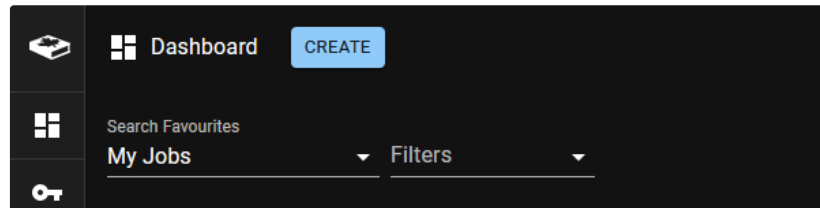# Create Job Wizard

Introduction

New to SPELLBOOK in 1.7.0 is the new **Job Wizard**, which allows you to create jobs in spellbook with ease. This feature improves upon existing job creation in spellbook, providing a number of new features that provide a faster, more convenient job creation experience.

Previously, the create job form did not provide any sort of template to build your job from, and required you to either manually copy an existing job, or write one from scratch. This led to a higher barrier of entry to users wanting to schedule a job.

Creating a Job

In order to create a job, you press the same button as before:

From here, instead of the old drawer, a new dialog will appear:

| 1 Basic Details | — | 2 DAG Template/Upload | — | 3 DAG Configuration | — | 4 Preview and Submit |

Please provide some additional information about your DAG. Required fields are marked with an asterisk.

Name *
Example DAG

Create as...
Rafuse, Matthew D.

Job key *
example_dag   ✓

This key must match the DAG id in your file. It must also be unique, consist of only alphanumeric characters and symbols (_ . -), and should not exceed 250 characters.

Description
This is an example DAG

Classification *
Unclassified   ▼

Tags

External References   +

No external references added.

HIDE EXTRA FIELDS   ⌃

NEXT   CANCEL

Similarly to before, you can specify the name of your job, the DAG ID, an optional description, and the classification of the job (if working in a multi-classification deployment). You can also expand the extra fields to add tags and external references, as before.

Next is the optional DAG Template/Upload page. This will allow you to pre-fill the DAG Configuration before editing it, either with a DAG Template provided by spellbook, or a file from your computer.

If you choose a DAG template, the Job Wizard will automatically prefill a number of the fields that a user had to manually fill out previously. Pressing next will move to the DAG Configuration Step.

> It isn't necessary to select a template or upload a file. If you want to write a DAG completely from scratch, simply press "Next".

```python
1    from datetime import datetime, timedelta
2
3    from airflow import DAG
4    from airflow.operators.python_operator import PythonOperator
5
6    default_args = {
7        'owner': 'Matthew.Rafuse@cyber.gc.ca',  # required
8        'start_date': datetime(2021, 4, 21),  # required
9        'email': 'Matthew.Rafuse@cyber.gc.ca',
10       # include this line if you don't want to be emailed when it fails
11       # 'email_on_failure': False,
12       # Enable these 3 lines to allow the job to retry.
13       # 'email_on_retry': True,
14       # 'retries': 1,
15       # 'retry_delay': timedelta(minutes=5)
16   }
17
18   dag = DAG(
19       dag_id='example_dag',
20       description='<optional_description>',
21       default_args=default_args,
22       schedule_interval='<cron_like_schedule>',
23       is_paused_upon_creation=False,
24       catchup=False
25   )
26
27
28   def python_callable():
29       # Any imports for this function need to be placed inside this function, not a
30       print("Do python stuff here")
31       return 'this is a returned xcom value'
32
33
34   python_operator = PythonOperator(
35       dag=dag,
```

PREVIOUS   NEXT   CANCEL

Once you proceed to the DAG configuration step, you'll note that there are a number of highlighted keys you will need to fill out before submitting your DAG. These include things like the crontab schedule, or the task names of the job's tasks. Note that some of these fields are automatically pre-filled, but can be edited, such as the owner, start date and email arguments.

```
17
18    dag = DAG(
19        dag_id='example_dag',
20        description='This is an example DAG',
21        default_args=default_args,
22        schedule_interval='@daily',
23        is_paused_upon_creation=False,
24        catchup=False
25    )
26
27
28    def python_callable():
29        # Any imports for this function need to be placed inside this function, not
30        print("Do python stuff here")
31        return 'this is a returned xcom value'
32
33
34    python_operator = PythonOperator(
35        dag=dag,
36        task_id='run_some_python',
37        python_callable=python_callable,
38        retries=3,
39        # Lineage inlets and outlets https://airflow.apache.org/docs/apache-airflow/
40        # inlets={
```

Once you have filled out all of the required fields, you can proceed to the Preview and Submit step. Here, the wizard validates your job and ensures there are no outstanding issues. If there are none, you should see the following:

✔ Basic Details —— ✔ DAG Template/Upload —— ✔ DAG Configuration —— 4 Preview and Submit

You're all set!
You can submit using the button below.

Basic Details	DAG Template/Upload	DAG Configuration
⊘ This step is complete, without errors.	⊘ This step is complete, without errors.	⊘ This step is complete, without errors.

PREVIOUS   SUBMIT   CANCEL

If there are outstanding issues, they will be listed here. Clicking on them will return you to the related page so you can fix them:

Once any issues are resolved, you can submit the job. Congrats, you've successfully used the Job Wizard for the first time!

Questions?

Any questions about this feature can be asked in the HOGWARTS support chat. If you have a DAG template you'd like to add, or any other suggestions, please let us know there. Thanks!