# Uputstvo za Djangular web app

## postavka foldera za backend-Django i frontend-Angular

**Dule**

**1/1/2021**

[Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.]

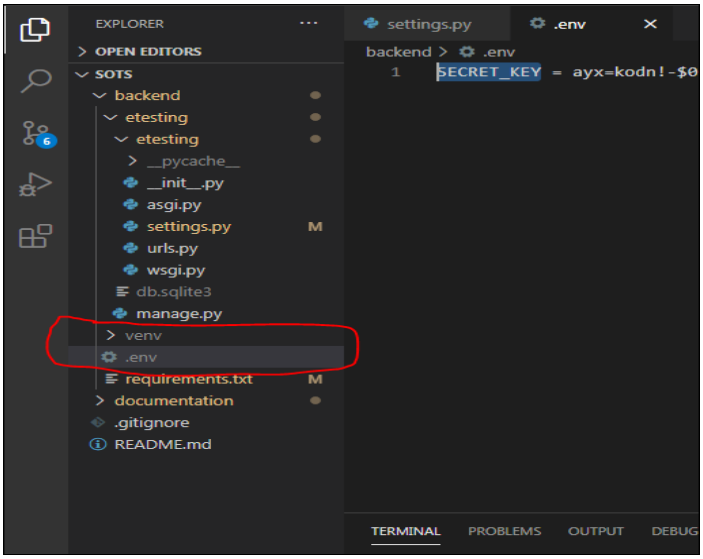
# Contents

## Podesavanja backend-a:

1. Napravi repo na github-u
2. Manuelno napravi folder backend
3. U backend folderu napravi virtuelno okruzenje komandom
   **python -m venv venv**
4. Aktiviraj napravljeno okruzenje komandom
   **venv\Scripts\activate.bat**
5. Eksportuju requirements.txt u backend folder
   **pip freeze > requirements.txt**
6. Instalirati sledece zavisnosti komandom
   **pip install django**
   **pip install djangorestframework**
7. Kreiranje Django projekta komandom
   **django-admin startproject etesting**
8. Pokreni napravljeni Django projekat
   **python manage.py runserver**
9. Uradi prvu migraciju u projektu
   **python manage.py migrate**
10. Podesavanje CORS – a https://dzone.com/articles/how-to-fix-django-cors-error
    **pip install django-cors-headers**
11. dfasdfa

## Podesavanje Django Secret Key exposed on GitHub (dotenv)

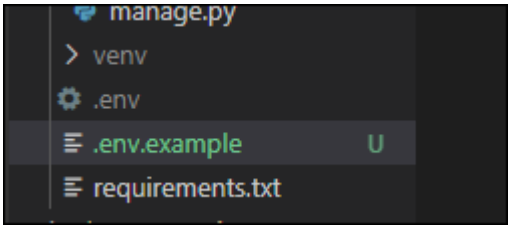
1. Instalirati zavisnost komandom
   **pip install python-dotenv**
2. U settings.py na vrh fajla copy-past sledece komande:
   **from dotenv import load_dotenv**
   **load_dotenv()**
3. U root folderu (backend folder) napraviti novi .env fajl. Napraviti novi .env fajl na nivou venv folder



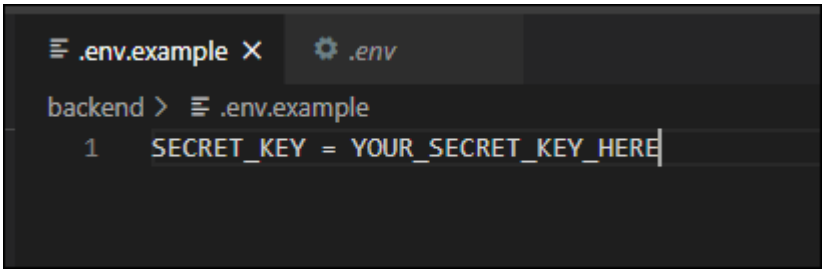

4. U .env fajl copy paste SECRET_KEY iz settings.py fajla. Kopirati po principu key = value samo vodi racuna da value bude bez navodnika
5. .env fajl je izuzet od push-ovanja sa .gitingore bas kao sto je i venv folder izuzet

```
# Environments
.env
.venv
env/
venv/
ENV/
env.bak/
venv.bak/
```

6. U settings.py zameniti SECRET_KEY sa SECRET_KEY = os.getenv('SECRET_KEY')
7. Dodati jos jedna fajl na nivou .env fajla pod imenom .env.example



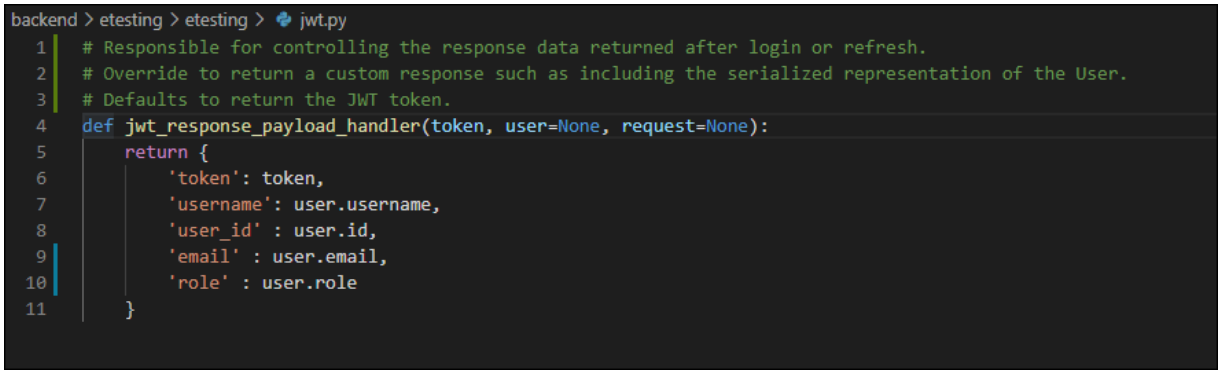

8. Svrha ovog .env.example fajla je da svima koji kloniraju ili skidaju projekata da uputstvo kako treba podesiti sistemske varijable da bi projekat moga da se pokrene. Ovaj fajl nije izuzet sa .gitignore fajlom.




9. Izvori:
   9.1. https://dev.to/vladyslavnua/how-to-protect-your-django-secret-and-oauth-keys-53fl
   9.2. https://pypi.org/project/python-dotenv/
   9.3. https://www.youtube.com/watch?v=17UVejOw3zA&t=446s

## Podasavanje JSON Web Token (JWT)

1. Instaliraj zavisnost:
   **pip install djangorestframework-jwt**
2. Dodavanje dodatnih stvari (id usera, role usera, sta god hoces da dodatno posaljes uz token) koje putuju zajedno sa tokenom na front.
   https://stackoverflow.com/questions/54575716/getting-user-id-returned-with-jwt
   Dodaj poseban fajl **jwt.py** na nivou projekta




3. dadfdasfdsfs

## Formiranje i punjenje tabela u bazi podataka

1. Formiranje praznih tabla (kostur): Sa ove dve komande formiraju se prazne tabele u bazi podataka koje nisu popunjene. Pravi se migracija od modela do praznih tabela zato se i zovu migracije.
   **python manage.py makemigrations**
   **python manage.py migrate**
2. Punjenje praznih tabela (populate db): Za punjenje baze podataka koriste se dummy podaci koji se zovu fixtures. Postoje globalni i lokalni fixtures. Lokalni fixtures su podaci koji se vezuju za svaku aplikaciju u projektu ponaosob i nalaze se u fixtures folderu svake aplikacije. Svaka aplikacija ima folder fixtures koji se rucno pravi. U tim folderim se nalaze lokalni fixtures sa podacima za konkretnu app u projektu. Lokalni fixtures se koriste za testiranje aplikacije. Globalni fixtures prave se na nivou projekta i obuhvataju dummy podatke na nivou citavog projekta (projekat = suma svih app(komponenti)). Globalni fixtures koristi se za punjenje baze podataka inicijalnim podacima. Globani fixture nalazi se na nivou projekta. Globani fixture se nalazi na nivou db.sqlite3 fajla.
   2.1. db.sqlite3 fajl predstavlja binarni fajl koji nije citljiv za ljude. U ovom fajlu se nalaze podaci iz lokalne baze podataka. Od push-ovanja je izuzet sa .gitignore fajlom. Ne moze da se koristi kao .sql skripta u Springu zato sto se vezuje za lokalni racunar. Sqlite3 baza podataka predstavlja embedded (ugradjenu) bazu podataka bas kao sto je H2 za Spring. db.sqlite3 fajl ne moze da se koristi na nivou github tima (ideja da ceo tim koristi iste inicijalne podatke) zato sto su svi podaci koji se nalaze u ovoj skripti vezani za lokani racunar i skripta je izuzeta od push-ovanja razlika u odnosu na Spring i .sql skriptu.
   2.2. Eksportovanje db.sqlite3 fajla u globalni fixtures: Pozicioniraj se na nivo db.sqlite3 skripte u folder projekta. Dodaj sve sto imas u db.sqlite3 lokalnu skriptu (dodavanje ide preko admin panela, fronta, terminala). Proveriti na sajtu https://sqliteonline.com da li su svi podaci dodati koji treba da budu dodati. Ici na opciju File open DB. Kada imas sve u lokalu exportuju db.sqlite3 u db.json fajl sledecom komandom. Na ovaj nacin lokani podaci su prebaceni u fajl koji nije izuzet od push-ovanja i ceo tim moze da koristi iste podatke. Push-uj projekat na git.
   **python manage.py dumpdata --indent 4 > db.json**
   2.3. Ucitavanje globanog fixture. Povuces sa gita najnoviju verziju projekta. Odradis dve gore navedene komande za migraciju. Ako izbiju greske izbrisati lokalni db.sqlite3 fajl koji je konfliktan. Ponovo pokrenuti komande za migraciju. Biti na nivou db.sqlite3 fajla i kucati sledecu komandu. Sa ovim je baza svih pojedinacnih clanova tima popunjena isitim inicijalnim podacima
   **python manage.py loaddata db.json**
   2.4. Proveriti na sajtu https://sqliteonline.com da li su svi podaci ucitani koji treba da budu ucitani. Ici na opciju File open DB

3. YAML db.yaml sve isto kao i sa db.json samo mora da se instalira sledeca zavisnost
   **pip install PyYAML**
   Nakon sto je instalirana zavisnost koriste se sledece komande za dump-ovanje i load-ovanje :
   **python manage.py dumpdata --indent 4 > db.yaml**
   **python manage.py loaddata db.yaml**

4. Izvori:
   4.1. https://docs.djangoproject.com/en/3.1/howto/initial-data/

## Podesavanje frontend-a

1. Pozicioniraj se u folder frontend i za intstalaciju angular projekta koristiti sledecu komandu:
   **ng new my-app**
2. Pozicioniraj se u folder frontend i za instalaciju angular material koristiti sledecu komandu:

**ng add @angular/material**

3. Material module
   **ng generate module material**
4. Toastr Angular Notification
   **npm install ngx-toastr --save**
5. Dodavanje toaster stila u angular.json fajl
   **"node_modules/ngx-toastr/toastr.css"**
6. Dodaj u folder frontend fajl **proxy.conf.json** koji sluzi za povezivanje frontend - a sa backend – om.
7. Generisanje core modula
   **ng generate module core**
8. Generisanje shared modula
   **ng generate module shared**
9. Generisanje authentication modula
   **ng generate module authentication**
10. Login komponenta u authentication modulu
    **ng g component authentication/login**
11. dfasdf

## Najcesce komande u Angularu2+

Dfasdaf

## Kreiranje custom User modela (override Default Django User modela)

Postupak kako su se menjali dokumenti:

1. settings.py u projektnom folderu
2. models.py u app folderu
3. serializers.py u app folderu
4. permissions.py u app folderu
5. views.py u app folderu
6. urls.py u app folderu
7. admin.py u app folderu

## Kreiranje custom superusera

Posto si izmenio Default (built in Django User model ) komanda python manage.py createsuperuser vise ne funkcionise. Ova komanda moze kada koristis default-ni User model. Ove sve naredne komande kucas u komandnoj liniji. Sledeci postupak za kreiranje custom superusera:

```
python manage.py shell
from accounts.models import User
user = User()
user.is_staff = True
user. is_superuser= True
user. is_active= True
user.username = 'admin'
user.set_password('admin123admin')
user.email = 'admin@gmail.com'
user.first_name = 'admin'
user.last_name = 'admin'
```

user.role = 'ADMIN'
user.save()
User.objects.all()
exit()

```
TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying sessions.0001_initial... OK

(venv) C:\Users\Dule\Desktop\SOTS\backend\etesting>python manage.py shell
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:57:15) [MSC v.1915 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from accounts.models import User
>>> user = User()
>>> user.is_staff = True
>>> user. is_superuser= True
>>> user. is_active= True
>>> user.username = 'admin'
>>> user.set_password('admin123admin')
>>> user.email = 'admin@gmail.com'
>>> user.first_name = 'admin'
>>> user.first_name = 'admin'
>>> user.last_name = 'admin'
>>> user.is_student = False
>>> user.is_teacher = False
>>> user.save()
>>> User.objects.all()
<QuerySet [<User: admin>]>
>>> exit()

(venv) C:\Users\Dule\Desktop\SOTS\backend\etesting>
```

## Debugging Angular Apps through Visual Studio Code

Prvo moras pokrenuti frontend aplikaciju pa tek onda pokrenuti debug mode. Jedno i drugo moraju biti pokrenuti.

1. Make sure VS Code, Google Chrome, and all the Angular parts are already installed.
2. Install the Debugger for Chrome extension in VS Code.
3. Create a *launch.json* config file (by clicking the gear icon in the Debug view).
4. Set an appropriate config spec in the *.vscode/launch.json* file (example below).
5. Set breakpoints in the editor.
6. Launch the Angular app separate from the debugger (such as by running “ng serve” from the command line).
7. Run the VS Code debugger “launch” job against the app (by clicking the green arrow in the Debug view).

https://stackoverflow.com/questions/48892311/angular-cli-1-7-0-and-visual-studio-code-cant-set-breakpoints

If you are using a WorkSpace with 1 or more Projects..

Working for me: launch.json

```
"version": "0.2.0",
"configurations": [
    {
        "type": "chrome",
        "request": "launch",
        "name": "ng serve",
        "url": "http://localhost:4300",
        "webRoot": "${workspaceFolder}/ProjectName",
        "sourceMapPathOverrides": {
            "webpack:/*": "${webRoot}/*"
        }
    }
]
```

1- Start Project ng serve --port 4300

2- Start Debugging