Universidad de Guadalajara

Ingeniería en computación

Computación Tolerante a Fallas 2023ª

“Workflow managers”

Olguín Hernández Jair Benjamín.

217439707

Un Workflow Manager, o Gestor de Flujo de Trabajo, es una herramienta o plataforma que ayuda a automatizar, coordinar y monitorear tareas y procesos dentro de una organización. Estos sistemas permiten la definición y ejecución de flujos de trabajo, los cuales son secuencias de tareas que se deben completar en un orden específico para lograr un objetivo. Los Workflow Managers pueden ser utilizados para automatizar procesos repetitivos y reducir el tiempo y los errores asociados con la realización de tareas manuales.

En nuestro ejemplo vamos a hacer extracción de la pagina dada por el profesor, “https://jsonplaceholder.cypress.io/todos”, donde simula ser información de algunos usuarios.

Donde tenemos que En la función transform, se pasará como parámetro la información obtenida de la extracción, la cual es una lista de diccionarios. Por cada objeto dentro de la lista, se crea un diccionario usando los valores de cada objeto, pero modificando las keys, y cada uno de estos nuevos diccionarios se agregan a una lista nueva y esta se retorna. En la función load, se pasará como parámetro la lista retornada por la función anterior y el archivo json en el que se guardará la información, se abrirá el archivo, se guardará la lista de diccionarios con ayuda de json.dump y se cerrará el archivo. Una vez completadas estas funciones, para convertirlas en un flujo de trabajo se necesita importar task y Flow de prefect, colocar los decoradores @task en la parte superior de cada función para convertirlas en una tarea, y después crear un flujo, en el que estarán todas las tareas a realizar. También se agrego un cronograma para que se ejecute el flujo cada determinado tiempo, en este caso dos minutos, esto con la ayuda de

IntervalSchedule de prefect.schedules. Este objeto schedule lo pasé como parámetro a la función donde se encuentra el flujo. Para ejecutar el flujo que nombré “flow” se realiza con flow.run. Para pasarle parámetros al flujo se necesita importar Parameter de prefect, en este caso pasé el link del cual obtendrá la información, y adentro del flujo se crea el parámetro con el mismo nombre que se le otorgó al pasar el parámetro a la hora de ejecutar el flujo.

Codigo:

```
import json
import requests
import datetime

from prefect import task, Flow, Parameter
from prefect.schedules import IntervalSchedule


@task(max_retries=10, retry_delay=datetime.timedelta(seconds=10))
def extract(url):
    resultado = requests.get(url)

    if not resultado:
        raise Exception('Datos no obtenidos.')

    return json.loads(resultado.content)

@task
def transform(datos):
    usuarios = []

    for usuario in datos:
        usuarios.append({
            'UsuarioID': usuario['userId'],
            'ID': usuario['id'],
            'Titulo': usuario['title'],
            'Completado': usuario['completed']
        })

    return usuarios
```

```python
@task
def load(datos, path):
    archivo = open(path, 'w')
    json.dump(datos, archivo, indent=5)
    archivo.close()


schedule = IntervalSchedule(interval=datetime.timedelta(minutes=2))

def prefect_flow(schedule):
    with Flow('etl_flow', schedule=schedule) as flow:
        url = Parameter(name='parametro_url', required=True)
        usuarios = extract(url)
        transformados = transform(usuarios)
        load(transformados, 'datos.json')

    return flow


flow = prefect_flow(schedule)
flow.visualize()
flow.run(parameters={
        'parametro_url': 'https://jsonplaceholder.cypress.io/todos'
    })
```

Resultados:

```
[2022-03-06 20:43:23-0600] INFO - prefect.etl_flow | Waiting for next scheduled run at 2022-03-07T02:44:00+00:00
[2022-03-06 20:44:00-0600] INFO - prefect.FlowRunner | Beginning Flow run for 'etl_flow'
[2022-03-06 20:44:00-0600] INFO - prefect.TaskRunner | Task 'parametro_url': Starting task run...
[2022-03-06 20:44:01-0600] INFO - prefect.TaskRunner | Task 'parametro_url': Finished task run for task with final state: 'Success'
[2022-03-06 20:44:01-0600] INFO - prefect.TaskRunner | Task 'extract': Starting task run...
[2022-03-06 20:44:01-0600] ERROR - prefect.TaskRunner | Task 'extract': Exception encountered during task execution!
Traceback (most recent call last):
  File "C:\Users\user\AppData\Local\Programs\Python\Python39\lib\site-packages\prefect\engine\task_runner.py", line 876, in get_task_run_st
ate
    value = prefect.utilities.executors.run_task_with_timeout(
  File "C:\Users\user\AppData\Local\Programs\Python\Python39\lib\site-packages\prefect\utilities\executors.py", line 468, in run_task_with_
timeout
    return task.run(*args, **kwargs)  # type: ignore
  File "c:\Users\user\Desktop\prefect\prefectflow.py", line 14, in extract
    raise Exception('Datos no obtenidos.')
Exception: Datos no obtenidos.
[2022-03-06 20:44:01-0600] INFO - prefect.TaskRunner | Task 'extract': Finished task run for task with final state: 'Retrying'
[2022-03-06 20:44:01-0600] INFO - prefect.TaskRunner | Task 'transform': Starting task run...
[2022-03-06 20:44:01-0600] INFO - prefect.TaskRunner | Task 'transform': Finished task run for task with final state: 'Pending'
[2022-03-06 20:44:01-0600] INFO - prefect.TaskRunner | Task 'load': Starting task run...
[2022-03-06 20:44:01-0600] INFO - prefect.TaskRunner | Task 'load': Finished task run for task with final state: 'Pending'
[2022-03-06 20:44:01-0600] INFO - prefect.FlowRunner | Flow run RUNNING: terminal tasks are incomplete.
[2022-03-06 20:44:01-0600] INFO - prefect.etl_flow | Waiting for next available Task run at 2022-03-07T02:44:11.781364+00:00
[2022-03-06 20:44:11-0600] INFO - prefect.FlowRunner | Beginning Flow run for 'etl_flow'
[2022-03-06 20:44:11-0600] INFO - prefect.TaskRunner | Task 'extract': Starting task run...
[2022-03-06 20:44:12-0600] ERROR - prefect.TaskRunner | Task 'extract': Exception encountered during task execution!
Traceback (most recent call last):
  File "C:\Users\user\AppData\Local\Programs\Python\Python39\lib\site-packages\prefect\engine\task_runner.py", line 876, in get_task_run_st
ate
    value = prefect.utilities.executors.run_task_with_timeout(
```

Por el contrario, si no hay ningún error, al ejecutar el flujo se muestra el momento en el que comienza y termina cada tarea (el flujo se vuelve a ejecutar cada dos minutos) y se carga la información obtenida en el archivo json:

```
[2022-03-06 20:33:23-0600] INFO - prefect.etl_flow | Waiting for next scheduled run at 2022-03-07T02:34:00+00:00
[2022-03-06 20:34:00-0600] INFO - prefect.FlowRunner | Beginning Flow run for 'etl_flow'
[2022-03-06 20:34:00-0600] INFO - prefect.TaskRunner | Task 'parametro_url': Starting task run...
[2022-03-06 20:34:01-0600] INFO - prefect.TaskRunner | Task 'parametro_url': Finished task run for task with final state: 'Success'
[2022-03-06 20:34:01-0600] INFO - prefect.TaskRunner | Task 'extract': Starting task run...
[2022-03-06 20:34:01-0600] INFO - prefect.TaskRunner | Task 'extract': Finished task run for task with final state: 'Success'
[2022-03-06 20:34:01-0600] INFO - prefect.TaskRunner | Task 'transform': Starting task run...
[2022-03-06 20:34:02-0600] INFO - prefect.TaskRunner | Task 'transform': Finished task run for task with final state: 'Success'
[2022-03-06 20:34:02-0600] INFO - prefect.TaskRunner | Task 'load': Starting task run...
[2022-03-06 20:34:02-0600] INFO - prefect.TaskRunner | Task 'load': Finished task run for task with final state: 'Success'
[2022-03-06 20:34:02-0600] INFO - prefect.FlowRunner | Flow run SUCCESS: all reference tasks succeeded
[2022-03-06 20:34:02-0600] INFO - prefect.etl_flow | Waiting for next scheduled run at 2022-03-07T02:36:00+00:00
[2022-03-06 20:36:00-0600] INFO - prefect.FlowRunner | Beginning Flow run for 'etl_flow'
[2022-03-06 20:36:00-0600] INFO - prefect.TaskRunner | Task 'parametro_url': Starting task run...
[2022-03-06 20:36:00-0600] INFO - prefect.TaskRunner | Task 'parametro_url': Finished task run for task with final state: 'Success'
[2022-03-06 20:36:00-0600] INFO - prefect.TaskRunner | Task 'extract': Starting task run...
[2022-03-06 20:36:00-0600] INFO - prefect.TaskRunner | Task 'extract': Finished task run for task with final state: 'Success'
[2022-03-06 20:36:00-0600] INFO - prefect.TaskRunner | Task 'transform': Starting task run...
[2022-03-06 20:36:00-0600] INFO - prefect.TaskRunner | Task 'transform': Finished task run for task with final state: 'Success'
[2022-03-06 20:36:00-0600] INFO - prefect.TaskRunner | Task 'load': Starting task run...
[2022-03-06 20:36:00-0600] INFO - prefect.TaskRunner | Task 'load': Finished task run for task with final state: 'Success'
[2022-03-06 20:36:00-0600] INFO - prefect.FlowRunner | Flow run SUCCESS: all reference tasks succeeded
```

Conclusión:

En conclusión, los gestores de flujo de trabajo son herramientas muy útiles en programación que permiten automatizar tareas y flujos de trabajo, lo que puede mejorar la eficiencia y la productividad de los equipos de desarrollo. Sin embargo, es importante tener en cuenta que los gestores de flujo de trabajo pueden aumentar la complejidad del código y requerir una curva de aprendizaje para su uso efectivo. Ya que en este caso desconocía sobre estos gestores y tuve que investigar para entenderlos.

Bibliografía:

Domínguez, V. H. M. (2016). *Los Sistemas Gestores de Flujos de Trabajo en la Gestión de Procesos Software* [Video]. https://www.redalyc.org/journal/5122/512253114009/html/#:~:text=Un%20Sistema%20Gestor%20de%20Flujos,cabo%20(Hollingsworth%2C%201995).

Prefect. (2020, April 17). *Getting Started with Prefect (PyData Denver)* [Video]. YouTube. https://www.youtube.com/watch?v=FETN0iivZps

PyData. (2019, January 4). *Task Failed Successfully - Jeremiah Lowin* [Video]. YouTube. https://www.youtube.com/watch?v=TlawR_gi8-Y