

# Zadatak za tehnički intervju - Python, Django, REST API, PostgreSQL

**Cilj zadatka:** Implementirati Django aplikaciju koja dohvaća podatke s REST API-ja (simuliranog ili pravog) Cisco DNA Center-a, obrađuje ih te pohranjuje u PostgreSQL bazu podataka.

## Detalji zadatka:

### 1. Postavke projekta

- Kreirajte novi Django projekt i aplikaciju unutar njega.
- Konfigurirajte PostgreSQL bazu podataka koristeći django.db

### 2. REST API poziv

- Potrebno je simulirati REST API endpoint koji vraća podatke o mrežnim uređajima (npr. JSON s podacima poput naziva uređaja, IP adrese, statusa itd.). Ovo možete postići pomoću ugrađenog Django runserver ili mock biblioteke poput responses.

Primjer odgovora API-ja:

```
json
[
  {
    "device_id": "abc123",
    "hostname": "Switch01",
    "ip_address": "192.168.1.1",
    "status": "active",
    "location": "Data Center A"
  },
  {
    "device_id": "def456",
    "hostname": "Router02",
    "ip_address": "192.168.1.2",
    "status": "inactive",
    "location": "Data Center B"
  }
]
```

### 3. Model baze podataka

Definiraj Django model koji će sadržavati sljedeća polja:

- device\_id (jedinstveni identifikator)
- hostname
- ip\_address
- status
- location

Kreiraj migracije i primijeni ih.

### 4. Obrada podataka

Napravi funkciju koja:

- Dohvaća podatke s REST API-ja.
- Parsira odgovor i sprema uređaje u bazu podataka.
- Ako uređaj već postoji u bazi (po device\_id), ažurira njegove podatke.

### 5. Admin sučelje

- Omogući pregled i uređivanje unesenih uređaja putem Django admin panela.

### 6. Endpoint za prikaz podataka

Implementiraj jedan Django REST Framework (DRF) endpoint (GET) koji vraća sve uređaje iz baze u JSON formatu.

### 7. (Opcionalno) Implementirati mogućnost filtriranja uređaja po statusu (active ili inactive) na GET endpointu.

### Kako dostaviti rješenje:

- Pošalji rješenje kao link na GitHub/GitLab repozitorij.
- U README.md dokumentu objasni kako:
  1. Pokrenuti projekt (instalacija ovisnosti, migracije, pokretanje servera).
  2. Testirati funkcionalnosti (mock API, dohvaćanje i pohrana podataka)