Задание

Необходимо разработать сервис "рандомный факт", сервис по запросу должен возвращать один случайный факт из своей БД, так же необходим метод для того чтобы пополнять базу новыми фактами и обновлять информацию о существующих

Описание API

Узнать случайный факт

Запрос: `GET /fact`

Ответ:

```
{
  "id": 1,
  "title": "високосная секунда",
  "description": "Дополнительная (високо́сная, скачущая) секу́нда[4][5][6], или секу́нда координа́ции[7] (англ. leap second) — секунда, иногда добавляемая (теоретически возможно и вычитание) в шкалу всемирного координированного времени (UTC) для согласования его со средним солнечным временем UT1.",
  "links": [
    "shorturl.at/dwzK9"
  ]
}
```

Получить факт по id

Запрос: `GET /fact/1`

Ответ:

```
{
  "id": 1,
  "title": "високосная секунда",
  "description": "Дополнительная (високо́сная, скачущая) секу́нда[4][5][6], или секу́нда координа́ции[7] (англ. leap second) — секунда, иногда добавляемая (теоретически возможно и вычитание) в шкалу всемирного координированного времени (UTC) для согласования его со средним солнечным временем UT1.",
  "links": [
    "shorturl.at/dwzK9"
  ]
}
```

Пополнить базу, запрос принимает на входе массив новых фактов, в ответе возвращает их id

Запрос:

```
POST /fact
{
  facts: [
    {
      "title": "без ссылок",
      "description": "некоторые факты не имеют ссылок на дополнительную информацию",
    },
    {
      "title": "с ссылками",
      "description": "некоторые факты имеют ссылки на дополнительную информацию",
      "links": [
        "http://ozon.ru"
      ]
    }
  ]
}
```

Ответ:

```
{
  "ids": [2, 3]
}
```

Обновить данные по факту

Запрос:

```
PUT /fact/3
{
  "id": 3,
  "title": "факс с ссылками",
  "description": "некоторые факты имеют ссылки на дополнительную информацию",
  "links": [
    "http://ozon.ru"
  ]
}
```

Ответ: `{}`

Требования

Обязательные:

- реализовать GET методы и POST метод для массовой загрузки данных
- в качестве БД можно использовать любую реляционную, например postgres
- язык программирования - golang
- библиотеки - можно пользоваться чем угодно, но лучше обойтись стандартной библиотекой golang и драйвером к БД (lib/pq, pgx в случае постреса)

Будет плюсом:

- реализовать PUT метод для обновления данных
- наличие юнит тестов
- docker compose и инструкция о том как собрать и запустить сервис
- валидация данных в POST/PUT методах