**Create a Blank Workflow**

The first thing we need to do is create a blank workflow. To do this, open up Alfred Preferences and navigate to the ‘**Workflows**’ tab. At the bottom of the workflows pane, you will be able to see a ‘**+**’ button. Click on ‘+’ and select ‘**Blank Workflow**’. We can name our blank workflow as ‘Climate Clock’.

A new window will pop-up. It contains a blank form with fields like Name, Description, Created By, etc. That information needs to be filled out.

Name- Climate Clock

Description- Displays information from the climateclock API

**Bundling Dependencies**

In order to make an API call within our Python script, we had to import the libraries that we used in our python code. We used requests, dateutil (pandas), json.

Here, requests, json is not part of the Python Standard Library. Due to this, if a user were to download our workflow, it would not work unless they had manually installed these libraries onto their machine.

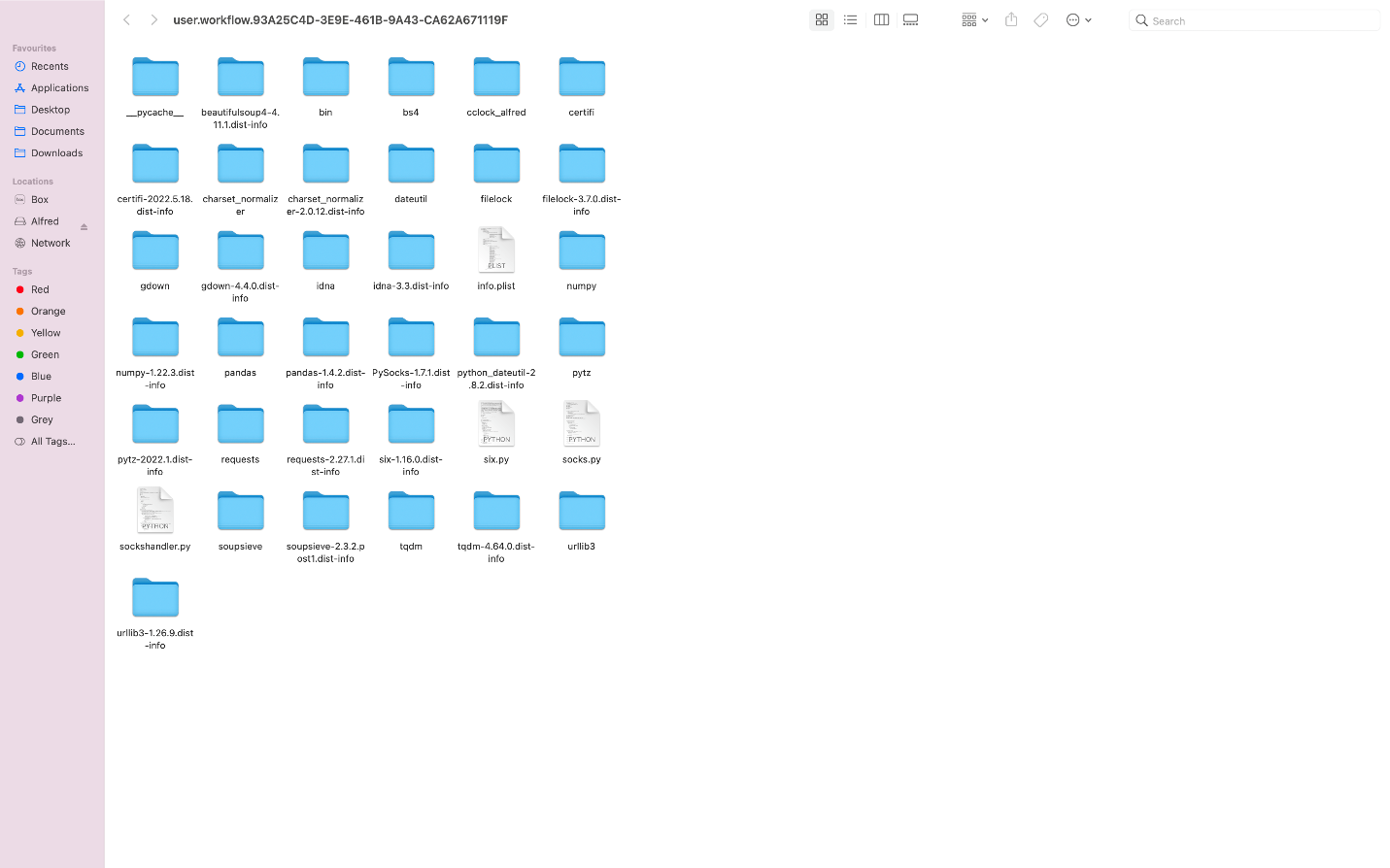
**Installing the libraries in the workflow directory**

In order to bundle these libraries, we need to install the dependencies within the workflow directory. To locate the workflow directory, right-click on the workflow that we created which we named as climate clock. Click ‘Open in Terminal’ and ‘Open in Finder’.

Within the terminal, run the following command to install the module within the current working directory.

* pip3 install –target=. requests
* pip3 install –target=. json
* pip3 install –target=. pandas

After the install completes, we can see that the libraries and its dependencies are within the directory list.



**Adding a Script Filter**

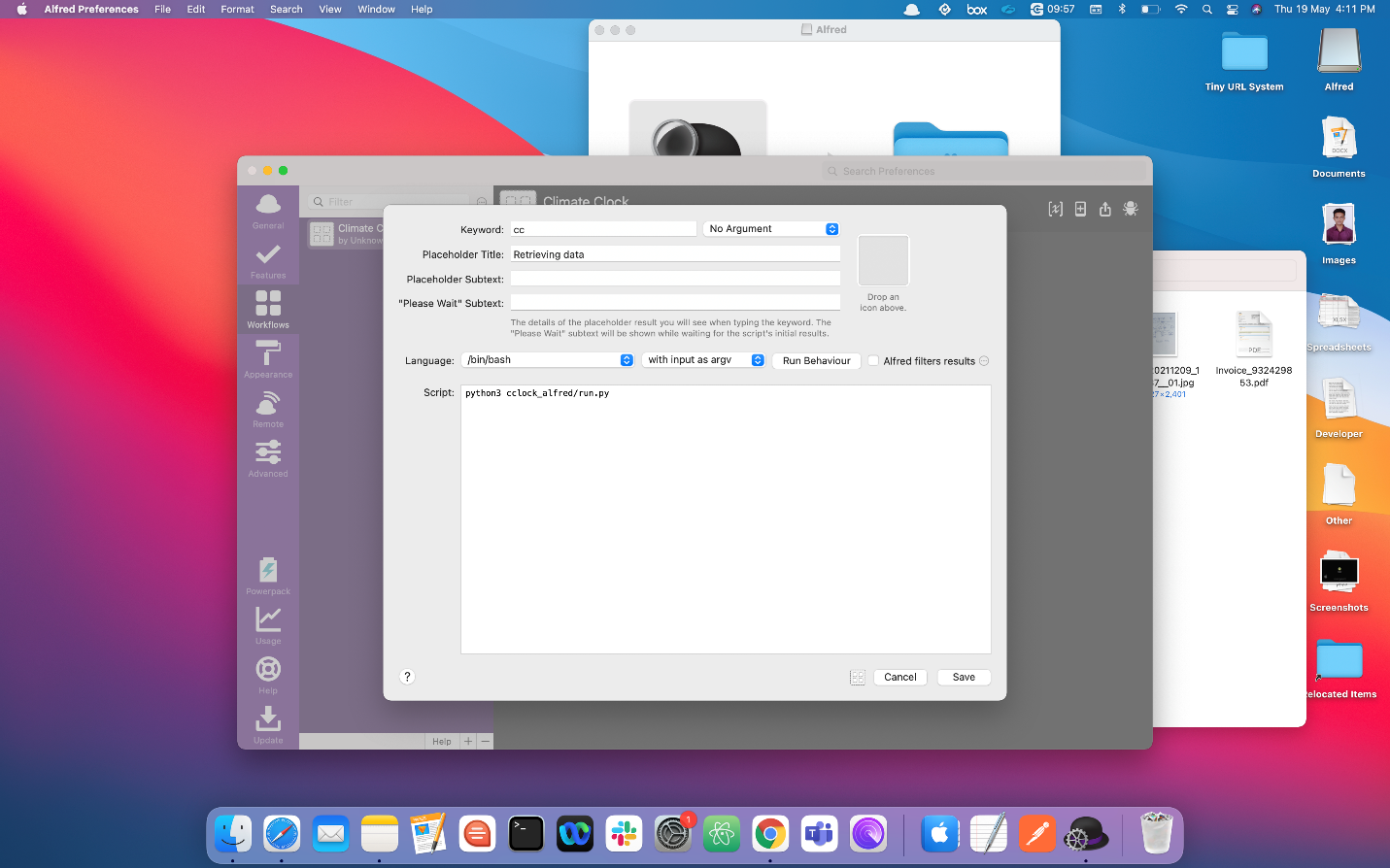
To create a script filter, right click on the blank area of the new workflow. Select Inputs and choose Script Filter. Right click on the script filter, select ‘Configure’.

Now you will be presented with a modal prompting you to fill out the fields

1. Keyword: This is the keyword that will be used to trigger the Alfred workflow. In our case, it is going to be ‘cc’.
2. Placeholder Title: The title that is displayed until the script filter is run. We can give something like ‘Retrieving data from Climateclock API’.
3. Placeholder Subtext, Please Wait Subtext are optional.
4. Script: The script to run when the workflow is activated. Before running the script, we need to place our python code in the current working directory. To do that, right click on the workflow. Go to ‘Open in Finder’. Place the code in the directory. Now, in the script section write ‘**python3 cclock\_alfred/run.py**’.

The code can be found in GitHub (<https://github.com/anudeepreddyv/climate_clock/tree/main/Climate%20Clock/cclock_alfred>).

1. Also, there are couple of other things that needs to be changed. For our workflow, arguments are not required. We need to change it to ‘Argument optional/No Argument’.

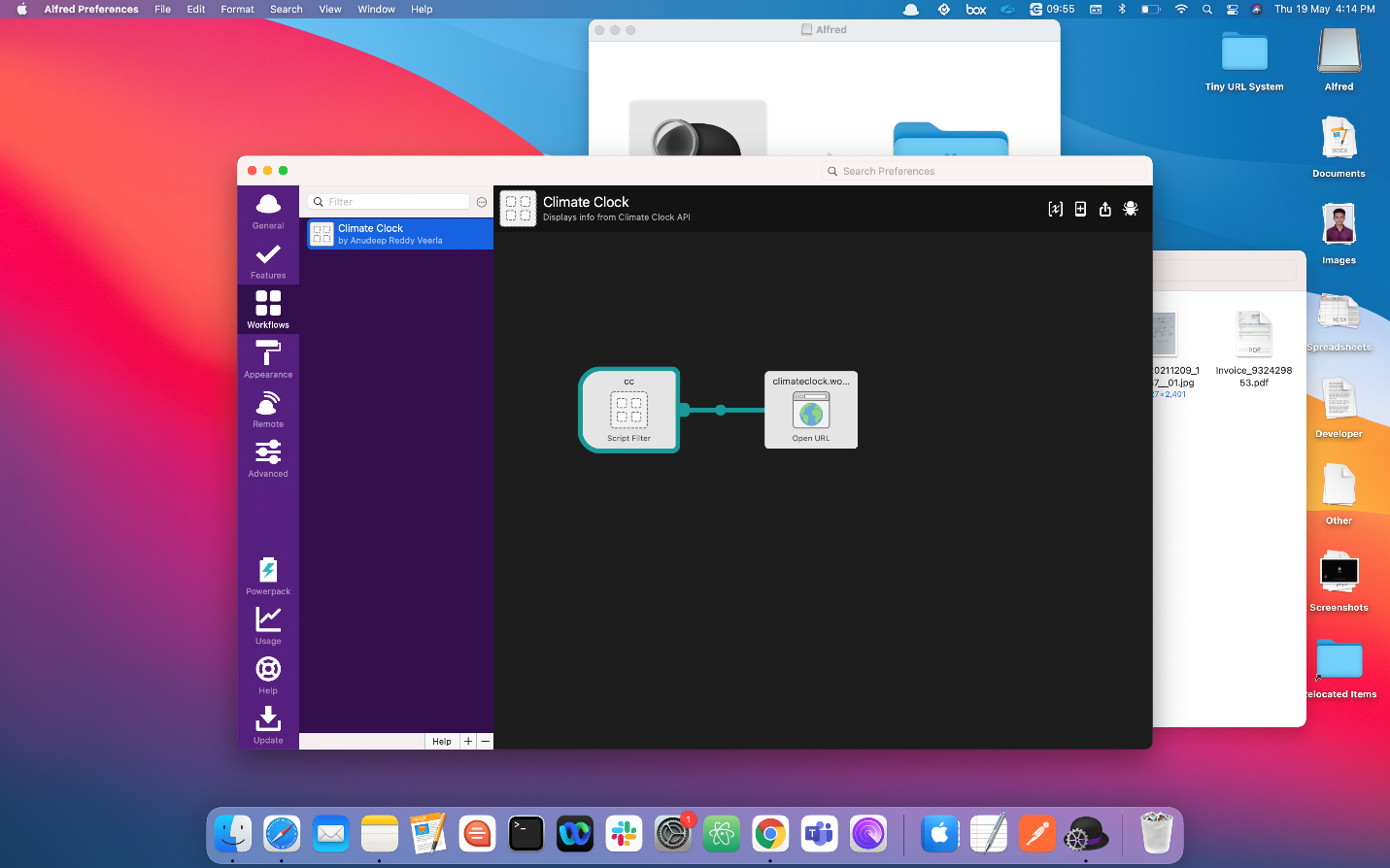
****

**Final Step**

We can also add an ‘Open URL’. For that, right click on the blank area, go to ‘Actions’ and select ‘Open URL’.

A modal pop-up will be opened and then we can give the URL as <https://www.climateclock.world>. Then, we need to link up the input script filter and the Open URL box.

When we run our script in Alfred, and by clicking on one of the rows of our output, it gets redirected to the climateclock website. This is the use of Open URL action. This step is optional.



Now that we have all the pieces connected, the Alfred part is finished. All we need to do is run the script by toggling Alfred using ‘cc’ command.