HTTP endpoints monitoring service

Snapp Team

November 4, 2021

Contents

1 Introduction

Design and implement a service with Golang programming language to monitor HTTP endpoints so that in some configurable periods (e.g., the 30s, 1m, 5m) this service sends HTTP requests to the endpoint and logs the response status code and registers an alert for that URL. Each URL must have an error threshold that indicates the maximum error count after which the service must trigger an alert for the user who the URL belongs to.

Successful HTTP call is indicated by an HTTP OK 2XX status code and the failed call is indicated by a non-2XX status code.

2 Database

Feel free to use any kind of database you want.

3 Design and develop a RESTFul API with the following features

3.1 Authentication

Use JWT Tokens for authenticating users

3.2 Users

This application needs simple user management and authentication so that users can register and receive a token in order to be able to use other endpoints to create URL, update URL, get the stats of the URLs and also get alerts for their URLs.

3.3 Endpoints

- Create a new user (public endpoint)

- Generate Token for the user (public endpoint)

URLs Each URL must have a threshold parameter that indicates the maximum error count which service must trigger and create an alert after that threshold.

Each user can create up to 20 URLs.

URL Endpoints Create a new URL (auth-required) List URLs by the user (auth-required) Get statistics for a specific URL by day (successes and failures). For example, count of successful call and failed call during today (auth-required)

Alerts Each time the number of errors exceeds the error threshold service must register a new alert for that URL. Users must be able to get the alerts for their URLs.

Alert Endpoint Get Alerts for a specific URL (auth-required)

Notice: The public endpoints are accessible by everyone and auth-required endpoints need the token for authentication

4 Instructions

Please create a GitHub repository and share it with us as soon as you started working on this assignment.

- Tell us about the Trade-Offs and the important decisions which you made during the design and development.

- Provide docker-compose file so we can easily run and see how your application works

- Use concise and good commit messages

- Design the API with RESTful standards

- We don't need 100% test coverage but do your best to write some tests for important parts of the application

- Write some documentation about your service in a file called README.md

After you finished working on the assignment please send us an email with the following information:

How much time you put on this assignment What did you learn by working on this assignment

Please be prepared to answer the follow-up question and don't hesitate to ask any questions you may have.

Good Luck! Snapp! Team