

Техническое задание

Обязательные требования

Язык программирования: Kotlin

Серверная часть: Ktor

ORM: Exposed

База данных: PostgreSQL

Тесты: Средства KTOR <https://ktor.io/docs/testing.html#test-app>

Требования

1. Написание микро-сервиса на основе Ktor
2. Создание endpoint `http://127.0.0.1:8080/saveData` куда приходит `uuid` в виде строки и `Graph`:

```
data class Graph(  
    val nodes: List<Node>,  
    val edges: List<Edge>,  
)  
  
data class Node(  
    val name: String,  
    val property: List<Property>,  
)  
  
data class Property(  
    val name: String,  
    val value: String,  
)  
  
data class Edge(  
    val name: String,
```

```
    val source: String,  
    val target: String,  
)
```

3. Необходимо сохранить эту информацию в `Postgres` в `database graph-base`
4. При успешном сохранении данных в БД должна происходить отправка `Graph` на `endpoint http://test-work:8080/data` который возвращает `true` или `false`.
5. Если соблюдены условия:
 - данные успешно сохранены в БД
 - `Graph` успешно отправлен на `http://test-work:8080/data` и он вернул `true`
То `endpoint http://127.0.0.1:8080/saveData` должен вернуть `true`, а при не соблюдении условий `false`
6. В программе должны быть обработаны ситуации:
 - данные пришли в неправильном формате и не десериализуются
 - произошла ошибка при сохранении в БД
 - `http://test-work:8080/data` не ответил или вернул `false`
7. Второй `endpoint http://127.0.0.1:8080/getData`, принимающий `uuid` в виде строки должен возвращать `json` ранее описанного формата `Graph` по его `uuid` или `bad request` если он не найден.
8. На оба `endpoint` необходимо написать по одному тесту средствами `Ktor`.
9. Задание необходимо выполнить в пределах 7 дней, сохранить его на своем `GitHub` и прислать на него ссылку.

Подробности: `http://test-work:8080/data` реализовывать **не надо**, необходимо лишь симулировать обращение к нему через средства `Ktor`. Хотя оно и будет бросать ошибку (которую надо обработать).