Building GitHub integration to automate branch protection for new repositories with webhooks and REST

Yoseph Buitrago
06/04/2022

Agenda

1. Introduction to REST APIs
2. Introduction to webhooks
3. Webhooks best practices
4. Demo
5. Q&A

REST APIs

It is a way of programmatically getting data from a server. In a request/response lifecycle when using HTTP, e.g.

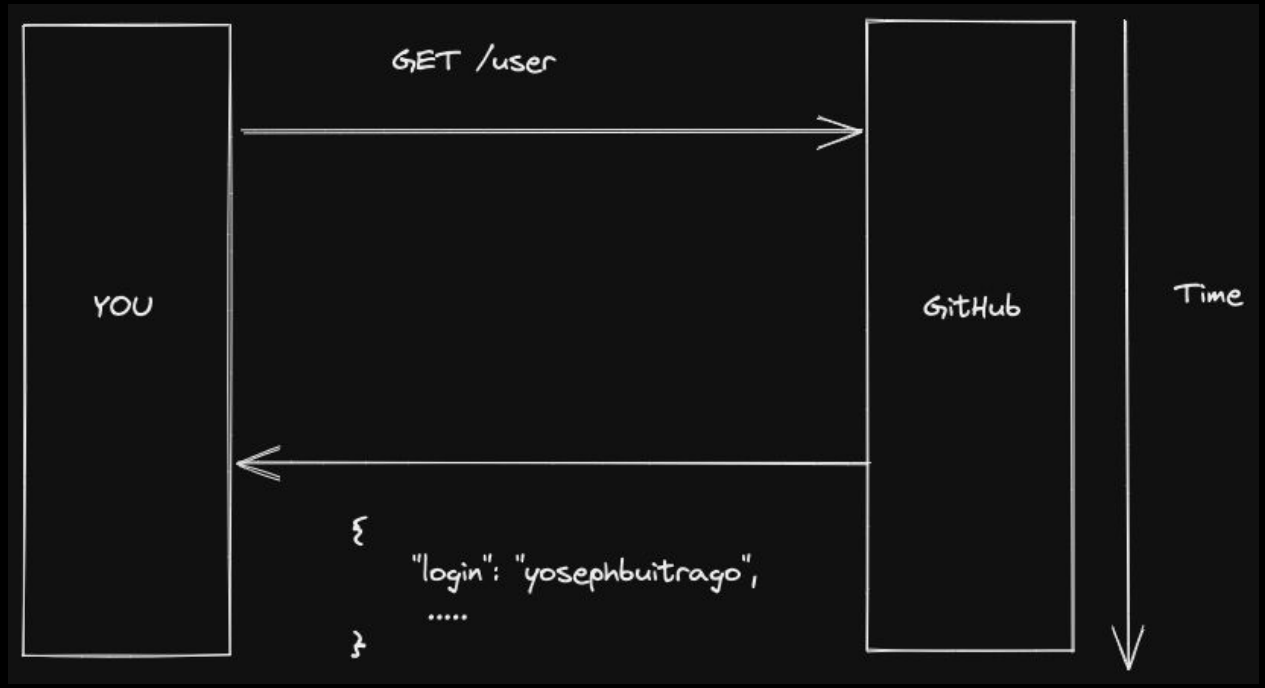
GET https://api.github.com/user

Compose of two parts:

- GET: verb operation
- Endpoint: https://api.github.com/user

More info: https://docs.github.com/en/rest

PULL Model



curl -X GET http://api.github.com/user

Client libraries: python

```python
from github import Github
# First create a Github instance:
# using an access token
g = Github("access_token")

# Github Enterprise with custom hostname
g = Github(base_url="https://{hostname}/api/v3", login_or_token="access_token")

# Then play with your Github objects:
print(g.get_user().login)
```

List of libraries: https://docs.github.com/en/rest/overview/libraries

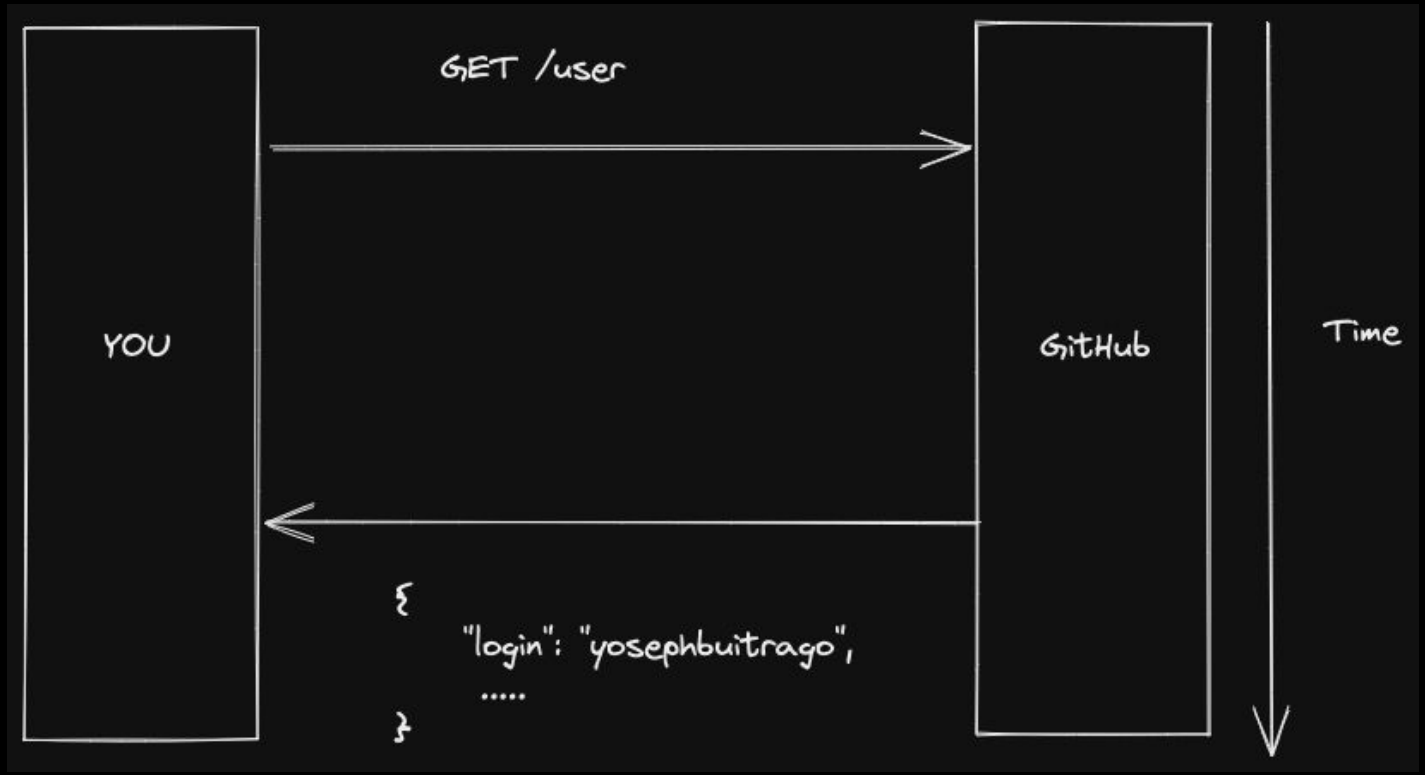
Limitations of PULL model

Is there a better solution to avoid these limitations of a PULL model?

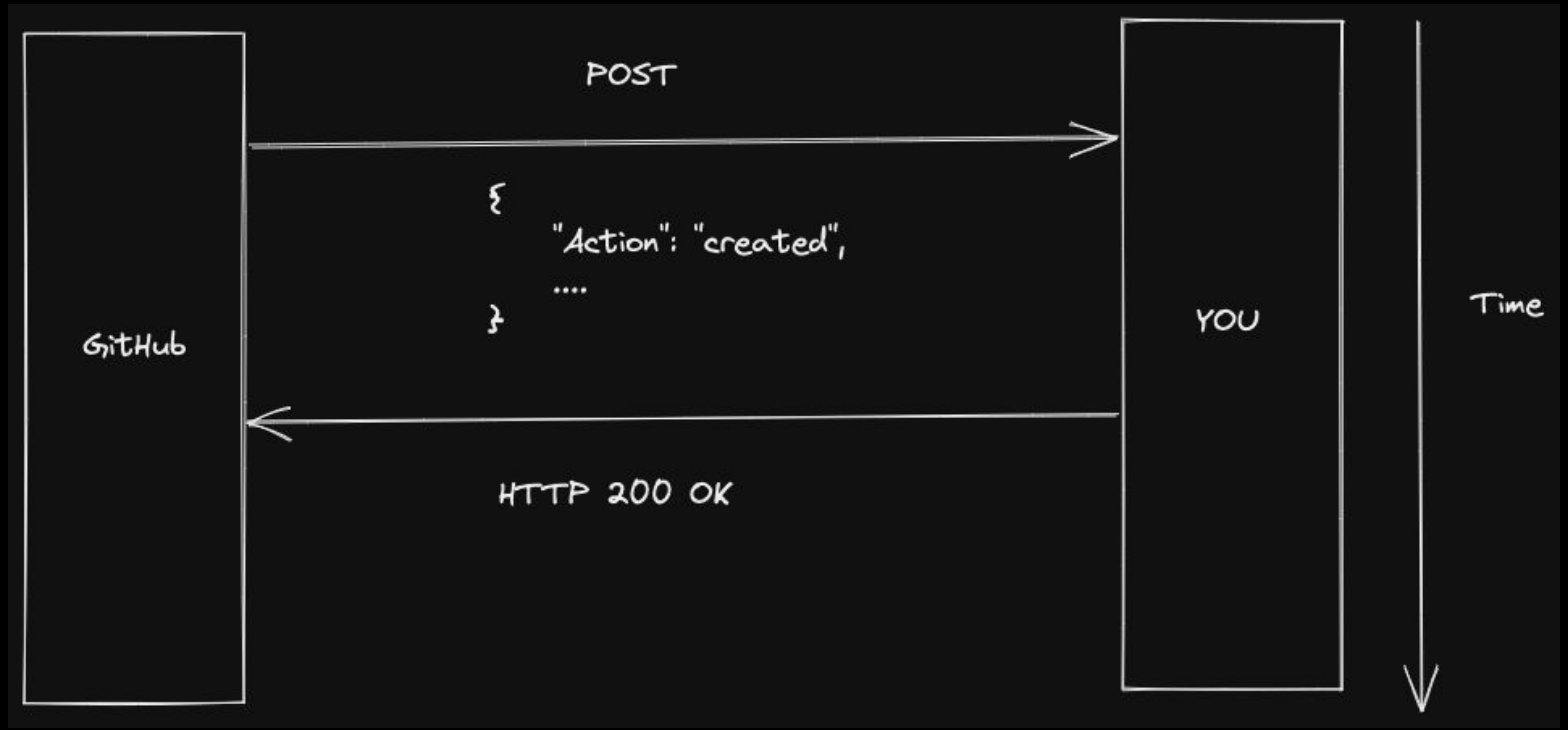
Webhooks

Instead of fetching the data GitHub will send a notification that an event has happened in a repository.

PULL model



Event notification model



Structure of webhook event

```
Request method: POST
Accept: */*
content-type: application/json
User-Agent: GitHub-Hookshot/ae56f53 #(the server sending the request.)
X-GitHub-Delivery: 8b3ee140-b2c8-11ec-902a-a625b65cf98e #(unique event identifier)
X-GitHub-Event: repository #(type of event)
X-GitHub-Hook-Installation-Target-Type: organization

{
  "action": "created",
  "repository": {
    "name": "repo2",
    "full_name": "YosephLab/repo2",
    ...
  },
  "organization": {
    "login": "YosephLab",
    ...
  }
  "sender": {
    "login": "yosephbuitrago",
    ...
  }
}
```

Installation targets for webhooks

- Organization level
- Repository
- Github apps
- …

Webhooks best practices

- Resilience:
  - Try to response to event in less that 10 seconds
  - Handle ordering of events (GitHub does best effort)
- Security
  - Verify authenticity of the events using HMAC signatures
  - IP whitelist *https://api.github.com/meta*
- HTTP
  - Use sensible HTTP status code for better debugging
    - 2XX: Success
    - 4XX: Client error
    - 5XX: Server error

Demo

Q&A