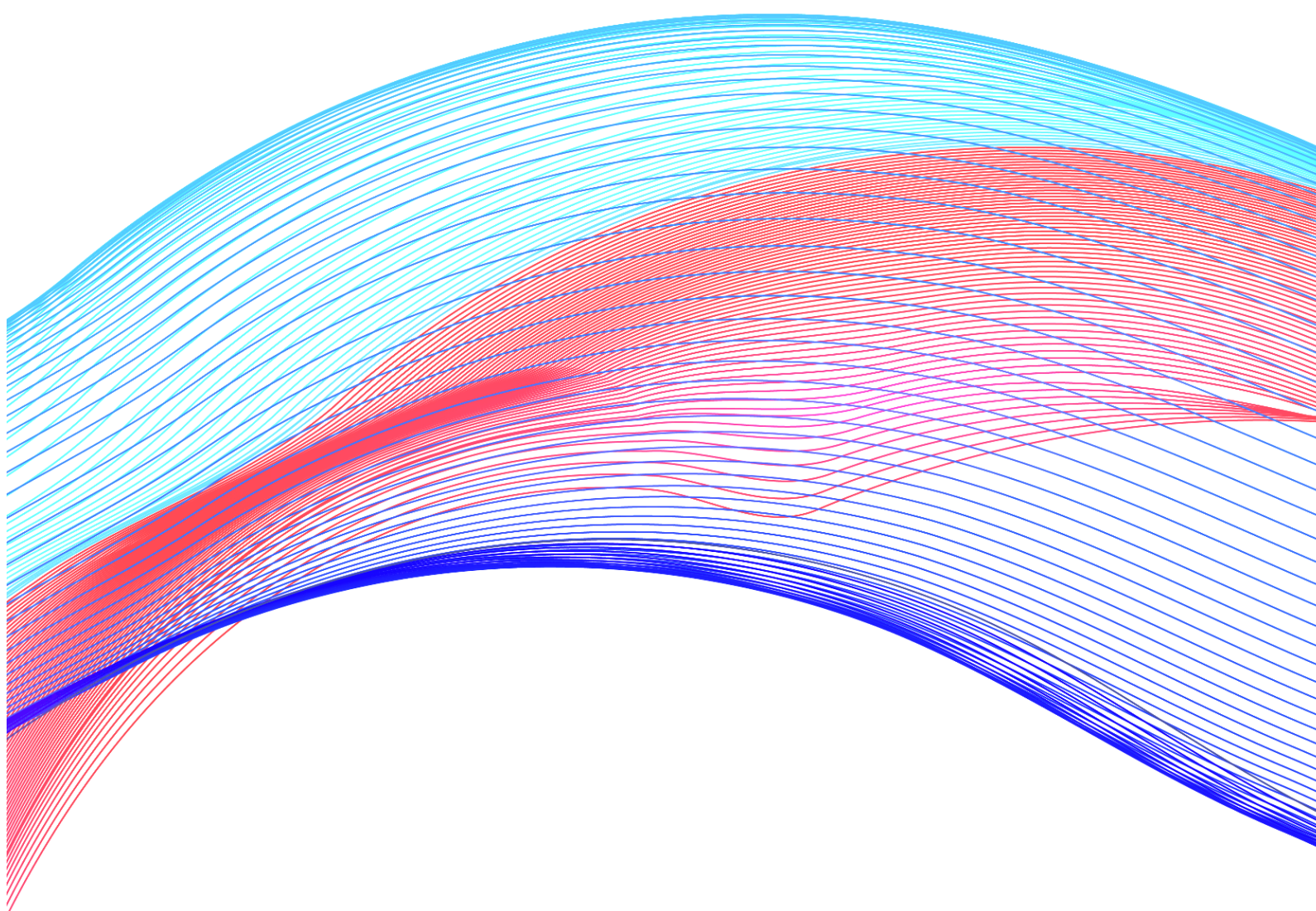




ООО "ТРИДИВИ"

# Image API 1.0.1

## Руководство пользователя



2023 г.

## Содержание

<b>Глоссарий</b>	<b>3</b>
<b>Обзор</b>	<b>4</b>
<b>Архитектура</b>	<b>5</b>
<b>Доступ к Image API через Swagger</b>	<b>7</b>
<b>Структура запроса в Swagger</b>	<b>8</b>
<b>Описание атрибутов</b>	<b>10</b>
<b>Применение</b>	<b>13</b>
Детекция лиц	13
Детекция силуэтов	13
Определение пола	14
Определение возраста	15
Оценка эмоций	15
Проверка liveness	17
Определение антропометрических точек лица	18
Проверка на наличие медицинской маски	20
Оценка качества изображения	21
Извлечение биометрического шаблона	22
Верификация лиц	22
<b>Комбинирование сервисов</b>	<b>24</b>

## Глоссарий

Термин	Определение
Сэмпл	Набор входных/выходных данных обработки. Сэмпл с входными данными поступает в сервис обработки, результат обработки также формируется и передается в виде сэмпла. В Image API сэмпл для обработки содержит изображение в формате base64.
Атрибуты объекта (лица или силуэта)	Пол, возраст, эмоции, антропометрические точки, liveness, наличие медицинской маски.
Верификация лиц	Сравнение двух изображений лиц с целью определить их принадлежность одному и тому же человеку; сопоставление 1:1 («один к одному»).
Биометрический шаблон лица	Уникальный набор биометрических признаков лица, извлеченных из изображения лица. Шаблоны позволяют сравнить два изображения лица и определить степень их схожести.

# Обзор

Image API представляет собой набор программных средств, предназначенных для выполнения работ по распознаванию, анализу, идентификации и верификации лиц и силуэтов на основе входных изображений и сэмплов данных. В сервисы обработки системы загружаются изображения в формате jpg, png или bmp, либо сэмплы, содержащие изображения в формате base64. Результатом обработки является сэмпл с набором выходных атрибутов.

Система разработана для решения следующих функциональных задач:

- Детекция лиц и силуэтов;
- Определение антропометрических точек лица и углов наклона/поворота головы;
- Определение возраста, пола и эмоций человека;
- Определение наличия маски на лице;
- Проверка liveness;
- Оценка качества изображения;
- Верификация лиц;
- Построение биометрического шаблона.

## Архитектура

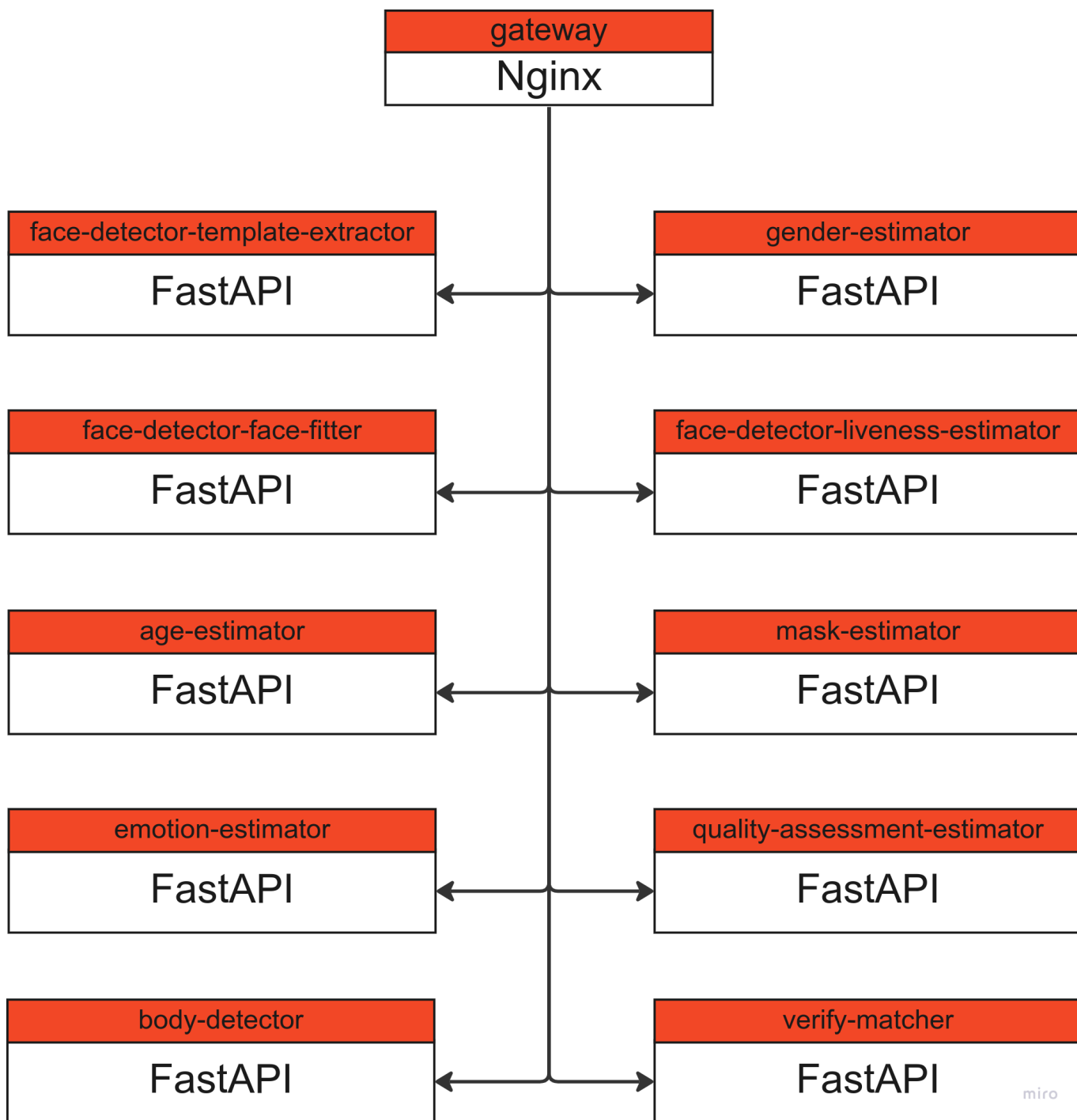


Image API состоит из следующих сервисов:

- **gateway**: сервис отвечает за доступ к Image API и взаимодействие с сервисами обработки, указанными ниже;

- **body-detector:** сервис предназначен для детекции силуэтов на изображении. Результатом детекции являются координаты ограничивающего прямоугольника (bbox) вокруг обнаруженного силуэта;
- **face-detector-face-fitter:** сервис используется для детекции лиц и определения антропометрических точек лица и углов наклона/поворота головы. Результатом работы компонента является набор из 21 антропометрической точки, а также значения углов наклона и поворота головы (yaw, pitch, roll);
- **emotion-estimator:** сервис используется для оценки эмоций человека по изображению лица. Результатом оценки является числовое значение степени проявления каждой оцениваемой эмоции;
- **age-estimator:** сервис используется для оценки возраста человека по изображению лица. Результатом оценки является числовое значение возраста человека;
- **gender-estimator:** сервис используется для оценки пола человека по изображению лица. Результатом оценки является определение принадлежности человеку к тому или иному полу;
- **face-detector-liveness-estimator:** сервис используется для детекции лиц и оценки принадлежности лица на изображении реальному человеку. Результатом является вердикт о принадлежности реальному человеку и числовое значение уверенности;
- **mask-estimator:** сервис позволяет определить наличие/отсутствие медицинской маски на лице человека с изображения. Результатом работы компонента является вердикт о наличии/отсутствии маски и числовое значение уверенности;
- **quality-assessment-estimator:** сервис предназначен для оценки качества изображения лица. Результатом является список обнаруженных лиц с подробным анализом качества;
- **verify-matcher:** сервис отвечает за сравнение двух лиц с изображений. Результатом сравнения является вердикт о принадлежности двух изображений одному и тому же человеку и числовое значение уверенности;
- **face-detector-template-extractor:** сервис предназначен для детекции лиц и извлечения биометрического шаблона лица.

Взаимодействие с сервисами осуществляется посредством REST API. В следующих разделах представлены структура и примеры API-запросов и ответов для решения основных функциональных задач Image API.

# Доступ к Image API

Для ознакомления с работой сервисов Image API предусмотрен специальный веб-интерфейс, расположенный по адресу `<image_api_url>`, откуда можно получить доступ к Swagger-документации по сервисам.

Для прямого доступа к Swagger-документации конкретного сервиса перейдите по ссылке `<image_api_url>/<service_name>/docs`, где `image_api_url` - адрес развернутого продукта, а `service_name` - имя сервиса.

Подробнее о работе со Swagger можно ознакомиться на сайте <https://swagger.io/>.

Список имен сервисов:

- face-detector-face-fitter
- face-detector-liveness-estimator
- face-detector-template-extractor
- age-estimator
- emotion-estimator
- gender-estimator
- mask-estimator
- quality-assessment-estimator
- body-detector
- verify-matcher

# Структура запроса

В веб-интерфейсе Swagger описаны следующие виды запросов: Process Sample и Process Image.

Processing		^
POST	/process/sample	Process Sample
POST	/process/image	Process Image

## Process Sample (Обработка сэмпла)

**Content-Type:** *application/json*

**Path:** *<image\_api\_url>/<service\_name>/process/sample*

В качестве тела запроса выступает сэмпл, содержащий:

- ***\$image*** - исходное изображение в формате base64
- ***objects*** - объекты обработки (лица и силуэты)

Для каждого объекта обработки предусмотрен определенный набор атрибутов, который отличается в зависимости от используемого сервиса. В ответ API возвращает атрибуты с вычисленными значениями.

**Примечание:** поля, отмеченные знаком "\$", например, *\$image*, определяют, что кодирование значения выполнено в формате base64.

## Process Image (Обработка изображения)

**Content-Type:** *multipart/form-data*

**Path:** *<image\_api\_url>/<service\_name>/process/image*

В качестве тела запроса выступает изображение в одном из форматов: jpg, png, bmp.

**Примечание:** данный интерфейс недоступен для сервисов *quality-assessment-estimator* и *verify-matcher*, т.к. для их работы требуются дополнительные атрибуты объектов в качестве входных данных. Доступность интерфейса конкретного сервиса можно проверить в веб-интерфейсе Swagger.



## Описание атрибутов

### objects:

- **angles:** углы наклона и поворота головы:
  - угол поворота (**yaw**): вращение вокруг вертикальной оси Y;
  - угол наклона (**pitch**): вращение вокруг горизонтальной оси Z;
  - угол отклонения (**roll**): вращение вокруг горизонтальной оси X.
- **age:** возраст;
- **bbox:** координаты ограничивающего прямоугольника вокруг лица. Рассчитываются относительно координат исходного изображения;
- **class:** имя класса объекта, например, лицо (face) или силуэт (body);
- **confidence:** числовое значение уверенности детекции;
- **emotions:**
  - **emotion:** 7 основных эмоций: гнев (angry), отвращение (disgusted), страх (scared), счастье (happy), нейтральное выражение лица (neutral), грусть (sad), удивление (surprised);
  - **confidence:** числовое значение уверенности оценки каждой эмоции. Диапазон значений: от 0 до 1.
- **fitter:**
  - **fitter\_type:** вид набора антропометрических точек. Набор *fda* обеспечивает высокое качество в широком диапазоне ракурсов (вплоть до профильных), при этом алгоритмы распознавания по-прежнему требуют, чтобы ракурс лица был максимально приближен к фронтальному. Набор *fda* содержит 21 точку;
  - **keypoints:** антропометрические точки лица;
  - **left\_eye:** координаты левого глаза;
  - **right\_eye:** координаты правого глаза;
- **gender:** пол;
- **id:** порядковый номер лица на изображении;
- **liveness:**
  - **confidence:** числовое значение уверенности принадлежности изображения реальному человеку. Диапазон значений: от 0 до 1;
  - **value:** вердикт: REAL - на изображении реальный человек, FAKE - изображение не принадлежит реальному человеку.

- **mask:**
  - **value:** вердикт: true - человек в маске, false - человек без маски;
  - **confidence:** числовое значение уверенности наличия/отсутствия маски на лице. Диапазон значений: от 0 до 1.
- **quality:**
  - **qaa** (Quality Assessment Algorithm) содержит следующий набор данных:
    - **totalScore:** числовое значение, обозначает общую оценку качества изображения в баллах от 0 до 100;
    - **isSharp:** логическое значение, обозначает резкость изображения;
    - **sharpnessScore:** числовое значение, обозначает оценку резкости в баллах от 0 до 100;
    - **isEvenlyIlluminated:** логическое значение, обозначает равномерность освещения на изображении;
    - **illuminationScore:** числовое значение, оценка равномерности освещения в баллах от 0 до 100;
    - **noFlare:** логическое значение, указывает на наличие или отсутствие вспышек на изображении;
    - **isLeftEyeOpened:** логическое значение, обозначает положение левого глаза (открыт/закрыт);
    - **leftEyeOpennessScore:** числовое значение, обозначает степень открытости глаза в баллах от 0 до 100;
    - **isRightEyeOpened:** логическое значение, обозначает положение правого глаза (открыт/закрыт);
    - **rightEyeOpennessScore:** числовое значение, обозначает степень открытости глаза в баллах от 0 до 100;
    - **isBackgroundUniform:** логическое значение, указывает на однородность фона;
    - **backgroundUniformityScore:** числовое значение, оценка однородности фона в баллах от 0 до 100;
    - **isDynamicRangeAcceptable:** логическое значение, показывает, что динамический диапазон интенсивности изображения в области лица превышает/не превышает значение 128;
    - **dynamicRangeScore:** числовое значение, оценка динамического диапазона интенсивности в баллах от 0 до 100;

- **isEyesDistanceAcceptable:** логическое значение, обозначает допустимое/недопустимое расстояние между глазами;
  - **eyesDistance:** числовое значение расстояния между глазами в px;
  - **isNotNoisy:** логическое значение, обозначает наличие/отсутствие шумов на изображении;
  - **noiseScore:** числовое значение, оценка зашумленности изображения в баллах от 0 до 100;
  - **isMarginsAcceptable:** логическое значение, обозначает допустимые/недопустимые отступы;
  - **marginInnerDeviation:** числовое значение внутреннего отклонения в px;
  - **marginOuterDeviation:** числовое значение внешнего отклонения в px;
  - **isNeutralEmotion:** логическое значение, обозначает наличие/отсутствие нейтрального выражения лица;
  - **neutralEmotionScore:** числовое значение, оценка степени нейтральных эмоций в баллах от 0 до 100;
  - **notMasked:** логическое значение, указывает на наличие/отсутствие маски на лице;
  - **notMaskedScore:** числовое значение, обозначает степень уверенности в отсутствии маски на лице от 0 до 100;
  - **hasWatermark:** логическое значение, указывает на наличие/отсутствие водяного знака на изображении;
  - **watermarkScore:** числовое значение, обозначает степень уверенности в наличии водяного знака на изображении от 0 до 100;
  - **isRotationAcceptable:** логическое значение, обозначает допустимый/недопустимый наклон или поворот головы;
  - **maxRotationDeviation:** числовое значение, максимальный градус отклонения для трех (yaw, pitch, roll) углов наклона и поворота головы.
- **\$template:** биометрический шаблон, закодированный в base64;
  - **template\_size:** размер биометрического шаблона (длина вектора).

### verification:

- **distance:** расстояние между векторами биометрических шаблонов;
- **fa\_r (False Acceptance Rate):** коэффициент ложной идентификации;
- **fr\_r (False Rejection Rate):** коэффициент ложного отклонения;
- **score:** параметр показывает степень схожести лиц от 0 (0%) до 1 (100%).

## Применение

### Детекция лиц

Детекция лиц реализована в следующих сервисах Image API:

- **face-detector-face-fitter**
- **face-detector-template-extractor**
- **face-detector-liveness-estimator**

### Детекция силуэтов

Запрос передается в сервис **body-detector**, который используется для детекции силуэтов.

В ответе возвращается следующий набор атрибутов с вычисленными значениями:

**objects:**

- **id**
- **class**
- **confidence**
- **bbox**

**Пример запроса:**

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {}
  ]
}
```

**Пример ответа:**

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {
      "id": 0,
      "class": "body",
      "confidence": 0.8266383409500122,
      "bbox": [
        0.648772656917572,
        0.13773296773433685,
        0.9848934412002563,
        0.8240703344345093
      ]
    },
    {
      "id": 1,
      "class": "body",
      "confidence": 0.7087612748146057,

```

```
    "bbox": [
      0.35164034366607666,
      0.15803256630897522,
      0.6833359003067017,
      0.8854727745056152
    ]
  }
]
```

## Определение пола

Запрос передается в сервис **gender-estimator**, который позволяет определить пол человека по изображению лица. В ответе возвращается следующий набор атрибутов с вычисленными значениями:

### objects:

- **id**
- **class**
- **gender**

### Пример запроса:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {}
  ]
}
```

### Пример ответа:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {
      "id": 0,
      "class": "face",
      "gender": "MALE"
    }
  ]
}
```

## Определение возраста

Запрос передается в сервис **age-estimator**, который используется для определения возраста человека по изображению лица. В ответе возвращается следующий набор атрибутов с вычисленными значениями:

### objects:

- **id**
- **class**
- **age**

### Пример запроса:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {}
  ]
}
```

### Пример ответа:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {
      "id": 0,
      "class": "face",
      "age": "25"
    }
  ]
}
```

## Оценка эмоций

Запрос передается в сервис **emotion-estimator**, который используется для оценки эмоций человека по изображению лица. В ответе возвращается следующий набор атрибутов с вычисленными значениями:

### objects:

- **id**
- **class**
- **emotions**
  - **emotion**
  - **confidence**

### Пример запроса:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {}
  ]
}
```

### Пример ответа:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {
      "id": 0,
      "class": "face",
      "emotions": [
        {
          "confidence": 0.10151619818478974,
          "emotion": "ANGRY"
        },
        {
          "confidence": 0.07763473911731263,
          "emotion": "DISGUSTED"
        },
        {
          "confidence": 0.20321173801223097,
          "emotion": "SCARED"
        },
        {
          "confidence": 0.08768639197580883,
          "emotion": "HAPPY"
        },
        {
          "confidence": 0.19000983487515088,
          "emotion": "NEUTRAL"
        },
        {
          "confidence": 0.08262699313446588,
          "emotion": "SAD"
        },
        {
          "confidence": 0.257314104700241,
          "emotion": "SURPRISED"
        }
      ]
    }
  ]
}
```

## Проверка liveness

Запрос передается в сервис **face-detector-liveness-estimator**, который используется для детекции лиц и оценки принадлежности лица на изображении реальному человеку. В ответе возвращается следующий набор атрибутов с вычисленными значениями:

### objects:

- **id**
- **class**
- **bbox**
- **confidence**
- **liveness:**

- **confidence**
- **value**

### Пример запроса:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {}
  ]
}
```

### Пример ответа:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {
      "id": 0,
      "class": "face",
      "confidence": 0.8233476281166077,
      "bbox": [
        0.375,
        0.12333333333333334,
        0.7645833333333333,
        0.42
      ],
      "liveness": {
        "confidence": 0.9989556074142456,
        "value": "REAL"
      }
    }
  ]
}
```

## Проверка на наличие медицинской маски

Запрос передается в сервис **mask-estimator**, который позволяет определить наличие/отсутствие медицинской маски на лице человека с изображения. В ответе возвращается следующий набор атрибутов с вычисленными значениями:

### objects:

- **id**
- **class**
- **mask:**
  - **value**
  - **confidence**

### Пример запроса:

```
{
  "$image": "изображение в формате base64",
  "objects": [
```



```
    {}  
  ]  
}
```

### Пример ответа:

```
{  
  "$image": "изображение в формате base64",  
  "objects": [  
    {  
      "id": 0,  
      "class": "face",  
      "mask": {  
        "confidence": 0.07230597734451294,  
        "value": false  
      }  
    }  
  ]  
}
```

## Определение антропометрических точек лица

Запрос передается в сервис **face-detector-face-fitter**, который используется для детекции лиц и определения антропометрических точек лица и углов наклона и поворота головы. В ответе возвращается следующий набор атрибутов с вычисленными значениями:

### objects:

- **id**
- **class**
- **bbox**
- **confidence**
- **fitter:**
  - **fitter\_type**
  - **keypoints**
  - **left\_eye**
  - **right\_eye**
- **angles:**
  - **yaw**
  - **pitch**
  - **roll**

### Пример запроса:

```
{  
  "$image": "изображение в формате base64",  
  "objects": [  
    {  
      "id": 0,  
      "class": "face",  
      "mask": {  
        "confidence": 0.07230597734451294,  
        "value": false  
      }  
    }  
  ]  
}
```

```
    {}  
  ]  
}
```

### Пример ответа:

```
{  
  "$image": "изображение в формате base64",  
  "objects": [  
    {  
      "id": 0,  
      "class": "face",  
      "confidence": 0.895225465297699,  
      "bbox": [  
        0.10445103857566766,  
        0.05966162065894924,  
        0.7008902077151336,  
        0.9243098842386465  
      ],  
      "fitter": {  
        "fitter_type": "fda",  
        "keypoints": [  
          344.24078369140625,  
          379.23858642578125,  
          0,  
          443.0493469238281,  
          364.8091125488281,  
          0,  
          547.5462646484375,  
          384.35833740234375,  
          0,  
          724.5175170898438,  
          385.01220703125,  
          0,  
          816.4994506835938,  
          366.4952697753906,  
          0,  
          899.7161865234375,  
          380.7967224121094,  
          0,  
          391.20654296875,  
          461.12066650390625,  
          0,  
          461.524169921875,  
          459.44287109375,  
          0,  
          531.2512817382812,  
          467.7398681640625,  
          0,  
          721.8792724609375,  
          468.227294921875,  
          0,  
          784.9144897460938,  
          461.4508056640625,  
          0,  
          854.609130859375,  
          465.002685546875,  
          0,  
          250.21035766601562,  
          657.1244506835938,  
        ]  
      }  
    }  
  ]  
}
```

```
    0,
    559.1598510742188,
    666.7738647460938,
    0,
    641.8836059570312,
    678.353515625,
    0,
    710.0083618164062,
    670.3438110351562,
    0,
    939.4479370117188,
    656.3207397460938,
    0,
    509.8494873046875,
    816.0798950195312,
    0,
    634.861083984375,
    823.5408325195312,
    0,
    748.5276489257812,
    813.4531860351562,
    0,
    633.5501098632812,
    1033.822509765625,
    0
  ],
  "left_eye": [
    461.524169921875,
    459.44287109375
  ],
  "right_eye": [
    784.9144897460938,
    461.4508056640625
  ]
},
"angles": {
  "yaw": 6.648662090301514,
  "roll": 0.3107689917087555,
  "pitch": -23.410654067993164
}
}
]
```

## Оценка качества изображения

Запрос передается в сервис **quality-assessment-estimator**, который предназначен для оценки качества изображения лица. В теле запроса необходимо передать значения атрибутов лица, полученных после обработки изображения сервисом **face-detector-face-fitter**. В ответе возвращается следующий набор атрибутов с вычисленными значениями:

### objects:

- id

- **class**
- **confidence**
- **bbox**
- **quality:**
  - **qaa:**
    - **totalScore**
    - **isSharp**
    - **sharpnessScore**
    - **isEvenlyIlluminated**
    - **illuminationScore**
    - **noFlare**
    - **isLeftEyeOpened**
    - **leftEyeOpennessScore**
    - **isRightEyeOpened**
    - **rightEyeOpennessScore**
    - **isBackgroundUniform**
    - **backgroundUniformityScore**
    - **isDynamicRangeAcceptable**
    - **dynamicRangeScore**
    - **isEyesDistanceAcceptable**
    - **eyesDistance**
    - **isNotNoisy**
    - **noiseScore**
    - **isMarginsAcceptable**
    - **marginInnerDeviation**
    - **marginOuterDeviation**
    - **isNeutralEmotion**
    - **neutralEmotionScore**
    - **notMasked**
    - **notMaskedScore**
    - **hasWatermark**
    - **watermarkScore**
    - **isRotationAcceptable**
    - **maxRotationDeviation**

### Пример запроса:

```
{
  "$image": "string",
  "objects": [
    {
      "id": 0,
      "class": "face",
      "confidence": 0.895225465297699,
      "bbox": [
        0.10445103857566766,
        0.05966162065894924,
        0.7008902077151336,
        0.9243098842386465
      ],
      "fitter": {
        "fitter_type": "fda",
        "keypoints": [
          344.24078369140625,
          379.23858642578125,
          0,
          443.0493469238281,
          364.8091125488281,
          0,
          547.5462646484375,
          384.35833740234375,
          0,
          724.5175170898438,
          385.01220703125,
          0,
          816.4994506835938,
          366.4952697753906,
          0,
          899.7161865234375,
          380.7967224121094,
          0,
          391.20654296875,
          461.12066650390625,
          0,
          461.524169921875,
          459.44287109375,
          0,
          531.2512817382812,
          467.7398681640625,
          0,
          721.8792724609375,
          468.227294921875,
          0,
          784.9144897460938,
          461.4508056640625,
          0,
          854.609130859375,
          465.002685546875,
          0,
          250.21035766601562,
          657.1244506835938,
          0,
          559.1598510742188,
          666.7738647460938,
          0,
          641.8836059570312,
```

```
        678.353515625,  
        0,  
        710.0083618164062,  
        670.3438110351562,  
        0,  
        939.4479370117188,  
        656.3207397460938,  
        0,  
        509.8494873046875,  
        816.0798950195312,  
        0,  
        634.861083984375,  
        823.5408325195312,  
        0,  
        748.5276489257812,  
        813.4531860351562,  
        0,  
        633.5501098632812,  
        1033.822509765625,  
        0  
    ],  
    "left_eye": [  
        461.524169921875,  
        459.44287109375  
    ],  
    "right_eye": [  
        784.9144897460938,  
        461.4508056640625  
    ]  
},  
"angles": {  
    "yaw": 6.648662090301514,  
    "roll": 0.3107689917087555,  
    "pitch": -23.410654067993164  
}  
}  
]
```

### Пример ответа:

```
{  
  "$image": "string",  
  "objects": [  
    {  
      "id": 0,  
      "class": "face",  
      "confidence": 0.69044026635,  
      "bbox": [  
        0.42242398858070374,  
        0.05838850140571594,  
        0.5360375642776489,  
        0.17216356098651886  
      ],  
      "quality": {  
        "qaa": {  
          "totalScore": 0,  
          "isSharp": true,  
          "sharpnessScore": 0,  
          "isEvenlyIlluminated": true,  
        }  
      }  
    }  
  ]  
}
```

```
"illuminationScore": 0,
"noFlare": true,
"isLeftEyeOpened": true,
"leftEyeOpennessScore": 0,
"isRightEyeOpened": true,
"rightEyeOpennessScore": 0,
"isRotationAcceptable": true,
"maxRotationDeviation": 0,
"notMasked": true,
"notMaskedScore": 0,
"isNeutralEmotion": true,
"neutralEmotionScore": 0,
"isEyesDistanceAcceptable": true,
"eyesDistance": 0,
"isMarginsAcceptable": true,
"marginOuterDeviation": 0,
"marginInnerDeviation": 0,
"isNotNoisy": true,
"noiseScore": 0,
"watermarkScore": 0,
"hasWatermark": true,
"dynamicRangeScore": 0,
"isDynamicRangeAcceptable": true,
"backgroundUniformityScore": 0,
"isBackgroundUniform": true
  }
}
]
}
```

## Извлечение биометрического шаблона

Запрос передается в сервис **face-detector-template-extractor**, который детектирует лица на изображении и формирует на их основе биометрические шаблоны. В ответе возвращается следующий набор атрибутов с вычисленными значениями:

**objects:**

- **id**
- **class**
- **confidence**
- **bbox**
- **\$template**
- **template\_size**

**Пример запроса:**

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {}
  ]
}
```

```
}
```

### Пример ответа:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {
      "id": 0,
      "class": "face",
      "confidence": 0.895225465297699,
      "bbox": [
        0.10445103857566766,
        0.05966162065894924,
        0.7008902077151336,
        0.9243098842386465
      ],
      "$template": "шаблон в формате base64",
      "template_size": 74
    }
  ]
}
```

## Верификация лиц

Запрос передается в сервис **verify-matcher**, который верифицирует лица на основе сравнения их биометрических шаблонов. В теле запроса необходимо передать значения атрибутов лица, полученных после обработки изображений сервисом **face-detector-template-extractor**. В ответе возвращается следующий набор атрибутов с вычисленными значениями:

### verification:

- **distance**
- **fa\_r**
- **fr\_r**
- **score**

### Пример запроса:

```
{
  "objects": [
    { "id": 0,
      "class": "face",
      "confidence": 0.9171707630157471,
      "bbox": [
        0.14427860696517414,
```



```
0.21912350597609562,  
0.8656716417910447,  
0.796812749003984  
],  
"$template": "шаблон в формате base64",  
"template_size": 74},  
{  
  "id": 0,  
  "class": "face",  
  "confidence": 0.8453116416931152,  
  "bbox": [  
    0.16477272727272727,  
    0.22272727272727272,  
    0.875,  
    0.7954545454545454  
  ],  
  "$template": "шаблон в формате base64",  
  "template_size": 74}  
]  
}
```

### Пример ответа:

```
{  
  "objects": [  
    {  
      "bbox": [  
        0.14427860696517414,  
        0.21912350597609562,  
        0.8656716417910447,  
        0.796812749003984  
      ],  
      "$template": "шаблон в формате base64",  
      "id": 0,  
      "class": "face",  
      "confidence": 0.9171707630157471,  
      "template_size": 74  
    },  
    {  
      "bbox": [  
        0.16477272727272727,  
        0.22272727272727272,  
        0.875,  
        0.7954545454545454  
      ],  
      "$template": "шаблон в формате base64",  
      "id": 0,  
      "class": "face",  
      "confidence": 0.8453116416931152,  
      "template_size": 74  
    }  
  ]  
}
```

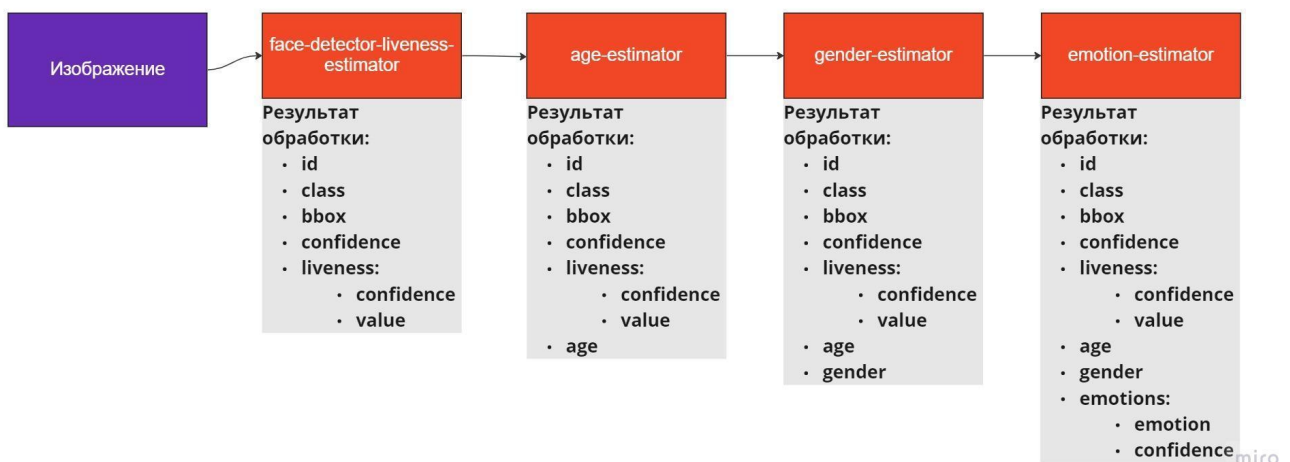
```
    "$template": "шаблон в формате base64",
    "id": 0,
    "class": "face",
    "confidence": 0.8453116416931152,
    "template_size": 74
  }
],
"verification": {
  "distance": 4796,
  "fa_r": 0,
  "fr_r": 0.522820770740509,
  "score": 0.9515298008918762
}
}
```

## Комбинирование сервисов

В Image API реализована возможность передачи результата обработки из одного сервиса в качестве входных данных для другого сервиса. Таким образом, можно получать обогащенные данные, содержащие композицию результатов, и выстраивать цепочки сервисов в зависимости от ваших задач.

Первым сервисом в цепочке должен идти сервис со встроенной функцией детекции лица. Такой сервис обрабатывает изображение, после чего можно передать результат детекции в следующий сервис цепочки в качестве входных данных. Функция детекции лица реализована в сервисах **face-detector-face-fitter**, **face-detector-template-extractor** и **face-detector-liveness-estimator**.

Например, для решения задачи по определению liveness, возраста, пола и эмоций человека по изображению, можно построить цепочку из следующих сервисов:



В сервис **face-detector-liveness-estimator** загружается изображение. Далее результат обработки (детекция лица и результат проверки liveness) передается на вход в сервис **age-estimator**, где к результату дописывается значение возраста. Затем полученные обогащенные данные поступают в сервис **gender-estimator**, где определяется пол человека, а оттуда - в **emotion-estimator**, где добавляется оценка эмоций.

**Пример запроса к сервису face-detector-liveness-estimator:**

```
{
  "$image": "изображение в формате base64",
  "objects": [
```

```
    {}  
  ]  
}
```

### Пример ответа:

```
{  
  "$image": "изображение в формате base64",  
  "objects": [  
    {  
      "id": 0,  
      "class": "face",  
      "confidence": 0.8233476281166077,  
      "bbox": [  
        0.375,  
        0.12333333333333334,  
        0.7645833333333333,  
        0.42  
      ],  
      "liveness": {  
        "confidence": 0.9989556074142456,  
        "value": "REAL"  
      }  
    }  
  ]  
}
```

### Пример запроса к сервису age-estimator:

```
{  
  "$image": "изображение в формате base64",  
  "objects": [  
    {  
      "id": 0,  
      "class": "face",  
      "confidence": 0.8233476281166077,  
      "bbox": [  
        0.375,  
        0.12333333333333334,  
        0.7645833333333333,  
        0.42  
      ],  
      "liveness": {  
        "confidence": 0.9989556074142456,  
        "value": "REAL"  
      }  
    }  
  ]  
}
```

### Пример ответа:

```
{  
  "$image": "изображение в формате base64",  
  "objects": [  
    {  
      "id": 0,  
      "class": "face",  
      "age": 21,  
      "liveness": {  
        "confidence": 0.9989556074142456,  
        "value": "REAL"  
      }  
    }  
  ]  
}
```

```
    "confidence": 0.9989556074142456,
    "value": "REAL"
  },
  "confidence": 0.8233476281166077,
  "bbox": [
    0.375,
    0.12333333333333334,
    0.7645833333333333,
    0.42
  ]
}
]
```

### Пример запроса к сервису gender-estimator:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {
      "id": 0,
      "class": "face",
      "age": 21,
      "liveness": {
        "confidence": 0.9989556074142456,
        "value": "REAL"
      }
    },
    "confidence": 0.8233476281166077,
    "bbox": [
      0.375,
      0.12333333333333334,
      0.7645833333333333,
      0.42
    ]
  ]
}
```

### Пример ответа:

```
{
  "$image": "изображение в формате base64",
  "objects": [
    {
      "id": 0,
      "class": "face",
      "gender": "FEMALE",
      "liveness": {
        "confidence": 0.9989556074142456,
        "value": "REAL"
      }
    },
    "bbox": [
      0.375,
      0.12333333333333334,
      0.7645833333333333,
      0.42
    ],
    "age": 21,
    "confidence": 0.8233476281166077
  ]
}
```

```
}  
]  
}
```

### Пример запроса к сервису emotion-estimator:

```
{  
  "$image": "изображение в формате base64",  
  "objects": [  
    {"id": 0,  
     "class": "face",  
     "gender": "FEMALE",  
     "liveness": {  
       "confidence": 0.9989556074142456,  
       "value": "REAL"  
     }  
    },  
    "bbox": [  
      0.375,  
      0.12333333333333334,  
      0.7645833333333333,  
      0.42  
    ],  
    "age": 21,  
    "confidence": 0.8233476281166077  
  ]  
}
```

### Пример ответа:

```
{  
  "$image": "изображение в формате base64",  
  "objects": [  
    {  
      "id": 0,  
      "class": "face",  
      "emotions": [  
        {  
          "confidence": 0.00046337815752155756,  
          "emotion": "ANGRY"  
        },  
        {  
          "confidence": 0.000010690175584454014,  
          "emotion": "DISGUSTED"  
        },  
        {  
          "confidence": 0.0000651997511831193,  
          "emotion": "SCARED"  
        },  
        {  
          "confidence": 0.050197473080296776,  
          "emotion": "HAPPY"  
        },  
        {  
          "confidence": 0.6730428226960512,  
          "emotion": "NEUTRAL"  
        },  
        {  
          "confidence": 0.274063680489088,  
          "emotion": "SURPRISED"  
        }  
      ]  
    }  
  ]  
}
```

```
    "emotion": "SAD"
  },
  {
    "confidence": 0.0021567556502748936,
    "emotion": "SURPRISED"
  }
],
"bbox": [
  0.375,
  0.12333333333333334,
  0.7645833333333333,
  0.42
],
"confidence": 0.8233476281166077,
"age": 21,
"liveness": {
  "confidence": 0.9989556074142456,
  "value": "REAL"
},
"gender": "FEMALE"
}
]
```