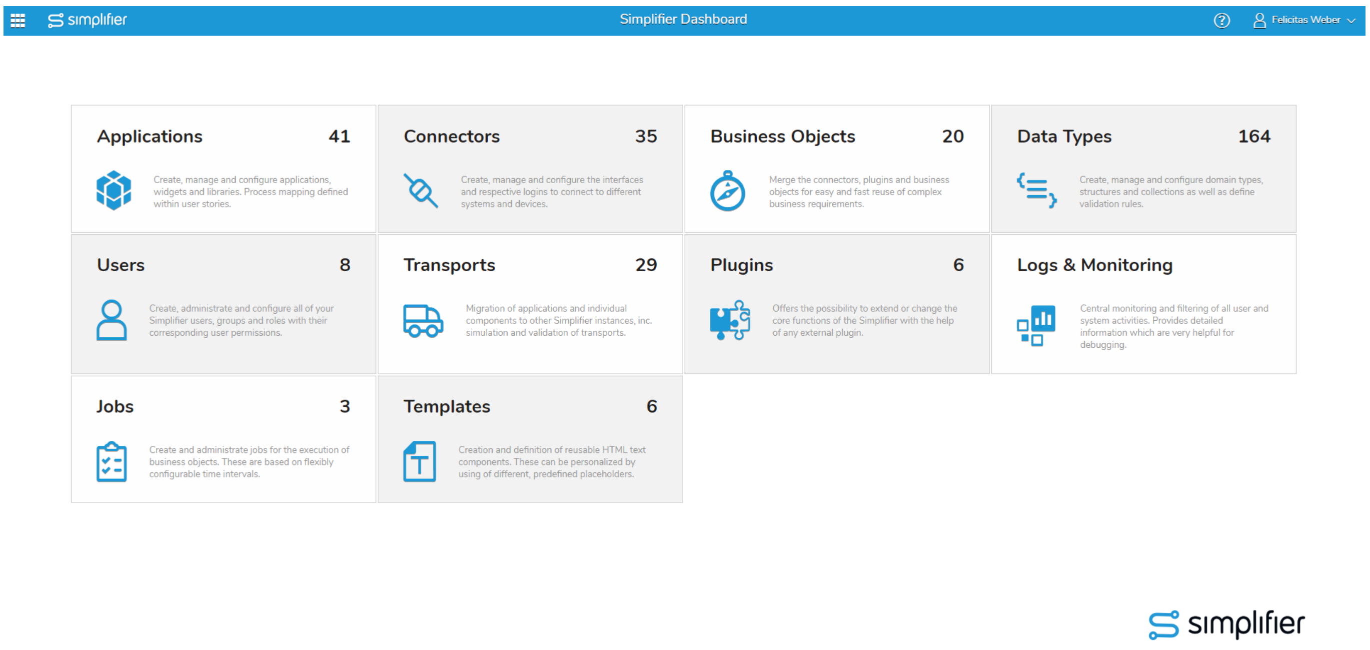
Simplifier is a low code platform for mapping business processes in integrated business and IoT applications and to interconnect internal and external IT infrastructures. Applications only need to be configured once to be available on any mobile device and operating system. Basically the functionality can be divided in two main categories:

- Application Creation, Operation and Maintenance
- abstract Integration Layer to connect external data sources

Main Features:

- Collaborative web-based Development Environment to configure integrated Mobile, Wearable and Browser applications
- Customization of Applications with UI Designer and Process Designer for visual Application Logic
- Customization of Backend Interfaces through standardized Connectors
- Rapid Deployment and Over-the-Air-Updates
- Contextual Technologies (Augmented Reality / Realtime Communication, Scanning, Device Sensors)
- Multi-Device (Browser, Smartphones, Tablets, Wearables)
- Multi Platform Mobile Client for Android and iOS

Using state-of-the-art technologies, we accelerate your application creation. We have designed and built our platform in terms of logic and usability to accommodate the modern, agile development processes within companies. Due to the low-code approach, applications no longer need to be elaborately programmed but instead can be easily configured and integrated into any system. Thus, applications can be mapped process-oriented.



Applications 41
Create, manage and configure applications, widgets and libraries. Process mapping defined within user stories.

Connectors 35
Create, manage and configure the interfaces and respective logins to connect to different systems and devices.

Business Objects 20
Merge the connectors, plugins and business objects for easy and fast reuse of complex business requirements.

Data Types 164
Create, manage and configure domain types, structures and collections as well as define validation rules.

Users 8
Create, administrate and configure all of your Simplifier users, groups and roles with their corresponding user permissions.

Transports 29
Migration of applications and individual components to other Simplifier instances, inc. simulation and validation of transports.

Plugins 6
Offers the possibility to extend or change the core functions of the Simplifier with the help of any external plugin.

Logs & Monitoring
Central monitoring and filtering of all user and system activities. Provides detailed information which are very helpful for debugging.

Jobs 3
Create and administrate jobs for the execution of business objects. These are based on flexibly configurable time intervals.

Templates 6
Creation and definition of reusable HTML text components. These can be personalized by using of different, predefined placeholders.

The main features of Simplifier can be accessed from the central Dashboard, that consists of the following parts:

- Applications
- Connectors
- Business Objects
- Data Types
- Users
- Transports
- Plugins
- Logs & Monitoring
- Jobs
- Templates

Basically, creating an application with Simplifier can be divided into the following 5 steps:

1. Connect systems

In the future, for each application, you can access the data that is needed contextually to make the integration process more efficient. Standardized connectors enable you to quickly connect to any back-end system and various data sources.

2. Create user interface

Easily, quickly and intuitively create the user-friendly interface for all your applications. Use the pre-designed elements designed for this purpose and create a uniform look and feel for improved user experience.

3. Configure processes

Configure the application logic of each application using the Process Designer. Based on individual user stories, reusable application logic is encapsulated within User Stories in the Process Dashboard. Each user story can be stored individually so that you can work with several people on different stories at the same time and thus be able to create application logic collaboratively.

4. Test application

Test your application at any time in the Simplifier Mobile client or in the browser. Intermediate testing allows for faster detection of misbehavior of your application at any point in time. The Simplifier Mobile Client supports the testing process by ensuring that applications can be used across devices and that the correct responsive display of the application on each end device can be ensured.

5. Publish application

The Simplifier transport system allows you to transfer the application to your productive system quickly and easily. And all without compilation or complex deployment processes. Quickly create a transport file of the finished application – it contains all the components of your applications and can be downloaded and imported directly into your production instance. Finished! The application is now available to any authorized user.

Glossar

https://developer.simplifier.io/documentation/glossar/

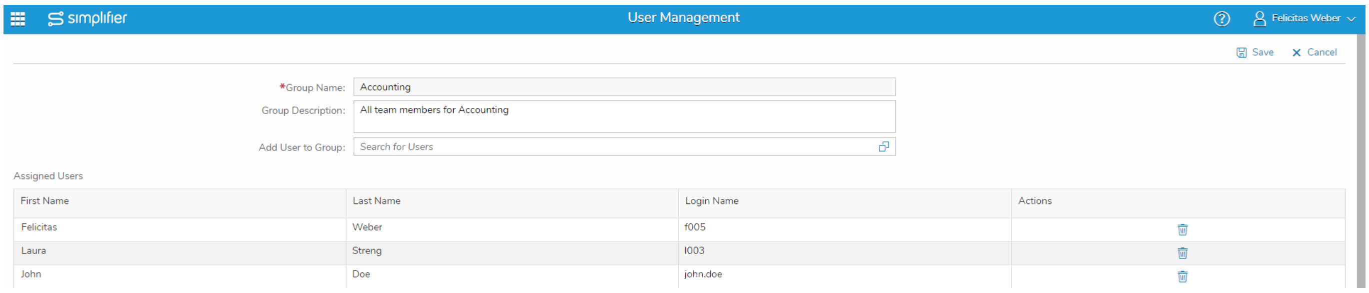
Here you will find general and Simplifier specific abbreviations, technical terms and their meaning.

Group Overview

https://developer.simplifier.io/documentation/user-management/group-overview/

A group contains several users and could be used for workflow logic in business apps like informing a team via email or push notification about a certain event or task.

Details View of a Group

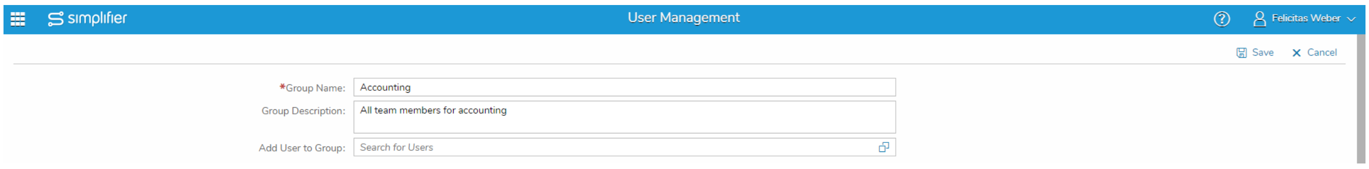


*Group Name:	Accounting
Group Description:	All team members for Accounting
Add User to Group:	Search for Users

Assigned Users

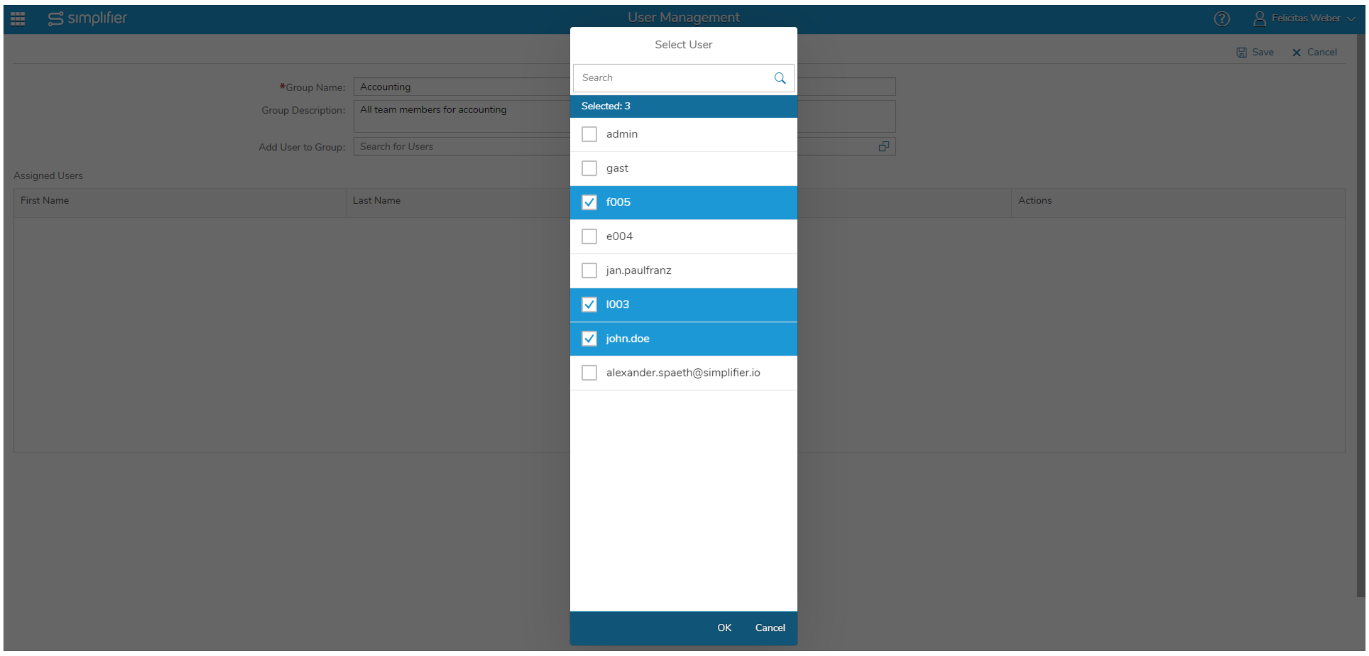
First Name	Last Name	Login Name	Actions
Felicitas	Weber	f005	🗑
Laura	Streng	l003	🗑
John	Doe	john.doe	🗑

To create a new group, you have to specify a **unique name** and an optional **description**, e.g. for a team or special task force group.



*Group Name: Accounting
Group Description: All team members for accounting
Add User to Group: Search for Users

To add users to the group, click into the 'Add User to Group' Field and search for specific usernames. Mark (optional) several users and click **OK** to add them to the group.



Accounting

All team members for accounting

Search for Users

Selected: 3

- [ ] admin
- [ ] gast
- [x] f005
- [ ] e004
- [ ] jan.paulfranz
- [x] l003
- [x] john.doe
- [ ] alexander.spaeth@simplifier.io

Save your changes.

Handling & Updating an On-Premise Installation

https://developer.simplifier.io/documentation/installation-instructions/on-premise/updating-premise-installation/

Docker Basic Commands

Start the Simplifier container:

```
$ docker start simplifier
```

Stop the Simplifier container:

```
$ docker stop simplifier
```

Restart the Simplifier container:

```
$ docker restart simplifier
```

Create a backup of the complete Simplifier data directory, e.g.

```
$ docker stop simplifier

$ tar cvzf simplifier_backup.tar.gz /home/simplifier
```

Updating

In case of updates, we will prepare a new docker image for you, preserving your personal settings. Please download the image to a temporary directory of your choice (e.g. /tmp) and change into the directory. Finally unpack and load it, as described in steps 3-4 in the installation instruction.

To perform the update, proceed as follows:

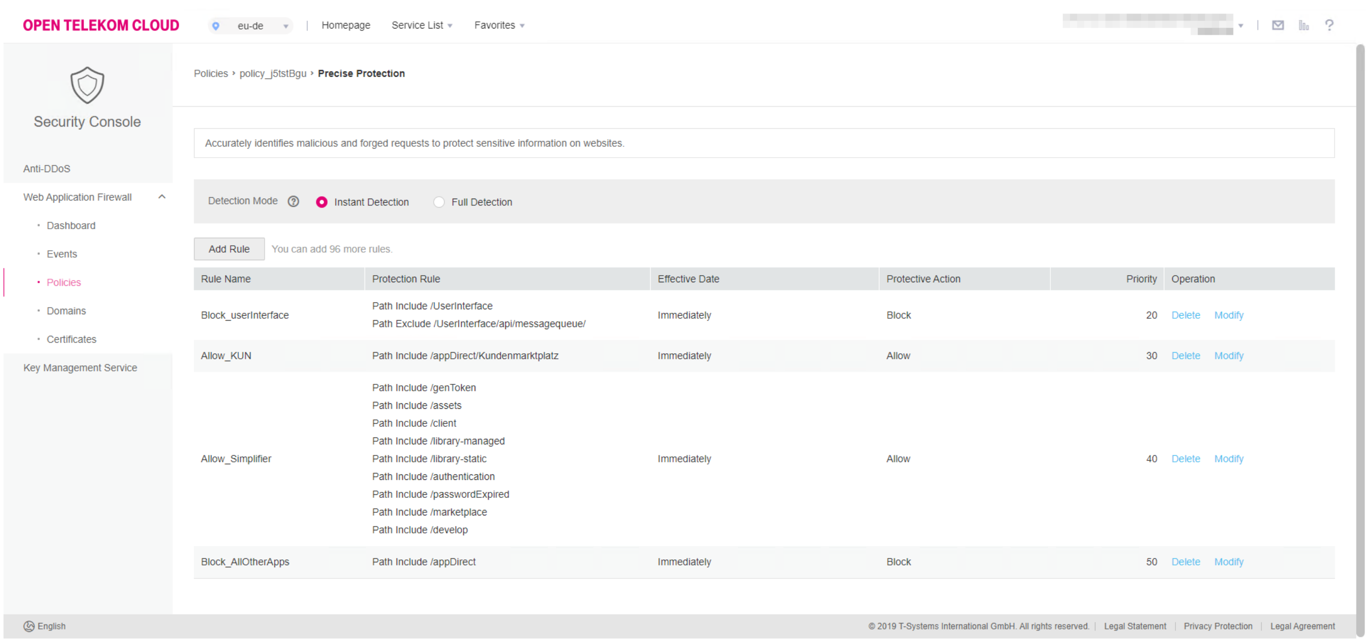
1. Stop the container and remove it from Docker. Take care NOT to remove your data directory /home/simplifier/data !
2. Perform the following commands in order:

```
$ docker stop simplifier

$ docker rm simplifier

$ docker run --name simplifier {additional options as in step 6 before}
```

Implementation of Web Application Firewalls

https://developer.simplifier.io/documentation/security-guidelines/implementation-of-web-application-firewalls/

As an example, a policy configuration of the OTC WAF for the use of the customer marketplace application "KUN" based on Simplifier



Policies › policy_j5tstBgu › **Precise Protection**

Accurately identifies malicious and forged requests to protect sensitive information on websites.

Detection Mode: [✓ Instant Detection] Full Detection

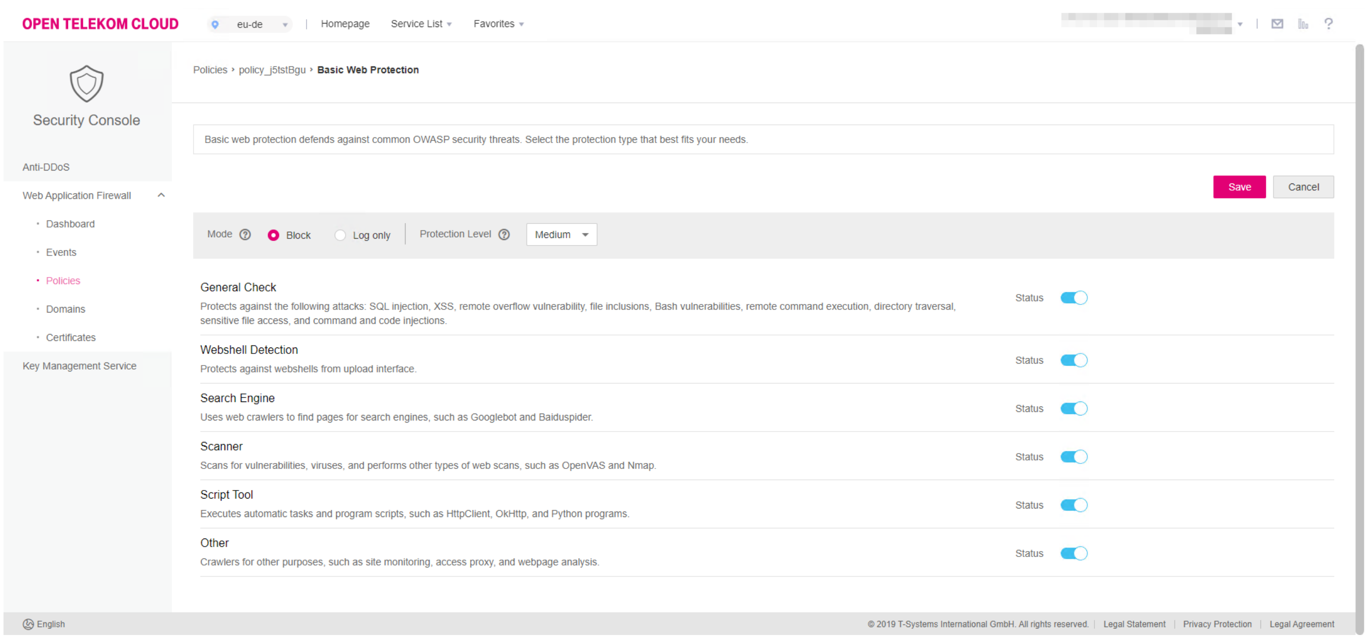
Add Rule — You can add 96 more rules.

Rule Name	Protection Rule	Effective Date	Protective Action	Priority	Operation
Block_userInterface	Path Include /UserInterface Path Exclude /UserInterface/api/messagequeue/	Immediately	Block	20	Delete Modify
Allow_KUN	Path Include /appDirect/Kundenmarktplatz	Immediately	Allow	30	Delete Modify
Allow_Simplifier	Path Include /genToken Path Include /assets Path Include /client Path Include /library-managed Path Include /library-static Path Include /authentication Path Include /passwordExpired Path Include /marketplace Path Include /develop	Immediately	Allow	40	Delete Modify
Block_AllOtherApps	Path Include /appDirect	Immediately	Block	50	Delete Modify

English
© 2019 T-Systems International GmbH. All rights reserved. | Legal Statement | Privacy Protection | Legal Agreement

Rule Name	Description
Block_userInterface	Blocks the user interface for external access
Allow_KUN	Allows dedicated access to the customer marketplace application
Allow_Simplifier	Allows basic functions
Block_AllOtherApps	Blocks all non-dedicated released applications

Activation of Basic Web Protection is recommended



Policies › policy_j5tstBgu › **Basic Web Protection**

Basic web protection defends against common OWASP security threats. Select the protection type that best fits your needs.

Mode: [✓ Block] Log only — Protection Level: Medium

General Check
Protects against the following attacks: SQL injection, XSS, remote overflow vulnerability, file inclusions, Bash vulnerabilities, remote command execution, directory traversal, sensitive file access, and command and code injections.
Status: on

Webshell Detection
Protects against webshells from upload interface.
Status: on

Search Engine
Uses web crawlers to find pages for search engines, such as Googlebot and Baiduspider.
Status: on

Scanner
Scans for vulnerabilities, viruses, and performs other types of web scans, such as OpenVAS and Nmap.
Status: on

Script Tool
Executes automatic tasks and program scripts, such as HttpClient, OkHttp, and Python programs.
Status: on

Other
Crawlers for other purposes, such as site monitoring, access proxy, and webpage analysis.
Status: on

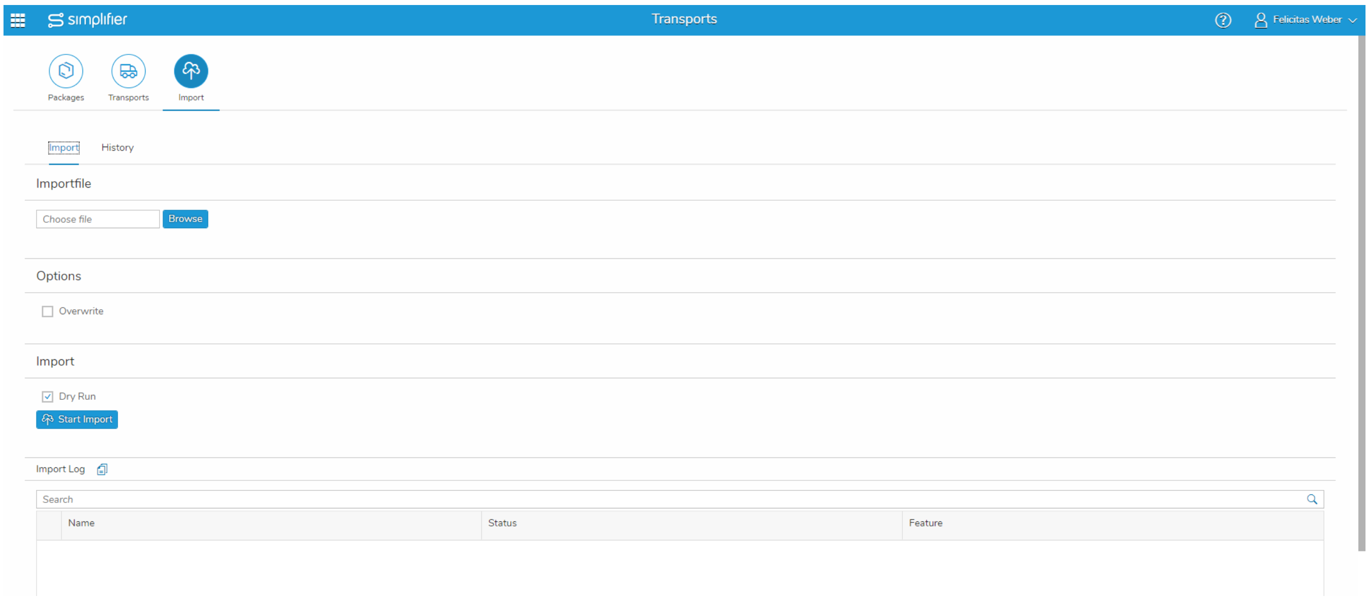
© 2019 T-Systems International GmbH. All rights reserved. | Legal Statement | Privacy Protection | Legal Agreement

Import Manual Transport

https://developer.simplifier.io/documentation/transports/import-manual-transport/

Vimeo Video

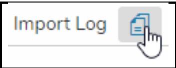
Switch to the tab Import to import a file to the Simplifier instance.



Importfile	Choose the file you want to import.

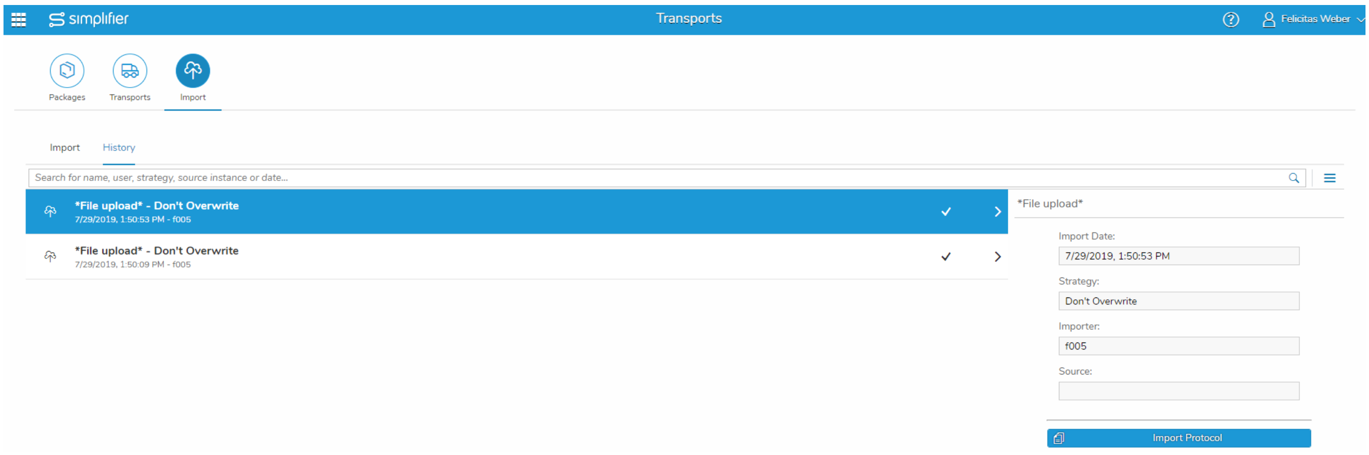
Options	If you select Overwrite, all features that already exist are overwritten with those from your transport file. Otherwise, only the new features are transported.
Import	If you select Dry Run, the content of the transport file is analyzed and a list of all features is displayed. It does not import the data so you can simply test how the transport would work out.

After selecting your transport file and setting it all up, click Start Import.

	The whole import log will be copied to clipboard.

History

Switch to the History tab to view an overview of all imported transports.



File upload - Don't Overwrite
7/29/2019, 1:50:53 PM - f005

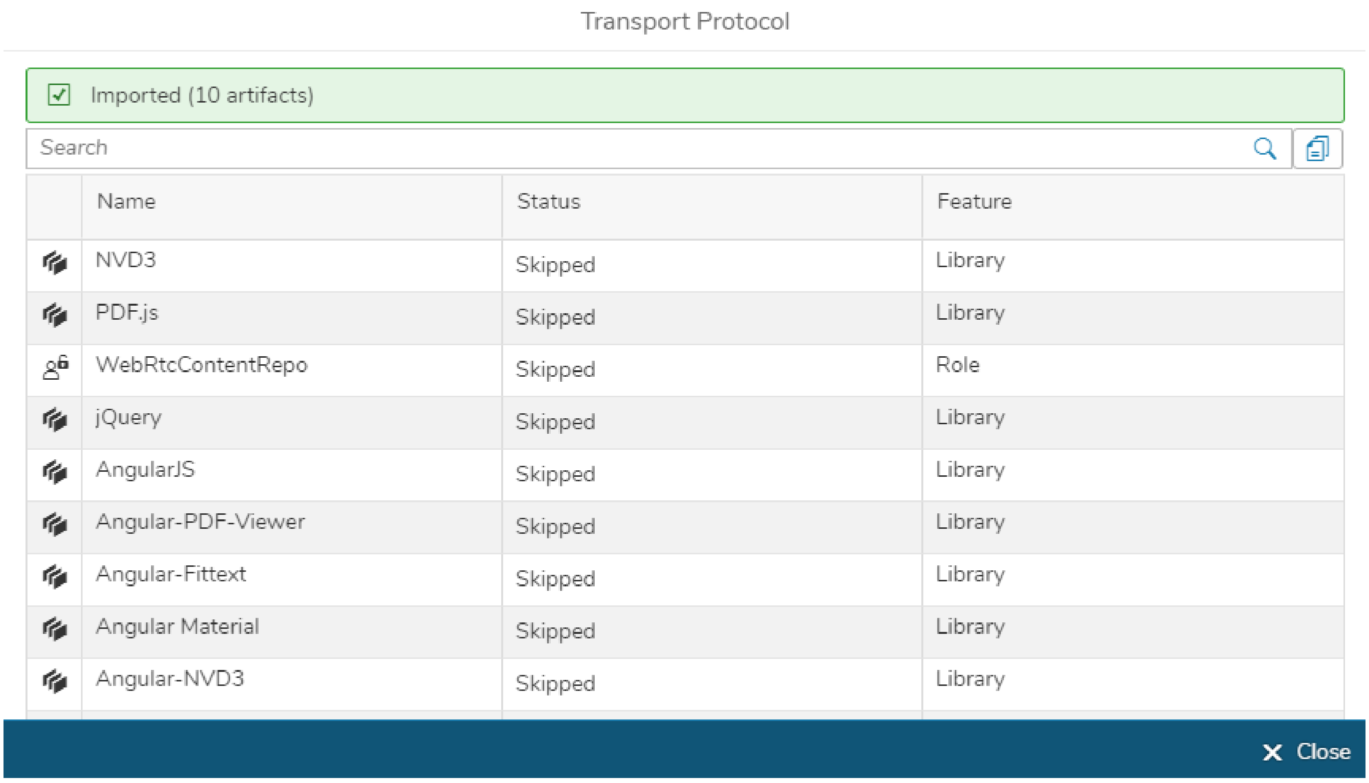
File upload - Don't Overwrite
7/29/2019, 1:50:09 PM - f005

File upload

Import Date: 7/29/2019, 1:50:53 PM
Strategy: Don't Overwrite
Importer: f005
Source:

On the right side, you get information about the Import Date, Strategy, Importer and Source.

If you click on Import Protocol, the list of all imported artifacts is displayed.


Transport Protocol

Imported (10 artifacts)

	Name	Status	Feature
	NVD3	Skipped	Library
	PDF.js	Skipped	Library
	WebRtcContentRepo	Skipped	Role
	jQuery	Skipped	Library
	AngularJS	Skipped	Library
	Angular-PDF-Viewer	Skipped	Library
	Angular-Fittext	Skipped	Library
	Angular Material	Skipped	Library
	Angular-NVD3	Skipped	Library

Installation PDF Plugin

https://developer.simplifier.io/documentation/plugins/pdf-plugin/installation-pdf-plugin/

Configuration

To use the pdf Plugin, you have to configure it first.
Copy the file "settings.conf.dist" from the directory "plugins/pdfPlugin/src/main/resources", save it as "settings.conf" and adjust it as follows:

In order to start the conversion, you need to install the program wkhtmltopdf on your operation system. The path to the wkhtmltopdf executable must be stated in the "settings.conf" file. Furthermore you need two folders, one to file your template and the other for the temporary data during the conversion. You can either use relative or absolute paths for the folders.

For example:

settings.conf

```
pdfPlugin {
     storageDir = "templates"
     tempDir = "tmp"
     wkhtmltopdf = "C:/Program Files/wkhtmltopdf/bin/wkhtmltopdf.exe"
}

....
```

NOTE:

If you use wkhtmltopdf on a Linux without the X11 Server, the error "**wkhtmltopdf: cannot connect to X server**" may occur.

In this case you need to install the program "xvfb" via the package manager to simulate the X11 server.

Create a wrapper (e.g. /usr/local/bin/wkhtmltopdf-xvfb) for the "wkhtmltopdf" program and write the path in the PdfPlugin Config.

wkhtmltopdf-xvfb

```
<#!/bin/bash>
xvfb-run --server-args="-screen 0, 1024x768x24"/usr/bin/wkhtmltopdf$*
```

Plugin Execution

The Plugin is located in the directory: plugins/pdfPlugins. It can be activated with the SBT/Activator via a "run" command. The STDIN command "stop" ends the Plugin execution.

You can adapt the logback-configuration file "plugins/pdfPlugin/src/main/resources/logback.xml" to configure the log output or display it in another file.

Installing an On-Premise Image

https://developer.simplifier.io/documentation/installation-instructions/on-premise/installing-premise-image/

We always prepare an all-in-one Docker image for our customers which contains all required components.

Given a target machine that matches the requirements described in the previous chapter, the installation is quite easy:

1. Create the directory which will host all external user-specific data:

```
$ mkdir -p /opt/simplifier/data

$ export SIMPLIFIER_DIR="/opt/simplifier/data"
```

2.1 If your server has an Internet connection, you can get the Docker image from Docker Hub.

```
$ docker pull simplifierag/onpremise:latest
# for the Onpremiseversion include MySQL and Nginx

$ docker pull simplifierag/netzportal:latest
# for the Netzportalversion with config files for your own Oracle DB
```

View the variants and their versions.

2.2 As an alternative we can provide a tarball for download. Copy the downloaded file with ending .tar.gz to a temporary directory on the target machine, e.g. /tmp and cd to this directory.

```
$ wget -O <filename>.tar.gz
```

Unpack the file in place:

```
$ tar xzvf <filename>.tar.gz
```

You will get two files: one readme.txt and the docker image with the ending .tar.

Inside the directory which contains the unpacked file, run the following commands as root- (super-) user:

```
$ docker load -i <imagefile.tar>
```

3. Install SSL certificates:

```
$ mkdir -p $SIMPLIFIER_DIR/certs

$ cp <certificate.crt> $SIMPLIFIER_DIR/certs/default.crt

$ cp <keyfile.key> $SIMPLIFIER_DIR/certs/default.key
```

4. Run docker image:

Alternative 1: with SSL/Certificates

```
$ docker run --name simplifier -v $SIMPLIFIER_DIR:/opt/simplifier/data \

-p 80:80 -p 443:443 -p 8090:8090 \

-d <Docker Tag>
```

Alternative 2: without SSL/Certificates

```
$ docker run --name simplifier -v $SIMPLIFIER_DIR:/opt/simplifier/data \

-p 80:8080 -p 8090:8091 \

-d <Docker Tag>
```

Replace the <Docker Tag> with the selected variant, e.g. simplifierag/onpremise:latest

5. Open your browser

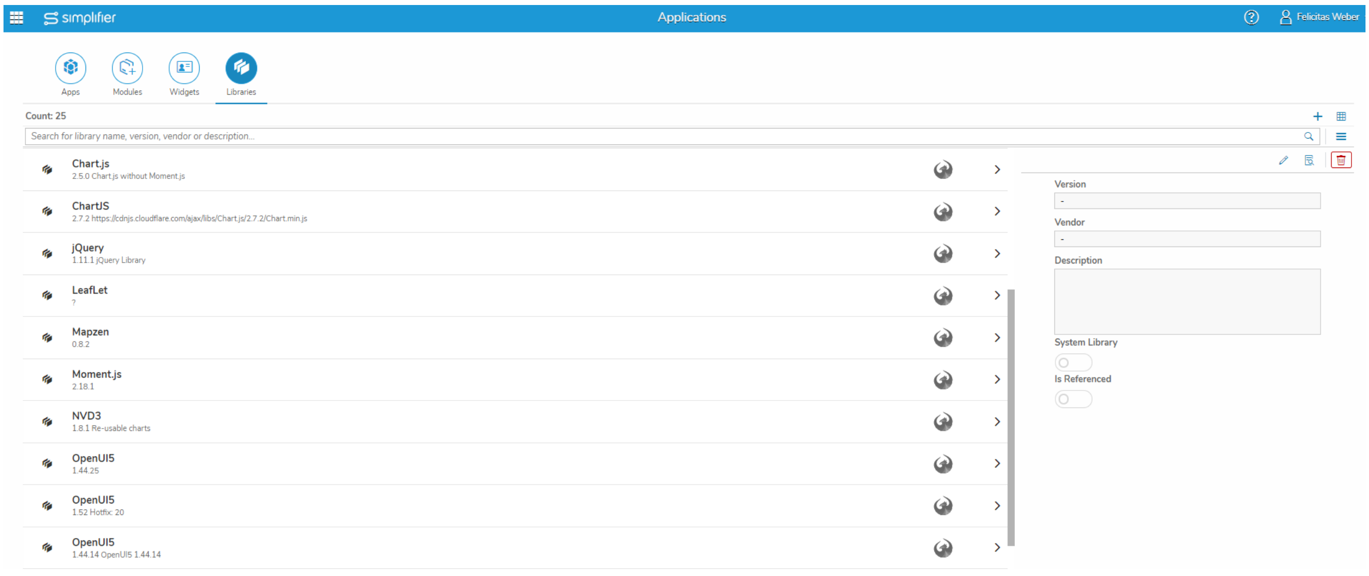
Now use your browser at your Client Computer to access http(s)://<IP> or <FQDN>/UserInterface. The Simplifier will prompt a license dialog. After pasting that license
you can start configuring Apps in the AdminUI.

Integration of external Libraries

https://developer.simplifier.io/documentation/applications/including-libraries/

Sometimes it is necessary to add an extra library to your app, e.g. if you want to display some special charts. You can upload and manage those external libraries under the "Libraries" tab in the Application tile.

If you want to know how to implement them into your application, go to "Libraries"



Count: 25

Search for library name, version, vendor or description...

Chart.js
2.5.0 Chart.js without Moment.js

ChartJS
2.7.2 https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.2/Chart.min.js

jQuery
1.11.1 jQuery Library

LeafLet
?

Mapzen
0.8.2

Moment.js
2.18.1

NVD3
1.8.1 Re-usable charts

OpenUI5
1.44.25

OpenUI5
1.52 Hotfix: 20

OpenUI5
1.44.14 OpenUI5 1.44.14

Version
-

Vendor
-

Description

System Library

Is Referenced

Standard Equipment

Simplifier provides the following library by default:

App Technology	Library	Version
UI5	OpenUI5	1.60

Integration of Libraries - addAfterInitHandler

https://developer.simplifier.io/documentation/applications/including-libraries/add-new-library/integration-libraries-addafterinithandler/

addAfterInitHandler

Parameter	**Type**	**Description**
Handler	Function	Callback function, which is called after all scripts have been loaded completely.

Integration of Libraries - addBeforeInitHandler

https://developer.simplifier.io/documentation/applications/including-libraries/add-new-library/integration-libraries-addbeforeinithandler/

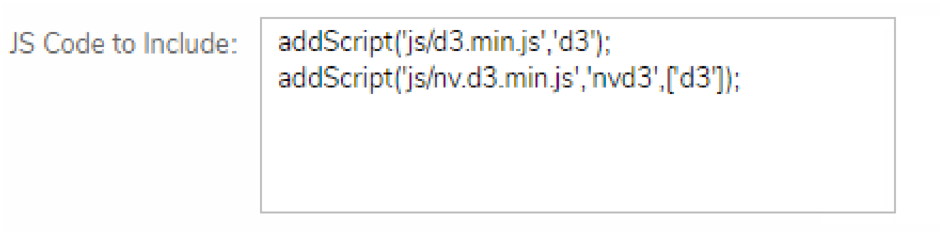
addBeforeInitHandler

Parameter	**Type**	**Description**
Handler	Function	Callback function, which is called immediately before the loading of the script begins.

Integration of Libraries - addScript

https://developer.simplifier.io/documentation/applications/including-libraries/add-new-library/integration-libraries-addscript/

To integrate the library with a js code snippet, use the following parameter:


JS Code to Include:

```
addScript('js/d3.min.js','d3');
addScript('js/nv.d3.min.js','nvd3',['d3']);
```

addScript(ScriptPath, Name, Dependencies)

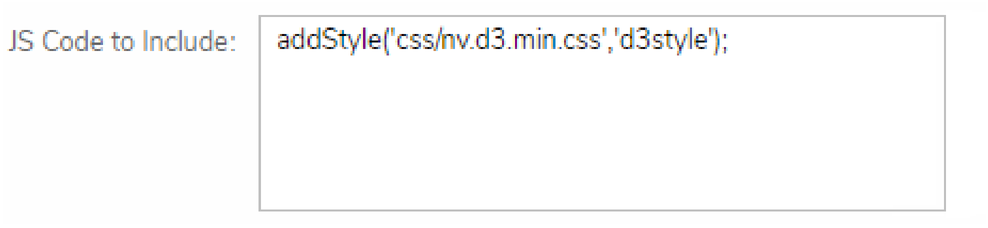
Parameter	Type	Description
ScriptPath	String	Relative path in the uploaded ZIP structure to the .js file you want to include (e.g. src/js/includedScript.js)
Name	String	Name of the library you can use to access the .js file (e.g. includedScript). By using "includedScript" in your script code you can now use all methods of your integrated library
Dependencies	Array<String>	Dependent scripts (refers to the parameter "name" of "addScript") It guarantees, that all dependencies are loaded beforehand. Use this if your library needs other libraries to work properly

It is important to ensure that all scripts specified under "dependencies" are either integrated into the same library, or a dependency is set on the library in which the script is integrated.

Integration of Libraries - addStyle

https://developer.simplifier.io/documentation/applications/including-libraries/add-new-library/integration-libraries-addstyle/

To integrate the library with a js code snippet, use the following parameter:


JS Code to Include:

```
addStyle('css/nv.d3.min.css','d3style');
```

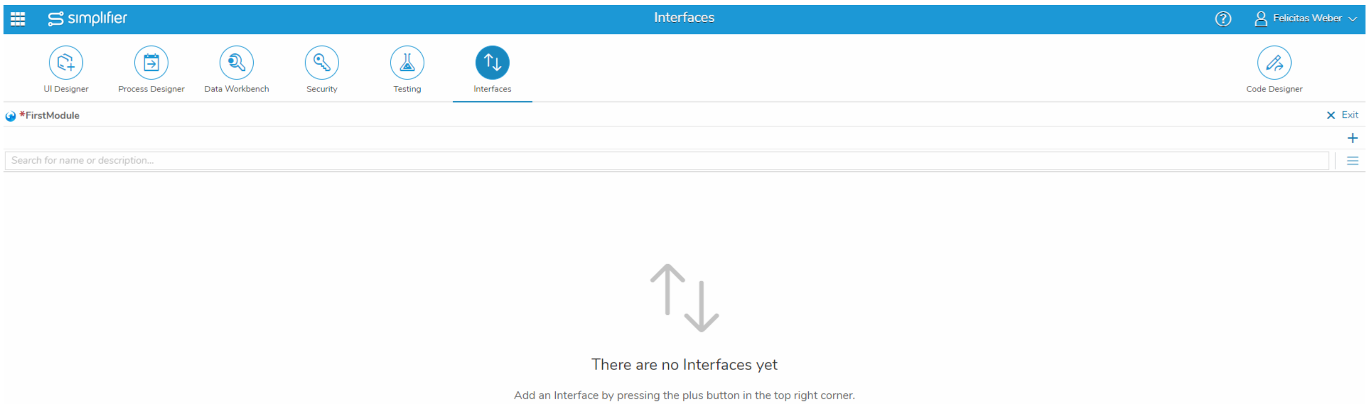
addSyle

Parameter	**Type**	**Description**
StyleURL	String	Relative path to the uploaded ZIP structure of the library
Name	String	Style name (optional)

Interfaces

https://developer.simplifier.io/documentation/applications/modules/interfaces/

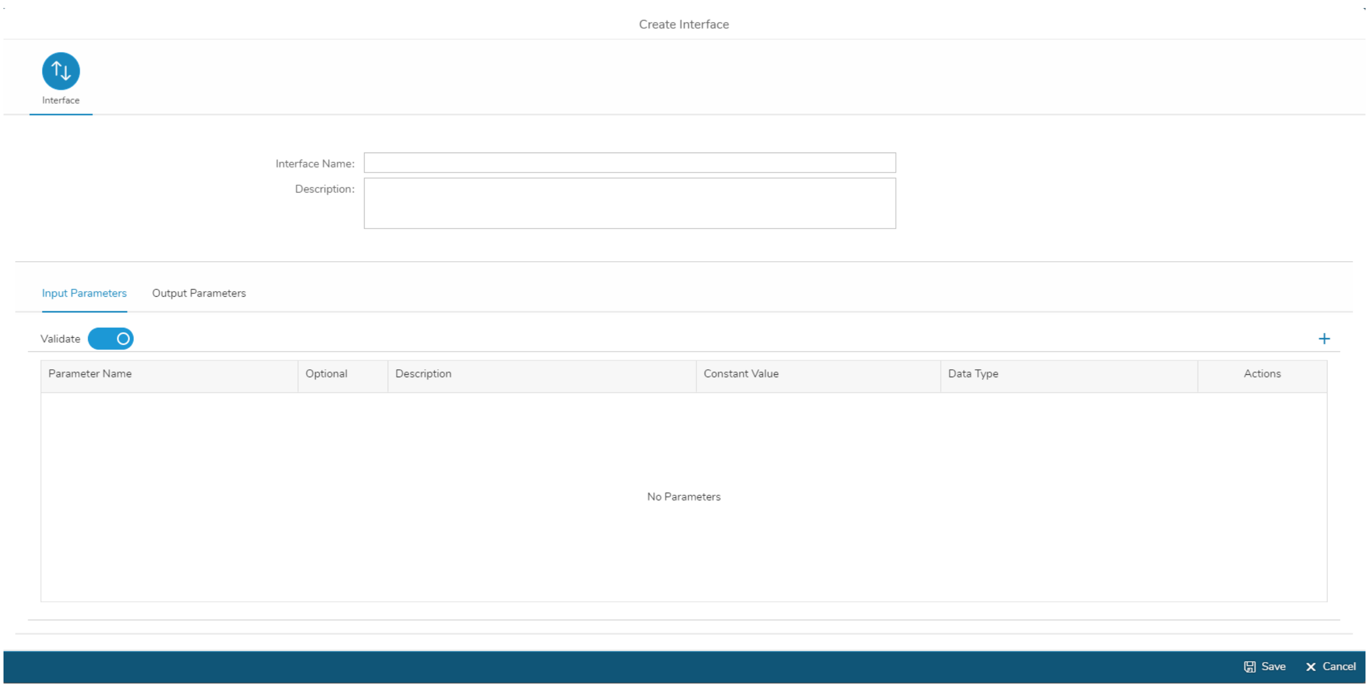
Interfaces are used for communication between the application and the modules to exchange data bidirectionally.



There are no Interfaces yet

Add an Interface by pressing the plus button in the top right corner.

When creating a new interface via the plus button on the top right, the following pop-up appears:

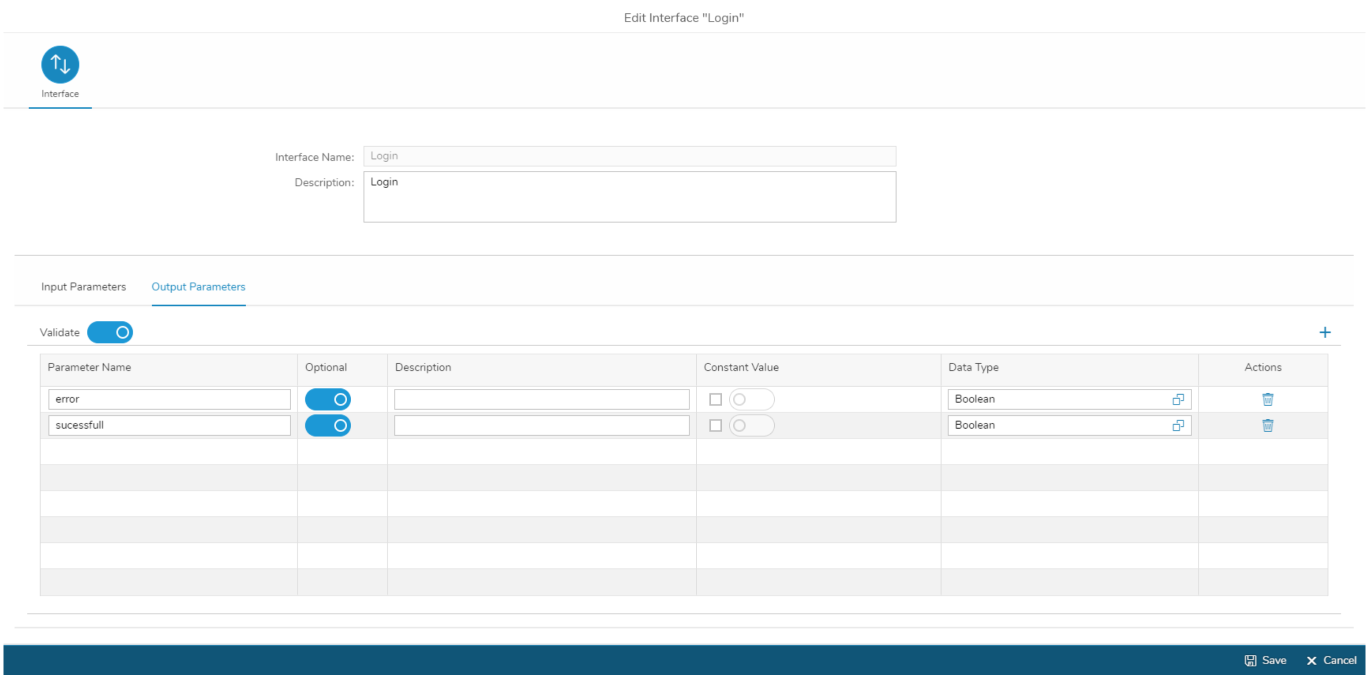


Interface Name:	
Description:	

Parameter Name	Optional	Description	Constant Value	Data Type	Actions

No Parameters

An interface of a module is defined by its unique name and a set of parameters, where **Input Parameters** are passed from the application to a module and **Output Parameters** are sent from a module to an application.

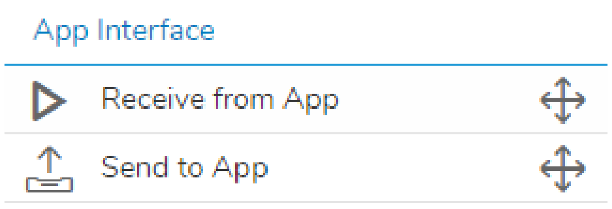


Interface Name: Login
Description: Login

Parameter Name	Optional	Description	Constant Value	Data Type	Actions
error	●		☐	Boolean	
sucessfull	●		☐	Boolean	

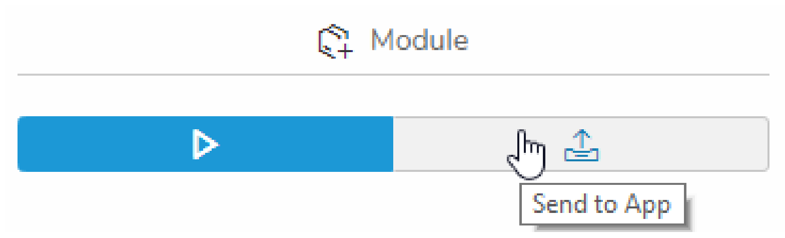
Use Interfaces in Process Designer

The Process Designer of the modules is similar to the Process Designer of the applications. However, among the activities, there is an explicit point for modules: **App Interface**.



Receive from App	This shape starts an action when the application is calling the module via an interface. Double click on it or open the selection helper on the right side to configure the shape by selecting an interface of the current module and the mapping of parameters, that are received from the app.
Send to App	This shape is used to return parameter data and/or trigger an action in the controlling app. Double click on it or open the selection helper on the right side to configure the shape by selecting an interface of the current module and the mapping of parameters to send it to the application.

You also have the option to switch between these two activities.



Simplifier Developer

Documentation & Community

PDF generated January 28, 2020 at 7:39 AM