





**PLATFORM DEEP DIVE / Plugins / 📦 Custom Plugins / 📦 Documents plugin /  
Using the plugin / Generating docs based on templates / generating-from-html-  
templates**

# Contents

- PLATFORM DEEP DIVE / Plugins /  Custom Plugins /  Documents plugin / Using the plugin / Generating docs based on templates / Generating from HTML templates
  - Creating a template
  - Sending the request
  - Reply

## PLATFORM DEEP DIVE / Plugins / Custom Plugins / Documents plugin / Using the plugin / Generating docs based on templates / Generating from HTML templates

The Document Management Plugin allows you to generate documents based on previously defined document templates. This example specifically covers generating documents using HTML templates.

### Creating a template

Use the **WYSIWYG** editor to create a document template.

**Process Definitions** Search by process defin

**Drafts / In progress**

Name	Version	Edited at	Edited by	
S...	1	30 Sep 2022, 5:41 PM	Silviu Grigore	▶ ✎ ⋮
d...	2	30 Sep 2022, 2:32 PM	andrei antal	▶ ✎ ⋮
t...	1	30 Sep 2022, 11:15 AM	QA FlowX	▶ ✎ ⋮
T...	4	30 Sep 2022, 10:20 AM	QA FlowX	▶ ✎ ⋮

**Published**

Name	Version	Published at	Published by	
A...	1	03 Oct 2022, 8:12 AM	QA FlowX	▶ ✎ ⋮
C...	19	30 Sep 2022, 3:08 PM	Silviu Grigore	▶ ✎ ⋮
d...	1	30 Sep 2022, 10:34 AM	Bogdan Ionescu	▶ ✎ ⋮
T...	1	30 Sep 2022, 10:18 AM	QA FlowX	▶ ✎ ⋮

**Navigation Sidebar:**

- Processes
  - Definitions
  - Active process
  - Process Instances
  - Failed process start
- Content Management
  - Enumerations
  - Substitution tags
  - Content models
  - Languages
  - Source systems
- Plugins
  - Task Manager
    - All tasks
    - Hooks
    - Stages
    - Allocation rules
    - Out of office
  - Notification templates

**User Profile:** John Doe

## Sending the request

1. Create a process that includes a **Kafka send event node** and a **Kafka receive event node** (one for sending the request and one for receiving the reply).
2. Configure the first node (Kafka Send Event) by adding a **Kafka send action**.

3. Add the **Kafka topic** to which the request should be sent.
4. Fill in the message with the following expected values in the request body:

#### Parameters

Custom

From integration

#### Topics

ai.flowx.in.qa.document.html.generate.v1

#### Message

```
1 {
2   "clientType": "PF",
3   "documentList": [
4     {
5       "customId": "123456",
6       "templateName": "test_doc",
7       "language": "en",
8       "data": {
9       },
10      "includeBarcode": true
11    }
12  ]
13 }
```

#### Advanced configuration

Show Headers ☒

```
1 {"processInstanceId": "${processInstanceId}"}
```

- **documentList**: A list of documents to be generated with properties (name and value to be replaced in the document templates)
- **customId**: Client ID
- **templateName**: The name of the template to be used
- **language**
- **includeBarcode**: True/False
- **data**: A map containing the values that should be replaced in the document template. The keys used in the map should match the ones defined in the

HTML template.

### INFO

Kafka topic names can be set by using (overwriting) the following environment variables in the deployment:

- **KAFKA\_TOPIC\_DOCUMENT\_GENERATE\_HTML\_IN** - default value: `ai.flowx.in.qa.document.html.generate.v1` - the topic that listens for the request from the engine
- **KAFKA\_TOPIC\_DOCUMENT\_GENERATE\_HTML\_OUT** - default value: `ai.flowx.updates.qa.document.html.generate.v1` - the topic on which the engine expects the reply

The above examples of topics are extracted from an internal testing environment. When setting topics for other environments, follow the pattern

`ai.flowx.updates.{{environment}}.document.generate.v1`.

### CAUTION

The engine listens for messages on topics with specific naming patterns. Make sure to use an outgoing topic name that matches the pattern configured in the engine.

## Reply

### INFO

You can view the response by accessing the **Audit log** menu.

The response will be sent on the output Kafka topic defined in the Kafka Receive Event Node. The response will contain the following information:

#### Audit log details

**Event:** process instance, message receive, 25 Oct 2022 at 5:57 PM

**Url:** ai.flowx.updates.qa.document.html.generate.v1

**Body:**

```
1  {"generatedFiles":{"123456":{"test_doc":{"customId":"123456",  
    "fileId":4746,"documentType":"test_doc",  
    "documentLabel":"GENERATED_PDF",  
    "minioPath":"qualitance-dev-paperflow-qa-process-id-759232/123456/  
    4746_test_doc.pdf","downloadPath":"internal/files/4746/download",  
    "noOfPages":1,"error":null}}},"error":null}
```

Values expected in the event body:

- **generatedFiles**: List of generated files.
  - **customId**: Client ID.
  - **fileId**: The ID of the generated file.
  - **documentType**: The name of the document template.
  - **documentLabel**: A label or description for the document.
  - **minioPath**: The path where the converted file is saved. It represents the location of the file in the storage system, whether it's a MinIO path or an S3 path, depending on the specific storage solution.
  - **downloadPath**: The download path for the converted file. It specifies the location from where the file can be downloaded.
  - **noOfPages**: The number of pages in the generated file.
  - **error**: If there were any errors encountered during the generation process, they would be specified here. In the provided example, the value is null, indicating no errors.

Example of generated file response received on

KAFKA\_TOPIC\_DOCUMENT\_GENERATE\_HTML\_IN topic :

```
{
  "generatedFiles": {
    "123456": {
      "test_doc": {
        "customId": "123456",
        "fileId": 4746,
        "documentType": "test_doc",
        "documentLabel": "GENERATED_PDF",
        "minioPath": "qualitance-dev-paperflow-qa-process-
id-759232/123456/4746_test_doc.pdf", //or S3 path, depending
on your storage solution
        "downloadPath": "internal/files/4746/download",
```

```
        "noOfPages": 1,  
        "error": null  
    }  
}  
},  
"error": null  
}
```

**Was this page helpful?**