RSS Scraper

Scrapping Really Simple Syndication feeds

**22 September 2023**

**Author: Maor Avitan**

# 

# Overview

This home assignment is a solution for RSS Scraper app that provides users the ability to subscribe to multiple feed URLs and retrieve feed items. the server will scrape the feed on-demand and automatically at regular intervals, adding items to the database to ensure that the item list is always up-to-date

# Technologies

Technology Stack for this App:

Server Side: FastAPI, Python 3

Database: SQLite

Testing: Pytest

# API list

# **@app.post("/login/")** Can use with two users: user1@sendcloud.com, user2@sendcloud.com Example: post call {"email": "[user1@sendcloud.com](mailto:user1@sendcloud.com)"} Response: {"access\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InVzZXIxQHNlbmRjbG91ZC5jb20iLCJpZCI6MX0.9Yz0kysK\_OuWdyIceBdDFVEV5CW4fMY3tQ-NB8XXMJ4","token\_type": "bearer"}

# **@app.post("/feeds/")** Add new feed related to user, user id fetched from token Example: post call {"url": "<http://www.nu.nl/rss/Algemeen>"}

Response:  
 {"message": "Subscribed to feed 'http://www.nu.nl/rss/Algemeen' successfully."}

# **@app.put("/follow-feeds")** Follow or unfollow list of feeds Example: put call [{"id": 1,"follow": **false**},{"id": 2,"follow": **true**}]

Response:

{"message": "Feed updated successfully"}

# **@app.get("/feeds/")** Get all user’s feed Response: [{"id": 1,"url": "http://www.nu.nl/rss/Algemeen","follow": **true**,"status": "","failed\_cnt": 0,"sync": **true**}] **@app.put("/{feed\_id}/force-sync/”)** Force feed to sync scrapping on demand Response: {"message": "30 items added to the feed"} **@app.get("/{feed\_id}/items")** Getting all feed items of the user, optional query params:unread\_only,read\_only, default all

[{"id": 4,"unread": **true**,"created\_time": "2023-09-22 07:46:12",

"url": "<https://www.nu.nl/binnenland/6282003/lastig-te-blussen-brand-bij-afvalwerker-in-rotterdam-woedt-al-24-uur.html>",

"title": "Lastig te blussen brand bij afvalwerker in Rotterdam woedt al 24 uur",

"description": "Bij afvalverwerker AVR in de Botlek ten westen van Rotterdam woedt al de hele nacht een brand die maar moeilijk onder controle te krijgen is. De brand ontstond gisterochtend al, maar laaide 's avonds weer op. De brandweer verwacht nog de hele dag met bluswerkzaamheden bezig te zijn.",

"feed\_id": 1

},

{"id": 8,"unread": **true**,"created\_time": "2023-09-22 07:25:59",

"url": "<https://www.nu.nl/tech/6281978/james-webb-telescoop-vindt-essentieel-onderdeel-van-leven-op-maan-jupiter.html>",

"title": "James Webb-telescoop vindt 'essentieel onderdeel van leven' op maan Jupiter",

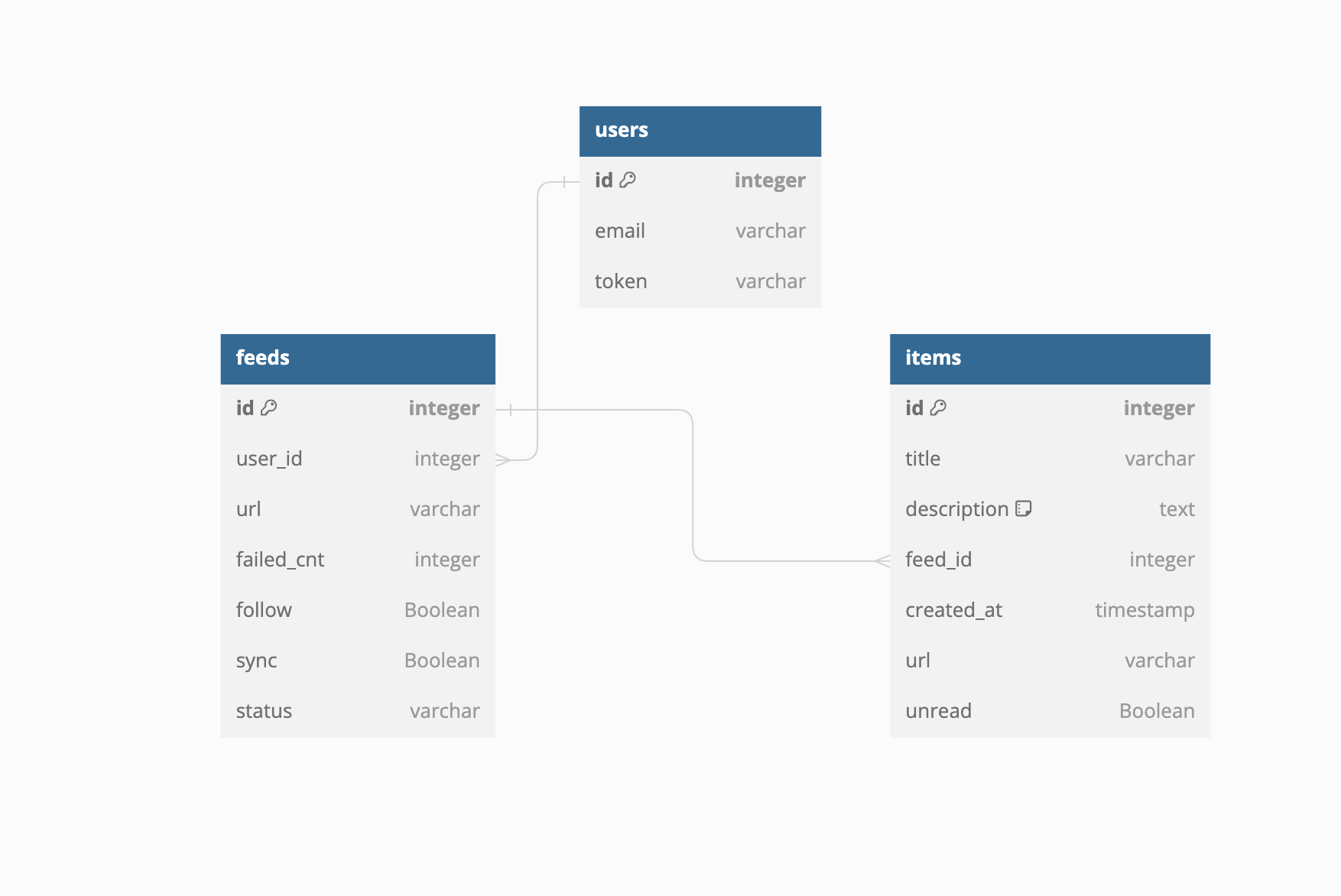
"description": "Onderzoekers zijn enthousiast over een vondst op de maan Europa, een van de manen van Jupiter. De James Webb-telescoop vond daar koolstof, een belangrijke voorwaarde voor leven, schrijft de Europese ruimtevaartorganisatie <a href=\"https://www.esa.int/Science\_Exploration/Space\_Science/Webb/Webb\_finds\_carbon\_source\_on\_surface\_of\_Jupiter\_s\_moon\_Europa\" target=\"\_blank\">ESA</a>.",

"feed\_id": 1},

**@app.put("/read-item/{item\_id}"**Mark item as read  
 Response:  
 {"message": "Item updated successfully"}

# 

# Database schema



# Scrapping feeds

We used the feedparser library to perform web scraping, extracting structured data from web feeds and inserting it into the Items table. This process is run asynchronously in a separate thread to ensure it doesn't impact server performance

schedule.every(1).hour.do(scrape\_feeds)

In case of failures multiple times feed will be set as not sync anymore

# Libraries

SQLAlchemy**~=**1.4.49

PyJWT**~=**2.8.0

fastapi**~=**0.103.1

starlette**~=**0.27.0

schedule**~=**1.2.0

uvicorn**~=**0.23.2

pydantic**~=**2.3.0

pytest**~=**7.4.2

dramatiq**~=**1.14.2

feedparser**~=**6.0.10

redis**==**3.5.3