



Super API (4.6.0)

License: [MIT](#)

А Доступ к методам REST API

Методы, обеспечивающие авторизацию пользователей

A.1 Авторизация ВИС с использованием УЗ ЕЦХД

POST /api/v2/auth/vis



Доступ к программным интерфейсам ЕЦХД (методы REST API) предоставляется пользователям с **Учетной записью ЕЦХД** при наличии **группового права** — «ИИПМ. Пользователь ИНС ОД 'приоритет'».

AUTHORIZATIONS: > externalLogin

REQUEST BODY SCHEMA: application/json
required

login required	string Логин УЗ ЕЦХД Пользователя
password required	string Пароль УЗ ЕЦХД Пользователя

Responses

✓ 200

Успешная авторизация

RESPONSE SCHEMA: application/json

accessToken required	string Access Token авторизации
refreshToken required	string Refresh Token авторизации



refreshTokenExpiresIn
required

integer
Время жизни refresh token (в секундах)

> 401

Ошибка авторизации

> 403

Отсутствие прав доступа к сервису

> 502

Ошибка авторизации вследствие внутренней ошибки во внешней Подсистеме авторизации и аутентификации ЕЦХД

> default

Внутренняя ошибка сервера

Request samples

Payload

Content type
application/json

Copy

```
{
  "login": "hi",
  "password": "test"
}
```

Response samples

200

401

403

502

default

Content type
application/json

Copy

```
{
  "accessToken": "dakdaksdk12k3addon12n",
  "refreshToken": "dsadk1k23k2ivdsjnv1pkeosmdam",
  "accessTokenExpiresIn": 3600,
  "refreshTokenExpiresIn": 86400
}
```



A.2 Обновление сессии

POST /api/v2/auth/refresh



Метод предназначен для обновления пользовательской сессии на основании ранее выданного refresh-токена.

Условия использования:

- Метод должен вызываться клиентским приложением в случае истечения срока действия access-токена.
- Обновление возможно только при наличии **валидного refresh-токена**, выданного в рамках предыдущей аутентификации.
- В ответе возвращается пара новых **access-** и **refresh-токенов**.

Ограничения:

- Если refresh-токен недействителен или истёк, пользователь должен пройти повторную аутентификацию.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

REQUEST BODY SCHEMA: application/json
required

refreshToken required	string Refresh Token авторизации
--------------------------	-------------------------------------

Responses

> 200

Успешное обновление токенов

RESPONSE SCHEMA: application/json

accessToken required	string Access Token авторизации
refreshToken required	string Refresh Token авторизации
accessTokenExpiresIn required	integer Время жизни access token (в секундах)
refreshTokenExpiresIn required	integer Время жизни refresh token (в секундах)

> 400

Некорректный формат запроса

> 401

Ошибка авторизации



Refresh-токен отозван

> default

Внутренняя ошибка сервера

Request samples

Payload

Content type

application/json

Copy

```
{
  "refreshToken": "dsadk1k23k2ivdsjnv1pkeosmdam"
}
```

Response samples

200

400

401

403

default

Content type

application/json

Copy

```
{
  "accessToken": "dakdaksdk12k3addon12n",
  "refreshToken": "dsadk1k23k2ivdsjnv1pkeosmdam",
  "accessTokenExpiresIn": 3600,
  "refreshTokenExpiresIn": 86400
}
```

Б Доступ к источникам и экспортерам данных

Система поддерживает следующие типы источников данных:

1. **schedule** — расписания ЕЦХД

Расписание снятия скриншотов, полученных с определенного списка камер в определенное время.

2. **odh** — КИНС

Аппаратно-программный комплекс, устанавливаемый на транспортных средствах, предназначенный для выявления нарушений и объектов контроля с использованием Интеллектуальных модулей.



4. **gin** — БВС (беспилотное воздушное судно).
5. **satellite** — спутниковые изображения, загружаемые в виде архивов.
6. **rabbitMqInternal** — очередь RabbitMQ, обслуживаемая Подсистемой ИИПМ.
7. **kafkaInternal** — очередь Kafka, обслуживаемая Подсистемой ИИПМ. Для источника kafkaInternal максимальный допустимый размер сообщения установлен на уровне 10 МБ. Ограничение обусловлено особенностями конфигурации брокера Kafka: при превышении указанного значения сообщение не принимается в топик.
8. **rabbitMqExternal** — очередь RabbitMQ, предоставляемая пользователями.
9. **kafkaExternal** — очередь Kafka, предоставляемая пользователями.
10. **camera** — камера ЕЦХД.
11. **cameraGroup** — источник данных, представляющий группу камер ЕЦХД, формируемую по фильтрам.
12. **cameraDevice** — источник данных, представляющий камеру ЕЦХД, к которой привязан внешний источник данных(device).
13. **cameraDeviceGroup** — источник данных, представляющий группу камер ЕЦХД, к которым привязан внешний источник данных(device), формируемую по фильтрам.
14. **archive** — архив изображений.
15. **device** — внешний источник данных.
16. **httpImage** — источник изображений, загружаемых по HTTP.

Источник предназначен для приёма изображений через HTTP-запросы в бинарном виде или по URL. Загруженные изображения автоматически обрабатываются всеми задачами, в конфигурации которых указан источник идентификатор источника **httpImage**.

Результаты обработки возвращаются в зависимости от режима вызова метода:

- при синхронном режиме — в ответе запроса;
- при асинхронном режиме — после завершения обработки, через стандартные механизмы получения результатов;
- при использовании webhook экспортера — посредством отправки HTTP POST запроса с результатами обработки в формате `application/json` на webhook endpoint, указанный в параметрах экспортера.

Б.1 Создание нового источника данных

POST /api/v2/sources



Условия использования:

Разрешено создание вручную только следующих типов источников:

1. rabbitMqInternal
2. kafkaInternal
3. rabbitMqExternal
4. kafkaExternal
5. satellite
6. archive



9. httpImage

Источники следующих типов создаются системой автоматически:

- 1. odh
- 2. panoramic
- 3. gin
- 4. device
- 5. schedule
- 6. camera
- 7. cameraDevice

Ограничения:

Параметр type обязателен и должен строго соответствовать одному из поддерживаемых значений. Попытка создания источника с неподдерживаемым или автоматически создаваемым типом приводит к ошибке 400 Bad Request.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

REQUEST BODY SCHEMA: application/json required

One of

rabbitMqInternalSourceCreate

kafkaInternalSourceCreate

rabbitMqExternalSourceCreate

kafkaExternalSourceCreate

cameraGroupSourceCreate

cameraGroupDeviceSourceCreate

httpImageSourceCreate

type required	string Тип источника данных. Должно быть строго указано значение: rabbitMqInternal — очередь RabbitMQ, обслуживаемая Подсистемой ИИПМ. Value: "rabbitMqInternal"
title required	string Название источника данных, задаваемое пользователем.
info > required	object Дополнительные параметры источника.

Responses

✓ 200

Successful creation

RESPONSE SCHEMA: application/json

One of

rabbitMqInternalSourceInfo

rabbitMqExternalSourceInfo

kafkaInternalSourceInfo

kafkaExternalSourceInfo

SatelliteGinPanoramicArchiveInfo

cameraGroupInfo

cameraGroupDeviceInfo

httpImageInfo



ИИПМ источник данных

Должно быть строго указано значение: `rabbitMqInternal` — очередь RabbitMQ, обслуживаемая Подсистемой ИИПМ.

Value: `"rabbitMqInternal"`

<code>title</code> <code>required</code>	<code>string</code> Название источника данных, задаваемое пользователем.
<code>info ></code> <code>required</code>	<code>object</code> Дополнительные параметры источника.
<code>id</code> <code>required</code>	<code>string <uuid></code> Уникальный Системный идентификатор источника данных.

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> default

Внутренняя ошибка сервера

Request samples

Payload

Content type
application/json

Example
rabbitMqInternalSourceCreate

Copy Expand all Collapse all

```
{
  "type": "rabbitMqInternal",
  "title": "string",
  - "info": {
    "format": "binary"
  }
}
```

Response samples

- 200
- 400
- 401
- default

Content type
application/json



Copy Expand all Collapse all

```
{
  "type": "rabbitMqInternal",
  "title": "очередь для обработки",
  - "info": {
    "format": "binary",
    "dsn": "amqp://localhost:5672/vhost",
    "exchange": "systemEventsExchange",
    "key": "imageEventsKey"
  },
  "id": "d3a8b5f2-4c1e-4f7b-9a6b-2e7d1c3a9b84"
}
```

Б.2 Получение источников, доступных пользователю

GET /api/v2/sources



AUTHORIZATIONS: > (bearerAuth and externalLogin)

QUERY PARAMETERS

type	Array of strings Items Enum: "rabbitMqInternal" "kafkaInternal" "rabbitMqExternal" "kafkaExternal" "schedule" "satellite" "gin" "panoramic" "camera" "archive" "device" "odh" "cameraGroup" "cameraDevice" "cameraDeviceGroup" "httpImage" Example: type=rabbitMqInternal&type=kafkaExternal Фильтрация по типу источника данных
sourceId	Array of strings <uuid> [items <uuid >] Example: sourceId=eb54723c-c3b0-4a75-b12c-f39a81d34d4d&sourceId=ffdc41d0-671c-4f9a-9e0a-b0f7b18aef23 Фильтрация по Системным идентификаторам источников данных
groupId	Array of strings <uuid> [items <uuid >] Example: groupId=eb54723c-c3b0-4a75-b12c-f39a81d34d4d&groupId=ffdc41d0-671c-4f9a-9e0a-b0f7b18aef23 Фильтрация по вхождению источников в группы. Передаются Системные идентификаторы групп
limit	integer >= 1 Default: 100 Максимальное количество возвращаемых элементов в одном ответе. Значение по умолчанию — 100.
offset	integer >= 0 Default: 0



scheduleId

Array of strings <uuid> [items <uuid >]

Example: `scheduleId=aal23456-1111-2222-3333-abcdefabcdef`

Фильтрация по расписанию ЕЦХД (сквозной идентификатор расписания ЕЦХД)

cameraId

Array of strings <uuid> [items <uuid >]

Example: `cameraId=a2123456-1111-2222-3333-abcdefabcdef`

Фильтрация по камере ЕЦХД (сквозной идентификатор камеры ЕЦХД)

Responses

✓ 200

Успешный ответ со списком всех источников

RESPONSE SCHEMA: application/json

`sources >`
`required`

Array of rabbitMqInternalSourceInfo (object) or rabbitMqExternalSourceInfo (object) or kafkaInternalSourceInfo (object) or kafkaExternalSourceInfo (object) or scheduleInfo (object) or deviceInfo (object) or SatelliteGinPanoramicArchiveInfo (object) or cameraInfo (object) or cameraGroupInfo (object) or cameraDeviceInfo (object) or cameraGroupDeviceInfo (object) or httpImageInfo (object) or OdhInfo (object)

`hasMore`
`required`

boolean

Признак наличия дополнительных элементов по переданным фильтрам. Параметр `hasMore` в ответе API указывает, есть ли ещё данные для выдачи. Если количество возвращённых элементов равно `limit` и в базе остаются данные, `hasMore` устанавливается в `true`, что сигнализирует клиенту о необходимости сделать следующий запрос с увеличенным `offset`. Если же возвращено меньше, чем `limit`, значит, данные закончились и `hasMore` будет `false`, что говорит клиенту, что дополнительных запросов делать не нужно.

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> 404

Запрашиваемый ресурс не найден

> default

Внутренняя ошибка сервера

Response samples



Content type

application/json

Copy Expand all Collapse all

```
{
  - "sources": [
    - {
      "type": "rabbitMqInternal",
      "title": "очередь для обработки",
    - "info": {
      "format": "binary",
      "dsn": "amqp://localhost:5672/vhost",
      "exchange": "systemEventsExchange",
      "key": "imageEventsKey"
    },
      "id": "d3a8b5f2-4c1e-4f7b-9a6b-2e7d1c3a9b84"
    },
  ],
  "hasMore": true
}
```

Б.3 Создание экспортера данных

POST /api/v1/exporters



Создание экспортера данных Подсистема должна поддерживать следующие типы источников, в которые направляются сообщения о результатах обработки изображений:

1. **rabbitMqInternal** — очередь RabbitMQ Подсистемы
2. **kafkaInternal** — очередь Kafka Подсистемы
3. **rabbitMqExternal** — очередь RabbitMQ пользователя
4. **kafkaExternal** — очередь Kafka пользователя
5. **webhook** — отправка HTTP POST запроса с результатами обработки в формате `application/json` на webhook endpoint, указанный в параметрах экспортера.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

REQUEST BODY SCHEMA: application/json
required

One of

rabbitMqExternalExporterCreate

kafkaExternalExporterCreate

RabbitMqInternalExporterCreate

KafkaInternalExporterCreate

webhookExporterCreate

type
required

string

Тип экспортера.

Для внешнего экспортера RabbitMQ должно быть строго значение

`rabbitMqExternal`.



title
required

string
Название экспортера.

info >
required

object
Параметры подключения к внешнему RabbitMQ экспортёру.

Responses

✓ 200

Успешное создание экспортера

RESPONSE SCHEMA: application/json

One of

- rabbitMqExternalExporterInfo
- ExporterRabbitInternalResponse
- kafkaExternalExporterInfo
- ExporterKafkaInternalResponse
- ExporterWebhookResponse

type
required

string
Тип экспортера.
Для внешнего экспортера RabbitMQ должно быть строго значение `rabbitMqExternal`.
Value: "rabbitMqExternal"

title
required

string
Название экспортера.

info >
required

object
Параметры подключения к внешнему RabbitMQ экспортёру.

id
required

string <uuid>
Уникальный Системный идентификатор экспортера.

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> default

Внутренняя ошибка сервера

Request samples

Payload



Example
rabbitMqExternalExporterCreate

Copy Expand all Collapse all

```
{
  "type": "rabbitMqExternal",
  "title": "some title",
  - "info": {
    "dsn": "amqp://localhost:5672/vhost",
    "exchange": "someEventsExchange",
    "key": "someEventsKey",
    - "auth": {
      "username": "rabbitUser",
      "password": "rabbitPass"
    }
  }
}
```

Response samples

- 200
- 400
- 401
- default

Content type
application/json

Example
rabbitMqExternalExporterInfo

Copy Expand all Collapse all

```
{
  "type": "rabbitMqExternal",
  "title": "some title",
  - "info": {
    "dsn": "amqp://localhost:5672/vhost",
    "exchange": "someEventsExchange",
    "key": "someEventsKey"
  },
  "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08"
}
```

Б.4 Получение списка экспортеров данных

GET /api/v1/exporters



Получение списка экспортёров данных



1. **rabbitMqInternal** — очередь RabbitMQ Подсистемы
2. **kafkaInternal** — очередь Kafka Подсистемы
3. **rabbitMqExternal** — очередь RabbitMQ пользователя
4. **kafkaExternal** — очередь Kafka пользователя
5. **webhook** — отправка HTTP POST запроса с результатами обработки в формате `application/json` на webhook endpoint, указанный в параметрах экспортера

AUTHORIZATIONS: > (bearerAuth and externalLogin)

QUERY PARAMETERS

limit	integer >= 1 Default: 100 Максимальное количество возвращаемых элементов в одном ответе. Значение по умолчанию — 100.
offset	integer >= 0 Default: 0 Смещение относительно начала выборки. Используется для постраничной навигации. Значение по умолчанию — 0.
type	Array of strings Items Enum: rabbitMqExternal kafkaExternal rabbitMqInternal kafkaInternal webhook Фильтрация по типу источников данных. Возможные значения: • <code>rabbitMqExternal</code> — очередь RabbitMQ пользователя • <code>kafkaExternal</code> — очередь Kafka пользователя • <code>rabbitMqInternal</code> — очередь RabbitMQ Подсистемы • <code>kafkaInternal</code> — очередь Kafka Подсистемы • <code>webhook</code> — отправка HTTP POST запроса с результатами обработки в формате <code>application/json</code> на webhook endpoint, указанный в параметрах экспортера.

Responses

✓ 200

Список экспортёров данных

RESPONSE SCHEMA: application/json

exporters > required	array
hasMore required	boolean Признак наличия дополнительных элементов по переданным фильтрам. Параметр <code>hasMore</code> в ответе API указывает, есть ли ещё данные для выдачи. Если количество возвращённых элементов равно <code>limit</code> и в базе остаются данные, <code>hasMore</code> устанавливается в <code>true</code> , что сигнализирует клиенту о необходимости



говорит клиенту, что дополнительных запросов делать не нужно.

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> 404

Запрашиваемый ресурс не найден

> default

Внутренняя ошибка сервера

Response samples

200

400

401

404

default

Content type

application/json

Copy Expand all Collapse all



```
- {
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "type": "rabbitMqExternal",
  "title": "some title",
  - "info": {
    "dsn": "amqp://localhost:5672/vhost",
    "exchange": "someEventsExchange",
    "key": "someEventsKey"
  }
},
- {
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "type": "kafkaExternal",
  "title": "some title",
  - "info": {
    "uri": "https://example.com/",
    "topic": "someImagesInputTopic"
  }
},
- {
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "type": "rabbitMqInternal",
  "title": "some title",
  - "info": {
    "dsn": "amqp://localhost:5672/vhost",
    "queue": "someQueue"
  }
},
- {
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "type": "kafkaInternal",
  "title": "some title",
  - "info": {
    "uri": "https://example.com/",
    "topic": "someImagesInputTopic"
  }
}
],
"hasMore": false
}
```

Б.5 Получение доступа к внутренним брокерам сообщений

GET /api/v1/broker-access



Получение доступа к внутренним брокерам сообщений



Полученные учётные данные применяются для следующих сценариев:

- загрузка изображений для последующей обработки во внутренние источники;
- получение событий и результатов обработки изображений из внутренних экспортеров;
- асинхронная передача экспертных оценок по результатам обработки изображений.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

Responses

✓ 200

Успешное получение учетных данных

RESPONSE SCHEMA: application/json

credentials >
required

object

Данные УЗ для подключения к брокерам Подсистемы ИИПМ

> 401

Ошибка авторизации

> default

Внутренняя ошибка сервера

Response samples

200

401

default

Content type

application/json

Copy Expand all Collapse all

```
{
  - "credentials": {
    "username": "broker_user",
    "password": "broker_pass",
    "consumerGroupTemplate": "managed-user-***"
  }
}
```

Б.6 Обновление источника на стороне Пользователя



Обновление информации об источнике данных

Поддерживаемые типы источников для обновления:

1. **rabbitMqExternal** — очередь RabbitMQ, предоставляемая пользователями.
2. **kafkaExternal** — очередь Kafka, предоставляемая пользователями.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

PATH PARAMETERS

id	string <uuid>
required	Уникальный Системный идентификатор источника данных

REQUEST BODY SCHEMA: application/json

required

One of rabbitMqExternalSourceUpdate kafkaExternalSourceUpdate

title	string
	Новое название источника данных. Если не указано, название остаётся без изменений.

info >	object
--------	--------

Responses

✓ 200

Источник успешно обновлён

RESPONSE SCHEMA: application/json

object
Общий формат успешного ответа

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> 404

Запрашиваемый ресурс не найден

> default

Внутренняя ошибка сервера



Request samples

Payload

Content type

application/json

Example

rabbitMqExternalSourceUpdate

Copy Expand all Collapse all

```
{
  "title": "string",
  - "info": {
    "dsn": "amqp://rmq1.example.com:5672,rmq2.example.com:5672,rmq3.example.com:5672/vhost",
    "queue": "someImagesInputQueue",
    - "auth": {
      "username": "rabbitUser",
      "password": "rabbitPass"
    }
  }
}
```

Response samples

200

400

401

404

default

Content type

application/json

Copy

```
{ }
```

Б.7 Обновление экспортера на стороне Пользователя

PATCH /api/v1/exporters/{id}



Обновление существующего экспортера данных по Системному идентификатору

Обновлению подлежат только следующие типы экспортёров:

1. **rabbitMqExternal** — очередь RabbitMQ пользователя
2. **kafkaExternal** — очередь Kafka пользователя
3. **webhook** — отправка HTTP POST запроса с результатами обработки в формате `application/json` на webhook endpoint, указанный в параметрах экспортера.

AUTHORIZATIONS: >

(bearerAuth and externalLogin)



id
required

string <uuid>
Уникальный Системный идентификатор экспортера

REQUEST BODY SCHEMA: application/json
required

One of

RabbitMqExternalExporterUpdate

kafkaExternalExporterUpdate

WebhookExporterUpdate

title

string
Обновляемое название экспортера.

info >

object
Объект с обновляемыми параметрами подключения к RabbitMQ.

Responses

✓ 200

Экспортер успешно обновлён

RESPONSE SCHEMA: application/json

object
Общий формат успешного ответа

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> 404

Запрашиваемый ресурс не найден

> default

Внутренняя ошибка сервера

Request samples

Payload

Content type
application/json

Example
RabbitMqExternalExporterUpdate



```
- "info": {
  "dsn": "amqp://localhost:5672/vhost",
  "exchange": "someEventsExchange",
  "key": "someEventsKey",
- "auth": {
  "username": "rabbitUser",
  "password": "rabbitPass"
}
}
```

Response samples

200

400

401

404

default

Content type

application/json

Copy

{ }

В Управление задачами

Методы создания, редактирования и просмотра задач обработки.

В.1 Получение списка доступных детекторов

GET /api/v1/detectors



Получение списка доступных детекторов

Перечень детекторов, доступных Пользователю, определяется Администратором Модуля на пользовательском портале Подсистемы ИИПМ.

Доступ формируется на основании назначенных прав доступа, закреплённых за Пользователем. В составе каждого детектора возвращается перечень классов данного детектора.

Метод возвращает **только те детекторы**, к которым у Пользователя есть разрешённый доступ

AUTHORIZATIONS: >

(bearerAuth and externalLogin)

Responses



Успешный ответ со списком доступных детекторов

RESPONSE SCHEMA: application/json

detectors >
required

Array of objects

Массив детекторов, доступных Пользователю

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> 404

Запрашиваемый ресурс не найден

> default

Внутренняя ошибка сервера

Response samples

200

400

401

404

default

Content type

application/json

Copy Expand all Collapse all

```
{
  - "detectors": [
    - {
      "id": 500,
      "name": "Крупное лицо",
      "processingMode": "frame",
    - "classes": [
      - {
        "id": 5001,
        "name": "Столб"
      }
    ]
  }
]
```



В.2 Создание задачи на обработку изображений

POST /api/v2/tasks



Метод предназначен для создания задачи на обработку изображений с заданными параметрами. Поддерживается работа с внутренними и внешними источниками и экспортёрами.

В параметре `sources` необходимо передавать массив объектов, каждый из которых должен соответствовать одному из следующих вариантов:

1. Создание задачи с передачей полной информации об источнике

(без необходимости предварительной регистрации в Системе).

В этом случае объект источника должен соответствовать одной из следующих схем:

- `rabbitMqInternalSourceCreate` для источника типа **rabbitMqInternal**
- `kafkaInternalSourceCreate` для источника типа **kafkaInternal**
- `rabbitMqExternalSourceCreate` для источника типа **rabbitMqExternal**
- `kafkaExternalSourceCreate` для источника типа **kafkaExternal**
- `scheduleSourceCreate` для источника типа **schedule**
- `cameraTaskCreate` для источника типа **camera**
- `cameraGroupTaskCreate` для источника типа **cameraGroup**
- `cameraDeviceTaskCreate` для источника типа **cameraDevice**
- `cameraGroupDeviceSourceCreate` для источника типа **cameraDeviceGroup**
- `deviceSourceBase` для источника типа **device**

2. Использование ранее зарегистрированного источника

(за исключением источника типа `cameraGroup`),

созданного методом **Б.1 Создание нового источника данных**.

В этом случае объект источника должен соответствовать схеме `InternalSource` и содержать поля:

- `type` — тип источника;
- `id` — системный идентификатор источника.

Разрешено создание задачи только по следующим типам источников данных:

- `schedule`
- `satellite`
- `rabbitMqInternal`
- `kafkaInternal`
- `rabbitMqExternal`
- `kafkaExternal`
- `camera`
- `archive`
- `panoramic`
- `gin`
- `device`
- `cameraGroup`
- `cameraDevice`
- `cameraDeviceGroup`
- `httpImage`



применяется к участку изображения, ограниченному координатами соответствующего объекта, если флаг `blur` установлен в значение `true` хотя бы на одном из уровней — детектора или класса.

При синхронном режиме `exporters` не указываются — результаты доступны по методу "**Д.1 Получение результатов анализа**".

Параметр `grouping` определяет способ передачи детектированных объектов/нарушений:

- при значении `true` все объекты/нарушения, детектированные на одном изображении, передаются в очередь событий в составе одного сообщения;
- при значении `false` каждый детектированный объект/нарушение передаётся в очередь событий отдельным сообщением сразу после его детектирования.

⚠ В разработке Необходимость включения в результат сведений о детекторах и классах детекторов, объекты/нарушения по которым не были обнаружены на изображении, определяется параметрами `detectors[].empty` и `detectors[].classesIncluded[].empty` соответственно.

Для источников типов `camera`, `cameraGroup` и `InternalCameraGroupSource` дополнительно требуется передача параметра `executionMode`, определяющего режим выполнения обработки видеопотока.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

REQUEST BODY SCHEMA: application/json

required

One of

TaskCreateSimple

TaskCreateVideo

title
required

string
Наименование задачи

detectors >
required

Array of objects `non-empty`
⚠ В разработке Список детекторов, участвующих в выполнении задачи, с индивидуальными настройками для каждого.

exporters >

Array of rabbitMqExternalExporterCreate (object) or kafkaExternalExporterCreate (object) or RabbitMqInternalExporterCreate (object) or KafkaInternalExporterCreate (object) or webhookExporterCreate (object) or InternalExporter (object)

timeout

integer `[ 5 .. 60 ]`
Максимальное время ожидания результата обработки в секундах. Время старта - получение изображения контейнером ИНС, время завершения - регистрация события в Базе данных Системы. Допустимые значения: от 5 до 60.

Особенности для источника типа `httpImage` :

- Для данного типа источника обработка выполняется в отдельном синхронном HTTP-методе (Г.1), вызываемом Клиентом напрямую.
- Таймаут, установленный в задаче, **не используется** при обработке изображений, поскольку ожидание результата регулируется таймаутом HTTP-запроса на стороне Клиента.
- При создании задачи с типом источника `httpImage` Система автоматически устанавливает таймаут серверного выполнения запроса Г.1 по умолчанию — **120 секунд**, независимо от указанного в запросе значения.

grouping

boolean
Default: `false`



передается в отдельном сообщении).

`sources` >
`required`

Array of `rabbitMqInternalSourceCreate` (object) or `kafkaInternalSourceCreate` (object) or `rabbitMqExternalSourceCreate` (object) or `kafkaExternalSourceCreate` (object) or `scheduleSourceCreate` (object) or `deviceSourceBase` (object) or `InternalSource` (object) or `cameraDeviceTaskCreate` (object) or `cameraGroupDeviceSourceCreate` (object) `non-empty`

Responses

✓ 200

Успешное создание задачи

RESPONSE SCHEMA: `application/json`

`task` >
`required`

object

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> 403

Недостаточно прав для доступа

> default

Внутренняя ошибка сервера

Request samples

Payload

Content type

`application/json`

Example

`TaskCreateSimple`

```
{  
  "title": "Обработка расписания Незаконная торговля",
```

Copy Expand all Collapse all



```
"id": 500,
"conf": 0.5,
"blur": false,
"drawBoxes": true,
"empty": false,
- "classesIncluded": [
  - {
    "id": 5001,
    "conf": 0.5,
    "blur": true,
    "drawBoxes": true,
    "empty": false
  }
]
},
],
- "exporters": [
  - {
    "type": "rabbitMqExternal",
    "title": "some title",
    - "info": {
      "dsn": "amqp://localhost:5672/vhost",
      "exchange": "someEventsExchange",
      "key": "someEventsKey",
      - "auth": {
        "username": "rabbitUser",
        "password": "rabbitPass"
      }
    }
  }
],
"timeout": 30,
"grouping": true,
- "sources": [
  - {
    "type": "rabbitMqInternal",
    "title": "string",
    - "info": {
      "format": "binary"
    }
  }
]
}
```

Response samples

[200](#)[400](#)[401](#)[403](#)[default](#)



Copy Expand all Collapse all





```

"title": "Обработка расписания Незаконная торговля",
"createdAt": "2023-10-04T15:43:42.574119+03:00",
"lastUpdatedAt": "2023-10-04T15:43:42.574119+03:00",
- "detectors": [
  - {
    "id": 500,
    "conf": 0.5,
    "blur": false,
    "drawBboxes": true,
    "empty": false,
    - "classesIncluded": [
      - {
        "id": 5001,
        "conf": 0.5,
        "blur": true,
        "drawBboxes": true,
        "empty": false
      }
    ]
  },
- "sources": [
  - {
    "type": "rabbitMqExternal",
    "title": "Внешняя очередь входящих изображений",
    - "info": {
      "format": "url",
      "dsn": "amqp://rmq1.example.com:5672,rmq2.example.com:5672,rmq3.example.com:5672/vhost",
      "queue": "externalImagesQueue"
    },
    "id": "b7d25b6e-9c4a-4e72-83d5-3d72c59f94c1"
  },
- "exporters": [
  - {
    "type": "rabbitMqExternal",
    "title": "some title",
    - "info": {
      "dsn": "amqp://localhost:5672/vhost",
      "exchange": "someEventsExchange",
      "key": "someEventsKey"
    },
    "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08"
  },
- "status": "InProgress",
"timeout": 30,
"grouping": true,
- "executionMode": {
  "mode": "realtime"

```



В.3 Получение списка задач

GET /api/v2/tasks



Метод возвращает список задач, созданных пользователем, а также задач, доступ к которым был предоставлен Администратором Подсистемы ИИПМ.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

QUERY PARAMETERS

limit	integer >= 1 Default: 100 Максимальное количество возвращаемых элементов в одном ответе. Значение по умолчанию — 100.
offset	integer >= 0 Default: 0 Смещение относительно начала выборки. Используется для постраничной навигации. Значение по умолчанию — 0.
status	Array of strings Items Enum: inProgress paused cancelled Статус активности задачи
order	string Default: desc Enum: asc desc Сортировка по времени создания задачи (createdAt)
detectorId	Array of integers Фильтр по списку детекторов/классов. Параметр принимает значения: - идентификаторы детекторов; - идентификаторы классов детекторов.
type	Array of strings Items Enum: rabbitMqInternal kafkaInternal rabbitMqExternal kafkaExternal schedule satellite gin panoramic camera archive device cameraGroup cameraDevice cameraDeviceGroup httpImage Фильтр по типу источника данных
sourceId	Array of strings <uuid> [items <uuid>] Фильтр по системному идентификатору источника данных
scheduleId	Array of strings <uuid> [items <uuid>] Фильтр по сквозному идентификатору расписания ЕЦХД
cameraId	Array of strings <uuid> [items <uuid>]



ИИПМ



deviceId	Array of strings <uuid> [items <uuid >] Фильтр по внешнему источнику данных ЕЦХД (сквозной идентификатор внешнего источника данных ЕЦХД)
exporterId	Array of strings <uuid> [items <uuid >] Фильтр по системному идентификатору экспортера
taskId	Array of strings <uuid> [items <uuid >] Фильтр по системному идентификатору задачи

Responses

✓ 200

OK

RESPONSE SCHEMA: application/json

tasks >
required

object

hasMore
required

boolean

Признак наличия дополнительных элементов по переданным фильтрам. Параметр `hasMore` в ответе API указывает, есть ли ещё данные для выдачи. Если количество возвращённых элементов равно `limit` и в базе остаются данные, `hasMore` устанавливается в `true`, что сигнализирует клиенту о необходимости сделать следующий запрос с увеличенным `offset`. Если же возвращено меньше, чем `limit`, значит, данные закончились и `hasMore` будет `false`, что говорит клиенту, что дополнительных запросов делать не нужно.

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> 404

Запрашиваемый ресурс не найден

> default

Внутренняя ошибка сервера

Response samples

200

400

401

404

default



Copy Expand all Collapse all





```

"title": "Обработка расписания Незаконная торговля",
"createdAt": "2023-10-04T15:43:42.574119+03:00",
"lastUpdatedAt": "2023-10-04T15:43:42.574119+03:00",
- "detectors": [
  - {
    "id": 500,
    "conf": 0.5,
    "blur": false,
    "drawBboxes": true,
    "empty": false,
    - "classesIncluded": [
      - {
        "id": 5001,
        "conf": 0.5,
        "blur": true,
        "drawBboxes": true,
        "empty": false
      }
    ]
  },
- "sources": [
  - {
    "type": "rabbitMqExternal",
    "title": "Внешняя очередь входящих изображений",
    - "info": {
      "format": "url",
      "dsn": "amqp://rmq1.example.com:5672,rmq2.example.com:5672,rmq3.example.com:5672/vhost",
      "queue": "externalImagesQueue"
    },
    "id": "b7d25b6e-9c4a-4e72-83d5-3d72c59f94c1"
  },
- "exporters": [
  - {
    "type": "rabbitMqExternal",
    "title": "some title",
    - "info": {
      "dsn": "amqp://localhost:5672/vhost",
      "exchange": "someEventsExchange",
      "key": "someEventsKey"
    },
    "id": "497f6eca-6276-4993-bfeb-53cbbba6f08"
  },
- "status": "InProgress",
"timeout": 30,
"grouping": true,
- "executionMode": {
  "mode": "realtime"
}

```

`"hasMore": true``}`

В.4 Обновление задачи

PATCH `/api/v1/tasks/{id}`

Подсистема ИИПМ должна обеспечивать централизованное управление жизненным циклом задач, включая их **создание, активацию, деактивацию и удаление**.

Задачи могут находиться в следующих статусах:

- `cancelled` — задача **отменена** (не может быть возобновлена после присвоения данного статуса);
- `paused` — задача **деактивирована** (может быть переведена в статус `inProgress` или `cancelled`);
- `inProgress` — задача **активна** (может быть переведена в статус `paused` или `cancelled`).

Метод предназначен для редактирования параметров существующей задачи.

Поддерживаются следующие операции:

- изменение статуса задачи (`status`);
- добавление новых детекторов/классов (`detectorsToAdd`);
- удаление детекторов/классов (`detectorIdsToRemove`);
- полная замена списка детекторов/классов (`detectors`);
- добавление новых источников данных (`sourcesToAdd`);
- удаление источников данных (`sourcesToRemove`);
- полная замена списка источников данных (`sources`).

Ограничения на использование параметров:

Параметр `detectors` (полная замена списка детекторов/классов) не допускается к передаче в одном запросе совместно с параметрами `detectorsToAdd` и/или `detectorIdsToRemove`.

Параметр `sources` (полная замена списка источников данных) не допускается к передаче в одном запросе совместно с параметрами `sourcesToAdd` и/или `sourcesToRemove`.

AUTHORIZATIONS: > *(bearerAuth and externalLogin)*

PATH PARAMETERS

`id`
`required`

string <uuid>
Идентификатор задачи

REQUEST BODY SCHEMA: application/json
`required`

Обновление задачи - статус, источники и/или детекторы

`non-empty`

`status`

string
Enum: `"paused"` `"inProgress"` `"cancelled"`



detectorsToAdd >

Array of objects `non-empty`

⚠ **В разработке** Список детекторов для добавления.

detectorIdsToRemove

Array of integers

Список детекторов для удаления

detectors >

Array of objects `non-empty`

⚠ **В разработке** Полный список детекторов для полной замены.

executionMode >

RealtimeExecution (object) or CronExecution (object) or SuperSnapshotExecution (object)

⚠ **Обновление executionMode доступно, но проверка ограничений в разработке**

При изменении параметра `executionMode` и/или состава детекторов должны соблюдаться следующие ограничения

1. Если в задаче присутствует хотя бы один детектор, зарегистрированный с параметром `processingMode = stream`, то
 - о значение `executionMode.mode` должно быть равно `realtime`;
 - о изменение `executionMode` на `cron` или `superSnapshotService` не допускается.
2. Если в задаче установлено значение `executionMode.mode`, отличное от `realtime`, добавление детектора с `processingMode = stream` не допускается.
3. В случае, если задача уже содержит детектор с `processingMode = stream`, изменение `executionMode` допускается только при условии одновременного удаления всех таких детекторов из задачи.
4. В случае, если при обновлении задачи одновременно
 - о сохраняется хотя бы один детектор с `processingMode = stream`; и
 - о передаётся значение `executionMode.mode`, отличное от `realtime`, запрос должен быть отклонён с ошибкой валидации

sourcesToAdd >

Array of scheduleSourceCreate (object) or cameraTaskCreate (object) or cameraGroupTaskCreate (object) or cameraDeviceTaskCreate (object) or cameraGroupDeviceSourceCreate (object) or deviceSourceBase (object) or InternalSource (object)

Источники данных, которые необходимо добавить

sourcesToRemove >

Array of scheduleSourceDelete (object) or cameraTaskDelete (object) or cameraDeviceTaskDelete (object) or deviceSourceBase (object) or InternalSource (object)

Источники данных, которые необходимо удалить

sources >

Array of scheduleSourceCreate (object) or cameraTaskCreate (object) or cameraGroupTaskCreate (object) or cameraDeviceTaskCreate (object) or cameraGroupDeviceSourceCreate (object) or deviceSourceBase (object) or InternalSource (object)

Полный список источников данных для замены

Responses



Успешное обновление статуса

RESPONSE SCHEMA: application/json

object

Общий формат успешного ответа

> 400

Некорректный запрос или логическая ошибка

> 401

Ошибка авторизации

> 403

Доступ запрещён – пользователь не является автором задачи

> default

Внутренняя ошибка сервера

Request samples

Payload

Content type

application/json

Copy Expand all Collapse all

```
{
  "status": "inProgress",
  - "detectorsToAdd": [
    - {
      "id": 500,
      "conf": 0.5,
      "blur": false,
      "drawBboxes": true,
      "empty": false,
    - "classesIncluded": [
      - {
        "id": 5001,
        "conf": 0.5,
        "blur": true,
        "drawBboxes": true,
        "empty": false
      }
    ]
  }
},
```



```
    ],
  - "detectors": [
    - {
      "id": 500,
      "conf": 0.5,
      "blur": false,
      "drawBboxes": true,
      "empty": false,
    - "classesIncluded": [
      - {
        "id": 5001,
        "conf": 0.5,
        "blur": true,
        "drawBboxes": true,
        "empty": false
      }
    ]
  }
],
- "executionMode": {
  "mode": "realtime"
},
- "sourcesToAdd": [
  - {
    "type": "schedule",
    "title": "string",
    - "info": {
      "scheduleId": "b7b4f318-018f-4d71-ac1a-f61e4bfaefbe"
    }
  }
],
- "sourcesToRemove": [
  - {
    "type": "schedule",
    "title": "string",
    - "info": {
      "scheduleId": "b7b4f318-018f-4d71-ac1a-f61e4bfaefbe"
    }
  }
],
- "sources": [
  - {
    "type": "schedule",
    "title": "string",
    - "info": {
      "scheduleId": "b7b4f318-018f-4d71-ac1a-f61e4bfaefbe"
    }
  }
]
}
```



response samples

- 200
- 400
- 401
- 403
- default

Content type
application/json

{ }

Copy

В.5 Получение статуса обработки последней итерации по задаче

GET /api/v1/tasks/{id}/last-iteration-status



Возвращает информацию о текущем статусе обработки последней итерации по каждому источнику данных в рамках указанной задачи.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

PATH PARAMETERS

id	string <uuid>
required	Идентификатор задачи

Responses

> 200

Статус итерации по источникам

RESPONSE SCHEMA: application/json

statuses >	Array of objects
required	Список статусов по каждому источнику данных

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> 404

Запрашиваемый ресурс не найден



Response samples

- 200
- 400
- 401
- 404
- default

Content type
application/json

Copy Expand all Collapse all

```
{
  - "statuses": [
    - {
      "sourceId": "1ac54723-aab0-4a75-b12c-f39a81d34fff",
      - "detectorsProgress": {
        "501": 4.5,
        "601": 100
      },
      "status": "InProgress",
      "started": "2025-05-04T15:43:42.574119+03:00",
      "approximateDuration": 5
    }
  ]
}
```

V.6 Настройка зоны интереса для камеры в контексте задачи

PUT /api/v1/tasks/{taskId}/camera-groups/{groupId}/cameras/{cameraId}/zone-of-interest

Установить зону интереса для конкретной камеры, входящей в указанную группу камер. Привязка зоны выполняется к задаче (taskId), группе камер (groupId) и сквозному идентификатору камеры (cameraId). В случае передачи null для схемы ZoneOfInterest система должна удалить ранее установленную зону интереса для указанной камеры в контексте соответствующей задачи (taskId) и группы камер (groupId).

AUTHORIZATIONS: > (bearerAuth and externalLogin)

PATH PARAMETERS

taskId required	string <uuid> Уникальный системный идентификатор задачи (UUID).
groupId required	string <uuid> Уникальный системный идентификатор группы источников (UUID). Допустимые типы источников: cameraGroup , cameraDeviceGroup .



сквозной идентификатор камеры в системе ЕЦХД (она должна быть у источника). Должен соответствовать UUID камеры ЕЦХД.

REQUEST BODY SCHEMA: application/json

required

zoneOfInterest >	object
	Зона интереса, определяемая пользователем для ограничения области детектирования. Содержит список координат, формирующих многоугольник, и флаг, определяющий режим поиска объектов (внутри или вне зоны).

Responses

✓ 200

Успешное выполнение операции.

- При передаче объекта `zoneOfInterest` — зона(ы) интереса установлена.
- При передаче `zoneOfInterest: null` — зона(ы) интереса удалена.

RESPONSE SCHEMA: application/json

taskId required	string <uuid> Идентификатор задачи
groupId required	string <uuid> Идентификатор группы источников
cameraId required	string <uuid> Сквозной идентификатор камеры (info.camerald)
zoneOfInterest >	object
	Зона интереса, определяемая пользователем для ограничения области детектирования. Содержит список координат, формирующих многоугольник, и флаг, определяющий режим поиска объектов (внутри или вне зоны).

> 400

Ошибка валидации данных

> 401

Ошибка авторизации

> 404

Задача, группа или камера не найдена

> default

Внутренняя ошибка сервера



Request samples

Payload

Content type

application/json

Copy

```
{
  "zoneOfInterest": null
}
```

Response samples

200

400

401

404

default

Content type

application/json

Copy

Expand all

Collapse all

```
{
  "taskId": "e6e9d88a-9b63-468a-aec3-b7a11de27af8",
  "groupId": "eb54e96e-21b8-4f54-9cd4-80fccbd06f55",
  "cameraId": "cbc3283e-6217-4855-9934-148717cbeefe",
  - "zoneOfInterest": {
    - "zones": [
      - [
        - {
          "x": 250,
          "y": 250
        },
        - {
          "x": 250,
          "y": 250
        },
        - {
          "x": 250,
          "y": 250
        }
      ]
    ],
    "rule": "inside"
  }
}
```

В.7 Получение зон интереса для всех камер группы



Получить список камер в указанной группе источников в рамках задачи вместе с их зонами интереса. Если зона интереса для камеры не задана, поле `zoneOfInterest` имеет значение `null`.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

PATH PARAMETERS

taskId
required

string <uuid>
Уникальный системный идентификатор задачи (UUID).

groupId
required

string <uuid>
Уникальный системный идентификатор группы источников (UUID). Допустимые типы групп: `cameraGroup`, `cameraDeviceGroup`.

QUERY PARAMETERS

limit

integer `>= 1`
Default: `100`

Максимальное количество возвращаемых элементов в одном ответе. Значение по умолчанию — 100.

offset

integer `>= 0`
Default: `0`

Смещение относительно начала выборки. Используется для постраничной навигации. Значение по умолчанию — 0.

Responses

✓ 200

Список камер с зонами интереса (может быть пустым)

RESPONSE SCHEMA: application/json

cameras >
required

Array of objects `>= 0 items`
Список камер с зонами интереса

hasMore
required

boolean
Признак наличия дополнительных элементов по переданным фильтрам. Параметр `hasMore` в ответе API указывает, есть ли ещё данные для выдачи. Если количество возвращённых элементов равно `limit` и в базе остаются данные, `hasMore` устанавливается в `true`, что сигнализирует клиенту о необходимости сделать следующий запрос с увеличенным `offset`. Если же возвращено меньше, чем `limit`, значит, данные закончились и `hasMore` будет `false`, что говорит клиенту, что дополнительных запросов делать не нужно.

> 400

Некорректный запрос — отсутствуют обязательные параметры или неверный формат данных



Ошибка авторизации

> 404

Задача или группа источников не найдена

> default

Внутренняя ошибка сервера

Response samples

200

400

401

404

default

Content type

application/json

Copy Expand all Collapse all

```
{
  - "cameras": [
    - {
      "cameraId": "cbc3283e-6217-4855-9934-148717cbeefe",
      - "zoneOfInterest": {
        - "zones": [
          - [
            - {
              "x": 250,
              "y": 250
            },
            - {
              "x": 250,
              "y": 250
            },
            - {
              "x": 250,
              "y": 250
            }
          ]
        ],
        "rule": "inside"
      }
    }
  ],
  "hasMore": true
}
```



Г Отправка изображений

Модели данных, связанные с вариантами получения изображений для ИНС анализа на стороне ИИПМ. При отправке изображений посредством брокеров сообщений в одном сообщении передается только одно изображение.

Содержит идентификатор изображения (`originScreenshotId`) в заголовке. Данный идентификатор является сквозным и в дальнейшем используется ИИПМ при отправке событий. Идентификатор `originScreenshotId` должен быть уникальным и представлен в формате GUID.

В теле сообщения передается само изображение в формате Binary. Дополнительно для каждого сообщения необходимо фиксировать время помещения в очередь (`enqueueTimestamp`). Для брокера сообщений Kafka значение назначается автоматически системой, для брокера сообщений RabbitMQ необходимо добавлять метку времени в заголовок сообщения.

KafkaBinaryImageMessage

any

Сообщение Kafka, содержащее изображение в бинарном виде.

- В заголовке сообщения (`headers`) передаётся параметр `originScreenshotId` — уникальный идентификатор изображения (UUID), используемый как сквозной идентификатор при отправке событий.
- В теле сообщения (`value`) передаются байты изображения **в бинарном формате**.

Copy

null

KafkaImageUrlMessage

any

Сообщение Kafka, содержащее ссылку на изображение.

- В заголовке сообщения (`headers`) передаётся параметр `originScreenshotId` — уникальный идентификатор изображения (UUID), используемый как сквозной идентификатор при отправке событий.



null

Copy

RabbitBinaryImageMessage

any

Сообщение RabbitMQ, содержащее изображение в бинарном виде.

- В заголовке сообщения (`headers`) передаются параметры:
 - `originScreenshotId` — уникальный идентификатор изображения (UUID), используемый как сквозной идентификатор при отправке событий;
 - `enqueueTimestamp` — время отправки сообщения в очередь, указывается в формате RFC3339 с часовым поясом UTC+3 и добавляется отправителем.
- В теле сообщения (`body`) передаются байты изображения **в бинарном формате**.

Copy

null

RabbitImageUrlMessage

any

Сообщение RabbitMQ, содержащее ссылку на изображение.

- В заголовке сообщения (`headers`) передаются параметры:
 - `originScreenshotId` — уникальный идентификатор изображения (UUID), используемый как сквозной идентификатор при отправке событий;
 - `enqueueTimestamp` — время отправки сообщения в очередь, указывается в формате RFC3339 с часовым поясом UTC+3 и добавляется отправителем.



Copy

null

Г.1 Загрузка изображения по источнику (моментальная обработка - синхронный режим)

POST /api/v1/sources/{id}/upload/sync

▼

Выполняет синхронную обработку изображения, переданного в запросе.

При получении изображения Система:

- определяет все задачи, связанные с указанным источником;
- выполняет обработку изображения в рамках этих задач;
- формирует результат обработки.

При формировании результата обработки задачи учитывается параметр `empty`, заданный в конфигурации детекторов (`DetectorItem`) и классов (`DetectorClassItem`). Формат ответа:

- При создании задачи с установкой у детектора или класса `empty = true` и отсутствии событий
Возвращается список идентификаторов детекторов/классов, для которых события не обнаружены.
- При создании задачи с установкой у детектора или класса `empty = false` и отсутствии событий
Возвращается код ответа 200 OK и пустое тело ответа, что означает отсутствие обнаруженных событий.
- При обнаружении событий Возвращается код ответа 200 OK и тело ответа в формате JSON, содержащее результаты обработки.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

PATH PARAMETERS

id
required string <uuid>
Идентификатор источника типа `httpImage`.

REQUEST BODY SCHEMA: multipart/form-data
required

image
required string <binary>
Бинарное изображение (JPEG).

originScreenshotId
required string <uuid>
Сквозной идентификатор изображения.



✓ 200

Успешная обработка изображения.

RESPONSE SCHEMA: application/json

One of

ImageEventsInfoAsync

События не обнаружены

image > required object

> 400

Ошибка обработки изображения.

> 401

Ошибка авторизации.

> 404

Указанный источник данных не найден.

> default

Внутренняя ошибка сервера при обработке изображения.

Request samples

Payload

Content type
multipart/form-data

Copy

```
{
  "originScreenshotId": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "image": "(binary JPEG)"
}
```

Response samples

200

400

401

404

default

Content type
application/json

Example
ImageEventsInfoAsync

Copy Expand all Collapse all





```
"createdAt": "2023-10-04T15:43:42.574119+03:00",
"snapshottedAt": "2023-10-04T15:43:42.574119+03:00",
"url": "http://example.com",
- "camera": {
  "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
  "pan": 0,
  "tilt": 0,
  "zoom": 0,
  "height": 0,
  "viewId": "string",
  "cameraTypeOrg": "string"
},
- "source": {
  "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
  "title": "string",
  "type": "rabbitMqInternal",
  "scheduleId": "b7b4f318-018f-4d71-ac1a-f61e4bfaefbe",
  "deviceId": "4de4adb9-21ee-47e3-aeb4-8cf8ed6c109a"
},
- "blurredImages": [
  - {
    "taskId": "e6e9d88a-9b63-468a-aec3-b7a11de27af8",
    "url": "http://example.com"
  }
],
- "processingErrors": [
  - {
    "detectorId": 0,
    "error": "err image download"
  }
],
- "emptyDetectorIds": [
  0
],
- "matchingImage": {
  "originScreenshotId": "d4df0173-ce22-4ef2-a2f5-c43c2d7cdba7",
  "createdAt": "2023-10-04T15:43:42.574119+03:00",
  "snapshottedAt": "2023-10-04T15:43:42.574119+03:00",
  "url": "http://example.com",
  - "camera": {
    "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
    "pan": 0,
    "tilt": 0,
    "zoom": 0,
    "height": 0,
    "viewId": "string",
    "cameraTypeOrg": "string"
  },
  - "source": {
    "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
```



```
"scheduleId": "b7b4f318-018f-4d71-ac1a-f61e4bfaefbe",
"deviceId": "4de4adb9-21ee-47e3-aeb4-8cf8ed6c109a"
}
},
- "geoCoordinates": {
  "latitude": 55.7558,
  "longitude": 37.6173,
  "speed": 100,
  "direction": 90,
  "valid": true
},
- "events": [
  - {
    "id": "497f6eca-6276-4993-bfeb-53cbbba6f08",
    "originEventId": "string",
    "objectId": "string",
    "isTrackingStart": true,
    "isTrackingEnd": true,
    "createdAt": "2023-10-04T15:43:42.574119+03:00",
    "originCreatedAt": "2023-10-04T15:43:40.120000+03:00",
    "detectorId": 601,
  - "taskIds": [
    "3fa85f64-5717-4562-b3fc-2c963f66afa6"
  ],
  - "objectDetections": [
    - {
      "id": "497f6eca-6276-4993-bfeb-53cbbba6f08",
      "parentObjectDetectionId": "49e7658b-7404-416d-a273-94c63f44de51",
      - "points": [
        - [
          0,
          0
        ],
        - [
          0,
          0
        ],
        - [
          0,
          0
        ],
        - [
          0,
          0
        ]
      ],
      "recognizedText": "string",
      "confidence": 0,
      "latitude": 55.9556,
      "longitude": 37.4536,
```



```
    },  
    - "class": {  
      "confidence": 0  
    },  
    "isLastEvent": true  
  }  
]  
}  
}
```

Г.2 Загрузка изображения по источнику (асинхронный режим)

POST /api/v1/sources/{id}/upload/async



Выполняет асинхронную обработку изображения, переданного в запросе.

При получении изображения Система:

- определяет все задачи, связанные с указанным источником;
- инициирует их выполнение в асинхронном режиме;
- сохраняет результаты обработки для последующего получения.

Результат обработки изображения доступен методом [Д.1 Получение результатов анализа GET api/v1/results](#) после завершения обработки.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

PATH PARAMETERS

id required	string <uuid> Идентификатор источника типа httpImage .
----------------	---

REQUEST BODY SCHEMA: multipart/form-data ▼
required

image required	string <binary> Бинарное изображение (JPEG).
-------------------	---

originScreenshotId required	string <uuid> Сквозной идентификатор изображения.
--------------------------------	--

Responses

✓ 200

Успешная обработка изображения.



object

Общий формат успешного ответа

> 400

Ошибка обработки изображения.

> 401

Ошибка авторизации.

> 404

Указанный источник данных не найден.

> default

Внутренняя ошибка сервера при обработке изображения.

Request samples

Payload

Content type

multipart/form-data

```
{
  "originScreenshotId": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "image": "(binary JPEG)"
}
```

Copy

Response samples

200

400

401

404

default

Content type

application/json

```
{ }
```

Copy

Д Получение результатов обработки



ImageEventsInfoAsync

image >
required

object

Copy Expand all Collapse all





```
"createdAt": "2023-10-04T15:43:42.574119+03:00",
"snapshottedAt": "2023-10-04T15:43:42.574119+03:00",
"url": "http://example.com",
- "camera": {
  "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
  "pan": 0,
  "tilt": 0,
  "zoom": 0,
  "height": 0,
  "viewId": "string",
  "cameraTypeOrg": "string"
},
- "source": {
  "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
  "title": "string",
  "type": "rabbitMqInternal",
  "scheduleId": "b7b4f318-018f-4d71-ac1a-f61e4bfaefbe",
  "deviceId": "4de4adb9-21ee-47e3-aeb4-8cf8ed6c109a"
},
- "blurredImages": [
  - {
    "taskId": "e6e9d88a-9b63-468a-aec3-b7a11de27af8",
    "url": "http://example.com"
  }
],
- "processingErrors": [
  - {
    "detectorId": 0,
    "error": "err image download"
  }
],
- "emptyDetectorIds": [
  0
],
- "matchingImage": {
  "originScreenshotId": "d4df0173-ce22-4ef2-a2f5-c43c2d7cdba7",
  "createdAt": "2023-10-04T15:43:42.574119+03:00",
  "snapshottedAt": "2023-10-04T15:43:42.574119+03:00",
  "url": "http://example.com",
  - "camera": {
    "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
    "pan": 0,
    "tilt": 0,
    "zoom": 0,
    "height": 0,
    "viewId": "string",
    "cameraTypeOrg": "string"
  },
  - "source": {
    "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
```



```

      "scheduleId": "b7b4f318-018f-4d71-ac1a-f61e4bfaefbe",
      "deviceId": "4de4adb9-21ee-47e3-aeb4-8cf8ed6c109a"
    }
  },
  - "geoCoordinates": {
    "latitude": 55.7558,
    "longitude": 37.6173,
    "speed": 100,
    "direction": 90,
    "valid": true
  },
  - "events": [
    - {
      "id": "497f6eca-6276-4993-bfeb-53cbbba6f08",
      "originEventId": "string",
      "objectId": "string",
      "isTrackingStart": true,
      "isTrackingEnd": true,
      "createdAt": "2023-10-04T15:43:42.574119+03:00",
      "originCreatedAt": "2023-10-04T15:43:40.120000+03:00",
      "detectorId": 601,
    - "taskIds": [
      "3fa85f64-5717-4562-b3fc-2c963f66afa6"
    ],
    - "objectDetections": [
      - {
        "id": "497f6eca-6276-4993-bfeb-53cbbba6f08",
        "parentObjectDetectionId": "49e7658b-7404-416d-a273-94c63f44de51",
        - "points": [
          - [
            0,
            0
          ],
          - [
            0,
            0
          ],
          - [
            0,
            0
          ],
          - [
            0,
            0
          ]
        ],
        "recognizedText": "string",
        "confidence": 0,
        "latitude": 55.9556,
        "longitude": 37.4536,

```



```
    },
    - "class": {
      "confidence": 0
    },
    "isLastEvent": true
  }
]
}
}
```

OksEventAttributes

oksYin	string УИН объекта капитального строительства (ОКС)
oksControlId	string Номер надзорного дела
oksFinance	string Тип финансирования ОКС (бюджетный/коммерческий)
oksOrg	string Застройщик
oksGroupOrg	string Группа компаний застройщика
oksGenpod	string Генеральный подрядчик
description	string Описание события

Copy

```
{
  "oksYin": "string",
  "oksControlId": "string",
  "oksFinance": "string",
  "oksOrg": "string",
  "oksGroupOrg": "string",
  "oksGenpod": "string",
  "description": "string"
```



ShelfOccupancyDetectionAttributes

extras >
required

object
Дополнительные атрибуты, связанные с анализом заполненности полок

Copy Expand all Collapse all

```
{
  - "extras": {
    "totalCount": 8,
    "totalArea": 100
  }
}
```

StoreVisitorDetectionAttributes

totalCount
required

integer
Общее количество посетителей

groups >
required

Array of objects
Список групп посетителей

Copy Expand all Collapse all

```
{
  "totalCount": 8,
```



```
      "groupCount": 3
    },
    - {
      "groupCount": 4
    }
  ]
}
```

SilhouetteDetectionAttributes

group_id	string Идентификатор силуэта
rotation_angle	integer Угол поворота силуэта
attributes	object Атрибуты силуэта (всегда пустой)

Copy Expand all Collapse all

```
{
  "group_id": "64375c5d",
  "rotation_angle": 15,
  "attributes": { }
}
```

FaceDetectionAttributes

id	string Уникальный идентификатор лица
----	---



Copy

```
{
  "id": "62ab559f2b760db3bc3dcd05-000"
}
```

VehicleDetectionAttributes

detection_score required	number Качество распознавания (берётся из качества номера)
bbox > required	object ВВох транспортного средства - его ограничивающий прямоугольник на картинке.
body > required	object Вид транспортного средства
design > required	object Дизайн транспортного средства
color > required	object Цвет транспортного средства
make > required	object Марка транспортного средства
model > required	object Модель транспортного средства
license_plate_country > required	object Страна номера
license_plate_number > required	object Номер транспортного средства
license_plate_region > required	object Регион номера
car_weight_types7 > required	object Категория транспортного средства по весу
car_categories > required	object Категория прав



ИИПМ



Copy Expand all Collapse all



```
- "bbox": {
  "left": 100,
  "top": 200,
  "right": 500,
  "bottom": 600
},
- "body": {
  "confidence": 0.95,
  "name": "taxi"
},
- "design": {
  "confidence": 0.87,
  "name": "suv"
},
- "color": {
  "confidence": 0.92,
  "name": "black"
},
- "make": {
  "confidence": 0.88,
  "name": "toyota"
},
- "model": {
  "confidence": 0.86,
  "name": "camry"
},
- "license_plate_country": {
  "confidence": 0.93,
  "name": "RU"
},
- "license_plate_number": {
  "name": "A978BC122RUS"
},
- "license_plate_region": {
  "confidence": 0.9,
  "name": "RUS"
},
- "car_weight_types7": {
  "confidence": 0.91,
  "name": "B_light"
},
- "car_categories": {
  "confidence": 0.89,
  "name": "B"
}
}
```



Д.1 Получение результатов анализа

GET /api/v1/results



Подсистема ИИПМ должна предусматривать получение событий по следующим типам Искусственных нейронных сетей (ИНС):

- 1. **Детекция** — процесс, при котором ИНС обнаруживает и локализует объекты на изображении, определяя их классы, уровень уверенности (confidence) и контуры. Контур каждого обнаруженного объекта задаётся последовательностью координатных точек (`points`), формирующих замкнутую область на изображении. Координаты задаются в формате [x, y], где началом отсчёта является верхний левый угол изображения.
- 2. **Распознавание** — процесс, при котором ИНС идентифицирует содержимое конкретного объекта, находящегося внутри выделенного `points` , предоставляя точные данные о его содержании. Например, распознавание текста внутри выделенного `bbox` .
- 3. **Классификация** — процесс, при котором ИНС анализирует изображение в целом и определяет его принадлежность к конкретному классу, не выделяя отдельные объекты. Классификация применяется ко всему изображению и отвечает на вопрос, что оно представляет в целом.
- 4. **Сегментация** — процесс, при котором ИНС разделяет изображение на области, соответствующие отдельным объектам, выделяя их точные контуры. Результатом работы модели сегментации являются маски или контуры, которые точно очерчивают каждый объект на изображении.

AUTHORIZATIONS: > (bearerAuth and externalLogin)

QUERY PARAMETERS

limit	integer [1 .. 3000] Количество элементов к выдаче (по умолчанию — 100, максимум — 3000)
offset	integer >= 0 Default: 0 Смещение относительно начала выборки. Используется для постраничной навигации. Значение по умолчанию — 0.
originScreenshotId	string <uuid> Фильтр по сквозному идентификатору изображения
createdAtFrom	string <date-time> Фильтр по дате и времени получения изображения Системой («от»)
createdAtTo	string <date-time> Фильтр по дате и времени получения изображения Системой («до»)
snapshottedAtFrom	string <date-time> Фильтр по дате и времени создания изображения («от»)
snapshottedAtTo	string <date-time> Фильтр по дате и времени создания изображения («до»)



Фильтр по идентификатору события	
eventCreatedAtFrom	string <date-time> Фильтр по дате и времени получения результатов от ИНС («от»)
eventCreatedAtTo	string <date-time> Фильтр по дате и времени получения результатов от ИНС («до»)
detectorId	Array of integers ⚠ В разработке Фильтр по списку детекторов/классов. Параметр принимает значения: - идентификаторы детекторов; - идентификаторы классов детекторов.
taskId	Array of strings <uuid> [items <uuid >] Фильтр по уникальному идентификатору задачи
cameraId	string <uuid> Фильтр по камере ЕЦХД (сквозной идентификатор камеры ЕЦХД)
scheduleId	string <uuid> Фильтр по расписанию ЕЦХД (сквозной идентификатор расписания ЕЦХД)
deviceId	string <uuid> Фильтр по внешнему источнику данных ЕЦХД (сквозной идентификатор внешнего источника данных ЕЦХД)
sourceId	Array of strings <uuid> [items <uuid >] Фильтр по Системному идентификатору источника данных
sourceType	Array of strings Items Enum: "rabbitMqInternal" "kafkaInternal" "rabbitMqExternal" "kafkaExternal" "schedule" "satellite" "gin" "panoramic" "camera" "archive" "device" "odh" "cameraGroup" "cameraDevice" "cameraDeviceGroup" "httpImage" Фильтр по типу источника данных
feedbackStatus	string Enum: "issue confirmed" "issue is absent" "issue was skipped" "empty screen" "ignored by expert" Фильтр по статусу результата экспертной оценки изображения. Используется для отбора изображений по итоговому статусу, установленному после отправки экспертной оценки.

Responses

✓ 200

Успешный ответ с результатами анализа

RESPONSE SCHEMA: application/json

images > required	Array of objects Информация об изображении
----------------------	--



признак наличия дополнительных элементов не передается фильтром.

Параметр `hasMore` в ответе API указывает, есть ли ещё данные для выдачи. Если количество возвращённых элементов равно `limit` и в базе остаются данные, `hasMore` устанавливается в `true`, что сигнализирует клиенту о необходимости сделать следующий запрос с увеличенным `offset`. Если же возвращено меньше, чем `limit`, значит, данные закончились и `hasMore` будет `false`, что говорит клиенту, что дополнительных запросов делать не нужно.

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> 403

Ошибка прав доступа

> default

Внутренняя ошибка сервера

Response samples

200**400****401****403****default**

Content type

application/json

Copy Expand all Collapse all





```
"originScreenshotId": "d4df0173-ce22-4ef2-a2f5-c43c2d7cdba7",
"createdAt": "2023-10-04T15:43:42.574119+03:00",
"snapshottedAt": "2023-10-04T15:43:42.574119+03:00",
"url": "http://example.com",
- "camera": {
  "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
  "pan": 0,
  "tilt": 0,
  "zoom": 0,
  "height": 0,
  "viewId": "string",
  "cameraTypeOrg": "string"
},
- "source": {
  "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
  "title": "string",
  "type": "rabbitMqInternal",
  "scheduleId": "b7b4f318-018f-4d71-ac1a-f61e4bfaefbe",
  "deviceId": "4de4adb9-21ee-47e3-aeb4-8cf8ed6c109a"
},
- "blurredImages": [
  - {
    "taskId": "e6e9d88a-9b63-468a-aec3-b7a11de27af8",
    "url": "http://example.com"
  }
],
- "processingErrors": [
  - {
    "detectorId": 0,
    "error": "err image download"
  }
],
- "emptyDetectorIds": [
  0
],
- "matchingImage": {
  "originScreenshotId": "d4df0173-ce22-4ef2-a2f5-c43c2d7cdba7",
  "createdAt": "2023-10-04T15:43:42.574119+03:00",
  "snapshottedAt": "2023-10-04T15:43:42.574119+03:00",
  "url": "http://example.com",
  - "camera": {
    "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
    "pan": 0,
    "tilt": 0,
    "zoom": 0,
    "height": 0,
    "viewId": "string",
    "cameraTypeOrg": "string"
  },
  - "source": {
```



```
      "type": "rabbitMqInternal",
      "scheduleId": "b7b4f318-018f-4d71-ac1a-f61e4bfaefbe",
      "deviceId": "4de4adb9-21ee-47e3-aeb4-8cf8ed6c109a"
    }
  },
  - "geoCoordinates": {
    "latitude": 55.7558,
    "longitude": 37.6173,
    "speed": 100,
    "direction": 90,
    "valid": true
  },
  - "events": [
    - {
      "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
      "originEventId": "string",
      "objectId": "string",
      "isTrackingStart": true,
      "isTrackingEnd": true,
      "createdAt": "2023-10-04T15:43:42.574119+03:00",
      "originCreatedAt": "2023-10-04T15:43:40.120000+03:00",
      "detectorId": 601,
      - "taskIds": [
        "3fa85f64-5717-4562-b3fc-2c963f66afa6"
      ],
      - "objectDetections": [
        - {
          "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
          "parentObjectDetectionId": "49e7658b-7404-416d-a273-94c63f44de51",
          - "points": [
            - [
              0,
              0
            ],
            - [
              0,
              0
            ],
            - [
              0,
              0
            ],
            - [
              0,
              0
            ]
          ],
          "recognizedText": "string",
          "confidence": 0,
          "latitude": 55.9556,
```



```
    },  
    - "class": {  
      "confidence": 0  
    },  
    "feedbackStatus": "issue confirmed"  
  }  
]  
}  
],
```



Е Отправка экспертной оценки

Методы передачи экспертной оценки по результатам обработки изображений.

Экспертная оценка может быть передана двумя способами:

1. **Синхронно**, через метод Е.1 «Отправка экспертной оценки».
2. **Асинхронно**, через внутренние брокеры сообщений системы (`rabbitMqInternal` , `kafkaInternal`) с использованием конфигурации, получаемой методом Е.2 «Получение конфигурации очередей для асинхронной передачи экспертных оценок».

Особенности работы с оценками:

- На один скриншот можно отправлять несколько экспертных оценок.
- В статистике и дальнейшей обработке учитывается **последняя отправленная оценка** для данного скриншота.

ExpertFeedbackMessageDefault

`taskId`
`required`

`string <uuid>`
Идентификатор задачи.

`status`
`required`

`string`
Enum: `"issue confirmed"` `"issue is absent"` `"issue was skipped"` `"empty screen"` `"ignored by expert"`

Статус нарушения:

- `"issue confirmed"` – нарушение подтверждено;
- `"issue is absent"` – нарушение отсутствует;
- `"issue was skipped"` – нарушение присутствует (чистый скриншот);
- `"empty screen"` – нарушение отсутствует (чистый скриншот);
- `"ignored by expert"` – результаты ИНС не учитывались.

Соответствие с метриками экспертной оценки:

- True Positive (TP) — событие, содержащее объект/нарушение, корректно классифицированное интеллектуальным модулем как содержащее объект/нарушение. Соответствует значению `issue confirmed`;
- False Negative (FN) — событие, содержащее объект/нарушение, ошибочно классифицированное интеллектуальным модулем как не содержащее объект/нарушение. Соответствует значению `issue was skipped`;



нарушение. Соответствует значению `empty screen`;

- False Positive (FP) — событие, не содержащее объекта/нарушения, ошибочно классифицированное интеллектуальным модулем как содержащее объект/нарушение. Соответствует значению `issue is absent`.

text

string

Оptionальный комментарий. При статусе "issue is absent" возможны следующие значения:

1. Не подлежит регистрации для данного объекта / неверно выбран объект; <номер нарушения ЦАФАП>
2. Не является нарушением; <номер нарушения ЦАФАП>
3. Отклонено при подтверждении / сезонное нарушение; <номер нарушения ЦАФАП>
4. Спорное нарушение; <номер нарушения ЦАФАП>
5. Типовая ситуация; <номер нарушения ЦАФАП>
6. Дубль; <номер нарушения ЦАФАП>
7. Несоответствие критериям автоматической фиксации; <процент вероятности нарушения>.

При значениях параметра text под номерами 1, 3, 4, 5, 6 результаты отработки ИНС считаются корректными; при значениях 2,7 – некорректными

detectorId
required

integer

Идентификатор детектора

points

Array of integers [items]

Массив координат точек объекта

recognizedText

string

Распознанный ИНС текст. Поле заполняется, если эксперт хочет указать корректный текст, в случае неверного распознавания ИНС.

originScreenshotId
required

string <uuid>

Сквозной UUID скриншота, по которому отправляется экспертная оценка.

Copy Expand all Collapse all

```
{
  "taskId": "f5a1d2c3-4567-89ab-cdef-1234567890ab",
  "status": "issue confirmed",
  "text": "Спорное нарушение;1234",
  "detectorId": 501,
```



```

100,
100
],
- [
200,
100
],
- [
200,
200
],
- [
100,
200
]
],
"recognizedText": "A123BC97",
"originScreenshotId": "d4df0173-ce22-4ef2-a2f5-c43c2d7cdba7"
}

```

ExpertFeedbackMessageOdh

taskId

string <uuid>

Идентификатор задачи.

status
required

string

Enum: "issue confirmed" "issue is absent" "issue was skipped" "empty screen" "ignored by expert"

Статус нарушения:

- "issue confirmed" – нарушение подтверждено;
- "issue is absent" – нарушение отсутствует;
- "issue was skipped" – нарушение присутствует (чистый скриншот);
- "empty screen" – нарушение отсутствует (чистый скриншот);
- "ignored by expert" – результаты ИНС не учитывались.

Соответствие с метриками экспертной оценки:

- True Positive (TP) — событие, содержащее объект/нарушение, корректно классифицированное интеллектуальным модулем как содержащее объект/нарушение. Соответствует значению `issue confirmed`;
- False Negative (FN) — событие, содержащее объект/нарушение, ошибочно классифицированное интеллектуальным модулем как не содержащее объект/нарушение. Соответствует значению `issue was skipped`;



нарушение. Соответствует значению `empty screen`;

- False Positive (FP) — событие, не содержащее объекта/нарушения, ошибочно классифицированное интеллектуальным модулем как содержащее объект/нарушение. Соответствует значению `issue is absent`.

text

string

Оptionальный комментарий. При статусе "issue is absent" возможны следующие значения:

1. Не подлежит регистрации для данного объекта / неверно выбран объект; <номер нарушения ЦАФАП>
2. Не является нарушением; <номер нарушения ЦАФАП>
3. Отклонено при подтверждении / сезонное нарушение; <номер нарушения ЦАФАП>
4. Спорное нарушение; <номер нарушения ЦАФАП>
5. Типовая ситуация; <номер нарушения ЦАФАП>
6. Дубль; <номер нарушения ЦАФАП>
7. Несоответствие критериям автоматической фиксации; <процент вероятности нарушения>.

При значениях параметра text под номерами 1, 3, 4, 5, 6 результаты отработки ИНС считаются корректными; при значениях 2,7 – некорректными

detectorId
required

integer

Идентификатор детектора

points

Array of integers [items]

Массив координат точек объекта

recognizedText

string

Распознанный ИНС текст. Поле заполняется, если эксперт хочет указать корректный текст, в случае неверного распознавания ИНС.

originScreenshotId
required

string <uuid>

Сквозной UUID скриншота, по которому отправляется экспертная оценка.

Copy Expand all Collapse all

```
{
  "taskId": "f5a1d2c3-4567-89ab-cdef-1234567890ab",
  "status": "issue confirmed",
  "text": "Спорное нарушение;1234",
  "detectorId": 501,
```



```

    100,
    100
  ],
  - [
    200,
    100
  ],
  - [
    200,
    200
  ],
  - [
    100,
    200
  ]
],
"recognizedText": "A123BC97",
"originScreenshotId": "d4df0173-ce22-4ef2-a2f5-c43c2d7cdba7"
}

```

Е.1 Отправка экспертной оценки

POST /api/v1/feedback/{originScreenshotId}



Метод предназначен для отправки **экспертной оценки** по скриншоту, идентифицируемому с помощью параметра `originScreenshotId`.

В рамках отправки экспертной оценки может быть передана информация как о зафиксированном ИНС объекте/нарушении, так и об отсутствии объекта/нарушения (чистый скриншот).

При передаче экспертной оценки по чистому скриншоту в теле запроса должны быть указаны значение параметра `status` и идентификатор `detectorId`, соответствующий детектору, который не был зафиксирован на изображении.

Передача идентификатора задачи (`taskId`) обязательна для всех источников, по которым в Системе создаётся задача (схема IssueRequestDefault). Для источника типа `odh`, по которому задача не создаётся, передача идентификатора задачи не осуществляется (схема IssueRequestOdh).

AUTHORIZATIONS: > (bearerAuth and externalLogin)

PATH PARAMETERS

<code>originScreenshotId</code> required	string <uuid> Сквозной UUID скриншота, по которому отправляется экспертная оценка.
---	---

REQUEST BODY SCHEMA: application/json
required



taskId
required

string <uuid>

Идентификатор задачи.

status
required

string

Enum: "issue confirmed" "issue is absent" "issue was skipped" "empty screen" "ignored by expert"

Статус нарушения:

- "issue confirmed" – нарушение подтверждено;
- "issue is absent" – нарушение отсутствует;
- "issue was skipped" – нарушение присутствует (чистый скриншот);
- "empty screen" – нарушение отсутствует (чистый скриншот);
- "ignored by expert" – результаты ИНС не учитывались.

Соответствие с метриками экспертной оценки:

- True Positive (TP) — событие, содержащее объект/нарушение, корректно классифицированное интеллектуальным модулем как содержащее объект/нарушение. Соответствует значению `issue confirmed`;
- False Negative (FN) — событие, содержащее объект/нарушение, ошибочно классифицированное интеллектуальным модулем как не содержащее объект/нарушение. Соответствует значению `issue was skipped`;
- True Negative (TN) — событие, не содержащее объекта/нарушения, корректно классифицированное интеллектуальным модулем как не содержащее объект/нарушение. Соответствует значению `empty screen`;
- False Positive (FP) — событие, не содержащее объекта/нарушения, ошибочно классифицированное интеллектуальным модулем как содержащее объект/нарушение. Соответствует значению `issue is absent`.

text

string

Оptionальный комментарий. При статусе "issue is absent" возможны следующие значения:

1. Не подлежит регистрации для данного объекта / неверно выбран объект; <номер нарушения ЦАФАП>
2. Не является нарушением; <номер нарушения ЦАФАП>
3. Отклонено при подтверждении / сезонное нарушение; <номер нарушения ЦАФАП>
4. Спорное нарушение; <номер нарушения ЦАФАП>
5. Типовая ситуация; <номер нарушения ЦАФАП>
6. Дубль; <номер нарушения ЦАФАП>
7. Несоответствие критериям автоматической фиксации; <процент вероятности нарушения>.

При значениях параметра text под номерами 1, 3, 4, 5, 6 результаты отработки ИНС считаются корректными; при значениях 2, 7 – некорректными

detectorId
required

integer

Идентификатор детектора

points

Array of integers [items]

Массив координат точек объекта

recognizedText

string



Responses

✓ 200

Экспертная оценка успешно принята.

RESPONSE SCHEMA: `application/json`

object

Общий формат успешного ответа

> 400

Некорректный формат запроса

> 401

Ошибка авторизации

> 403

Пользователь не является автором задачи и не может отправить экспертную оценку.

> 404

Запрашиваемый ресурс не найден

> default

Внутренняя ошибка сервера

Request samples

Payload

Content type

`application/json`

Example

`IssueRequestDefault`

```
{
  "taskId": "f5a1d2c3-4567-89ab-cdef-1234567890ab",
  "status": "issue confirmed",
  "text": "Спорное нарушение;1234",
  "detectorId": 501,
```

Copy Expand all Collapse all



```
100,
100
],
- [
200,
100
],
- [
200,
200
],
- [
100,
200
]
],
"recognizedText": "A123BC97"
}
```

Response samples

200

400

401

403

404

default

Content type

application/json

Copy

{ }

Е.2 Получение конфигурации очередей для асинхронной передачи экспертных оценок

GET /api/v1/message-queues/expert-feedback



Метод предназначен для получения пользователем информации об очередях брокеров сообщений, используемых Системой для асинхронной передачи экспертных оценок. Учётные данные (логин и пароль) для подключения к указанным брокерам сообщений предоставляются посредством метода **Б.5 Получение доступа к внутренним брокерам сообщений**.

AUTHORIZATIONS: >

(bearerAuth and externalLogin)

Responses



RESPONSE SCHEMA: application/json

queues >
required

Array of RabbitMqInternalQueue (object) or KafkaInternalQueue (object)

Список очередей брокеров сообщений, доступных пользователю для асинхронной передачи экспертных оценок.

> 401

Ошибка авторизации

> default

Внутренняя ошибка сервера

Response samples

200

401

default

Content type

application/json

Copy Expand all Collapse all

```
{
  - "queues": [
    - {
      "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
      "type": "rabbitMqInternal",
      "title": "string",
      - "info": {
        "dsn": "string",
        "exchange": "string",
        "key": "string"
      }
    }
  ]
}
```