 Python

**Hvad er Python, og hvad bruges det til?**

Python er et ” high level programming language”, det kan bruges til primært 3 forskellige ting. Det kan bruges til datavidenskab (analyse, og visualisering) hvilket det oftest også bruges til, maskinelæring og webudviking som jeg har brugt det til.

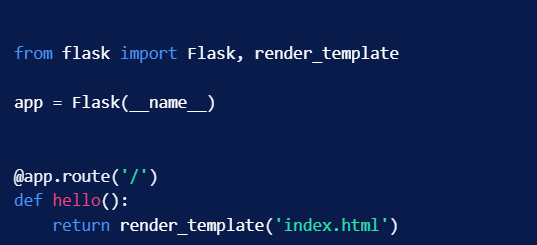
Python er et back-end sprog der kører på en server der snakker sammen med applikationer, databaser og API’er. Mange anbefaler at hvis man skal bruge Python til at bygge en webapplikation skal man bruge Flask som udvider Python.

Flask er et mikro web framework som sagt udvider python en smule, det gør det nemmere at bygge en webapplikation.



Det her er et eksempel med Flask. Vores output vil her være ”Hello, World!” på vores index side (”/”)

Man kan installere en masse ekstra ting til Flask alt efter hvad man vil lave. Hvis man fx ville lav en nav bar, kan man skrive ” pip install flask-nav”. Her installerer man alt det man skal bruge for at kunne lave en nav.

På den øverste linje importerer jeg Flask, og en render\_template. Nede i vores return, der returnerer vi en render\_template(’index.html’), det vil sige vi viser index.html filen.

Jeg har lavet et Flask + JSX projekt, for at vise hvordan det hænger sammen, og hvordan front-end og back-end snakker sammen.

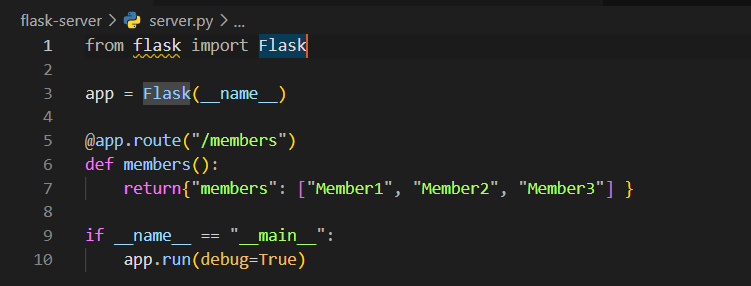
Jeg har en mappe der hedder ”flask-server” hvor alt mit back-end ligger i, dvs. min server.py fil ligger der. Der ligger også en venv (virtual enviroments) mappe, det vigtige i den mappe er at der ligger en activate fil som der skal aktiveres for at det hele kan køre. Den holder styr på alt for en, hvis der skulle opstå en konflikt mellem back-enden og front-enden løser den problemet. Normalt når man har en fejl og man går ind i browseren ville man ikke kunne komme videre men når vores script er aktiveret er det lige meget om der er fejl eller ikke.



Hvis man tilføjer den nederste linje slipper man for at genstarte severen hver gang der er sket en ændring.

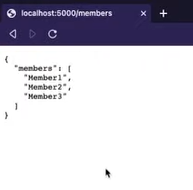
For at vi Flask og JSX til at snakke sammen kræver det man laver et helt nyt projekt og starter med at sætte back-enden op. Bagefter kan man så lave en ny mappe uden for back-enden og lave en react-app nede i den mappe. Og som de sidste skal man så ind i package.json og lave en linje der sætter proxy til at køre på en bestemt url.

  "proxy": "http://127.0.0.1:5000",

For at starte vores python server skriver man ”cd flask-server” (afhængig af hvilken mappe man står i), når man har tjekket man er i den rigtige mappe skriver man python ”server.py” også starter en server. Bagefter kan man så starte ens frontend, ved at ”cd frontendapp” (alt efter hvilken mappe man er i), også ”npm start” også kører det hele. Serverene er afhængige af hinanden, så man kan ikke bare lukke den ene ned fordi man ikke bruger den.

På linje 7 laver vi et return, hvor vi faktisk konstruerer vores eget lille ”API”.

Vores back-end server kan vi selvfølgelig også se, det ser sådan her ud:



Det her API kan jeg fetch på ligesom vi plejer, det giver det her output:

