Python Practice

Реализация api Приватбанка. Логирование

Автор курса

Кременtарь Ксения

Ведущий Python разработчик

Системный архитектор

в компании K-Solutions

После урока обязательно

Повторите этот урок в видео формате на ITVDN.com

Проверьте как Вы усвоили данный материал на TestProvider.com

Реализация api Приватбанка. Логирование

На этом уроке

1. Добавим логирование в проект
2. Добавим модуль config.py для хранения констант настроек
3. Добавим модуль с реализацией api Приватбанка
4. Рассмотрим, что такое парсинг
5. Напишем тесты для нового модуля

Логирование

Логирование - это процесс фиксации действий, совершаемых в коде скрипта, приложения, проекта. Фиксация может осуществляться в консоль, файл, базу данных.

Цели логирования:

- отражение хода выполнения программы, независимо от того, идет все штатно или возникла ошибка
- источник информации для дальнейшего анализа работы приложения - количество ошибок, время выполнения того или действия, общая нагрузка на приложение и так далее, анализировать можно в различных направлениях и плоскостях, админы будут благодарны за грамотные логи
- сохранение максимальной информации для анализа причины ошибки

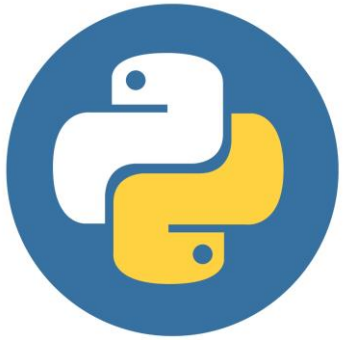
Основные принципы построения логирования

- Отображение основных событий приложения - запуск-остановка, начало-конец обработки запросов, возникновение ошибок
- Правильное использование различных уровней логирования, DEBUG, INFO, WARNING, ERROR, FATAL для логирование различных событий
- Все логи должны соответвовать определенному формату
- Должны быть легко анализируемыми, давать понимание об источнике записи лога
- Одна строка — одно сообщение.
- Логи должны быть лаконичными, стоит избегать избыточности, особенно для уровня логирование выше DEBUG.
- Стоит избегать сложной логики, выполнения каких-либо действий при логировании - основная задача просто отобразить (значение переменной, ее тип, текст ответа и тп)

Модуль logging

Стандартный модуль для организации логирования в языке Python.

https://docs.python.org/3.6/library/logging.html



Добавление логирования в файл

1. Создать объект логгера с помощью logging.getLogger(logger_name)
2. Создать объект-обработчиков логов, хендлер. Выбираем логирование в файл, поэтому нужно указать и имя файла с помощью logging.FileHandler(file_path)
3. Установить уровни логирования у логгера и хендлера
4. Указать нужный формат логов - с помощью logging.Formatter(format)

```
import logging

log = logging.getLogger("LoggerName")
fh = logging.FileHandler("app.log")
fh.setLevel(logging.DEBUG)
fh.setFormatter(logging.Formatter(
    "%(asctime)s [%(levelname)s] - %(name)s:%(message)s"))
log.addHandler(fh)
log.setLevel(logging.DEBUG)

log.debug("Hello from app!")
```

Модуль config.py

Параметры для конфигурация логгера, путь к файлу базы данных и другие настройки рекомендуется выносить в отдельный модуль, для удобства управления ими. Обычно такой модуль с конфигурационной информацией называют config.py

```
import logging


DB_NAME = "golden-eye.db"

LOGGER_CONFIG = dict(level=logging.DEBUG,
                     file="app.log",
                     formatter=logging.Formatter("%(asctime)s [%(levelname)s] - %(name)s:%(message)s")
                     )
```

Двигаемся дальше!

Необходимо реализовать новый модуль, который будет обновлять курс USD -> UAH в базе, предварительно получая новое значение курса по api Приватбанка.

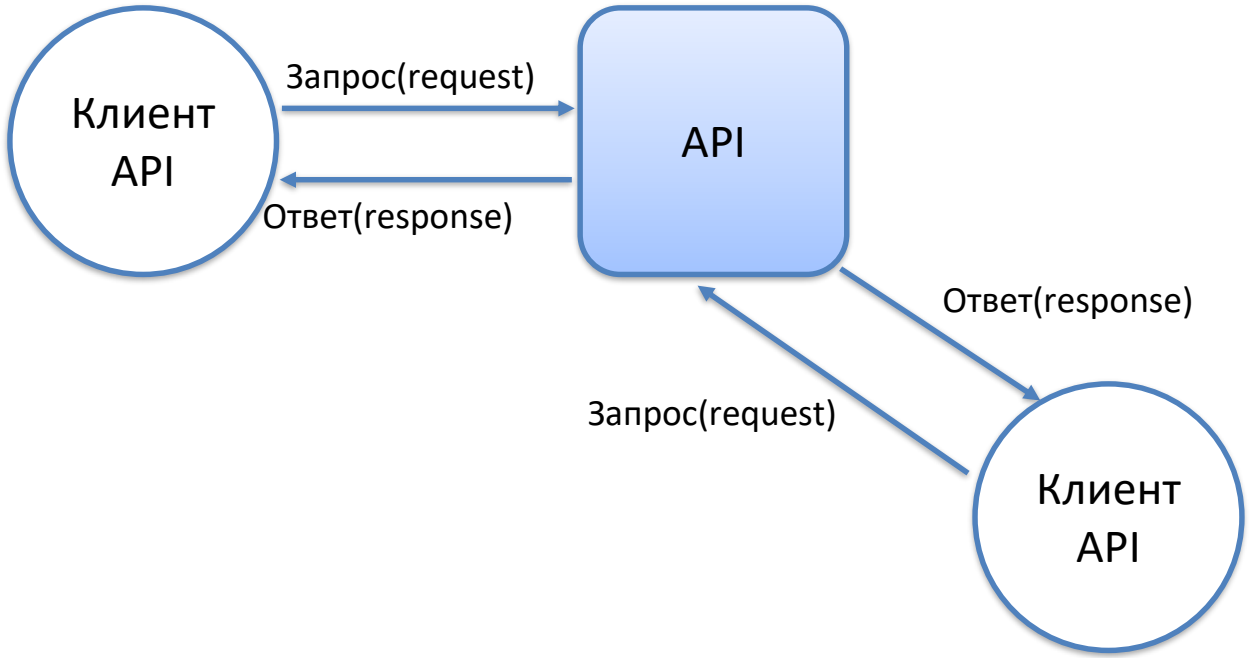
Что такое api Приватбанка?

Как получать информацию по api?

Какие python библиотеки нам понадобятся?

WEB API

Web API - специальные службы, доступные онлайн.



Все web api имеют:

- URL, точка входа (идентифицируют api)
- Протокол (правила) взаимодействия

Описание API Приватбанка

Правила: https://api.privatbank.ua/#p24/exchange

Точка входа: https://api.privatbank.ua/p24api/pubinfo?exchange&json&coursid=11

Необходимо отправить GET запрос!

Библиотеки для отправки запросов

Стандартные

- http.client https://docs.python.org/3.6/library/http.client.html
- urllib https://docs.python.org/3.6/library/urllib.html#module-urllib

Pythonic, простая
Requests

Библиотека requests

Мощнейшая библиотека, де-факто ставшая стандартом для отправки http запросов в Python

Основные преимущества:

- Удобство
- Простота
- Красота

http://docs.python-requests.org/

Отправка GET запроса

Вызов метода requests.get(url), в качестве параметра указать url удаленного API.

Результатом выполнения будет объект requests.Response, который содержит текст ответа, хедеры, HTTP код ответа и много другой полезной информации.
Подробнее http://docs.python-requests.org/en/master/api/#requests.Response

```
import requests

response = requests.get("https://api.privatbank.ua/p24api/pubinfo?exchange&json&coursid=11")

print("response.text:", response.text)
```

Реализация модуля privat_api.py

Модуль - privat_api.py. Его задача:

- получать данные о курсе, который нужно изменить, из базы
- отправить запрос на api Приватбанка, получить нужный курс из ответа
- обновить данные: поле updated и значение курса, полученное по api

Итак, напишем этот модуль, в нем опишем функцию update_xrates(from_currency, to_currency). Параметры from_currency, to_currency - это валюты курса из таблицы xrates, будут использованы для поиска записи, в которой нужно изменить курс. И протестируем его, напишем тест для проверки работы.

Как осуществлять поиск нужного курса?

Приватбанк возвращает ответ в формает JSON.

Это список.
Необходимо пройтись по всем элементам списка и найти такой элемент, у которого атрибут *ccy* равен USD.
И взять у него элемент *sale*!

Это и будет наш курс!

```python
def find_usd_rate(data):
    for e in data:
        if e["ccy"] == "USD":
            return e["sale"]
```


```
[
    {
        "ccy": "USD",
        "base_ccy": "UAH",
        "buy": "26.25000",
        "sale": "26.52520"
    },
    {
        "ccy": "EUR",
        "base_ccy": "UAH",
        "buy": "30.50000",
        "sale": "31.05590"
    },
    {
        "ccy": "RUR",
        "base_ccy": "UAH",
        "buy": "0.40500",
        "sale": "0.42501"
    },
    {
        "ccy": "BTC",
        "base_ccy": "USD",
        "buy": "6953.0833",
        "sale": "7684.9868"
    }
]
```

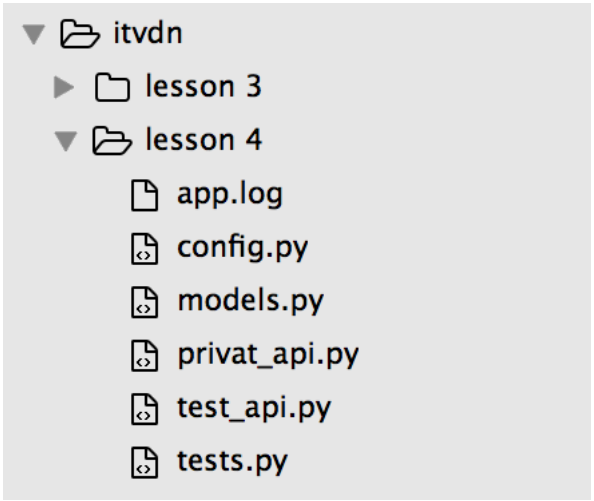
Парсинг

Парсинг — это процесс разбора текстовой информации разнообразного формата, получение необходимых данных и сохранение их в унифицированном формате и дальнейшая обработка.

Парсер — программа или скрипт, осуществляющая парсинг. Под каждый формат - свой парсер. Например, XML парсер, JSON парсер и тп.

Что имеем к настоящему моменту

Полученная структура проекта

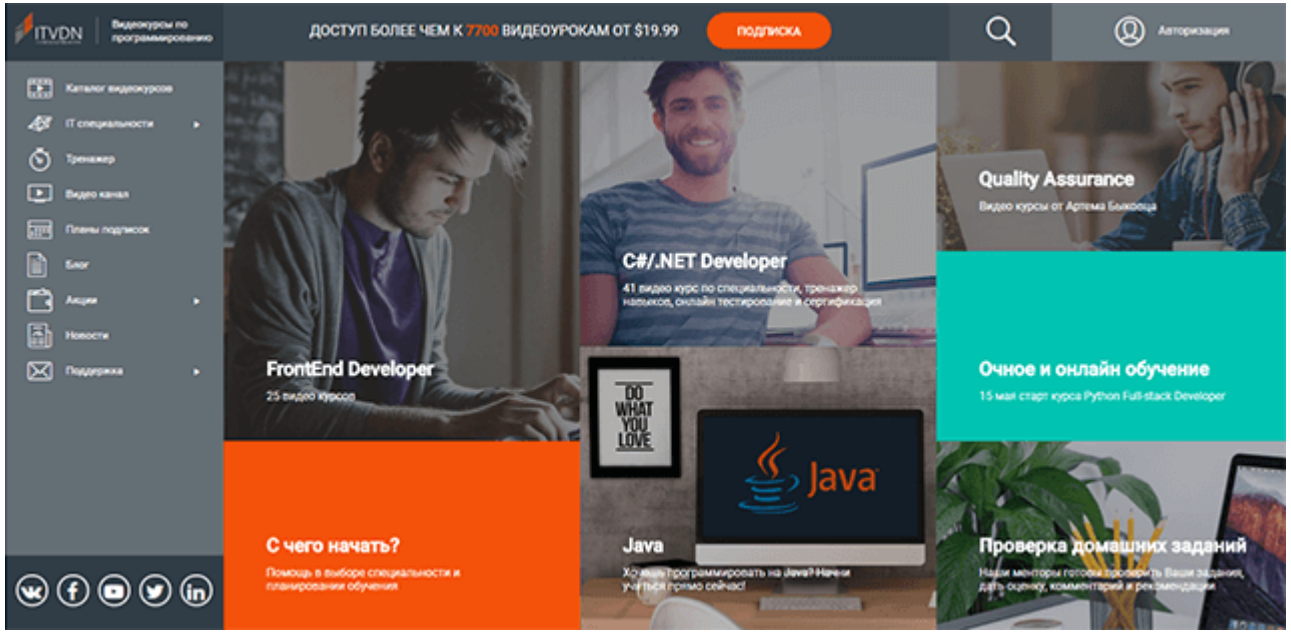


Что дальше?

- Реализация первого api для получения курса USD -> RUB по api ЦБР.
- Реализация XML парсера для получения курса по XML.
- Первый рефакторинг проекта.

Смотрите наши уроки в видео формате

ITVDN.com



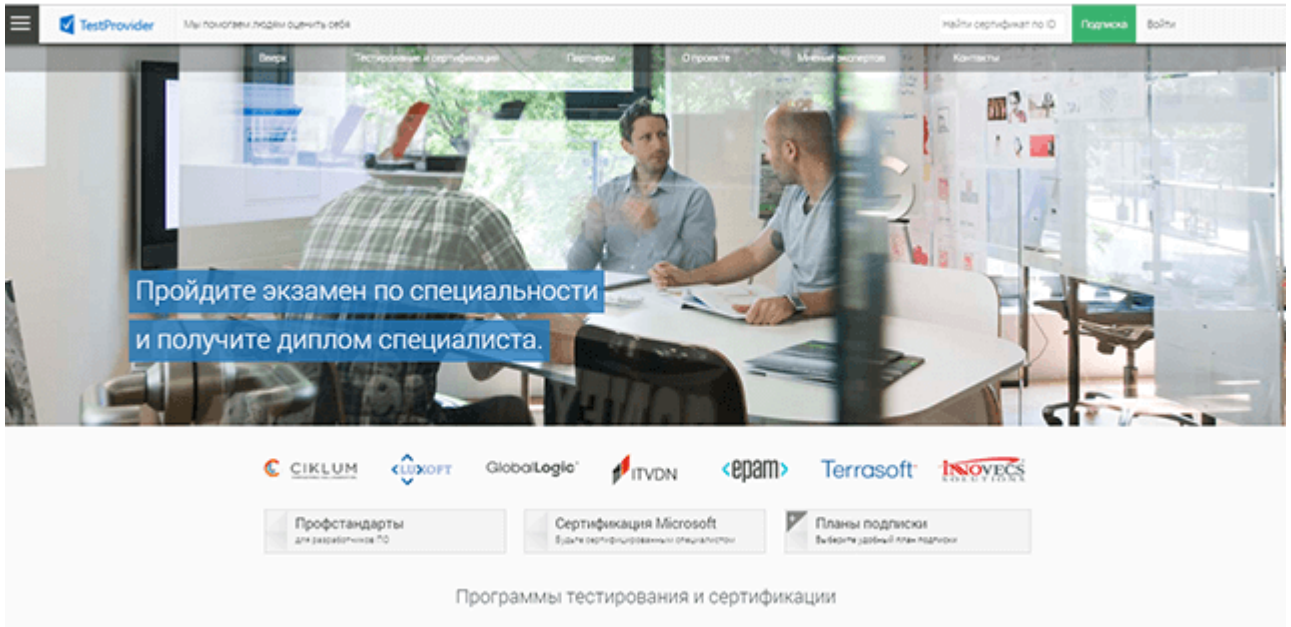
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.

ITVDN
IT VIDEO DEVELOPERS NETWORK

Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

TestProvider

Q&A

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association