

Тестовое задание "Begemotic MVP"

Менеджер продукта begemotic (BEstplace GEO Machine fOr Tlptop Calculations) хочет запустить сервис расчета геофакторов по HTTP API. Он просит тебя для начала реализовать MVP сервиса: расчет некоторых видов геоагрегаций на одном датасете — сэмпле многоквартирных домов Москвы (см. [apartments.csv](#)). Описание полей "apartments.csv" см. ниже.

Поле	Описание поля
id	уникальный идентификатор дома
geopos	координаты дома в формате geojson (порядок координат: долгота, широта)
apartments	количество квартир в доме
price	средняя стоимость квартир в доме за 1 кв. м
year	год постройки дома

Требования к MVP

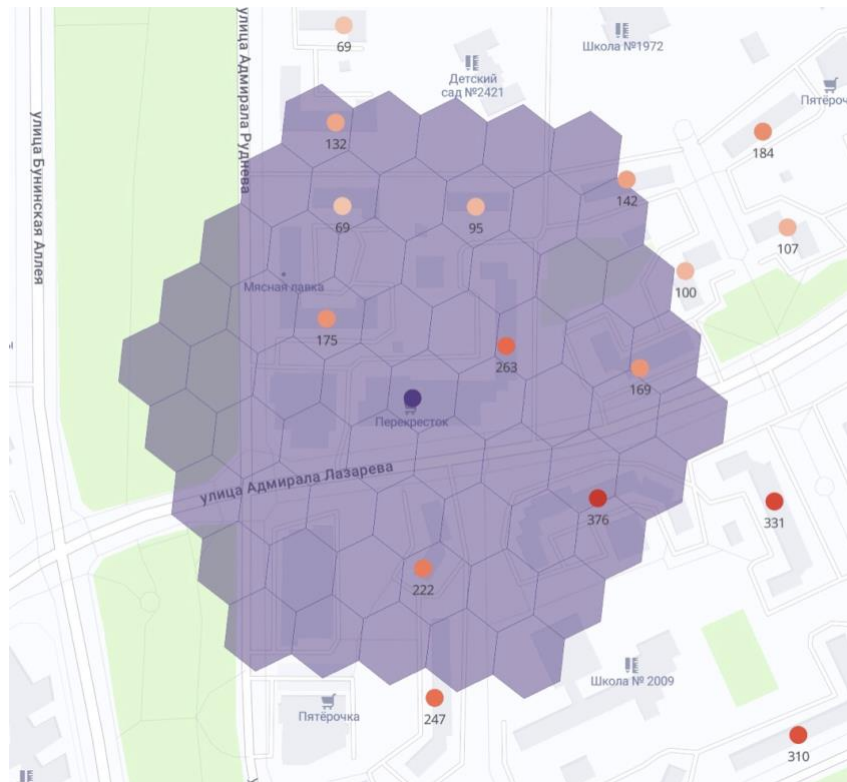
- Сервис должен быть реализован с помощью библиотек [h3-py](#) (версии 3.x и 4.x сильно отличаются, в этом документе приведены функции из версии 3.x) и [fastapi](#).
- Для расчетов агрегаций следует использовать h3 точность (resolution) = 11. Длина стороны h3 гексов такой точности в Москве примерно равна 30м.
- Сервис должен обладать следующими API методами:
 - Посчитать агрегацию в радиусе k гексов от точки (см. [h3.geo_to_h3](#) и [h3.k_ring](#)). На вход метод должен принимать следующие параметры:

Параметр	Описание
geometry	координаты точки в формате geojson
field	поле датасета
aggr	тип агрегации (допустимые варианты: sum — сумма, avg — среднее значение, min — минимальное значение, max — максимальное значение)
r	размер радиуса в гексах

Пример данных POST запроса к методу (расчет суммы квартир в радиусе 4
гексов от заданной точки):

```
{
  "geometry": {
    "type": "Point",
    "coordinates": [37.517259, 55.542444]
  },
  "field": "apartments",
  "aggr": "sum",
  "r": 4
}
```

Для данного примера результат расчёта будет равен 1501:



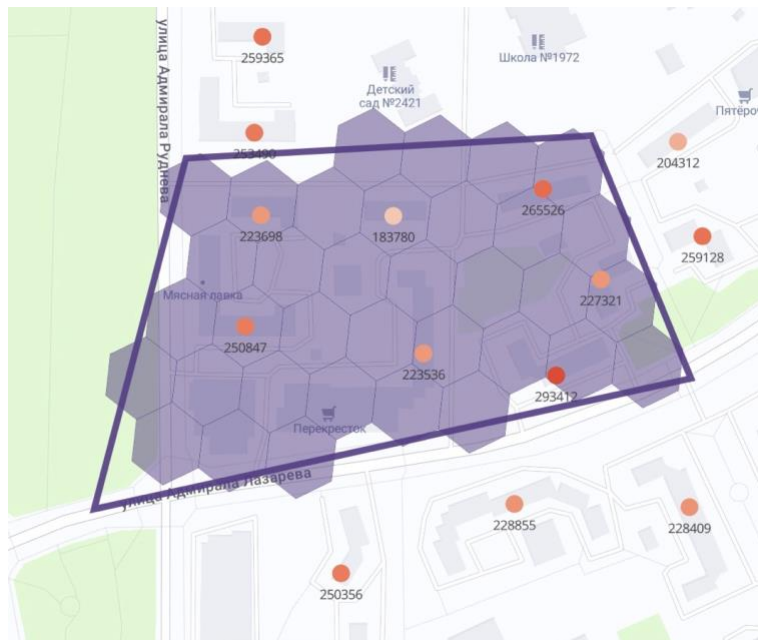
- b. Посчитать агрегацию в заданном полигоне, полигон должен аппроксимироваться с помощью гексов (см. [h3.polyfill](#)). На вход метод должен принимать следующие параметры:

Параметр	Описание
geometry	координаты точки в формате geojson
field	поле датасета
aggr	тип агрегации (допустимые варианты: sum — сумма, avg — среднее значение, min — минимальное значение, max — максимальное значение)

Пример данных POST запроса к методу (расчет средней стоимости квадратного метра в заданном полигоне):

```
{
  "geometry": {
    "type": "Polygon",
    "coordinates": [[
      [37.520123, 55.54413],
      [37.515671, 55.54399],
      [37.514662, 55.541793],
      [37.521218, 55.542612],
      [37.520123, 55.54413]
    ]]
  },
  "field": "price",
  "aggr": "avg"
}
```

Для данного примера результат расчёта будет равен примерно 238302.86:



4. В случае некорректно вызванного метода API сервис должен возвращать понятное описание ошибки.
5. Сервис должен разворачиваться внутри docker образа.

Результат

После выполнения задания нужно отправить архив со следующим содержанием:

1. Исходный код проекта, написанный на python3.
2. Файл docker-compose.yml, с помощью которого можно собрать и развернуть контейнер сервиса.
3. Файл readme с описанием процесса разворачивания и (по желанию) комментариями.