



**PLATFORM DEEP DIVE / Core components / Core extensions / scheduler**

# Contents

- PLATFORM DEEP DIVE / Core components / Core extensions / Scheduler
  - Overview
  - Using the scheduler

## PLATFORM DEEP DIVE / Core components / Core extensions / Scheduler

### Overview

The Scheduler is part of the core components of the

The fallback content to display on prerendering

. It can be easily added to your custom FLOWX deployment to **enhance the core platform capabilities with functionality specific to scheduling messages.**

The service offers the possibility to schedule a message that you only need to process after a configured time period.

It can be quickly deployed on the chosen infrastructure and then connected to the

The fallback content to display on prerendering  
through Kafka events.

Let's go through the steps needed in order to deploy and set up the service:

» [Scheduler setup guide](#)

We've prepared some examples of various use cases where this service is useful:

## Using the scheduler

After deploying the scheduler service in your infrastructure, you can start using it to schedule messages that you need to process at a later time.

One such example would be to use the scheduler service to expire processes that were started but haven't been finished.

### DANGER

First you need to check the configured topics match the ones configured in the engine.

For example the engine topics

KAFKA\_TOPIC\_PROCESS\_SCHEDULE\_OUT\_SET and

KAFKA\_TOPIC\_PROCESS\_SCHEDULE\_OUT\_STOP **should be the same with the ones configured in the scheduler** (KAFKA\_TOPIC\_SCHEDULE\_IN\_SET and KAFKA\_TOPIC\_SCHEDULE\_IN\_STOP)

When a process is scheduled to expire, the engine sends the following message to the scheduler service (on the topic KAFKA\_TOPIC\_SCHEDULE\_IN\_SET):

```
{  
  "applicationName": "onboarding",
```

```
"applicationId": "04f82408-ee66-4c68-8162-b693b06bba00",
"payload": {
  "scheduledEventType": "EXPIRE_PROCESS",
  "processInstanceId": "04f82408-ee66-4c68-8162-
b693b06bba00"
},
"scheduledTime": 1621412209.353327,
"responseTopicName": "ai.flowx.process.expire.staging"
}
```

The scheduled time should be defined as `java.time.Instant`.

At the scheduled time, the payload will be sent back to the response topic defined in the message, like so:

```
{
  "scheduledEventType": "EXPIRE_PROCESS",
  "processInstanceId": "04f82408-ee66-4c68-8162-
b693b06bba00"
}
```

If you don't need the scheduled message anymore, you can discard it by sending the following message (on the topic `KAFKA_TOPIC_SCHEDULE_IN_STOP`)

```
{
  "applicationName": "onboarding",
  "applicationId": "04f82408-ee66-4c68-8162-b693b06bba00"
}
```

These fields, `applicationName` and `applicationId` are used to uniquely identify a scheduled message.

**Was this page helpful?**