

Ohsiha harjoitustyö vaihe 3

Ohjelmallinen sisällönhallinta kevät 2019

Joona Prehti, joona.prehti@tuni.fi

Koostepalvelu suunnitelma ja API

Kurssin alusta alkaen minusta on tuntunut luontevimmalta lähteä kehittämään jotain koostepalvelua. Aluksi minulla oli ollut suunnitelmissa kehittää jokin cryptovaluutta-dataa visualisoiva API, mutta huomasin että historia datan saaminen API:sta olisi maksanut useita kymmeniä dollareita kuukaudessa. Vaihdoin API:a Fingridin API:iin, josta saan haettua paljon erilaista sähköverkkoon liittyvää dataa. Ajattelin että olisi hauska visualisoida, vaikka Venäjän Suomeen myymän sähkön määrää megawatteina viimeisen 6 kuukauden ajalta. Samalla voisi tutkia vaikka, miten vuodenaajat ja sääilmiöt vaikuttavat Suomen Venäjältä ostaman sähkön määrään.

Datarajapinnan käyttöönotto

Aluksi tein projektiin uuden JavaScript moduulin nimeltä `api_fingrid.js`, joka hoitaa datan hakemisen Fingridin REST API:sta. Kyseinen moduuli hoitaa myös tarvittaessa datan tallentamisen PostgreSQL tietokantaa, jos haluan myöhemmin tallentaa dataa myös sinne. Datan tallentaminen tietokantaan ei ole silti välttämätöntä, koska voin aina tarvittaessa hakea sen uudestaan API:n kautta. API:n datan tallentaminen myös tietokantaan olisi melko suoraviivaista koska minulla on jo tietokanta pystyssä ja tallennan sinne käyttäjien salasanoja ja käyttäjänimiä.

Saadakseni dataa Fingridin API:sta minun piti ensin rekisteröityä, jotta sain käyttööni "x-api-key" -avaimen. Avaimen avulla kykenen tunnistautumaan API kutsuja tehdessäni ja Fingridin päässä kyetään paremmin valvomaan, että kuka käyttää API:a ja minkä verran. Avain laitetaan kutsussa header -kohtaan ja se otetaan vastaan API:n päässä, jossa se validius tarkastetaan. Alla kuvakaappaus `api_fingrid.js` -moduulista, jossa API:sta haetaan dataa Venäjän ja Suomen välisestä sähkönmyynnistä ensimmäisen 3 kuukauden ajalta vuonna 2019:

```

1  const router = require('express').Router();
2  const request = require('request');
3
4  router.get('/data', function (req, res) {
5    request({
6      url: 'https://api.fingrid.fi/v1/variable/58/events/json?'
7        + 'start_time=2019-01-01T01%3A01%3A01Z&end_time=2019-03-01T01%3A01%3A01Z',
8      json: true,
9      headers: {
10       'x-api-key': 'API KEY TULEE TÄHÄN MUTTA POISTIN SEN KUVAN AJAKSI TURVALLISUUS SYISTÄ'
11     }
12   }, (error, response, body) => {
13     res.json(body);
14   });
15
16 });
17
18 module.exports = router;

```

Kuten kuvakaappauksesta voikin huomata niin datan hakeminen onnistuu request nimisen kirjaston avulla melko yksinkertaisesti. Kun käyttäjä menee osoitteeseen `"/data"`, niin hänelle näytetään json -muodossa sähköön myynti dataa. Käyttäjä kykenee ottamaan yhteyttä `"/data"` osoitteeseen vain ja ainoastaan, jos hän on aiemmin onnistuneesti kirjautunut ja hänellä on siis validi json-web-token.

Ohjelma on vasta testivaiheessa ja siksi data näytetään vielä json -muodossa. Harjoitustyön neljännessä vaiheessa käytän jotain kivaa visualisointi kirjastoa yhdessä bootstrap:in kanssa ja laitan datan näkymään kuvaajassa. Ajattelin käyttää jotain valmista "bootstrap templatea", jonka avulla saan sivun näyttämään tyylikkäämmältä.

Helpot ja vaikeat toimenpiteet

Rajapinnan käyttäminen Node.js:n request -kirjaston avulla oli helppoa ja ei vaatinut montaakaan riviä koodia. Request -kirjasto on suunniteltu REST API -kutsujen tekemiseen ja se säästää hyvinkin paljon ohjelmoijien hermoja.

Datan lähettäminen clientille json -muodossa onnistui helposti koodipätkällä: `"res.json(body);"` jossa body sisälsi API:n lähettämän datan json -muodossa. Samalla data näkyi selaimessa oikealla tavalla sisennettynä, koska käytän chrome:ssa JSONView -extension:ia, joka sisentää automaattisesti json -tiedostot.

Fingridin API:lla oli hieman sekava dokumentaatio ja se vaikeutti aluksi käyttöönottoa. Sitten kun sain kasattua oikeanlaisen URL:in, niin loppu oli helppoa request -kirjaston avulla.

Hyödylliset lähteet ja linkit

- <https://www.npmjs.com/> (Node.js kirjasto manageri)
- <https://d3js.org/> (Hyvä JavaScript visualisointi kirjasto)
- <https://startbootstrap.com/templates/sb-admin/> (Hyvä bootstrap template)
- <https://data.fingrid.fi/fi/pages/api> (Fingridin API)