

Python:

Ensure you have Python 3.7+ installed. Install the following libraries aiohttp, asyncpg, temporalio and sendgrid

Command

```
pip install aiohttp asyncpg temporalio sendgrid
```

PostgreSQL:

Ensure you have a PostgreSQL database named temporal and a table news_articles with the following schema:

```
CREATE TABLE news_articles ( id SERIAL PRIMARY KEY, title TEXT, published_at  
TIMESTAMPTZ, description TEXT, content TEXT);
```

SendGrid Account:

Sign up for a SendGrid account and obtain an API key.

Code process:

activities.py :

Fetches news articles from the News API.

workflows.py:

Workflow that triggers the fetch news activity and returns the result.

run_workflows.py:

Contains the functions for insert, fetch, delete, and update articles in the database and send the articles via email notification using the send grid library.

insert_articles(articles[0:10]) this inserts only 10 articles

delete_all_articles() this delete all the rows from the news_articles table

update_article() this updates the content based on the id of the article

fetch_articles() this retrieves all the articles from the news_articles table

send_email(subject, content) this sends an email using send grid library with the fetched articles. It takes the subject of the mail and content as the argument.

run_worker.py:

This executes the code of workflows and activity

Brief Explanation

First I got the understanding how temporal work. I was trying to run the quick start code but was it was not running then I solved the problem by running the temporal server using docker. Once the say hello code got successfully executed. I gone through the complete code but it was going all over my head so, I found they have listed some banker example code and community post.

After reading the codes thoroughly and it's explanation I got a hand how the structure is and where should I put my code blocks so that it can run efficiently. I created a branch from main as version_0.1 in this I just fetched the raw data and printed it then I formatted it and got the things I want which were title, published_at, description and content. Once the formatting was in place I also tried printing it just to make sure everything was correct and then pushed the code and created a pull request and merge it with main.

Now I checked out on the version_0.1 to main then created a new branch version_0.2 in which I wanted the add the SQL operation part I created the function one by one and tested them they all worked perfectly as I wanted. Then I committed the code and pushed to create a pull request then merged it with main.

Again I did checkout to main and created new branch version_0.3 I which I wanted to add the send email notification so I found I needed a API key for that but the send grid was not allowing me to sign up. So I added the code for sending the email notification but not tested it. I have the email body set in html format for better looking. I have pushed and created a pull request for version_0.3 and committed the code. When I have the API key I will test the code again then merge with the main branch.