DOCUMENTATION

Set B

Code

import asyncio  
import os  
import aiohttp  
  
from gidgethub.aiohttp import GitHubAPI  
  
async def main():  
 async with aiohttp.ClientSession() as session:  
 gh = GitHubAPI(session, "joker\_bot", oauth\_token=os.getenv("joker"))  
 await gh.post("/repos/sohanjokerinfini/mayadata/pulls",  
 data={"title": "hello github bot ",  
 "body" : "Thanks!"})  
  
asyncio.run(main())

this code I tried to auto comment “thanks” after creating the new pull request

before pull request I tried with issue that worked in mayadata respitory

await gh.post("/repos/sohanjokerinfini/mayadata/pulls/110/merge",

now while including web service , I tried with opening a issue ,it was running in  [http://127.0.0.1:8080](http://127.0.0.1:8080/)

I connected my github with Heroku and tried to run but their was some problem with the git and it was not runing

import os  
import aiohttp  
  
from aiohttp import web  
  
from gidgethub import routing, sansio  
from gidgethub import aiohttp as gh\_aiohttp  
  
routes = web.RouteTableDef()  
  
router = routing.Router()  
  
@router.register("issues", action="opened")  
async def issue\_opened\_event(event, gh, \*args, \*\*kwargs):  
 *"""  
 Whenever an issue is opened, greet the author and say thanks.  
 """* url = event.data["issue"]["comments\_url"]  
 author = event.data["issue"]["user"]["login"]  
  
 message = f"Thanks for the report @{author}! I will look into it ASAP! (I'm a bot)."  
 await gh.post(url, data={"body": message})  
  
@routes.post("/")  
async def main(request):  
 body = await request.read()  
  
 secret = os.environ.get("GH\_SECRET")  
 oauth\_token = os.environ.get("joker")  
  
 event = sansio.Event.from\_http(request.headers, body, secret=secret)  
 async with aiohttp.ClientSession() as session:  
 gh = gh\_aiohttp.GitHubAPI(session, "sohan",  
 oauth\_token=oauth\_token)  
 await router.dispatch(event, gh)  
 return web.Response(status=200)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app = web.Application()  
 app.add\_routes(routes)  
 port = os.environ.get("PORT")  
 if port is not None:  
 port = int(port)  
  
 web.run\_app(app, port=port)