Viewstats Backend Eng Coding Assignment

ABOUT PROJECT

Your task is to build an application that checks whether a video is playable on youtubekids.com, given the video's ID. To do so, it should consume messages from RabbitMQ, send an HTTP request (or multiple if needed) to youtubekids.com, parse the result, then write the result to a PostgreSQL database. In addition, the application should provide some API endpoints itself.

You can test if a video is playable on YouTube kids by placing it's video ID in the URL and seeing if the video player shows up. The application is supposed to do the same - but based on raw HTTP request(s) and in an unattended fashion.

Provided by us:

- Schema of the RabbitMQ messages
- Database schema
- A server with RabbitMQ and PostgreSQL pre-installed, along with some sample data + all credentials needed
- A set of test data (in database on server)

Services and ports:

- Your IP address has been sent to you along with this document
- SSH
  - Username:
  - Password:
- RabbitMQ management interface:
  - Username:
  - Password:
- RabbitMQ port:
  - (Same credentials as the management interface)
- PostgreSQL
  - Port:
  - Username:
  - Password:
- PGAdmin management interface:
  - Email:
  - Password:

Technology:

- Programming language: Java only
- Libraries: Any of your choice
    - Hint: You can but do not have to use an ORM framework
- Package manager: Maven is preferred
- Please stick to sending raw HTTP requests yourself; do not run a full browser with remote debugging (like Puppeteer, Selenium), or a headless browser (with Playwright)
- The application should be built in a way that scaling it (running multiple instances) is simple and does no require huge resources
- Hint: Use a debug proxy like HTTP Toolkit, Burp or Zap; or Firefox, as it can modify and replay HTTP requests (Chrome can just replay, not modify)

Expected result:

- An application that does the following:
    - Receives messages from RabbitMQ
    - Determines whether a YT video is currently playable on YT kids (as in, the player shows up, or an error message does)
        - *Hint: youtubekids.com*
    - If the video is either newly present on YT kids or is no longer available on YT kids (ie if the YT kids status changes)
        - Update the status on the videos table
        - Create an entry on the video_events table
    - APIs to provide:
        - GET /videos/:id/checkKids
            - Checks if a video is currently available on YouTube kids. This endpoint does not need to interact with database; simply returning true or false is fine
        - GET /videos/:id/events
            - Retrieves a video from the database and all of the video_events attached to it (newest first)
        - The API should be protected by authorization using a Bearer token. You can generate one yourself
- Expect failure on every level (namely network failure, rate limits and RabbitMQ disconnects). Have your application deal with those errors accordingly; no message from RabbitMQ should get lost if an error occurs or the application crashes or gets killed halfway through

Submission:

DUE DATE: 72 hours after receiving the project.

Grading Rubric:

Criteria	Excellent (5)	Good (4)	Satisfactory (3)	Needs Improvement (1-2)	Unacceptable (0)
Scraping HTTP request(s)	Sends only the request(s) needed, has the request reduced to only what's needed	Sends only the request(s) needed	Sends more requests than needed, but the desired data is present somewhere	Requests have no error handling, application crashes if response is unexpected	Used a browser debugging framework like Selenium
RabbitMQ	Will not drop messages unless submitted to database, reconnects on network failure	Will not drop messages unless submitted to database	Will execute messages fine in normal operation	Major implementation issues like using some features wrong	Does not connect to RabbitMQ
Java and frameworks	Used a modern version of Java and modern framework versions	Used outdated versions of libraries (patch versions are fine)	Used libraries that are barely contribute to the solution	Used libraries that don't contribute to the solution	Different language (including Kotlin and Scala); Major implementation issues like parsing Java with String.split and .substring
API endpoints	API does exactly what's specified and does not fail when	API does exactly what's specified, but headers are not	API errors out a lot (upstream errors are of course fine)	API has additional endpoints (like default management or admin functionality)	API is not functional

	requests are (slightly) malformed	correct			
Database	Uses PostgreSQL, follows SQL best practises, uses good data types	Uses PostgreSQL, mostly follows SQL best practises	Minor implementation issues	Major implementation issues, including those introduced by poor framework selection	Does not use a database or uses a different database