

# Описание протокола интеграции с сервисом распознавания номеров автомобилей Автомаршал

Automarshal SDK Service API

Ревизия от 28.02.2019

## Содержание

<b>Краткое описание принципа работы сервиса распознавания</b>	<b>8</b>
Быстрый старт	9
<b>Получить все подписки</b>	<b>13</b>
Описание	13
Запрос	13
Заголовки	13
Ответ	13
Код ответа	13
Заголовки	13
Тело ответа	13
Возможные коды ответов	13
<b>Получить подписки по фильтру</b>	<b>14</b>
Описание	14
Запрос	14
Заголовки	14
Тело запроса	14
Поля тела запроса	14
Схема тела запроса	15
Ответ	15
Код ответа	15
Заголовки	15
Тело ответа	15
Коды ошибок	15
Возможные коды ответов	16
<b>Получить подписку по идентификатору</b>	<b>17</b>
Описание	17
Запрос	17
Параметры	17
Ответ	17
Код ответа	17
Тело ответа	17
Схема ответа	17
Коды ошибок	17
<b>Подписаться</b>	<b>18</b>
Описание	18
Запрос	18
Тело запроса	18

# Automarshal SDK Service API

Поля объекта запроса	18
Схема тела запроса	20
Ответ	20
Код ответа	20
Тело ответа	20
Коды ошибок	20
<b>Отписаться</b>	<b>21</b>
Описание	21
Запрос	21
Параметры	21
Ответ	21
Код ответа	21
Коды ошибок	21
<b>Обновить подписку</b>	<b>22</b>
Описание	22
Запрос	22
Тело запроса	22
Поля объекта запроса	22
Схема тела запроса	23
Ответ	24
Код ответа	24
Тело ответа	24
Коды ошибок	24
<b>Объект Subscription</b>	<b>25</b>
Поля объекта	25
<b>Получить зарегистрированные типы событий</b>	<b>27</b>
Описание	27
Запрос	27
Заголовки	27
Ответ	27
Код ответа	27
Заголовки	27
Тело ответа	27
Возможные коды ответов	27
<b>Получить всех инициаторов событий</b>	<b>28</b>
Описание	28
Запрос	28
Ответ	28
Код ответа	28
Тело ответа	28

# Automarshal SDK Service API

Коды ошибок	28
<b>Получить инициатора событий по идентификатору</b>	<b>29</b>
Описание	29
Запрос	29
Параметры	29
Ответ	29
Код ответа	29
Тело ответа	29
Коды ошибок	29
<b>Добавить инициатора</b>	<b>30</b>
Описание	30
Запрос	30
Параметры	30
Тело запроса	30
Ответ	30
Код ответа	30
Тело ответа	30
Коды ошибок	30
<b>Обновить инициатора</b>	<b>32</b>
Описание	32
Запрос	32
Параметры	32
Тело запроса	32
Ответ	32
Код ответа	32
Тело ответа	32
Коды ошибок	32
<b>Удалить инициатора</b>	<b>34</b>
Описание	34
Запрос	34
Параметры	34
Ответ	34
Код ответа	34
Коды ошибок	34
<b>Объект Event</b>	<b>35</b>
Поля объекта	35
<b>Объект Publisher</b>	<b>36</b>
<b>Произвести оповещение</b>	<b>37</b>
Описание	37

# Automarshal SDK Service API

Запрос	37
Параметры	37
Тело запроса	37
Ответ	37
Код ответа	37
Тело ответа	38
Схема ответа	38
Коды ошибок	38
<b>Получить результаты распознавания</b>	<b>40</b>
Описание	40
Запрос	40
Параметры	40
Тело запроса	40
Поля объекта запроса	40
Схема тела запроса	41
Ответ	41
Код ответа	41
Тело ответа	41
Поля объекта массива ответа	41
Схема ответа	41
Коды ошибок	41
<b>Получить изображение</b>	<b>42</b>
Описание	42
Запрос	42
Параметры	42
Ответ	42
Код ответа	42
Тело ответа	42
Коды ошибок	42
<b>Начать обработку</b>	<b>44</b>
Описание	44
Запрос	44
Параметры	44
Ответ	44
Код ответа	44
Тело ответа	45
Схема ответа	45
Коды ошибок	45
<b>Остановить обработку</b>	<b>46</b>
Описание	46

# Automarshal SDK Service API

Запрос	46
Параметры	46
Ответ	46
Код ответа	47
Коды ошибок	47
<b>Получить список токенов</b>	<b>48</b>
Описание	48
Запрос	48
Ответ	48
Код ответа	48
Тело ответа	48
Схема ответа	48
<b>Получить конфигурационный файл</b>	<b>49</b>
Описание	49
Запрос	49
Ответ	49
Код ответа	49
Тело ответа	49
Коды ошибок	49
<b>Задать конфигурационный файл</b>	<b>50</b>
Описание	50
Запрос	50
Тело запроса	50
Ответ	50
Код ответа	50
Коды ошибок	50
<b>Получить протокол работы ядра</b>	<b>51</b>
Описание	51
Запрос	51
Ответ	51
Код ответа	51
Тело ответа	51
Коды ошибок	51
<b>Получить информацию о ядре</b>	<b>52</b>
Описание	52
Запрос	52
Ответ	52
Код ответа	52
Тело ответа	52
Поля объекта ответа	53

# Automarshal SDK Service API

Схема ответа	56
Коды ошибок	56
<b>Получить текущий статус ядра</b>	<b>57</b>
Описание	57
Запрос	57
Ответ	57
Код ответа	57
Тело ответа	57
Поля ответа	58
Схема ответа	59
Коды ошибок	59
<b>Получить изображение</b>	<b>60</b>
Описание	60
Запрос	60
Параметры	60
Ответ	61
Код ответа	61
Тело ответа	61
Коды ошибок	61
<b>Событие recognition</b>	<b>62</b>
Поля объекта	63
Схема решения	66
Фильтр события	67
Поля фильтра	67
Допустимые значения поля decisionComposition	68
Допустимые значения поля imagesComposition	68
Схема фильтра	69

## Краткое описание принципа работы сервиса распознавания

Ядро распознавания получает видеопоток с камеры или видеофайла, анализирует последовательность кадров и генерирует события о распознанном номере для каждого последующего автомобиля. Ядро может работать в двух режимах: *freeflow* - непрерывное распознавание номеров ТС на входном видеопотоке, *trigger* - запуск/останов распознавание по внешнему событию.

Ядро распознавания содержит модуль захвата изображения с различных видеоисточников (аналоговые и IP-камеры, видеофайлы и т.п.), что освобождает разработчика от необходимости реализовывать свой способ получения видео.

Ядро распознавания защищено специальным ключом защиты (usb-ключ + файл лицензии) и не может быть использовано без него. Параметры лицензии указываются в файле лицензии, который привязан к ключу защиты. При изменении параметров лицензии (расширение числа каналов распознавания, расширение списка поддерживаемых стран, изменение скорости 30/270 км/ч и др.) достаточно заменить файл лицензии. Замена ключа защиты не требуется.

Ядро распознавания реализовано в виде сервиса операционной системы Windows. Взаимодействие с ядром распознавания осуществляется по протоколу HTTP. Для получения [результатов распознавания](#) необходимо подписаться на событие [recognition](#) с указанием адреса обратного вызова и фильтра события. При распознавании номера ядро отправляет на указанный в подписке адрес сообщение с результатами распознавания номера транспортного средства. Пример получения результатов распознавания схематично представлен на рисунке 1.

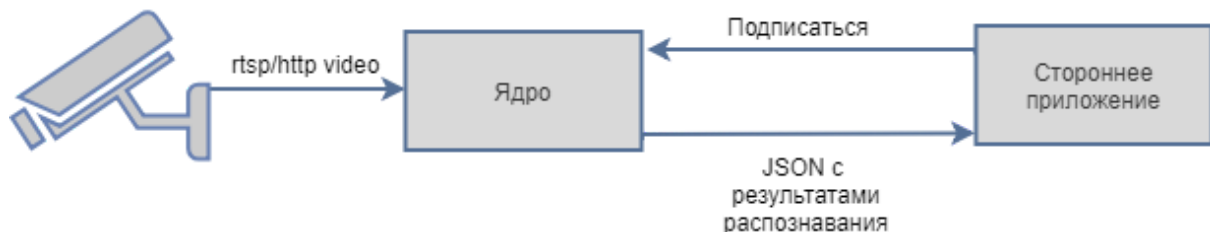


Рисунок 1 - получение результатов распознавания

Конфигурация ядра распознавания выполняется с помощью приложения "Конфигуратор ядра распознавания" или путём передачи файла конфигурации посредством [HTTP-запроса](#).



## Быстрый старт

Для быстрого старта работы с ядром распознавания необходимо выполнить следующие шаги:

1. Подписаться на событие [recognition](#)
2. Ожидать уведомлений о результатах распознавания

Однажды подписавшись на событие, стороннее приложение будет всегда получать уведомления о нём до тех пор, пока не отпишется. Взаимодействие стороннего приложения и ядра посредством уведомлений представлено на рисунке 2.

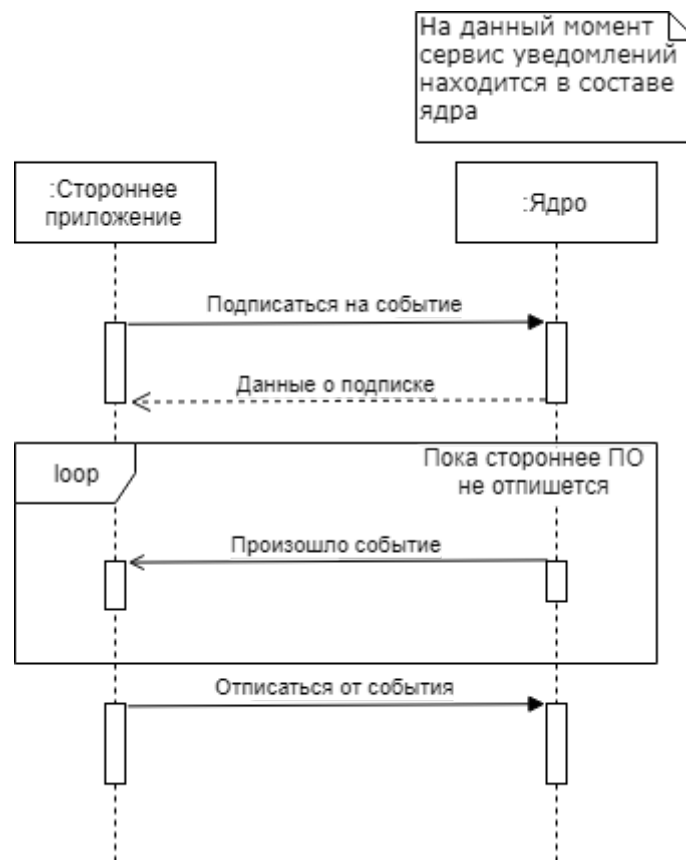


Рисунок 2 - диаграмма последовательности взаимодействия стороннего приложения и ядра

# Сервис уведомлений

Сервис уведомлений предназначен для уведомления подписчиков о каком-либо событии в системе.

На рисунке 3 представлена диаграмма последовательности, отражающая взаимодействие стороннего приложения, сервиса хранения результатов распознавания, сервиса уведомлений, и ядра.

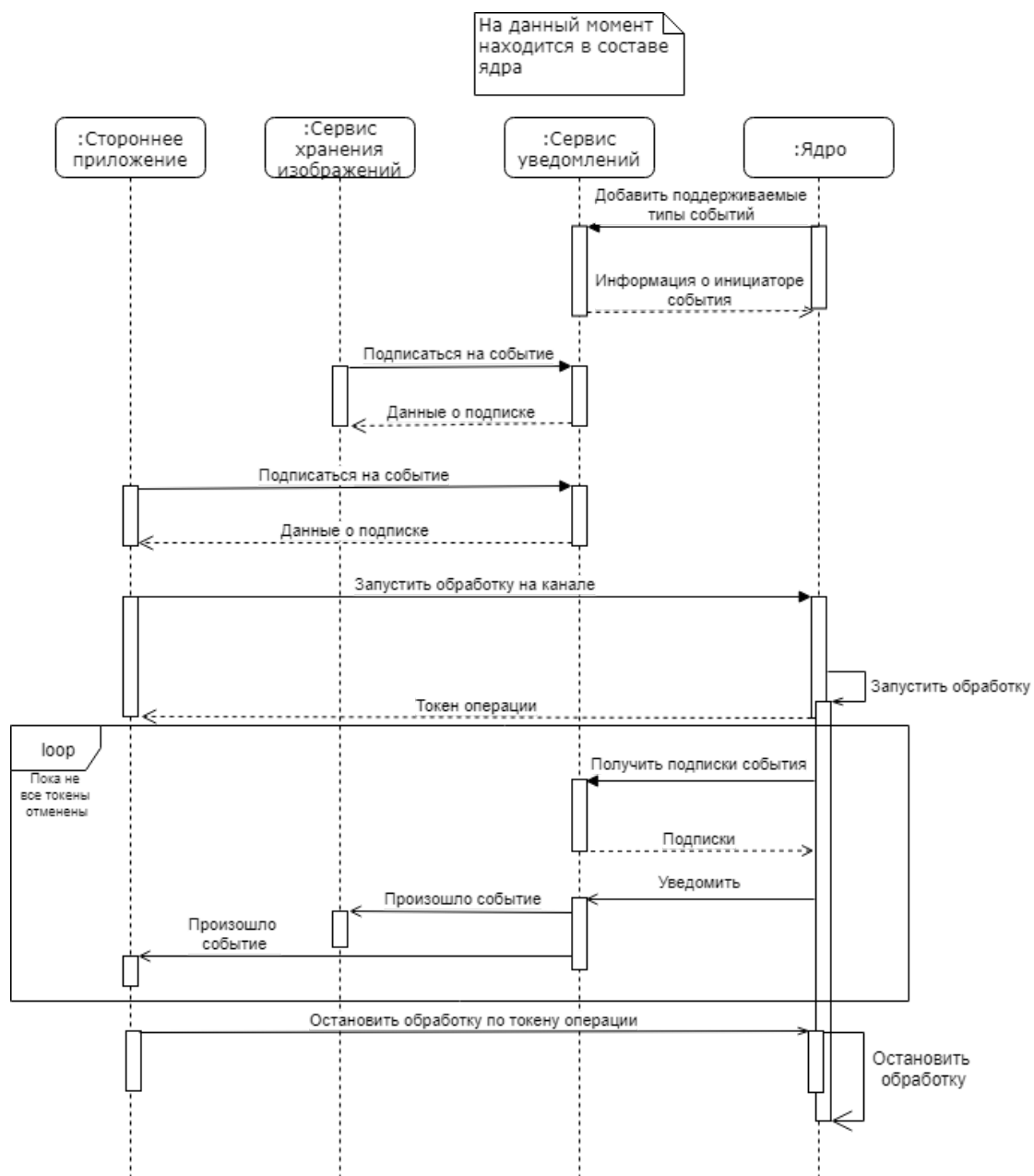


Рисунок 3 - диаграмма последовательности взаимодействия стороннего приложения, сервиса хранения результатов распознавания, сервиса уведомлений, и ядра

## Automarshal SDK Service API

При запуске ядра, оно регистрирует доступные события в сервисе уведомлений. После этого, любое приложение может подписаться на зарегистрированные типы событий. На рисунке 3 сервис хранения изображений и стороннее приложение подписываются на событие ядра. На текущий момент, доступно только событие распознавания. Подробнее об этом событии написано в разделе API ядра распознавания.

Стороннему приложению необязательно подписываться на события ядра. Оно может в любой момент запросить результаты распознавания и связанные с принятым решением изображения в сервисе хранения изображений. Но предположим что стороннее приложение в режиме реального времени отображает результаты распознавания (только строка с номером), а при необходимости получает изображения из соответствующего сервиса.

Чтобы начать распознавание, стороннему приложению необходимо отправить команду в ядро на начало обработки видео на конкретном канале. Эта команда подробнее описана в разделе API ядра распознавания. В ответ на команду, ядро вернёт токен операции. Он понадобится в дальнейшем для завершения конкретной операции обработки видео.

Во время обработки, ядро будет у сервиса уведомлений вызывать команду, выполняющую оповещение подписчиков о том, что произошло принятие решения. Но перед этим ядро получит всех подписчиков на событие распознавания, а затем отфильтрует каждого подписчика по имеющемуся у него фильтру. Затем, вызовет команду на уведомление, передав идентификаторы подписчиков, удовлетворяющих условиям их фильтров.

Ядро будет производить оповещение по каждому принятому решению до тех пор, пока не будут отменены все запущенные операции обработки видео.

Стороннее приложение может отменить операцию обработки, подав соответствующую команду ядру с токеном операции в качестве параметра.

На текущий момент сервис уведомлений входит в состав ядра. В дальнейшем планируется вынести его в отдельный микросервис.

Сервис уведомлений, ядро распознавания, а также сервис хранения результатов распознавания используют кодировку **UTF-8**.

Для всех запросов требуется заголовок Accept с значением \*/\*

Для запросов, требующих передачу тела JSON объекта требуется заголовок Content-Type с значением application/json.

Ниже представлен список доступных действий для сервиса уведомлений.

### 1. Подписки на уведомления

#### 1.1. Получить все подписки

## Automarshal SDK Service API

- 1.2. Получить подписки по фильтру
  - 1.3. Получить подписку по идентификатору
  - 1.4. Подписаться
  - 1.5. Отписаться
  - 1.6. Обновить подписку
2. Типы событий
- 2.1. Получить зарегистрированные типы событий
3. Управление инициатором событий
- 3.1. Получить всех инициаторов событий
  - 3.2. Получить инициатора событий по идентификатору
  - 3.3. Добавить инициатора
  - 3.4. Обновить инициатора
  - 3.5. Удалить инициатора
4. Публикация событий
- 4.1. Произвести оповещение
5. Уведомление подписчиков

## Получить все подписки

### Описание

Возвращает все подписки сервиса уведомлений

### Запрос

```
HTTP GET /api/v1/subscriptions
```

### Заголовки

Запрос имеет механизм кеширования. Для его включения требуется отправить заголовок If-None-Match с хеш-кодом заголовка ETag. Если содержимое ответа на запрос не изменилось, то вместо тела запроса и кода 200 OK будет возвращён ответ без тела с ответом 304 Not Modified.

### Ответ

#### Код ответа

200 OK

#### Заголовки

ETag - содержит хеш-код содержимого ответа. Требуется для включения механизма кеширования.

#### Тело ответа

В качестве тела ответа на запрос, сервер вернёт массив [объектов Subscription](#).

### Возможные коды ответов

304 Not Modified - нет изменений

## Получить подписки по фильтру

### Описание

Получает подписки по фильтру

### Запрос

```
HTTP POST /api/v1/subscriptions
```

### Заголовки

Запрос имеет механизм кеширования. Для его включения требуется отправить заголовок If-None-Match с хеш-кодом заголовка ETag. Если содержимое ответа на запрос не изменилось, то вместо тела запроса и кода 200 OK будет возвращён ответ без тела с ответом 304 Not Modified.

### Тело запроса

```
{
  "id": null,
  "callback": {
    "address": null,
    "method": null
  },
  "type": "recognition",
  "isPermanent": null
}
```

### Поля тела запроса

Наименование	Описание	Тип	Обязательное
id	идентификатор подписки	string	нет
callback	объект обратного вызова	object	нет

## Automarshal SDK Service API

callback.address	адрес обратного вызова. Должен обязательно начинаться с http или https	string	нет
callback.method	тип HTTP-запроса обратного вызова	string	нет
type	тип события	string	нет
isPermanent	требуется ли восстановить подписку после перезагрузки сервиса уведомлений	bool	нет

### Схема тела запроса

--

### Ответ

#### Код ответа

200 OK

#### Заголовки

ETag - содержит хеш-код содержимого ответа. Требуется для включения механизма кэширования.

#### Тело ответа

В качестве тела ответа на запрос, сервер вернёт массив [объектов Subscription](#).

### Коды ошибок

404 Not Found - не найдено ни одного совпадения

500 Internal Server Error - внутренняя ошибка сервера

## Automarshal SDK Service API

### **Возможные коды ответов**

304 Not Modified - нет изменений



## Получить подписку по идентификатору

### Описание

Возвращает подписку по её идентификатору

### Запрос

```
HTTP GET /api/v1/subscriptions/subscription?id=[Id]
```

### Параметры

Наименование	Описание	Тип	Обязательное
Id	Идентификатор подписки	string	да

### Ответ

#### Код ответа

200 ОК

#### Тело ответа

В качестве тела ответа на запрос, сервер вернёт [объект Subscription](#).

#### Схема ответа

--

### Коды ошибок

404 Not Found - указанная подписка не найдена

## Подписаться

### Описание

Подписывается на событие указанного типа с определенным фильтром. Схема фильтра зависит от типа запроса.

### Запрос

```
HTTP POST api/v1/subscriptions/new
```

### Тело запроса

```
{
  "callback": {
    "address": "http://example.com/subscription",
    "method": null
  },
  "type": "recognition",
  "isPermanent": null,
  "notificationsQueueCount": null,
  "notificationLifeTime": null,
  "notificationResendTimeout": null,
  "filter": "{ \"channels\": [0, 1] }"
}
```

### Поля объекта запроса

Наименование	Описание	Тип	Обязательное	Значение по умолчанию
callback	объект обратного вызова	object	да	-
callback.address	адрес обратного вызова. Должен обязательно	string	да	-

## Automarshal SDK Service API

	начинаться с http или https			
callback.method	тип HTTP-запроса обратного вызова	string	нет	POST
type	тип события	string	да	-
isPermanent	требуется ли восстановить подписку после перезагрузки сервиса уведомлений	bool	нет	false
notificationsQueueCount	длина очереди уведомлений. Принимает значения от 0 до 10	int	нет	5
notificationLifetime	время жизни уведомления. Принимает значения от 0 до 10000. Измеряется в миллисекундах	int	нет	3000
notificationResendTimeout	время повторного отправления уведомления. Принимает значения от 500 до 5000. Измеряется в миллисекундах	int	нет	1000
filter	фильтр события. Фильтр должен соответствовать схеме фильтра события	string	да	-

# Automarshal SDK Service API

## Схема тела запроса

--

## Ответ

### Код ответа

200 ОК

### Тело ответа

В качестве тела ответа на запрос, сервер вернёт [объект Subscription](#).

## Коды ошибок

400 Bad Request - неверный формат сообщения

406 Not Acceptable - не удалось распарсить тело запроса

500 Internal Server Error - внутренняя ошибка сервера

500 One of this is empty: filter, type, address - один из следующих параметров пустой: фильтр, тип события, или адрес

500 This address and method already subscribed - на указанный обратный адрес и метод уже оформлена подписка

500 Specified subscriptions type does not exists - указанный тип события не существует

500 Filter does not valid - фильтр не прошёл проверку схемы фильтра типа события

500 Notifications queue count out of range - длина очереди подключений вне допустимого диапазона (0-10)

500 Notifications life timeout of range - время жизни уведомления вне допустимого диапазона (0-10000)

500 Notifications resend timeout of range - время повторной отправки уведомления вне допустимого диапазона (500-5000)

500 Invalid address - возвратный адрес указан неверно.

## Отписаться

### Описание

Отписывается от события

### Запрос

```
HTTP DELETE api/v1/subscriptions/subscription?id=[Id]
```

### Параметры

Наименование	Описание	Тип	Обязательное
Id	Идентификатор подписки	string	да

### Ответ

Код ответа

200 ОК

### Коды ошибок

404 Not Found - не найдено ни одного совпадения

## Обновить подписку

### Описание

Обновляет фильтр подписки

### Запрос

HTTP PUT `api/v1/subscriptions/subscription`

### Тело запроса

```
{
  "id": "7e5e16cb-0c47-4a89-93a3-df31aa21b96a",
  "callback": {
    "address": null,
    "method": "POST"
  },
  "type": null,
  "isPermanent": null,
  "notificationsQueueCount": null,
  "notificationLifeTime": null,
  "notificationResendTimeout": null,
  "filter": null
}
```

### Поля объекта запроса

Наименование	Описание	Тип	Обязательное
id	идентификатор подписки	string	да
callback	объект обратного вызова	object	нет
callback.address	адрес обратного вызова. Должен обязательно начинаться с http	string	нет

## Automarshal SDK Service API

	или https		
callback.method	тип HTTP-запроса обратного вызова	string	нет
type	тип события	string	нет
isPermanent	требуется ли восстановить подписку после перезагрузки сервиса уведомлений	bool	нет
notificationsQueueCount	длина очереди уведомлений. Принимает значения от 0 до 10	int	нет
notificationLifeTime	время жизни уведомления. Принимает значения от 0 до 10000. Измеряется в миллисекундах	int	нет
notificationResendTimeout	время повторного отправления уведомления. Принимает значения от 500 до 5000. Измеряется в миллисекундах	int	нет
filter	фильтр события. Фильтр должен соответствовать схеме фильтра события	string	нет

### Схема тела запроса

--

# Automarshal SDK Service API

## Ответ

Код ответа

200 ОК

Тело ответа

В качестве тела ответа на запрос, сервер вернёт [объект Subscription](#).

## Коды ошибок

400 Bad Request - неверный формат сообщения

404 Not Found - указанная подписка не найдена

406 Not Acceptable - не удалось распарсить тело запроса

500 Internal Server Error - внутренняя ошибка сервера

500 Specified subscriptions type does not exists - указанный тип события не существует

500 Filter does not valid - фильтр не прошёл проверку схемы фильтра типа события

500 Notifications queue count out of range - длина очереди подключений вне допустимого диапазона (0-10)

500 Notifications life timeout of range - время жизни уведомления вне допустимого диапазона (0-10000)

500 Notifications resend timeout of range - время повторной отправки уведомления вне допустимого диапазона (500-5000)

500 Invalid address - возвратный адрес указан неверно.



## Объект Subscription

```
{
  "id": "7e5e16cb-0c47-4a89-93a3-df31aa21b96a",
  "createdAt": "2019-01-22T12:43:31.9711861+03:00",
  "updatedAt": "2019-01-22T12:58:33.2591967+03:00",
  "callback": {
    "address": "http://example.com/subscription",
    "method": "POST"
  },
  "type": "recognition",
  "isPermanent": false,
  "notificationsQueueCount": 5,
  "notificationLifeTime": 3000,
  "notificationResendTimeout": 1000,
  "filter": "{ \"channels\": [3, 4] }"
}
```

### Поля объекта

Наименование	Описание	Тип
id	идентификатор подписки	string
createdAt	Дата и время регистрации подписки	дата и время в строковом представлении (формат RFC 3339)
updatedAt	Дата и время обновления подписки	дата и время в строковом представлении (формат RFC 3339)
callback	объект обратного вызова	object
callback.address	адрес обратного вызова. Должен обязательно начинаться с http или https	string
callback.method	тип HTTP-запроса обратного вызова Допустимые значения: <b>GET</b> <b>POST</b>	string

## Automarshal SDK Service API

	<b>PUT DELETE</b>	
type	тип события	string
isPermanent	требуется ли восстановить подписку после перезагрузки сервиса уведомлений	bool
notificationsQueueCount	длина очереди уведомлений. Принимает значения от 0 до 10	int
notificationLifeTime	время жизни уведомления. Принимает значения от 0 до 10000. Измеряется в миллисекундах	int
notificationResendTimeout	время повторного отправления уведомления. Принимает значения от 500 до 5000. Измеряется в миллисекундах	int
filter	фильтр события. Фильтр должен соответствовать схеме фильтра события	string

В этом примере был использован [фильтр](#) события [recognition](#).

## Получить зарегистрированные типы событий

### Описание

Получает типы событий и схемы их фильтров

### Запрос

```
HTTP GET api/v1/notifications/types
```

### Заголовки

Запрос имеет механизм кеширования. Для его включения требуется отправить заголовок If-None-Match с хеш-кодом заголовка ETag. Если содержимое ответа на запрос не изменилось, то вместо тела запроса и кода 200 ОК будет возвращён ответ без тела с ответом 304 Not Modified.

### Ответ

#### Код ответа

200 ОК

#### Заголовки

ETag - содержит хеш-код содержимого ответа. Требуется для включения механизма кеширования.

#### Тело ответа

В качестве ответа на запрос, будет возвращен массив [объектов Event](#).

### Возможные коды ответов

304 Not Modified - нет изменений

## Получить всех инициаторов событий

### Описание

Возвращает всех зарегистрированных инициаторов событий

### Запрос

```
HTTP GET api/v1/publishers
```

### Ответ

#### Код ответа

200 OK

#### Тело ответа

В качестве тела ответа на запрос будет возвращён массив [объектов Publisher](#).

### Коды ошибок

Нет

## Получить инициатора событий по идентификатору

### Описание

Получает инициатора события по его идентификатору

### Запрос

```
HTTP GET api/v1/publishers/publisher?id=[Id]
```

### Параметры

Наименование	Описание	Тип	Обязательное
Id	идентификатор инициатора событий	string	да

Id - GUID инициатора событий

### Ответ

#### Код ответа

200 ОК

#### Тело ответа

В качестве тела ответа на запрос будет возвращён [объект Publisher](#).

### Коды ошибок

404 Not Found - не найден указанный инициатор событий

## Добавить инициатора

### Описание

Добавляет нового инициатора событий

### Запрос

```
HTTP POST api/v1/publishers/new
```

### Параметры

Нет

### Тело запроса

В качестве тела запроса требуется [объект Publisher](#). При этом, поля **createdAt** и **updatedAt** сервером не учитываются. Таким образом, эти поля можно опустить при составлении тела запроса.

### Ответ

#### Код ответа

200 OK

#### Тело ответа

В качестве тела ответа на запрос будет возвращён [объект Publisher](#).

### Коды ошибок

400 Bad Request - неверный формат сообщения

406 Not Acceptable - не удалось распарсить тело запроса

500 Internal Server Error - внутренняя ошибка сервера

500 Id or Events empty - не указан идентификатор инициатора или его список событий пуст

500 This publisher already exists - инициатор с таким идентификатором уже существует

500 Next events are duplicating: [типы событий] - приведенные типы событий уже зарегистрированы

500 One of events is null or empty - имя типа одно из событий не указано

## Automarshal SDK Service API

500 Event [тип события] has empty filter schema - событие не имеет фильтра

500 Event [тип события] has unsupported version of schema - событие имеет схему не поддерживаемой версии

## Обновить инициатора

### Описание

Обновляет данные об инициаторе событий

### Запрос

```
HTTP PUT api/v1/publishers/publisher
```

### Параметры

Нет

### Тело запроса

В качестве тела запроса требуется [объект Publisher](#). При этом, поля **createdAt** и **updatedAt** сервером не учитываются. Таким образом, эти поля можно опустить при составлении тела запроса.

### Ответ

#### Код ответа

200 ОК

#### Тело ответа

В качестве тела ответа на запрос будет возвращён [объект Publisher](#).

### Коды ошибок

400 Bad Request - неверный формат сообщения

406 Not Acceptable - не удалось распарсить тело запроса

404 Not Found - не найден инициатор событий

500 Internal Server Error - внутренняя ошибка сервера

500 Next events are duplicating: [типы событий] - приведенные типы событий уже зарегистрированы

500 One of events is null or empty - имя типа одно из событий не указано

500 Event [тип события] has empty filter schema - событие не имеет фильтра



## Automarshall SDK Service API

500 Event [тип события] has unsupported version of schema - событие имеет схему не поддерживаемой версии

## Удалить инициатора

### Описание

Удаляет инициатора событий

### Запрос

```
HTTP DELETE api/v1/publishers/publisher?id=[Id]
```

### Параметры

Наименование	Описание	Тип	Обязательное
Id	идентификатор инициатора событий	string	да

### Ответ

#### Код ответа

200 OK

### Коды ошибок

404 Not Found - не найден указанный инициатор событий

## Объект Event

```
{
  "type": "error",
  "schema": "{\\"definitions\\": {},\\"$schema\\":
  \\"http://json-schema.org/draft-06/schema#",\\"$id\\":
  \\"http://example.com/root.json\\",\\"type\\": \\"object\\",\\"title\\": \\"The
  Root Schema\\",\\"required\\": [\\"errorLevel\\"],\\"properties\\":
  {\\"errorLevel\\": {\\"$id\\": \\"#/properties/errorLevel\\",\\"type\\":
  \\"integer\\",\\"title\\": \\"The errorLevel schema\\",\\"default\\":
  0,\\"examples\\": [3]}}}"
}
```

### Поля объекта

Наименование	Описание	Тип	Обязательное
type	тип события	string	да
schema	схема фильтра подписки на событие	string	да

## Объект Publisher

```
{
  "id": "KERNEL",
  "createdAt": "2019-01-22T09:30:45.3997652+03:00",
  "updatedAt": "2019-01-22T09:36:42.3552201+03:00",
  "events": [Event]
}
```

Наименование	Описание	Тип	Обязательное
id	идентификатор инициатора событий	string	да
createdAt	дата и время регистрации инициатора. При добавлении или изменении инициатора, поле не учитывается	дата и время в строковом представлении (формат RFC 3339)	нет
updatedAt	дата и время обновления инициатора. При добавлении или изменении инициатора, поле не учитывается	дата и время в строковом представлении (формат RFC 3339)	нет
events	список событий инициатора	Array< <a href="#">Event</a> >	да

## Произвести оповещение

### Описание

Сообщить сервису уведомлений о том, что произошло событие

### Запрос

```
HTTP POST api/v1/notifications/notify
```

### Параметры

Нет

### Тело запроса

В качестве тела запроса принимается только multipart/form-data.

состав:

- поле subscribers содержит перечисление идентификаторов подписок, которые необходимо уведомить о наступлении события;
- поле eventData содержит json с данными события
- также допустимо присутствие любого количества файлов. Имена полей с файлами должны начинаться со слова file

```
Content-Disposition: form-data; name="subscriptions"
```

```
["1297301d-cc1c-40f3-97e4-b964641f06db"]
```

```
Content-Disposition: form-data; name="eventData"
```

```
{}
```

```
Content-Disposition: form-data; name="file0";  
filename="C:\Users\admin\Downloads\132.html"
```

### Ответ

#### Код ответа

200 OK

## Automarshal SDK Service API

Тело ответа

Нет

Схема ответа

Нет

### Коды ошибок

400 Bad Request - неверный формат сообщения

500 Internal Server Error - внутренняя ошибка сервера

# Сервис хранения результатов распознавания

Данный сервис подписывается на событие распознавания ядра. По наступлению события, сохраняет полученные данные на диск.

Для данного сервиса доступно следующее API:

1. Слушатель уведомлений
2. Получить результаты распознавания
3. Получить изображение

## Получить результаты распознавания

### Описание

Возвращает стоп-кадр и результат распознавания

### Запрос

```
HTTP POST api/v1/recognitionStorage/decisions
```

### Параметры

Нет

### Тело запроса

```
{
  "channelId":3,
  "dateTimeFrom":"2017-03-17T12:00:25.543804071+02:00",
  "dateTimeTo":"2017-03-17T16:00:25.543804071+02:00"
}
```

### Поля объекта запроса

Наименование	Описание	Тип	Обязательное
channelId	идентификатор канала	int	нет
dateTimeFrom	дата и время начала диапазона поиска сохраненных решений	дата и время в строковом представлении (формат RFC 3339)	нет
dateTimeTo	дата и время конца диапазона поиска сохраненных решений	дата и время в строковом представлении (формат RFC 3339)	нет



# Automarshall SDK Service API

## Схема тела запроса

--

## Ответ

### Код ответа

200 OK

### Тело ответа

```
[
  {
    "decision":{/** о б ъ е к т Decision **/},
    "images":[
      "/api/v1/recognitionStorage/images?id=3"
    ]
  }
]
```

### Поля объекта массива ответа

Наименование	Описание	Тип
decision	JSON с результатами принятия решения	<a href="#">Decision</a>
images	массив ссылок на изображения, связанных с принятым решением	Array<string>

## Схема ответа

--

## Коды ошибок

400 Bad Request - неверный формат сообщения

404 Not Found - не найдено ни одного совпадения

500 Internal Server Error - внутренняя ошибка сервера

## Получить изображение

### Описание

Возвращает изображение результата распознавания.

### Запрос

```
HTTP GET /api/v1/recognitionStorage/images?id=[Id]
```

### Параметры

Наименование	Описание	Тип	Обязательное
Id	идентификатор изображения	string	да

### Ответ

#### Код ответа

200 ОК

#### Тело ответа

Изображение

### Коды ошибок

404 Not Found - не найдено ни одного совпадения

# API ядра распознавания

Для ядра распознавания доступно следующее API:

1. Начать обработку
2. Остановить обработку
3. Получить конфигурационный файл
4. Задать конфигурационный файл
5. Получить протокол работы ядра
6. Получить информацию о ядре
7. Получить изображение
8. Получить текущий статус ядра

## Начать обработку

### Описание

Запускает обработку на канале. Для запуска распознавания на канале только по внешним сигналам параметр **MotionDetectMethod** должен иметь значение "ExternalPlugin". Данный параметр находится в файле конфигурации ядра, секция **ProcessUnit**, которая находится по следующему пути Kernel.VideoChannels.

### Запрос

```
HTTP GET /api/v1/kernel/processing/start?chId=[ChId]&stopAfter=[Ms]
```

### Параметры

Наименование	Описание	Тип	Обязательное	Значение по умолчанию
ChId	номер канала на котором требуется запустить обработку	int	да	-
Ms	задержка автоматической остановки обработки на канале. Измеряется в миллисекундах	int	нет	null - бесконечная обработка

### Ответ

#### Код ответа

200 OK

# Automarshal SDK Service API

## Тело ответа

```
"730f81e2-884f-460b-8e27-43c7821b0526"
```

## Схема ответа

--

## Коды ошибок

500 Channel not configured - MotionDetectMethod канала не установлен в значение ExternalPlugin

500 Internal Server Error - внутренняя ошибка сервера

## Остановить обработку

### Описание

Останавливает обработку по токену

### Запрос

```
HTTP GET
/api/v1/kernel/processing/stop?token=[Token]&delay=[Ms]&force=[Force]
```

### Параметры

Наименование	Описание	Тип	Обязательное	Значение по умолчанию
Token	токен операции, которую требуется остановить	string	да	-
Ms	задержка автоматической остановки обработки на канале. Измеряется в миллисекундах	int	нет	null - операция будет завершена немедленно
Force ( <b>не реализован в текущей версии</b> )	требуется ли немедленно сформировать результат распознавания	bool	нет	false

**Параметр Force будет реализован в одной из следующих версий.**

# Automarshal SDK Service API

## Ответ

Код ответа

200 OK

## Коды ошибок

404 Not Found - не найдена задача на обработку видеопотока с указанным токеном

500 Internal Server Error - внутренняя ошибка сервера

## Получить список токенов

### Описание

Возвращает список токенов операции "Обработка".

### Запрос

```
HTTP GET /api/v1/kernel/processing/tokens
```

### Ответ

#### Код ответа

200 OK

#### Тело ответа

```
[  
  "55ss1df5-s214-df56-1f8x-2df46j5cdfh6",  
  "32dd43e4-dece-4a8c-aaf3-8b1d21595f33"  
]
```

#### Схема ответа

--



## Получить конфигурационный файл

### Описание

Получить файл настроек ядра распознавания.

### Запрос

```
HTTP GET /api/v1/kernel/settings
```

### Ответ

#### Код ответа

200 ОК

#### Тело ответа

Файл в формате .xml

### Коды ошибок

Нет

## Задать конфигурационный файл

### Описание

Устанавливает настройки ядра распознавания.

### Запрос

```
HTTP POST /api/v1/kernel/settings
```

### Тело запроса

Файл в формате .xml

### Ответ

#### Код ответа

200 OK

### Коды ошибок

500 Internal Server Error - внутренняя ошибка сервера

## Получить протокол работы ядра

### Описание

Возвращает файл с логом ядра распознавания

### Запрос

```
HTTP GET /api/v1/kernel/logs/log
```

### Ответ

#### Код ответа

200 OK

#### Тело ответа

Возвращает файл формата .log

### Коды ошибок

Нет

## Получить информацию о ядре

### Описание

Возвращает информацию о ядре распознавания

### Запрос

```
HTTP GET /api/v1/kernel/info
```

### Ответ

Код ответа

200 OK

Тело ответа

```
{
  "core": {
    "version": "2.19.3",
    "commitDate": "2017-03-17T12:00:25.543804071+02:00",
    "commitHash": "113d-s5d546-d4f45d6",
    "buildDate": "2017-03-17T12:00:25.543804071+02:00"
  },
  "api": {
    "version": "1",
    "commitDate": "2017-03-17T12:00:25.543804071+02:00",
    "commitHash": "113d-s5d546-d4f45d6",
    "buildDate": "2017-03-17T12:00:25.543804071+02:00"
  },
  "license": {
    "channelsCountInfo": {
      "processing": 1,
      "surveillance": 3
    },
    "maxProcessingFps": 90,
    "countries": ["RU"],
    "validUntil": "2018-03-17T12:00:25.543804071+02:00",
  }
}
```

## Automarshal SDK Service API

```
"createdData": "2017-03-17T12:00:25.543804071+02:00",
"supportUntil": "2017-03-17T12:00:25.543804071+02:00",
"status": 2
},
"dongle": {
  "number": "be3aeb87",
  "owner": "Петров Пётр"
},
"hardware": {
  "ram": "2147483648",
  "memory": [
    {
      "volume": "C:\\",
      "label": "Windows 10",
      "totalFree": "1909719040 bytes",
      "totalSize": "127440777216 bytes"
    },
    {
      "volume": "D:\\",
      "label": "Files storage",
      "totalFree": "736163074048 bytes",
      "totalSize": "1000068870144 bytes"
    }
  ]
},
"cpu": {
  "cpuName": "Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz",
  "cpuCores": 4,
  "cpuFrequency": "3.60GHz"
},
"software": {
  "os": {
    "name": "Windows 10 Pro",
    "architecture": "x64",
    "version": "1809"
  }
}
}
```

Поля объекта ответа

Наименование	Описание	Тип
--------------	----------	-----

## Automarshal SDK Service API

core	информация о ядре	object
core.version	версия ядра	string
core.release	дата и время релиза	дата и время в строковом представлении (формат RFC 3339)
core.commitDate	дата и время коммита	дата и время в строковом представлении (формат RFC 3339)
core.commitHash	хеш-код коммита	string
core.buildDate	дата и время сборки	дата и время в строковом представлении (формат RFC 3339)
api	информация о протоколе взаимодействия	object
api.version	версия протокола	string
api.commitDate	дата и время коммита	дата и время в строковом представлении (формат RFC 3339)
api.commitHash	хеш-код коммита	string
api.buildDate	дата и время сборки	дата и время в строковом представлении (формат RFC 3339)
license	информация о лицензии	object
license.channelsCountInfo	информация о каналах	object
license.channelsCountInfo.processing	каналов обработки	int
license.channelsCountInfo.surveillance	каналов наблюдения	int
license.maxProcessingFps	ограничение по количеству обрабатываемых кадров	int
license.countries	список доступных стран	Array<string>
license.validUntil	срок окончания действительности лицензии	дата и время в строковом представлении (формат RFC 3339)

## Automarshal SDK Service API

license.createdData	дата и время создания лицензии	дата и время в строковом представлении (формат RFC 3339)
license.supportUntil	срок окончания поддержки	дата и время в строковом представлении (формат RFC 3339)
license.status	статус лицензии. <b>0</b> - лицензия не действительна <b>1</b> - лицензия действительна <b>2</b> - нет лицензии	enum
dongle	информация о ключе	object
dongle.number	номер ключа	string
dongle.owner	владелец ключа	string
hardware	информация о железе ПК ядра	object
hardware.ram	количество оперативной памяти	string
hardware.memory	информация о памяти	Array<object>
hardware.memory[i].volume	том	string
hardware.memory[i].label	метка тома	string
hardware.memory[i].totalFree	всего свободно памяти	string
hardware.memory[i].totalSize	всего памяти	string
hardware.cpu	информация о ядре	object
hardware.cpu.cpuName	наименование процессора	string
hardware.cpu.cpuCores	количество ядер в процессоре	int
hardware.cpu.cpuFrequency	частота процессора	string
software	информация о ПО ПК ядра	object
software.os	информация о операционной системе	object

## Automarshal SDK Service API

software.os.name	наименование операционной системы	string
software.os.architecture	разрядность операционной системы	string
software.os.version	версия операционной системы	string

### Схема ответа

--

### Коды ошибок

Нет



## Получить текущий статус ядра

### Описание

Возвращает текущее состояние ядра.

### Запрос

```
HTTP GET /api/v1/kernel/status
```

### Ответ

Код ответа

200 OK

Тело ответа

```
{
  "channels": [{
    "id": 0,
    "fps": 30,
    "framesSkipped": 10,
    "framesProcessed": 54546,
    "isProcessingStarted": true,
    "channelName": "ch1",
    "timestamps": {
      "firstFrame": "2019-01-22T12:58:33.2591967+03:00",
      "lastFrame": "2019-01-22T12:58:33.2591967+03:00"
    },
    "resolution": "1920 x 1080",
    "videoSourceStatus": {
      "statusCode": 0,
      "statusDescription": "off"
    }
  }
  ],
  "kernelStatus": 5,
  "kernelStartedAt": "2019-01-22T12:58:33.2591967+03:00",
  "memory": {
    "lastUpdate": "2019-01-22T12:58:33.2591967+03:00",
    "totalBytes": 67475775488,
    "consumedBytes": 67061047296
  }
}
```

## Automarshal SDK Service API

```
}
}
```

### Поля ответа

Наименование	Описание	Тип
channels	массив описаний каналов	Array<object>
channels[i].id	идентификатор канала	int
channels[i].fps	количество кадров в секунду	int
channels[i].framesSkipped	количество пропущенных кадров с момента запуска ядра	int
channels[i].framesProcessed	количество обработанных кадров с момента запуска ядра	int
channels[i].isProcessingStarted	запущена ли обработка	bool
channels[i].channelName	наименование канала	string
channels[i].timestamps	информация о дате и времени получения кадров каналом	object
channels[i].timestamps.firstFrame	дата и время получения первого кадра	дата и время в строковом представлении (формат RFC 3339)
channels[i].timestamps.lastFrame	дата и время получения последнего кадра	дата и время в строковом представлении (формат RFC 3339)
channels[i].videoSourceStatus	информация о статусе видеоисточника	object
channels[i].videoSourceStatus.statusCode	код статуса видео источника <b>-2147483646</b> - открывается <b>-2147483640</b> - закрывается <b>0</b> - источник не сконфигурирован <b>1</b> - источник закрыт	enum

## Automarshal SDK Service API

	<b>4</b> - источник запущен <b>16</b> - нет сигнала	
channels[i].videoSourceStatus.description	описание кода статуса видеисточника	string
channels[i].resolution	разрешение видеопотока канала	string
kernelStatus	отображает состояние видеисточников <b>2</b> - останавливается <b>3</b> - остановлено <b>4</b> - запускается <b>5</b> - запущено	enum
kernelStartedAt	дата и время запуска ядра	дата и время в строковом представлении (формат RFC 3339)
memory	описание состояния оперативной памяти	object
memory.lastUpdate	дата и время последнего обновления статуса памяти	дата и время в строковом представлении (формат RFC 3339)
memory.totalBytes	всего оперативной памяти на ПК	int
memory.usedBytes	израсходовано памяти приложением	int

### Схема ответа

--

### Коды ошибок

Нет

## Получить изображение

### Описание

Возвращает изображение с канала

### Запрос

```
HTTP GET
/api/v1/kernel/videochannels/image?id=[Id]&format=[Format]&timestamp=[Timestamp]
```

### Параметры

Наименование	Описание	Тип	Обязательный	Значение по умолчанию
Id	идентификатор канала с которого требуется получить изображение	int	да	-
Format	параметр, указывающий формат отдаваемого изображения (JPEG, PNG, BMP)	string	нет	JPEG
Timestamp (Не реализован в текущей версии)	дата и время изображения с видеоисточника. Будет возвращен кадр из циклического буфера, временная метка которого	дата и время в строковом представлении (формат RFC 3339)	нет	дата и время последнего доступного кадра

## Automarshal SDK Service API

	ближе всего к Timestamp.			
--	-----------------------------	--	--	--

**Параметр Timestamp будет реализован в одной из следующих версий.**

### Ответ

Код ответа

200 ОК

Тело ответа

Изображение формата Format или изображение формата Format, соответствующее временной метке Timestamp. Timestamp изображения будет передан в заголовке Date.

### Коды ошибок

404 Not Found - В буфере не найден подходящий кадр.

## Событие recognition

Во время работы ядро посылает решения через сервис уведомлений. Событием генерации решения по распознаванию является событие recognition. При вызове этого события к уведомлению будет приложено решение следующего вида

```
{
  "tokens": [
    "df5585-dfg5-455d5-45ffj",
    "dfdfg1-51gfh5-51dgf5-151df"
  ],
  "id": 245,
  "channel": {
    "id": 0,
    "name": "Видео канал 1"
  },
  "timestamps": {
    "bestFrame": "2019-01-22T12:43:31.9711861+03:00",
    "firstFrame": "2019-01-22T12:43:31.9711861+03:00",
    "lastFrame": "2019-01-22T12:43:31.9711861+03:00"
  },
  "plate": {
    "number": "A123AA35",
    "confidence": 0.9987,
    "stencil": {
      "id": "RU_A000AA00",
      "country": {
        "id": "ru",
        "iso31066_1_Alpha2": "RU",
        "iso31066_1_Alpha3": "RUS",
        "iso31066_1_Numeric": 643
      }
    }
  },
  "bounds": [
    {
      "x": 120,
      "y": 110
    },
    {
      "x": 80,
      "y": 120
    },
    {
      "x": 80,
      "y": 120
    }
  ]
}
```

## Automarshall SDK Service API

```
    },
    {
      "x": 80,
      "y": 120
    }
  ],
  "symbols": [
    {
      "char": "A",
      "order": 0,
      "confidence": 0.8945,
      "bounds": [
        {
          "x": 120,
          "y": 110
        },
        {
          "x": 80,
          "y": 120
        },
        {
          "x": 80,
          "y": 120
        },
        {
          "x": 80,
          "y": 120
        }
      ]
    }
  ]
},
"movement": {
  "angle": 42,
  "direction": 1,
  "passage": 2,
  "name": "из Череповца в Вологду"
},
"additionalInfo": {
  "framecount": 15,
  "reason": "left movement [15 frames with movement]",
  "isBlocked": false
}
}
```

## Automarshal SDK Service API

### Поля объекта

Наименование	Описание	Тип
tokens	список токенов операций обработки видео	Array<string>
Id	сквозной идентификатор решения. При каждом перезапуске ядра обнуляется	int
channel	информация о видеоканале	object
channel.id	идентификатор видеоканала	int
channel.name	наименование видеоканала	string
timestamps	информация о дате и времени принятия решения	object
timestamps.bestFrame	дата и время лучшего кадра принятия решения	дата и время в строковом представлении (формат RFC 3339)
timestamps.firstFrame	дата и время первого кадра принятия решения	дата и время в строковом представлении (формат RFC 3339)
timestamps.lastFrame	дата и время последнего кадра принятия решения	дата и время в строковом представлении (формат RFC 3339)
plate	информация о номере	object
plate.number	распознанный номер	string
plate.confidence	достоверность распознавания	long
plate.stencil	информация о шаблоне номера	object
plate.stencil.id	идентификатор шаблона номера	string



## Automarshal SDK Service API

plate.stencil.country	информация о стране шаблона	object
plate.stencil.country.id	идентификатор страны шаблона номера	string
plate.stencil.country.iso31066_1_Alpha2	код страны по стандарту iso 31066-1-Alpha2	string
plate.stencil.country.iso31066_1_Alpha3	код страны по стандарту iso 31066-1-Alpha3	string
plate.stencil.country.iso31066_1_Numeric	код страны по стандарту iso 31066-1-Numeric	int
plate.bounds	границы углов номерной пластины	Array<object>
plate.bounds[i].x	координата точки по оси x	int
plate.bounds[i].y	координата точки по оси y	int
plate.symbols[i]	информация о символе из номера	object
plate.symbols[i].char	распознанный символ номера	string
plate.symbols[i].order	порядковый номер символа в распознанном номере из поля plate.number. Для неотображаемых символов символов имеет значение null	int   null
plate.symbols[i].confidence	достоверность распознавания	long
plate.symbols[i].bounds	границы символа номерной пластины	Array<object>
plate.symbols[i].bounds[i].x	координата точки по оси x	int
plate.symbols[i].bounds[i].y	координата точки по оси y	int
movement	информация о перемещении	object
movement.direction	направление перемещения ТС 0 - не определено 1 - въезд на территорию	enum

## Automarshal SDK Service API

	<b>2</b> - выезд с территории	
movement.passage	направление движения по кадру <b>0</b> - неизвестно <b>1</b> - снизу вверх <b>2</b> - сверху вниз	enum
movement.name	наименование перемещения	string
additionalInfo	дополнительная информация	object
additionalInfo.framecount	число кадров, понадобившихся для принятия решения	int
additionalInfo.reason	причина принятия решения	string
additionalInfo.isBlocked	является ли решение заблокированным	bool

Помимо JSON с решением могут быть несколько изображений.

### Схема решения

--

## Фильтр события

Событие recognition требует следующий фильтр

```
{
  "channels": [3],
  "decisionComposition": ["Plate", "Bounds", "Channel"],
  "imagesComposition": ["DecisionFrame", "LinkedFrames",
    "NumberPlateImage"],
  "isBlocked": true
}
```

## Поля фильтра

Наименование	Описание	Тип	Обязательное	Значение по умолчанию
channels	Список идентификаторов видеоканалов. <b>null</b> - все видеоканалы <b>пустой массив</b> - ни одного видеоканала	Array<int>	нет	null
decisionComposition	список требуемых в объекте решения полей. <b>null</b> - все поля <b>пустой массив</b> - пустой JSON объект. <a href="#">Допустимые значения</a>	Array<string> или Array<int>	нет	null
imagesComposition	список требуемых изображений. <b>null</b> - все изображения. <b>пустой массив</b> - изображения не будут отправлены. <a href="#">Допустимые значения</a>	Array<string> или Array<int>	нет	null

## Automarshal SDK Service API

	<a href="#">значения</a>			
isBlocked	требуется ли оповещать о заблокированных решениях	bool	нет	false

Допустимые значения поля decisionComposition

Значение в строковом представлении	Численное представление	Описание
Tokens	0	Добавляет список токенов операции распознавания
Channel	1	Добавляет информацию о видеоканале
Timestamps	2	Добавляет информацию временных метрах
Plate	3	Добавляет информацию о номере
PlateStencil	4	Добавляет информацию о шаблоне номера
PlateBounds	5	Добавляет информацию о границах номера
PlateSymbols	6	Добавляет информацию о символах номера
Movement	7	Добавляет информацию о перемещении ТС
AdditionalInfo	8	Добавляет дополнительную информацию

Допустимые значения поля imagesComposition

Значение в строковом представлении	Численное представление	Описание
------------------------------------	-------------------------	----------

## Automarshal SDK Service API

DecisionFrame	0	Добавляет кадр решения
LinkedFrames	1	Добавляет связанные кадры с кадром решения
NumberPlateImage	2	Добавляет изображение номерной пластины

### Схема фильтра

--