



Developer Guide | PUBLIC  
Document Version: 2402

# Business Process Extensions – Developer's Guide

# Content

- 1 Business Process Extensions. . . . . 3**
- 2 Architectural Overview. . . . . 4**
- 3 SAP Business Technology Platform Extensions. . . . . 5**
  - 3.1 Serverless Functions in the SAP BTP, Kyma Runtime. . . . . 5
  - 3.2 Customer-Built Applications in SAP BTP. . . . . 5
- 4 Registering Services in the SaaS Tenant of SAP Digital Manufacturing . . . . . 6**
  - 4.1 API Services. . . . . 7
    - Create an API Service (RESTful and OData). . . . . 8
    - Import API Services (SOAP). . . . . 13
    - Copy, Move, and Delete an API Service. . . . . 14
    - Import MII Transactions. . . . . 15
  - 4.2 Schemas. . . . . 16
    - Create a Schema. . . . . 17
    - Copy, Move, and Delete a Schema. . . . . 22
  - 4.3 UI Extensions. . . . . 23

# 1 Business Process Extensions

A business process extension service is a reusable function which can assist in data formatting, data enrichment, or other use cases for production process design. It can be written in either the SAP BTP, Kyma Runtime, or customer-built applications in SAP BTP. Furthermore, it can also be written in a business application within a customer landscape exposed as an API and registered in SAP Digital Manufacturing.

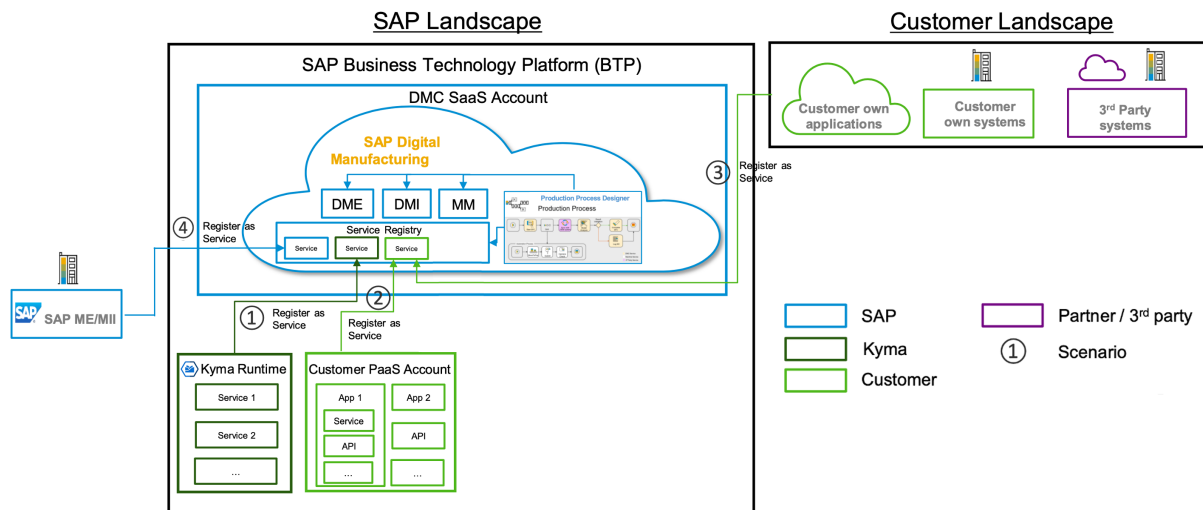
This guide provides you with a deeper understanding of the respective extensibility patterns.

## 2 Architectural Overview

The architecture in the following diagram shows different components from SAP and the customer landscape coming together to create distinct patterns for building extensions.

The following patterns are possible:

1. Extend with serverless functions developed in the SAP BTP, Kyma Runtime
2. Extend with services from customer-built applications in SAP Business Technology Platform
3. Extend with services from customer's own application (cloud or on-premise) out of SAP Business Technology Platform
4. Extend with services from other SAP systems, e.g. SAP ME/MII, etc.



Apart from the above user-defined services, you can also use other API services provided by SAP in the service registry.

# 3 SAP Business Technology Platform Extensions

The scope of this chapter is limited to the following two patterns:

1. Serverless functions in the SAP BTP, Kyma Runtime
2. Customer-built applications in SAP Business Technology Platform

## 3.1 Serverless Functions in the SAP BTP, Kyma Runtime

Kyma is a platform to develop and deploy applications with serverless functions and microservices. You can use SAP BTP, Kyma Runtime to extend the Next Number generation functionality of the SAP Digital Manufacturing for execution by implementing in-app extensions.

For details, see the [Installation and Configuration Guide](#) .

## 3.2 Customer-Built Applications in SAP BTP

You can use your PaaS account (Cloud Foundry environment) to create applications with SAP Business Application Studio. SAP Business Application Studio is a service within SAP Business Technology Platform that provides a development environment for SAP Cloud Foundry.

For details, see the [Developer Guide](#) .

# 4 Registering Services in the SaaS Tenant of SAP Digital Manufacturing

After APIs have been built, you can register them in SAP Digital Manufacturing using the [Manage Service Registry](#) app.

To view [Manage Service Registry](#) app on the SAP Fiori launchpad, and create, update, and delete one or all service entries and schemas, you must have the role of `Manufacturing Extension Provider` or `Production Engineer`. To read one or all service entries and schemas, you must have the role of `Manufacturing Extension Provider`, `Production Operator`, `Production Supervisor`, `Automation Engineer`, or `Production Engineer`.

The [Manage Service Registry](#) app supports the following service types to be registered and they are displayed in groups and subgroups:

Type / Group	Description
DMC	<p>API services provided by SAP Digital Manufacturing. Automatically available in service registry. Cannot edit, copy, move, or delete.</p> <p>SAP Digital Manufacturing public API packages, as well as some internal services. For details, see <a href="#">SAP Business Accelerator Hub</a>.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>The actual services available are more than those listed under the public API packages in SAP Business Accelerator Hub. There are some internal services (marked in the header section) as well. You can check all services under DMC group in the <a href="#">Manage Service Registry</a> app.</p> </div> <p>You can see the software product and product version of SAP Digital Manufacturing services in the header section. Use the combo box next to the service name to switch between different versions.</p>
Production Process	<p>Production processes published from production process designer to service registry. Cannot edit, copy, move, or delete from service registry side but the process is removed from service list once you unpublish or delete it in production process designer. You can see the name, version, and description of the related production process design and production process in the source section. Jump to the target directly with the hyperlink.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>If you want to invoke the APIs associated with these production processes, refer to the host URL for the public APIs in the SAP Business Accelerator Hub.</p> </div>
Indicator Service	Services for reading or writing values from indicators on asset/equipment.
User-Defined	Any services that you manually register with SAP Digital Manufacturing. You can create your own subgroups for these services.


Type / Group	Description
MII	User-defined transactions in SAP Manufacturing Integration and Intelligence (MII). You can edit service name when importing the first time and reimport to update. The transaction display follows the same structure as you defined in MII.

You can check the service statuses on both the list and detail page:

- **Changed (compatible):** There are compatible changes in the service in comparison with the previous release.
- **Changed (incompatible):** There are incompatible changes in the service in comparison with the previous release.
- **Deprecated:** The service is to be deprecated in future releases.

### Note

Compatible and incompatible changes are different. For example, changing the name and description, adding optional parameters, and response code, are compatible service changes, while changing the API path, adding mandatory parameters, and changing the HTTP method, web server, or response code, are incompatible service changes. Making changes to a service may affect its usage in its consumer. A compatible change won't cause runtime errors while an incompatible one will do. You can also see if there is any usage status change due to service updates in its where-used list. See [Create an API Service \(RESTful and OData\) \[page 8\]](#) for details.

Choose  (*Refresh*) to load the latest service list.

All schemas of services are centrally managed under the [Schemas](#) tab. For details, see [Schemas \[page 16\]](#).

Besides the services above, the [Manage Service Registry](#) app also supports registering extensions that can be plugged into existing UIs, for example, POD plugins. You can see these UI extensions under the [UI Extensions](#) tab. For details, see [UI Extensions \[page 23\]](#).

## Related Information

[Manage Service Registry](#)

## 4.1 API Services

You can manually register or import API services in the service registry.

You must have the role of `Manufacturing Extension Provider` or `Production Engineer` to perform the action.

## 4.1.1 Create an API Service (RESTful and OData)

You can manually register REST and OData APIs as services.

### Procedure

1. Go to the *Manage Service Registry* app and under the *API Services* tab, choose **Create > New Service**.
2. Fill in relevant fields.

Section	Field	Description
Header	Name	This name will be displayed in service library of the production process designer later when you select services.
	ID	The unique identifier of a service. The ID can only contain A–Z, a–z, 0–9 and underscores. It must start with a letter and end either with a letter or a number. Double underscore is not allowed. Do not use “dmc” as a prefix for API type services. The ID is automatically generated as you type the name. You can of course also change the ID independently in this field. The ID cannot be changed once a service is created.
	Description	Optional. The description of the service.
	Group Path	You can either select an existing subgroup under "User-Defined" group or manually enter a new subgroup name to create one. You can also change the subgroup after a user-defined service is created in service detail page.  Alternatively, choose <i>Manage Groups</i> on the service list page to edit "User-Defined" subgroups.  You can delete a "User-Defined" subgroup if there is no service in it.
Overview	Availability	<ul style="list-style-type: none"><li>• <b>Yes</b>: Allows the service to be used.</li><li>• <b>No</b>: Prevents the service from being used.</li></ul>
	Type	Support HTTP only.
	Protocol	Support RESTful and limited support for OData. Some parameters are not supported in OData, such as \$batch.
	Method	API Method. Single select from: GET, POST, PUT, PATCH, DELETE
	Web Server	Assign a web server you created to the service. For details on how to create a web server, see <i>Manage Web Servers</i> .



Section	Field	Description
	URL / Path	<p>Associate the service with web server first if you want to use it in production process designer. Then enter a relative path in URL / Path field. To override the host/virtual URL specified in the web server, enter an absolute path.</p> <p>Web server association is not required if the service is to be used in the <i>Manage Next Numbers</i> app. In this case, enter an absolute path.</p> <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p><b>Note</b></p> <p>The system uses the URL with this order of priority when calling cloud services:</p> <ol style="list-style-type: none"> <li>1. Absolute path set in the <i>Manage Service Registry</i> app</li> <li>2. Destination added for web server when no absolute path is set in the <i>Manage Service Registry</i> app</li> <li>3. Host URL set in the web server along with the relative path set in the <i>Manage Service Registry</i> app, if no destination is added for web server</li> </ol> <p>The system uses the URL with this order of priority when calling on-premise services:</p> <ol style="list-style-type: none"> <li>1. Absolute path set in the <i>Manage Service Registry</i> app</li> <li>2. Virtual URL in destination added for web server when no absolute path is set in the <i>Manage Service Registry</i> app</li> <li>3. Virtual URL set in the web server along with the relative path set in the <i>Manage Service Registry</i> app, if no destination is added for web server</li> </ol> <p>See <a href="#">Manage Web Servers</a> for details.</p> </div>
	Is Extension	Check to use the service in the <i>Manage Next Numbers</i> app. For details, see <a href="#">Manage Next Numbers</a> .

3. Service registry supports two editing modes: form view and code view. Switch between modes by choosing *Switch to Code* or *Switch to Form*.

## Form View

Fill in relevant fields of the *Service Definition* section.

Section	Field	Description
Parameters	Name	Required.
	Description	Optional. The description of the parameter.
	Location	<p>There are three types: <code>Path</code>, <code>Query</code>, <code>Header</code>. The path parameter is automatically generated based on the path or URL field.</p> <p>You cannot edit the path parameter name or data type but you can edit its other values and enter a description.</p>

Section	Field	Description
	Data Type	Support <code>String</code> , <code>DateTime</code> , <code>Byte</code> , <code>Boolean</code> , <code>Integer</code> , <code>Short</code> , <code>Long</code> , <code>Number</code> , <code>Double</code> , <code>Float</code> and arrays of these primitive data types.  <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>Note</b> Array type of query parameter only supports format like <code>name=a , b , c</code>.</p> </div>
	Enum	List of values that the parameter accepts. Press <i>Enter</i> to add each value.
	Item Enum	List of values that items of array parameter accept. Press <i>Enter</i> to add each value.
	Default Value	If any enum values are defined, default value must be one of them. You can also add a default value without defining enum.
	Required	Check the box if the parameter is mandatory.
<b>Request Body</b>	Required	Choose <i>Yes</i> if the request body is mandatory.
	Description	Description of the request body.
	Content Type	You can either input freely or select one or more content types in the following combinations: 1. <code>application/json</code> 2. <code>text/plain</code>  <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>Note</b> Timers and business service type rules in the <i>Manage Automatic Triggers</i> app only support services with <code>application/json</code> as content type.</p> </div>
	Data Type	Support <code>String</code> , <code>DateTime</code> , <code>Byte</code> , <code>Boolean</code> , <code>Integer</code> , <code>Short</code> , <code>Long</code> , <code>Number</code> , <code>Double</code> , <code>Float</code> , <code>Structure</code> and arrays of these data types.
	Schema	If the data type is <code>Structure</code> or <code>StructureArray</code> , you can define a schema for the structure. You can define the schema inline as part of the service definition or reference a schema from the <i>Schemas</i> tab. The schemas are described in JSON objects.  <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>Note</b> The system generates schema automatically under the <i>Schemas</i> tab from inline schema of a new service.  Inline schema for structure array is for the schemas of the items (object) of the array. You are recommended to create the schema under the <i>Schemas</i> tab first and then reference it if the schema is complex. Array-of-array structure, <code>[ [ 1 , 2 , 3 ] , [ 4 , 5 , 6 ] ]</code>, for example, is not supported.</p> </div>
<b>Response</b>	Response Code	HTTP response status codes indicate whether a specific HTTP request has been successfully completed. List HTTP status codes clearly like "200", "401". Codes like "2XX" are not accepted; <b>default</b> is allowed.
	Description	Required. Description of response code.

Section	Field	Description
	Content Type	<p>You can either input freely or select one or more content types in the following combinations: 1. application/json (default) 2. text/plain</p> <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p><b>Note</b></p> <p>Timers and business service type rules in the <a href="#">Manage Automatic Triggers</a> app only support services with application/json as content type.</p> <p>If text/plain is defined as content type in response, only string data type is allowed in the Production Process Designer.</p> </div>
	Data Type	Support String, DateTime, Byte, Boolean, Integer, Short, Long, Number, Double, Float, Structure and arrays of these data types.
	Schema	<p>If the data type is Structure or StructureArray, you can define a schema for the structure. You can define the schema inline as part of the service definition or reference a schema from the <a href="#">Schemas</a> tab. The schemas are described in JSON objects.</p> <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p><b>Note</b></p> <p>The system generates schema automatically under the <a href="#">Schemas</a> tab from inline schema of a new service.</p> <p>Inline schema for structure array is for the schemas of the items (object) of the array. You are recommended to create the schema under the <a href="#">Schemas</a> tab first and then reference it if the schema is complex. Array-of-array structure, [ [ 1 , 2 , 3 ] , [ 4 , 5 , 6 ] ], for example, is not supported.</p> </div>

## Code View

Edit the service definition in the text field. Refer to [OpenAPI 3.0](#) for the definition of parameters and responses. You need to create the service JSON following the structure in the sample below:

### Sample Code

```
{
  "timeout": 90,
  "parameters": [
    {
      "name": "petId",
      "in": "path",
      "description": "Pet ID",
      "required": true,
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "country",
      "in": "query",
      "description": "Country of origin",
      "required": false,
      "schema": {
        "type": "string"
      }
    }
  ]
}
```

```

    ],
    "requestBody": {
      "description": "Pet information update request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "/User Defined/petUpdate"
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "Message successfully processed.",
        "content": {
          "application/json": {
            "schema": {
              "type": "array",
              "items": {
                "$ref": "/User Defined/pet"
              }
            }
          }
        }
      }
    }
  }
}

```

#### 📘 Note

The path parameter defined in the [Overview](#) section should also be defined in the service definition. If you have declared a path parameter in code view of service definition, you will still need to add the path parameter in *Path* field of the [Overview](#) section. The path parameter needs to be defined in curly brackets, for example, `/pets/{petId}`.

All API services previously following OpenAPI 2.0 standards are automatically converted to OpenAPI 3.0.

#### 4. Choose *Create*.

#### 📘 Note

Choose *Edit* to edit existing services under the "User-Defined" group. If the service has been used in other places, the system will detect it automatically and you can check in the warning message box where this service is used once you save your edit. You can continue with your update or cancel. Alternatively, you can also choose *Check Where-Used* in the service detail page to check where the service has been used. Refresh the list to see the latest changes if the service gets updated. Besides the general information about the consumers provided in the where-used list, the usage status indicates if the service is used in an up-to-date manner by its consumers. You should pay attention and take certain actions to those services with "outdated" usage status. Possible usage status:

- **Normal:** The service is being used normally by the consumer.
- **Normal (deprecated):** The service is being used normally by the consumer, but you need to check further to see if any action is needed as the service is to be deprecated in the future.
- **Outdated (compatible):** The service is outdated in the consumer due to compatible changes. Recommend checking further.
- **Outdated (incompatible):** The service is outdated in the consumer due to incompatible changes. You need to synchronize the service to keep using it.

- **Unavailable:** The service is not available to the consumer due to one of the following reasons:
  - the service has been disabled
  - the service web server has been removed or changed
  - the web server connection has been deleted or changed
  - production process published to the service registry has been unpublished
- **Unsupported:** The service is no longer supported in the consumer.

Navigate directly to each consumer through its link and check the service accordingly.

For how to use these API services as part of the production process in the Production Process Designer after registration, see [Create and Define a Production Process](#).

## 4.1.2 Import API Services (SOAP)

You can register SOAP APIs as services.

### Procedure

1. Go to the [Manage Service Registry](#) app and under the [API Services](#) tab, choose **Import** **SOAP Service**.
2. In the [Select SOAP Service](#) window, assign a web server you created to the service. The system gets the URL from the web server you added. For details on how to create a web server, see [Manage Web Servers](#).
3. In the [Get WSDL from](#) field, choose [URL](#) or [File](#).
4. If you choose [URL](#), enter a relative path in the [Path](#) field, choose a destination you added in the [Manage Web Servers](#) app or leave it blank. The destination is just for authentication purposes. The system uses the host URL set in the web server along with the relative path set here in the [Manage Service Registry](#) app to call the service. If you choose [File](#), upload your own WSDL file for the system to fetch the related information.

#### Note

The policy configuration and reference details in the WSDL file can cause errors in runtime if you assign an on-premise web server to the SOAP service. In this case, make sure to remove all relevant parts before uploading, for example:

#### Sample Code

```
<wsp:Policy>
  <sp:HttpsToken>
    <wsp:Policy>
      <sp:HttpBasicAuthentication/>
    </wsp:Policy>
  </sp:HttpsToken>
</wsp:Policy>
<wsp:PolicyReference URI="#SoapConnectorLibrary.SoapEndpoint_policy" />
```

5. Choose *Go*.
6. Choose a *Target Service Group* from existing service groups or input a name to create a new service group.
7. Select SOAP services you want to register from the list. Change the service name and ID if needed.
8. Choose *Add*.
9. You can view and edit the SOAP service you have registered on the service detail page. For details, see [Create an API Service \(RESTful and OData\) \[page 8\]](#).

#### Note

For how to use these API services as part of production process in Production Process Designer after registration, see [Create and Define a Production Process](#).

For restrictions on SOAP service, see the related section in [3379404](#) for details.

## 4.1.3 Copy, Move, and Delete an API Service

You can create a new service by copying an existing one. Meanwhile, you can also move the service into another subgroup, or delete it.

### Copy

1. Go to the *Manage Service Registry* app. Under the *API Services* tab, choose an existing service of the "User-Defined" group.
2. Choose *Copy* on the service detail page.
3. In the *Copy Service* window, add a name, ID and select an existing subgroup under the "User-Defined" group.

#### Note

Selecting a subgroup is not mandatory. The service will be copied to the root "User-Defined" group by default if no subgroup is selected.

4. Choose *Confirm*.

### Move

1. Go to the *Manage Service Registry* app. Under the *API Services* tab, choose one or more existing services of the "User-Defined" group.
2. Choose *Move* on the service list page.
3. In the *Select Group* window, choose an existing subgroup under the "User-Defined" group.

#### Note


Alternatively, you can just change the group when editing the service on the service detail page.

## Delete

1. Go to the [Manage Service Registry](#) app. Under the [API Services](#) tab, choose one or more existing services of the "User-Defined" group..
2. Choose [Delete](#) on the service list page or go to the service detail page and choose [Delete](#).

### Note

SAP Digital Manufacturing services and production processes cannot be deleted. You need to uncheck the [Publish to Service Registry](#) in the Production Process Designer to remove production process services in the [Manage Service Registry](#) app.

If the service has been used in other places, the system will detect it automatically and you can check in the warning message box where this service is used once you perform the delete action. You can continue with your deletion or cancel. If deletion is continued in this case, an  (*error*) will be displayed next to the service in the consumer and the service can no longer be used.

To delete the current service and all the schemas used in this service, select the checkbox.

You cannot delete schemas used in other schemas, services, or schemas generated from production processes in the service registry. When you perform the delete action, a list will give you information on which schemas can be deleted and which not.

## 4.1.4 Import MII Transactions


You can import SAP Manufacturing Integration and Intelligence (MII) transactions as services to service registry. These services are then available for use in the production process designer.

### Prerequisites

- You have the role of `Manufacturing Extension Provider Or Production Engineer`.
- You have configured a destination for your MII system in your subaccount.

### Note

For more information on the integration setup, see [Integration with SAP MII](#).

There is a limitation for MII versions earlier than 15.3. For more information, see [2979122](#) .

The transaction can only have primitive data types as input or output parameters. List, map, and XML aren't supported.

- You have set up a web server for your MII system and configured the destination above in the web server in the [Manage Web Servers](#) app. Note that you need to connect the runtime web server with the MII web server if you want to use the MII transactions in production processes. For details, see [Manage Web Servers](#) and [Setting up Connections Between Web Servers](#).

## Procedure

1. Go to the *Manage Service Registry* app and under the *API Services* tab, choose **Import** **SAP MII**.
2. In the *Import API* window, select the web server for your MII system and then select one destination from the web server to load MII transactions.
3. Select one or more transactions that you want to import.

### Note

The transaction display follows the same structure as you defined in MII. You need to change the service ID when importing if it does not follow the naming conventions of service registry. See the service ID information in [Create an API Service \(RESTful and OData\) \[page 8\]](#) for naming conventions. If there is already service with the same ID in the same folder, the service is then updated.

4. Choose *Import*.

## Related Information

[Integration with SAP MII  
Available Services and Subprocesses](#)

## 4.2 Schemas

The service registry provides a central place for defining structures as schemas.

Schemas coming from SAP Digital Manufacturing services are automatically added to the *Schemas* tab when these services are registered in the *Manage Service Registry* app. You can also create your own schemas from scratch. The schemas can then be referenced and used in Production Process Designer (PPD) and automatic triggers for various purposes.

### Note

The system also generates schema automatically under the *Schemas* tab from inline schema of a newly-created service and schema created inside another schema.

The service registry supports structure (comprised of components, each of which has a name and a data type) to be defined as schema under the *Schemas* tab. The structures are described in JSON objects.

You can create and edit schema details, including its structure and field properties under the *Schemas* tab in both form view and code view.

You must have the role of `Manufacturing Extension Provider Or Production Engineer` to perform the action.



## 4.2.1 Create a Schema

You can create schemas. All created schemas are listed under the [Schemas](#) tab.

### Procedure

1. Go to the [Manage Service Registry](#) app and under the [Schemas](#) tab, choose [Create](#).
2. Fill in relevant fields.

Field	Description
Name	Required. Must be unique under each subgroup. Case-insensitive, ≤ 128 characters. Pound signs (#), slashes (/) are not allowed.
Group Path	All schemas you created will be under "User-Defined" group. You can either select an existing subgroup under "User-Defined" group or manually enter a new subgroup name to create one.  Alternatively, choose <a href="#">Manage Groups</a> on the schema list page to edit "User-Defined" subgroups.  You can delete a "User-Defined" subgroup if there is no schema in it.
Description	Optional. ≤ 255 characters.

3. The service registry supports two editing modes: form view and code view. Switch between modes by choosing [Switch to Code](#) or [Switch to Form](#).

### Form View

1. In [Schema Definition](#) section, choose [Add](#).
2. Fill in relevant fields in the right panel.

Field	Description
Name	Required.
Description	Optional. The description of the node.
Required	Choose <a href="#">Yes</a> if the node is mandatory.

Field	Description
Data Type	<p>Support <code>String</code>, <code>DateTime</code>, <code>Byte</code>, <code>Boolean</code>, <code>Integer</code>, <code>Short</code>, <code>Long</code>, <code>Number</code>, <code>Double</code>, <code>Float</code>, <code>Structure</code> and arrays of these data types.</p> <div data-bbox="821 454 1394 1016" style="background-color: #f0f0f0; padding: 10px;"> <p><b>Note</b></p> <p>For <code>Structure</code> or <code>StructureArray</code>, you can reference a schema or create the structure directly inline. To create the structure directly inline, choose <a href="#">Add Child</a> and then you can save the structure as a separate schema by defining a name for it. The inline schema will be saved to the same group of the parent schema you are currently creating.</p> <p>To reference a schema, check the box of <a href="#">Use Existing Schema</a> and select schema from list.</p> <p>The type of schema's outermost layer is always an object and cannot be an array. Nesting in a schema an array-of-array structure, <code>[ [ 1 , 2 , 3 ] , [ 4 , 5 , 6 ] ]</code>, for example, is also not supported.</p> </div>
Enum	<p>List of values that the schema accepts. Press <a href="#">Enter</a> to add each value.</p> <div data-bbox="821 1120 1394 1272" style="background-color: #f0f0f0; padding: 10px;"> <p><b>Note</b></p> <p>Only <code>String</code> enum can be consumed in production process designer.</p> </div>
Item Enum	<p>List of values that the items of array schema accept. Press <a href="#">Enter</a> to add each value.</p> <div data-bbox="821 1375 1394 1527" style="background-color: #f0f0f0; padding: 10px;"> <p><b>Note</b></p> <p>Only <code>String</code> item enum of array can be consumed in production process designer.</p> </div>
Default Value	<p>If any enum values are defined, default value must be one of them. You can also add a default value without defining enum.</p>

## Code View

In [Schema Definition](#) section, fill in the data structure definition and some additional information like required fields.

### Note

You can create nested schema under the [Schemas](#) tab and nested as references, or define embedded schemas inline. The inline schema will be saved separately along with the parent schema if you specify "title" of an object. Don't specify the "title" if you don't want the inline schema to be saved separately. To nest a schema, reference to it using its group path. Example for creating a schema with a nested one:

### Sample Code

```
{
  "type": "object",
  "properties": {
    "id": {
      "type": "string",
      "description": "Pet ID"
    },
    "name": {
      "type": "string",
      "description": "Pet name"
    },
    "honors": {
      "type": "array",
      "items": {"$ref": "/User Defined/pets/petHonor"}
    }
  }
}
```

After creation, the nested schema path will be replaced with an ID.

4. Choose *Create*.

### Note

Here is an example of schema definition in form view and code view.

### Example

#### Form View

Field	Description	Data Type	Required
name	Pet name	String	Yes
▼ petOwner	Pet owner	Structure	No
name	Pet owner's name	String	Yes
▼ adoptionHistory	Pet's adoption history	StructureArray	No

#### Code View

### Sample Code

```
{
  "type": "object",
  "required": [
    "name"
  ],
  "properties": {
    "name": {
      "description": "Pet name",
      "type": "string"
    }
  }
}
```

Form View

Code View

adoptionHistory - Structure (adoption- Yes  
History)

```
    }  
  ,  
  "petOwner  
  ": {  
  
    "descript  
    ion":  
    "Pet  
    owner ",  
  
    "type":  
    "object",  
  
    "required  
    ": [  
  
      "name"  
  
    ],  
  
    "properti  
    es": {  
  
      "name": {  
  
        "descript  
        ion":  
        "Pet  
        owner's  
        name",  
  
        "type":  
        "string",  
  
        "title":  
        ""  
  
      }  
  
    }  
  },  
  "adoption  
  History":  
  {  
  
    "descript  
    ion":
```

**Form View****Code View**

```
"Pet's
adoption
history",

"type":
"array",

"items":
{

"$ref":
"d9650d1b
-2f50-4bb
a-
bf26-2736
a84bd44b"

}
},

"title":
"pet"
}
```

Choose [Edit](#) to edit existing schemas under the "User-Defined" group. You can edit both the direct properties of the current schema and the properties of its referenced child schema on the same page. Your changes are simultaneously refreshed across all instances where this child schema is used under the current schema. If the schema has been used in other places, the system will detect it automatically and you can check in the warning message box where this schema is used once you save your edit. You can save a copy of all schemas and child schemas, continue with your update or cancel. Alternatively, you can also choose [Check Where-Used](#) in the schema detail page to check where the schema has been used. Refresh the list to see the latest changes if the schema gets updated.

Besides the general information about the consumers provided in the where-used list, the usage status under the [Production Processes](#) tab indicates if the schema is used in an up-to-date manner by its consumers. You should pay attention and take certain actions to those schemas with "outdated" usage status. Possible usage status:

- **Normal:** The schema is being used normally by the consumer.
- **Outdated (compatible):** The parameter or variable of the consumer using the schema is outdated due to schema compatible changes. Recommend checking further.
- **Outdated (incompatible):** The parameter or variable of the consumer using the schema is outdated due to schema incompatible changes. You need to adjust the structure value assignment accordingly to keep using it.

Navigate directly to each consumer through its link and check the schema accordingly.

After the update, in the Production Process Designer or subscription in the [Manage Automatic Triggers](#) app the service or the message payload structure using this schema will be marked as outdated. You need to

synchronize and redo the value mapping accordingly. The notification of service outdate is not applicable to simply updating the description field of the schema.

## Related Information

[Registering Services in the SaaS Tenant of SAP Digital Manufacturing \[page 6\]](#)

[Manage Service Registry](#)

### 4.2.2 Copy, Move, and Delete a Schema

You can create a new schema by copying an existing one. Meanwhile, you can also move the schema into another subgroup, or delete it.

#### Copy

1. Go to the [Manage Service Registry](#) app. Under the [Schemas](#) tab, choose an existing schema of the "User-Defined" group.
2. Choose [Copy](#) on the schema detail page.
3. In the [Copy Schema](#) window, add a name and select an existing subgroup under the "User-Defined" group.

#### Note

To copy the current schema and all of its corresponding child schemas, select the checkbox.

4. Choose [Confirm](#).

#### Move

1. Go to the [Manage Service Registry](#) app. Under the [Schemas](#) tab, choose one or more existing schemas of the "User-Defined" group.
2. Choose [Move](#) on the schema list page.
3. In the [Select Group](#) window, choose an existing subgroup under the "User-Defined" group.

#### Note

Alternatively, you can just change the group when editing the schema on the schema detail page.

## Delete

1. Go to the [Manage Service Registry](#) app. Under the [Schemas](#) tab, choose one or more existing schemas of the "User-Defined" group.
2. Choose [Delete](#) on the schema list page or go to the schema detail page and choose [Delete](#).

### Note

To delete the current schema and all of its corresponding child schemas, select the checkbox.

You cannot delete schemas used in other schemas, services, or schemas generated from production processes in the service registry. However, you can check in the warning message box where this schema is used once you perform the delete action.

## 4.3 UI Extensions

You can register POD plugins as services in the service registry.

### Prerequisites

You must have the role of `Manufacturing Extension Provider Or Production Engineer` to perform the action.

### Register

1. Go to the [Manage Service Registry](#) app and under the [UI Extensions](#) tab, choose [Create](#).
2. Fill in the relevant fields.

Field	Description
Name	The name can only contain A–Z, a–z, 0–9 and underscores. It must start with a letter and end either with a letter or a number. Double underscore is not allowed.
Description	Optional. The description of UI extension.
Type	Supports POD plugins only. This service type must include the server and location of the plugin extensions. For more information, see <a href="#">Creating and Deploying Custom POD Plugins</a> .

Field	Description
URL	Required. UI5 application URL.
	<div style="border-left: 2px solid #0070C0; padding-left: 10px; background-color: #F0F0F0;"> <p><b>Note</b></p> <p>Only the URL under SAP Digital Manufacturing default domain (end with "hana.ondemand.com" or "kyma.ondemand.com") is supported.</p> </div>
Path	Required. HTML5 module name. For more information, see <a href="#">Creating and Deploying Custom POD Plugins</a> .
Namespace	The application namespace minus the "path" name. For more information, see <a href="#">Creating and Deploying Custom POD Plugins</a> .

3. Choose *Create*.

#### **Note**

For information on how to use POD plugins in the POD Designer after registration, see [Configuring Plugin Extensions](#).

## Delete

Choose *Delete* to delete the current UI extensions under the UI extension detail page or delete in batch in the UI extension list page.





# Important Disclaimers and Legal Information

## Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

## Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

## Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

## Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

## Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2024 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.