

Instrukcija, kā palaist Python Flask serveri savā datorā.

Noder, lai izmēģinātu servera darbību savā datorā, pirms to izliek publiskai lietošanai.

Balstīta uz Pāvila Jurjāna skaidrojumiem.

1. Sagatavojam nepieciešamās datnes:
 - a. HTML datne, kas būs web lapa, kurā lietotājs darīs kaut ko, kā dēļ vajadzēs vērsties pie servera pēc datiem. Piemēram, spēlē "Karātavas" pieprasīt serverim jaunu vārdu. Es esmu sagatavojusi ilustratīvu piemēru **test.html**, kurā uz random uzģenerē it kā spēles laiku, paprasa spēlētāja vārdu, un tad laiku kā spēlē iegūtos punktus kopā ar ievadīto vārdu sūta uz serveri glabāšanai. Atpakaļ saņem sarakstu, kurā spēlētāji sakārtoti punktu secībā, un parāda to web lapā. Datnē ir komentāri, lai saprastu kam domātas attiecīgās komandas.
 - b. PY datne, kas būs serveris, kurā notiks pieprasījumu apstrāde. Esmu sagatavojusi piemēru **test.py**, kurā serveris saņem pieprasījumu no web lapas ar spēlētāja vārdu un punktiem, nolasa no datnes **vardi.txt** iepriekšējo spēlētāju rezultātus, pievieno jauno, saglabā izmaiņas datnē **vardi.txt**, tad sakārto izmainīto spēlētāju un punktu sarakstu un sūta atpakaļ web lapai. Datnē test.py ir komentāri, lai saprastu kam domātas attiecīgās komandas, bet pietrūkst kļūdu apstrādes, piemēram, ko darīt, ja nav datu datnes **vardi.txt**.
 - c. Datu datne(s), šajā gadījumā **vardi.txt**. Datnes saturam var būt dažādi formāti, es izvēlējos **JSON**, jo tad visas datnes rindas var nolasīt un ierakstīt ar vienu **json** objekta komandu, kas, protams, sevī slēpj ciklus. Tas šobrīd ir populārs formāts.
 - d. Datnēm jābūt strukturētām pa mapēm, konkrēti laikam(?) **html** datnei jābūt apakšmapē **static**. Datu datni **vardi.txt** arī noliku **static** mapē, bet nezinu, vai tā ir pareizākā stratēģija.

2. Ja datorā vēl nav instalēt Flask, tad to instalē no komandrindas šādi:

```
Windows:
cmd
> py -m pip install Flask
Linux:
bash
$ pip install Flask
```

3. Pasaka Flask(am), kura Python programma būs jālaiž kā serveris. Šajā gadījumā tā ir programma **test.py**. Teorētiski tur vajadzētu varēt norādīt ceļu līdz datnes **test.py** atrašanās vietai, praktiski man neizdevās, tāpēc pirms šīs komandas izpildes es konsolē vispirms ar **cd** (change directory) aizgāju uz mapi, kurā stāv **test.py** datne.

```
Windows
cmd
> set FLASK_APP=test.py
Linux
bash
$ export FLASK_APP=test.py
```

4. Palaiž Flask. Arī to es darīju, atrodoties iepriekš minētajā mapē. Tajā vēl jāatrodas arī iepriekš minētajai **static** mapei.

```
Windows
cmd
> py -m flask run
Linux

bash
$ python -m flask run
```

5. Ja viss notiek veiksmīgi, tad konsolē parādās uzraksts, ka serveris palaists adresē <http://127.0.0.1:5000> un ka to var apturēt, spiežot CTL-C. Atverot šo adresi pārlūkprogrammā, tajā sāks darbu **test.html** kods.