



Проект за курс CITV331 Програмиране с Python

Преподавател: гл. ас. д-р Ф.Андонов

Автор: Радослав Бонев

Фак. номер: F88092

Идея

Приложение с графичен интерфейс, което взема информация от REST API за телевизионната програма. Приложението дава възможност при клик на някое предаване, в часът му на започване, потребителят да получи нотификация на компютъра си, за да не пропусне предаването.

Технологии

- Python
- WX Python

Екрани

- Начален - екран, на който се визуализират изображенията на доставчиците на телевизионни услуги, които се поддържат от приложението. Доставчиците се взимат от REST API-то. То е json масив, като всеки обект задължително осигурява url към графично изображение в мрежата, което се визуализира. Примерен отговор изглежда така:

```
[
  {
    "id": 1,
    "name": "BNT",
    "image": "https://i.ibb.co/NWND035/BNT.png"
  },
  {
    "id": 2,
    "name": "BTV",
    "image": "https://upload.wikimedia.org/wikipedia/commons/8/8f/BTV-logo.png"
  },
  {
    "id": 3,
    "name": "NOVA",
    "image": "https://jobs.gaijinpot.com/employer/image/index/size/300x300/id/4721"
  }
]
```

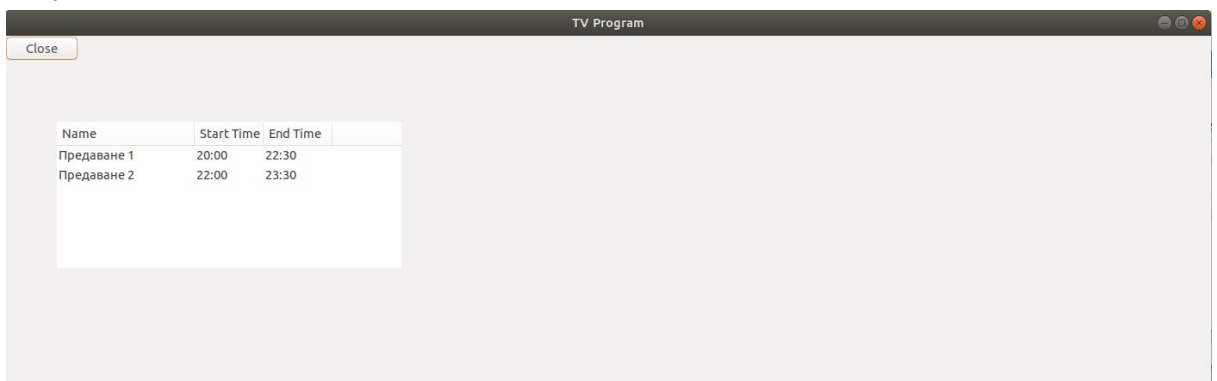
Екрана изглежда по този начин:



- Екран на програмата - За конкретен provider_id (id), който също се съдържа в отговора на първото API, се извиква нова заявка. Тя връща отговор, подобен на този:

```
[
  {
    "id": 1,
    "name": "Предаване 1",
    "start_time": "20:00"
  },
  {
    "id": 2,
    "name": "Предаване 2",
    "start_time": "22:00"
  }
]
```

Визуално екрана изглежда така:



Основни моменти в технически аспект

- REST API

За да бъдат динамични и актуални данните, беше необходимо да се намери API, което ги предоставя. При направено проучване не установих съществуване на такова публично API. За това съм създам python логиката, която набавя тези данни, за

която става дума по-късно. Тя обработва информация и я съхранява в база данни. Крайният резултат може да се достъпи като се извикват с метод GET на url-и:

<https://demo1623113.mockable.io/providers>

<http://46.101.233.213/programs/2>

● Обработка на данни от страница

Както вече стана дума, не беше установено наличие на публично API, което да предоставя информация за телевизионната програма. За целта, всеки път, когато се стартира програма, се изпълнява и Python логика, която влиза в уеб страницата <https://www.btv.bg/programata/> и според вече проучената вече структура на html страницата, я обхожда и събира данни за име на програмата и началното време. Обхождането взима за отправна точка конкретни html тагове и техни класове. Така се постига възможно най-голяма актуалност на данните, тъй като идват от сайта на медията директно и се обновяват при стартиране на програмата. За обработката на html-а се използва библиотеката [BeautifulSoup](#). След това така събраните данни се изпращат към API с метод POST, което е написано на PHP и е вързано с MySQL база данни. Тялото на заявката е JSON-форматирано и представлява масив от предавания, съдържащи име и начален час. При всяко добавяне, автоматично се изтрива предишното съдържание на таблицата. Сървърът е droplet от DigitalOcean. Това е пример от моментното съдържание на таблицата в базата данни. Това съдържание, както стана дума, е събрано от Python скрипта.

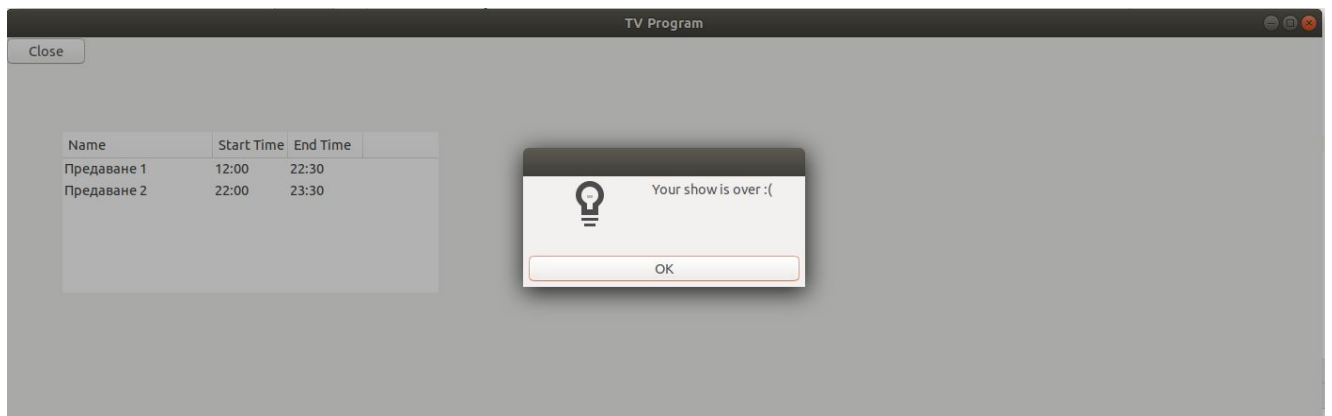
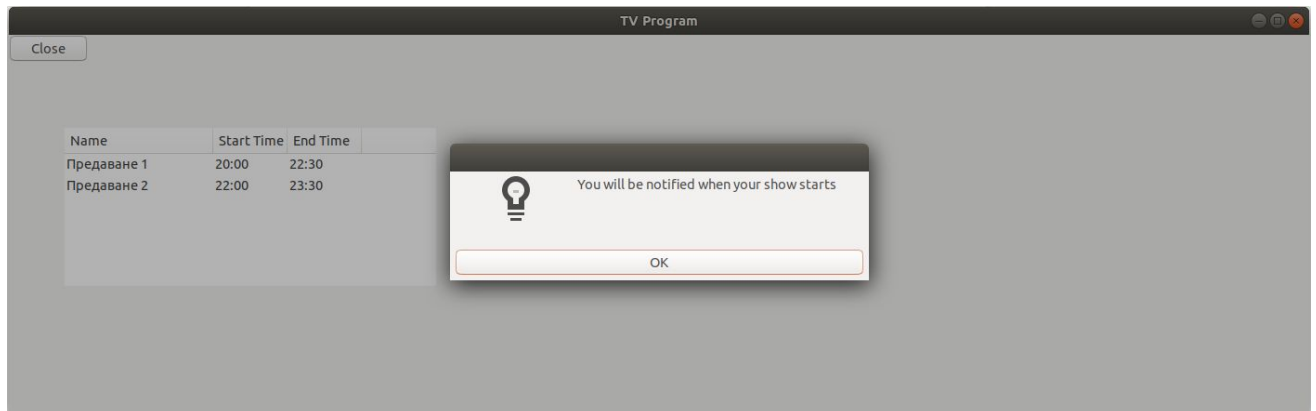
#	id	name	start_time	created_at
1	1	Тежки престъпления	00:00	2019-06-11 22:26:29
2	2	Вечно твоя - Индия	01:00	2019-06-11 22:26:29
3	3	bTV Новините /п./	02:10	2019-06-11 22:26:29
4	4	Преди обед	02:40	2019-06-11 22:26:29
5	5	Опасни улици	04:40	2019-06-11 22:26:29
6	6	Лице в лице	05:30	2019-06-11 22:26:29
7	7	Трансформърс: Роботи под...	06:00	2019-06-11 22:26:29
8	8	Тази сутрин	06:30	2019-06-11 22:26:29
9	9	Преди обед	09:30	2019-06-11 22:26:29
10	10	bTV Новините	12:00	2019-06-11 22:26:29
11	11	Шоуто на Слави	12:30	2019-06-11 22:26:29
12	12	Втори шанс	13:30	2019-06-11 22:26:29

● Графичен интерфейс

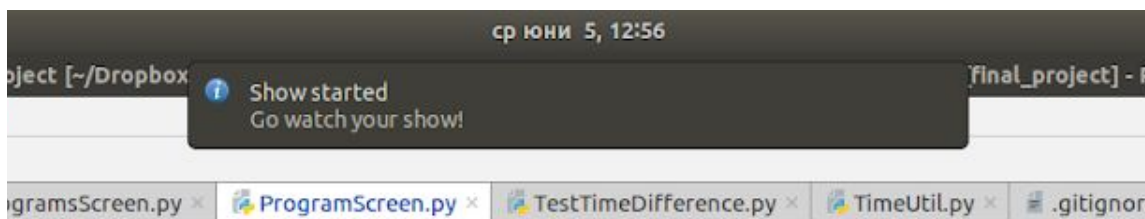
За удобство на потребителя е реализиран графичен интерфейс с помощта на wxPython. Използвани са доста класове, като част от елементите се визуализират динамично. Например изображенията на доставчиците са толкова на брой, колкото са и обектите в масива на отговора. Използван е и клас, който създава списък с елементи, които могат да бъдат избирани.

- Нотификации

При избиране на предаване от екрана за програми, ще бъдете уведомени, че ще получите нотификация в момента, в който предаването ще започне. Ако предаването започва в час, по-ранен от текущия час за системата, ще се появи описателно съобщение за грешка.



Самото изпращане на нотификации се извършва чрез вградена в Python 2.7 библиотека - subprocess. Нотификацията се изпълнява като се задава Timer, на който е зададено сметнатото оставащо време до предаването. Под Ubuntu 18.04 нотификацията изглежда така:



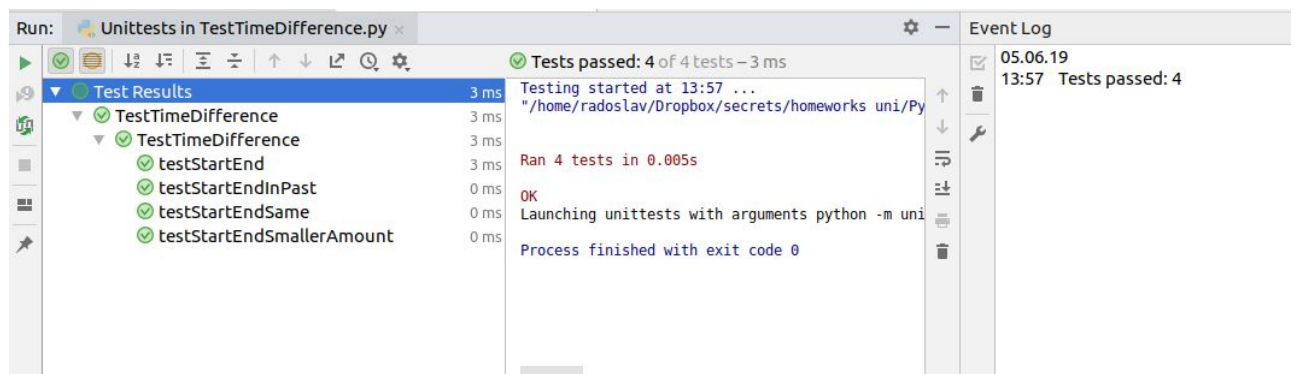
- Смятане на оставащото време

Логиката за смятане на оставащото време е отделена в utility клас. Тя работи така: тя приема два часа във формат "HH:mm", parse-ва ги до число и вади началния

от крайния. Резултатът е цяло число, което е минути до предаването. Това число може да бъде 0 или отрицателно, което означава, че предаването започва сега или вече е минало. Тогава се появява съобщението за грешка, описано горе. Тъй като апито на Timer очаква секунди, минутите трябва да се умножат по 60. Всичките необходими константи са отделени във файл с цел четимост и сегментация на кода.

- Unit-тестове

Логиката за смятане на оставащото време е отделена в малки методи, за да може да бъде лесно тествана. Реализирани са unit-тестове, които дават различни часове на методите и очакват конкретен резултат. Това е резултатът е от стартирането на unit тестовите:



- SVC (System Version Control)

В процеса на разработка е използван GIT и кода и историята му на развитие може да бъде видяна в хранилището:

<https://github.com/rado9818/TV-programs>

Резултат

Резултатът е приложение с графичен интерфейс, което може да бъде захранено с данни от мрежата. Потребителят може да си зададе напомняне, когато започва дадено предаване. Приложението е обезпечено с тестове и в процеса на разработка е използвана система за контрол на версиите, за да може да се управлява по-лесно процесът на разработка.