# Зад. 01. Restful API.

## Выберите произвольную предметную область. Для нее создайте Restful API.

Обратите внимание, что контролеры не должны работать с сущностями ORM, только с DTO.

Между уровнем контролеров и сервисов поместите уровень фасадов. Методы фасадов принимают и возвращают только DTO или простые классы типа String (ни в коем случае не ORM сущности). Внутри фасадов используются сервисы. Фасады нужны, в т. ч. чтобы упростить интерфейс взаимодействия для контроллеров. Внутри их методов можно вызывать несколько сервисов.

В контроллеры не инжектировать сервисы, а только фасады.

Сущности возвращаются сервисным уровнем. На сервисном же уровне находится бизнес логика. Методы у сервисов не делайте большими, чтобы их легко было переиспользовать.

Много сущностей и эндпоинтов не создавайте. Покажите качество кода, а не его количество. Также примените различные технологии и подходы. Покажите целесообразное применение паттернов, различных аннотаций ORM, аутентификацию и прочее.

## Обязательные требования:

- Протестировать веб сервисы с помощью Rest Assured.

## Опциональные требования:

- аутентификация

## Stack:

- Java 17

- любая SQL DB

- Hibernate или Spring Data

- Spring Boot

- Rest Assured

## Некоторые частые ошибки, когда задание **не сдано**:

- нет уровня фасадов

- в контроллере есть хотя бы один java импорт сервиса, репозитория (DAO) или сущности

- хотя бы один метод фасада возвращает или принимает сущность.

- если код конвертации из сущности в DTO (или наоборот) происходит внутри метода фасада. Конвертацию нужно вынести в отдельный компонент Маппер, чтобы можно было переиспользовать.

- если встречаются JSON аннотации в сущностях. Такие аннотации должны быть в DTO.

- в сервисах импортируются DTO. DTO должны быть выше уровня сервисов, то есть начиная с фасадов.

- мапперы для конвертации сущностей в DTO не должны быть на уровне сервисов.

- сервисы и фасады написаны просто классами, без интерфейсов. Нужно использовать интерфейсы, инжектировать в другие компоненты их, а не классы.

- есть неиспользуемые импорты в классах