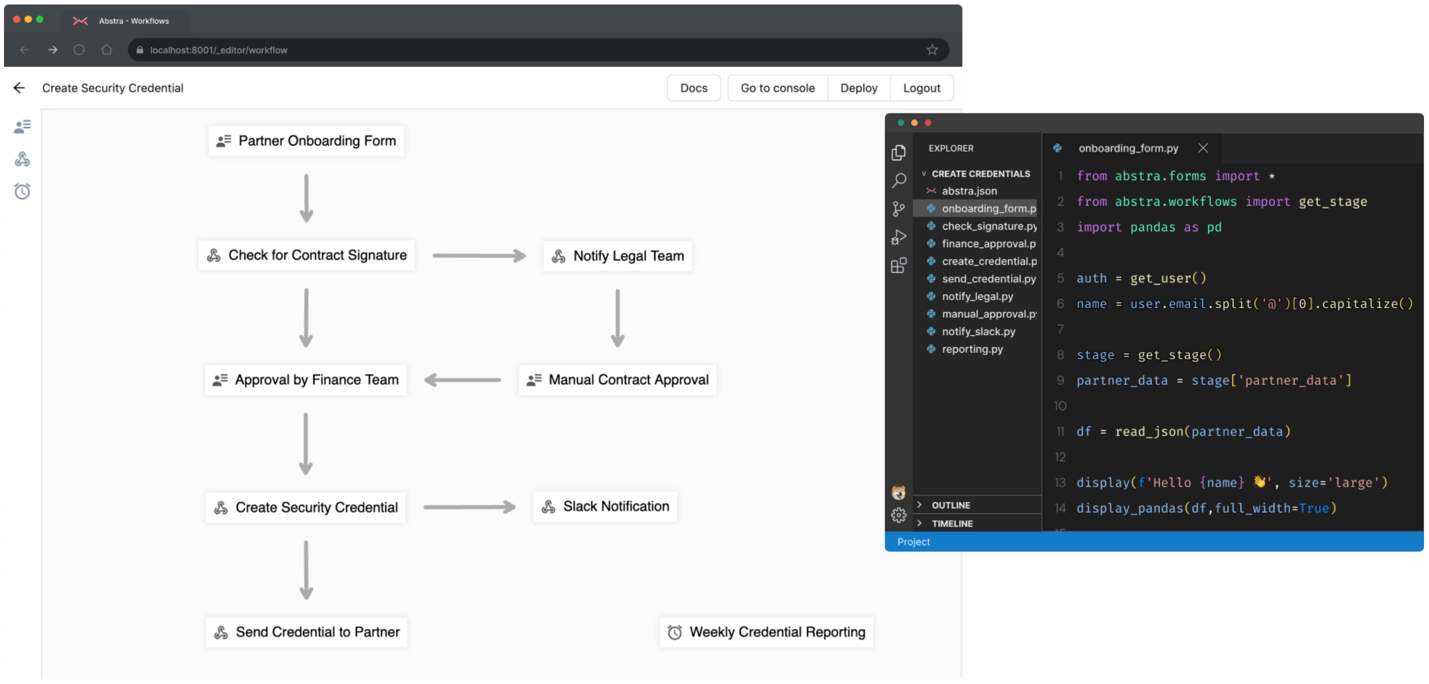
Workflows

INFO

The workflow is scheduled for an official release in early January. 🥳



```python
from abstra.forms import *
from abstra.workflows import get_stage
import pandas as pd

auth = get_user()
name = user.email.split('@')[0].capitalize()

stage = get_stage()
partner_data = stage['partner_data']

df = read_json(partner_data)

display(f'Hello {name} 👋', size='large')
display_pandas(df,full_width=True)
```

Workflows is a tool for automating business processes, perfect for processes that require a mix of manual tasks and integrations between systems.

A Workflow is made up of Python-coded stages, which are assembled and integrated visually in the Workflow editor.

Types of Stages

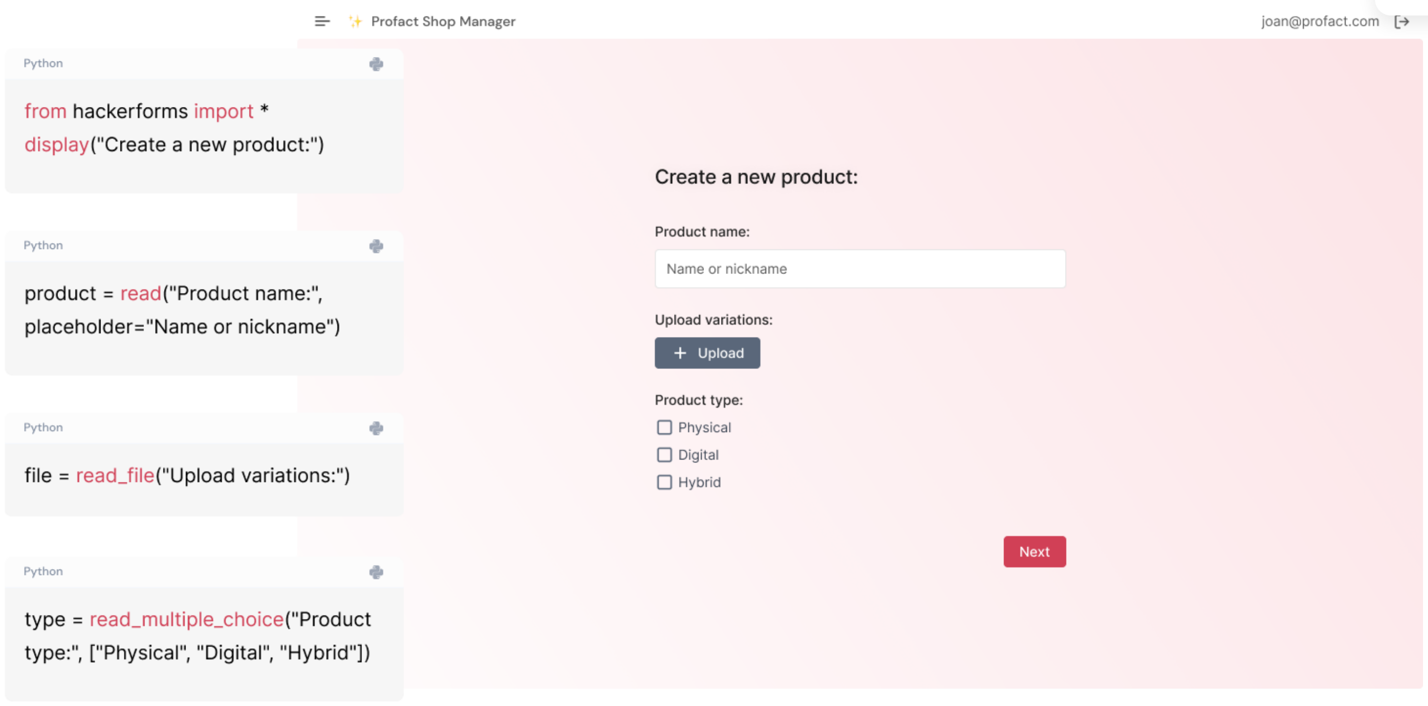
A stage is a single Python file, deployed as a Form, Hook, Job or Script. Let's get into each type.

Forms

Forms can be added into a Workflow whenever user interaction is needed in a process - for example, providing personal data, reviewing documents or approving requests. They are built with Abstra's UI-generating lib and created similarly to terminal programs.

When a user is assigned to a Form, the Workflow's run depends on their input to proceed. They will receive an email notification about this pending step. More on that on `Assignees`.

Read more on Forms here.



```python
from hackerforms import *
display("Create a new product:")
```

```python
product = read("Product name:", placeholder="Name or nickname")
```

```python
file = read_file("Upload variations:")
```

```python
type = read_multiple_choice("Product type:", ["Physical", "Digital", "Hybrid"])
```

Hooks

Hooks are stages triggered by HTTP requests. Whenever a Hook is added to a Workflow, an authorized HTTP request sent to the Hook's endpoint will trigger the script's execution.

Read more on Hooks here.

Jobs

Jobs are stages scheduled to be executed periodically. After writing a Job's code, you can select the frequency you'd like it to run on the editor - hourly, daily, weekly or monthly, and the selected time.

Since Jobs are time-based steps, they cannot be triggered by any other step within the Workflows. They can, however, trigger following events.

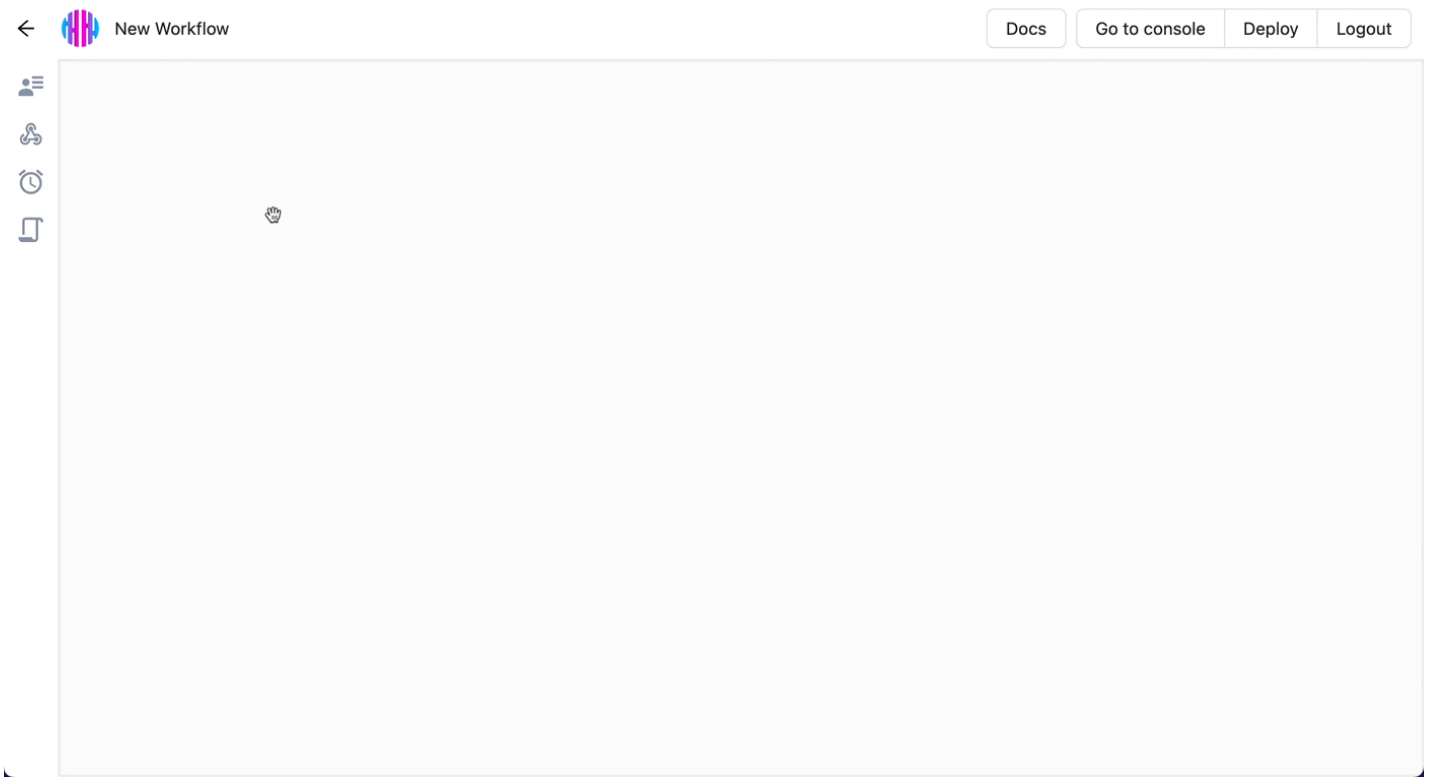
Scripts

Very straightforward - a Script is a plain Python script. It's trigger is simply the completion of the previous step on the Workflow - it's not user-based, time-based or based on an external event.

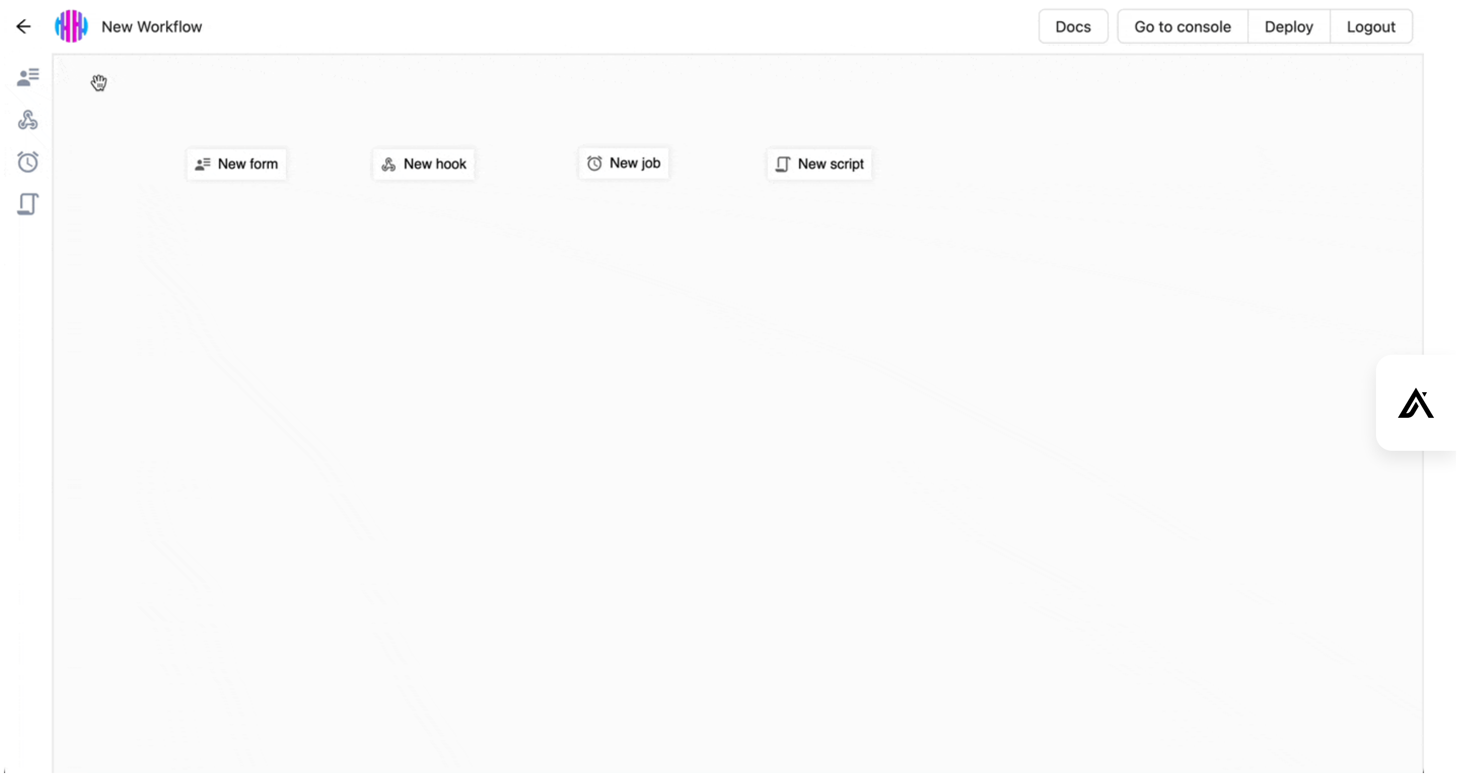
Add a Script to your Workflow whenever you need to simply run some Python code within your sequential process.

Adding stages to a Workflow

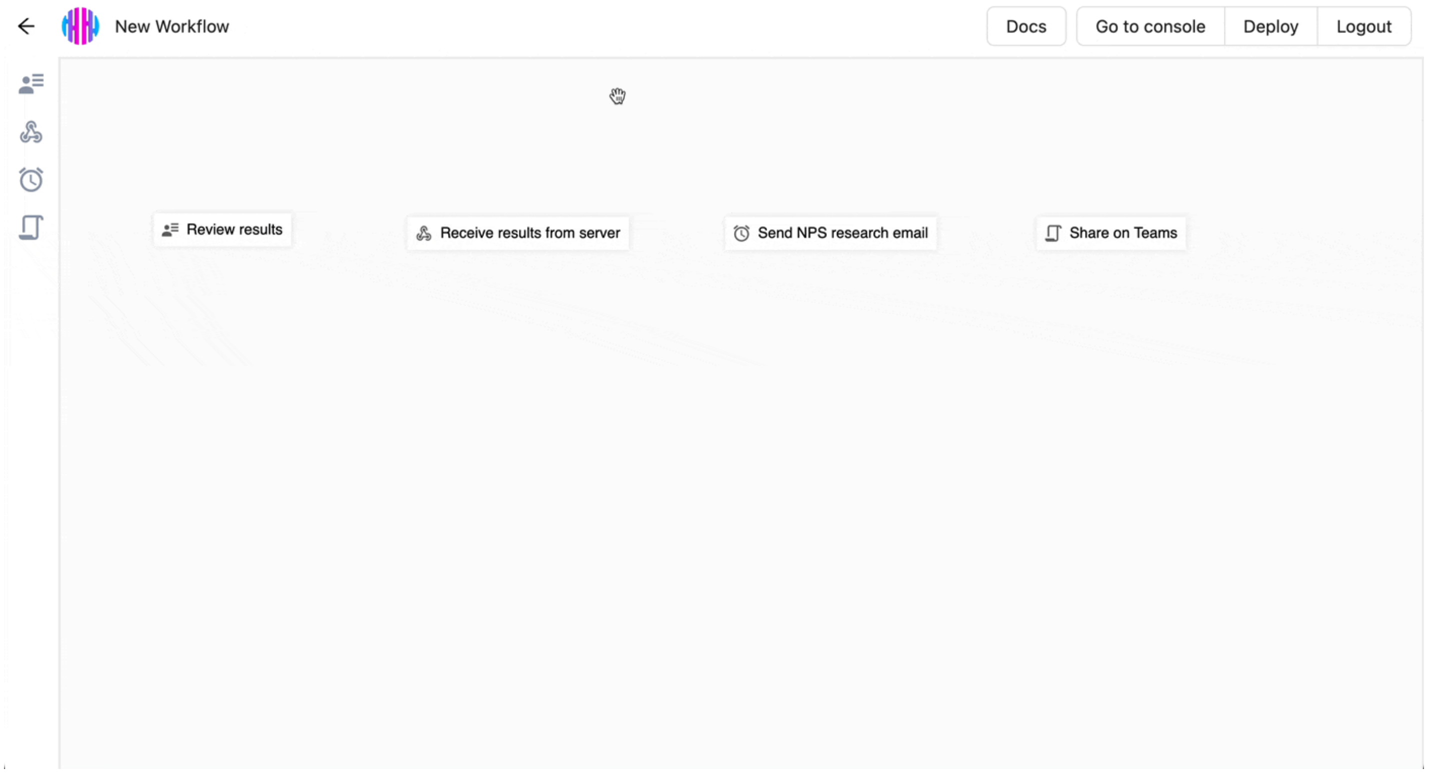
You can create a new stage directly in the Workflow editor, by dragging and dropping it from the left sidebar into the canvas.



By double-clicking a stage, an `Options` sidebar will appear. There you can edit settings, including the stage's code. Click on `Open file` to open the code on your preferred IDE.



Define your Workflow's sequence by connecting stages with directional transitions, represented by arrows. Click on a stage, select `Add transition (T)"` and drag the arrow to the stage you'd like to follow.



You can also create a Form, Job or Hook in their standalone editor and it will appear as an unconnected stage in the Workflow editor. Simply add a transition to incorporate it into the process.

NOTE

It's also possible to not connect a stage to your Workflow, keeping it as a self-contained task that runs independently within the project.

An example of this is a Job that runs weekly to send reporting info to the team's Slack.

Sharing variables between stages

Share relevant variables between multiples stages using the `get_stage()` function from the `abstra.workflows` lib.

```
from abstra.workflows import get_stage

# Get data that was previously stored in a stage
stage = get_stage()
email = stage['email']

# Save new data to be used in next stages
first_name = "Anna"
last_name = "Smith"
stage['full_name'] = first_name + last_name
```

Defining an `assignee` for a stage

Use the `next_stage()` function from the `abstra.workflows` lib to select a specific user or team to complete a certain step. This user or team will be referred to as an assignee.

When your Workflow's run reaches this step, they will receive an email notification.

Assignees are more common for Form stages, since they require user interaction.

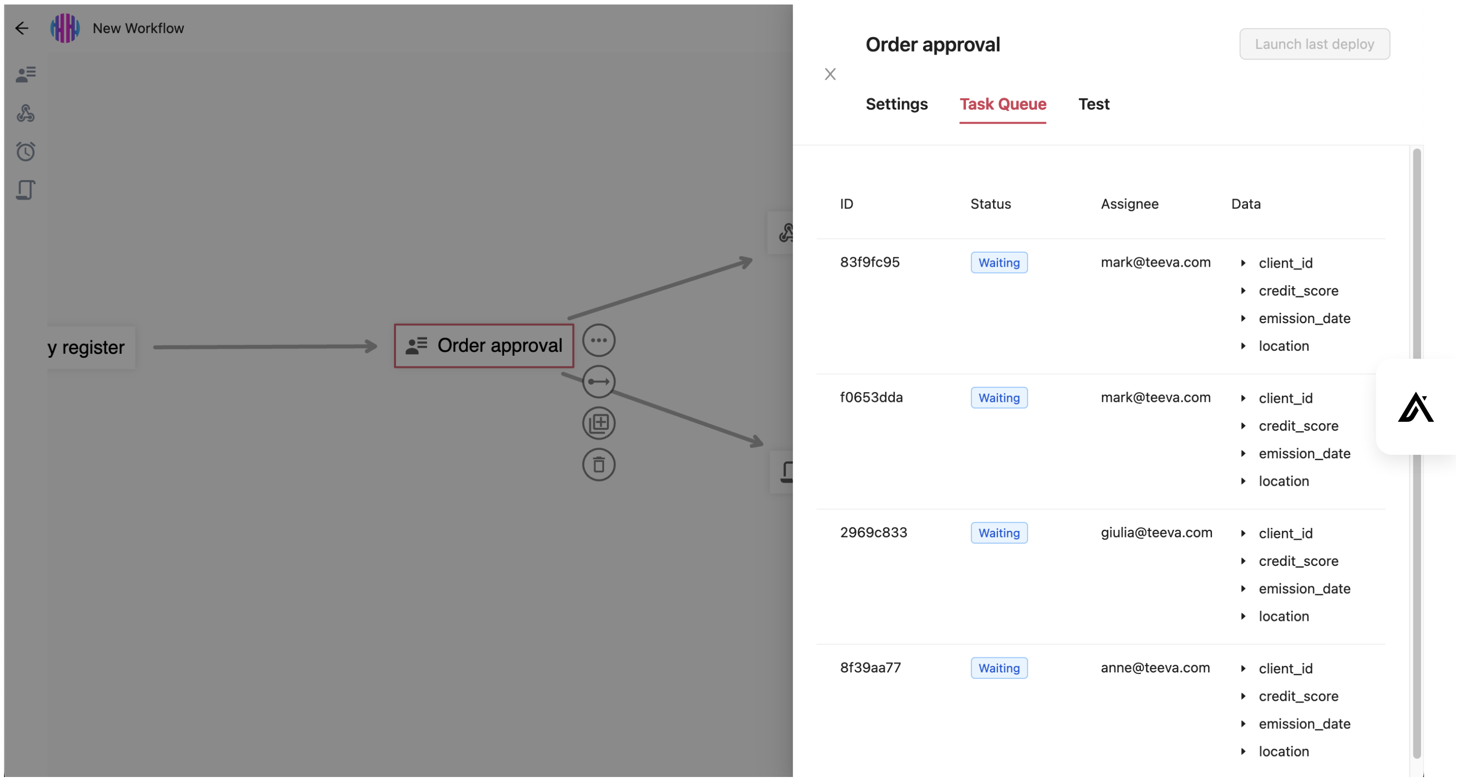
```
from abstra.workflows import next_stage
```

```
# Set assignee for next stage
next_stage([
        {
        "assignee": "mark@company.com",
        "data": {
            "team": "finance"
        },
        "stage": "contract-analysis" # use the next stage's path
    },
        {
        "assignee": "kiara@company.com",
        "data": {
            "team": "product"
        },
        "stage": "contract-analysis" # use the next stage's path
    },
])
```

Testing your Workflow

Every stage in a project can be tested in the `Options` sidebar. Navigate to the `Test` tab to run it.

By running a step, this will create a test Workflow run that will continue to next steps. This is useful to test continuity and a complete Workflow run.



ID	Status	Assignee	Data
83f9fc95	Waiting	mark@teeva.com	client_id credit_score emission_date location
f0653dda	Waiting	mark@teeva.com	client_id credit_score emission_date location
2969c833	Waiting	giulia@teeva.com	client_id credit_score emission_date location
8f39aa77	Waiting	anne@teeva.com	client_id credit_score emission_date location

Relaunching the server will erase all test runs.

Deploying your Workflow

`Deploy` your entire Workflow in a single click, by hitting the `Deploy` button on the top right corner.

CAUTION

Always remember to add a `requirements.txt` file to your project's folder, with every Python package you're using in each stage. Your Deploy won't work without it!

Running your Workflow

After deploy, a Workflow run will be triggered every time the first step is completed. It will run through all steps as indicated in your Workflow.

This first step can be triggered manually, like filling a Form, or automatically, by receiving an HTTP request with a Hook or running a scheduled Job.

Currently, you can keep track of step execution on the Project's `Logs` tab on Abstra Cloud.

INFO

Coming late January: a detailed Kanban view to manage Workflow runs and granular access control.