



Backend Developer Challenge

Write a RESTful API that downloads files from a given URL

You are given a URL, and your task is to download the file served by the URL.
(Usage of requests package is allowed)

ENDPOINTS

POST /download?url=<input>

Where input is the URL to the file you want to download.
The response should return a unique ID that can be used to track the progress of the download.

GET /status?id=<input>

Where input is the ID returned from a POST request to /download endpoint. The response should contain all important details like

1. Total file size
2. Downloaded file size
3. Remaining file size
4. Status
5. Estimated time to complete

CONSTRAINTS

1. The downloader should be multithreaded (parallel threads, processes, etc)
2. File writing and download workers should be independently scalable (X download workers, Y file writers), using queues is allowed.
3. The API should be rate limited.
4. The flask application should be containerized with an application server of your choice, reverse proxies by a web server (nginx, apache, etc)

REQUIREMENTS

1. Please host a public code repository with gradual and meaningful commits.
2. Please add basic documentation explaining your approach and the design.
3. Please host this in a publicly reachable location like heroku.
4. Databases should be persistent (i.e not file based and not a part of your container for services like heroku)

Bonus:

1. Bonus points for an asyncio implementation (with an optional ASGI server setup).
2. Bonus points for a POST /control?id=<input> endpoint that can be used to pause, resume or stop a download.
3. Bonus points for test coverage.
4. Bonus points for a GET /file?id=<input> endpoint to serve the downloaded files.
5. Bonus points for a POST /upload endpoint with multipart upload support.
6. Bonus points for a UI

SUBMISSION

Code has to be submitted in a public code repository, with relevant configurations for your tech stack and a dockerfile