

Django Auth + tests, GitHub Actions, GitFlow

Današnji termin vežbi namenjen je za upoznavanje sa:

- Django sistemom za autentifikaciju i autorizaciju
- Django testovima
- *GitHub Actions* alatom za CI/CD
- *GitFlow*-om

Kao referentni primer, koristi se sledeći repozitorijum.

1 Django Auth

Django radni okvir poseduje ugrađeni sistem za autentifikaciju i autorizaciju korisnika.

U funkciji `_napravi_korisnike` obratiti pažnju na sledeće:

- dinamičko kreiranje permisija
- dinamičko kreiranje grupa
- dodavanje permisije grupi
- dodela korisnika grupi

U `kasa_view.py` obratiti pažnju na sledeće:

- kako se radi login
- kako se radi logout
- kako se ograničava pristup pojedinim view funkcijama.

Takođe obratiti pažnju na sledeće:

- `kase_login.html` *template*
- `logout` u `base.html` *template*-u

2 Django Testiranje

Django podržava automatizovano testiranje. Konvencija je da se testovi navode unutar `tests` foldera i da nazivi modula u kojima se nalaze testovi počinju sa prefiksom `test`. Testovi se pišu unutar klasa koje nasleđuju `unittest.TestCase`, a navedene su u pomenutim modulima.

2.1 Jedinično testiranje

Primer pisanja jediničnog testa možete pronaći na linku.

2.2 Integracioni testovi

Klasa `KaseLoginTests` predstavlja primer skupa integracionih testova. Obratiti pažnju na:

- inicijalizaciju baze za ovaj skup testova (`setUpTestData`)
- Pripremu test klijenta za svaki test (`setUp`).
- Slanje zahteva serveru upotrebom pripremljenog klijenta.
- Provera vrednosti koje server vrati kao odgovor na pomenuti zahtev.

2.2.1 Pokretanje `KaseLoginTests` skupa testova

Za pokretanje testova navedenih unutar pomenute klase, izvršiti sledeću komandu:

```
python manage.py test prodavnice.tests --noinput
```

Obratiti pažnju da parametar `--noinput` dovodi do brisanja sadržaja baze nakon završetka testova.

2.3 Pokretanje testova

Ostale načine za pokretanje automatizovanih testove možete pronaći ovde.

3 GitHub Actions

GitHub Actions predstavlja jedan od alata koji služe za CI/CD. Da biste ga koristili, dovoljno je samo da specificirate *workflow* unutar `.github/workflows/`

direktorijuma. Datoteka `.github/workflows/django.yml` predstavlja jednostavan primer *workflow*-a. Više informacije o značenju elemenata *YAML workflow* datoteke možete pronaći u zvaničnoj dokumentaciji. Rezultati svih *workflow*-ova (*build* planova) dostupni su u *Actions* kartici direktorijuma.

Informacije o upotrebi CI/CD alata za potrebe ovog projekta možete pronaći u dokumentu. Posebno obratiti pažnju na pitanja koja se tiču *SCM*-a i *CI/CD* alata.

4 GitFlow

Za potrebe projektnog zadatka iz ovog predmeta, git je potrebno koristiti po preporučenom modelu koji se zove *GitFlow*. Više informacija možete pronaći u sledećem dokumentu. Posebno obratiti pažnju na pitanja i odgovore koji se tiču upotrebe *git* alata.

5 Upotreba GitHub platforme

Smernice za upotrebu GitHub platforme tokom izrade projektnog zadatke pogledati u sledećem dokumentu. Posebno obratiti pažnju na pitanja i odgovore koji se tiču upotrebe *GitHub* platforme.