



Python Practice

Работа с XML документами. Добавление нового источника валют.

Python Practice

Автор курса



Крементарь Ксения

Ведущий Python разработчик

Системный архитектор

в компании K-Solutions

Python Practice

После урока обязательно



Повторите этот урок в видео формате на
[ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на
[TestProvider.com](http://testprovider.com)

Работа с XML документами. Добавление НОВОГО ИСТОЧНИКА валют.

Python Practice

На этом уроке

1. Познакомимся с принципами работы с XML документами.
2. Рассмотрим библиотеки для работы с XML.
3. Добавим модуль с реализацией API ЦБР и тесты.
4. Произведем рефакторинг проекта.

Python Practice

Описание API ЦБР

Правило: необходимо отправить GET запрос, ответ придет в формате XML.

Точка входа: http://www.cbr.ru/scripts/XML_daily.asp

Python Practice

Что такое XML?

XML (Extensible Markup Language) — расширяемый язык разметки. Предназначен для передачи и хранения информации.

XML документ — это структура данных (в памяти, в файле и т.п.) в формате XML.

Каждый XML документ состоит из элементов, каждый элемент состоит из тегов (открывающийся и закрывающийся) и содержимого. У элементов могут быть атрибуты. Содержимым элемента могут быть другие элементы, которые называются - подэлементы (дочерние элементы). Соответственно, начальный элемент называется родительским.

Корневой элемент — это родительский элемент для всех других элементов.

Python Practice

Примеры XML документов

```
<name>
<first>Kseniia</first>
<last>Krementar</last>
</name>

<person>
<name first="Kseniia" last="Krementar"></name>
<prof>py-dev</prof>
</person>

<person ID="1234567">
  <name>
    <first>Kseniia</first>
    <last>Krementar</last>
  </name>
  <address>
    <city>Kiev</city>
    <district>Obolon</district>
  </address>
</person>
```

```
<employees>
  <person ID="1234567">
    <name>
      <first>Kseniia</first>
      <last>Krementar</last>
    </name>
    <address>
      <city>Kiev</city>
      <district>Obolon</district>
    </address>
  </person>

  <person ID="7654321">
    <name>
      <first>Ivan</first>
      <last>Petrov</last>
    </name>
    <address>
      <city>Kiev</city>
      <district>Pechersk</district>
    </address>
  </person>
</employees>
```


Python Practice

Библиотеки для работы с XML

- стандартная xml, <https://docs.python.org/3.6/library/xml.html>
- простая xmltodict, <https://pypi.org/project/xmltodict/>
- универсальная BeautifulSoup, <https://www.crummy.com/software/BeautifulSoup/>

Python Practice

Встроенная библиотека xml

Позволяет работать с XML документами как с объектами в терминах XML. В результате парсинга с помощью метода `fromstring` получаем объект класса `Element` (`xml.etree.ElementTree.Element`) — корневой элемент XML документа.

У каждого экземпляра класса `xml.etree.ElementTree.Element` есть методы:

- `get(key)` - для получения значения атрибута с именем `key`;
- `find(tag_name)` - для получения первого дочернего элемента с тегом `tag_name`;
- `findall(tag_name)` - для получения списка дочерних элементов с тегом `tag_name`.

Python Practice

Библиотека `xmltodict`

Позволяет работать с разобранным XML документом, как со словарем в Python (dict).

В результате парсинга с помощью функции `xmltodict.parse` получается объект типа `OrderedDict`.

Для получения доступа к дочернему элементу необходимо указать в качестве ключа имя этого искомого элемента.

Для получения доступа к атрибуту элемента необходимо в качестве ключа указать: символ `@` + имя атрибута.

Python Practice

Двигаемся дальше!

Итак, мы готовы приступить к реализации нового модуля, `cbr_api.py`.

- получать данные о курсе, который нужно изменить, из базы.
- отправить запрос на API ЦБР, получить нужный курс из XML ответа.
- обновить данные: поле `updated` и значение курса, полученное по API.

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

Python Practice

Пример ответа по API ЦБР

```
<ValCurs Date="14.07.2018" name="Foreign Currency Market">
  <Valute ID="R01010">
    <NumCode>036</NumCode>
    <CharCode>AUD</CharCode>
    <Nominal>1</Nominal>
    <Name>Австралийский доллар</Name>
    <Value>46,7002</Value>
  </Valute>

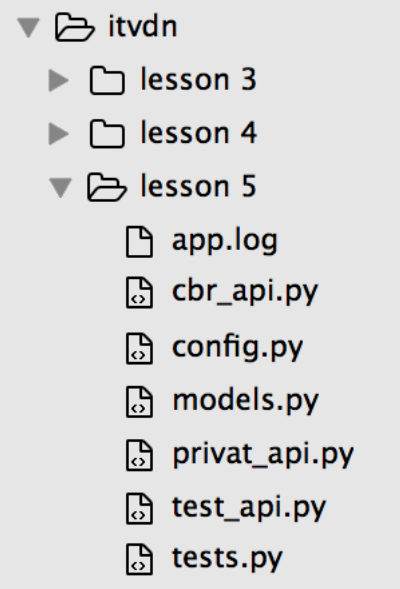
  <Valute ID="R01020A">
    <NumCode>944</NumCode>
    <CharCode>AZN</CharCode>
    <Nominal>1</Nominal>
    <Name>Азербайджанский манат</Name>
    <Value>37,1551</Value>
  </Valute>

  <Valute ID="R01235">
    <NumCode>840</NumCode>
    <CharCode>USD</CharCode>
    <Nominal>1</Nominal>
    <Name>Доллар США</Name>
    <Value>63,2194</Value>
  </Valute>
</ValCurs>
```

Python Practice

Что имеем к настоящему моменту

Полученная структура проекта:



Python Practice

Рефакторинг

Рефакторинг — процесс внесения изменений в структуру написанного кода с целью оптимизации, улучшения читабельности кода.

При реализации нового проекта разработчик должен часто и критично оглядывать свой код, анализировать его на предмет необходимости рефакторинга. После реализации проекта, при дальнейшей его поддержке, также необходимо проводить рефакторинг, но уже гораздо аккуратнее, с тщательным тестированием.

Python Practice

Рефакторинг проекта golden-eye

- Выделение пакета (package) API.
- Добавление классов для каждой реализации API.
- Вынесение общей логики в базовый класс _Api:
 - получение курса из БД.
 - сохранение в БД обновленного.
 - создание логгера.

Python Practice

Структура проекта golden-eye после рефакторинга

```
▼ lesson 5_refactored
  ► __pycache__
  ▼ api
    ► __pycache__
    ├── __init__.py
    ├── cbr_api.py
    ├── privat_api.py
    └── test_api.py
  ├── app.log
  ├── config.py
  ├── models.py
  └── tests.py
```

```
2018-07-21 15:30:58,343 [INFO] - CbrApi:Started update for: 840=>643
2018-07-21 15:30:58,344 [DEBUG] - CbrApi:rate before: XRate(840=>643): 1.0
2018-07-21 15:30:58,570 [DEBUG] - CbrApi:response.encoding: windows-1251
2018-07-21 15:30:58,572 [DEBUG] - CbrApi:response.text: <?xml version="1.0" encoding="utf-8">
2018-07-21 15:30:58,575 [DEBUG] - CbrApi:rate after: XRate(840=>643): 63.4888
2018-07-21 15:30:58,576 [INFO] - CbrApi:Finished update for: 840=>643
2018-07-21 15:30:58,586 [INFO] - TestApi:Started update for: 840=>980
2018-07-21 15:30:58,590 [DEBUG] - TestApi:rate before: XRate(840=>980): 1.0
2018-07-21 15:30:58,592 [DEBUG] - TestApi:rate after: XRate(840=>980): 1.01
2018-07-21 15:30:58,592 [INFO] - TestApi:Finished update for: 840=>980
2018-07-21 15:30:58,605 [INFO] - PrivatApi:Started update for: 840=>980
2018-07-21 15:30:58,606 [DEBUG] - PrivatApi:rate before: XRate(840=>980): 1.0
2018-07-21 15:30:59,088 [DEBUG] - PrivatApi:Privat response: [{'ccy': 'USD', 'base_currency': 'RUB', 'rate': 26.5252}
2018-07-21 15:30:59,090 [DEBUG] - PrivatApi:rate after: XRate(840=>980): 26.5252
2018-07-21 15:30:59,091 [INFO] - PrivatApi:Finished update for: 840=>980
```

Python Practice

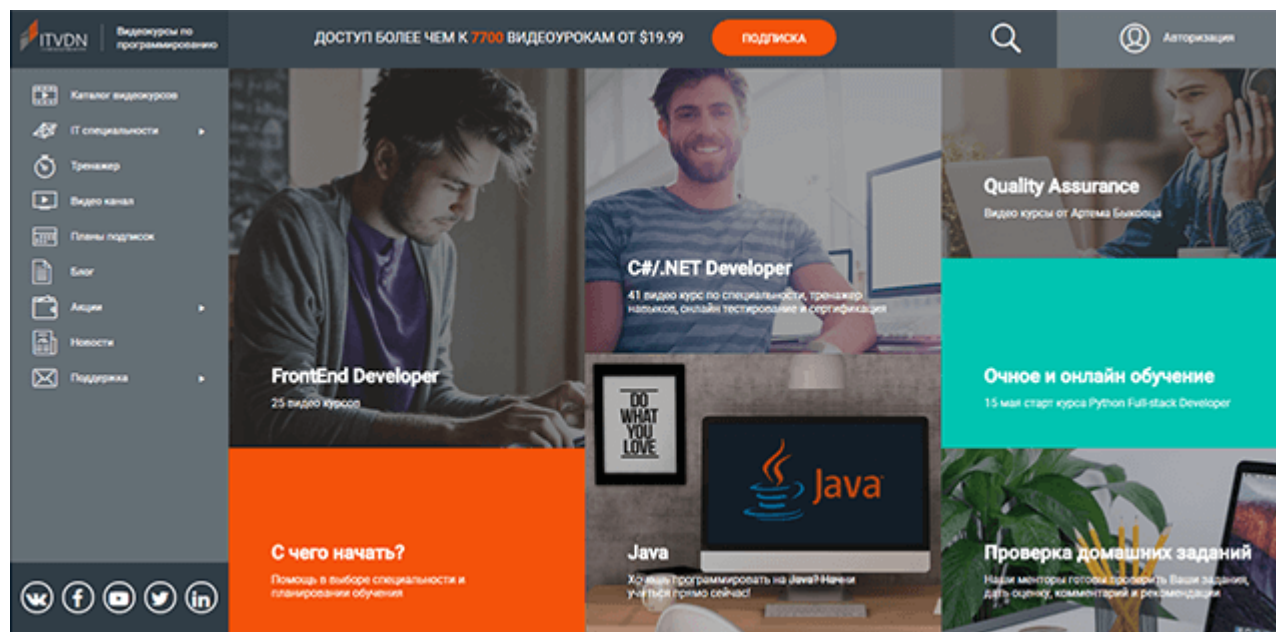
Что дальше?

Усложнение логики проекта:

- Добавление логирования запросов-ответов от API в БД.
- Дальнейший рефакторинг.

Смотрите наши уроки в видео формате

ITVDN.com



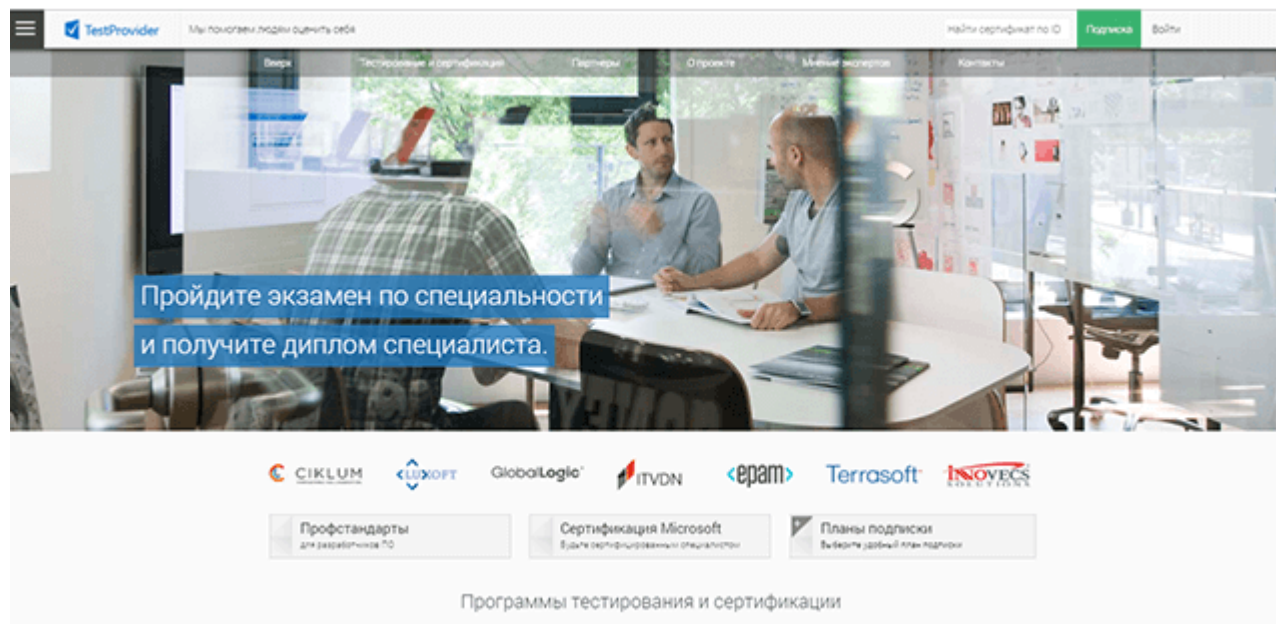
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

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



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](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Python Practice

Q&A

Информационный видеосервис для разработчиков программного обеспечения

