

Universidad de Guadalajara.

Computación Tolerante a fallas.



Diego Ivan Becerra Gonzalez.

Sección D06.

Objetivo.

Seguir el tutorial siguiente y ejecutarlo en tu máquina, después modificarlo y Generar un ejemplo.

Desarrollo.

Para esta práctica se utilizó la librería de prefect, la cual es una herramienta que nos permite conocer como es el flujo de los procesos en nuestro programa, es de gran utilidad ya que nos permite conocer el estado de diferentes tareas, la dependencia entre ellas y nos da información detallada de donde fallaron.

Para este trabajo se utilizo la API de coingecko, la cual nos da información sobre diversas criptomonedas.

La primer task se encargo de hacer una consulta a la API para obtener el precio de la moneda Dogecoin, se utilizo el decorador de cache para evitar hacer constantes llamadas al API.

```
## extract
@task(cache_for=datetime.timedelta(days=1))
def get_price():
    r = requests.get("https://api.coingecko.com/api/v3/simple/price?ids=dogecoin&vs_currencies=mxn")
    response_json = json.loads(r.text)
    return response_json
```

La segunda tarea hacia un tratamiento de los datos, extrayéndolos del objeto json.

```
@task
def parse_data(raw):
    price = raw['dogecoin']['mxn']
    return price
```

Finalmente, la tercer tarea lo que pretendía hacer es utilizar un bot para enviarnos un mensaje por telegram, un que nos indicara comprar cuando el precio del Dogecoin fuera menor a 1 peso mexicano, y otro que nos indicara vender si fuera mayor a 2.5.

```

## load
@task
def send_notificacion(parsed):
    token = "6191852587:AAEyCD59d37jirp0B2d0Zq-wMiTLBjCKjfo"
    url = f"https://api.telegram.org/bot{token}"
    if parsed < 1:
        params = {"chat_id": "5473415339", "text": "Compraaaa!!!" }
        r = requests.get(url + "/sendMessage", params=params)
    elif parsed > 2.5:
        params = {"chat_id": "5473415339", "text": "vende!!!" }
        r = requests.get(url + "/sendMessage", params=params)

```

Sin embargo, esto no fue posible por problemas de configuración del bot en la aplicación de telegram, por lo que no se logro hacer llegar los mensajes. A pesar de esto, la petición al API de telegram fue correcta, por lo que al ejecutar el programa no nos arrojaba errores.

Finalmente se agrego un temporizador que repitiera el flujo de trabajo cada 10 segundos, pero solo para fines de prueba.

```

schedule = IntervalSchedule(interval=datetime.timedelta(seconds=10))

with Flow("my etl flow",schedule) as f:
    raw = get_price()
    parsed = parse_data(raw)
    send_notificacion(parsed)

```

Aquí se muestra que corrió correctamente.

```

[2023-03-06 23:18:51-0600] INFO - prefect.my etl flow | Waiting for next scheduled run at 2023-03-07T05:19:00+00:00
[2023-03-06 23:19:00-0600] INFO - prefect.FlowRunner | Beginning Flow run for 'my etl flow'
[2023-03-06 23:19:00-0600] INFO - prefect.TaskRunner | Task 'get_price': Starting task run...
[2023-03-06 23:19:00-0600] WARNING - prefect.TaskRunner | Task 'get_price': Can't use cache because it is now invalid
[2023-03-06 23:19:01-0600] INFO - prefect.TaskRunner | Task 'get_price': Finished task run for task with final state: 'Cached'
[2023-03-06 23:19:01-0600] INFO - prefect.TaskRunner | Task 'parse_data': Starting task run...
[2023-03-06 23:19:01-0600] INFO - prefect.TaskRunner | Task 'parse_data': Finished task run for task with final state: 'Success'
[2023-03-06 23:19:01-0600] INFO - prefect.TaskRunner | Task 'send_notificacion': Starting task run...
[2023-03-06 23:19:01-0600] INFO - prefect.TaskRunner | Task 'send_notificacion': Finished task run for task with final state: 'Success'
[2023-03-06 23:19:01-0600] INFO - prefect.FlowRunner | Flow run SUCCESS: all reference tasks succeeded
[2023-03-06 23:19:01-0600] INFO - prefect.my etl flow | Waiting for next scheduled run at 2023-03-07T05:19:10+00:00

```

Conclusión.

Prefect es una herramienta muy completa que nos permite conocer cómo se interrelacionan nuestras tareas, ejecutarlas como si se tratara de un servicio y

conocer a detalle porque fallaron, sin embargo, en proyectos demostrativos como esta actividad, no se puede decir que se aprovechó completamente.