ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

 ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики

Компьютерные системы и сети

Курс: Методология и технология разработки корпоративных обучаемых экспертных систем

РУКОВОДСТВО РАЗРАБОТЧИКА

к проекту «Сервис по поиску тренера на основе ресурса Портал открытых данных Правительства Москвы»

Бригада:

Меркудинов Сергей

Климкович Александр

Сапко Анастасия

Группа: МКС-183

Преподаватель: Поляков Константин Львович

Оглавление

[**Структура проекта** 3](#_Toc30631029)

[Основной каталог: 3](#_Toc30631030)

[Настройки приложения 4](#_Toc30631031)

[**Инструкция по запуску проекта** 4](#_Toc30631032)

## **Структура проекта**

## Основной каталог:

* **DataAggregator** - каталог со всем исходным кодом приложения;
* **DataReport** - каталог с файлами общего Django-проекта DataReport, внутри которого создано приложение DataAggregator;
* **db.sqlite3** - дефолтная SQLite база фреймфорка;
* **manage.py** - дефолтный скрипт фреймворка.

В проекте используется структура папок и файлов, предлагаемая фреймворком Django. В контексте основного проекта DataReport создано приложение DataAggregator и находится в соответствующем разделе. Все разработанные скрипты также находятся там.

* **data** - каталог с файлами, которые являются источниками данных. Путь к файлу с данными прописан в настройках apps.py.
* **library** - каталог с общими скриптами и классами.  
  В нем описан класс DataLoader, который используется для извлечения данных. Класс предоставляет единый интерфейс для извлечения данных через метод extract(). Класс оперирует настройками, которые задаются в файле apps.py.
* **migrations** - дефолтный каталог django с файлами миграций.
* **static** - каталог с файлами CSS-стилей и JS-скриптов. В проекте для стилевого оформления использован CSS-фреймворк Bootstrap, а для JS-скриптов используется jQuery.
* **templates** – каталог, где хранятся HTML-шаблоны страниц, а также их компонентов. Основным файлом является base\_template.html. Он представляет собой макет страницы и ссылается на 2 компонента: шапку(navbar.html) и подвал(footer.html). На основе базового шаблона страницы строятся все страницы проекта через механизм наследования.
* **init.py** - дефолтный инициализационный файл
* **admin.py** - дефолтный файл моделей для админ-панели django
* **apps.py** - файл с настройками приложения. Можно задать способ извлечения данных, параметры извлечения и максимальное количество записей
* **models.py** - файл моделей данных. В нём описана модель данных сущности "Тренер"(Coach), на основе которой создаются объекты и обогащаются данными от загрузчика DataLoader.
* **tests.py** - дефолтный файл для тестов.
* **urls.py** - файл для настройки url-маршрутизации на основе url-паттернов. В зависимости от запрашиваемого url происходит вызов соответствующего отображения(view), которое рендерит и выводит конкретный шаблон(template).
* **views.py** - файл с отображениями (views). Используются представления на основе функций (Function-Based Views, FBV), которые подготавливают данные и вызывают рендер соответствующего HTML-шаблона

## Настройки приложения

Сервис умеет работать с данными в двух режимах:

1. Загружать данные из csv файла
2. Получать данные через API портала открытых данных (data.mos.ru)

Настройки находятся в файле apps.py в каталоге DataAggregator:

* **DATA\_LOAD\_METHOD** - Способ загрузки данных. Принимает 1 из 2 возможных значений: API или CSV
* **DATA\_LOAD\_CSV\_PARAMS** - Настройки для извлечения данных из csv-файла
* **DATA\_LOAD\_API\_PARAMS** - Настройки для подключения к внешнему API
* **DATA\_LOAD\_MAX\_RECORDS** - Максимальное количество извлекаемых записей

## **Инструкция по запуску проекта**

Windows

1. Установить Python 3 (<https://www.python.org/downloads/>)
2. Установить Django:

pip install django

1. Скопировать репозиторий (при наличии установленного git) или просто скачать файлы из репозитория:

git clone https://github.com/SergeyMerkudinov/DataReport.git

1. Перейти в папку с проектом и запустить сервер в командной строке:

python manage.py runserver

1. Открыть браузер и перейти по адресу localhost:8000.