

# Никонов Андрей Алексеевич

## БПИ 238

### Основная информация

- Приложение разрабатывалось на Golang 1.23.6 (это было заранее согласовано с семинаристом)
- Была использована REST архитектура

### Запуск




Предварительно нужно скачать приложение с <https://github.com/nessornot/kpokr2>




Затем

```
docker-compose up --build
```






### Информация для проверяющего

#### Критерии оценки

1. Реализация основных требований к функциональности – **2 балла**
  2. Распределение функциональности по микросервисам – **4 балла**
    - a. Серверная часть веб-приложения состоит минимум из двух микросервисов.
    - b. Обработка ошибок (один из микросервисов упал).
  3. Реализация коллекции Postman / Swagger, которая должна демонстрировать функциональность реализованных микросервисов, охватывая все API – **1 балл**
  4. Качество кода и документация – **2 балла**
    - a. Хорошо организованный, модульный и “чистый” код.
    - b. Документация, включая краткое описание архитектуры системы и спецификацию API.
  5. Покрыто тестами более 65% кода (**1 балл**)
- 
1. Реализация основных требований к функциональности - **выполнено**  . Приложение выполняет подсчет статистики.
  2. Распределение функциональности по микросервисам - **выполнено**  . Серверная часть состоит из 3х микросервисов (api\_gateway, file\_storing, file\_analysis). Ошибки обрабатываются корректно.
  3. Реализация коллекции Swagger - **выполнено**  . Доступно для проверки по адресу `http://localhost:8080/swagger/index.html` (предварительно запустив приложение).

4. Качество кода - **выполнено**  . Код разбит на логические элементы, все понятно и красиво. Документация - **частично**  . Присутствуют комментарии в коде, но полноценную документацию я писать не буду.
5. Покрыто тестами более 65% кода. - не выполнено  . Я не писал тесты.

Мое мнение об итоговой оценке КР2:

1. Реализация основных требований к функциональности – **2 балла** 
2. Распределение функциональности по микросервисам – **4 балла** 
3. Реализация коллекции Postman / Swagger, которая должна демонстрировать функциональность реализованных микросервисов, охватывая все API – **1 балл** 
4. Качество кода и документация – **2 балла** 
5. Покрыто тестами более 65% кода (**1 балл**) 

$$2\checkmark + 4\checkmark + 1\checkmark + 1\text{orange} + 0\text{red} = 8 - 2 \text{ (просрочка 2 дня)} = 6$$

- Итого 6