Python Practice

Реализация ядра системы. Тесты.

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 ядра системы. Тесты.

На этом уроке

1. Выберем СУБД для проекта golden-eye, выберем библиотеку python для работы с базой данных
2. Опишем модели
3. Создадим базу и таблицы в ней. Поиграемся с интерфейсом peewee для выборки, создания записей, редактирования
4. Напишем модуль тестового api, который будет менять курс в базе данных
5. Напишем тест для запуска и проверки работы модуля тестового api
6. Коснемся вопроса тестирования и особенностей тестирования в интерпретируемых языках

Выбор базы данных

Распространенные СУБД:

подходит для простейших проектов или прототипов. Не требует сервера базы данных. Не рекомендуется к использованию на продакшн среде.

простая в установке и администрировании СУБД, доступна на всех популярных операционных системах.

мощная СУБД, доступна на всех популярных операционных системах.

Oracle и Microsoft SQL Server - мощные СУБД, распространяются по коммерческой лицензии.

Sqlite

Sqlite база данных - это простая база данных, написанная на C. Она не требует сервера БД, данные могут сохраняться в файле на диске или в памяти системы. Работа с ней доступа с версии Python 2.5 - не требует никакой дополнительной установки для начала работы. Встроенный модуль sqlite3 позволяет быстро начать работу с базой.

Библиотеки ORM

ORM(Object-relational mapper) — объектно-реляционное отображение, или преобразование.

ORM библиотеки — специальные библиотеки, автоматизирующие работу с данными, хранимыми в БД.

- Избавляют разработчика от необходимости писать SQL код для взаимодействия с БД - позволяют использовать только код языка программирования.
- Обеспечивают интерфейс для CRUD операций над данными.
- Автоматизируют перевод данных из реляционных БД в объекты, которые могут быть использованы в коде приложения.

Python ORM библиотеки

Django ORM - используется в Django фреймворке. Хорошо работает с простыми запросами, средней сложности, но при сложных выборках генеруются очень сложные SQL запросы. Используется в Django. Поддерживаемые СУБД: PostgreSQL, MySQL и SQLite.

SQLAlchemy - мощная библиотека, позволяет разработчику использовать все возможности SQL с помощью языка Python. Поддерживаемые СУБД: SQLite, Postgresql, MySQL, Oracle, MS-SQL.

Peewee - позиционируется как более простая в написании и использовании, нежели SQLAlchemy. Более легкая, более быстрая. Может быть использована в большинстве фреймворков, а также без него. Поддерживаемые СУБД: PostgreSQL, MySQL и SQLite.

PonyORM - легкая, простая ORM библиотека с приятным pythonic синтаксисом. Поддерживаемые СУБД: PostgreSQL, MySQL и SQLite.

Peewee ORM

Открытая Python библиотека для связывания данных, хранящихся в таблицах реляционных баз данных, с объектами Python.

http://docs.peewee-orm.com/en/latest/

Peewee ORM

```
from peewee import (SqliteDatabase, Model, IntegerField, DoubleField,
                    DateTimeField, datetime as peewee_datetime)

db = SqliteDatabase("golden-eye.db")


class XRate(Model):
    class Meta:
        database = db
        db_table = "xrates"
        indexes = (
            (("from_currency", "to_currency"), True),
        )

    from_currency = IntegerField()
    to_currency = IntegerField()
    rate = DoubleField()
    updated = DateTimeField(default=peewee_datetime.datetime.now)
```

xrates	
from	int, not null
to	int, not null
rate	double, not null
updated	datetime, not null

Реализация модуля test_api.py

Первый модуль - test_api.py. Его задача:

- получать данные о курсе из базы
- обновить данные: поле updated и значение курса, увеличенное на 0.01.

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.

Тестирование

Встроенная библиотека unittest - фреймворк для unit тестирования. Unit тестирование - это метод тестирования, который позволяет в автоматическом режиме проверить поведение модулей системы в определенных условиях. Каждый юнит тест проверяет определенное ожидаемое поведение системы в тех или иных условиях, логику работы отдельных функций, классов, модулей.

Для создания тестов, необходимо создать класс, унаследованный от класса unittest.TestCase. А у этого класса определить методы, название которых начинается с test_. Эти методы и будут являться тест-кейсами.

Запуск тестов из консоли

```
python -m unittest test_module

python -m unittest test_module.TestClass

python -m unittest test_module.TestClass.test_method
```

Особенности тестирования в Python

Так как python - интерпретируемый язык, то в отличие от компилируемых языков некторые строки кода могут быть и не запущены никогда (например, редкое условие if или функция, которая описана, но не вызывается) или запущены в очень редких случаях. А в них может скрываться ошибка. Например, Exception() вместо raise Exception() - это бывает чаще, чем вы думаете - и в результате ошибка появляется совсем не там и не так, как ты ожидаешь.

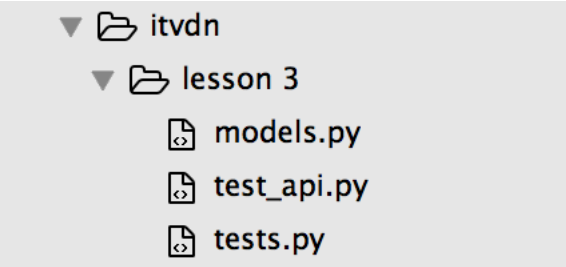
Поэтому в python и в других интерпретируемых языках тесты используются еще и для того, чтоб покрыть исходный код, выполнить все if и все функции и методы и тп. Это позволяет разработчику глубже проникнуть в бизнес-логику и понять ее, что тоже очень важно!

Тестирование. Первый тест.

```python

import unittest

import test_api
import models


class Test(unittest.TestCase):
    def setUp(self):
        models.init_db()

    def test_main(self):
        xrate = models.XRate.get(id=1)
        self.assertEqual(xrate.rate, 1.0)
        test_api.update_xrates(840, 980)
        xrate = models.XRate.get(id=1)

        self.assertEqual(xrate.rate, 1.01)

```

Что имеем к настоящему моменту

Полученная структура проекта



Пример запуска теста из консоли

```
krementar@192:~/projects/itvdn/lesson 3$ python -m unittest tests
```

Что дальше?

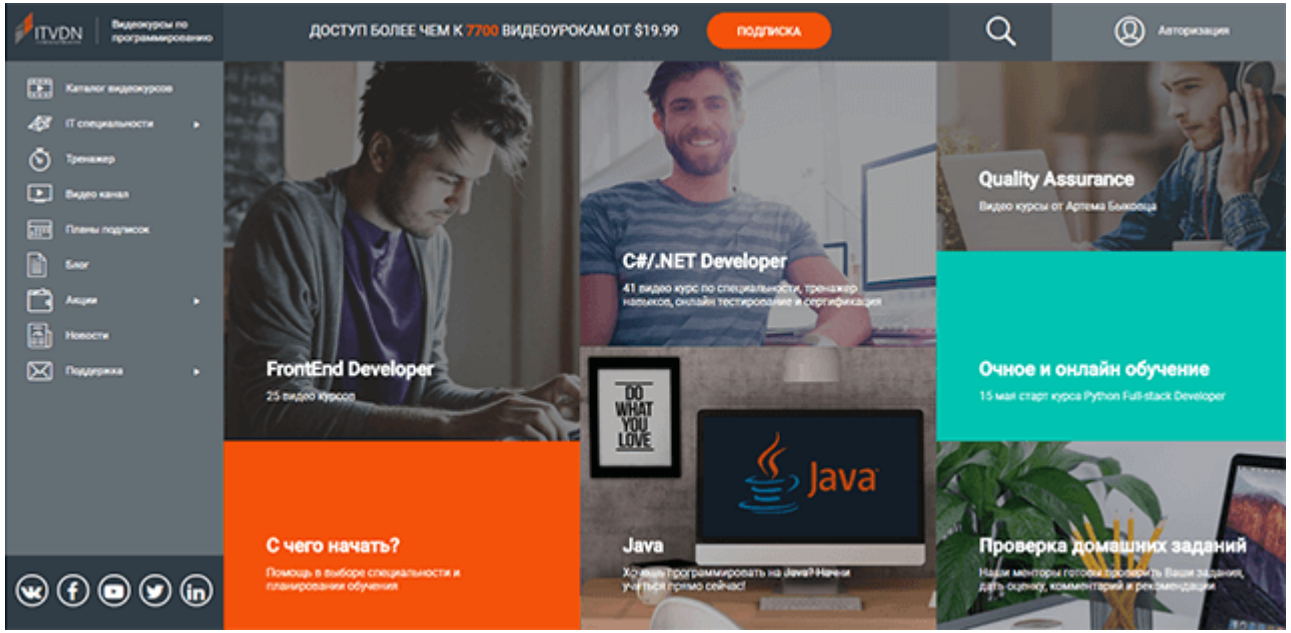
Реализация первого api для получения реального курса валют - api Приватбанка!

Усложенение структуры проекта, добавление новых тестов.

Добавление логирования в проект.

Смотрите наши уроки в видео формате

ITVDN.com



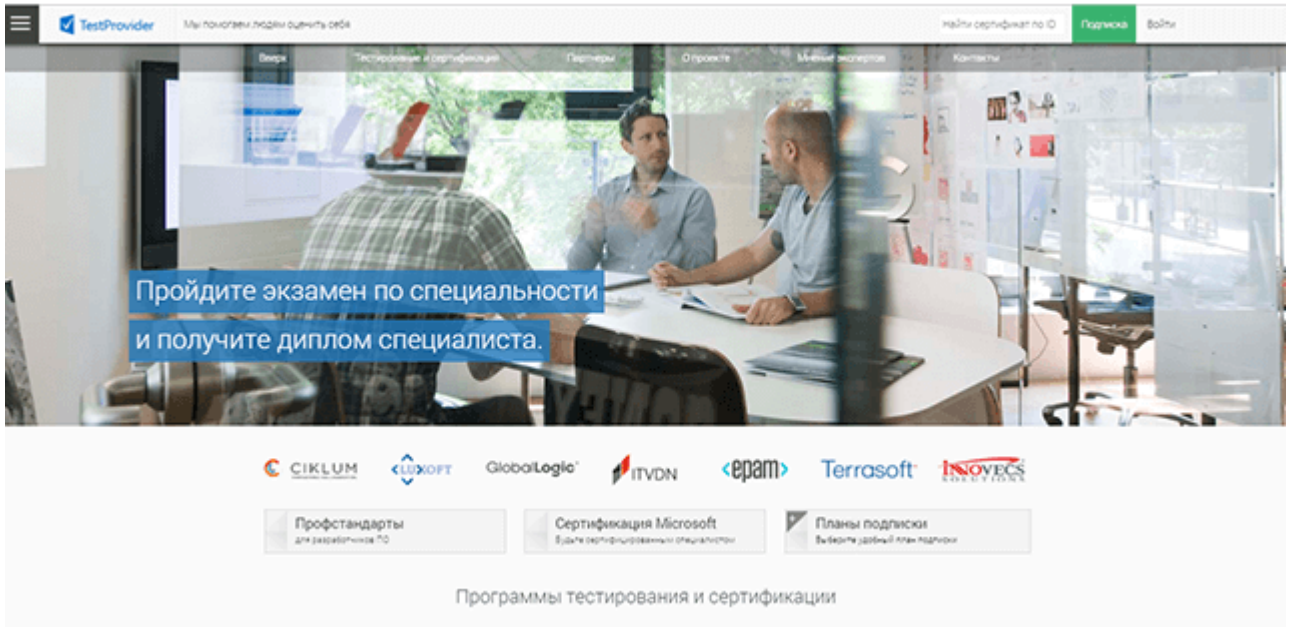
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

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