

TLO-32400 Ohjelmallinen sisällönhallinta

Harjoitustyön vaihe 3

Tommi Honkanen

tommi.honkanen@tuni.fi

Muutokset aiempaan

Sain idean harjoitustyön toteutukseen vasta kolmannen vaiheen aikana, jonka takia tein rajun muutoksen käytettävien työkalujen suhteen ja vaihdoin alustaksi Django ja Pythonin tarvittavine kirjastoineen. Tietokantaratkaisuna toimii Django vakiototeutus Sqlite3.

Datan integrointi osaksi palvelua

Sovellus ei varsinaisesti käytä mitään virallista verkon yli toimivaa rajapintaa, vaan sen data on saatu ulkopuolisen sivuston kokoamista datadumpeista, jotka ovat csv-tiedostoja. Tiedostot sisältävät virtuaalisten esineiden markkinahintoja tietyllä aikajaksolla (league), ja tiedostot on koostettu aikajaksoittain. Sovelluksen ideana on visualisoida eri esineiden hintakehitystä ajan kuluessa, ja tarkoituksena on myös luoda aikasarjaan perustuva hintaennustaja.

Alla olevassa kuvassa on esitelty datadumpmien läpikäyntiin kirjoitettu skripti, jossa jokaisen aikajakson tiettyjen esineiden hintatiedot on otettu talteen ja kirjoitettu uudestaan csv-muotoon. Tietosisällön ja datan kontekstin vuoksi talteen otetaan hintatiedot vain yhden virtuaalisen valuutan suhteen, ja palvelussa esiteltävien esineiden määrää on rajattu paljon dumpmien koko sisällöstä.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Apr  4 17:08:31 2019
4
5 @author: Tommi Honkanen
6 """
7
8 import csv
9 import glob
10 import os
11
12 currencies = ["Exalted Orb", "Orb of Annulment", "Orb of Transmutation", "Orb of Chance", "Silver Coin", "Orb of Alchemy", "Chromatic Orb", "Orb of Regret",
13              "Orb of Scouring", "Orb of Alteration", "Blacksmith's Whetstone", "Orb of Augmentation"]
14
15 write_rows = []
16 data_path = "data/originals"
17
18 files = glob.glob(data_path + os.sep + "*.csv")
19
20 for name in files:
21     write_rows = []
22
23     # We loop through all .csv dump files which contain the price information for leagues
24     with open(name) as csv_file:
25         csv_reader = csv.reader(csv_file, delimiter=',')
26         line_count = 0
27         for row in csv_reader:
28             if line_count == 0:
29                 print(f'Column names are {", ".join(row)}')
30                 write_row = ["League", "date", "Currency", "PriceCurrency", "Value", "CurrencyID"]
31                 write_rows.append(write_row)
32                 line_count += 1
33
34             # for now we are using only information that's based on chaos orbs, every currency has a value that's based on it
35             else:
36                 if row[3] == "Chaos Orb" and row[2] in currencies:
37                     index = str(currencies.index(row[2]))
38                     line_count += 1
39                     write_row = [row[0], row[1], row[2], row[3], str(row[4]), index]
40                     write_rows.append(write_row)
41
42         print(f'Processed {line_count} lines.')
43         base = os.path.basename(name[:-4])
44         out_name = "data/" + base + "_currencies.csv"
45
46         # We write the parsed price data to new .csv files which are then read to the database
47         with open(out_name, mode='w', newline='') as currency_file:
48             currency_writer = csv.writer(currency_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
49
50             for row in write_rows:
51                 currency_writer.writerow([row[0], row[1], row[2], row[3], row[4], row[5]])
52
53
54
55

```

Kun haluttu hintadata on saatu kerättyä, se kirjoitetaan uudestaan muotoon, josta se voidaan lukea palvelun tietokantaan. Tätä varten kirjoitin toisen skriptin, jossa Django-projektin Sqlite3-tietokantaan otetaan yhteyttä ja sinne kirjoitetaan haluttujen tiedostojen data.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Apr  4 19:51:32 2019
4
5 @author: Tommi Honkanen
6 """
7
8 import sqlite3
9 import pandas as pd
10 import glob
11 import os
12
13 # We create a connection to the project's sqlite3 database and assign a cursor for it
14 conn = sqlite3.connect('db.sqlite3')
15 c = conn.cursor()
16
17 data_path = 'data'
18
19 files = glob.glob(data_path + os.sep + "*.csv")
20
21 # We go through all parsed .csv files and used pandas library to read the data from the files
22 # and then insert it to the database
23 for name in files:
24
25     read_clients = pd.read_csv (name)
26     read_clients.to_sql('datavis_currencydata', conn, if_exists='append', index = False)
27
28

```

Palvelun runko on tavallinen Django-projekti, johon olen luonut erillisen Django app-sovelluksen, joka sisältää palvelun reitit, näkymät ja datamallit. Models.py sisältää tietokantataulun, johon on määritelty palvelun dataa vastaavat kentät. Palvelun reitit ovat tällä hetkellä rajattu localhost:8000/datavis -sivulle, jossa html-dokumentin sisältämään taulukkoon luetaan tietokannan sisältämä data.

League is Abyss

Below you can see data for the league.

	League	Date		Currency		Price in chaos
Abyss		Dec. 9, 2017		Exalted Orb		26.166439999999998
Abyss		Dec. 10, 2017		Exalted Orb		35.92556
Abyss		Dec. 11, 2017		Exalted Orb		43.5739
Abyss		Dec. 12, 2017		Exalted Orb		47.99698
Abyss		Dec. 13, 2017		Exalted Orb		45.45841
Abyss		Dec. 14, 2017		Exalted Orb		41.32816
Abyss		Dec. 15, 2017		Exalted Orb		42.488890000000005
Abyss		Dec. 16, 2017		Exalted Orb		41.789229999999996
Abyss		Dec. 17, 2017		Exalted Orb		41.5251
Abyss		Dec. 18, 2017		Exalted Orb		45.1072
Abyss		Dec. 19, 2017		Exalted Orb		60.965109999999996
Abyss		Dec. 20, 2017		Exalted Orb		69.671219999999999
Abyss		Dec. 21, 2017		Exalted Orb		71.29334
Abyss		Dec. 22, 2017		Exalted Orb		74.122
Abyss		Dec. 23, 2017		Exalted Orb		79.050209999999999
Abyss		Dec. 24, 2017		Exalted Orb		85.535
Abyss		Dec. 25, 2017		Exalted Orb		90.74893
Abyss		Dec. 26, 2017		Exalted Orb		85.32706
Abyss		Dec. 27, 2017		Exalted Orb		80.56075
Abyss		Dec. 28, 2017		Exalted Orb		82.916980000000001
Abyss		Dec. 29, 2017		Exalted Orb		83.99195
Abyss		Dec. 30, 2017		Exalted Orb		84.24766
Abyss		Dec. 31, 2017		Exalted Orb		85.36966
Abyss		Jan. 1, 2018		Exalted Orb		85.3875
Abyss		Jan. 2, 2018		Exalted Orb		87.69038
Abyss		Jan. 3, 2018		Exalted Orb		90.22788
Abyss		Jan. 4, 2018		Exalted Orb		91.21405
Abyss		Jan. 5, 2018		Exalted Orb		90.675119999999999
Abyss		Jan. 6, 2018		Exalted Orb		91.116819999999999
Abyss		Jan. 7, 2018		Exalted Orb		90.393819999999999
Abyss		Jan. 8, 2018		Exalted Orb		90.05419
Abyss		Jan. 9, 2018		Exalted Orb		89.946930000000001
Abyss		Jan. 10, 2018		Exalted Orb		89.87451
Abyss		Jan. 11, 2018		Exalted Orb		89.6155
Abyss		Jan. 12, 2018		Exalted Orb		88.813330000000001
Abyss		Jan. 13, 2018		Exalted Orb		86.78846

Helpot ja vaikeat asiat

- Helppoa: Django-projektin alustaminen. Django on web frameworkina äärettömän helppokäyttöinen ja monipuolinen, ja toimivan serverin kevyellä html-sivulla saa aikaan noin viidessä minuutissa. Lisäksi laajempaan palveluun tarvittavia toiminnallisuuksia voi hyvin helposti toteuttaa Django mukana tulevilla kirjastoilla.
- Vaikeaa: Datan lukeminen manuaalisesti Sqlite3-kantaan ilman Djangoa/tietokantaan kuuluvaa serveriä. Normaalisti Django käyttämään kantaa voisi populoida helposti admin-paneelin tai erillisen näkymän ja API:n yhdistelmällä, mutta tässä tapauksessa kannan täyttö tapahtui offline-tilassa. Django models.py:n perusteella luomat taulut ovat nimittäin kantaan luodessa omalla erillisellä nimellä, jonka selvittämiseen ja kannan toiminnan testaamiseen löysin onneksi DB Browser-nimisen ohjelmiston, joka on suunniteltu Sqlite-kannoille. Yrittäessäni populoida datataulua se ei aluksi tähtynyt väärän nimen takia, ja Python ei antanut tästä erillistä ilmoitusta.
- Helppoa: Tietokannan sisällön verkkoon siirtäminen. Django MTV-paradigma on yksinkertainen tapa esittää mitä tahansa tietoa serveriin liitetystä kannasta. Yhden näkymän ja html-tiedoston avulla kannasta voi lukea annetuilla parametreilla dataa, ja sen voi liittää html-elementteihin Django template-menetelmillä.

Hyödylliset linkit

Djangon models-toteutus <https://docs.djangoproject.com/en/2.2/topics/db/models/>

Sqlite3-kannan luonti Pythonilla <https://datatofish.com/create-database-python-using-sqlite3/>

Satunnaiset ohjeet Django-projektiin ja Sqlite3-kantaan <https://stackoverflow.com>