Вам нужно разработать сервис с REST API на основе HTTP с передачей данных в формате JSON. Сервис должен вести список заданий (Task). В задание загружаются изображения (Image). Затем задание запускается на обработку. В рамках обработки с помощью стороннего сервиса на изображениях необходимо найти лица (Face), определить их пол и возраст. На основе полученных данных в рамках задания нужно получить статистики.

Задание имеет один из статусов:

- Формируется — задание создано и пока ещё в него можно загружать изображения.
- Обрабатывается — начата обработка задания. В задание с таким статусом загружать новые изображения нельзя.
- Обработка завершена — все изображения обработаны и доступны вычисленные статистики. В задание с таким статусом загружать новые изображения нельзя.
- Ошибка обработка — одно или несколько изображений не удалось обработать. В задание с таким статусом загружать новые изображения нельзя, запускать повторную обработку также нельзя.

Методы

API сервиса должен поддерживать следующие методы:

- Добавление задание.
- Получение задания. В ответе на запрос должны быть:
  - Идентификатор задания.
  - Статус задания.
  - Список изображений в задании, содержащий для каждого изображения:
    - Его название.
    - Список лиц, найденных на изображении, содержащий для каждого лица:
      - Bounding box лица.
      - Пол.
      - Возраст.
  - Статистики:
    - Суммарное число лиц на всех изображениях в задании.
    - Суммарное число мужчин и женщин на всех изображениях в задании.
    - Средний возраст мужчин на всех изображениях в задании.
    - Средний возраст женщин на всех изображениях в задании.
- Запуск обработки задания. Обработка должна выполняться асинхронно. Данный метод возвращает ответ сразу после запуска обработки и не дожидается её завершения.
- Удаление задания. При удалении задания все связанные с ним данные должны быть удалены. Удалять задание в статусе "Обрабатывается" пользователю API должно быть запрещено.
- Добавление изображения в задание. Вместе с изображением передается его название.

Требования

- Версия Golang не ниже 1.20.
- В качестве стороннего сервиса необходимо использовать FaceCloud.
    - Документация к API сервиса доступна по адресу https://docs.facecloud.tevian.ru.
    - Про создание аккаунта для использования сервиса можно узнать в разделе How to create demo account.
    - Вам понадобится запрос POST /api/v1/detect.
    - Обратите внимание, что FaceCloud работает только с JPEG изображениями. Разрабатываемый сервис должен уметь принимать как минимум JPEG изображения. Поддержка других типов не обязательна.
- Допустимо при перезапуске сервиса терять данные о добавленных заданиях, изображениях и статусе их обработки. Использование СУБД для хранения этих данных не обязательно.
- Файлы с изображениями должны храниться на диске с использованием файловой системы.
- Запросы к FaceCloud должны отправляться параллельно. На каждое изображение нужно запустить свою горутину, либо обрабатывать изображения в пуле из заданного числа горутин (не менее 4-х).
- Должна быть возможность с помощью environment variables сконфигурировать:
    - Логин и пароль для FaceCloud.
    - URL API FaceCloud-а.
- К сервису нужно подготовить README, которое будет содержать:
    - Инструкцию по запуску сервиса (что нужно установить, что нужно настроить, какие команды ввести). Инструкция должна в том числе содержать и инструкцию по компиляции сервиса. Прикладывать готовый бинарный файл в задание нельзя.
    - Описание доступных настроек сервиса.
    - Описание API сервиса.

Необязательные требования (будет плюсом)

Ниже перечислены необязательные требования, выполнение любого из которых будет считаться плюсом. Можно выполнить только часть необязательных требований.

- Предоставлен Dockerfile, позволяющий собрать контейнер с сервисом.
- Предоставлен docker-compose.yml, позволяющий запустить сервис и другие связанные сервисы, которые потребуются.
- Для хранения данных (кроме изображений) используется СУБД. Предпочтительно использование PostgreSQL, но другие СУБД допустимы.
    - Для работы с СУБД можно использовать любой удобный вам подход: ручное формирование запросов, использование ORM библиотек.
- Доступ к API должен быть защищен с помощью HTTP Basic Auth. Логин и пароль должны конфигурироваться через environment variables.
- Нужно разрешить удалять задания в статусе "Обрабатывается" и при удалении корректно прекращать запущенную обработку.

- Нужно разрешить запускать повторную обработку заданий со статусом "Ошибка обработки".
    - Дополнительным плюсом будет считаться, если уже обработанные успешно изображения в таком случае не будут обработаны повторно.

Прочие примечания

- Код сервиса и README необходимо прислать в виде zip-архива.
- Если вы испытываете трудности с использованием FaceCloud или у вас возникли вопросы по заданию, не стесняйтесь обратиться к нам за помощью.