

Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
Факультет инфокоммуникационных технологий

ЛАБОРАТОРНАЯ РАБОТА №3
Реализация серверной части на django rest.
Документирование API.

Выполнил:

Литвак И. Г., гр. К33401

Проверил:

Говоров А. И.

Цель работы: овладеть практическими навыками реализации серверной части (backend) приложений средствами Django REST framework. **Оборудование:** компьютерный класс

Программное обеспечение: Python 3.9, библиотеки Python: Django, Django REST Framework, Djoser, Swagger, MkDocs

Выполнение работы

Вариант: приложение погоды

Реализован список городов, авторизация, добавление избранных городов пользователю, автоопределение города по координатам, получение прогноза для города (результат кэшируется)

Swagger:

The screenshot displays the Swagger UI for the 'Weather API documentation v1'. At the top, the Swagger logo is on the left, the URL 'http://127.0.0.1:8000/swagger/?format=openapi' is in the center, and an 'Explore' button is on the right. Below the title, the base URL is specified as 'http://127.0.0.1:8000/api/'. A 'Schemes' dropdown menu is set to 'HTTP'. On the right, there are 'Django Login' and 'Authorize' buttons. A 'Filter by tag' input field is present. The API endpoints are organized into two main sections: 'auth' and 'cities'. The 'auth' section includes two POST endpoints: '/auth/jwt/create/' (auth_jwt_create_create) and '/auth/jwt/refresh/' (auth_jwt_refresh_create). The 'cities' section includes ten endpoints: GET '/cities/' (cities_list), GET '/cities/closest/' (cities_closest), GET '/cities/favourite/' (cities_favourite_list), POST '/cities/favourite/' (cities_favourite_create), GET '/cities/favourite/{id}/' (cities_favourite_read), PUT '/cities/favourite/{id}/' (cities_favourite_update), PATCH '/cities/favourite/{id}/' (cities_favourite_partial_update), DELETE '/cities/favourite/{id}/' (cities_favourite_delete), GET '/cities/{id}/' (cities_read), and GET '/cities/{id}/forecast/' (cities_forecast). Each endpoint is color-coded by method (green for POST, blue for GET, orange for PUT, red for DELETE, light green for PATCH) and includes a lock icon.

Документация MkDocs (наполнение создано с использованием библиотеки openapi-generator):

Weather API docs

Search docs

Home

Documentation for API Endpoints

Documentation for Models

Documentation for Authorization

Auth

Cities

Next »

Docs » Home

Documentation for Weather API documentation

Documentation for API Endpoints

All URIs are relative to `http://localhost:8000/api`

Class	Method	HTTP request	Desc
AuthApi	authJwtCreateCreate	POST /auth/jwt/create/	Take
AuthApi	authJwtRefreshCreate	POST /auth/jwt/refresh/	Take
AuthApi	authJwtVerifyCreate	POST /auth/jwt/verify/	Take
AuthApi	authUsersActivation	POST /auth/users/activation/	
AuthApi	authUsersCreate	POST /auth/users/	
AuthApi	authUsersDelete	DELETE /auth/users/{id}/	
AuthApi	authUsersList	GET /auth/users/	
AuthApi	authUsersMeDelete	DELETE /auth/users/me/	

Weather API docs

Search docs

Home

Auth

Cities

« Previous

Docs » Cities

CitiesApi

All URIs are relative to `http://localhost:8000/api`

Method	HTTP request	Description
citiesClosest	GET /cities/closest/	
citiesFavouriteCreate	POST /cities/favourite/	
citiesFavouriteDelete	DELETE /cities/favourite/{id}/	
citiesFavouriteList	GET /cities/favourite/	
citiesFavouritePartialUpdate	PATCH /cities/favourite/{id}/	
citiesFavouriteRead	GET /cities/favourite/{id}/	
citiesFavouriteUpdate	PUT /cities/favourite/{id}/	
citiesForecast	GET /cities/{id}/forecast/	
citiesList	GET /cities/	
citiesRead	GET /cities/{id}/	

Weather API docs

Search docs

Home

Auth

Cities

« Previous

citiesRead

City citiesRead(id)

ViewSet for cities

Parameters

Name	Type	Description	Notes
id	Integer	A unique integer value identifying this City.	[default to null]

Return type

City

Authorization

Basic

HTTP request headers

Вывод: я закрепил навыки Django REST framework, научился создавать автоматическую документацию при помощи Swagger и MkDocs