Добавление отображения курсов валют.

№ урока:	9	**Курс:**	Python Practice

Средства обучения:	Интерпретатор Python, virtualenv, текстовый редактор

Обзор, цель и назначение урока

Цель урока: расширить функционал созданного web приложения, добавить отображение курсов валют на странице сайта. Спроектировать структуру web проекта.

Изучив материал данного занятия, учащийся:

- Научится проектировать структуру web сервиса на основе Flask фреймворка, группировать функционал в модули
- Научится использовать jinja2 шаблонизатор для отображения более сложных html конструкций, а также использовать jinja2 конструкции for, if и тп.
- Научится обрабатывать ошибки, возникающие при работе приложения и возвращать неуспешный ответ

Содержание урока

1. Краткое резюме, что мы уже имеем к данному момент и обзор дальнейших действий.
2. Проектирование web сервиса, разделение кода на модули, обзор этих модулей.
3. Добавление view-функции для отображения курсов валют и реализация контроллера с логикой.
4. Добавление обработки ошибки при работе контроллера, возврат "ошибочного" ответа, HTTP коды ошибок.
5. Разработка шаблона для отображения курсов валют в виде html таблицы, использование конструкции for, if.

Резюме

- Итак, мы уже умеем создавать web приложения на основе Flask фреймворка, осталось добавить нужные страницы нашего сайта и нужные методы api.
- Несмотря на то, что web приложение на Flask можно поместить в один модуль, и этого будет достаточно для того, чтоб оно работало, более-менее большие проекты, конечно же, содержат более одного модуля и код сгруппирован в модули и пакеты.
- Типичная структура Flask приложения содержит несколько модулей. Названия модулей, конечно же, могут варьироваться, на усмотрение разработчика, но суть такова:
  - app.py - модуль, в котором осуществляется создание Flask приложения, создание экземпляра класса Flask.
  - модуль runserver.py - модуль, в котором происходит запуск приложения на тестовом локальном сервере.
  - views.py - модуль, в котором описаны функции и соответствующие им url'ы приложения.
  - controllers.py - модуль или пакет с функциями или классами, в которых описана основная логика работы.
  - models.py - модуль или пакет для доступа к базе данных.
  - config.py - модуль с настройками приложения.
  - utils.py - модуль с утилитными функциями, которые используются из различных модулей.
  - tests.py - модуль или пакет с тестами.
  - Папка templates для хранения шаблонов приложения.

CyberBionic systematics
CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

Рассмотрим эти модули подробнее на примере приложения Golden eye.

- Сегодня нам нужно будет добавить страницу с отображением всех курсов валют. Для этого мы опишем view функцию view_rates, результатом которой будет html шаблон. Логика получения данных о курсах, рендеринг шаблона и вся остальная логика, которая необходима для обработки этого запроса, описана в controllers.py, в соответствующем контроллере, рассмотрим его подробнее.
- При рендеринге шаблона view_rates.html курсы валют, полученные из базы, удобно передавать списком, а для динамического "встраивания" информации о курсах воспользоваться конструкцией for. Синтаксисе jinja для конструкции for имеет вид: {% for item in iterable %} html конструкции или выражения {{}} {% endfor %}. Обращаю внимание, что {% endfor %} элемент обязателен, означает конце блока for.
- Синтаксис if конструкции похожий, есть опциональные блоки {% elif %} и {% else %}, но блок {% endif %} обязателен, как и {% endfor %}. Рассмотрим использование конструкций на примере шаблона view_rates.html, где отображение курсов будет в html таблице, строки которой будут генерироваться в цикле {% for %}.
- По умолчанию, если обработка запроса во Flask приложении завершилась без ошибок, клиенту возвращается HTTP статус 200, который свидетельствует о корректной обработке запроса. В случае, если что-то пошло не так, может возвращаться иной статус. Flask позволяет явно указать HTTP код обработки запроса, метод make_response принимает в качестве аргументов (body, status, headers), где body — тело ответа, строка, status — HTTP статус ответа и headers — словарь заголовков ответа. С помощью функции make_response можно реализовать обработку ошибки, возникшей при обработке запроса.
- В результате внесенных изменений, имеем Flask проект с описанной выше структурой, с одним url для отображения курсов валют в виде html таблицы. В случае ошибки, вернется текст ошибки и HTTP код 500.

Закрепление материала

- Зачем разбивать код проекта на модули?
- Какие модули можно выделить в типичном Flask проекте?
- Почему блоки endfor и endif являются обязательными в синтаксисе jinja?
- Какой HTTP код обозначает корректную обработку запроса? Как вернуть корректный HTTP код во Flask приложении (два варианта, минимум)?

Дополнительное задание

Задание

Добавить проверку списка курсов валют в шаблоне, а если список курсов пустой, отображать строку - Курсы не найдены в базе данных, вместо пустой таблицы.

Самостоятельная деятельность учащегося

Задание 1

Добавить отображение валют не кодами, а алиасами, на странице отображения курсов.

Задание 2

Написать тест для вызова url с отображением курсов валют, проверить, что возвращается корректный HTTP код. Написать тест для генерации ошибки и проверить, что возвращается HTTP код 500 в ответе.

Рекомендуемые ресурсы

http://flask.pocoo.org/docs/1.0/

CyberBionic systematics

CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

http://jinja.pocoo.org/docs/2.10/
http://jinja.pocoo.org/docs/2.10/templates/#list-of-control-structures
http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make_response

CyberBionic systematics
CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com