



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"
  }
]
```

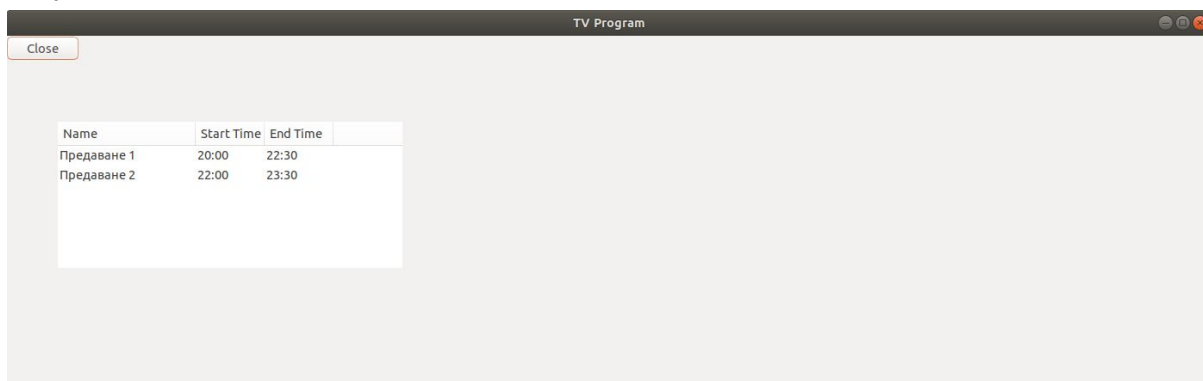
Екрана изглежда по този начин:



- 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:

```
[
  {
    "name": "Предаване 1",
    "start_time": "20:00",
    "end_time": "22:30"
  },
  {
    "name": "Предаване 2",
    "start_time": "22:00",
    "end_time": "23:30"
  }
]
```

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, което да връща програмата на нито една българска телевизия. За целта съм създам такова с mock-данни. Тъй като REST API-то е извън контекста на

проекта, то е реализирано чрез платформата <https://www.mockable.io>. То винаги връща едни и същи отговори. За да се достъпят апитата, се извикват с метод GET на url-и:

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

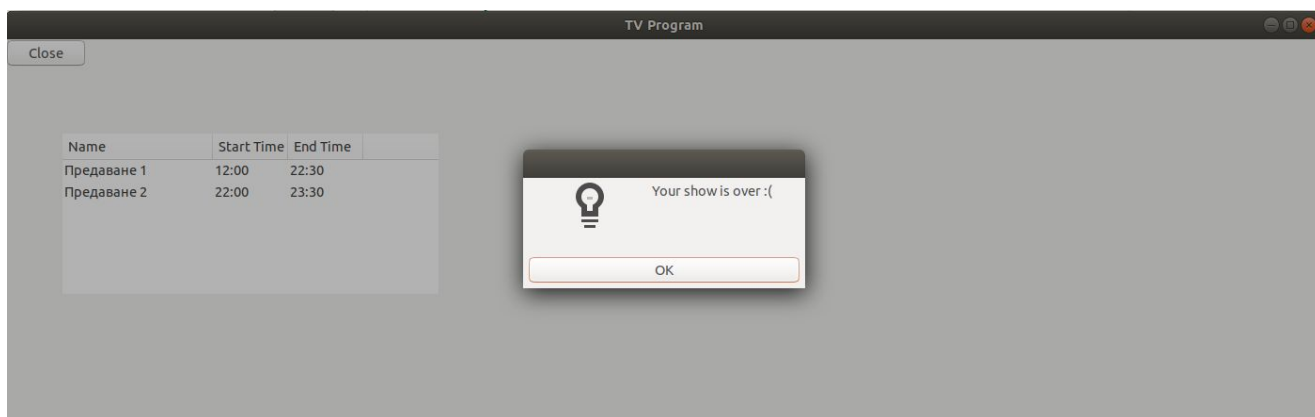
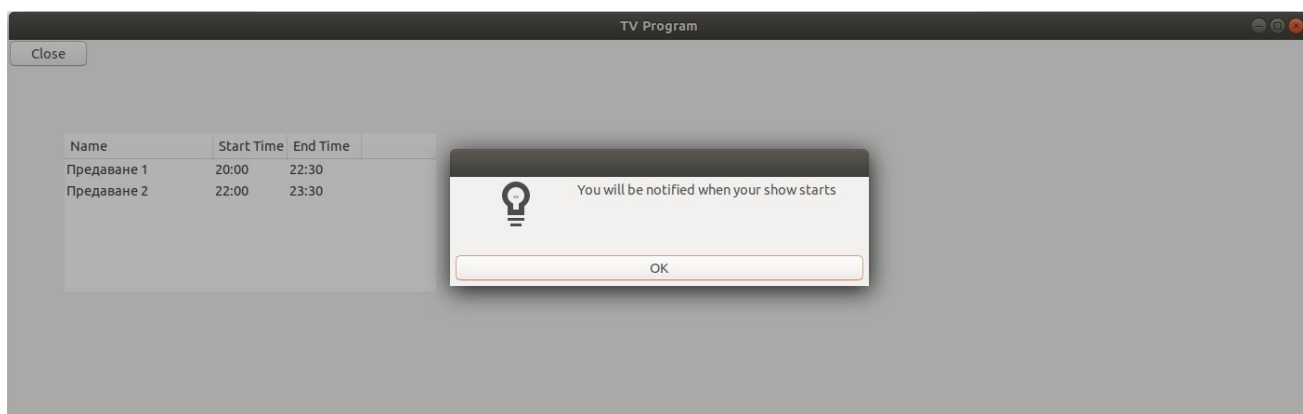
<https://demo1623113.mockable.io/providers/{id}/program>

- **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.

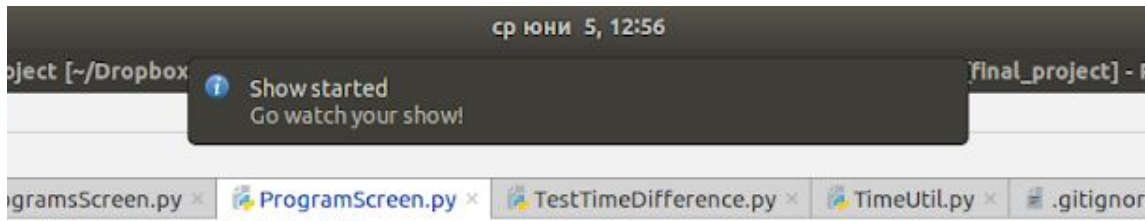
- **Нотификации**

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:

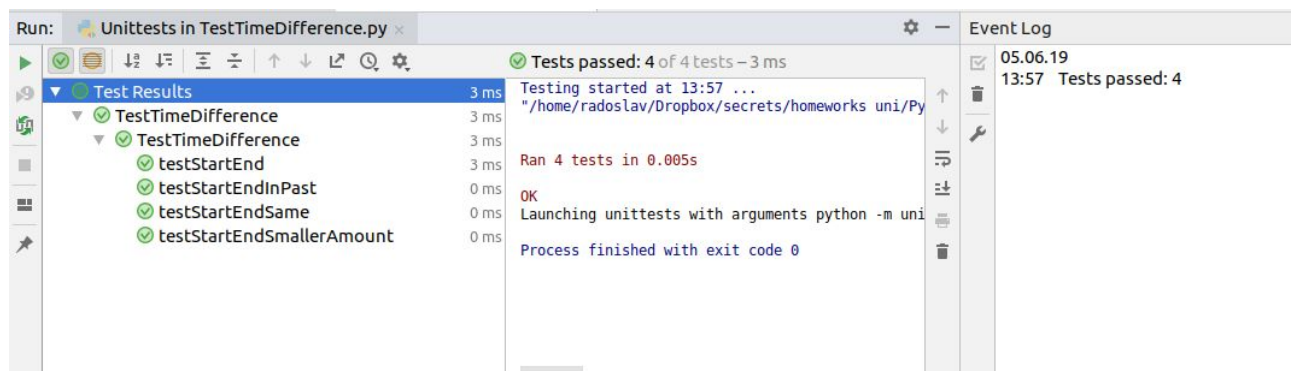


- **Смятане на оставащото време**

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.