

Webservices

Tuki tiedon lukemiseen WEB Service rajapintojen ylitse

Tuki lisätty 6.2.2018:

Pääsääntöisesti RESTful datan lukeminen web service rajapinnan kautta ei tarvitse laisinkaan ohjelmointia, mikäli data tarjotaan XML tai JSON muodossa. Kahden edellä mainitun datamuodon lisäksi web service -toteutus osaa hakea tiedon n.s. RAW muodossa, joka tarkoittaa sitä että web palvelun tarjoamaa dataa ei pyritä parsimaan millään tavalla. Sen sijaan siihen voidaan soveltaa Lua:n merkkijonoille tarkoitettuja pattern-matching funktiota, joiden avulla pyritään löytämään tärkeät osuudet datasta.

Toiminta:

Web service rajapinta hyödyntää cURL kirjastoa ja komentorivityökalua.

Käyttö:

Web service rajapintaa käytetään luomalla 'webService' -tyyppinen piste tietokantaan. Kyseisiä pisteitä pollataan noin 5 sekunnin välein, ja mikäli pisteen kohdalla edellisestä kyselystä kauemmin kuin pisteen updateInterval -kentässä on määritelty, suoritetaan uusi kysely.

Pisteeseen täytyy syöttää kaksi arvoa, dataSource joka on web palvelun URL osoite, ja selects -taulukko, joka kertoo mitä arvoja datasta halutaan kerätä talteen. Ohjelma kerää web palvelun palauttamasta datasta selects -taulukon perusteella arvoja, ja kirjoittaa ne pisteen ".data" -taulukoon.

Esimerkki 1:

Luodaan webService tyyppinen piste "openWeatherMap"

web services URL osoite on:

<http://api.openweathermap.org/data/2.5/weather>

Lisäksi kyselyyn pitää asettaa parametrit appid (ikään kuin salasana), q (sijainti), sekä units (yksikkö)

query parametrit:

appid=a6d3f6617ad0c5ad24cc00d673ddceec&q=Pirkkala,fi&units=metric

Niinpä pisteen "openWeatherMap.dataSource" arvoksi tulee:

<http://api.openweathermap.org/data/2.5/weather?appid=a6d3f6617ad0c5ad24cc00d673ddceec&q=Pirkkala,fi&un>

Vastauksena saadaan JSON data:

```
{
  "coord": {
    "lon": 23.65,
```

```

    "lat":61.47 },
    "weather":[
      {
        "id":600,
        "main":"Snow",
        "description":"light snow",
        "icon":"13d"
      }
    ],
    "base":"stations",
    "main": {
      "temp":-13,
      "pressure":1009,
      "humidity":92,
      "temp_min":-13,
      "temp_max":-13
    },
    "visibility":10000,
    "wind":{
      "speed":0.5,
      "deg":40
    },
    "clouds":{
      "all":75
    },
    "dt":1517988000,
    "sys":{
      "type":1,
      "id":5045,
      "message":0.0045,
      "country":"FI",
      "sunrise":1517985075,
      "sunset":1518014928
    },
    "id":641491,
    "name":"Pirkkala",
    "cod":200
  }
}

```

Jos meillä on tarkoitus lukea luetusta datasta esimerkiksi:

ulkolämpötila, ulkokosteus, auringon nousuaika, auringon laskuaika ja pilvisuus

Luodaan "openWeatherMap.selects" taulukkoon seuraavat rivit (selite suluissa, sitä ei lisätä oikeasti pisteeseen):

```

#1 main.temp      (ulkolämpötila)
#2 main.humidity  (kosteus tieto)
#3 sys.sunrise    (auringon nousuaika)
#4 sys.sunset     (auringon laskuaika)
#5 clouds.all     (pilvisuus)

```

Syntaksi selects taulukossa on sama kuin Lua -kielessä taulukon soluihin viittaaminen!

Jos luetussa JSON datassa on taulukoita, soluihin viitataan numerolla [] sulkujen sisällä, ja indeksointi alkaa numerosta yksi (1), ei nollasta.

Selects taulukon viittaamat data solut löytyvät "openWeatherMap.data" -taulukosta samoilla indekseillä. Eli koska main.temp on ensimmäisellä rivillä .selects taulukossa, löytyy ulkolämpötila .data taulukon ensimmäiseltä riviltä, ja toisella rivillä on suhteellinen kosteus.

-

Esimerkki 2:

Parsitaan HTML sivulta Chuck Norris "faktoja".

Luo piste "ws/norrisFacts"

Aseta web services URL osoitteeksi:

<http://bleacherreport.com/articles/43450-humor-top-100-chuck-norris-facts>

Encoding parametriksi pisteeseen annetaan "RAW", joka tarkoittaa että vastauksena tulee suoraan HTML/teksti-tyyppistä dataa, eikä sille tehdä dekoddausta.

Selects taulukko tarvitsee vain yhden rivin, jonka arvoksi tulee "[%w%s%p]-".

Tämän jälkeen pisteen data taulukosta pitäisi löytyä noin 108 riviä. Niitä tutkimalla voidaan havaita, että 3 ensimmäistä, ja 13 viimeistä eivät ole oikeata dataa. Ne tulee jättää huomiotta kun dataa haetaan pisteestä

Luomme tässä esimerkissä vielä pisteen, joka noukkii satunnaisen faktan ".data" riveiltä.

Luo piste "norris/fact", joka on tyyppiä "none".

lisää siihen kenttä "script", ja anna sille arvoksi seuraava pieni skriptin pätkä:

```
if ( ( os.time() % 60 ) > 0 ) then return; end; local raw = Data.get ('ws/norrisFacts.data'); local n = (#raw or 0); local
```

Nyt pisteen pv -kenttään haetaan satunnainen Chuck Norris fakta, jonka voi vaikkapa näyttää käyttöliittymässä.

Revision #1

Created 30 May 2022 11:02:11 by Severi Hiltunen

Updated 10 June 2022 11:32:14 by Severi Hiltunen