## Задание

Имеется множество объектов (зданий), состояние которых контролируется с помощью датчиков. С одним объектом может быть связано несколько датчиков. Все датчики измеряют один параметр (температуру) и отправляют измерения на сервер.

Вам надо реализовать сервис, который будет хранить измерения датчиков и помогать пользователям работать с этими данными. Доступ к сервису должен выполняться через REST API со следующими функциями:

* */api/save:* сохранить измерения датчиков, передаваемые как список json структур (объект, датчик, время в секундах, значение).

Пример тела запроса:

[

{

"objectId": 1,

"sensorId": 1,

"time": 1565796600,

"value": 9.4

},

{

"objectId": 1,

"sensorId": 2,

"time": 1565796720,

"value": 14.2

}

]

* */api/history:* выдать все измерения заданного датчика за заданный интервал времени;

Пример запроса:

/api/history?id=1&from=1565654400&to=1565827200

* */api/latest:* выдать текущие (последние на данных момент) значения всех датчиков для заданного объекта.

Пример запроса:

/api/*latest*?id=1

* */api/avg:* выдать среднее из текущих значений датчиков для каждого объекта.

Пример запроса:

/*api/avg*

Измерения датчиков должны храниться в PostgreSQL. Реализация сервиса, включая SQL-запросы, должна обеспечивать эффективную работу с большим числом (суммарно более 1М измерений по всем датчикам) хранимых данных.

Для тестирования приложения к заданию предоставляется генератор тестовых данных *generate\_sensor\_data.py.*

*Пример запуска генератора:*

*./generate\_sensor\_data.py -t 1514764800 -p 10 -N 2 -n 3 -s 42*

## Пожелания и требования к выполнению задания

### Пожелания

Ожидаемое время выполнения задания: 4 часа

### Требования

БД приложения должна включать таблицы объектов наблюдения, датчиков и результатов измерений.

На случай отсутствия таблиц в БД необходимо иметь средства их инициализации.

Требования к используемым инструментам: Java 8 или выше, Spring Framework, Gradle, PostgreSQL.

Помимо реализации сервиса ваше решение должно включать минимальную документацию (ReadMe с примерами сборки, запуска приложения и юнит тестов), скрипты для инициализации БД и набор юнит тестов для проверки функционала сервиса.

# Дополнение

generate\_sensor\_data.py - скрипт на python, генерирующий данные для тестового задания по Java. По-умолчанию (без аргументов) генерирует 2.592 М точек = по 3 объектам и 10 сенсорам с каждого объекта и 86400 наблюдений с каждого сенсора. Результат генерации - это валидный json документ - список точек с данными

Подготовительные действия:

chmod 755 ./generate\_sensor\_data.py

Просмотр помощи:

./generate\_sensor\_data.py --help

Как запускать для генерации данных, результата кладется в data.json:

./generate\_sensor\_data.py