



VisionLabs
MACHINES CAN SEE

VISIONLABS ACCESS CONTROL SERVER

Инструкция по эксплуатации

ООО «ВижнЛабс»

123458, г. Москва, ул. Твардовского д. 8, стр. 1

☎ +7 (499) 399 3361

✉ info@visionlabs.ru

🌐 www.visionlabs.ru

Содержание

Введение	3
1. Создание аккаунта	4
2. Создание обработчика	5
2.1. Выполнение запроса	6
2.2. Результат выполнения запроса	6
3. Генерация события и обработка handler	8
3.1. Описание запроса	8
3.2. Выполнение запроса	8
3.3. Результат выполнения запроса	8

Введение

В данном документе приводится пример создания обработчика, используя CURL-запрос. С помощью данного обработчика возможно:

- детектировать лицо;
- определять базовые атрибуты (пол, возраст и этническую принадлежность);
- извлекать дескриптор;
- сохранять биометрический образец на диск (bucket);
- сохранять лицо в базу данных Faces.

Также в инструкции приведен пример генерации события, используя изображение, которое будет сравниваться с изображением из списка.

1. Создание аккаунта

Первым этапом необходимо создать аккаунт. Аккаунт создается с помощью запроса "create account".

Обработчик можно создать с помощью двух типов авторизации - BasicAuth (логин и пароль), BearerAuth (токен). В данном примере будет использоваться авторизация по логину и паролю.

Для использования авторизации BasicAuth в curl-запросе, требуется конвертировать логин и пароль в формат Base64. Конвертировать логин и пароль можно разными способами, например, с помощью ресурса <https://www.base64encode.org/>. Для конвертации требуется указать логин и пароль в формате login:password. В данном случае - the_best_user@visionlabs.ru:the_best_password. Закодированные в Base64 логин и пароль выглядят следующим образом:

```
bi5wcm9za3VyaW5Adm1zaW9ubGFicy5haTpyb290
```

2. Создание обработчика

Назначение: Запрос позволяет создать обработчик в Access Control Server, который будет использоваться в дальнейшем для создания событий.

Тип запроса: POST

Запрос: 127.0.0.1:5000/6/handlers

Запрос выполняется с параметрами, указанными в Таблице 1.

Таблица 1. Параметры запроса для создания обработчика

Название параметра	Описание	Значение
Заголовки запроса		
Content-Type	Тип содержимого в запросе	application/json
Authorization	Basic-авторизация. Логин и пароль в формате Base64.	dGhIX2Jlc3RfdXNlckB2aXNpb25sYWJzLnJ1OnRoZV9iZXN0X3Bhc3N3b3Jk
Тело запроса		
	<p>Строка для ввода данных.</p> <p>Необходимо указать:</p> <ul style="list-style-type: none"> • description - описание обработчика • policies - политики обработчика <ul style="list-style-type: none"> ○ detect_policy > detect_face - параметр определения лица ○ extract_policy > extract_basic_attributes - параметр извлечения базовых атрибутов (пол, возраст, этническая принадлежность) ○ extract_policy > extract_face_descriptor - параметр извлечения дескриптора ○ storage_policy > face_policy > store_face - параметр сохранения лица в БД Faces • handler_type - тип обработчика (0 - 	См. ниже

Статический	статический, 1 - динамический)
-------------	-----------------------------------

Примечание. Обработчик можно создать без указания политик, однако это не значит, что обработчик будет создан без них. Некоторые параметры политик включены по умолчанию, поэтому даже без указания политик будут оцениваться некоторые параметры и сохраняться некоторые данные.

2.1. Выполнение запроса

Пример cURL-запроса для выполнения запроса из командной строки:

```
curl --location --request POST 'http://127.0.0.1:5000/6/handlers' \
--header 'Authorization: Basic bi5wcm9za3Vyaw5AdmlzaW9ubGFicy5haTpyb290' \
--header 'Content-Type: application/json' \
--data '{
  "description": " face detection, descriptor extraction, saving a face in the Faces
database ",
  "policies": {
    "detect_policy": {
      "detect_face": 1
    },
    "extract_policy": {
      "extract_basic_attributes": 1,
      "extract_face_descriptor": 1
    },
    "storage_policy": {
      "face_sample_policy": {
        "store_sample": 1
      },
      "face_policy": {
        "store_face": 1
      }
    }
  },
  "handler_type": 0
}'
```

2.2. Результат выполнения запроса

При успешном выполнении запроса система возвращает ID и URL обработчика.

Пример ответа на запрос:

```
{"handler_id": "3f0ae6fe-85e6-4960-b980-bbc36fb41857", "url": "\/6\/handlers\/3f0ae6fe-85e6-4960-b980-bbc36fb41857", "external_url": "http://127.0.0.1:5000\/6\/handlers\/3f0ae6fe-85e6-4960-b980-bbc36fb41857"}
```

```
{"handler_id": "3f0ae6fe-85e6-4960-b980-bbc36fb41857", "url": "\/6\/handLers\/3f0ae6fe-85e6-4960-b980-bbc36fb41857", "external_url": "http://127.0.0.1:5000\/6\/handLers\/3f0ae6fe-85e6-4960-b980-bbc36fb41857"} [root@techwrite ~]#
```

Примечание. Полученный "handler_id" необходимо сохранить для последующего использования.

3. Генерация события и обработка handler

По созданному обработчику теперь необходимо сгенерировать событие, используя изображение, которое будет сравниваться с изображением из списка.

3.1. Описание запроса

Назначение: Запрос позволяет сгенерировать события и обработать их соответствующим обработчиком.

Тип запроса: POST

Запрос: 127.0.0.1:5000/6/handlers/{handler_id}/events

Запрос выполняется с параметрами, указанными в Таблице 6.

Таблица 6. Параметры запроса для генерации события

Название параметра (key)	Описание	Значение
Заголовки запроса		
Luna-Account-Id	Идентификатор аккаунта	6d071cca-fda5-4a03-84d5-5bea65904480
Content-Type	Тип содержимого в запросе	image/jpeg
Тело запроса		
	Путь до изображения. Изображение заранее загружено на тестовый стенд VisionLabs.	(см. ниже)

3.2. Выполнение запроса

Пример cURL-запроса для выполнения запроса из командной строки:

```
curl --location --request POST 'http:// 127.0.0.1:5000/6/handlers/b3413af3-31c4-4a62-bf0e-1aedaf784d9c/events' \
--header 'Luna-Account-Id: 6d071cca-fda5-4a03-84d5-5bea65904480' \
--header 'Content-Type: image/jpeg' \
--data-binary '@/home/vlabs/Image1.jpg'
```

3.3. Результат выполнения запроса

При успешном выполнении запроса система возвращает поля, определение которых было задано в обработчике.

Примеры ответа на запрос по наиболее интересующим полям представлены ниже.

Поле "events" содержит в себе все данные созданного события.

Поле "events" > "face_attributes". Поскольку в обработчике, созданном в п.2 было задано извлечение базовых атрибутов, то в событии отображаются все базовые атрибуты загруженного изображения.

```
...
"face_attributes": {
  "samples": [
    "7d011620-a778-4026-8d7c-a60d8719b62e"
  ],
  "attribute_id": null,
  "url": null,
  "basic_attributes": {
    "ethnicities": {
      "predominant_ethnicity": "caucasian",
      "estimations": {
        "asian": 3.3201965834450675e-6,
        "indian": 2.709296950342832e-6,
        "caucasian": 0.9999136924743652,
        "african_american": 8.021500252652913e-5
      }
    },
    "age": 21,
    "gender": 0
  },
  "score": 0.9876164793968201
},
...
```

Поле "events" > "match". Поскольку в обработчике были заданы правила сравнения загружаемого лица с лицом из списка, то в поле "match" > "candidates" будут отображаться все кандидаты для сравнения (в данном случае только один кандидат). Параметр "similarity" означает степень схожести загружаемого изображения с изображением из списка от 0 до 1. В данном случае "similarity" близится к 1, поскольку было загружено тоже самое лицо.

```
...
"matches": [
  {
    "candidates": [
      {
        "face": {
          "face_id": "260023a3-9e7b-42a7-8e94-223278c83600",
          "account_id": "6d071cca-fda5-4a03-84d5-5bea65904480",
          "event_id": null,
          "create_time": "2021-11-02T11:00:38.722556+03:00",
          "external_id": "",
          "avatar": "",
          "lists": [
            "fecfaefb-7cf2-490f-9f89-006344535d76"
          ]
        },
        "similarity": 0.9999998808
      }
    ],
    "label": ""
  }
]
```

```
],
```

```
...
```

Поле "attributes" > "liveness". Поле содержит данные Liveness загружаемой фотографии. Параметр "prediction" может принимать три значения - "real", "spoof", "unknow". Для загружаемого изображения поле имеет значение "real", что означает, что лицо прошло проверку на Liveness.

```
...
```

```
"attributes": {  
  "liveness": {  
    "prediction": "real",  
    "estimations": {  
      "quality": 0.9958741664886475,  
      "score": 0.9993811249732971  
    }  
  }  
}
```

```
...
```