



**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](mailto:info@visionlabs.ru)

🌐 [www.visionlabs.ru](http://www.visionlabs.ru)

## Содержание

Глоссарий.....	3
Введение .....	4
1. Эксплуатация Системы .....	5
1.1. Создание списка .....	5
1.1.1. Описание запроса .....	5
1.1.2. Выполнение запроса.....	5
1.1.3. Результат выполнения запроса.....	6
1.2. Загрузка фотоизображений и создание биометрического образца .....	6
1.2.1. Описание запроса .....	6
1.2.2. Выполнение запроса.....	7
1.2.3. Результат выполнения запроса.....	7
1.3. Извлечение биометрического шаблона и базовых атрибутов .....	7
1.3.1. Описание запроса .....	8
1.3.2. Выполнение запроса.....	8
1.3.3. Результат выполнения запроса.....	8
1.4. Создание лица и прикрепление лица к списку .....	9
1.4.1. Описание запроса .....	9
1.4.2. Выполнение запроса.....	9
1.4.3. Результат выполнения запроса.....	10
1.5. Создание обработчика .....	10
1.5.1. Описание запроса .....	10
1.5.2. Выполнение запроса.....	11
1.5.3. Результат выполнения запроса.....	11
1.6. Генерация события и обработка handler .....	11
1.6.1. Описание запроса .....	12
1.6.2. Выполнение запроса.....	12
1.6.3. Результат выполнения запроса.....	12

## Глоссарий

<b>Термин</b>	<b>Определение</b>
Базовые атрибуты	Возраст, пол, этническая принадлежность.
БД	База данных
Биометрический образец, БО	Фотоизображение, приведенное в формат, который соответствует требованиям Системы.
Биометрический шаблон, БШ	Набор данных в закрытом двоичном формате, подготавливаемый системой на основе анализируемой характеристики. Представляет из себя составной вектор признаков фотоизображения лица человека.
Лица	Изменяемые объекты, содержащие информацию об одном лице.
Параметры лиц	Эмоции, параметры рта, положение головы и т.д.

## **Введение**

В документе описывается процесс эксплуатации VisionLabs Access Control Server (далее – Система). Для эксплуатации Системы необходимо подключение к тестовому стенду VisionLabs (см. документ «Инструкция для проведения проверки ПО»).

Дополнительная документация по Системе может быть запрошена у компании VisionLabs.

## 1. Эксплуатация Системы

Заказчик последовательно отправляет HTTP-запросы к сервисам Access Control Server через сервис API и получает в ответ пакет данных. Для некоторых запросов предварительно требуется выполнить запрос, чтобы получить сущность, идентификатор которой будет использоваться в следующем запросе.

Система генерирует уникальные идентификаторы, которые отличаются от тех, что приведены в документации. Для корректной эксплуатации необходимо заменять идентификаторы на те, что были получены заказчиком в результате выполнения предыдущего запроса.

В данном документе приведена следующая последовательность действий:

- создание списка, к которому будет прикреплено фотоизображение;
- загрузка фотоизображения в Access Control Server и создание биометрического образца;
- извлечение биометрического шаблона и базовых атрибутов;
- создание объекта «лицо» и прикрепление его к списку;
- создание обработчика, в котором должны содержаться правила по сравнению загружаемого лица с лицом из списка;
- обработка ранее созданного обработчика и генерация события.

### 1.1. Создание списка

#### 1.1.1. Описание запроса

Назначение	Тип запроса	Запрос
Запрос позволяет создать список в Access Control Server. К такому списку могут быть прикреплены лица	POST	10.16.30.82:5000/6/lists

Запрос выполняется с параметрами, указанными в Таблице 1.

**Таблица 1.** Параметры запроса для создания списка

Параметр (key)	Описание	Значение
<b>Headers</b>		
Luna-Account-Id	Идентификатор аккаунта	6d071cca-fda5-4a03-84d5-5bea65904480
Content-Type	Тип содержимого в запросе	application/json
<b>Body</b>		
	user_data - название списка	(см. ниже)

#### 1.1.2. Выполнение запроса

Пример cURL-запроса для выполнения запроса из командной строки:

```
curl --location --request POST 'http://10.16.30.82:5000/6/lists' \
--header 'Luna-Account-Id: 6d071cca-fda5-4a03-84d5-5bea65904480' \
--header 'Content-Type: application/json' \
--data '{
  "user_data": "List"
}'
```

### 1.1.3. Результат выполнения запроса

При успешном выполнении запроса Система возвращает ID и URL списка.

Пример ответа на запрос:

```
{"list_id": "fecfaefb-7cf2-490f-9f89-006344535d76", "url": "\/6\/lists\/fecfaefb-7cf2-490f-9f89-006344535d76"}
```

Полученный "list\_id" необходимо сохранить для последующего использования.

## 1.2. Загрузка фотоизображений и создание биометрического образца

После того, как список создан, необходимо загрузить лицо в Access Control Server и создать БО. В качестве загружаемого лица будет использоваться лицо, показанное на рисунке ниже. Данное изображение уже загружено на тестовый стенд VisionLabs.

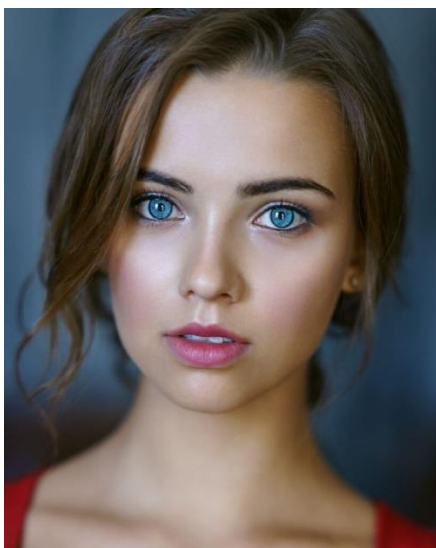


Рисунок 1. Загружаемое лицо

### 1.2.1. Описание запроса

Назначение	Тип запроса	Запрос
Запрос позволяет загрузить изображение в Access Control Server, детектировать лица на изображении, извлечь свойства и создать БО для каждого детектированного лица	POST	10.16.30.82:5000/6/detector

Запрос выполняется с параметрами, указанными в Таблице 2.

Таблица 2. Параметры запроса на создание БО

Параметр (key)	Описание	Значение
<b>Headers</b>		
Luna-Account-Id	Идентификатор аккаунта	6d071cca-fda5-4a03-84d5-5bea65904480
Content-Type	Тип содержимого в запросе	image/jpeg
<b>Body</b>		
	Путь до изображения. Изображение (см. ниже) ранее загружено на тестовый стенд VisionLabs.	

### 1.2.2. Выполнение запроса

Пример cURL-запроса для выполнения запроса из командной строки:

```
curl --location --request POST 'http://10.16.30.82:5000/6/detector' \
--header 'Luna-Account-Id: 6d071cca-fda5-4a03-84d5-5bea65904480' \
--header 'Content-Type: image/jpeg' \
--data-binary '@/root/Image1.jpg'
```

### 1.2.3. Результат выполнения запроса

При успешном выполнении запроса Система возвращает основные параметры обнаружения и идентификатор БО "sample\_id".

Пример ответа на запрос:

```
{"images":[{"filename":"raw
image","error":{"error_code":0,"desc":"Success","detail":"Success"},"status":1,"detections
":{"samples":[{"face":{"detection":{"rect":{"x":272,"y":332,"width":854,"height":1067},"la
ndmarks5":[[282,405],[681,439],[478,736],[286,825],[582,845]]},"url":"\\/6\\/samples\\/faces\\
/18a1b0c9-4358-4ac2-b919-25d33a448c0b","sample_id":"18a1b0c9-4358-4ac2-b919-
25d33a448c0b"}]}],"filtered_detections":{"face_detections":[]}]}}
```

Полученный "sample\_id" необходимо сохранить для последующего использования.

## 1.3. Извлечение биометрического шаблона и базовых атрибутов

Для последующей работы необходимо выполнить запрос на извлечение и сохранение в базу данных БШ и базовых атрибутов загруженной фотографии.

Все извлекаемые параметры временны и удаляются из БД по истечении заданного времени (TTL).

### 1.3.1. Описание запроса

Назначение	Тип запроса	Запрос
Запрос позволяет извлекать из образцов БШ и базовые атрибуты	POST	10.16.30.82:5000/6/extractor

Запрос выполняется с параметрами, указанными в Таблице 3.

**Таблица 3.** Параметры запроса для извлечения БШ и атрибутов

Параметр (key)	Описание	Значение
<b>Headers</b>		
Luna-Account-Id	Идентификатор аккаунта	6d071cca-fda5-4a03-84d5-5bea65904480
Content-Type	Тип содержимого в запросе	image/jpeg
<b>Params</b>		
extract_descriptor	Извлекать ли БШ	1
extract_basic_attributes	Извлекать ли базовые атрибуты (пол, возраст, расовая принадлежность)	1
ttl	Время существования временных атрибутов в секундах	3000
<b>Body</b>		
	Идентификатор БО, сохраненный в <a href="#">п.1.2.3</a> из которого необходимо извлечь БШ и базовые атрибуты	(см. ниже)

### 1.3.2. Выполнение запроса

Пример cURL-запроса для выполнения запроса из командной строки:

```
curl --location --request POST
'http://10.16.30.82:5000/6/extractor?extract_descriptor=1&extract_basic_attributes=1&ttl=3000' \
  --header 'Luna-Account-Id: 6d071cca-fda5-4a03-84d5-5bea65904480' \
  --header 'Content-Type: application/json' \
  --data '[
    "0eb491c0-b2f8-4c05-af0f-fc8183ca8e76"
  ]'
```

### 1.3.3. Результат выполнения запроса

При успешном выполнении запроса Система возвращает ID и URL атрибутов, а также базовые атрибуты для каждого изображения.

Пример ответа при успешном выполнении запроса:



```
[{"samples":["18a1b0c9-4358-4ac2-b919-25d33a448c0b"],"attribute_id":"53af8fb4-4149-41b4-b3e9-250c24beb63c","url":"\\6\\attributes\\53af8fb4-4149-41b4-b3e9-250c24beb63c","basic_attributes":{"ethnicities":{"predominant_ethnicity":"caucasian","estimations":{"asian":2.193369937231182e-6,"indian":2.1315634057827992e-6,"caucasian":0.9999216794967651,"african_american":7.401195034617558e-5}},"age":20,"gender":0},"score":0.9902883768081665}]
```

Полученный "attribute\_id" необходимо сохранить для последующего использования. Временные атрибуты будут удалены через 3000 секунд и может потребоваться заново выполнить запрос на извлечение атрибутов.

## 1.4. Создание лица и прикрепление лица к списку

Далее необходимо выполнить запрос, создающий объект "лицо" и прикрепляющий его к списку.

### 1.4.1. Описание запроса

Назначение	Тип запроса	Запрос
Запрос позволяет создать лицо, прикрепить к нему БШ и прикрепить лицо к списку	POST	10.16.30.82:5000/6/faces

Запрос выполняется с параметрами, указанными в Таблице 4.

**Таблица 4.** Параметры запроса для создания лица и прикрепления лица к списку

Параметр (key)	Описание	Значение
<b>Заголовки запроса</b>		
Luna-Account-Id	Идентификатор аккаунта	6d071cca-fda5-4a03-84d5-5bea65904480
Content-Type	Тип содержимого в запросе	image/jpeg

#### Тело запроса

attribute\_id – идентификатор атрибута, (см. ниже) полученный в [п.1.3.3](#);

lists – идентификатор списка, полученный в [п.1.1.3](#), к которому необходимо прикрепить лицо

### 1.4.2. Выполнение запроса

Пример cURL-запроса для выполнения запроса из командной строки:

```
curl --location --request POST 'http://10.16.30.82:5000/6/faces' \
--header 'Luna-Account-Id: 6d071cca-fda5-4a03-84d5-5bea65904480' \
--header 'Content-Type: application/json' \
--data '{ "attribute": { "attribute_id": "53af8fb4-4149-41b4-b3e9-250c24beb63c" }, "lists": [ "fecfaefb-7cf2-490f-9f89-006344535d76" ]}'
```

### 1.4.3. Результат выполнения запроса

При успешном выполнении запроса Система возвращает ID и URL лица.

Пример ответа на запрос:

```
{"face_id": "260023a3-9e7b-42a7-8e94-223278c83600", "url": "\/6\/faces\/260023a3-9e7b-42a7-8e94-223278c83600"}
```

## 1.5. Создание обработчика

Теперь, когда создан список и в него добавлено лицо, можно приступить к сравнению лиц. Дальнейшие действия разделены на два запроса – создание набора правил для обработки изображения (создание обработчика) и генерация события по созданному обработчику.

### 1.5.1. Описание запроса

Назначение	Тип запроса	Запрос
Запрос позволяет создать обработчик в Access Control Server, который будет использоваться в дальнейшем для создания событий	POST	10.16.30.82:5000/6/handlers

Запрос выполняется с параметрами, указанными в Таблице 5.

**Таблица 5.** Параметры запроса для создания обработчика

Параметр (key)	Описание	Значение
<b>Headers</b>		
Luna-Account-Id	Идентификатор аккаунта	6d071cca-fda5-4a03-84d5-5bea65904480
Content-Type	Тип содержимого в запросе	application/json
<b>Body</b>		
	<ul style="list-style-type: none"> <li>• description – краткое описание обработчика;</li> <li>• detect_policy – параметры политики детекции:               <ul style="list-style-type: none"> <li>• detect_face – определять ли лицо;</li> </ul> </li> <li>• extract_policy – параметры политики извлечения атрибутов:               <ul style="list-style-type: none"> <li>• extract_basic_attributes – извлекать ли базовые атрибуты;</li> <li>• extract_face_descriptor – извлекать ли БШ;</li> </ul> </li> <li>• match_policy – параметры политики сравнения:               <ul style="list-style-type: none"> <li>• candidates – определяет фильтры для обрабатываемых лиц;</li> <li>• origin – тип источника (лицо или событие);</li> </ul> </li> </ul>	(см. ниже)

Параметр (key)	Описание	Значение
	<ul style="list-style-type: none"> <li>list_id – идентификатор списка, в котором должен находиться кандидат.</li> </ul>	

### 1.5.2. Выполнение запроса

Пример cURL-запроса для выполнения запроса из командной строки:

```
curl --location --request POST 'http://10.16.30.82:5000/6/handlers' \
--header 'Luna-Account-Id: 6d071cca-fda5-4a03-84d5-5bea65904480' \
--header 'Content-Type: application/json' \
--data '{
  "description": "Обработчик для генерации события",
  "policies": {
    "detect_policy": {
      "detect_face": 1,
    },
    "extract_policy": {
      "extract_basic_attributes": 1,
      "extract_face_descriptor": 1
    },
    "match_policy": [
      {
        "candidates": {
          "origin": "faces",
          "list_id": "4ceacdb4-8ff6-4574-8c8a-1c423845cbad"
        }
      }
    ]
  }
}'
```

### 1.5.3. Результат выполнения запроса

При успешном выполнении запроса Система возвращает ID и URL обработчика.

Пример ответа на запрос:

```
{"handler_id": "b3413af3-31c4-4a62-bf0e-1aedaf784d9c", "url": "\/6\/handlers\/b3413af3-31c4-4a62-bf0e-1aedaf784d9c"}
```

Полученный "handler\_id" необходимо сохранить для последующего использования.

## 1.6. Генерация события и обработка handler

По созданному обработчику теперь необходимо сгенерировать событие, используя изображение, которое будет сравниваться с изображением из списка. В данном примере будет использоваться то же изображение, что было добавлено ранее в список.

### 1.6.1. Описание запроса

Назначение	Тип запроса	Запрос
Запрос позволяет сгенерировать события и обработать их соответствующим обработчиком	POST	10.16.30.82:5000/6/handlers/{handler_id}/events

Запрос выполняется с параметрами, указанными в Таблице 6.

**Таблица 6.** Параметры запроса для генерации события

Параметр (key)	Описание	Значение
<b>Headers</b>		
Luna-Account-Id	Идентификатор аккаунта	6d071cca-fda5-4a03-84d5-5bea65904480
Content-Type	Тип содержимого в запросе	image/jpeg
<b>Body</b>		
	Путь до изображения. Изображение загружено на тестовый стенд VisionLabs.	заранее (см. ниже)

### 1.6.2. Выполнение запроса

Пример cURL-запроса для выполнения запроса из командной строки:

```
curl --location --request POST 'http://10.16.30.82:5000/6/handlers/b3413af3-31c4-4a62-bf0e-1aedaf784d9c/events' \
  --header 'Luna-Account-Id: 6d071cca-fda5-4a03-84d5-5bea65904480' \
  --header 'Content-Type: image/jpeg' \
  --data-binary '@/root/Image1.jpg'
```

### 1.6.3. 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.1.5.3](#) 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": ""  
    }  
  ],  
  ...  
}
```