

## Harjoitustyö vaihe 4: Datan visualisointi

Kuvaan tässä harjoitustyön osan 4 raportissa, miten visualisoin sovelluksen keräämän datan. Käytän rajapintana OpenWeatherMapin tarjoaman Current Weather Data API:a. Tässä vaiheessa teen datalle visualisoinnin. Päädyin tekemään visualisoinnin Highchartsin avulla, sillä moni muukin on käyttänyt sitä, joten sain suosituksia sen käytöstä.

### Visualisoinnin luominen

Tutustuin aluksi aiempien harjoitustöiden pohjalta Highchartseihin ja googlettamalla lähteitä. Valitsin Highchartsien demoista sopivan kaavion. Ennen kuin visualisoin API:sta kerättävää dataa testasin Highchartsin toimintaa yksinkertaisella valmiilla koodilla. Kun se onnistui, siirryin oman datan visualisointiin. Tässä vaiheessa päätin, että käytän visualisointini pohjalla sekä current weather dataa että 3 tunnin ennuste 5 päivälle dataa. Sovelluksessa on mahdollista vertailla eri kaupunkien 3 tunnin ennustetta viidelle seuraavalle päivälle.

Kaaviota varten tarvitsin 5 day / 3 hour forecast dataa. Mietin ja googlailin hetken, miten saan haettua uutta ennustedataa. Lueskeltuani aikaisempia harjoitustöitä löisin PyOWM:n, joka Pythonin kirjasto OpenWeatherMapin API:elle ja mahdollistaa API:n dataan keruun ja tallentamisen palveluun. PyOWM:n omilla sivuilla oli kattava dokumentaatio sen käyttämisestä. Aluksi asennettiin PyOWM komentorivillä.

```
pip3 install pyowm
```

Sen jälkeen importasin PyOWM:n *views.py*-tiedostoon, jossa sitä lopulta tarvittiin.

### Datan visualisoinnin esittäminen

En aluksi saanut luomaani datan visualisointia näkyviin millään palvelimella. Hommasta ei meinannut tulla yhtään mitään, vaikka etsin ja korjasin virheitä ja yritin saada sivua näkyviin päiviä. Otin kandin takia kuitenkin viikon etäisyyttä tähän työhön. Se oli hyvä päätös, sillä palattuani takaisin Djangoon pariin löysin *visualization.html*-tiedoston javascript-osuudesta pieniä ja turhauttavia virheitä. Käytyäni työtä läpi uudestaan sain Highchartsit toimimaan ja sääkuvaaja tuli näkyviin iloisesti palvelimella.

Varsinaista koodausta oli tässä kohtaa harjoitustyötä enemmän kuin aikaisemmin. Käytin hyväkseni edellisten palautusten pohjia. Lisäsin viime palautuksen *weather*-applikaation *views.py*-tiedostoon *visualization*-funktion, joka kerää datan 5 day / 3 hour forecast API:sta. PyOWM:n dokumentaation mukaisesti importasin sen *views.py*-tiedoston alkuun, lisäsin alkuun *API\_key*:n ja kutsun PyOWM:lle ja API-avaimen välille. Alla on kuva *views.py*-tiedoston *visualization*-funktioista. HUOM! Kuvassa ei ole koko koodia.

```
46
47 def visualization(request):
48
49     cities = City.objects.all()
50     temp_data = []
51
52     for city in cities:
53         fc = owm.three_hours_forecast(city.name)
54         f = fc.get_forecast()
55
56         temp_list = list()
57         for weather in f:
58             temp = weather.get_temperature(unit='celsius')['temp']
59             temp_list.append(temp)
60
61         forecast = {
62             'name' : city.name,
63             'data' : temp_list
64         }
65
66         temp_data.append(forecast)
67
68     context = {'temp_data' : json.dumps(temp_data)}
69     return render(request, 'weather/visualization.html', context)
70
```

Loin projektitason *templates*-kansioon *weather*-kansioon *visualization.html* tiedoston. Sinne sijoitin Highchartsin koodipohjan.

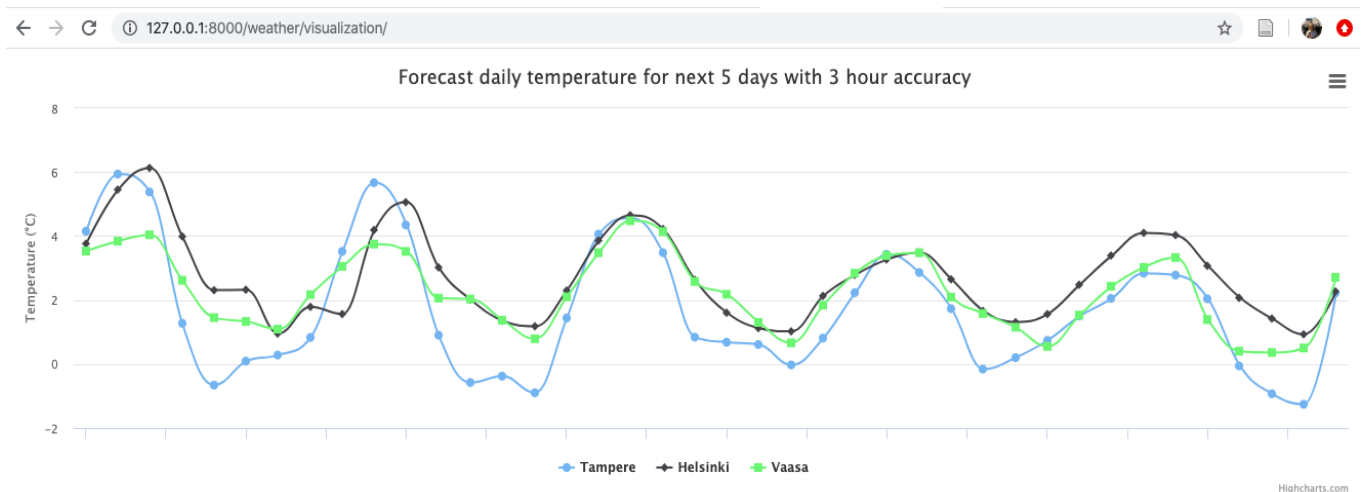
```
mysite > templates > weather > visualization.html > html
1 <!doctype html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <title>Weather visualization</title>
6 </head>
7 <body>
8     <div id="container"></div>
9     <script type = "text/javascript">
10         var series = "{{temp_data}}"
11     </script>
12     <script src="https://code.highcharts.com/highcharts.js"></script>
13     <script src="https://code.highcharts.com/modules/exporting.js"></script>
14     <script src="https://code.highcharts.com/modules/export-data.js"></script>
15
16     <div id="container" style="min-width: 310px; height: 400px; margin: 0 auto"></div>
17
18     <script>
19         Highcharts.chart('container', {
20             chart: {
21                 type: 'spline',
22                 scrollablePlotArea: {
23                     minWidth: 600,
24                     scrollPositionX: 1
25                 }
26             },
27             title: {
28                 text: 'Forecast daily temperature for next 5 days with 3 hour accuracy'
29             },
30             xAxis: {
31                 labels: {
32                     enabled: false
33                 }
34             },
35             yAxis: {
36                 title: {
37                     text: 'Temperature (°C)'
38                 }
39             },

```

```
39      },
40      plotOptions: {
41        line: {
42          dataLabels: {
43            enabled: true
44          },
45          enableMouseTracking: false
46        }
47      },
48      series: {{ temp_data|safe }}
49    }); </script>
50 </body>
51 </html>
```

Suurimmat muokkaukset tein siis *views.py* ja *visualization.html*-tiedostoihin. Tein myös pieniä muutoksia tässä vaiheessa projektitason *urls.py* ja *settings.py*-tiedostoihin ja *weather*-applikaation *urls.py*-tiedostoon.

Toteutin lämpötilaennusteen Highchartsilla OWM:n tarjoamasta API-datasta. Kun sain visualisoinnin vihdoin näkyviin, sivusto näytti halutun alla esitetyn lämpötilakuvaajan.



## Versionhallinta GitHubiin

Saatuani työni toimimaan halutusti lisäsin uudet ja päivitetyt tiedostot GitHubiin. Tämä tapahtui komentorivillä suurin piirtein seuraavilla komennoilla

```
git . add
git commit -m "API:sta saatiedot saava saapalvelu"
git push -u origin master
```

Harjoitustyön 4. vaiheen projektin versio löytyy GitHubistani  
<https://github.com/anutamminen/ohsiha>.

## Helppoa/haasteellista

- Javascriptin käyttö oli yllättävän haastavaa, sillä oli vaikea ymmärtää, missä on virhe. Sitä alkoi kuitenkin ymmärtämään käyttäessä.
- En saanut visualization.html-tiedoston Highcharts-kaaviota aluksi näkyviin. Se johtui container divin puuttumisesta ja virheistä javascriptissa. Se oli turhauttavaa, mutta etäisyyden ottaminen omasta työstä auttoi löytämään virheet ja ymmärtämään toimintaa paremmin. Highchartsien käyttö oli kuitenkin selkeää, kun sain ne lopulta toimimaan sovelluksessa ja luin dokumentaatiota huolella.
- PyOWM oli mukava lisä. Data oli helppo kerätä sen avulla API:sta.
- Views.py-tiedoston linkittyminen koko muuhun ohjelmaan oli välillä hämmentävää. Toisaalta onnistuin, opin Djangoa paljon uutta ja ymmärrän paremmin sen toimintaa nyt.

## Lähteet

Highchartsien käyttäminen

<https://simpleisbetterthancomplex.com/tutorial/2018/04/03/how-to-integrate-highcharts-js-with-django.html>

PyOWM:n dokumentaatio

<https://pyowm.readthedocs.io/en/latest/index.html>

OpenWeatherMapin API:n dokumentaatio

<https://openweathermap.org/api>

JSFiddle apuna Highcharts-visualisoinnissa

<https://jsfiddle.net/>

Aikaisemmat harjoitustyöt ja Stack Overflow pienien ongelmien ratkaisemissa