# Тестовое задание по Java

# Вариант A (легкий)

## Сделать сервис Producer

Сервис должен генерировать и отправлять в кафку сообщения в формате json-объекта {code: string, label: string}  
Сервис должен предоставлять REST-интерфейс для генерации одного или нескольких сообщений

## Сделать сервис Consumer

 Сервис должен читать из кафки сообщения, отправленные сервисом Producer и сохранять их в БД.  
 Также продемонстрировать, как создать объекты в БД.

## Сделать сервис UI-Back

Сервис должен предоставлять CRUD+список REST-интерфейс к данным в БД, загруженным сервисом Consumer.

# Вариант Б (усложненный)

## Сделать сервис Producer

Сервис должен генерировать и отправлять в кафку сообщения в формате json-объекта {code: string, label: string}

В кафке использовать партиционирование, грамотно использовать Key сообщения, настроить время жизни сообщений. Конфигурацию топиков при этом определить через Java-код.

Сервис должен предоставлять REST-интерфейс для генерации одного или нескольких сообщений.

## Сделать сервис Consumer

Сервис должен читать из кафки сообщения, отправленные сервисом Producer и сохранять их в БД.

Сервис должен быть масштабируемым, т.е. запускаться в нескольких экземплярах, при этом обеспечить однократное вычитывание сообщений из кафки.

Также продемонстрировать, как создать объекты в БД несколькими вариантами - с помощью средств Java, с помощью SQL. Будет плюсом демонстрация работы Liquibase.

## Сделать сервис UI-Back

Сервис должен предоставлять CRUD+список REST-интерфейс к данным в БД, загруженным сервисом Consumer. Дополнительно в методе для списка сделать пагинацию.

## Сделать сервис CallExternalRest

Сервис должен по расписанию запрашивать новые или отредактированные записи через REST-методы сервиса UI-BACK и сохранять их в журнал изменений. Журнал можно реализовать в БД или в Кафке.

Иделаьный варинт - реализовать журнал изменений в Elastic Search. При необходимости можно дополнить таблицу в БД и методы UI-Back необходимыми полями.