

# CLICKUP BOT

AI-powered task management automation

---

Python · Flask · Groq AI · GitHub API · ClickUp API

Projektna dokumentacija

**Pavle Kapuran**

[pavlekapuran.pages.dev](https://pavlekapuran.pages.dev)

# // sadrzaj

- 01 Opis projekta
- 02 Funkcionalnosti
- 03 Tech Stack
- 04 Arhitektura projekta
- 05 Struktura projekta
- 06 Flow podataka
- 07 API endpointi
- 08 Autentifikacija i sigurnost
- 09 Konfiguracija okruzenja

## // opis projekta

ClickUp Bot je interni alat za automatizaciju koji povezuje GitHub repozitorije sa ClickUp projektним menadžment sistemom. Bot funkcioniše kao webhook server koji sluša GitHub push event-e, automatski analizira promjene koda koristeći AI (Groq LLaMA 3.3 70B model), i generiše strukturirane sazetke na srpskom jeziku koje potom postavlja kao komentare na odgovarajuće ClickUp taskove.

Osnovni cilj projekta je ubrzavanje komunikacije unutar razvojnog tima. Umjesto da project manageri i ostali članovi tima moraju citati git logove ili pregledati diff-ove, bot automatski generiše citljive, AI-analizirane sazetke promjena direktno na ClickUp karticama koje tim već koristi za praćenje zadataka.

### Glavni tokovi korištenja

1. Developer push-a kod na GitHub repozitorij (branch ime sadrži ClickUp Task ID)
2. GitHub šalje webhook payload na Flask server
3. Bot parsira payload, dohvaca detalje commita
4. Commit podaci se šalju Groq AI na analizu
5. AI generiše strukturiran sažetak na srpskom
6. Bot ekstrahuje Task ID iz branch-a ili poruke
7. Formatirani sažetak se postavlja na ClickUp task

## // funkcionalnosti

### Webhook prijemnik

Flask server izložen na /webhook endpoint-u prima POST zahtjeve od GitHub-a. Server validira da je payload push event (provjerava prisustvo "commits" ključa) i obradjuje svaki commit pojedinačno. Iz payload-a se ekstrahuju informacije o repozitoriju, branch-u i pusheru.

### GitHub integracija

Koristi PyGithub biblioteku za dohvatanje detaljnih informacija o svakom commitu. Za svaki commit prikuplja: SHA hash, commit poruku, autora, datum, listu izmijenjenih fajlova sa statusom (added/modified/removed), statistike (dodato/obrisano linija) i patch sadržaj za male promjene (< 500 znakova). Ograničava prikaz na maksimalno 10 fajlova i patch za prva 3 fajla.

## AI analiza koda

Commit podaci se salju Groq API-ju koji koristi LLaMA 3.3 70B Versatile model. AI generise sazetak na srpskom jeziku koji ukljucuje: kratak opis (2 recenice), listu izmijenjenih fajlova sa objasnjenjima, nove/izmijenjene funkcionalnosti sa imenima metoda, i naglasene bitne promjene za tim. Temperature je 0.3 za preciznost, max tokena 800.

## Task ID ekstrakcija

Bot automatski ekstraktuje ClickUp Task ID iz branch imena ili commit poruke koristeci regex pattern-e. Podrzava format sa slash-om (feature/86c74hx82-opis) i direktni format (86c74hx82-opis). Task ID je 9-znakovni alfanumericki string.

## ClickUp integracija

Nakon ekstrakcije Task ID-a, bot provjerava postojanje taska putem ClickUp API-a i postavlja formatiran komentar sa AI sazetkom, statistikama i metapodacima commita.

## Pregled funkcionalnosti

Funkcionalnost	Opis	Eksterni servis
Webhook prijemnik	Prima GitHub push event-e	GitHub Webhooks
Commit analiza	Dohvaca detalje (fajlovi, patch, stats)	GitHub API
AI sazetak	Generise sazetak na srpskom	Groq API (LLaMA 3.3)
Task ID ekstrakcija	Regex pretraga branch-a i poruke	Interno
ClickUp posting	Postavlja komentar na task	ClickUp API v2
Health check	Testira konekcije svih servisa	Svi

# // tech stack

## Backend

Tehnologija	Verzija	Namjena
Python	3.x	Primarni programski jezik
Flask	3.0.0	Web framework za webhook server
Gunicorn	21.2.0	WSGI HTTP server za produkciju

## Biblioteke i SDK-ovi

Biblioteka	Verzija	Namjena
PyGithub	2.1.1	GitHub API klijent za commit detalje
groq	Latest	Groq AI SDK za LLaMA 3.3 model
requests	2.31.0	HTTP klijent za ClickUp API pozive
python-dotenv	1.0.0	Ucitavanje .env varijabli

## Eksterni servisi

Servis	API verzija	Namjena
GitHub API	REST v3	Dohvacanje commit podataka i diff-ova
Groq API	Latest	AI analiza putem LLaMA 3.3 70B
ClickUp API	v2	Pronalazenje taskova i postavljanje komentara

## // arhitektura projekta

Projekat koristi monolitnu arhitekturu u obliku jednog Flask servera. Cijela aplikacijska logika je sadržana u jednom Python fajlu (app.py) što je primjereno za velicinu i kompleksnost projekta. Arhitektura je event-driven — bot reaguje na webhook event-e koji dolaze od GitHub-a.

### Ključni principi

- Event-driven — bot se aktivira samo na webhook event
- Sinhrona obrada — svaki commit se procesira sekvencijalno
- Fail-safe — greska u jednom commitu ne zaustavlja ostale
- Loosely coupled — svaka integracija je nezavisna funkcija
- Graceful degradation — nedostajuci Task ID se loguje, ne crash-uje

## // struktura projekta

Projekat ima minimalnu i fokusiranu strukturu fajlova, primjerenu za single-purpose automation tool.

Fajl	Opis
app.py	Glavni fajl — sva logika (~386 linija)
requirements.txt	Python dependency-ji (7 paketa)
README.md	Instrukcije za setup i pokretanje
.env	API ključevi (nije u git-u)
.gitignore	Isključuje venv/, .env, __pycache__/

## Detaljna struktura app.py

Linija	Funkcija	Opis
1-17	Importi i config	Flask init, env varijable
19-51	get_commit_details()	Dohvaca commit detalje sa GitHub API
54-123	analyze_with_ai()	Salje commit AI-ju, generise sazetak
126-147	format_clickup_message()	Formatira poruku za ClickUp
149-181	extract_task_id()	Regex ekstrakcija Task ID-a
184-210	find_clickup_task()	Pronalazi task u ClickUp-u
213-241	post_to_clickup()	Postavlja komentar na task
243-293	Health + Test routes	Monitoring endpointi
295-386	/webhook route	Glavni webhook handler

## // flow podataka

### Glavni flow: GitHub Push → ClickUp komentar

#### Korak 1: GitHub Webhook

Developer push-a kod na GitHub. GitHub salje POST zahtjev sa JSON payload-om na /webhook endpoint. Payload sadrzi informacije o repozitoriju, branch-u, pusheru i listi commit-ova.

#### Korak 2: Parsiranje payload-a

Flask server prima payload, validira prisustvo "commits" kljuca, i ekstraktuje repo\_name, branch (bez "refs/heads/" prefiksa), i pusher name.

#### Korak 3: Dohvacanje detalja sa GitHub API

get\_commit\_details() koristi PyGithub za dohvacanje pune informacije: skraceni SHA (7 znakova), poruka, autor, datum, lista fajlova sa statusom, broj dodanih/obrisanih linija, i patch sadrzaj za male promjene.

## Korak 4: AI analiza

`analyze_with_ai()` konstruise prompt sa commit metapodacima, statistikama, listom fajlova (max 10) i kodom promjena (max 3 fajla, <500 znakova). Prompt se salje Groq API-ju sa LLaMA 3.3 70B modelom (`temperature=0.3`, `max_tokens=800`).

## Korak 5: Formatiranje i posting

`format_clickup_message()` kombinuje commit detalje i AI sazetak. `extract_task_id()` trazi 9-znakovni alfanumericki ID u branch imenu ili commit poruci. `find_clickup_task()` verifikuje task, i `post_to_clickup()` postavlja finalni komentar.

## Tok gresaka

Bot implementira graceful error handling na svakom koraku procesa:

Situacija	Ponasanje
Commit detalji nedostupni	Commit se preskace (continue)
AI analiza ne uspije	Vraca error string umjesto sazetka
Task ID nije pronadjen	Loguje poruku, ne salje na ClickUp
ClickUp task ne postoji	Loguje upozorenje, nastavlja dalje
Webhook handler crash	Vraca 500 status sa error porukom

## // api endpointi

Metoda	Endpoint	Opis	Odgovor
GET	/	Health check	200 — "Bot radi!"
GET	/test	Test konekcija	200 — JSON statusi
POST	/webhook	GitHub webhook	200/500 — JSON

## POST /webhook — Detalji

Ocekuje standardni GitHub push webhook payload sa `repository.full_name`, `ref`, `pusher.name`, i `commits[]` nizom. Uspjesan odgovor: `{ status: "success", processed: N }`. Neuspjesan: `{ status: "error", message: "..." }`. Ignorirani eventi: `{ status: "ignored", reason: "Not a push event" }`.

# // autentifikacija i sigurnost

## API autentifikacija

Servis	Tip	Varijabla	Koristenje
GitHub	Personal Access Token	GITHUB_TOKEN	PyGithub konstruktor
Groq	API Key	GROQ_API_KEY	Groq klijent init
ClickUp	API Token	CLICKUP_TOKEN	Authorization header

## Sigurnosne mjere

- ▶ Svi API ključevi u .env fajlu (isključen iz git-a)
- ▶ Environment varijable preko python-dotenv
- ▶ Nema hardcode-ovanih kredencijala u kodu
- ▶ ClickUp API koristi Authorization header za svaki zahtjev

## Poznata ograničenja

- ▶ Nema verifikacije GitHub webhook secret-a (X-Hub-Signature)
- ▶ Nema rate limiting-a na webhook endpoint-u
- ▶ HTTPS konfiguracija zavisi od reverse proxy-ja
- ▶ Debug mod treba isključiti za produkciju

# // konfiguracija okruženja

## Environment varijable

Varijabla	Obavezna	Opis
GITHUB_TOKEN	Da	GitHub PAT sa repo pristupom
GROQ_API_KEY	Da	Groq API ključ za AI analizu
CLICKUP_TOKEN	Da	ClickUp API token
GITHUB_REPO	Da	Puno ime repozitorija (owner/repo)
PORT	Ne	Port servera (default: 5000)

## Pokretanje projekta

### Lokalni razvoj

```
$ pip install -r requirements.txt
$ cp .env.example .env
$ # Popuniti .env sa API kljucevima
$ python app.py
→ Server slusa na http://0.0.0.0:5000
```

### Produkcija

```
$ gunicorn app:app
→ Server mora biti javno dostupan za GitHub webhook
→ Alternativno: ngrok za testiranje
```

## AI model konfiguracija

Parametar	Vrijednost	Opis
Model	llama-3.3-70b-versatile	Groq-hosted LLaMA 3.3
Temperature	0.3	Preciznost > kreativnost
Max tokens	800	Max duzina sazetka
Jezik	Srpski	System prompt na srpskom