FLOWX.AI

PLATFORM DEEP DIVE / Plugins / Custom Plugins / Notifications plugin / Using the plugin

Contents

PLATFORM DEEP DIVE / Plugins / Custom Plugins / Notifications plugin / Using the plugin / Managing notification templates

You can create and manage notification templates using

The fallback content to display on prerendering
web app, by accessing the dedicated section.

Configuring a template
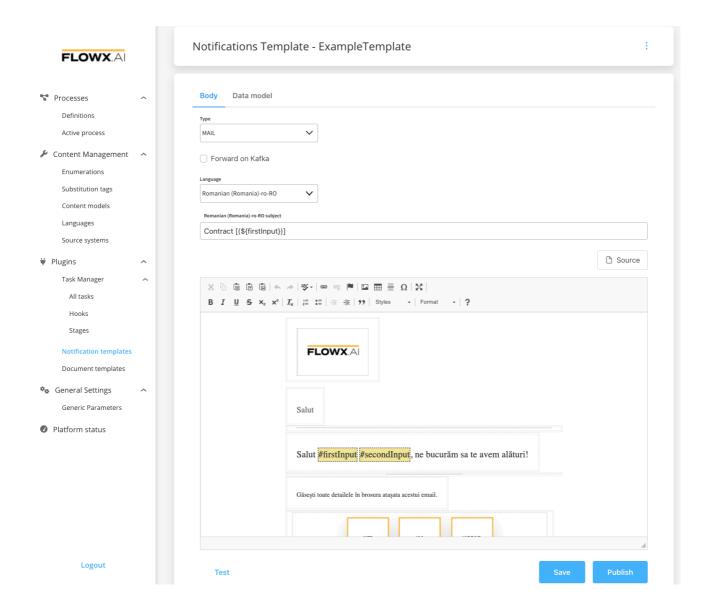
To configure a document template, first, you need to select some information stored in the **Body**:

1. **Type** - could be either MAIL or SMS notifications

2. **Forward on Kafka** - if this checkbox is ticked, the notification is not being sent directly by the plugin to the destination, but forwarded to another adapter (this is mandatory for SMS notifications templates, as they require an external adapter)

3. **Language** - choose the language for your notification template

4. **Subject** - enter a subject

Editing the content

You can edit the content of a notification template by using the WYSIWYG editor embedded in the body of the notification templates body.

Configuring the data model

Using the
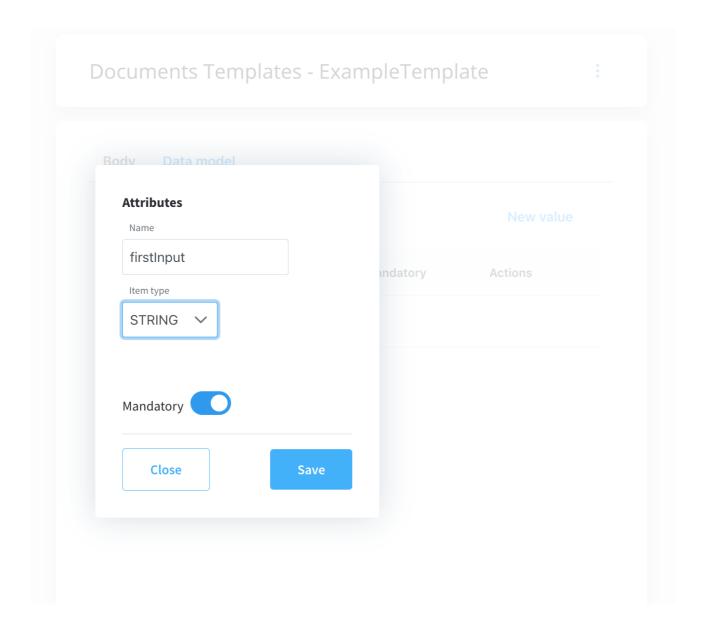
The fallback content to display on prerendering

, you can define key pair values (parameters) that will be displayed and reused in the editor. Multiple parameters can be added:

- STRING

- NUMBER

- BOOLEAN

- OBJECT

- ARRAY (which has an additional `item` field)

Documents Templates - ExampleTemplate

Body   Data model

Attributes

Name

firstInput

Item type

STRING

New value

andatory   Actions

Mandatory

Close   **Save**
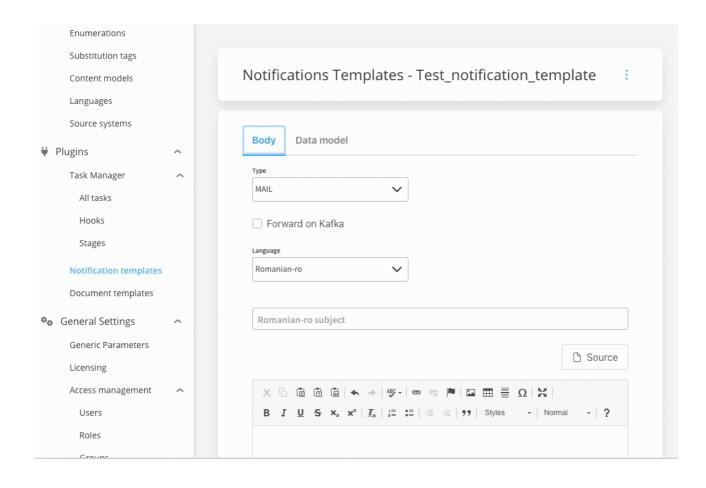
After you defined some parameters in the **Data Model** tab, you can type "**#**" in the editor to trigger a dropdown where you can choose which one you want to use/reuse.
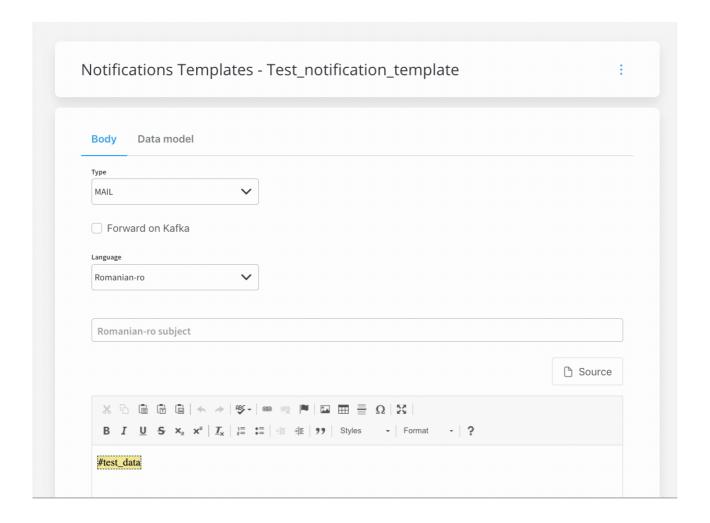
» WYSIWYG Editor

Testing the template
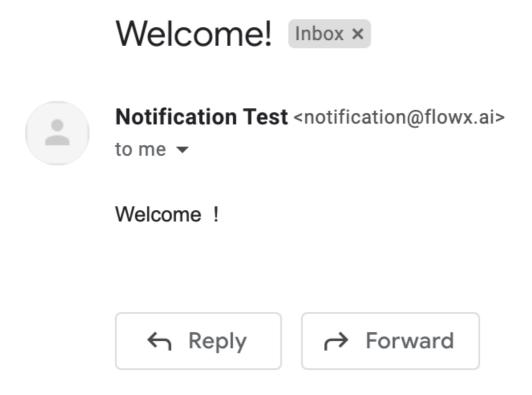
You can use the test function to ensure that your template configuration is working as it should before publishing it.

In the example above, some keys (marked as mandatory) were not used in the template, letting you know that you've missed some important information. After you enter all the mandatory keys, the notification test will go through:

Welcome! Inbox ×

Notification Test <notification@flowx.ai>

to me ▾

Welcome !

← Reply   → Forward
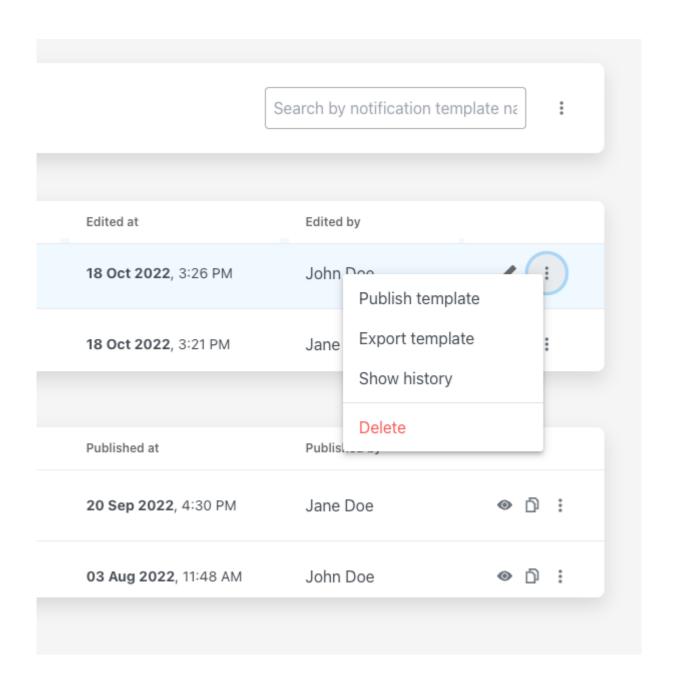
Other actions

When opening the contextual menu (accessible by clicking on the breadcrumbs button), you have multiple actions to work with the notifications templates:

- Publish template - publish a template (it will be then displayed in the **Published** tab), you can also clone published templates
- Export template - export a template (JSON format)
- Show history - (version history and last edited)

Was this page helpful?

PLATFORM DEEP DIVE / Plugins / Custom Plugins / Notifications plugin

/ Using the plugin / Sending a notification

The plugin can be used for sending many kinds of notifications such as emails or SMS notifications. It can be easily integrated in one of your business processes.

Configuring the process

To configure a business process that sends notifications you must follow the next steps:

- use

  The fallback content to display on prerendering

  web app to create/edit a notification template

- use

  The fallback content to display on prerendering

  to add a **Message send task** and a **Message received task**

- configure the needed actions

- configure the request body

- configure the needed Kafka topics

The following values are expected in the request body:

Key	Definition	
language	The language that should be used	Mandatory
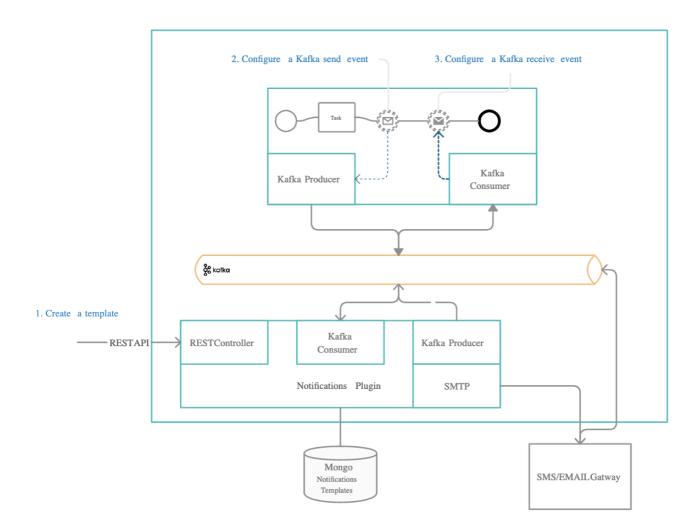
Key	Definition	
templateName	The name of the notification template that is used	Mandatory
channel	Notification channel: SMS/MAIL	Mandatory
receivers	Notification receivers: email/phone number	Mandatory
senderEmail	Notification sender email	Optional
senderName	Notification sender name	Optional
attachments	Attachments that are sent with the notification template (only used for MAIL notifications)	Optional

> ⓘ **INFO**
>
> Check the detailed example below.
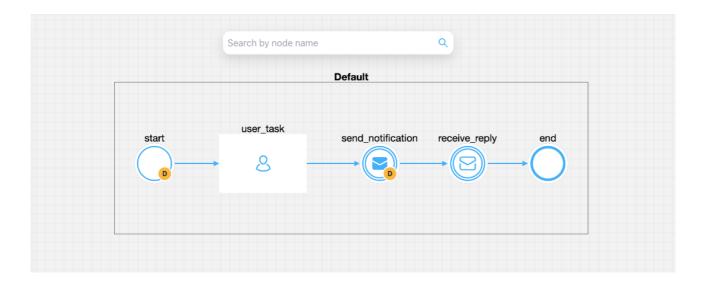
Define needed Kafka topics

Kafka topic names can be set by using environment variables:

- `KAFKA_TOPIC_NOTIFICATION_INTERNAL_IN` - topic used to trigger the request to send a notification

- `KAFKA_TOPIC_NOTIFICATION_INTERNAL_OUT` - topic used for sending replies after sending the notification

Example: send a notification from a business flow
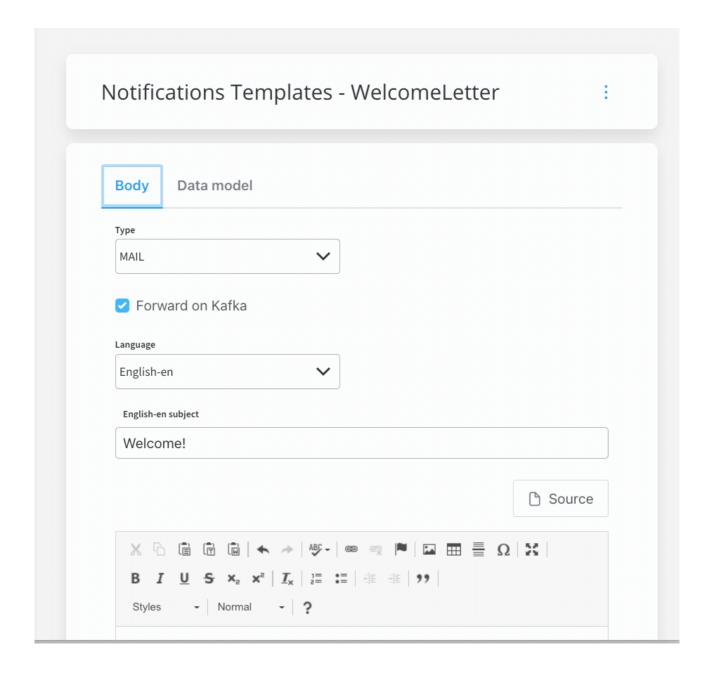
Let's pick a simple use-case, say we need to send a new welcome letter when we onboard a new customer. The steps are the following:

1. Configure the template that you want to use for the welcome email, see the previous section, Managing notification templates for more information.

2. Use the FLOWX.AI Designer to add a **Message send task** and a **Message received task**.

3. On the **Message send task** add a proper configuration to the action, the Kafka topic and request body message to be sent:

- **Topics** - `KAFKA_TOPIC_NOTIFICATION_INTERNAL_IN` - `flowx-notifications-qa`
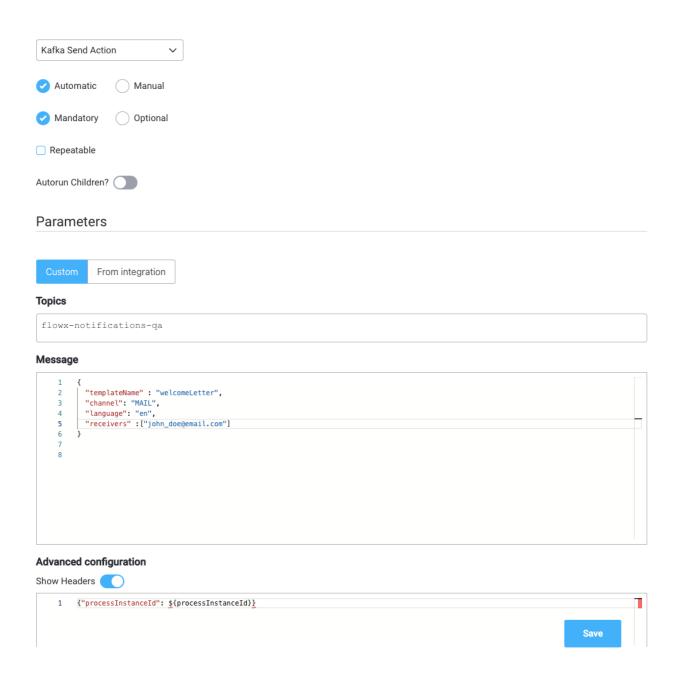
- **Message** (expected parameters):
  - templateName
  - channel
  - language
  - receivers
- **Headers** - it is always `{"processInstanceId": ${processInstanceId}}`
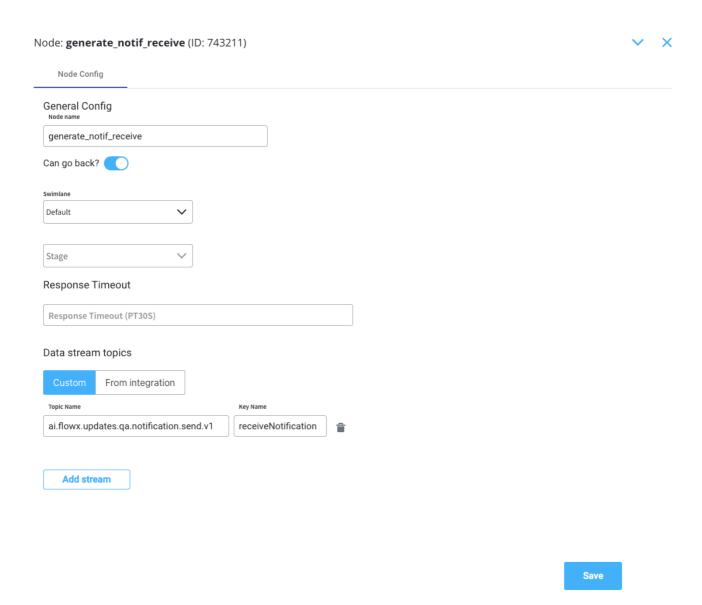
4. On the **Message received task** add the needed topic to receive the kafka response - `KAFKA_TOPIC_NOTIFICATION_INTERNAL_OUT` - `ai.flowx.updates.qa.notification.request.v1`.

FLOWx.AI

Node: **generate_notif_receive** (ID: 743211)

Node Config

General Config
Node name

generate_notif_receive

Can go back?

Swimlane

Default

Stage

Response Timeout

Response Timeout (PT30S)

Data stream topics

| Custom | From integration |

Topic Name | Key Name

ai.flowx.updates.qa.notification.send.v1 | receiveNotification

Add stream

Save
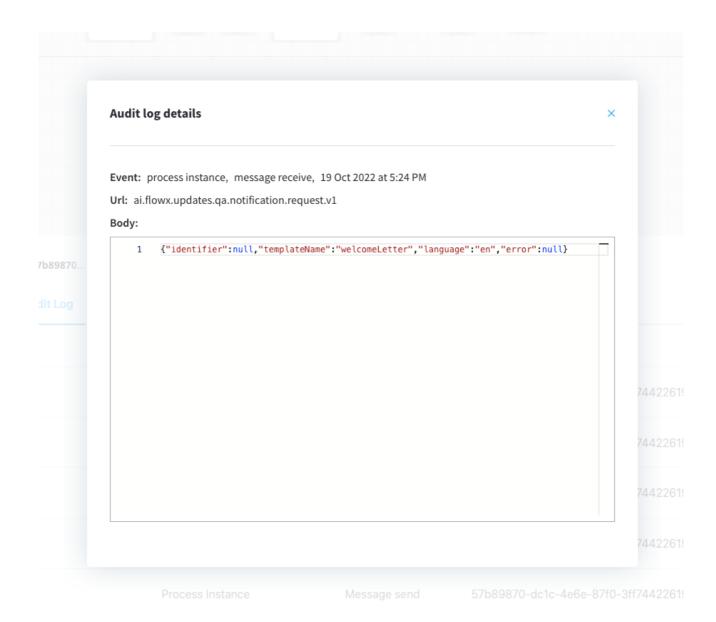
5. Run the process and look for the response (you can view it via the **Audit log**)
or checking the responses on the Kafka topic defined at
`KAFKA_TOPIC_NOTIFICATION_INTERNAL_OUT` variable.

Audit log details ✕

Event: process instance, message receive, 19 Oct 2022 at 5:24 PM

Url: ai.flowx.updates.qa.notification.request.v1

Body:

```
1    {"identifier":null,"templateName":"welcomeLetter","language":"en","error":null}
```

7b89870..

dit Log

7442261

7442261

7442261

7442261

Process Instance   Message send   57b89870-dc1c-4e6e-87f0-3ff7442261

Response example at `KAFKA_TOPIC_NOTIFICATION_INTERNAL_OUT`:

```
{
  "identifier": null,
  "templateName": "welcomeLetter",
  "language": "en",
  "error": null
}
```

Was this page helpful?

PLATFORM DEEP DIVE / Plugins / Custom Plugins / Notifications plugin / Using the plugin / Sending an email with attachments

To use the notification plugin for sending emails with attachments, you must define the same topic configuration as for sending regular notifications. A notification template must be created, and the corresponding Kafka topics must be defined.

» **Send a notification**
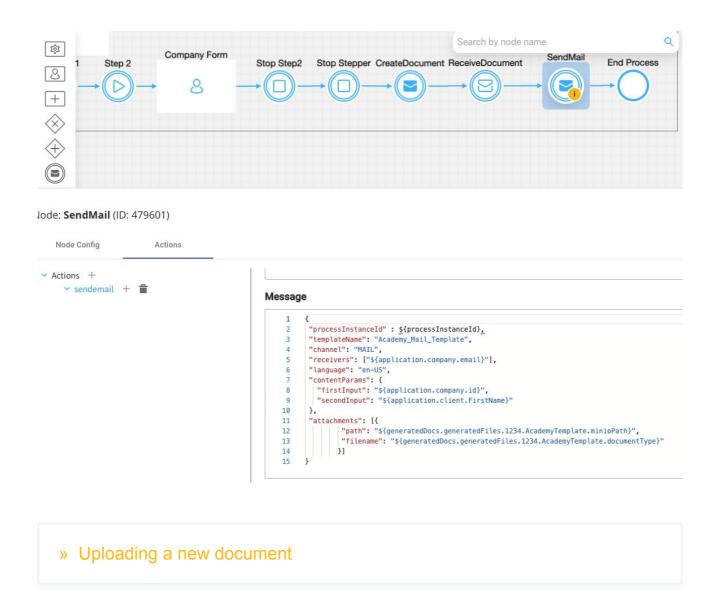
Defining process actions

Example: send an email notification with attached files from a business flow

Let's pick a simple use-case. Imagine we need to send a copy of a contract signed by a new customer. Before setting the action for the notification, another action must be defined, so the first one will save the new contract using the documents plugin.

» Uploading a new document

The steps for sending the notification are the following:

Step 1: Configure the template that you want to use for the email, see the Managing notification templates section for more information.

Step 2: Check that the needed topics are defined correctly on the following environment variables:

```
KAFKA_TOPIC_NOTIFICATION_INTERNAL_IN
```

`KAFKA_TOPIC_NOTIFICATION_INTERNAL_OUT`

Step 3: Use the

The fallback content to display on prerendering
to add a new Kafka send event action to the correct node in the process definition.

Step 4: Add the proper configuration to the action, the Kafka topic and message to be sent.

The message to be sent to Kafka will look something like:

```
{
  "templateName" : "contractCopy",
  "identifier" : "text",
  "language": "en",
  "receivers" : [ "someone@somewhere.com" ],
  "contentParams" : {
    "clientId" : "clientId",
    "firstName" : "first",
    "lastName" : "last"
  },
  "attachments" : [ {
    "filename" : "contract",
    "path" : "MINIO_BUCKET_PATH/contract.pdf"
  } ]
}
```

Was this page helpful?

PLATFORM DEEP DIVE / Plugins / Custom Plugins / Notifications plugin / Using the plugin / Forward notifications to an external system

If the Notification service is not directly connected to an SMTP / SMS server and you want to use an external system for sending the notifications, you can use the notification plugin just to forward the notifications to your custom implementation.

Define needed Kafka topics

Kafka topic names can be set by using environment variables:

- `KAFKA_TOPIC_NOTIFICATION_INTERNAL_IN` - topic used to trigger the request to send a notification
- `KAFKA_TOPIC_NOTIFICATION_EXTERNAL_OUT` - the notification will be forwarded on this topic to be handled by an external system
- `KAFKA_TOPIC_NOTIFICATION_INTERNAL_OUT` - topic used for sending replies after sending the notification

Example: send a notification from a business flow

Let's pick a simple use case. Imagine we need to send a new welcome letter when we onboard a new customer. You must follow the next steps:

1. Configure the template that you want to use for the welcome email, use the WYSIWYG Editor
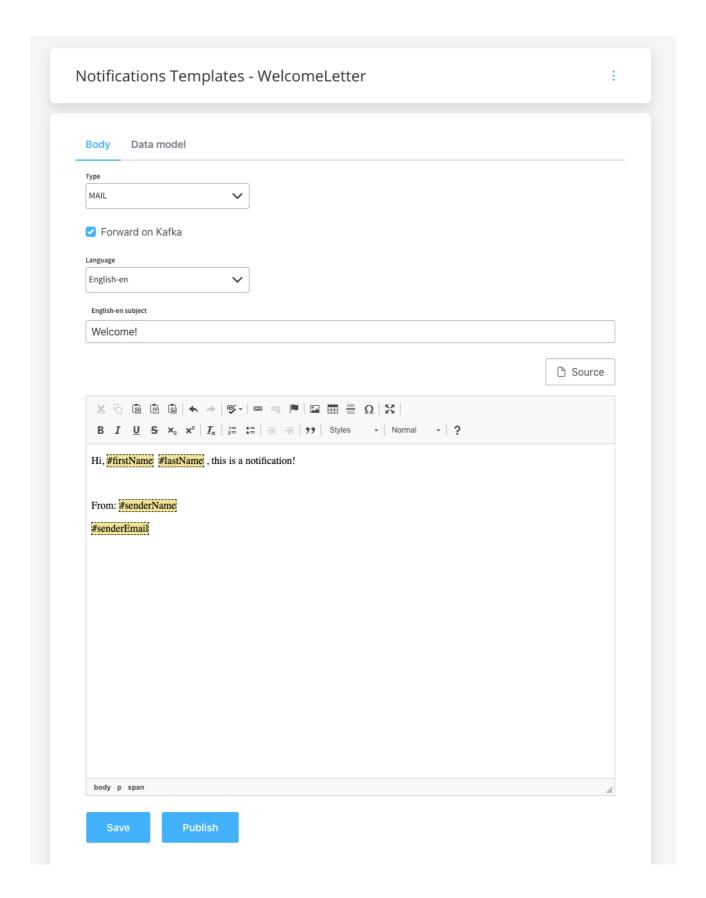
> ⚠️ **CAUTION**
>
> Make sure that the **Forward on Kafka** checkbox is ticked, so the notification
> will be forwarded to an external adapter.
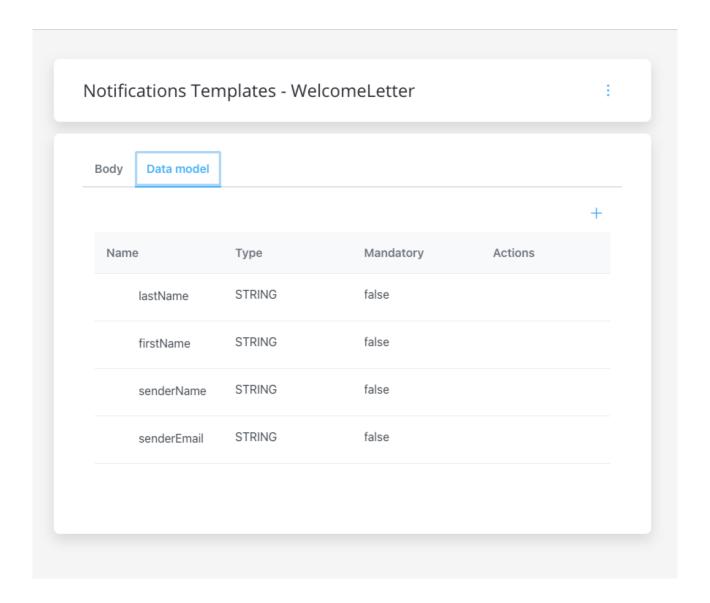
2. Configure the data model for the template.

3. To configure a document template, first, you need to define some information stored in the Body:

- **Type** - MAIL (for email notifications)
- ❗ **Forward on Kafka** - if this box is checked, the notification is not being sent directly by the plugin to the destination, but forwarded to another adapter
- **Language** - choose the language for your notification template
- **Subject** - enter a subject

Notifications Templates - WelcomeLetter

Body   Data model

Type

MAIL ⌄

☑ Forward on Kafka

Language

English-en ⌄

English-en subject

Welcome!

📄 Source

Hi, #firstName #lastName , this is a notification!

From: #senderName

#senderEmail

body p span

Save   **Publish**

Notifications Templates - WelcomeLetter

Body | **Data model**

Name	Type	Mandatory	Actions
lastName	STRING	false	
firstName	STRING	false	
senderName	STRING	false	
senderEmail	STRING	false	

4. Use the FLOWX.AI Designer to create a process definition.

5. Add a **Kafka send event node** and a **Kafka receive event node** (one to send the request, one to receive the reply).

6. Check if the needed topic (defined at the following environment variable) is configured correctly: `KAFKA_TOPIC_NOTIFICATION_INTERNAL_IN`.

7. Add the proper configuration to the action, the Kafka topic, and the body message.

Kafka Send Action ▾

☑ Automatic   ◯ Manual

☑ Mandatory   ◯ Optional

☐ Repeatable

Autorun Children? ◯—

Parameters

[Custom] [From integration]

Topics

```
flowx-notifications-qa
```

Message

```
1  {
2    "templateName" : "welcomeLetter",
3    "channel": "MAIL",
4    "language": "en",
5    "receivers" :["john_doe@email.com"]
6  }
7
8
```

Advanced configuration

Show Headers ●—

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

[Save]
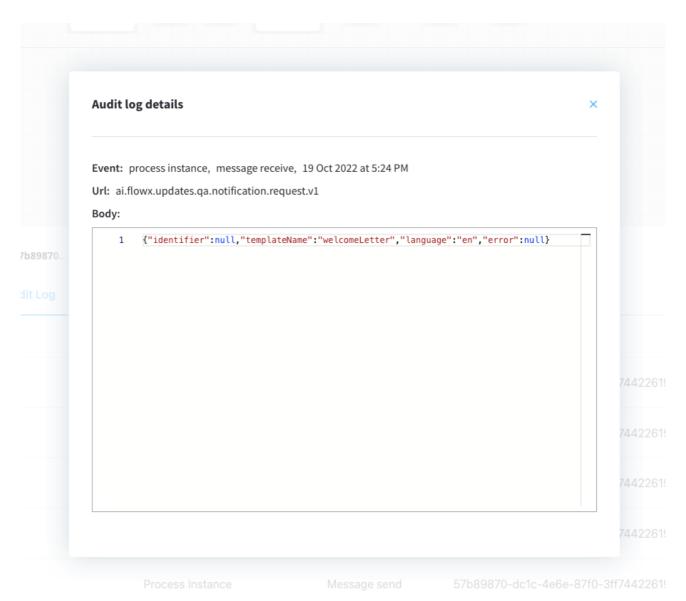
⚠ **INFO**

Forward on Kafka option will forward the notification to an external adapter, make sure the needed Kafka topic for forwarding is defined/overwritten using the following environment variable: `KAFKA_TOPIC_EXTERNAL_OUT`.

7. Run the process and look for the response (you can view it via the **Audit log**)
or by checking the responses on the Kafka topic

Audit log details   ✕

Event: process instance, message receive, 19 Oct 2022 at 5:24 PM

Url: ai.flowx.updates.qa.notification.request.v1

Body:

```
1    {"identifier":null,"templateName":"welcomeLetter","language":"en","error":null}
```

Process Instance   Message send   57b89870-dc1c-4e6e-87f0-3ff74422619

Response example at `KAFKA_TOPIC_NOTIFICATION_INTERNAL_OUT`:

```
{
  "templateName": "welcomeLetter",
  "receivers": [
```

```
        "john.doe@mail.com"
    ],
    "channel": "MAIL",
    "language": "en"
}
```
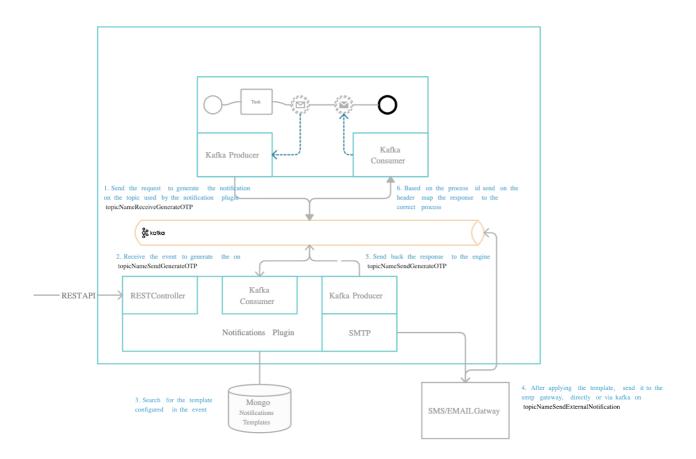
Was this page helpful?

PLATFORM DEEP DIVE / Plugins / Custom Plugins / Notifications plugin / Using the plugin / OTP flow / Generate OTP

There are some cases when you will need to generate an OTP (One Time Password) from a business flow, for example when validating an email account.

The notifications plugin handles both the actual OTP code generation and sending the code to the user using a defined notification template.

Define needed Kafka topics

Kafka topic names can be set by using environment variables:

- `KAFKA_TOPIC_OTP_GENERATE_IN`
- `KAFKA_TOPIC_OTP_GENERATE_OUT` - after the OTP is generated and sent to the user, this is the topic used to send the response back to the Engine.

> ⚠️ **CAUTION**
>
> The Engine is listening for messages on topics with names of a certain pattern, make sure to use an outgoing topic name that matches the pattern configured in the Engine.

Request to generate an OTP

Values expected in the request body:

- templateName: the name of the notification template that is used (created using the WYSIWYG editor)

- channel: notification channel: SMS / MAIL

- recipient: notification receiver: email / phone number

- notification template content parameters (for example, clientId): parameters that should be replaced in the notification template

Parameters

Custom | From integration

Topics

```
ro.flowx.qa.notification.otp.in.generate.v1
```

Message

```
1   {
2       "templateName": "otpMail",
3       "channel" : "MAIL",
4       "language": "en-US",
5       "clientId" : "${application.client.identificationData.personalIdentificationNumber}",
6       "recipient" : "${application.client.contactData.email.emailAddress}",
7       "contentParams":
8           {
9               "clientId":"${application.client.identificationData.personalIdentificationNumber}"
10          }
11
12  }
13
```

Advanced configuration

Show Headers 🔵

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

Response from generate OTP
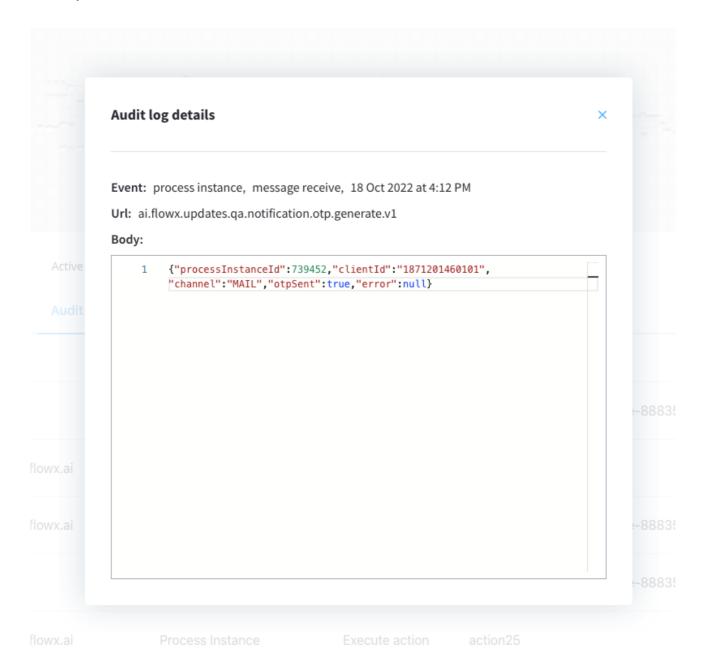
Values expected in the reply body:

- processInstanceId = process instance ID

- clientId = the client id (in this case the SSN number of the client)

- channel = notification channel used

- otpSent = confirmation if the notification was sent: true or false

- error = error description, if any

Example:

Audit log details

Event: process instance, message receive, 18 Oct 2022 at 4:12 PM

Url: ai.flowx.updates.qa.notification.otp.generate.v1

Body:

```
1    {"processInstanceId":739452,"clientId":"1871201460101",
     "channel":"MAIL","otpSent":true,"error":null}
```

Example: generate an OTP from a business flow
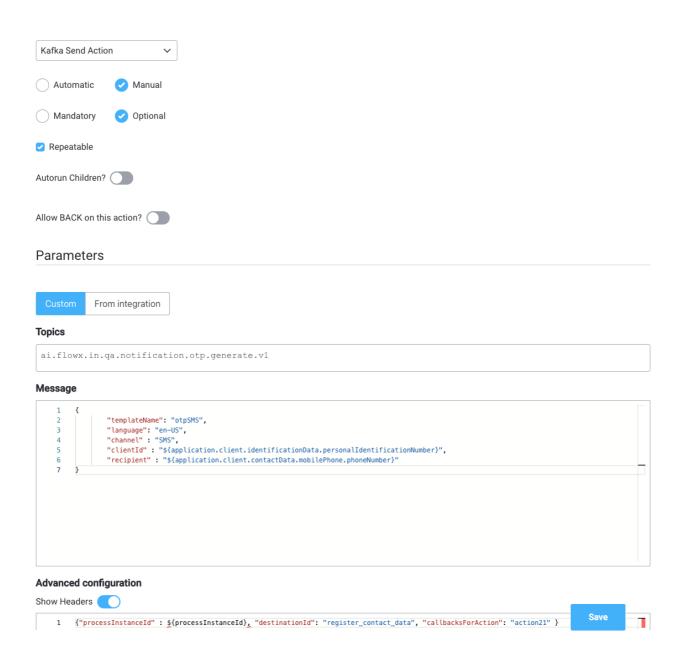
It is important to identify what is the business identifier that you are going to use to validate that OTP, it can be, for example, a user identification number.

1. Configure the templates that you want to use (for example, an SMS template).
2. Check that the needed topics are configured correctly: the topic used to generate OTP (`KAFKA_TOPIC_OTP_GENERATE_IN`) and the topic used to receive the response (`KAFKA_TOPIC_OTP_GENERATE_OUT`).
3. Use the FLOWX.AI Designer to add a new Kafka send event to the correct node in the process definition.
4. Add the proper configuration to the action, the Kafka topic, and configure the body message.

Kafka Send Action ∨

○ Automatic ✓ Manual

○ Mandatory ✓ Optional

✓ Repeatable

Autorun Children? ⬤

Allow BACK on this action? ⬤

Parameters

[Custom] [From integration]

Topics

```
ai.flowx.in.qa.notification.otp.generate.v1
```

Message

```
1  {
2      "templateName": "otpSMS",
3      "language": "en-US",
4      "channel" : "SMS",
5      "clientId" : "${application.client.identificationData.personalIdentificationNumber}",
6      "recipient" : "${application.client.contactData.mobilePhone.phoneNumber}"
7  }
```

Advanced configuration

Show Headers ⬤

```
1  {"processInstanceId" : ${processInstanceId}, "destinationId": "register_contact_data", "callbacksForAction": "action21" }
```

Save

5. Add a node to the process definition (for the Kafka receive event).

6. Configure on what key you want to receive the response on the process instance params.

Topic Name	Key Name
ro.flowx.updates.qa.notification.otp.generate.v1	otpCheck

PLATFORM DEEP DIVE / Plugins / Custom Plugins / Notifications plugin / Using the plugin / OTP flow / Validate OTP

Define needed Kafka topics

Kafka topic names can be set by using environment variables:

- `KAFKA_TOPIC_OTP_VALIDATE_IN` - the event sent on this topic (with an OTP and an identifier) will check if the OTP is valid

- `KAFKA_TOPIC_OTP_VALIDATE_OUT` - the response for this request will validate an OTP, the reply is sent back to the Engine on this topic

> ⚠️ **CAUTION**
>
> The Engine is listening for messages on topics with names of a certain pattern, make sure to use an outgoing topic name that matches the pattern configured in the Engine.

Request to validate an OTP

Values expected in the request body:

- processInstanceId = process instance ID

- client id = the user unique ID in the system

- channel = notification channel: SMS/MAIL

- otp = OTP code that you received, used to compare with the one that was
  sent from the system

Example:

```
{
    "processInstanceId": 12345,
    "clientId": "1871201460101",
    "channel": "MAIL",
    "otp": "1111"
}
```
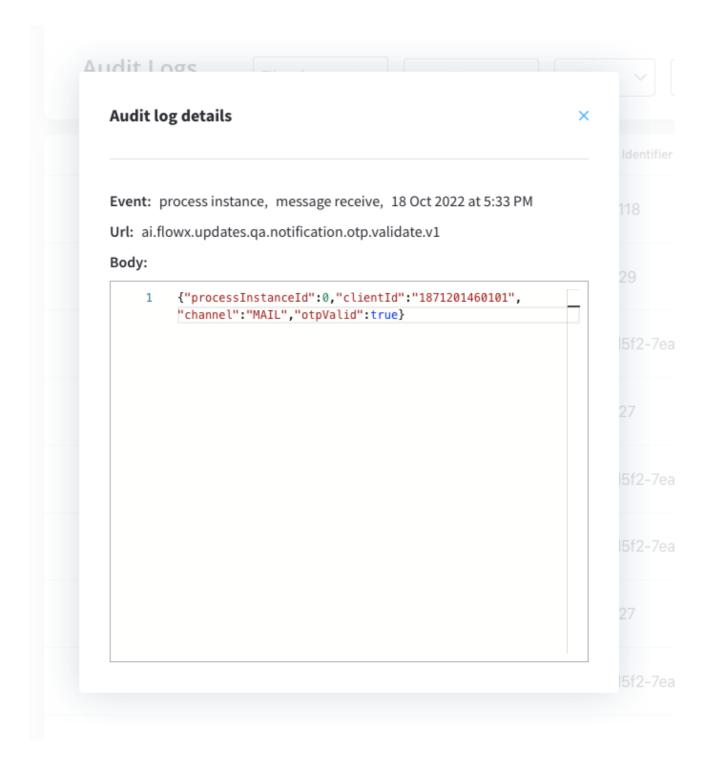
Reply from validate OTP

Values expected in the reply body:

- client id = the user unique id in the system

- channel = notification channel used

- otpValid = confirmation if the provided OTP code was the same as the one
  sent from the system

Example:

Audit log details ✕

Event: process instance, message receive, 18 Oct 2022 at 5:33 PM

Url: ai.flowx.updates.qa.notification.otp.validate.v1

Body:

```
1    {"processInstanceId":0,"clientId":"1871201460101",
     "channel":"MAIL","otpValid":true}
```

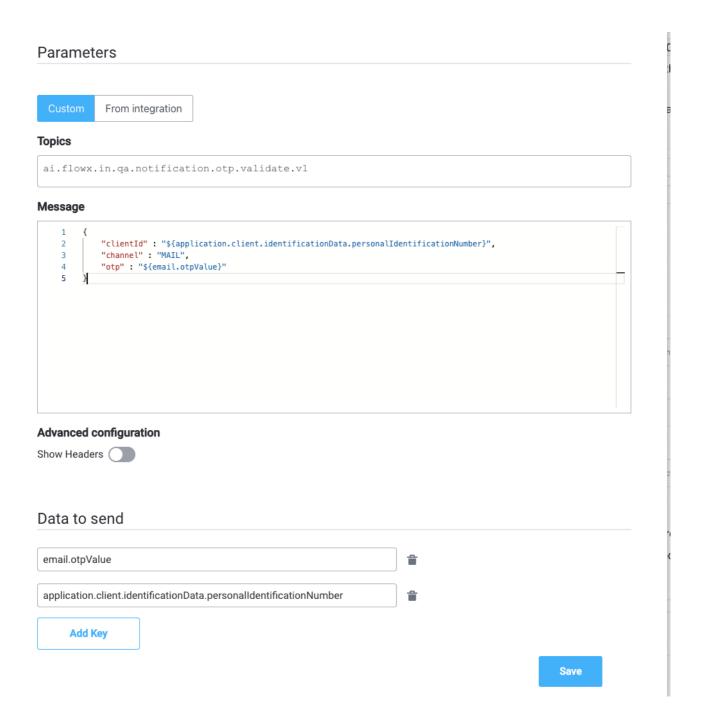# Example: validate an OTP from a business flow

Similar to the generation of the OTP you can validate the OTP that was generated for an identifier.

1. Check that the needed topics are configured correctly:
   (`KAFKA_TOPIC_OTP_VALIDATE_IN` and
   `KAFKA_TOPIC_OTP_VALIDATE_OUT`)

2. Add the actions for sending the request to validate the OTP on the node that contains the 'Generate OTP' actions

3. Add the proper configuration to the action, the Kafka topic and configure the body message.

Parameters

| Custom | From integration |

Topics

```
ai.flowx.in.qa.notification.otp.validate.v1
```

Message

```
1  {
2      "clientId" : "${application.client.identificationData.personalIdentificationNumber}",
3      "channel" : "MAIL",
4      "otp" : "${email.otpValue}"
5  }
```

Advanced configuration

Show Headers

Data to send

```
email.otpValue
```

```
application.client.identificationData.personalIdentificationNumber
```

Add Key

Save

4. Add a node to the process definition (for the Kafka receive event)

5. Configure on what key you want to receive the response on the process instance parameters

FLOWx.AI

Node: **register_contact_data** (ID: 735010)

Node Config	Actions

Custom	From integration

Topic Name

ai.flowx.updates.qa.notification.otp.valida

Key Name

otpValid

Task Management

Update task management?

ⓘ Force Task Management Plugin to update information about this process after this node.

Was this page helpful?