УДК 004.4

СОЗДАНИЕ НОВОСТНОГО САЙТА-ПАРСЕРА СО СВОИМ API И ЕГО ИСПОЛЬЗОВАНИЕ ЧЕРЕЗ МОБИЛЬНОЕ ПРИЛОЖЕНИЕ

Кивишев К.А., студент гр. ПИб-192, III курс
Черкасова М.О., студент гр. ПИб-192, III курс
Научный руководитель: Тайлакова А.А., старший преподаватель
Кузбасский государственный технический университет
имени Т.Ф. Горбачева
г. Кемерово

Существует множество сайтов, которые решают определенные задачи. Одни существуют для продажи различной одежды, другие предоставляют пользователю информацию о расписании автобусов или авиабилетов и т.д. Как правило, на подобных ресурсах информация является уникальной и не берётся из каких-либо других источников в интернете. В тоже самое время, существуют сайты-парсеры. Суть парсеров в том, чтобы брать из различных источников информацию и каким-либо образом использовать её. Такие сайты в основном берут информацию и используют её для анализа, к примеру, прогноз изменения валют.

В данной статье, описана разработка сайта для сбора информации с новостных ресурсов и дальнейшего изучение интересов пользователей (какого типа статьи нравится читать пользователям – политические, экономические, всемирные и т.д.).

Сайт был разработан с помощью фреймворка Django. Архитектура данного фреймворка состоит из представлений, контроллера и моделей. Представления – это страницы сайта и в основном хранят в себе дизайн и функцию вызова определенной процедуры при нажатии на кнопку. Контроллер отвечает за логику сайта, он распределяет что должно будет происходить на самом сайте: будет выполняться поиск по наименованию статьи, регистрация или авторизация, переход на страницу статьи и др. Модели являются сущностями для базы данных и могут хранить в себе некоторую логику.

База данных сайта состоит из 3 сущностей – Пользователь, Профиль и Новость. Пользователь – это встроенный пользователь Django, содержит несколько полей, в т. ч. id (первичный ключ пользователя), login (логин пользователя), password (пароль). Профиль – это сущность, которая дополняет информацию о пользователе, имеет связь один к одному и хранит в себе следующие данные – id_User (внешний ключ пользователя), isCreator (логическое значение, обозначающее является ли пользователь редактором или нет). Новость – это сущность, которая содержит следующие поля – id (первичный ключ), id_User (внешний ключ пользователя), title (заголовок), image (изображение), text (текст), dateCreate (дата создания), isNews (логическое значение: является ли запись статьей или новостью).

Авторизация на сайте происходит с помощью POST запроса, который отправляет на сервер значения логина и пароля (рис. 1). Далее с помощью встроенного метода Django authenticate мы отправляем введенные логин и пароль, и при подтверждении прав пользователя, метод вернет объект Пользователь, в ином случае ничего не вернет.

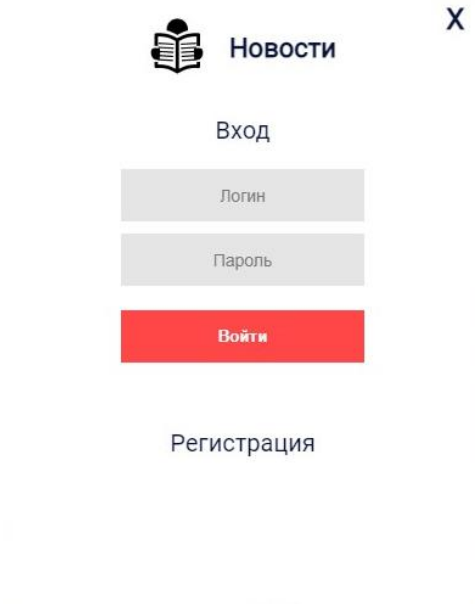

Рисунок 1 – Структура поля авторизации

Регистрация происходит практически аналогичным образом. Пользователь вводит логин, пароль и его подтверждение (рис. 2). После нажатия на кнопку, введенные данные отправляются на сервер и создается запись пользователя Django и профиля. Для всех пользователей, которые регистрируются через сайт, изначально присваивается ложное значение isCreator, которое означает, что пользователь является обычным читателем и не имеет права писать статьи.

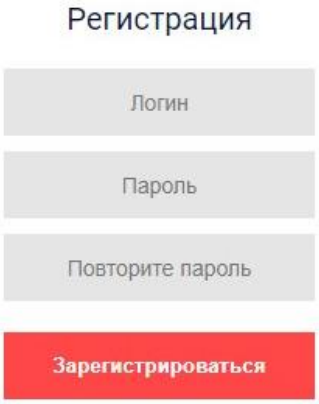

Рисунок 2 – Структура страницы регистрации

Так же на сайте присутствует функция поиска, которая принимает на вход строковое значение, и после на его основе выполняет поиск статей,

в заголовке которых встречается введенная фраза без учета регистра. После того как программа находит все записи, она переводит пользователя на другую страницу поиска и выводит все найденные ранее записи (рис. 3).

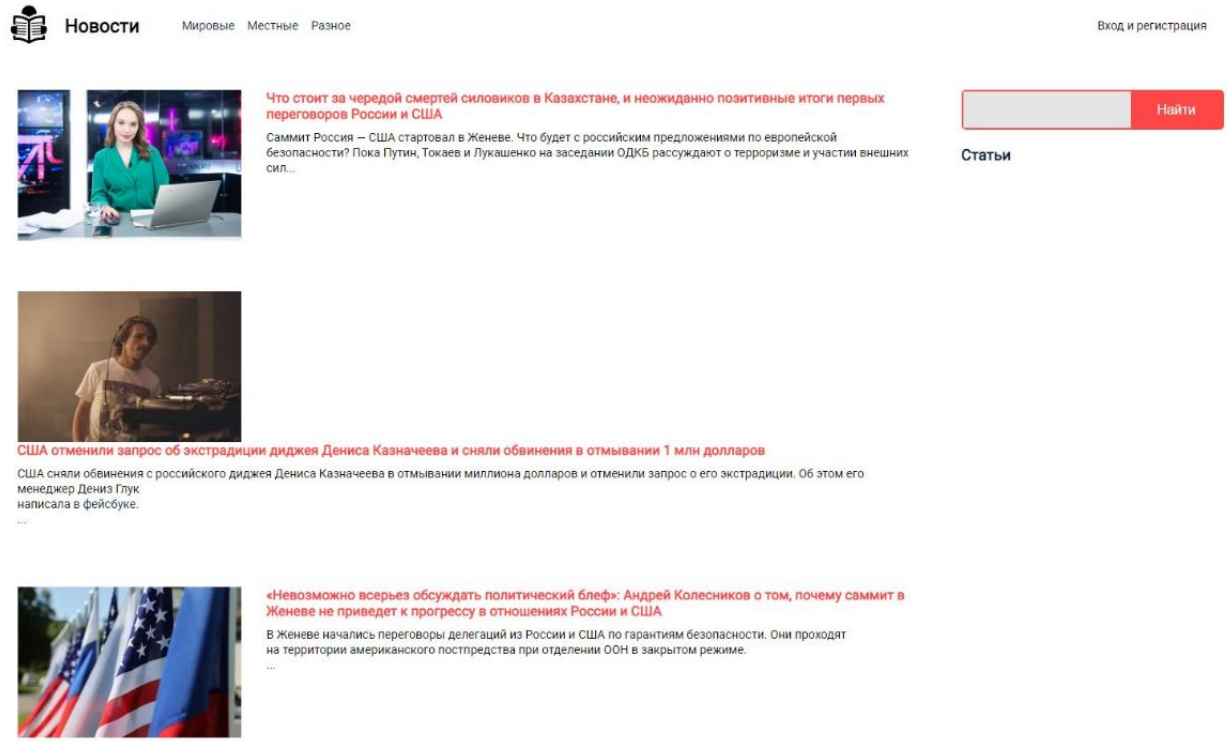

Рисунок 3 – Страница найденных записей

Было разработано мобильное приложение на языке Java для использования API созданного сайта, для дальнейшего управления. Для этого на сайте была реализована архитектура REST API, которая позволяет передавать информацию о введенном ID пользователя (или пользователей).

Архитектура REST API позволяет реализовывать 4 вида запросов, GET, POST, DELETE и PUT. GET – это метод HTTP-запроса, который используется только для получения информации. POST – это метод, который используется для ввода информации на сайт. DELETE – метод удаления информации. PUT – метод редактирования информации.

Для мобильного приложения использовалась библиотека retrofit. Retrofit – это REST клиент для Java и Android. Он позволяет легко получить и загрузить JSON (или другие структурированные данные) через веб-сервис на основе REST. В Retrofit вы настраиваете, какой конвертер используется для сериализации данных. Обычно для JSON используется GSon, но вы можете добавлять собственные конвертеры для обработки XML или других протоколов. В Retrofit используется библиотека OkHttp для HTTP-запросов[1].

В мобильном приложении был создан класс User, который имеет аргументы – id, username, date_joined. С помощью GET запроса мы обращаемся к сайту по ссылке «…/users/» и принимаем выходные данные – id, username и date_joined. Таким образом приложение может получить более подробную информацию о конкретном пользователе (рис. 4, рис. 5).

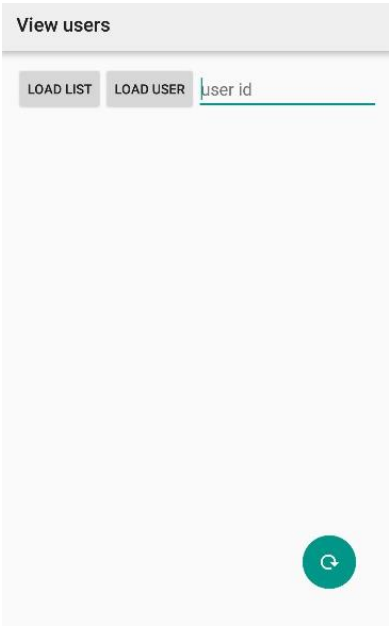

Рисунок 4 – Дизайн мобильного приложения

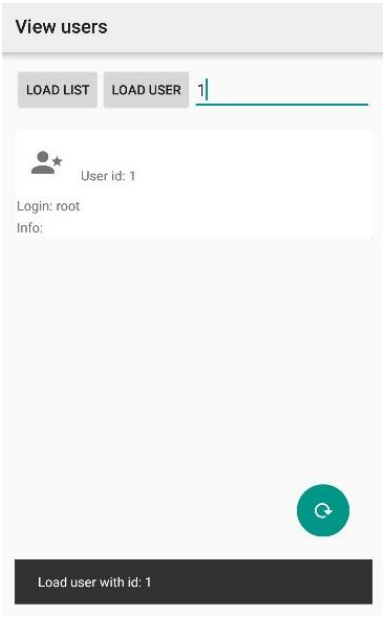

Рисунок 5 – Вывод подробной информации о конкретном пользователе

В результате выполнения работы был реализован сайт-парсер для анализа предпочтений пользователей новостных ресурсов. Реализовано мобильное приложение для доступа к функциям сайта.

Список литературы:

1. Использование Retrofit 2.x в качестве REST клиента — Tutorial [Электронный ресурс] – Режим доступа: https://habr.com/ru/post/428736/ свободный (дата обращения: 11.01.2022).