Учебен център Нет Ит

Курс по програмиране на Python

Курсов проект

на

Валентин Красимиров Мандев

Тема: „Проектиране и разработване на програма събираща информация относно песни от жанра Eurobeat от различни източници и представяща я в удобен за потребителя вид”

София 2020

Съдържание

[Увод 3](#_Toc30494086)

[1 Въведение 4](#_Toc30494087)

[2 Проектиране 5](#_Toc30494088)

[3 Разработване 8](#_Toc30494089)

[4 Експерименти и тестове 11](#_Toc30494090)

[Заключение 12](#_Toc30494091)

[Източници 13](#_Toc30494092)

# Увод

Целта на проекта е да се разработи опростен графичен интерфейс, който използва няколко файла с Python код, за да запълни база данни с информация за песни – линк към уеб-страница с текст, която в по-следващ етап да бъде достъпена, за да се вземе текста, година на издаване, музикална компания, продуцент, продължителност на песента и линк за търсене в YouTube.

Информацията се взема от два различни източника, като чрез едно просто натискане на бутон в главната страница на графичния интерфейс базата данни се ъпдейтва в рамките на 2-3 минути.

Включена е търсачка, където потребителя може да потърси песни по име, изпълнител, музикална компания или година на издаване.

При избор на конкретна песен се отваря втория екран – той съдържа информацията отбелязана по-горе, както и текста на песента.

# Въведение

Основният проблем на жанрове като Евробийт-а (с по-ниска популярност, ограничени до тесен кръг любители и с продуценти и изпълнители работещи предимно за удоволствие) винаги е бил липсата на достатъчно информация, дори и в наши дни, когато сме свикнали всичко от което се интересуваме да е на няколко клика или едно бързо търсене в Google.

За щастие все пак съществува уеб-сайта Eurobeat-prime.com, създаден и поддържан от фенове, които пък комуникират и споделят информация във форума и Discord канала. Така или иначе няма единен източник, който да позволява удобен и бърз достъп до достатъчно информация – обикновено се рови в Discogs, фейсбук страници на музикални компании, изпълнители и продуценти, разпитват се познати и т.н.

Макар и да няма претенции да обхваща всички разпилени в пространството данни, направих приложението Eurobeat song guide така, че да се свързва с два различни източника и да успява да достъпи наличната информация за максимален брой песни, за да могат потребителите, да намират максимално бързо и лесно информация с минимална консумация на интернет трафик.

От една страна липсата на API ключ за сайта на Eurobeat-prime ме накара да използвам библиотеката bs4 (BeautifulSoup) за достъп до отделни страници, от които да взема линкове до страниците с текстове, което наложи повърхностно запознаване с HTML, за да успея да изпълня задачата си, нещо което така или иначе ще ми е от полза в бъдещото ми развитие като програмист.

В същото време пък вторият източник на информация – Spreadsheet в Google Drive предостави сравнително лесен достъп от програмата ми чрез библиотеки вършещи чудесна работа за четене и извличане от ексел файлове.

Макар и събрани, данните от двата различни източника имат нужда и от последваща обработка, за да ги преведа в подходящ вид, който да представя във финалния аутпут.

Също така за голям брой от песните информация просто липсва – дали ще е година на издаване или продуцент, или пък нещо друго. Но пък източниците постоянно се обновяват, добавят се и нови песни поради тази причина добавих възможността да ъпдейт на базата данни чрез бутон в графичния софтуер.

# Проектиране

В завършен вид програмата представлява търсачка – в нея потребителя може да въведе име на песен, изпълнител, музикална компания или година. Проектирал съм търсачката така, че да не е case-sensitive. Освен това може да се въведат едновременно име на изпълнител и песен за по-бързо и лесно намиране на конкретна песен. Работи както с натискане на клавиша „Enter”, така и с кликване върху бутона „Search”.

Също така съм сложил опции за избиране на прозиволна песен, година или пък музикална компания, в случай че потребителя не търси нещо, а предпочита да бъде изненадан.

A screenshot of a cell phone

Description automatically generated

Без значение дали се използва търсачката или някой от random бутоните, на екрана се появява меню, в което като опции излизат всички песни отговарящи на търсенето. При избор на една от тях се отваря втория екран с информация. Ако няма намерени песни, то излиза текст подканващ потребителя да опита отново.

Екрана с информация за песента съдържа година на издаване, музикална компания, продуцент, автор, продължителност на песента, линк за търсене на песента в YouTube и текст. В скийншота по-долу виждаме, че цялата информация е налична за тази песен, но има случай, когато в базата данни липсват неща.

A screenshot of a cell phone

Description automatically generated

След това потребителя може да се върне към основния екран с „Back to search screen”.

Основния екран също така съдържа часовник и дата, както и бутон за ъпдейт на базата данни, който извлича отново информация и от двата източника и я обединява в едно. Самия процес отнема 2-3 минути и е задължително да се направи при първо пускане на програмата на даден компютър.

След края излиза съобщение за успешно обновяване и приканване за рестартиране на приложението.

A screenshot of a cell phone

Description automatically generated

Приложението се затваря стандартно с натискане на X горе вдясно на който и да е от двата екрана.

# Разработване

Използвани библиотеки:

* urllib.request
* bs4
* os
* openpyxl
* sqlite3
* pandas
* re
* random
* tkinter
* webbrowser
* datetime
* time

Програмата се състои от пет файла (по ред на създаване):

1. get\_titles\_and\_links.py

Тук достъпвам сайта [www.eurobeat-prime.com](http://www.eurobeat-prime.com) Първо си създавам списък със страниците съдържащи текстовете на песни. Страниците са според първата буква от името на изпълнителя. След това минавам през всички страници (общо 27) и използвайки BeautifulSoup взимам имената на песните, изпълнителите и линкове към страниците с текстове на песните. Всяка песен си има ID в сайта, което в крайна сметка използвам и в моята база данни.

Правя си списък с всички получени данни дотук, който впоследствие да импортирам.

1. get\_info\_from\_excel\_database.py

Чрез втория скрипт свалям екселска база данни с голям брой шийтове от Google Drive. След това ползвам библиотеката openpyxl, за да взема необходимата информация от файла. Минавайки през всеки един шийт пълня списък с данни от съответните колони.

1. update\_songs\_database.py

Тук събирам данните от двата източника в една обща база. За целта първо проверявам кои песни се срещат и на двете места ползвайки формат „Име на изпълнител – Име на песен”. За тези, които съвпаднат просто добавям данните от екселския файл към тези от сайта. Отделно си пълня друг списък с имената на общите, за да ги изтрия от списъка с екселски данни, тъй като съм ги ползвал и вече не ми трябват.

След това обработвам двата списъка поотделно – поставям празни стрингове за екселските елементи за песните, които се срещат само в сайта и обратно.

Сега вече мога да събера двата списъка, като съм сигурен, че всяка песен се съдържа по веднъж и има нужната информация или празен стринг за всеки един показател, който ми трябва.

Накрая от този обобщен списък, включващ всички песни и данни за

ID, изпълнител, име на песента, линк към страницата с текст, линк за търсене в YouTube, година на издаване, продължителност, автор, продуцент и музикална компания си създавам SQLite база.

Рънването на този файл води до изпълнение и на предходните два и до цялостен ъпдейт на базата данни.

1. search\_for\_a\_song.py

Този файл съдържа функциите по намирането на определена песен и представянето на наличната информация за нея. Повече информация в следващата точка.

1. gui.py

Основния файл, който рънваме – графичен интерфейс, който използва останалите 4 скрипта. Както вече споменах, имаме два прозореца – един за търсене и друг – за информация за песните. Направил съм ги в 2 отделни функции, като прехвърляме от едната към другата, когато сменяме екраните.

Без значение дали ползваме полето за въвеждане на текст или random бутоните – принципа на действие е един и същ – при въведения инпут се изпълнява функцията show\_info от search\_for\_a\_song.py. Използвам RE, за да направя така, че търсенето да не е case sensitive. Възможните начини за търсене, които съм включил са по име на песен, изпълнител, година, музикална компания или комбинация от изпълнител и име на песен. Пак ползвам списък – с всички песни, които отговарят на търсенето, за да може потребителя да избере тази, която го интересува.

При избор на дадена песен се записва временен файл с името ѝ, за да намерим в базата данни и да вземем нужната информация чрез print\_song\_info функцията. Вадя си ID-то на песента и чрез BeautifulSoup отварям страницата в сайта, на която е текста (стига да има такъв) и го обработвам, за да премахна излишните HTML символи и празни редове и да го представя в изчистен вид. Вземам и останалата част от информацията.

Отваря се втория екран – където излиза цялата налична информация и единствено един бутон – за връщане към екрана за търсене.

За random нещата ползвам random.choise.

При създаването на всички текстове и бутони ползвах класове.

Включил съм дата и час, които се рефрешват постоянно (датата на една секунда, а часът на 200 милисекунди).

Кликването върху бутона за ъпдейт води до изпълнение на скрипта „update\_songs\_database.py“ чрез os.system('update\_songs\_database.py')

Както вече споменах е нужен рестарт след ъпдейт, понеже базата данни е отворена и няма как да бъде презаписана. Затова записвам друг .db файл с различно име, а при рестарт се изтрива стария и се преименува новия преди да бъде отворен.

# Експерименти и тестове

От началото на проекта не спрях да правя тестове при създаването на всеки един скрипт, които ми показваха различни неточности, които успявах да оправя.

Затрудни ме например да докарам списъците с двата вида данни до такова положение, че песните във всеки един от тях да имат еднакъв брой елементи от информация и подредени по един и същ начин, за да мога да ги merge-на правилно. След определен брой опити се получи.

Друг проблем беше невъзможността да се събере целия текст на песента, в момента съм го направил с минимален шрифт, но все още достатъчен, за да се чете лесно и ясно на екрана на 14-инчовия ми лаптоп.

Също така имах известни затруднения с продължителността на песента. Излизаше ми във формат hh:mm:ss, а за тези, за които нямаше данни стандартно беше None, но също така за някои песни имаше (LP), (Z) или други стрингове. В крайна сметка ползвах exception, за да проверя, кои започват с цифра, тоест има информация, а за всички останали записах текст 'Song duration: Unknown'.

При търсенето в YouTube, за песни/изпълнители съдържащи знака „&”, текста за търсене се изрязваше до този знак. Примерно за Dave & Domino – Sunshine In Your Eyes ми търсеше единствено „Dave”. Оправих го като единствено за youtube линковете се сменя „&” с „and”.

При затварянето на екрана с информация за песента не се затваряше програмата, а се отваряше отново главния прозорец. Това оправих като просто добавих exit() в края на window2 функцията.

За проблема с презаписването на базата данни и решението чрез рестарт вече писах по-горе.

# Заключение

Проекта със сигурност ми помогна да преговоря голяма част от изучения материал, както и да придобия нови познания. Радвам се, че успях да изпълня плана си и да създам нещо от нулата, без готови темплейти и копирани програми, за да усетя през какви стъпки се минава, какви проблеми мога да срещна и как трябва да сглобя отделните парченца в едно цяло. Това със сигурност ще ми помогне в бъдеще, макар че едва ли ще се случва често да започна без някакви предварително подготвени материали и скриптове.

Идеята ми оттук нататък е да доразвия приложението, търсейки информация от още източници, да добавя още страници с информация за изпълнители и новини, евентуално дори за вграждане на видео/аудио.

Бих желал да го направя също така и под формата на уеб-приложение.

Лично аз като любител на жанра вече си го ползвам като удобен и бърз източник на информация.

# Източници

Нет ИТ курс по Пайтън

Денис Колесниченко – „Python – практическо програмиране“

[www.docs.python.org](http://www.docs.python.org)

<https://stackoverflow.com/>

<https://realpython.com/python-gui-tkinter/>

<https://www.geeksforgeeks.org/python-gui-tkinter/>

<http://www.pythonlake.com/tkintertkgeometry>