

# Harjoitustyön neljäs vaihe

---

Tässä vaiheessa harjoitustyötä olen toteuttanut jo kirjautumis- sekä rekisteröitymispalvelun, CRUD-toiminnot ja rajapinnan käyttöönoton sovellukseeni. Harjoitustyön neljänteen vaiheeseen toteutin visualisoinnin Highchartsilla. Visualisointi e esittää vuoden 2018 dataa tapahtumien määrästä, joihin liittyy musiikki, tanssi, ja kulttuuri.

## Teknologiat

- Django 2.0.2
- Python 3.6.4
- SQLite
- Atom

## Näkymän luominen

Ensimmäiseksi loin haluamani Highcharts-visualisoinnin JSFiddle-palvelulla, jonka rakenteen sai helposti kopioitua omaan html-tiedostoon.

```

<script src="https://code.highcharts.com/highcharts.js"></script>
<script src="https://code.highcharts.com/modules/data.js"></script>
<script src="https://code.highcharts.com/modules/exporting.js"></script>
<script src="https://code.highcharts.com/modules/export-data.js"></script>

<div id="container" style="min-width: 310px; height: 400px; margin: 0 auto"></div>

<table id="datatable">
  <thead>
    <tr>
      <th></th>
      <th>Musiikki</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>Musiikki</th>
      <td>{{kategoriat.musiikki|safe}}</td>
    </tr>
    <tr>
      <th>Kulttuuri</th>
      <td>{{kategoriat.kulttuuri}}</td>
    </tr>
    <tr>
      <th>Tanssi</th>
      <td>{{kategoriat.tanssi}}</td>
    </tr>
  </tbody>
</table>
<script>

```

```

<script>
    Highcharts.chart('container', {
    data: {
        table: 'datatable'
    },
    chart: {
        type: 'column'
    },
    title: {
        text: 'Tapahtumien määrä kategorioittain Tampereella vuonna 2018'
    },
    yAxis: {
        allowDecimals: false,
        title: {
            text: 'Tapahtumien määrä'
        }
    },
    tooltip: {
        formatter: function() {
            return '<b>' + this.series.name + '</b><br/>' +
                this.point.y + ' ' + this.point.name.toLowerCase();
        }
    }
    });
</script>
</body>
</html>

```

Seuraavaksi loin `views.py` -kansioon funktion, joka käy läpi rajapinnan dataa laskien sieltä tapahtumien määrää. Funktio hakee dataa jo valmiiksi kategorioihin rajatuista json-tiedostoista.

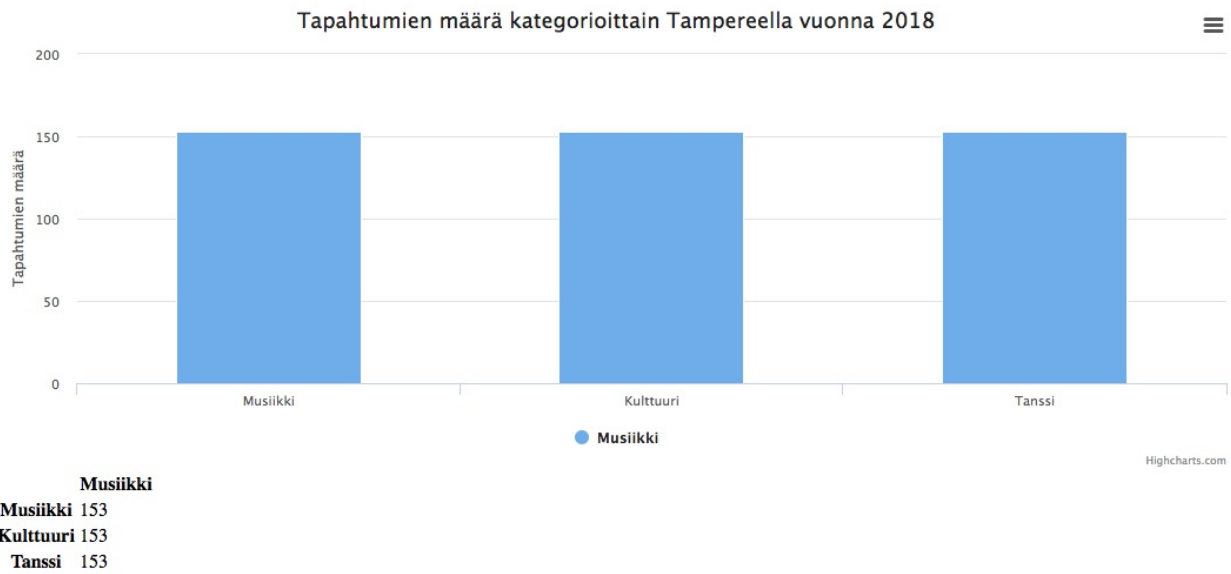
```

if requests.get('https://visittampere.fi:443/api/v1/event?tag=music&limit=1000000').status_code==200:
    response = requests.get('https://visittampere.fi:443/api/v1/event?tag=music&limit=1000000')
    print(response.status_code)
    musicdata = response.json()
    i = 0

    for event in musicdata:
        #if event['start_datetime'] > 1522684233000:
        # for event['times'] in event:
        #     i += 1
        if event['start_datetime'] >= 1514764800000:
            if event['start_datetime'] < 1546128000000:
                i += 1
                print(i)
    kategoriat['musiikki'] = i
    i = 0
    print(kategoriat)

```

Myös `urls.py` -kansioon piti muistaa luoda polku html-sivun näkymälle. Seuraavaksi on esitetty html-sivulla luotu Highcharts-näkymä tapahtumadatalle.



Palkit yltävät yhtä korkeiksi, sillä tulevat tapahtumat sisältävät kaikki kategoriat.

## Hyviä lähteitä

- How to Integrate Highcharts.js with Django:  
<https://simpleisbetterthancomplex.com/tutorial/2018/04/03/how-to-integrate-highcharts-js-with-django.html>
- JSFiddle: <https://jsfiddle.net/>
- Highcharts demo: <https://www.highcharts.com/demo>

## Ilmenneet hankaluudet ja muut huomiot

- Rajapinnasta oli vaikea keksiä hyvää dataa esitettäväksi
- Dictin käsittely html-sivulla tuotti vaikeuksia
- Javascript tuntuu vielä aika vieraalta