

POSTGIS

Добавляем полигоны
Удаляем полигоны
Смотрим на карте
Получаем API

ИНСТРУКЦИЯ ДЛЯ РАЗРАБОТЧИКА

Разработал: Евгений Москвин

ВВЕДЕНИЕ

Web сервис добавления полигонов в базу данных PostgreSQL реализован на языке программирования Python с использованием фреймворка Django.

База данных PostgreSQL (с использованием Postgis) развернута по адресу **45.132.18.6:54320**

Для визуального просмотра данных БД можно воспользоваться pgAdmin, развернутый по адресу <http://45.132.18.6:5050> (login: admin@admin.ru pass: root).

PostgreSQL и pgAdmin развернуты с использованием docker-compose.

Для работы с данными в проекте используется Django ORM.

Frontend реализован с использованием шаблонизатора Django, фреймворка Bootstrap 5 (для адаптивной сетки), с применением JQuery.

ТЕРМИНЫ

Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC.

ORM (англ. Object-Relational Mapping, рус. объектно-реляционное отображение, или преобразование) — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

Bootstrap (Twitter Bootstrap) — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML и CSS шаблоны оформления для типографики, веб форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript расширения.

PostgreSQL — свободная объектно-реляционная система управления базами данных (СУБД).

pgAdmin — Представляет собой программное обеспечение с открытым исходным кодом для разработки и администрирования баз данных PostgreSQL

ОПИСАНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ

База данных содержит 1 «пользовательскую» таблицу:

1. **PolygonsModel** - Таблица полигонов

Таблица возможно редактирования из административной панели django, находящейся по адресу: *****/admin

POLYGONSMODEL (ТАБЛИЦА ПОЛИГОНОВ)

1. **id** - id номер записи, автоинкремент, тип int
2. **name_of_polygon** - Название полигона, тип CharField, max_lenght=100
3. **polygon** - Полигон, тип PolygonField
4. **antimeridian** - Антимеридиан, тип BooleanField, default=False

ПРИЛОЖЕНИЯ В ПРОЕКТЕ

Проект содержит два приложения:

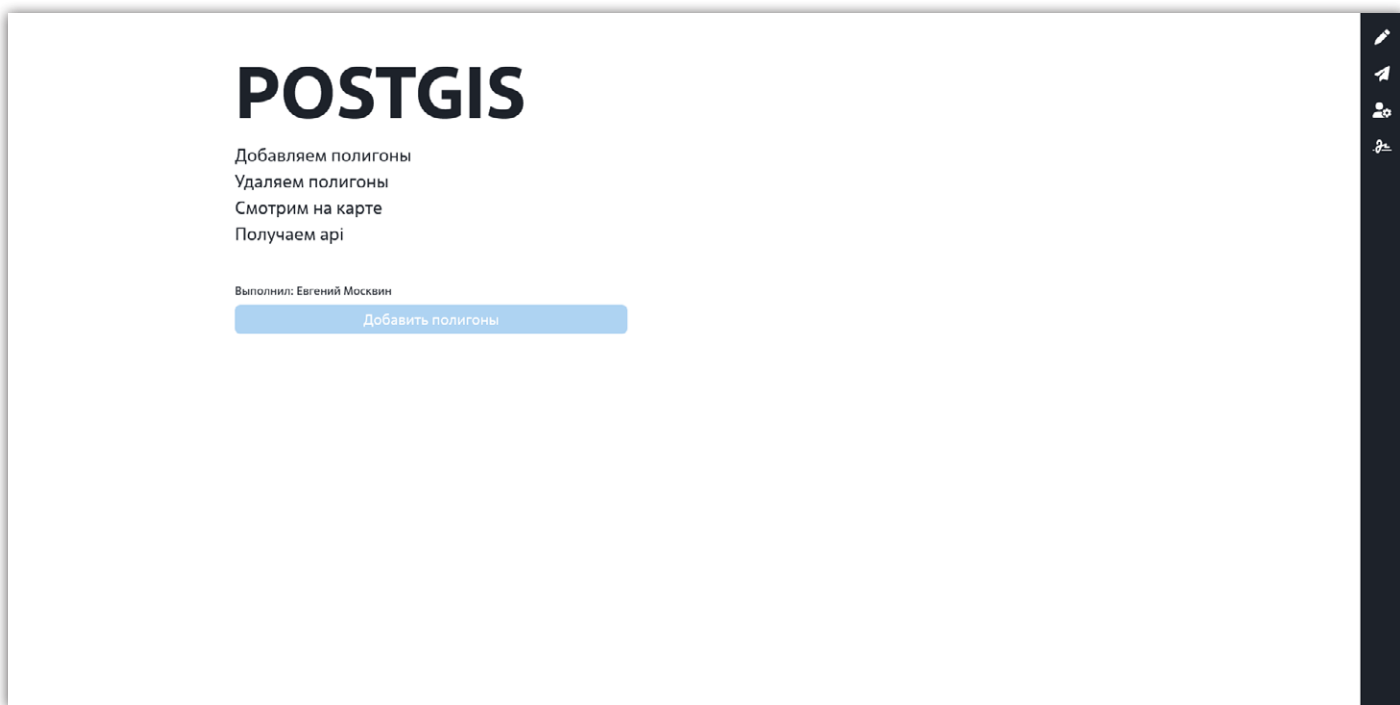
1. **postgis_app** - Приложение в котором описаны все модели базы данных и представления для дальнейшего взаимодействия с проектом путем веб страниц
2. **postgis_api** - Приложение в котором реализовано api с использованием Django REST Framework (DRF)

VIEWS.PY (POSTGIS_APP)

В основной части проекта реализованы 4 основных представления (View)

ABOUTVIEW (GET)

Представление AboutView предназначено исключительно для отображения страницы About,



Страница About

INDEXVIEW (GET, POST)

Представление IndexView предназначено для добавления данных о полигонах в БД.

На клиентской части формируется массив координат `array_to_send`.

С использованием ajax (jquery) полученные данные преобразовываются в json строку и отправляются в IndexView методом post, где происходит обработка полученных данных.

Основные переменные представления IndexView:

`name_polygon` - название полигона, строка полученная из `request.POST.get('namePolygon')`

`send_text` - json строка с координатами полигона, полученная из `request.POST.get('sendText')`

`data_from_json` - список пар координат, полученный преобразованием через `json.loads(send_text)`

Полученный список data_from_json проверяется циклом for, определяются пустые значения, если таковые были и приравнием их к нулю, определяем антимеридиан (в случае если долгота больше 180°, добавляем к записи флаг antimeridian = True) и добавляем это значение в result_list.

В конечном результате формируем новую запись в PolygonsModel(name_of_polygon=name_polygon, polygon=polygon, antimeridian=antimeridian), и возвращаем id данной записи.

На клиентской стороне выполняем переход на страницу с картой по данной записи.

Добавить точки к полигону

Широта

Долгота

Добавить

Данные для отправки

Название полигона

Координаты:

Отправить

Страница «Добавить полигон»

SHOWLISTPOLYGONSVIEW (GET)

Представление ShowListPolygonsView предназначено для отображения информации о всех полигонах в БД.

Для более читаемого представления координат используется templatetag: get_coordinates

Все полигоны

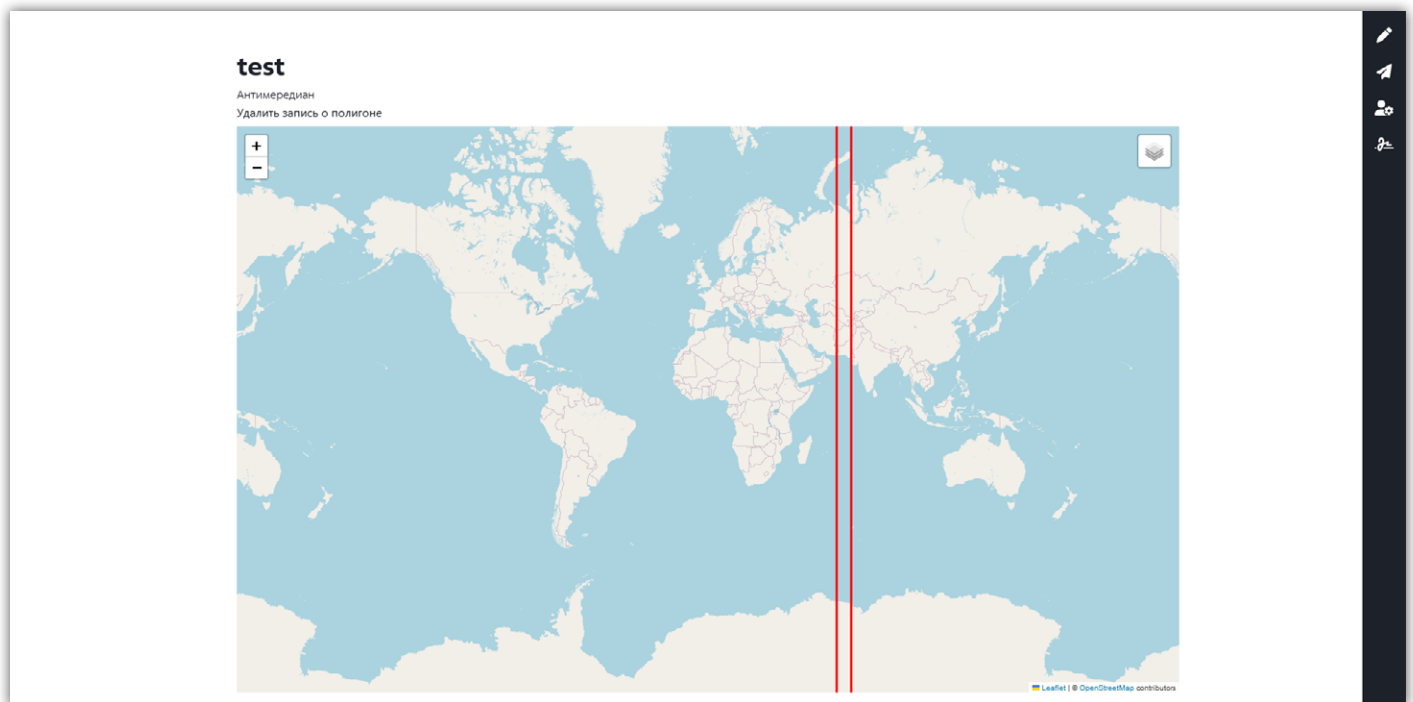
#	Название полигона	Координаты полигона	Антимеридиан
1	Тест изменяем по ари	[[(-0.0170889922606532, 0.0023850229093604), (-0.0172297563051993, -0.0002543029251871), (-0.0127956889019978, -7.9658472355e-06), (-0.0131124080022265, 0.003370371219933), (-0.0170889922606532, 0.0023850229093604)]]	
2	Тест через админку 2	[[[30.251658873134755, 60.05986571563659], (30.17439425733208, 59.61666654478877), (31.075814775029972, 59.63294556454888), (31.19171169873399, 60.101615183672784), (30.251658873134755, 60.05986571563659)]]	
3	тест загрузки	[[0.0, 0.0], (0.0, 50.0), (50.0, 50.0), (50.0, 0.0), (0.0, 0.0)]	
4	тест загрузки	[[0.0, 0.0], (0.0, 50.0), (50.0, 50.0), (50.0, 0.0), (0.0, 0.0)]	
5	тест загрузки	[[0.0, 0.0], (0.0, 50.0), (50.0, 50.0), (50.0, 0.0), (0.0, 0.0)]	
6	тест загрузки	[[0.0, 0.0], (0.0, 50.0), (50.0, 50.0), (50.0, 0.0), (0.0, 0.0)]	
7	test	[[69.3493386397765, 174.19921875000003], (69.3493386397765, -155.03906249999997), (62.103882522897884, -154.51171874999997), (62.186013857194226, 173.67187500000003), (69.3493386397765, 174.19921875000003)]]	✓
8	Тест с формы2	[[123.0, -39.0], (112.0, -107.0), (310.0, -70.0), (123.0, -39.0)]	✓
9	test redirect	[[100.0, -160.0], (200.0, -110.0), (300.0, 150.0), (100.0, -160.0)]	✓
10	тест запятые	[[45.321, 12.0], (45.321, 12.0), (45.321, 12.0), (45.321, 12.0)]	
11	тест точки	[[123.25, -174.68], (180.25, -129.68), (190.25, -109.68), (123.25, -174.68)]	✓
12	123	[[123.0, -39.0], (80.0, -39.0), (80.0, -39.0), (123.0, -39.0)]	✓
13	121	[[123.0, -39.0], (123.0, -39.0), (123.0, -39.0), (123.0, -39.0)]	✓
14	12312	[[123.0, -39.0], (123.0, -39.0), (123.0, -39.0), (123.0, -39.0)]	✓
15	3244	[[50.0, 60.0], (50.0, 60.0), (50.0, 60.0), (50.0, 60.0)]	
16	3244	[[50.0, 60.0], (50.0, 60.0), (50.0, 60.0), (50.0, 60.0)]	

Страница просмотра всех полигонов

SHOWONMAPVIEW (GET, POST)

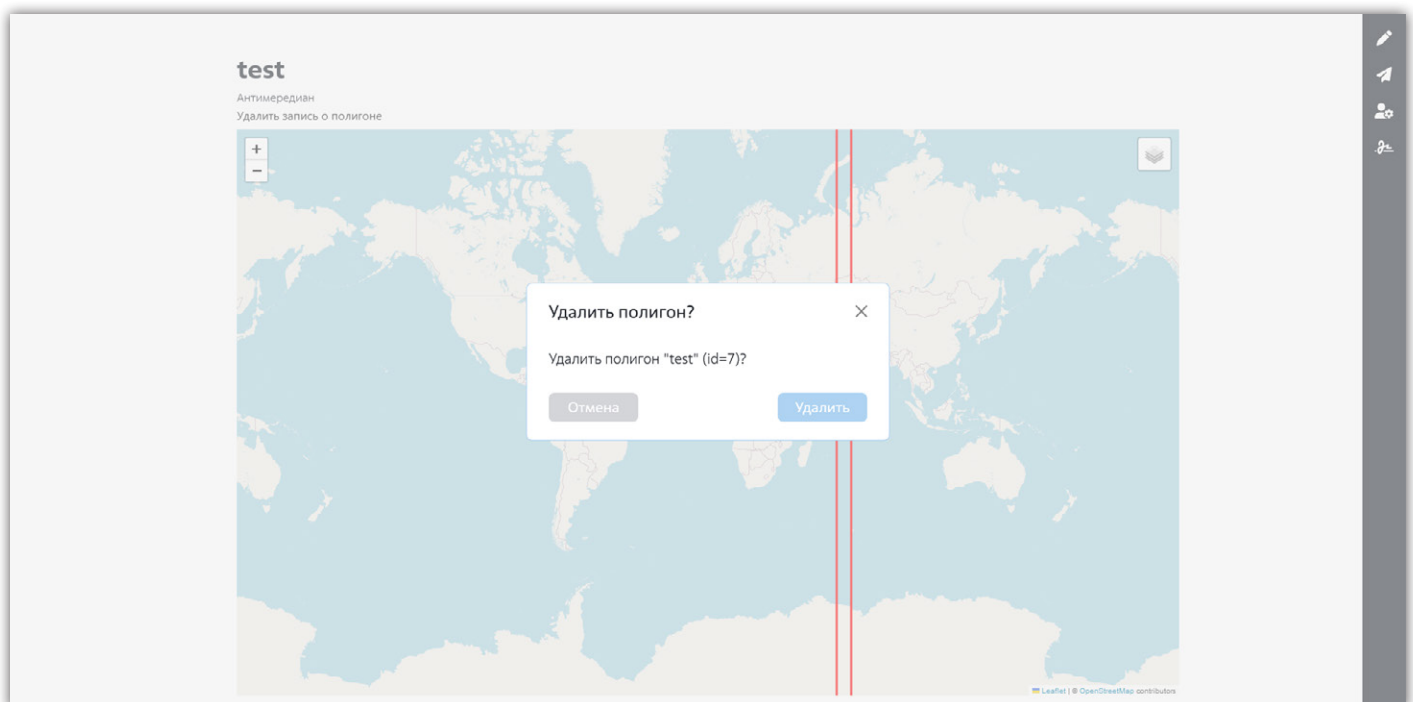
Представление ShowOnMapView предназначено для отображения информации о конкретном полигоне.

В методе get реализовано отображение данного полигона на карте.
Для отрисовки карты используется библиотека folium.



Страница просмотра информации о полигоне

В методе post реализовано удаление записи о данном полигоне



Модальное окно для удаления полигона

VIEWS.PY (POSTGIS_API)

В данном проекте API реализовано с использованием Django REST Framework (DRF), путем наследования от существующих классов

Реализованы 3 основных представления:

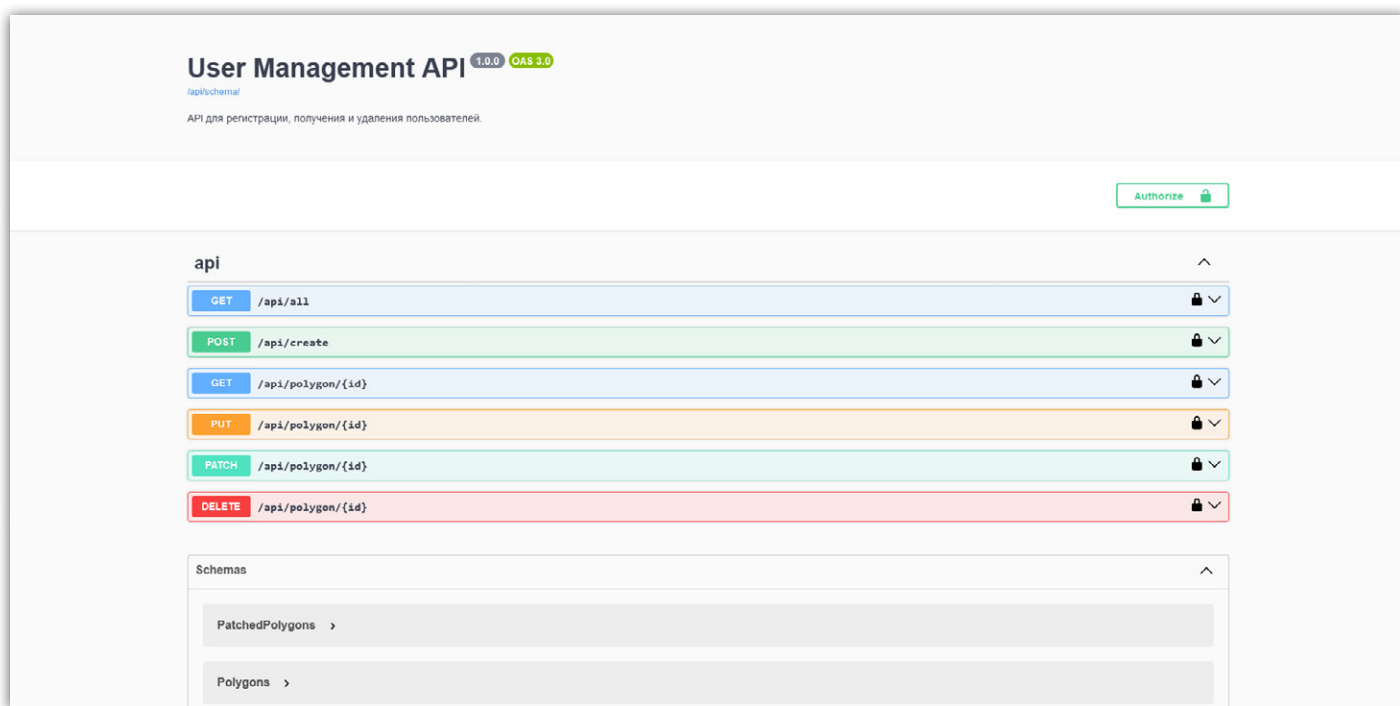
1. **CreatePolygonAPIView** - создание записи в БД, унаследовано от класса CreateAPIView
2. **ListPolygonsAPIView** - просмотр записей в БД, унаследовано от класса ListAPIView
3. **OnePolygonApiGenericView** - просмотр, редактирование и удаление записи в БД, унаследовано от класса RetrieveUpdateDestroyAPIView

SERIALIZERS.PY (POSTGIS_API)

Сериализация данных проходит с использованием класса ModelSerializer

SWAGGER

В проект добавлена библиотека Swagger для тестирования и документирования API



Swagger