



# Система управления доступом «Smilart-КПП»

Руководство по использованию API v 9.1.2

# Содержание

О компании .....	2
О системе .....	3
Общие сведения .....	4
Основа протокола .....	4
Общее описание запросов .....	4
Общее описание ответов .....	5
Коды состояний .....	5
Принятые соглашения .....	7
Особенности работы с RabbitMQ .....	8
Основные понятия КПП .....	9
Описание протокола .....	10
Авторизация .....	10
Управление камерами .....	11
Получение списка PID камер .....	11
Получение информации о камере .....	11
Получение информации обо всех камерах .....	12
Получение последнего кадра с камеры .....	13
Управление базой персон .....	15
Добавление персоны .....	15
Удаление персоны .....	16
Редактирование анкеты персоны .....	16
Получение списка персон .....	17
Обучение персоны .....	18
Обучение фотографией, не находящейся в базе КПП .....	18
Добавление шаблона, созданного на основе фотографии .....	18
Добавление шаблона, созданного на основе изображения лица с координатами глаз .....	19
Автообучение .....	20
Начать автообучение .....	20
Прогресс .....	24
Результат автообучения .....	26
Остановить автообучение .....	26
Получение последних зафиксированных лиц .....	27
Получение последних зафиксированных лиц с координатами глаз .....	28
Получение шаблонов по id .....	29
Получение событий детектов, идентификации, верификации в КПП .....	30
Получение детектов КПП .....	30
Настройки очереди RabbitMQ для получения событий .....	30
Пример события детекта .....	30

Получение идентификаций КПП .....	33
Настройки очереди RabbitMQ для получения событий .....	33
Пример события идентификации .....	33
Верификации в КПП .....	35
Настройки очередей RabbitMQ для запросов и результатов верификаций .....	35
Пример запроса на верификацию .....	35
Пример ответа о верификации .....	36
Контакты .....	38

Авторские права компании «**Простые решения**» защищены законом.

Несанкционированное копирование, демонстрация, распространение или другое использование разрабатываемых компанией продуктов и документации к ним в какой бы то ни было форме и любыми средствами являются противозаконными.

# О компании

Компания **ООО «Простые решения»** осуществляет разработку и поставку высокоточных комплексных систем распознавания и идентификации объектов. Компания обеспечивает клиентов прогрессивными, удобными, гибкими и настраиваемыми продуктами по распознаванию, предлагает как конечные, так и базовые системы, которые клиенты могут самостоятельно адаптировать и развивать исходя из собственных потребностей.

# О системе

Система управления доступом «Smilart КПП» предназначена для разграничения доступа людей в помещения, здания и на территории охраняемых объектов. Программа работает на базе технологии распознавания лиц, что обеспечивает её быструю, надежную работу и эффективное применение для доступа в любое помещение.

Принципы работы системы:

1. Человек подходит к преграждающему устройству (дверь, турникет, ворота, шлагбаум) и смотрит в камеру.
2. Система сравнивает лицо объекта с лицами людей, которые имеют доступ в данное помещение.
3. В случае успешной идентификации подается сигнал устройству и человеку предоставляется доступ в помещение, в противном случае доступ будет закрыт.

Система управления доступом «Smilart КПП» работает с устройствами сторонних компаний. На данный момент поддерживаются контроллеры «Эра-2000», производимые ООО «Эра новых технологий», и IP-домофоны компании MОВOTIX.

Система управления доступом «Smilart КПП» также имеет систему оповещений об идентификациях. В этом случае при успешной идентификации отправляется сообщение по email. Если основной задачей является мониторинг проходов через КПП, а не управление доступом, то наличие преграждающих устройств необязательно.

Гибкость и масштабируемость системы обуславливает широкую область применения: от квартир и подъездов до крупных режимных объектов (предприятий, заводов и банков).

# Общие сведения

## Основа протокола

Протокол основан на текстовом формате [JSON](#). Передача данных происходит посредством обмена AMQP-сообщениями через сервер [RabbitMQ](#).

## Общее описание запросов

Все запросы (кроме запроса на авторизацию) содержат следующие данные:

1. Поле `"protocol"` — версия протокола, по которой происходит взаимодействие с КПП.
2. Поле `"session key"` — ключ сессии, полученный при авторизации.
3. Объект `"command"` — содержит описание команды, выполнение которой требуется в запросе:
  - a. Поле `"name"` — имя команды, которую требуется выполнить;
  - b. Объект `"parameters"` (опционально) — данные для выполнения команды.

Пример запроса без параметров:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "command name"
  }
}
```

Пример запроса с параметрами:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "command name",
    "parameters": {
      // параметры команды
    }
  }
}
```

# Общее описание ответов

Все ответы содержат следующие данные:

1. Поле *"protocol"* — версия протокола, по которой происходит взаимодействие с КПП.
2. Поле *"response name"* — имя ответа на запрос.
3. Поле *"status"* — код состояния запроса.
4. Поле *"description"* (опционально) — описание ошибки, если код состояния указывает на ошибку выполнения запроса.
5. Объект *"result"* (опционально) — данные, возвращаемые при выполнении запроса.

## Коды состояний

Успех

- **100 OK** — успешное выполнение запроса.

Ошибки доступа

- **200 AUTHORIZATION FAILED** — не корректная сессия авторизации;
- **201 SESSION EXPIRED** — истекло время жизни сессии;
- **202 PERMISSION DENIED** — запрос не доступен ни для одной из доменных групп, в которых состоит пользователь;
- **203 LICENCE RESTRICTIONS** — запрос невозможно выполнить из-за лицензионных ограничений.

Ошибки пользователя

- **300 INVALID REQUEST FORMAT** — ошибка в формате JSON-запроса;
- **301 INVALID REQUEST DATA** — не корректные данные авторизации (неверные логин/пароль, либо пользователь не принадлежит ни к одной из поддерживаемых КПП доменных групп);
- **302 TEMPLATE DUPLICATE** — невозможно выполнить изменения в наборе шаблонов персоны, так как в результате один шаблон будет использован для идентификации одной или нескольких персон многократно;
- **303 NOT UNIQUE PERSON KEY** — персона с указанным ключом уже существует, хотя на базе персон стоит уникальный индекс по ключу;
- **304 EMPTY PERSON KEY** — не указан ключ персоны, хотя на базе персон стоит уникальный индекс по ключу.

Ошибки сервера

- **400 INTERNAL ERROR** — внутренняя ошибка сервера;
- **401 TIMEOUT** — выполнение операции заняло недопустимое количество времени;

- **402 DATA IS NOT AVAILABLE** — не удалось найти запрашиваемые данные (например, если объект с id, указанным в запросе, был удален до выполнения запроса).

*Пример ответа об успешном выполнении запроса (без возврата данных):*

```
{
  "protocol": "2.0",
  "response name": "command name",
  "status": "100 OK"
}
```

*Пример ответа об успешном выполнении запроса (с возвратом данных):*

```
{
  "protocol": "2.0",
  "response name": "command name",
  "status": "100 OK",
  "result": {
    // данные с результатами запроса
  }
}
```

*Пример ответа об ошибке при обработке запроса:*

```
{
  "protocol": "2.0",
  "response name": "command name",
  "status": "400 INTERNAL ERROR",
  "description": "some error description"
}
```

# Принятые соглашения

Приняты следующие соглашения:

- ☑ запросы на авторизацию отправляются в AMQP exchange **api-authorization**,
- ☑ все авторизованные запросы отправляются в AMQP exchange **api-requests**,
- ☑ двоичные данные передаются закодированными в base64,
- ☑ имена всех полей запросов пишутся в несколько слов.

Мы стараемся сделать всё для того, чтобы все запросы были корректно обслужены. Несмотря на это, не стоит полагаться на то, что ответ на запрос будет получен всегда. Разумным будет установить определенное время ожидания ответа, и считать запрос неудавшимся по истечении этого времени.

# Особенности работы с RabbitMQ

В параметрах всех AMQP-сообщений с запросами, необходимо указать следующие данные:

1. Очередь для ответа, в которую будут отправлены все ответы на данный запрос (поле **reply-to** свойств AMQP-сообщения).
2. Correlation id сообщения (поле **correlation-id** свойств AMQP-сообщения).

Все AMQP-сообщения с ответом на конкретный запрос будут отправлены с параметром **correlation id**, указанным в этом запросе.

В случае если требуемые параметры AMQP сообщения не будут указаны, запрос будет проигнорирован (не будет отправлено никакого ответа).

Библиотеки для работы с RabbitMQ для языка программирования, который вы используете, можно найти на официальном сайте на странице [Clients & Developers Tools](#).

# Основные понятия КПП

- *Камера* — видеокамера, участвующая в работе КПП.
- *PID (persistent ID)* — уникальный идентификатор объекта.
- *Сессия* — период времени, в течение которого пользователь использует полученный при авторизации ключ сессии для выполнения запросов. По истечении сессии требуется повторная авторизация.
- *Время жизни сессии* — период времени, на который продлевается сессия пользователя после каждого полученного от него запроса.
- *База данных КПП* (далее "база") — совокупность данных, которые КПП использует в работе.
- *Персона* — объект в базе, представляющий конкретного человека, содержащий определенный набор данных, описывающих этого человека, имеющий уникальный идентификатор.
- *Лицо* — фотография в базе, которую КПП отметило как содержащую лицо.
- *Шаблон* — лицо, подготовленное для идентификации персоны по нему.
- *Обучение персоны* – соотнесение шаблонов с конкретной персоной. КПП будет использовать шаблоны сопоставленные персоне для идентификации этой персоны в потоке кадров с камер.

# Описание протокола

## Авторизация

Авторизация требуется для выполнения любых других запросов. В процессе авторизации серверу передаются логин и пароль пользователя домена, указанного при запуске КПП. В ответ сервер отправляет ключ сессии, который нужно указывать в каждом запросе в течение сессии, и список ролей, выполняемых пользователем. Запросы на авторизацию отправляются в AMQP exchange **api-authorization**.

*Запрос:*

```
{
  "protocol": "2.0",
  "login": "login",
  "password": "password"
}
```

*Ответ:*

```
{
  "protocol": "2.0",
  "response name": "authorization",
  "status": "100 OK",
  "result": {
    "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
    "roles": [
      "role1",
      "role2",
      // ...
      "roleN"
    ]
  }
}
```

# Управление камерами

## Получение списка PID камер

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "get camera pids"
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "get camera pids",
  "status": "100 OK",
  "result": {
    "camera pids": [
      "Camera1",
      "Camera2",
      // ...
      "CameraN"
    ]
  }
}
```

## Получение информации о камере

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "get camera info",
    "parameters": {
      "camera pid": "Camera1"
    }
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "get camera info",
  "status": "100 OK",
  "result":{
    "camera info": {
      "pid": "Camera1",
      "type": "mjpeg.arecont",
      "vendor": "smilart",
      "active": True,
      "exchangeName": "arecontCamera",
      "routingKey": "camera.arecont.mjpeg",
      "parameters": {
        "ipaddr": "192.168.0.15",
        "x0": 0,
        "y0": 0,
        "x1": 2048,
        "y1": 1536,
        "res": "full",
        "quality": 12,
        "fps": 5,
        "login": "login",
        "password": "password"
      }
    }
  }
}
```

## Получение информации обо всех камерах

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "get all cameras info"
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "get all cameras info",
  "status": "100 OK",
  "result":{
    "cameras info": [
      {
        "pid": "Camera1",
        "type": "mjpeg.arecont",
        "vendor": "smilart",
        "active": True,
        "exchangeName": "arecontCamera",
        "routingKey": "camera.arecont.mjpeg",
        "parameters": {
          "ipaddr": "192.168.0.15",
          "x0": 0,
          "y0": 0,
          "x1": 2048,
          "y1": 1536,
          "res": "full",
          "quality": 12,
          "fps": 5,
          "login": "login",
          "password": "password"
        }
      },
      // ...
    ]
  }
}
```

## Получение последнего кадра с камеры

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "get camera last frame",
    "parameters": {
      "camera pid": "Camera1"
    }
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "get camera last frame",
  "status": "100 OK",
  "result": {
    "frame": {
      "id": "frame id",
      "content": "base64 encoded image"
    }
  }
}
```

# Управление базой персон

Имя персоны не должно быть пустым или содержать более 100 символов.

## Добавление персоны

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "add person",
    "parameters": {
      "person name": "Person Name",
      "person key": "000-00001",
      "person description": "Something about person",
      "templates": [
        "template1 id",
        "template2 id",
        // ...
        "templateN id"
      ]
    }
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "add person",
  "status": "100 OK",
  "result": {
    "person id": "id"
  }
}
```

## Удаление персоны

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "delete person",
    "parameters": {
      "person id": "id"
    }
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "delete person",
  "status": "100 OK"
}
```

## Редактирование анкеты персоны

В запросе поле *"person id"* объекта *"parameters"* следует указать id персоны, в анкету которой необходимо внести изменения, а всем остальным полям присвоить желаемые значения.

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "update person",
    "parameters": {
      "person id": "id",
      "person name": "Some Name",
      "person key": "000-00001",
      "person description": "Something about person",
      "templates": [
        "template1 id",
        "template2 id",
        // ...
        "templateN id"
      ]
    }
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "update person",
  "status": "100 OK"
}
```

## Получение списка персон

Запрос:

```
{
  "protocol": "2.0",
  "session key": "8f61e130-af6a-4d48-878d-d2898662e576",
  "command": {
    "name": "get persons"
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "get persons",
  "status": "100 OK",
  "result": {
    "persons": [
      {
        "person id": "id",
        "person name": "Some Name",
        "person key": "000-00001",
        "person description": "Something about person",
        "templates": [
          "template1 id",
          "template2 id",
          // ...
          "templateN id"
        ]
      },
      // ...
    ]
  }
}
```

# Обучение персоны

## Обучение фотографией, не находящейся в базе КПП

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "add photo to person templates",
    "parameters": {
      "person id": "id",
      "photo": "base64 encoded image"
    }
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "add photo to person templates",
  "status": "100 OK"
}
```

## Добавление шаблона, созданного на основе фотографии

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "create template from photo",
    "parameters": {
      "photo": "base64 encoded image"
    }
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "create template from photo",
  "status": "100 OK",
  "result": {
    "template id": "new template id"
  }
}
```

## Добавление шаблона, созданного на основе изображения лица с координатами глаз

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "create template from face with eyes",
    "parameters": {
      "content": "base64 encoded image",
      "eyes": {
        "left": {
          "x": 98.7477,
          "y": 101.113
        },
        "right": {
          "x": 296.047,
          "y": 101.113
        }
      }
    }
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "create template from face with eyes",
  "status": "100 OK",
  "result": {
    "template id": "new template id"
  }
}
```

# Автообучение

Автообучение используется для автоматического выбора нескольких фотографий обучаемого человека из набора кадров с указанной камеры.

Необходимо, чтобы обучаемый человек находился перед камерой, с которой происходит автообучение.

В случае если в течение некоторого времени не будет получено кадров, на которых удалось распознать лицо, автообучение будет остановлено.

Для начала автообучения нужно отправить запрос с указанием камеры, с которой будут получаться кадры.

В ответ будет отправлено уведомление об успешном начале автообучения.

Далее КПП будет отправлять ответы с долей (число от 0 до 1.0) полученных и принятых для обучения кадров с лицом от их количества, требуемого для завершения автообучения.

Когда с камеры будет получено достаточное для автообучения число кадров, содержащих лицо, КПП отправит ответ с результатами автообучения: id созданных по выбранным фотографиям шаблонов.

Процесс автообучения можно остановить, прислав соответствующий запрос.

## Начать автообучение

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "start autolearning",
    "parameters": {
      "camera pid": "Camera1",
      "addHeadPoseStatistics": True,
      "headPoseMessageFields": ["position", "angles"]
    }
  }
}
```

Параметры:

- `camera pid` — обязательный параметр. ID камеры, с которой пойдут кадры с лицами для автообучения;
- `addHeadPoseStatistics` — опциональный параметр. Если указан данный параметр и автообучение осуществляется по поиску ракурсов (направлений) головы, то в [сообщении о прогрессе автообучения](#) появится статистика о количестве найденных ракурсов;
- `headPoseMessageFields` — опциональный параметр. Если указан данный параметр и автообучение осуществляется по поиску ракурсов головы, то в [сообщении о прогрессе автообучения](#) появится сообщение о последнем ракурсе головы с указанными полями: `angles` — если поле указано, то в [сообщении](#) будут указаны углы поворота головы.

**position** — если поле указано, то в **сообщении** будет указан последний ракурс из таблицы ниже.

<b>OutsideTop</b> — голова направле на выше допустим ого угла	
--	--

Outsidel  
eft

—  
го  
ло  
ва  
на  
пр  
ав  
ле  
на  
ле  
ве  
е  
до  
пу  
ст  
и  
м  
ог  
о  
уг  
ла

о	о	о	о
Do	Do	Do	уг
wn	wn	wn	ла
Pi	Pi	Pi	
tc	tc	tc	
hL	hM	hR	
ef	id	igt	
tY	dL	ht	
aw	eY	Ya	
—	aw	w	
го	—	—	
ло	го	го	
ва	ло	ло	
на	ва	ва	
пр	на	на	
ав	пр	пр	
ле	ав	ав	
на	ле	ле	
вн	на	на	
из	вн	вн	
вл	из	из	
ев		вп	
о		ра	
		во	

**OutsideDow**  
**n** — голов  
а  
направле  
на ниже  
допустим  
ого угла

Ответ:

```
{
  "status": "100 OK",
  "protocol": "2.0",
  "result": {
    "type": "3x3",
    "minCoverageRatioPerPose": 0.5,
    "minCollectedDetectsPerPose": 1
  },
  "response name": "start autolearning - start"
}
```

В ответе **result** находятся некоторые параметры конфигурации, с которыми началось автообучение:

- **type** — тип автообучения, какие положения головы (ракурсы) накапливать для обучения:

**3x3** — 9 положений головы (верх лево, верх, верх право, лево, центр, право, низ лево, низ, низ право);

**cross** — 5 положений головы (верх, лево, центр, право, низ);

**horizontal** — 3 положения головы (лево, центр, право);

**center** — 1 положение головы (центр);

**allDetects** — набирать каждый детект без сортировки по ракурсам.

- **minCoverageRatioPerPose** — минимальная площадь покрытия набранными детектами внутри ракурса.
- **minCollectedDetectsPerPose** — количество лиц, которое нужно набрать для каждого ракурса, для успешного завершения автообучения, если отключено определение лучшего лица по площади (**minCoverageRatioPerPose: 0**)

## Прогресс

Если тип (**type**) автообучения, который пришел в **ответ на старт** — **allDetects**, то в процессе посылается только прогресс автообучения — **progress** (Растёт от 0 до 1.0 и соответствует доле кадров с лицом, полученных с камеры и принятых для обучения, от числа кадров с лицом, необходимых для автообучения)

Ответ:

```
{
  "protocol": "2.0",
  "response name": "start autolearning - progress",
  "status": "100 OK",
  "result": {
    "progress": 0.5809523463249207
  }
}
```

Если тип (**type**) автообучения, который пришел в **ответ на старт** отличен от **allDetects**, то присылается расширенная информация по прогрессу, согласно полям заданным при **старте автообучения**

Ответ:

```
{
  "protocol": "2.0",
  "response name": "start autolearning - progress",
  "status": "100 OK",
  "result": {
    "progress": 0.5809523463249207,
    "headPose": {
      "position": "DownPitchLeftYaw",
      "angles": {
        "yaw": 22.276460647583008,
        "roll": -6.815710544586182,
        "pitch": 12.593829154968262
      }
    },
    "collectedHeadPoses": {
      "DownPitchMiddleYaw": 0.0,
      "MiddlePitchRightYaw": 0.0,
      "DownPitchLeftYaw": 0.0,
      "MiddlePitchLeftYaw": 0.9523809552192688,
      "TopPitchLeftYaw": 0.0,
      "DownPitchRightYaw": 0.0,
      "MiddlePitchMiddleYaw": 1.0,
      "TopPitchRightYaw": 0.0,
      "TopPitchMiddleYaw": 0.9523809552192688
    }
  }
}
```

Поля запроса ответа:

- **progress** — значение прогресса. Растёт от 0 до 1.0 и соответствует доле кадров с лицом, полученных с камеры и принятых для обучения, от числа кадров с лицом, необходимых для автообучения;
- **collectedHeadPoses** — статистика о заполненности соответствующего ракурса, появляется, если задан параметр в [запросе старта автообучения](#) (Растёт от 0 до 1.0);
- **headPose** — информация о последнем ракурсе головы с полями указанными в [запросе старта автообучения](#).
  - position** — (может отсутствовать) последний поворот головы из [таблицы ракурсов](#);
  - angles** — (может отсутствовать) углы поворота головы (когда человек смотрит прямо в объектив камеры, все углы равны 0):
    - pitch** — угол наклона головы вверх-вниз (положительное значение — голова наклонена вниз, отрицательное — вверх);
    - yaw** — угол поворота головы влево-вправо (положительное значение — голова повернута влево, отрицательное — вправо);
    - roll** — угол наклона головы влево-вправо (положительное значение — голова наклонена влево, отрицательное — вправо).

## Результат автообучения

Ответ:

```
{
  "protocol": "2.0",
  "response name": "start autolearning - result",
  "status": "100 OK",
  "result": {
    "templates": [
      "template1 id",
      "template1 id",
      // ...
      "templateN id"
    ]
  }
}
```

## Остановить автообучение

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "stop autolearning"
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "stop autolearning",
  "status": "100 OK"
}
```

## Получение последних зафиксированных лиц

В процессе работы КПП сохраняется некоторое количество последних кадров с камер, на которых (кадрах) было найдено лицо, независимо от того было ли это лицо идентифицировано как персона из базы или нет. Это может быть полезно для обучения новой персоны.

Для получения этих кадров нужно отправить запрос с указанием камеры, с которой они были получены, и их максимальное количество.

В ответ будут присланы id и содержимое кадров. Количество кадров в ответе может быть меньше, чем было указано в запросе, если в базе не имеется достаточное количество кадров с указанной камеры.

*Запрос:*

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "get last faces",
    "parameters": {
      "camera pid": "Camera1",
      "count": 20 // количество лиц в ответе
    }
  }
}
```

*Ответ:*

```
{
  "protocol": "2.0",
  "response name": "get last faces",
  "status": "100 OK",
  "result": {
    "faces": [
      {
        "id": "face id",
        "content": "base64 encoded image"
      },
      // ...
    ]
  }
}
```

# Получение последних зафиксированных лиц с координатами глаз

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "get last faces with eyes",
    "parameters": {
      "camera pid": "Camera1",
      "count": 20 // количество лиц в ответе
    }
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "get last faces with eyes",
  "status": "100 OK",
  "result": {
    "faces": [
      {
        "id": "face id",
        "content": "base64 encoded image",
        "eyes": {
          "left": {
            "x": 98.7477,
            "y": 101.113
          },
          "right": {
            "x": 296.047,
            "y": 101.113
          }
        }
      },
      // ...
    ]
  }
}
```

# Получение шаблонов по id

Запрос:

```
{
  "protocol": "2.0",
  "session key": "b559dbf4-1cb0-4e46-a93a-b9435715ef30",
  "command": {
    "name": "get templates",
    "parameters": {
      "templates": [
        "template1 id",
        "template2 id",
        // ...
        "templateN id"
      ]
    }
  }
}
```

Ответ:

```
{
  "protocol": "2.0",
  "response name": "get templates",
  "status": "100 OK",
  "result": {
    "templates": [
      {
        "id": "template id",
        "content": "base64 encoded image data"
      },
      // ...
    ]
  }
}
```

# Получение событий детектов, идентификации, верификации в КПП

В отличие от запросов, описанных выше, для получения событий КПП авторизация не требуется.

## Получение детектов КПП

### Настройки очереди RabbitMQ для получения событий

При настройке системы можно сконфигурировать очередь в которую будут отправляться результаты детектов.

*Параметры конфигурации по умолчанию*

```
DetectEventSenderConfigurator.setConfig({exchangeName:"platform.info.fd",  
exchangeType:"fanout"})
```

При детекте происходит отправка в очередь AMQP:

exchange name = `exchangeName`

exchange type = `exchangeType`

routing key = `cameraPID.schemaPID`

где

`cameraPID` — PID камеры, с которой пришел кадр с лицом

`schemaPID` — PID схемы в КПП, в котором участвует данная камера

### Пример события детекта

```
{  
  "camera pid": "arecont1",  
  "face": {  
    "face document": {  
      "members": {  
        "id": "74",  
        "faceBlobURL": "http://smilart4:80/fac7f42a-188f-428e-b5f0-  
2e290d6d8734/9af8b45a-e12b-41ae-8545-a967a10f606a/976823367402987859/3143878-160000",  
        "faceJpegBlobURL": "http://smilart4:80/fac7f42a-188f-428e-b5f0-  
2e290d6d8734/9af8b45a-e12b-41ae-8545-a967a10f606a/976823367402987859/3303886-24918",  
        "corresponding response id": "56ebf8b23ca3de000197991a",  
        "type": "image/MONO",  
        "width": 400,  
        "height": 400,  
        "eyes": {  
          "left": {  
            "x": 282.88263,  
            "y": 163.34508
```

```

    },
    "right": {
      "x": 142.63243,
      "y": 155.79536
    }
  },
  "face quad": {
    "top left": {
      "x": 104.61316,
      "y": 95.37991
    },
    "top right": {
      "x": 314.0938,
      "y": 95.37991
    },
    "bottom left": {
      "x": 104.61316,
      "y": 304.86044
    },
    "bottom right": {
      "x": 314.0938,
      "y": 304.86044
    }
  }
},
"face rectangle": {
  "top left": {
    "x": 3833.9604,
    "y": 3117.5122
  },
  "top right": {
    "x": 4660.8853,
    "y": 3117.5122
  },
  "bottom left": {
    "x": 3833.9604,
    "y": 3944.4365
  },
  "bottom right": {
    "x": 4660.8853,
    "y": 3944.4365
  }
},
"left eye outer corner": {
  "x": 4537.679,
  "y": 3385.8047
},
"right eye outer corner": {
  "x": 3984.0415,
  "y": 3356.0022
}

```

```

}
},
"detect parameters": {
  "detect type": "light",
  "detect version": 9,
  "cutting parameters": {
    "shift x face fraction": 0,
    "shift y face fraction": 0,
    "face rectangle max height pixels": 400,
    "face rectangle max width pixels": 400,
    "border scale factor": 2
  },
  "min outer eye corners distance pixels": 75,
  "preprocessing max answers": 1,
  "is cutting horizontal": true,
  "number of 90 degree clockwise rotations": 1
},
"detect statistics": {
  "camera driver times": {
    "camera driver host": "smilart4",
    "frame arrived from camera to driver java timestamp ms": 1458305202575,
    "arrival from camera to frame stored duration ms": 1458305202575
  },
  "detect times": {
    "workflow host": "smilart4",
    "frame arrived java timestamp ms": 1458305202581,
    "arrival to face detect request sent duration ms": 1,
    "arrival to face detect response arrived duration ms": 48
  }
},
"frame blob url": "http://smilart4:80/fac7f42a-188f-428e-b5f0-2e290d6d8734/4ae512c5-6ec1-443e-9210-6cf7fa11f96a/5573303906368852961/5365062312-2789936"
}

```

# Получение идентификаций КПП

## Настройки очереди RabbitMQ для получения событий

При настройке системы можно сконфигурировать очередь в которую будут отправляться результаты идентификаций.

Параметры конфигурации по умолчанию

```
IdentificationConfigurator.setConfig({resultExchangeName:"checkpoint.face-recognitions", resultExchangeType:"topic"})
```

При идентификации происходит отправка в очередь AMQP:

exchange name = `resultExchangeName`

exchange type = `resultExchangeType`

routing key = `cameraPID.schemaPID`

где

`cameraPID` — PID камеры, с которой пришел кадр с лицом

`schemaPID` — PID схемы в КПП, в котором участвует данная камера

## Пример события идентификации

```
{
  "face from camera": {
    "right eye outer corner": {
      "y": 162.99729919433594,
      "x": 119.18872833251953
    },
    "left eye outer corner": {
      "y": 156.23081970214844,
      "x": 265.41778564453125
    },
    "face document": {
      "members": {
        "eyes": {
          "right": {
            "y": 162.9973,
            "x": 119.18873
          },
          "left": {
            "y": 156.23082,
            "x": 265.4178
          }
        }
      },
      "faceBlobURL": "http://smilart2:80/a7f36cd1-7a73-4a62-98fd-215567e93aab/6c59ac61-ed83-4804-99ef-683930a33ecb/1504365059835814950/151989482751-160000",
      "corresponding response id": "578f694af1c92900014920ed",
    }
  }
}
```

```

    "faceJpegBlobURL": "http://smilart2:80/a7f36cd1-7a73-4a62-98fd-
215567e93aab/6c59ac61-ed83-4804-99ef-683930a33ecb/1504365059835814950/151989642759-
30360",
    "height": 400,
    "width": 400,
    "type": "image/MONO"
  }
},
"person description": "from arecont1",
"camera PID": "arecont1",
"identification statistics": {
  "camera driver times": {
    "frame arrived from camera to driver java timestamp ms": 1469016394699,
    "arrival from camera to frame stored duration ms": 1469016394699,
    "camera driver host": "smilart2"
  },
  "identification workflow times": {
    "workflow host": "smilart2",
    "arrival to identification response arrived duration ms": 253,
    "frame arrived java timestamp ms": 1469016394700,
    "arrival to identification request sent duration ms": 194,
    "arrival to identification signal sending duration ms": 253,
    "arrival to face detect response arrived duration ms": 194,
    "arrival to face detect request sent duration ms": 0
  }
},
"face JPEG from camera": "{140480 bytes binary base64 image}",
"person key": "222-222",
"correlation": 0.9907351732254028,
"scheme PID": "arecont1Scheme",
"person name": "Aurora",
"face JPEG from base": "{116572 bytes binary base64 image}",
"eyes from base": {
  "right": {
    "y": 162.0,
    "x": 119.0
  },
  "left": {
    "y": 156.0,
    "x": 265.0
  }
},
"template ID": "578e3f1bf1c929000126d060",
"identification threshold": 0.7200000286102295,
"person ID": "578e3f2bf1c929000126d2bb"
}

```

# Верификации в КПП

## Настройки очередей RabbitMQ для запросов и результатов верификаций

При настройке системы можно сконфигурировать очередь в которую будут отправляться запросы и результаты верификаций.

*Параметры конфигурации по умолчанию*

```
VerificationConfigurator.setConfig({apiExchangeName:"checkpoint.verification-requests", resultExchangeName:"verificationResult", resultExchangeType:"topic"})
```

Отправлять запрос на верификацию необходимо в очередь AMQP:

exchange name = `apiExchangeName`  
exchange type = `topic`  
routing key = `#`

При верификации в КПП происходит отправка в очередь AMQP:

exchange name = `resultExchangeName`  
exchange type = `resultExchangeType`  
routing key = `cameraPID.schemaPID.answerType`  
где

`cameraPID` — PID камеры, с которой пришел кадр с лицом

`schemaPID` — PID схемы в КПП, в котором участвует данная камера

`answerType` — тип ответа на запрос верификации, один из:

`unknownPersonKey` — персона не найдена по ключу

`success` — пришел ответ на запрос верификации

`timeout` — вышел таймаут запроса на верификацию

`failure` — верификация не удалась

## Пример запроса на верификацию

```
{  "person key": "111-111",  "camera pid": "camera"}
```

## Пример ответа о верификации

Результат успешной верификации (routing key = camera.cameraScheme.success)

```
{
  "person description": "",
  "camera pid": "camera",
  "person key": "111-111",
  "scheme pid": "cameraScheme",
  "person name": "Man1",
  "result": {
    "best correlation": 0.93490404,
    "face document": "http://station421:80/4c69f501-f863-4e78-bf1c-
a12a989cc177/f32db3d0-47c9-4804-b0c3-6bf7154bb71f/2128565726090271650/157280745319-
22805",
    "correlation template id": "57680c7464a6c200019d82ce",
    "right eye outer corner": {
      "y": 154.91404724121094,
      "x": 134.696044921875
    },
    "statistic": {
      "verificationWorkflowTimes": {
        "workflowHost": "station421",
        "sentToFdDurationMs": 0,
        "frameArrivedJavaTimestampMs": 1469020317258,
        "gotFdResultDurationMs": 43,
        "sentToVerificationDurationMs": 43,
        "gotVerificationResultDurationMs": 91,
        "verificationFinishedDurationMs": 91
      },
      "cameraDriverTimes": {
        "saveFrameToBlobStoreMs": 1469020317256,
        "cameraDriverHost": "station421",
        "frameArrivedFromCameraToDriverJavaTimestampMs": 1469020317256
      }
    },
    "verification threshold": 0.72,
    "left eye outer corner": {
      "y": 161.96359252929688,
      "x": 258.24658203125
    }
  },
  "person id": "57680c7e64a6c200019d82e1"
}
```

Результат, когда вышел таймаут верификации (`routing key = camera.cameraScheme.timeout`)

```
{
  "person description": "",
  "camera pid": "camera",
  "person key": "111-111",
  "scheme pid": "cameraScheme",
  "person name": "Man1",
  "person id": "57680c7e64a6c200019d82e1"
}
```

Результат, когда персона не найдена по ключу (`routing key = camera.cameraScheme.unknownPersonKey`)

```
{
  "camera pid": "camera",
  "person key": "111-111",
  "scheme pid": "cameraScheme"
}
```

Результат, когда произошла ошибка во время верификации (`routing key = camera.cameraScheme.failure`)

```
{
  "camera pid": "camera",
  "person key": "111-111"
}
```



В случае отсутствия ответов на запросы проверьте, что камера, указанная в запросах, присутствует в конфигурации КПП и имеется схема с типом "combined" или "verification", к которой она привязана.

# Контакты

Адрес: 127051, Россия, г. Москва, 2-й Колобовский переулок, д. 9/2, стр. 4

Телефон: +7 (499) 704-25-34 (Московский офис)

+7 (8172) 50-14-62 (Вологодский офис)

Е-mail: [info@smilart.com](mailto:info@smilart.com)

Техническая поддержка: [support@smilart.com](mailto:support@smilart.com)