

# Добавление возможности обновления курсов с сайта и отображения логов из БД.

**№ урока:** 11 **Курс:** Python Practice

**Средства обучения:** Интерпретатор Python, virtualenv, текстовый редактор

## Обзор, цель и назначение урока

Цель урока: расширить функционал созданного web приложения, добавить возможность обновления курсов с сайта, а также отображение логов общения с удаленными api.

## Изучив материал данного занятия, учащийся:

- Закрепит навыки по построению роутинга url во Flask приложениях.
- Закрепит навыки по работе с GET запросами Flask.
- Закрепит навыки работы с ORM библиотекой peewee.
- Узнает, что такое пагинация и варианты ее реализации.

## Содержание урока

1. Краткое резюме, что мы уже имеем к данному момент и обзор дальнейших действий
2. Добавление view-функции для обработки GET запроса на обновление конкретного курса валют или всех курсов.
3. Реализация контроллера с логикой обновления курсов валют с помощью пакета api.
4. Добавление на страницу отображения курсов валют ссылки для обновления курса. Добавление ссылки для обновления всех курсов. Динамическое построение url с помощью функции url\_for.
5. Добавление функционала по отображению логов — view-функция с динамическим url для передачи типа логов, реализация контроллера и добавление нового шаблона с отображением логов.
6. Добавление пагинации при отображении логов.
7. Проверка нового функционала "вручную".
8. Написание и запуск тестов для проверки нового функционала с использованием библиотеки requests.

## Резюме

- Для обновления курсов со страницы отображения курсов валют будем использовать пакет api, реализованный ранее. Необходимо только "связать" нажатие кнопки на сайте с отправкой запроса на нужное api.
- Прежде всего нужно будет добавить новую view-функцию и новый url, при вызове которого должно будет осуществляться обновление курсов. При этом, параметры курса - from\_currency, to\_currency могут быть установлены либо в самом url, либо переданы в качестве GET параметров запроса. Можно реализовать с передачей параметров в url, динамическими параметрами.
- Во Flask есть возможность добавлять несколько url для одной и той же view функции — просто указав один декоратор над другим. Воспользуемся такой возможностью для добавления двух вариантов обновления курсов — либо конкретный курс при вызове url вида /update/<from\_currency>/to/<to\_currency>, либо все курсы при вызове url вида /update/all. При этом обработка указанных url будет осуществляться одной и той же view-функцией.

- Само тело новой view-функции представляет собой вызов нового контроллера с передачей параметров из url и дальнейший редирект на страницу отображения курсов.
- При реализации нового контроллера будет осуществляться вызов функции update\_rate из пакета api с параметрами конкретного курса или получение списка курсов из БД и последовательный вызов с каждым курсом.
- Осталось последнее — добавление ссылки на обновление каждого курса валют в html таблице на странице отображения курсов. Для каждого курса url уникален, формируется в зависимости от from\_currency и to\_currency. Для динамического формирования url удобно использовать функцию url\_for.
- Следующий этап — добавление отображение логов из БД. Как и остальной новый функционал, мы начинаем с создания новой view-функции и привязки url к ней. Затем нужно будет добавить новый шаблон для рендеринга страницы логов.
- Так как при постоянной работе приложения количество логов в базе данных будет расти, удобнее будет отображать не все содержимое таблиц с логами, а постранично, то есть определенными партиями, например, по 10 штук за один раз. Такое отображение называется пагинацией. При этом в GET аргументах запроса будет использоваться параметр page, отвечающий за номер требуемой страницы.
- Для реализации пагинации в библиотеке peewee есть метод paginate. При вызове его у какого-либо объекта select запроса необходимо указать первым параметром номер страницы, вторым — количество элементов в выборке. Добавим использование этого метода в контроллере по работе с логами.
- В результате внесенных изменений и добавления нового функционала можно будет обновлять курсы со страницы отображения курсов валют, а также просматривать логи общения с удаленными api, с пагинацией.

### Закрепление материала

- Как привязать несколько url к одной и той же view-функции во Flask приложении?
- Что такое пагинация? Зачем она используется в web приложениях?
- Каким образом можно получить данные для пагинации при использовании библиотеки peewee?

### Дополнительное задание

#### Задание

Добавить пагинатор на страницу отображения логов. Добавить переход на следующую страницу (Next) и предыдущую (Prev), если текущая страница больше 1.

### Самостоятельная деятельность учащегося

#### Задание 1

Добавить фильтрацию логов по id, времени создания, использование GET аргументов в url.

#### Задание 2

Добавить параллельную отправку запроса на удаленные api при обновлении всех курсов валют. Использовать стандартную библиотеку threading.

### Рекомендуемые ресурсы

[http://flask.pocoo.org/docs/1.0/api/#flask.url\\_for](http://flask.pocoo.org/docs/1.0/api/#flask.url_for)  
<http://flask.pocoo.org/docs/1.0/api/#flask.Request.args>  
<http://docs.peewee-orm.com/en/latest/peewee/queries.html#paginating-records>