Developer Take Home Interview

Problem

We are building a service that sync GitHub issues to 3rd-party services. While GitHub has a Webhook that notifies our service when an issue is opened/closed, it does not notify us when the title and description of an issue is updated.

Objective

Your objective is to write a simple service (server) that simulates the GitHub Webhook. In addition to relaying events already supported by GitHub Webhook, your service will also **simulate an event when the title and description of an issue is updated**, which is not supported by the original Github Webhook.

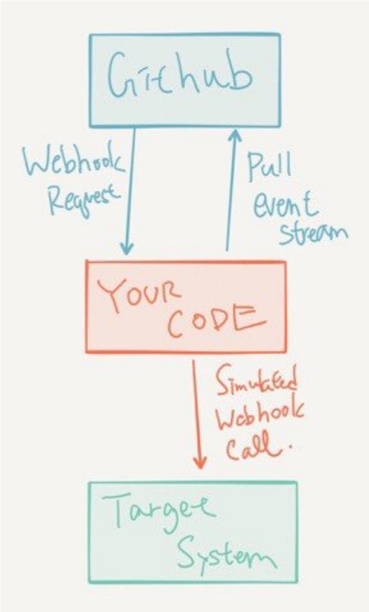
Overview

Your simple service will accept all webhook events from GitHub and relay the events to a 3rd-party Target System. You may assume the webhook is registered at GitHub manually.

It will:

- Accept and replay the request from GitHub.
- Detect when issue description or title is updated. Simulate GitHub Webhook event when issue title and description is updated, by sending a webhook to the Target System.
- Able to work with **multiple** repos.

The diagram below illustrates the data flow:



Tasks

1. Understand the related GitHub API and Webhook.
2. Design a mechanism that will observe the GitHub API and accept events from GitHub Webhook. Try to make it **efficient**.
3. Implement the basic relay of GitHub Webhook to Target System
4. Implement the issue title / description update Webhook to Target System.

The format of simulated issue update webhook should be similar to the format of the issue create webhook, with the ‘`action`’ being ‘`updated`’. Example is as follows:

```
{
  ''action'': ''updated'',
  ''issue'': {
    ''url'': ''https://api.github.com/repos/oursky/hooksim/issues/1347'',
    ''number'': 1347,
    ...
  },
  ''repository'' : {
    ''id'': 1296269,
    ''full_name'': ''oursky/hooksim'',
    ''owner'': {
      ''login'': ''chpapa'',
      ''id'': 1445,
      ...
    },
    ...
  },
  ''sender'': {
    ''login'': ''rickmak'',
    ''id'': 123,
    ...
  }
}
```

Tips

- How we know when the title and description is updated?

    - Observe the issue event stream: https://developer.github.com/v3/issues/events/.
    - Query the `GET /repos/:owner/:repo/issues` API: https://developer.github.com/v3/issues/.
- Do we need to have local store? What to store?
- When to pull the event stream?
- What fault tolerance level do we need?

Other requirements

- Use your choice of technology stack
- Use any open sources libraries
- Please init the project with git and submit the exercise with meaningful git history
- There is no time limit. Just tell us how much time you spent. (We expect around 4 hours of work)

How we evaluate

- We look for clean code
- We look for maintainability
- We look for atomic commit
- We will ask for reason why the code is structured in a specific way