

# Техническое задание

---

## На разработку API

Реализовать простейшее API для учета товаров на складе. Товар выглядит следующим образом:

```
{
  "id" : 0,
  "title" : "New Chair9000",
  "amount" : 150,
  "price" : 12.5
}
```

У API нужно реализовать следующие 5 методов:

- **GET /items** - возвращает все товары на складе и код **200** если есть хотя бы один товар. Код **404** и сообщение `{"Error" : "No one items in stock!"}` - в противном случае.
- **GET /item/{id}** - возвращает **200** и товар с **id**, если он существует в БД. Иначе **404** и сообщение `{"Error": "Item with that id not found!"}`
- **POST /item** - добавляет товар в БД и возвращает **201**. Поля товара указаны выше. Гарантируется, что все товары будут иметь уникальный **id**, будут указаны в порядке возрастания от 0 и выше с шагом в единицу.
- **PUT /item/{id}** - обновляет информацию о товаре с **id**. Возвращает **202** и обновленный товар. Если товара с **id** БД нет - **404** и сообщение `{"Error": "Item with that id not found!"}`
- **DELETE /item/{id}** - удалить товар из БД с **id**. Возвращает **202** и удаленный товар. Если товара с **id** БД нет - **404** и сообщение `{"Error": "Item with that id not found!"}`.

Требования по коду:

- Можно использовать любые зависимости
- БД - только локальная (слайс или мапа)

Решение предоставить в виде ссылки на ваш репозиторий в docker hub в виде: `docker pull <user>/<repo>:latest`

Для тобавления сторонних зависимостей в проект добавьте в **Dockerfile** следующие строки перед этапом сборки проекта в контейнере:

```
...
COPY go.mod .
COPY go.sum .
```

```
RUN go mod download
```

```
...
```