

WEB-ПРИЛОЖЕНИЕ, ОБЪЕДИНЯЮЩЕЕ НОВОСТНЫЕ ЛЕНТЫ СОЦИАЛЬНЫХ СЕТЕЙ

Горюнова М.А., Комаров И.Н.

Лепустин А.В.

Томский политехнический университет

mag8@tpu.ru

Введение

Пользователи социальных сетей получают ежедневно большое количество новостей от страниц и сообществ, на которые они подписаны. Для пользователя гораздо удобнее читать новостные ленты со всех источников и аккаунтов в одном месте, поэтому целью работы была реализация web-приложения, объединяющего все новостные ленты и отображающего на одном ресурсе.

Описание приложения

Данное web-приложение предоставляет пользователю возможность добавить несколько учетных записей для социальной сети «ВКонтакте» и видеохостинга «YouTube».

Приложение формирует новостную ленту из всех подписок пользователя в добавленных аккаунтах, упорядочивает ее по времени добавления и отображает. В новостной ленте отображается текст, изображения, аудио и видео файлы, с возможностью просмотра видео и прослушивания аудио на страницах web-приложения, без перехода на внешние ресурсы.

Пользователь имеет возможность перейти из web-приложения к оригиналу просматриваемой новости.

Реализация серверной части приложения

Серверная часть приложения реализована на высокоуровневом языке программирования общего назначения – Python. Основные архитектурные черты языка – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных.

Веб приложение реализовано с помощью фреймворка Django. Django – это фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MTV [1].

MTV (Model-Template-View) – шаблон проектирования во многом схожий с шаблоном MVC

- Model – слой доступа к данным.
- Template – слой представления данных. Этот слой принимает решения относительно представления данных.

View – слой бизнес-логики. Этот слой содержит информацию о том, как получить доступ к моделям и применять соответствующий шаблон.

Этот слой можно рассматривать, как мост между моделями и шаблонами.

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Один из основных принципов фреймворка – DRY (англ. *Don't repeat yourself*)[2].

В приложении реализован следующий функционал:

- регистрация;
- авторизация;
- личный кабинет;
- добавление новых учетных записей;
- удаление учетных записей;
- отключение учетных записей;
- пользовательские настройки;
- формирование новостной ленты.

Связь с сервисами производится с помощью API. API (интерфейс программирования приложений, интерфейс прикладного программирования) – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах [3].

Связь с API производится по HTTP протоколу, с помощью POST и GET запросов. Для доступа к API соответствующих сервисов используются токены доступа, то есть в базе данных приложения не хранятся данные учетных записей социальных сетей (Логин/Пароль), что обеспечивает безопасность данных пользователя. В приложении используются offline-токены, они отличаются тем, что не требуют обновления токена через отрезок времени. Их необходимо заменять только в случае смены пароля.

После авторизации пользователю становится доступен личный кабинет, где он может изменить указанные при регистрации данные. Также, личный кабинет пользователя предоставляет ему возможность управления учетными записями своих социальных сетей. Пользователь может добавить более одной учетной записи для одной социальной сети. Также пользователь может удалить учетную запись социальной сети или отключить её (новостная лента этой учетной записи отображаться не будет). Из новостей, получаемых в режиме реального времени по запросу пользователя, формируется одна новостная лента. Дублирующие друг друга записи объединяются в одну.

Реализация клиентской части приложения.

Клиентская часть приложения реализована с использованием языка разметки HTML, метаязыка SASS, формального языка описания внешнего вида документа CSS и библиотеки jQuery.

При реализации клиентской части используется интерфейсный фреймворк Bootstrap, включающий в себя HTML и CSS шаблоны оформления для типографики, форм, кнопок, блоков и прочих интерфейсных компонентов. К достоинствам данного фреймворка можно отнести:

- простоту использования – для использования фреймворка достаточно базовых знаний HTML и CSS;
- возможность редактирования стандартных CSS-стилей;
- наличие большого количества документации (в том числе русскоязычной);
- совместимость с браузерами;
- открытость – бесплатная загрузка фреймворка и открытый исходный код [4].

Данный фреймворк позволяет поддерживать все принципы адаптивной верстки – легко масштабировать проект с одной базой кода от телефонов и планшетов до настольных компьютеров. На рисунке 1 показан внешний вид страницы регистрации на настольном компьютере, на рисунке 2 показана эта же страница при просмотре с мобильного телефона.



Рис. 1 – Страница регистрации на настольном компьютере

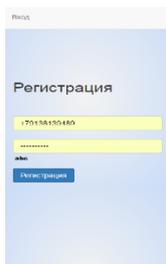


Рис. 2 – страница регистрации на мобильном телефоне

Стили приложения описаны с использованием метаязыка SASS. Для компиляции стилей, написанных на SASS, в понятный браузер CSS использовался сборщик проектов GULP. Данный сборщик позволяет «отслеживать» изменения в указанных SASS файлах и компилировать их в CSS при наличии изменений.

Библиотека jQuery, используемая при разработке, сфокусирована на взаимодействии JavaScript и HTML. Она позволяет получить доступ к DOM элементам страницы, получать атрибуты и менять их значения [5].

Подобно тому, как CSS служит для отделения визуализации от структуры HTML, jQuery отделяет поведение.

В данном приложении библиотека jQuery использовалась для реализации модальных окон, сворачивающихся/разворачивающихся блоков и эффектов анимации элементов.

С помощью библиотеки реализована возможность свернуть новость (вместо шапки, включающей источник новости и текст, отображается только шапка). Для удобной навигации по новостям реализована возможность перемещения к первой записи с любой позиции при просмотре новостей.

На клиентской части реализованы следующие страницы:

- страница регистрации нового пользователя;
- страница авторизации;
- личный кабинет пользователя (включает в себя настройки профиля, позволяет добавлять учетные записи и управлять добавленными учетными записями);
- страница просмотра новостной ленты.

Заключение

Web-приложение, реализованное с использованием фреймворка Django, позволяет пользователю просматривать новостные ленты с различных социальных сетей в одном месте. Для реализации клиентской части был выбран фреймворк Bootstrap, как один из самых распространённых фреймворков с большим количеством русскоязычной документации.

В перспективе планируется реализовать возможность добавления аккаунтов таких ресурсов как «Instagram», «Facebook», «Habrhabr» и усовершенствовать клиентскую часть приложения.

Список использованных источников

1. Фреймворк Django. [Электронный ресурс]. – Режим доступа: <https://www.djangoproject.com/>, свободный (дата обращения: 11.04.2016).
2. Документация Django. [Электронный ресурс]. – Режим доступа: <http://djbook.ru/rel1.9/>, свободный (дата обращения: 01.03.2016).
3. The Django Book. [Электронный ресурс]. – Режим доступа: <http://www.djangobook.com/en/2.0/index.html>, свободный (дата обращения: 8.04.2016).
4. Документация Bootstrap. [Электронный ресурс]. – Режим доступа: <http://bootstrap-3.ru/index.php>, свободный (дата обращения: 10.03.2016).
5. Учебник jQuery, HTML, CSS. [Электронный ресурс]. – Режим доступа: <http://www.wisdomweb.ru/>, свободный (дата обращения 02.03.2016).