Logo

Description automatically generated

10/21/2021

Prof. îndrumător: Ovidiu Valea

Student: Sorina - Rodica Ioja

Aplicație Django REST

Administrarea sistemelor de operare

Partea I

Contents

[Cerințele rezolvate 2](#_Toc85698023)

[Modul de rezolvare 2](#_Toc85698024)

[Probleme întâlnite și modul de rezolvare 6](#_Toc85698025)

[Concluzii 6](#_Toc85698026)

# Cerințele rezolvate

Prima parte a proiectului are ca scop realizarea unei aplicații Django REST framework avand ca principale entitati liste si carduri.

# Modul de rezolvare

Initial am vrut sa lucrez in Ubuntu, respectiv masina virtuala, dar avand blocari destul de frecvente am renuntat rapid la idee si am create proiectul in Windows.

Pentru inceput am verificat daca am instalat python 3 **python –version**, ulterior am incercat sa imi creez un virtual environment pentru a instala in acesta o anumita versiune a Django si a altor pachete si biblioteci. Nu am reusit sa il creez asa ca am rulat **pip install virtualenv**. Apoi am create virtual environment-ul cu ajutorul comenzii **python -m venv venv\_django**, iar apoi l-am activat **venv\_django\Scripts\activate**.

In acest virtual environment instalez Django framework **pip install Django djangorestframwork**. In folderul care contine virtual environment-ul rulam **django-admin startproject project.** Instalam pycharm, deschidem proiectul si adaugam interpretorul de python din venv-ul creat.

Pentru a creea lista de task-uri rulam **python manage.py stratapp scrumboard**, iar ulterior o declaram in INSTALLED\_APPS. Am definit modelele List si Card, pentru a adduce modificarile in baza de data am rulat **python manage.py makemigrations**, dupa care **python manage.py migrate**.

Graphical user interface, application

Description automatically generatedAdaugam un superuser astfel **pyhton manage.py createsuperuser** si urmam instructiunile din terminal. Adaugam in admin urmatoarele

Putem rula **python manage.py runserver**, aplicatia va porni pe portul 800 si putem intra la **localhost:8000/admin** unde dupa logare va fi astfel:

A screenshot of a computer

Description automatically generated

Aici putem adauga o lista si carduri. Includem in INSTALLED\_APPS si ‘rest\_framework’. Creem serializers.py, unde adaugam doua clase care vor ajuta la reprezentarea informatiei extrase din baza de date.

Graphical user interface, text, application

Description automatically generated

Iar in api.py expunem cate un REST API pentru fiecare din obiecte:

Graphical user interface, text, application, email

Description automatically generated

Adaugam path-ul si in urls.py principal.

Rulam python manage.py si accesand /scrumboard/cards sau /scrumboard/lists vom observa cardurile adaugate mai devreme din admin.

Graphical user interface, text

Description automatically generated

# Probleme întâlnite și modul de rezolvare

Daca vom trece in INSTALLED\_APPS prima data ‘scrumboard’ si doar ulterior rulam **python.exe .\manage.py startapp scrumboard**, atunci aplicatia nu va recunoaste scrumboard. Operatiile trebuie realizate in ordine inversa.

# Concluzii

Din prima parte a proiectului am invatat cum se porneste/incepe o aplicatie folosind Django REST framework.