



# Интеграция с порталом

*REST API Документация вер. 1.14*

Дата	Версия	Замечания по релизу
02/12/2013	1.0	Первый публичный релиз документа
05/06/2014	1.5	<ol style="list-style-type: none"> <li>1. Расширен API GET /passes для получения статистической отчетности о картах и значений полей карт.</li> <li>2. Расширен API GET /templates для получения статистической отчетности о шаблонах.</li> <li>3. Расширен API GET /templates/tname для скачивания графических файлов шаблона.</li> <li>4. Исправлены некоторые ошибки и неточности.</li> </ol>
10/08/2014	1.6	<ol style="list-style-type: none"> <li>1. Добавлен метод группового обновления API PUT /passesintemplate/templatename для одновременного обновления всех карт шаблона.</li> <li>2. Расширен API /passes для получения статуса карты и индивидуальной графической информации</li> </ol>
01/07/2015	1.7	<ol style="list-style-type: none"> <li>1. Документированы многие методы и параметры запросов, ранее не публиковавшиеся в публичной версии API интерфейса.</li> </ol>
15/10/2015	1.8	<ol style="list-style-type: none"> <li>1. Добавлены методы для динамического пуш и акций.</li> <li>2. Добавлены методы регистрации карт</li> </ol>
26/11/2015	1.9	<ol style="list-style-type: none"> <li>1. Документированы дополнительные методы для работы с регистрацией карт.</li> </ol>
10/12/2015	1.10	<ol style="list-style-type: none"> <li>1. Расширено описание метасимволов для отправки почты и использования в шаблонах маркетинговых кампаний.</li> <li>2. Изменена структура ссылок на приложения карты cardApps в шаблоне карты.</li> <li>3. Добавлено описание поля altValue в карте и шаблоне</li> <li>4. Добавлены дополнительные фильтры в запрос /passes</li> </ol>
13/03/2016	1.11	<ol style="list-style-type: none"> <li>1. Обновленный API для работы с маркетинговыми кампаниями /campaigns2. Доступна работа с заранее заданным списком карт. Рекомендуется к использованию вместо /campaigns</li> </ol>
10/06/2016	1.12	<ol style="list-style-type: none"> <li>1. Документированы методы сбора статистики и получения уведомлений по событиям с картами</li> </ol>
26/06/2016	1.13	<ol style="list-style-type: none"> <li>1. Обновлен метод создания карты</li> <li>2. Добавлен механизм HTTP Basic Authentication в URL события</li> <li>3. Добавлены методы группового создания и обновления карт</li> </ol>
08/07/2016	1.14	<ol style="list-style-type: none"> <li>1. Добавлены методы работы с резервированием карт</li> </ol>

<b>Предисловие.....</b>	<b>10</b>
<b>Общие положения.....</b>	<b>10</b>
Доступ к API.....	10
Авторизация.....	10
<b>Поддерживаемые команды .....</b>	<b>11</b>
<b>Запросить информацию по карте .....</b>	<b>12</b>
extendedInfo .....	12
<b>Формат возвращаемых данных.....</b>	<b>12</b>
label.....	12
value .....	12
altValue.....	12
changeMsg.....	12
status .....	13
statusCode.....	13
voided .....	13
expiryDate.....	13
template .....	14
serialNo .....	14
limit.....	15
locations .....	15
geo .....	15
message.....	15
barcode.....	15
images .....	15
colors.....	15
<b>Запросить список доступных шаблонов .....</b>	<b>16</b>
serialTotal .....	16
serialActive .....	16
deviceCount.....	16
<b>Запросить информацию о шаблоне .....</b>	<b>17</b>
<b>Формат возвращаемых данных.....</b>	<b>17</b>
<b>Обновить значения шаблона.....</b>	<b>19</b>
<b>Формат данных .....</b>	<b>19</b>
<b>Формат возвращаемых данных.....</b>	<b>20</b>
<b>Создать новый шаблон .....</b>	<b>20</b>
<b>Формат данных .....</b>	<b>20</b>
<b>Редактировать шаблон .....</b>	<b>21</b>
<b>Запросить список карт .....</b>	<b>22</b>
template .....	22
activeOnly.....	22
status .....	22
stats.....	22
graphics.....	23
barcode.....	23
fields .....	23
page.....	23
filterVoided.....	23
filterStatus.....	23
geo .....	23

<b>Формат возвращаемых данных - стандартный .....</b>	<b>25</b>
<b>Формат возвращаемых данных - расширенный .....</b>	<b>25</b>
statusCode.....	27
serialNo .....	27
template .....	27
downloaded .....	27
updated .....	27
registered .....	27
created .....	27
devices .....	27
limit .....	27
fields .....	27
<b>Обновить значение карты .....</b>	<b>28</b>
<b>Формат данных .....</b>	<b>29</b>
void.....	29
expiryDate.....	29
barcode.....	29
images .....	29
colors .....	30
<b>Формат возвращаемых данных.....</b>	<b>30</b>
<b>Обновить все карты в шаблоне .....</b>	<b>30</b>
<b>Создать новую карту .....</b>	<b>31</b>
<b>Формат возвращаемых данных.....</b>	<b>31</b>
<b>Удалить карту.....</b>	<b>32</b>
<b>Возвращаемые данные.....</b>	<b>32</b>
<b>Отправить ссылку на загрузку карты по СМС .....</b>	<b>32</b>
<b>Формат данных .....</b>	<b>33</b>
message .....	33
sender .....	33
<b>Отправить ссылку на загрузку карты по email.....</b>	<b>33</b>
<b>Формат данных .....</b>	<b>34</b>
from .....	34
fromName .....	34
subject.....	34
body .....	34
Метасимволы.....	35
<b>Отправить ссылку на загрузку карты по email (2) .....</b>	<b>35</b>
<b>Запросить ссылку на загрузку карты .....</b>	<b>36</b>
<b>Формат данных .....</b>	<b>36</b>
type .....	36
<b>Формат возвращаемых данных.....</b>	<b>36</b>
<b>Переместить карту на другой шаблон .....</b>	<b>37</b>
<b>Отправить PIN-код по СМС .....</b>	<b>38</b>
<b>Формат данных .....</b>	<b>38</b>
smsText.....	38
length .....	38
<b>Возвращаемые данные.....</b>	<b>39</b>
<b>Проверить ранее выданный PIN-код.....</b>	<b>39</b>

token.....	39
pin.....	39
<b>Возвращаемые данные.....</b>	<b>39</b>
<b>Запросить общие параметры сервиса .....</b>	<b>40</b>
<b>Формат возвращаемых данных.....</b>	<b>40</b>
helper : text : ru.....	41
helper : text : en .....	41
sms .....	41
email .....	41
<b>Изменить общие параметры сервиса .....</b>	<b>42</b>
<b>Проверить подключение к сервису .....</b>	<b>42</b>
<b>Формат возвращаемых данных.....</b>	<b>43</b>
<b>Текстовый поиск по содержимому полей карт.....</b>	<b>43</b>
text.....	43
front .....	43
back.....	43
fields .....	43
templates .....	44
<b>Формат возвращаемых данных.....</b>	<b>44</b>
serial.....	44
content.....	44
<b>Работа с графической библиотекой .....</b>	<b>45</b>
<b>Запросить список доступных графических файлов .....</b>	<b>45</b>
<b>Формат возвращаемых данных.....</b>	<b>46</b>
images .....	46
totalSize.....	46
imgType .....	46
imgId.....	46
imgDescription.....	46
usageCount .....	46
<b>Запросить информацию о графическом файле из библиотеки.....</b>	<b>47</b>
<b>Формат возвращаемых данных.....</b>	<b>47</b>
cards .....	47
imgData .....	48
general .....	48
imgDTS .....	48
imgType .....	48
imgDescription.....	48
imgId.....	48
imgSize.....	49
<b>Добавить новый графический файл в библиотеку.....</b>	<b>49</b>
<b>Возвращаемые данные.....</b>	<b>49</b>
<b>Параметры .....</b>	<b>49</b>
imgDescription.....	49
imgType .....	49
imgData .....	49
<b>Изменить существующий графический файл в библиотеке.....</b>	<b>50</b>
<b>Возвращаемые данные.....</b>	<b>50</b>
imgData .....	51

imgDescription.....	51
<b>Удалить существующий графический файл из библиотеки .....</b>	<b>51</b>
Возвращаемые данные.....	51
<b>Безусловно удалить существующий графический файл из</b>	
<b>библиотеки и обновить карты .....</b>	<b>51</b>
Возвращаемые данные.....	52
<b>Автоматическая выдача карт .....</b>	<b>53</b>
Метасимволы.....	54
<b>Запросить список компаний.....</b>	<b>55</b>
<b>Запросить параметры компании .....</b>	<b>55</b>
<b>Формат возвращаемых данных.....</b>	<b>57</b>
active .....	57
linkActive.....	57
cards .....	57
pages.....	57
stdHeaders.....	57
link.....	58
type .....	58
dates .....	58
name .....	58
template .....	58
poolcards .....	58
<b>Создать новую кампанию.....</b>	<b>59</b>
<b>Редактировать параметры кампании .....</b>	<b>59</b>
<b>Удалить кампанию.....</b>	<b>61</b>
<b>Регистрация автоматически выданных карт .....</b>	<b>62</b>
<b>Получить статус регистрации карты .....</b>	<b>62</b>
<b>Обновить статус регистрации карты.....</b>	<b>63</b>
<b>Параметры .....</b>	<b>63</b>
status .....	63
<b>Обновить статус регистрации карты и сохранить данные он-лайн</b>	
<b>регистрации.....</b>	<b>63</b>
<b>Параметры .....</b>	<b>64</b>
status .....	64
saveData.....	64
group .....	64
values .....	64
<b>Получить список регистрационных групп.....</b>	<b>65</b>
<b>Получить поля данных регистрационной группы .....</b>	<b>65</b>
<b>Получить регистрационные данные для карт.....</b>	<b>66</b>
<b>Удалить регистрационные данные карт.....</b>	<b>67</b>
<b>Параметры .....</b>	<b>67</b>
registrations .....	67
<b>Общие замечания по регистрации карт .....</b>	<b>67</b>

<b>Резервирование карт.....</b>	<b>68</b>
<b>Получить информацию о ссылке резервирования карты .....</b>	<b>69</b>
<b>Формат возвращаемых данных.....</b>	<b>69</b>
status .....	69
serial.....	69
param1,2,3 .....	69
registered .....	70
maxTry.....	70
attempt .....	70
dts .....	70
link.....	70
template .....	70
<b>Получить информацию о группе ссылок резервирования карт .....</b>	<b>71</b>
<b>Формат возвращаемых данных.....</b>	<b>71</b>
<b>Параметры .....</b>	<b>71</b>
registeredOnly .....	71
start.....	72
<b>Создать ссылки на резервирование карт.....</b>	<b>72</b>
<b>Параметры.....</b>	<b>73</b>
<b>Формат возвращаемых данных.....</b>	<b>73</b>
serial.....	73
RCODE.....	73
link.....	73
<b>Обновить ссылки на резервирование карт .....</b>	<b>73</b>
<b>Параметры .....</b>	<b>74</b>
attempt .....	74
link.....	74
<b>Маркетинг .....</b>	<b>75</b>
<b>Отправить пуш сообщение .....</b>	<b>75</b>
<b>Параметры .....</b>	<b>76</b>
serials.....	76
templates .....	76
allCards.....	76
start.....	76
message.....	76
<b>Отправить динамическую акцию .....</b>	<b>76</b>
<b>Параметры .....</b>	<b>77</b>
serials.....	77
templates .....	77
allCards.....	77
start.....	77
end.....	77
label.....	77
message.....	77
body .....	77
<b>Работа со статистикой.....</b>	<b>78</b>
<b>Запросить общую статистику.....</b>	<b>78</b>
<b>Формат возвращаемых данных .....</b>	<b>79</b>
markedDeleted.....	79

created .....	79
active .....	79
inactive .....	79
finalDeleted .....	79
total (cards) .....	79
total (devices) .....	79
ios .....	79
android .....	79
<b>Запросить историческую статистику .....</b>	<b>79</b>
<b>Параметры .....</b>	<b>80</b>
start .....	80
end .....	80
<b>Запросить статус доставки пуш сообщения .....</b>	<b>81</b>
<b>Параметры .....</b>	<b>81</b>
start .....	81
serialNo .....	81
<b>Формат возвращаемых данных .....</b>	<b>81</b>
updated .....	81
delivered .....	81
serialNo .....	81
<b>Работа с уведомлениями о событиях .....</b>	<b>82</b>
<b>Получить список настроенных уведомлений .....</b>	<b>83</b>
<b>Формат возвращаемых данных .....</b>	<b>83</b>
info .....	84
method .....	84
url .....	84
event .....	84
port .....	84
user & password .....	84
<b>Настроить список уведомлений .....</b>	<b>84</b>
<b>Параметры .....</b>	<b>84</b>
method .....	84
url .....	85
event .....	85
port .....	85
user & password .....	85
<b>Формат возвращаемых данных .....</b>	<b>85</b>
<b>Групповые операции .....</b>	<b>86</b>
<b>Групповое создание карт .....</b>	<b>86</b>
<b>Параметры .....</b>	<b>87</b>
serial .....	87
template .....	87
data .....	87
<b>Формат возвращаемых данных .....</b>	<b>87</b>
serial .....	88
RCODE .....	88
link .....	88
<b>Групповое обновление карт .....</b>	<b>88</b>
<b>Параметры .....</b>	<b>89</b>



serial.....	89
push .....	89
data.....	89
<b>Формат возвращаемых данных.....</b>	<b>89</b>
serial.....	89
RCODE.....	89
<b>ПРИЛОЖЕНИЕ 1. Расширенные сообщения об ошибках.....</b>	<b>90</b>
<b>ПРИЛОЖЕНИЕ 2. Примеры использования .....</b>	<b>92</b>
Выдача новой карты .....	92
Автоматическое изменение поля.....	92
Задание индивидуального внешнего вида карт .....	92
<b>ПРИЛОЖЕНИЕ 3. Использование библиотек .....</b>	<b>93</b>
сURL.....	93
Среда 1C.....	93
<b>ПРИЛОЖЕНИЕ 4. Работа с баркодами.....</b>	<b>94</b>
Поддерживаемые типы баркодов.....	94
Общие замечания.....	94
Структура баркода.....	94
format.....	94
protect.....	94
encoding.....	95
messageType.....	95
message.....	95
signature.....	95
signatureType.....	95
show (шаблон).....	96
show (карта) .....	96
showSignature (шаблон) .....	96
showSignature (карта) .....	96
Примеры.....	97
Расширенная информация о баркоде .....	97
Стандартная информация о баркоде.....	98
Баркод шаблона .....	98
Обновление информации для баркода карты.....	98

## Предисловие

Набор интеграционных REST API (далее просто API) предназначен для автоматизации работы с электронными картами, выдаваемыми и поддерживаемыми порталом OSMICards. С помощью API становится возможным создавать дополнительные приложения по работе с шаблонами и картами, динамически изменять параметры карты, выдавать новые карты и отсылать их клиентам через смс.

*Набор API постоянно совершенствуется и дополняется, поэтому для работы всегда пользуйтесь новейшей версией документации.*

## Общие положения

Доступ к API осуществляется по принципам стандартного механизма REST (Representational state transfer) с использованием стандартных HTTP запросов. Общая идеология API предусматривает поддержку следующих HTTP команд:

GET	Запросить данные о ресурсе/ресурсах
PUT	Изменить значения ресурса
POST	Создать новый ресурс
DELETE	Удалить ресурс

На все команды возвращаются стандартные HTTP статусы:

200	Запрос обработался без ошибок, сервер вернул данные клиенту
204	Запрос обработался без ошибок, нет данных для ответа клиенту
400	Ошибка при обработке запроса; вероятно ошибка в данных
404	Неправильно указан ресурс в запросе
500	Внутренняя ошибка сервера

## Доступ к API

Адрес доступа: `https://api.osmicards.com/v2/`  
Content-Type: `application/json`

*Адрес доступа назначается компанией ОСМИ при создании учетной записи клиента и может отличаться от вышеприведенного.*

## Авторизация

Каждая команда из набора API требует авторизации. Только после успешной авторизации команда будет принята сервером к исполнению. При авторизации требуется передать `apiId` и `apiKey`. Эти два параметра выдаются зарегистрированным клиентам компанией ООО "ОСМИ" по запросу и однозначно идентифицируют клиента.

При авторизации необходимо использовать HTTP Digest Authentication. Преимущество этого метода по сравнению с другими в том, что **apiKey** никогда не передается по каналу интернета.

Примеры формата запросов к REST API с использованием программы cURL

```
curl --digest -u {apild}:{apiKey} -X GET "https://api.osmicards.com/v2/{method URL}"
```

```
curl --digest -u {apild}:{apiKey} -X PUT "https://api.osmicards.com/v2/{method URL}" --data "{данные и параметры}" --header "Content-Type: application/json"
```

Результат исполнения команд PUT и POST всегда возвращается в теле ответа в формате json и в заголовке ответа будет поле "Content-Type: application/json". При передаче параметров в запрос используется стандартная конвенция `?par1=xxx&par2=yyy ...`. Строка запроса к URL ресурсу должна быть URL-encoded.

### !!! Внимание !!!

Для правильного распознавания данных и параметров сервером, кодировка всех передаваемых через API полей должна быть в формате UTF-8!

## Поддерживаемые команды

В настоящее время поддерживаются следующие группы API команд

- Запросить информацию по картам
- Запросить информацию по шаблонам
- Создание и изменение карт
- Перемещение карт
- Создание и изменение шаблонов
- Работа со статистикой
- Проверка доставки пуш сообщений
- Отправка карт по почте, смс, QR
- Маркетинговые сообщения и акции
- Регистрация карт и регистрационные группы
- Настройка параметров сервиса
- Асинхронные уведомления по событиям с картами и устройствами
- Работа с библиотекой графических файлов для индивидуальных карт
- Групповые операции по одновременному созданию и редактированию большого количества карт

## Запросить информацию по карте

Метод: GET  
Ресурс: /passes/{номер карты}  
Параметры: необязательный extendedInfo=true

Эта команда позволяет запросить информацию о текущих значениях карты. Формат команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X GET "https://api.osmicards.com/v2/passes/osmibs0002"
```

```
curl --digest -u {apild}:{apiKey} -X GET  
"https://api.osmicards.com/v2/passes/osmibs0002?extendedInfo=true"
```

### extendedInfo

Этот параметр необязателен. Если он использовался в запросе, то информация о баркоде вернется в расширенном формате. Детали общего и расширенного формата баркода приведены в **Приложении 4**.

Если команда выполнилась без ошибок, то она вернет данные о карточке. Пример ответа приведен ниже.

## Формат возвращаемых данных

### label

Это метка поля данных. Задается в шаблоне. Используется для идентификации поля данных и должна быть уникальна в контексте шаблона карты. При обновлении данных на карте или в шаблоне доступ к полю данных осуществляется по этой метке.

### value

Это поле данных. Может быть пустым. Если поле пустое, то вернется строка "-empty-". Дополнительно, если поле пустое, то само поле и его метка не отобразятся визуально на электронной карте хотя и будут присутствовать в параметрах карты.

Данные могут быть любыми символами в кодировке UTF-8. Можно задавать гиперлинки типа <http://www.mysite.com/product1> или `<a href='www.site.com'>мой сайт</a>`. Телефон автоматически распознает гиперлинки, даты, адреса и телефонные номера при отображении карты и делает их активными, те можно по нажатию на ссылку можно произвести звонок или перейти в карту.

### altValue

Это альтернативное поле данных. Если не пустое, то при отображении карты на телефоне данные из этого поля отобразятся вместо данных из поля **value**. В остальном ведет себя как поле **value**.

### changeMsg

Поле нотификации, показываемое телефоном при изменении значения в поле "value". **В дополнении к обычному тексту или внутри**

**текста требуется добавить комбинацию символов %@.** При изменении значения в поле "value" на телефоне покажется всплывающее сообщение с текстом из поля "changeMsg", в котором %@ заменится на значение поля "value".

### status

Текущий статус регистрации карты (текст). Может принимать разные значения в зависимости от текущего состояния карточки.

### statusCode

Текущий статус регистрации карточки (цифровое значение). Может принимать разные значения в зависимости от текущего состояния карты. Возможные статусы показаны в таблице:

Статус	Значение
1	Карта создана в системе но еще ни разу не ставилась ни на один телефон.
2	Карта стоит на одном или более телефонах и хотя-бы один телефон зарегистрирован на получение автоматических обновлений.
3	После регистрации клиент выключил получение автоматических обновлений или карта удалена с телефона. Если телефонов с данной картой больше одного, то это условие должно выполняться для всех телефонов.
7	Карта помечена в системе как удаленная. Удаленные карты не получают обновлений. На карте отображается приглашение пользователю удалить карту с телефона.
10	Пользователь удалил карту с телефона после получения сообщения на карту об удалении карты из системы.

### voided

Если установлено в `true`, то телефон относится к карте как к недействительной. Визуально на карте это выражается в пропадании серийного номера карты под бар кодом и сильном снижении контраста и яркости бар кода до такой степени, что он уже не будет читаться сканерами кодов.

### expiryDate

Дата и время когда карта начнет считаться недействительной. Если значение отсутствует, то у карты нет срока окончания действия. Телефон сам отслеживает текущие дату и время и сравнивает их со значением в этом поле. Если текущие дата и время превысили это установленное значение, то визуальный эффект будет такой же, как и при установке поля "voided" в значение `true`

```

{
  "values" : [
    {
      "label" : "Скидка",
      "value" : "15%",
      "altValue" : "-empty-",
      "changeMsg" : "Ваша скидка %@"
    },
    {
      "label" : "Имя",
      "value" : "Иван Петров",
      "altValue" : "-empty-",
      "changeMsg" : "Имя изменилось на %@"
    },
    {
      "label" : "Акция",
      "value" : "Новая коллекция",
      "altValue" : "-empty-",
      "changeMsg" : "Новая акция: %@"
    },
    {
      "label" : "Адрес",
      "value" : "Большой Саввинский\ппереулок, 12, с. 10Б\пТелефон:\п+7 (499) 234 6688",
      "changeMsg" : "наш адрес изменился: %@"
    }
  ],
  "barcode" : {
    "barcode": {
      "format": "QR",
      "showSignature": false,
      "protect": true,
      "message": "-serial-",
      "show": true,
      "signature": "-empty-",
      "encoding": "iso-8859-1"
    },
    "general" : {
      "status" : "Карта зарегистрирована на телефоне",
      "voided" : false,
      "statusCode" : 2,
      "expiryDate" : "2014-06-30 23:59:59",
      "serialNo" : "osmibs0002",
      "template" : "Мой шаблон"
    },
    "locations" : [
      {
        "message" : "Мы рядом, заходите",
        "geo" : "0.33333,-1.11111"
      },
      {
        "message" : "Наш бутик на 4-м этаже, из лифта направо",
        "geo" : "0.123123,0.334455"
      }
    ]
  }
}

```

### template

Имя шаблона, которому принадлежит данная карта.

### serialNo

Серийный номер карты. Это тот же самый номер, использованный в запросе.

### limit

Максимальное количество регистраций, позволенных для этой карты. Ограничивает количество устройств с картой, которые будут получать автообновления.

### locations

Список гео позиций. Карта может иметь до 10-ти гео позиций. Каждая позиция состоит из координат точки и сообщения.

### geo

Позиция точки на карте в гео координатах. Если отсутствует, то гео локация не добавляется к телефону.

### message

Сообщение, которое будет показано на телефоне, когда телефон определил текущую позицию и она совпала с заданной в поле `geo` позицией

### barcode

Информация о баркоде карты. Детали общего и расширенного формата баркода приведены в **Приложении 4**.

### images

Информация о графических файлах, используемых картой. Пример структуры показан ниже.

Наличие информации в этой структуре означает, что у этой карты при ее отображении будут использоваться соответствующие графические файлы из библиотеки графики, а не графика из полей шаблона карты. Работа с библиотекой графических файлов описана в разделе **Работа с графической библиотекой**.

В данном примере при отображении карты на телефоне графика типа `strip` будет заменена на картинку из библиотеки с индексом `"AA11BB33CC22"`. Графика иконки и логотипа отобразятся так, как было задано в шаблоне карты. Индекс `imgId` и описание графического файла `imgDescription` являются свойствами файла из библиотеки графических файлов.

### colors

Информация об индивидуальных цветах, присвоенных соответствующим элементам карты.

Если цвета заданы (т.е. не установлены в `"-empty-"`), то они будут использованы при отображении карты вместо цветов, заданных в шаблоне.

В приведенном примере цвет фона карты будет установлен в "#AABBCC", а цвета текста и ярлыков будут установлены согласно шаблону карты. Параметры цвета используют стандартную нотацию HTML кодировки "#RRGGBB".

## Запросить список доступных шаблонов

Метод: GET  
Ресурс: /templates  
Параметры: необязательный stats=true

Эта команда запрашивает список всех шаблонов, доступных клиенту. Формат команды приведен ниже.

Формат команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X GET "https://api.osmicards.com/v2/templates"
```

Если команда выполнилась без ошибок, то она вернет список доступных шаблонов.

Если был использован параметр ?stats=true то в дополнении к списку шаблонов добавится и суммарная статистическая информация.

```
{
  "templates" : [
    "Стандартный шаблон",
    "День рождения",
    "Новый Год",
    "Специальная акция",
  ]
}

{
  "templates" : [
    "Стандартный шаблон",
    "День рождения",
    "Новый Год",
    "Специальная акция"
  ],
  "stats" : {
    "serialActive" : 21,
    "serialTotal" : 35,
    "deviceCount" : 29
  }
}
```

### serialTotal

Общее количество серийных номеров карт, принадлежащих к указанным шаблонам.

### serialActive

Общее количество серийных номеров карт, которые в настоящий момент активны, то есть карты с этими номерами установлены на телефонах пользователей и сконфигурированы на автоматическое получение обновлений.

### deviceCount

Количество телефонов пользователей, на которых установлены карты, сконфигурированные на получение автоматических обновлений. Так как



карта с одним серийным номером может потенциально быть установлена одновременно на нескольких устройствах, то количество активных карт и устройств может различаться.

## Запросить информацию о шаблоне

Метод: GET  
Ресурс: /templates/{имя шаблона}[/data]  
Параметры: необязательный stats=true  
              необязательный showKeys=true

Пример команды запроса информации о шаблоне с использованием необязательных параметров. Формат команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X GET  
"https://api.osmicards.com/v2/templates/ДР%201/data?stats=true&showKeys=true"
```

{имя шаблона} должно быть URL-Encoded.

Если команда выполнена без ошибок, то она вернет детальные данные о запрошенном шаблоне.

### Формат возвращаемых данных

Формат полей в основном такой-же, как и в команде /passes/{serial}.

Имеющиеся отличия:

1. Возвращаемые значения являются значениями по умолчанию. Это означает, что при создании новой карты на основе этого шаблона ее поля автоматически заполнятся значениями из полей шаблона.
2. формат поля `barcode` отличается, так как включает только информацию о параметрах шаблона.
3. Поле `stats` является информативным и возвращается только при использовании параметра `?stats=true`.
4. Поле `images` является информативным и возвращается только при использовании ресурса `/data`. Формат этого поля соответствует параметрам работы с изображениями и детально описан в разделе Обновить Значение Карты.

```

{
  "values" : [
    {
      "Key" : "B1",
      "label" : "Кутузовский",
      "value" : "Кутузовский пр., 14\nРезервирование столов: +7(499) 243-20-15",
      "altValue" : "-empty-",
      "changeMsg" : "Новый адрес: %@"
    },
    {
      "Key" : "H1",
      "label" : "Скидка",
      "value" : "5%",
      "altValue" : "-empty-",
      "changeMsg" : "Ваша скидка %@"
    },
    {
      "Key" : "S1",
      "label" : "Имя",
      "value" : "-empty-",
      "changeMsg" : "Имя: %@"
    }
  ],
  "colors" : { "label" : "#777099", "background" : "#C7C6CB", "foreground" : "#0E0017" },
  "barcode" : {
    "format" : "QR",
    "signatureType" : "-serial-",
    "showSignature" : true,
    "show" : true,
    "protect" : false,
    "messageType" : "-serial-",
    "encoding" : "iso-8859-1"
  },
  "general" : {
    "logoText" : "-empty-",
    "style" : "storeCard",
    "limit" : "-empty-",
    "cardApps" : {
      "AppleID" : "123456789",
      "GoogleID" : "com.company.appname",
      "AmazonID" : "-empty-",
      "AndroidTitle" : "Мое приложение"
    }
  },
  "images" : {
    "icon" : "iVBORw0KGgoAAAANSUhEUgAAAHIAAAByCAYAAA ... ",
    "logo" : "iVBORw0KGgoAAAANSUhEUgAAAGUAAABkCAYAAACfIP5q ... ",
    "strip" : "iVBORw0KGgoAAAANSUhEUgAAAmYAAAD2CAYAAAB81Uv ..."
  },
  "stats" : { "serialActive" : 21, "serialTotal" : 35, "deviceCount" : 29 },
  "locations" : [
    { "message" : "Мы рядом, заходите", "geo" : "0.3456789012,-1.1111111111" },
    { "message" : "бутик на 2-м этаже", "geo" : "0.1231231231,0.334455667788" }
  ]
}

```

## Обновить значения шаблона

Метод: PUT  
Ресурс: /templates/{имя шаблона} [/push]  
Параметры: нет

Команда изменения данных в шаблоне. Пример команды с использованием cURL:

```
curl --digest -u {apiId}:{apiKey} -X PUT  
"https://api.osmicards.com/v2/templates/День%20рождения" --data ' { "values" : [  
  { "label" : "Акция", "value" : "Новая Олимпийская коллекция", "changeMsg" : "Новая  
акция" }, { "label" : "Адрес", "value" : "Большой Саввинский\nпереулок, 12, с. 10Б\nЧасы  
работы:\n10:00-22:00 пн-вс\nТелефон:\n+7 (499) 246 6085", "changeMsg" : "наш адрес  
изменился: %@" }, { "label" : "Скидка", "value" : "-empty-", "changeMsg" : "Ваша скидка  
%@" }, { "label" : "Имя", "value" : "-empty-", "changeMsg" : "Имя изменилось на %@" } ],  
  "locations" : [ { "message" : "мы рядом", "geo" : "0.34567,-1.111111" }, { "message" : "бутик  
на 2-м этаже", "geo" : "0.123123,0.334455" } ] , "barcode" : {  
  "format" : "PDF417", "signatureType" : "-serial-", "showSignature" : true, "show" : true,  
  "messageType" : "-text-", "encoding" : "iso-8859-1" } }' --header "Content-Type:  
application/json"
```

{имя шаблона} должно быть URL-Encoded.

При использовании ресурса /push активные карты получают пуш-нотификацию. Параметр /push с этой командой следует использовать тогда, когда в шаблоне изменилась графика, цвета или тип баркода. Если же обновление затронуло только содержимое существующих полей данных шаблона, то /push не приведет к видимому эффекту на картах.

### Формат данных

Формат запроса повторяет формат возвращаемых данных из команды «Запросить информацию о шаблоне».

Если требуется изменить значение поля в шаблоне, то его нужно включить в соответствующий блок данных, посылаемый с командой. Если поле не включено в блок данных команды, то оно не обновится. Если нужно стереть данные в конкретном поле, то нужно установить его в "-empty-".

Исключение из этого правила составляют поля "locations". Нужно всегда задавать полный список требуемых гео позиций. Например, если шаблон ранее имел 8 позиций, а в новой команде были переданы только 3 позиции, в шаблон занесутся только 3 переданные позиции, а ранее установленные 5 позиций удаляться. Если передать "locations": [], то все ранее присутствующие гео позиции в шаблоне удалятся.

### Ограничения:

1. Нельзя изменить значение полей label.
2. Поля signatureType и messageType могут принимать только значения "-serial-" и "-text-"

## Формат возвращаемых данных

При успешном выполнении команды возвращается обновленный шаблон в том же самом формате, как и после исполнения команды «Запросить информацию о шаблоне».

## Создать новый шаблон

Метод: POST  
Ресурс: /templates/{имя шаблона}  
Параметры: необязательный edit=true

Команда создания нового шаблона. Пример команды с использованием cURL:

```
curl --digest -u {apiId}:{apiKey} -X POST
"https://api.osmicards.com/v2/templates/День%20рождения" --data ' { "values" : [
{ "key" : "B1", "label" : "Акция", "value" : "Новая Олимпийская коллекция", "changeMsg" :
"Новая акция" }, { "key" : "B2", "label" : "Адрес", "value" : "Большой Саввинский\ппереулок,
12, с. 10Б\пЧасы работы:\п10:00-22:00 пн-вс\пТелефон:\п+7 (499) 246 6085",
"changeMsg" : "наш адрес изменился: %@" }, { "key" : "H1", "label" : "Скидка", "value" : "-
empty-", "changeMsg" : "Ваша скидка %@" }, { "key" : "S1", "label" : "Имя", "value" : "-empty-
", "changeMsg" : "Имя изменилось на %@" } ], "locations" : [ { "message" : "мы рядом",
"geo" : "0.34567,-1.111111" }, { "message" : "бутик на 2-м этаже",
"geo" : "0.123123,0.334455" } ], "limit":1, "barcode" : { "format" : "PDF417", "signatureType" :
"-serial-", "showSignature" : true, "show" : true, "messageType" : "-text-", "encoding" : "iso-
8859-1", "protection" : true } }' --header "Content-Type: application/json"
```

{имя шаблона} должно быть URL-Encoded.

## Формат данных

Формат запроса повторяет формат возвращаемых данных из команды «Запросить информацию о шаблоне» с некоторыми изменениями:

1. Секция `general` не используется и поля `style`, `limit`, `logoText` и структуру `cardApps` следует передавать в основном теле запроса.
2. Секция `stats` не передается так является информационной.
3. В секции `Barcode` значение «protect»:true имеет смысл только при установке значения поля «messageType»:«-text-».

Значения, заданные в полях «Key», определяют в какой визуальной зоне на телефоне будет расположено соответствующее поле. Цифра в поле «Key» определяет последовательность поля в списке полей данного типа. Возможные значения этих полей приведены в таблице ниже.

Key	Местоположение поля
Hn	В правом верхнем углу на лицевой стороне карты
Pn	В поле основной картинки. Размер символов увеличен.
Sn	Под основной картинкой
An	Под строчкой полей типа S, но может и попасть в строчку S
Bn	На оборотной стороне карты

При успешном выполнении команды возвращается обновленный шаблон в том же самом формате, как и после исполнения команды «Запросить информацию о шаблоне».

## Редактировать шаблон

Метод: POST  
Ресурс: /templates/{имя шаблона}[/push]  
Параметры: обязательный `edit=true`

Команда редактирования существующего шаблона полностью повторяет формат команды «Создать шаблон», за исключением добавления параметра `?edit=true` и необязательного ресурса `/push`. Все поля, требуемые в шаблоне, должны присутствовать в теле запроса. Данные запроса полностью переписывают текущие значения шаблона. Выполнение команды можно приравнять к удалению старого шаблона и созданию полностью нового шаблона под таким-же именем.

Параметр `/push` с этой командой следует использовать тогда, когда в шаблоне изменилась графика, цвета, количество и/или имена полей, их расположение и тип баркода. Если-же обновление затронуло только содержимое существующих полей данных шаблона, то `/push` не приведет к видимому эффекту на картах.

При успешном выполнении команды возвращается обновленный шаблон в том же самом формате, как и после исполнения команды «Запросить информацию о шаблоне».

## Запросить список карт

Метод:	GET																						
Ресурс:	/passes																						
Параметры:	<table><tr><td>необязательный</td><td>activeOnly=true</td></tr><tr><td>необязательный</td><td>template={имя шаблона}</td></tr><tr><td>необязательный</td><td>stats=true</td></tr><tr><td>необязательный</td><td>page={номер страницы}</td></tr><tr><td>необязательный</td><td>fields=поле1[,поле2]...</td></tr><tr><td>необязательный</td><td>status=true</td></tr><tr><td>необязательный</td><td>graphics=true</td></tr><tr><td>необязательный</td><td>barcode=true</td></tr><tr><td>необязательный</td><td>geo=true</td></tr><tr><td>необязательный</td><td>filterVoided=true   false</td></tr><tr><td>необязательный</td><td>filterStatus={status code}</td></tr></table>	необязательный	activeOnly=true	необязательный	template={имя шаблона}	необязательный	stats=true	необязательный	page={номер страницы}	необязательный	fields=поле1[,поле2]...	необязательный	status=true	необязательный	graphics=true	необязательный	barcode=true	необязательный	geo=true	необязательный	filterVoided=true   false	необязательный	filterStatus={status code}
необязательный	activeOnly=true																						
необязательный	template={имя шаблона}																						
необязательный	stats=true																						
необязательный	page={номер страницы}																						
необязательный	fields=поле1[,поле2]...																						
необязательный	status=true																						
необязательный	graphics=true																						
необязательный	barcode=true																						
необязательный	geo=true																						
необязательный	filterVoided=true   false																						
необязательный	filterStatus={status code}																						

Команда запроса списка серийных номеров карточек, известных системе. Опционально дает возможность запросить имеющуюся статистику по картам и значение полей карт. Примеры команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X GET "https://api.osmicards.com/v2/passes"
```

```
curl --digest -u {apild}:{apiKey} -X GET  
"https://api.osmicards.com/v2/passes?activeOnly=true&template=День%20города"
```

```
curl --digest -u {apild}:{apiKey} -X GET  
"https://api.osmicards.com/v2/passes?fields=Баланс,Имя&stats=true&page=5"
```

### template

Этот параметр не обязателен. Если он использовался в запросе, то в список выданных карточек попадут только карты из указанного шаблона. Если поле не использовано, то в список попадут карты из всех шаблонов клиента. При задании имени шаблона в команде, имя шаблона должно быть URL-Encoded.

### activeOnly

Этот параметр не обязателен. Если он использовался в запросе и установлен в true, то в список попадут только активные карты, т.е. только те, которые зарегистрированы на телефонах.

### status

Этот параметр не обязателен. Если он использовался в запросе и установлен в true, то в списке для каждой карты отобразится ее статус **statusCode**.

### stats

Этот параметр не обязателен. Если он использовался в запросе и установлен в true, то формат выдаваемого списка измениться по сравнению со списком, выдаваемым командой без параметров. В список карт для каждой карты добавится секция **stats**, содержащая данные о

регистрации карты, когда она была последний раз загружена телефоном, на скольких устройствах она стоит и когда телефон последний раз зарегистрировался для получения автообновлений.

### graphics

Этот параметр не обязателен. Если он использовался в запросе и установлен в `true`, то в списке для каждой карты (при наличии у карты прикрепленной картинки и/или цветов, отличных от цветов шаблона) добавится секция **graphics** с подсекциями **images** и **colors**.

### barcode

Этот параметр не обязателен. Если он использовался в запросе и установлен в `true`, то в списке для каждой карты добавится секция **barcode**. Содержимое секции детально описано в **Приложении 4**, Работа с баркодами.

### fields

Этот параметр не обязателен. Если он использовался в запросе, то он должен быть установлен в строку названий полей шаблона карты, разделенных запятыми. Строка полей должна быть **url-encoded**.

В результате в список карт для каждой карты добавится секция `fields`, содержащая данные из запрошенных полей карты.

?fields=Владелец%20карты,Баланс,контакты

### page

Этот параметр не обязателен. Если он использовался в запросе, то он должен быть установлен в номер страницы (N), от 1 до 100000. Этот параметр используется для ограничения максимального объема информации, отправляемой сервером клиенту в результате запроса. Размер страницы ограничен в 1000 номеров карт. Если этот параметр не используется или установлен в 1, то выдадутся первые 1000 номеров. Если общее количество номеров, потенциально могущих попасть в ответ сервера, меньше чем  $N \cdot 1000$ , то вернется пустой список.

### filterVoided

Этот параметр не обязателен. Если он использовался в запросе, то в список будут отфильтрованы по статусу `voided`.

### filterStatus

Этот параметр не обязателен. Определяет статус карты. Если он использовался в запросе, то производится дополнительная фильтрация по значению этого параметра при условии когда параметр `status` установлен в `true`.

### geo

Этот параметр не обязателен. Если он использовался в запросе, то он должен быть установлен в номер страницы (N), от 1 до 100000.



Для совместимости с предыдущими версиями API, если параметры `fields` и `stats` не используются, то возвращаются все карты, без учета деления на страницы.

### Примеры использования фильтров

[/v2/passes?filterStatus=2&status=true&filterVoided=true](#)

В данном случае получаем список карт, у которых статус =2 и у которых стоит флажок `voided=true`

[/v2/passes?stats=true&filterVoided=true](#)

Получаем список карт не зависимо от статуса карты, но в список войдут только карты у которых `voided=true`

[/v2/passes?stats=true&filterVoided=false](#)

Получаем список карт не зависимо от статуса карты, но в список войдут только карты у которых `voided= false`

[/v2/passes?stats=true&filterVoided=false&status=true](#)

Получаем список карт не зависимо от статуса карты, но в список войдут только карты у которых `voided=false`

В полученном списке для каждой карты дополнительно будет отображаться статус карты.

[/v2/passes?stats=true&filterVoided=false&status=true&filterStatus=3&activeOnly=true](#)

Получаем список карт у которых статус карты = 2, но в список войдут только карты у которых `voided=true`

В полученном списке для каждой карты дополнительно будет отображаться статус карты.

Почему = 2? для совместимости с ключом `activeOnly`

Если этот ключ установлен в запросе то `filterStatus` игнорируется при обработке



## Формат возвращаемых данных - стандартный

При успешном выполнении команды возвращается список выданных клиентом карточек, возможно отфильтрованный по дополнительным параметрам `activeOnly` и `template`.

Пример ответа без использования параметров `stats` и `fields` показан слева.

## Формат возвращаемых данных - расширенный

При успешном выполнении команды возвращается список выданных клиентом карточек с дополнительной информацией `stats` и `fields`, возможно отфильтрованный по дополнительным параметрам `activeOnly` и `template`.

Пример ответа с использованием обоих параметров `stats` и `fields` показан ниже. Справа приведен пример секции ответа с использованием параметров `status=true` и `graphics=true`

```
{
  "cards" : [
    "osmibs000221",
    "osmibs000222",
    "osmibs000223",
    "osmibs000224",
    "osmibs000225",
    "osmibs000226",
    "osmibs000226",
    "osmibs000228",
    "osmibs000228"
  ]
}
```

```
{
  "statusCode" : 2,
  "serialNo" : "3050236",
  "template" : "CarvariCard"
},
{
  "statusCode" : 1,
  "graphics" : {
    "images" : {
      "strip" : "AXCR3QJQU0T6"
    }
  },
  "serialNo" : "3100471",
  "template" : "CarvariCard"
},
{
  "statusCode" : 2,
  "serialNo" : "3050237",
  "template" : "CarvariCard"
},
,
```

```

{
  "cards" : [
    {
      "fields" : {
        "Скидка" : "10%",
        "Имя" : "Вася Иванов",
        "Акция" : "Спец предложение на часы"
      },
      "statusCode" : 3,
      "stats" : {
        "downloaded" : "2013-10-10 04:49:13",
        "updated" : "2013-10-08 23:02:47",
        "registered" : "2013-10-04 16:25:19",
        "created" : "2013-08-18 06:36:24",
        "devices" : 0
      },
      "serialNo" : "OSMINIKA0001",
      "template" : "ОСМИ Ника"
    },
    {
      "fields" : {
        "Скидка" : "-notexists-",
        "Имя" : "Иван Сергеев",
        "Акция" : "-notexists-"
      },
      "statusCode" : 2,
      "stats" : {
        "downloaded" : "2014-04-22 13:47:20",
        "updated" : "2014-04-22 13:46:58",
        "registered" : "2013-08-19 15:01:23",
        "created" : "2013-08-19 14:59:03",
        "devices" : 1
      },
      "serialNo" : "OSMICP0001",
      "template" : "День Рождения"
    }, ...
  ]
}

```

### statusCode

Кодовое выражение состояния карты. Значения те-же что и в вызове «Запросить информацию по карте».

### serialNo

Серийный номер карты.

### template

Имя шаблона, которому принадлежит карта. Если в запросе установлен фильтр с именем шаблона, то это поле отсутствует.

### downloaded

Дата и время когда карта была последний раз загружена телефоном с сервера. Поле может быть пустым, если после создания карты она еще не была установлена на телефон. Если карта установлена на нескольких телефонах, то показаны дата и время самой последней загрузки с любого из устройств с картой.

### updated

Дата и время когда карта была последний раз обновлена системой. При создании карты принимает значение поля **created**.

### registered

Дата и время когда телефон, на котором установлена карта, включил режим автоматического обновления значений карты.

### created

Дата и время когда система создала карту с данным серийным номером.

### devices

Количество физических устройств, на которых установлена данная карта. Считается по количеству устройств, у которых включен режим автоматического обновления значений карты.

### limit

Количество разрешенных регистраций карты. Так как карта с одним номером может быть одновременно установлена на разных устройствах, каждое устройство пытается зарегистрировать себя на получение автоматических обновлений для установленной карты. Данный параметр ограничивает максимально возможное количество устройств, получающих автоматические обновления значений для карты.

### fields

Список запрошенных полей карты с соответствующими значениями. Если запрошенное поле не присутствует в шаблоне карты, то его значение вернется как "-notexists-". Если запрошенное поле существует в шаблоне, но в нем нет данных, то вернется "-empty-".

## Обновить значение карты

Метод: PUT  
Ресурс: /passes/{номер карты}[/push]  
Параметры: нет

/push

Этот ресурс не обязателен. Если он присутствует, то вызывает немедленную посылку нотификации на карту об обновлении данных. Нотификации будут разосланы только активным картам.

Команда обновить значение карты без посылки нотификации:

```
curl --digest -u {apild}:{apiKey} -X PUT "https://api.osmicards.com/v2/passes/osmibs000231"
--data ' { "values" : [
  { "label" : "Акция", "value" : "Новая Олимпийская коллекция" }, { "label" : "Адрес", "value"
: "Большой Саввинский\u000A\u000Aпереулок, 12, с. 10Б\u000AЧасы работы:\u000A10:00-22:00 пн-
вс\u000AТелефон:\u000A+7 (499) 246 6085" }, { "label" : "Скидка", "value" : "-empty-" }, { "label" :
"Имя", "value" : "-empty-" } ], "locations" : [ { "message" : "мы рядом с вами", "geo" :
"0.34567, -1.111111" }, { "message" : "бутик на 2-м этаже", "geo" : "0.123123, 0.334455" } ],
"void" : false, "expiryDate" : "2013-12-04T12:22:33+03:00", "barcode" : { "show" : true,
"showSignature" : true, "message" : "My barcode message", signature : "AA 12345" },
"images" : { "strip" : "AABBCC11DD", "logo" : "2233AADDCC", "colors" : { "label" : "#AA55BB",
"foreground" : "#00FF00" } } ' --header "Content-Type: application/json"
```

Команда отправить нотификацию об обновлении без изменения значений карты:

```
curl --digest -u {apild}:{apiKey} -X PUT
https://api.osmicards.com/v2/passes/osmibs000231/push"
```

*Если значения полей карты не изменились, а была запрошена отсылка нотификации, то эта ситуация будет рассматриваться как передача заведомо неправильных данных. При многократной передаче заведомо неправильных данных доступ к серверу может быть заблокирован.*

Команда обновить значение карты с отправкой нотификации об обновлении:

```
curl --digest -u {apild}:{apiKey} -X PUT
https://api.osmicards.com/v2/passes/osmibs000231/push" --data ' { "values" : [
  { "label" : "Акция", "value" : "Новая Олимпийская коллекция" }, { "label" : "Адрес", "value"
: "Большой Саввинский\u000A\u000Aпереулок, 12, с. 10Б\u000AЧасы работы:\u000A10:00-22:00 пн-
вс\u000AТелефон:\u000A+7 (499) 246 6085" }, { "label" : "Скидка", "value" : "-empty-" }, { "label" :
"Имя", "value" : "-empty-" } ], "limit" : 2, "locations" : [ { "message" : "мы рядом", "geo" :
"0.34567, -1.111111" }, { "message" : "бутик на 2-м этаже", "geo" : "0.123123, 0.334455" } ],
"void" : false, "expiryDate" : "2013-12-04T12:22:33+03:00", "barcode" : { "show" : true,
"showSignature" : true, "message" : "My barcode message", signature : "AA 12345" },
"images" : [ "strip" : "AABBCC11DD", "logo" : "2233AADDCC", "colors" : { "label" : "#AA55BB",
"foreground" : "#00FF00" } } ' --header "Content-Type: application/json"
```

При необходимости обновления большого количества карт, например при пакетной обработке результатов дня, рекомендуется пользоваться методами из раздела “Групповые операции”.

### Формат данных

По формату пакета данных команда очень похожа на команду обновления шаблона `/templates/{имя шаблона}`. Принципиальная разница в том, что эта команда обновляет текущие значения карты, а при обновлении шаблона обновляются значения по умолчанию. Значения по умолчанию используются только в момент выдачи новой карты, когда они автоматически копируются в текущие значения вновь создаваемой карты. Так же эта команда не обновляет поле `"changeMsg"`, которое задается в шаблоне.

Дополнительные поля описаны ниже.

#### void

Это поле не обязательное. Устанавливает поле `"voided"` в `true` или `false`. Смотри команду получить информацию о карте `/passes/{номер карты}` для механизма работы поля `"voided"`.

**Примечание для iPhone:** после установки этого поля в `true` и отправке нотификации на телефон, телефон перестает обрабатывать полученные нотификации на изменение значений полей. Для возобновления обработки обновлений необходимо установить это поле в `false` и затем отправить нотификацию на телефон. После этого все последующие нотификации на обновление значений начнут обрабатываться телефоном как и прежде.

#### expiryDate

Это поле не обязательное. Устанавливает значение поля `"expiryDate"`. Если нужно убрать блокировку карты по дате и времени, то необходимо в этом поле задать значение `"-empty-"`. Смотри команду "получить информацию о карте" `/passes/{номер карты}` для механизма работы поля `"expiryDate"`. Значение даты и времени должно соответствовать формату [W3C](#)

#### barcode

Это поле необязательное. Позволяет задать некоторые параметры баркода, уникальные для выбранной карты. Детали общего и расширенного формата баркода приведены в **Приложении 4**.

#### images

Если значение `imgId` установлено в `"-empty-"`, то при отображении карты для данного элемента будет использована графика, заданная для него в шаблоне. Сам графический файл из библиотеки графики при этом не удаляется.

При присваивании `imgId` элементу карты, система проверяет совпадение типа элемента (`logo`, `strip` и тд.) у файла с `imgId` из

библиотеки с типом в запросе. При несовпадении типов запрос будет отвергнут с соответствующей ошибкой. Смотри раздел **Работа с графической библиотекой** для полного описания работы с библиотекой графики.

### colors

Если значение цвета установлено в "-empty-", то при отображении карты для данного элемента будет использован цвет, заданный для него в шаблоне.

### Формат возвращаемых данных

При успешном выполнении команды возвращается обновленное содержимое карты в формате, соответствующему запросу "получить информацию о карте" `/passes/{номер карты}`

## Обновить все карты в шаблоне

Метод:	PUT
Ресурс:	<code>/passesintemplate/templatename[/push]</code>
Параметры:	нет

`/push`

Этот ресурс не обязателен. Если он присутствует, то вызывает немедленную посылку нотификации на все карты, обновленные данной командой. Нотификации будут разосланы только активным картам.

Эта команда повторяет формат команды "обновить карту", но ее действие применяется не к одной карте, а ко всем картам, заданным именем шаблона `templatename`.

При успешном исполнении команды возвращается HTTP код 200 и **json** структура с общим числом обновленных карт и количеством устройств, на которые была отослана нотификация об обновлении. Если опция `push` не была использована, то количество нотификаций в структуре будет равно 0.

### Примечание

Действие этой команды не распространяется на параметры баркодов карт.

## Создать новую карту

Метод: POST  
Ресурс: /passes/{номер карты}/{имя шаблона}  
Параметры: необязательный ?withValues=true

Команда создать новую карту на основе заданного шаблона. Пример команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X POST
"https://api.osmicards.com/v2/passes/osmibs00333/День%20Рождения"

curl --digest -u {apild}:{apiKey} -X POST
"https://api.osmicards.com/v2/passes/osmibs00333/День%20Рождения?withValues=true" --
data ' { "values" : [ { "label" : "Акция", "value" : "Новая Олимпийская коллекция" }, { "label"
: "Адрес", "value" : "Большой Саввинский\нпереулок, 12, с. 10Б\нЧасы работы:\n10:00-
22:00 пн-вс\нТелефон:\n+7 (499) 246 6085" }, { "label" : "Скидка", "value" : "-empty-" }, {
"label" : "Имя", "value" : "-empty-" } ], "limit" : 2, "locations" : [ { "message" : "мы рядом",
"geo" : "0.34567,-1.111111" }, { "message" : "бутик на 2-м этаже", "geo" :
"0.123123,0.334455" } ], "void" : false, "expiryDate" : "2013-12-04T12:22:33+03:00",
"barcode" : { "show" : true, "showSignature" : true, "message" : "My barcode message",
signature : "AA 12345" }, "images" : [ "strip" : "AABBCC11DD", "logo" : "2233AADDCC" ],
"colors":{"label":"#AA55BB", "foreground" : "#00FF00"} }' --header "Content-Type:
application/json"
```

Карта создается с запрошенным серийным номером. Значения полей карты при создании карты копируются из значений полей шаблона. Если в запросе задан номер уже существующей карты, то вернется HTTP сообщение об ошибке 400 и код ошибки 319 в теле ответа.

Если используется параметр `withValues=true`, то в теле запроса ожидается блок данных для заполнения полей карты вместо соответствующих значений, скопированных в карту из шаблона. Это позволяет в процессе создания карты сэкономить на вызове “Обновить карту”.

Формат передаваемого блока данных полностью соответствует формату данных команды “Обновить карту”.

При необходимости создания большого количества карт за один раз, например для массовой рассылки новых карт клиентам, предпочтительнее пользоваться методами из раздела “Групповые операции”. Это позволит существенно сэкономить время и ресурсы.

### Формат возвращаемых данных

При успешном выполнении команды возвращается содержимое новой карты в формате, соответствующему запросу “получить информацию о карте” /passes/{номер карты}

## Удалить карту

Метод: DELETE  
Ресурс: /passes/{номер карты}[/push]  
Параметры: нет

/push

Этот ресурс не обязателен. Если он присутствует, то вызывает немедленную посылку нотификации на карту об удалении. Нотификации будут разосланы только активным картам.

```
curl --digest -u {apild}:{apiKey} -X DELETE  
"https://api.osmicards.com/v2/passes/osmibs000231/push"
```

Следует помнить, что невозможно удаленно стереть карту с телефона. Пользователю только можно предложить удалить карту. Так-же невозможно достоверно проконтролировать, удалил ли пользователь карту с телефона или просто выключил автоматические обновления для карты.

Поэтому при вызове команды «Удалить карту», карта физически не удаляется из базы данных карт, а принимает статус удаленной. Внешний вид удаленной карты изменяется и приглашает пользователя физически удалить карту с телефона. Удаленные карты не получают обновления.

*Номер удаленной таким образом карты не может быть использован для создания новой карты.*

Если требуется полностью удалить карту из системы с целью использования старого серийного номера карты для создания новой карты, обратитесь в службу технической поддержки. Но в большинстве случаев достаточно обновить карту с новым содержанием без удаления.

### Возвращаемые данные

При успешном исполнении команды возвращается HTTP статус 204 и пустое тело ответа.

## Отправить ссылку на загрузку карты по СМС

Метод: GET  
Ресурс: /passes/{номер карты}/sms/{телефон}  
Параметры: необязательный ?message=aaabbccc...  
?sender=Romashka...

Команда отправить ссылку на загрузку карты через смс на мобильный телефон. Пример команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X GET  
"https://api.osmicards.com/v2/passes/osmibs00333/sms/79256662233?message=карта%20по%20ссылке%20%7Blink%7D"
```



## Формат данных

{ телефон }

Мобильный телефон пользователя. *Рекомендуется использовать нормализованный формат 79993332211, где первая цифра (цифры) являются кодом страны.* Допустимые форматы номеров:

- +7(999) 555-44-22
- +7-999-555-44-22
- 7 999 555 44 22
- другие похожие

При отсылке на Российские мобильные номера телефонов допускается использовать формат номера телефона, начинающегося с первой цифрой 8, например 8-925-555-33-22. Тогда номер будет распознан как Российский. СМС сообщения отправляются через СМС-провайдера, зарегистрированного в учетной записи пользователя.

### message

Этот параметр не обязателен. Если он использовался в запросе, то смс будет отправлено с указанным в параметре текстом вместо текста, заданного в параметре учетной записи пользователя. В тексте должна обязательно присутствовать последовательности символов {link}. Эта последовательность при отправке заменится на фактическое значение ссылки на загрузку карты. Текст должен быть в формате UTF-8 и URL-Encoded. Обязательно оставляйте минимум один пробел между текстом и {link} с обеих сторон для правильного распознавания телефоном URL ссылки на карту!

### sender

Этот параметр не обязателен. Если он использовался в запросе, то смс будет отправлено с указанным именем отправителя вместо имени, привязанного к учетной записи пользователя. Для успешной отправки смс с альтернативным именем отправителя, СМС Провайдер должен поддерживать возможность выбора отправителя при работе с одной учетной записью.

## Отправить ссылку на загрузку карты по email

Метод:	GET
Ресурс:	/passes/{номер карты}/email/{email}
Параметры:	необязательный from=xxxxx...
	необязательный fromName=yyyyy...
	необязательный subject=zzzzz...
	необязательный body=aaabbccc...
	необязательный attachment=true

Команда отправить ссылку на карту и (опционально) саму карту как аттачмент на электронную почту клиента. Пример команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X GET  
"https://api.osmicards.com/v2/passes/osmibs00333/email/ivan%40mail.ru?attachement=true  
&subject=ваша%20карта"
```

Все необязательные параметры при использовании должны быть в кодировке UTF-8 и URL-encoded. Если необязательный параметр не задан в запросе, то его значение возьмется из параметров учетной записи пользователя.

**Этот метод оставлен только для совместимости с предыдущими версиями API. Рекомендуется использовать новый метод отправки через POST!**

### Формат данных

`{email}`

URK-encoded адрес электронной почты получателя. Электронная почта клиенту отправляется через почтовый сервер и почтовый ящик, зарегистрированные в учетной записи пользователя.

#### from

Этот параметр не обязателен. Если он использовался в запросе, то к конверту письма будет добавлен параметр Reply-To с заданным значением электронного адреса. Это может использоваться в случае когда нужно выделить специальный адрес для ответов клиентов, получивших карту по электронной почте.

#### fromName

Этот параметр не обязателен. Если он использовался в запросе, то к конверту письма в поле Reply-To будет добавлено текстовое представление адреса для ответа на письмо. Это текстовое представление будет показано почтовой программой клиента в поле Reply-To вместо адреса электронной почты.

#### subject

Этот параметр не обязателен. Если он использовался в запросе, тема письма будет задана текстом, переданным в этом параметре.

#### body

Этот параметр не обязателен. Если он использовался в запросе, то текст письма будет сформирован из содержимого этого параметра. Текст может задаваться в html формате. Для этого текст должен быть обернут в тэги `<html><body>...</body></html>`. Допустимо использование встроенных CSS стилей `style='...'` и других допустимых html тэгов в теле письма. Секции `<head>` и `<style>` не поддерживаются.

## Метасимволы

В тексте письма могут быть использованы метасимволы, которые при отправке заменяются на актуальные параметры. Допустимые метасимволы приведены ниже:

### {link}

заменится на актуальную ссылку на карту. Если текст письма в формате HTML, то заменится на `<a href=aa.bb.cc/dd> aa.bb.cc/dd </a>`. В противном случае заменится просто на `aa.bb.cc/dd`

### {linkurl}

Заменится на `aa.bb.cc/dd`, являющейся актуальной ссылкой на скачивание карты.

### {linkqr}

Заменится на ссылку `aa.bb.cc/dd.png`

Может использоваться в структуре `<img src='{linkqr}'>`. В этом случае покажет QR код для сканирования телефоном. В QR код будет зашита ссылка на скачивание карты.

### {QR}

Если текст письма в в формате HTML, то заменится на конструкцию `` для встраивания картинки в тело письма в URI формате.

В противном случае картинка QR кода на скачивание карты прикрепится к письму в виде вложения.

Некоторые почтовые клиенты (google mail и др) не отображают картинки в URI формате. Поэтому рекомендуется использовать {linkqr}

## Отправить ссылку на загрузку карты по email (2)

Метод: POST  
Ресурс: /passes/{номер карты}/email/{email}  
Параметры: нет

Команда отправить ссылку на карту и (опционально) саму карту как вложение на электронную почту клиента. Команда очень похожа на предыдущую команду отправки через GET, но использует метод POST с теми же опциональными параметрами, но теперь в json структуре в теле запроса. Пример команды с использованием cURL:

```
curl --digest -u {apid}:{apiKey} -X POST  
"https://api.osmicards.com/v2/passes/osmibs00333/email/ivan%40mail.ru" --data  
'{"from":"test@test.com", "useAttachment":true}' --header "Content-Type: application/json"
```

Использование этого метода рекомендуется вместо предыдущего метода GET, так как размер текста письма не ограничивается максимальной длиной строки запроса браузера в методе GET. Так же многие корпоративные прокси сервера ограничивают длину запроса еще больше, вплоть до 200-300 символов.

## Запросить ссылку на загрузку карты

Метод: GET  
Ресурс: /passes/{номер карты}/link  
Параметры: необязательный type={URL | QR}

Команда получить ссылку на загрузку карты для последующей ее отправки по СМС, по почте, показу на вэб сайте и др. Пример команды с использованием cURL:

```
curl --digest -u {apiId}:{apiKey} -X GET  
"https://api.osmicards.com/v2/passes/osmibs00333/link?type=QR"
```

### Формат данных

{номер карты}

Серийный номер карты.

#### type

Необязательный параметр. Задаёт требуемый формат получаемой ссылки. Может принимать следующие значения:

Отсутствует	- запросить строку URL
type=URL	- запросить строку URL
type=QR	- запросить строку URI с mime типом <code>img/png</code> в кодировке Base64.

### Формат возвращаемых данных

При успешном выполнении команды возвращается ссылка на карту в запрошенном формате. Пример ответа при запросе ссылки в формате URL показан ниже.

```
{  
  "link" : "https://a.osmicards.com/34e5345352w253434632e"  
}
```

Это актуальная ссылка на скачивание карты. Можно добавить к полученному URL суффикс `.png` и тогда ссылка станет указывать на картинку QR кода, в который будет зашита актуальная ссылка на загрузку карты.

Если был запрошен формат QR, то сервер сгенерирует картинку QR кода со ссылкой на загрузку карты, закодирует ее в формате URI ресурса и вернет в теле ответа. URI ресурс можно сразу показывать на веб страничках, используя тэг ``, или сохранить (после декодирования из Base64) в графическом файле формата PNG для последующей обработки. Приложения PassBook и PassWallet уже имеют встроенные сканеры QR кодов и позволяют установить новую карту на телефон при сканировании правильного QR кода.

Пример ответа при запросе ссылки в формате QR приведен ниже.

```
{
  "link" : "data:img/png;base64,iVBORw0KGgoAAAANSUgAAA...IQAAACE5CYII="
}
```

## Переместить карту на другой шаблон

Метод: PUT  
Ресурс: /passes/move/{номер карты}/{шаблон} [/push]  
Параметры: нет

Команда переместить карту с одного шаблона на другой. После перемещения графика, цвета, баркод и количество полей у карты будут определяться новым шаблоном, если только они не были индивидуализированы с использованием графических файлов из библиотеки графики или индивидуальными цветами элементов. Поля "label" и "changeMsg" карты тоже будут определяться новым шаблоном. Поля "value" и "location" будут определяться содержимым полей карты.

Команда переместить карту без отправки нотификации:

```
curl --digest -u {apild}:{apiKey} -X PUT
"https://api.osmicards.com/v2/passes/osmibs00333/День%20рождения"
```

Команда переместить карту с отправкой нотификации об обновлении:

```
curl --digest -u {apild}:{apiKey} -X PUT
"https://api.osmicards.com/v2/passes/osmibs00333/День%20рождения/push"
```

/push

Этот ресурс не обязателен. Если он присутствует, то вызывает немедленную посылку нотификации на карту об обновлении данных. Нотификации будут разосланы только активным картам.

## Отправить PIN-код по СМС

Метод: POST  
Ресурс: /activation/sendpin/{телефон}  
Параметры: нет

Команда установить верификационный пинкод и отправить его по СМС клиенту. Далее можно проверить введенный клиентом пинкод на соответствие отправленному коду, используя запрос “Проверить PIN-код”. Пример команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X POST  
"https://api.osmicards.com/v2/activation/sendpin/79256662233" --data ' { "smsText" : "Ваш  
пинкод для транзакции {pin}", "length": 5 }' --header "Content-Type: application/json"
```

в теле запросе можно не указывать json данные. Тогда сообщение сформируется со стандартным текстом “Ваш пинкод {pin}” (см ниже).  
Срок жизни пинкода равен 10 минутам с момента его выдачи, и максимальное число попыток ввода пинкода ограничено 5-ю попытками.

### Формат данных

{ телефон }

Мобильный телефон пользователя. *Рекомендуется использовать нормализованный формат 79993332211, где первая цифра (цифры) являются кодом страны.* Допустимые форматы номеров:

- +7(999) 555-44-22
- +7-999-555-44-22
- 7 999 555 44 22
- другие похожие

При отсылке на Российские мобильные номера телефонов допускается использовать формат номера телефона, начинающегося с первой цифрой 8, например 8-925-555-33-22. Тогда номер будет распознан как Российский. СМС сообщения отправляются через СМС-провайдера, зарегистрированного в учетной записи пользователя.

### smsText

Этот параметр не обязателен. Если он использовался в теле запроса, то смс будет отправлено с указанным в параметре текстом вместо стандартного текста. Если параметр используется то в тексте должна присутствовать последовательность символов {pin}, которая при отправке заменится на случайный цифровой код длиной в 4 символа, или заданной длины.

### length

Этот параметр не обязателен. Если он использовался в теле запроса, то значение переменной {pin} при отправке будет длиной в length цифр. Это параметр может принимать значения от 4 до 6 включительно.

Если этот параметр не задан или его размер вне границ, то длина пинкода будет равна 4 символа.

### Возвращаемые данные

При успешном исполнении команды вернется значение token, которое нужно будет использовать в последующей команде проверки введенного пинкода:

```
{  
  "token" : "29516681a0d7022ed3de25cbb7862fca689fb97c"  
}
```

## Проверить ранее выданный PIN-код

Метод: POST  
Ресурс: /activation/checkpin  
Параметры: нет

Команда позволяет проверить пинкод, выданный ранее с использованием команды «Отправить pin-код по СМС». Пример команды с использованием cURL:

```
curl --digest -u {apiId}:{apiKey} -X POST "https://api.osmicards.com/v2/activation/checkpin"  
--data ' { "token" : "75a288ffbeffcad54cd1632a021192a6ac93abc3", "pin": "12345" } ' --header  
"Content-Type: application/json"
```

в теле запросе можно не указывать json данные. Тогда сообщение сформируется со стандартным текстом «Ваш пин код {pin}» (см ниже).

### token

Этот параметр обязателен. Его нужно установить в значение, полученное ранее из команды «Отправить pin-код по СМС».

### pin

Этот параметр обязателен. Он устанавливается в значение, введенное пользователем после получения пинкода по СМС.

### Возвращаемые данные

При успешном исполнении команды возвращается HTTP статус 204 и пустое тело ответа. При неуспешной проверке пинкода возвращается HTTP статус 400 и соответствующие сообщения об ошибках:

```
{"RCODE":357,"RMESSAGE":"FAILED"}  
{"RCODE":357,"RMESSAGE":"FAILED: 3 LEFT"}  
{"RCODE":357,"RMESSAGE":"FAILED: TOO MANY ATTEMPTS"}
```

Если при запросе будет передан неправильное значение `token`, то всегда будет выдаваться первый вариант ответа.

Если `token` был передан правильно, но пинкод не верен, то будут выдаваться 2 остальных варианта ответа до того момента, пока количество попыток не превысит максимально разрешенное (5). После этого будет опять выдаваться первый вариант ответа.

## Запросить общие параметры сервиса

Метод: GET  
Ресурс: /defaults[/all | /sms | /email | /help]  
Параметры: нет

Команда позволяет запросить некоторые параметры сервиса, установленные при создании учетной записи клиента. Возможно запросить следующие параметры сервиса:

- шаблон текста смс рассылок
- дополнительный текст на страничке загрузки карты.

Команда запросить все доступные общие параметры:

```
curl --digest -u {apild}:{apiKey} -X GET "https://api.osmicards.com/v2/defaults"  
curl --digest -u {apild}:{apiKey} -X GET "https://api.osmicards.com/v2/defaults/all"
```

## Формат возвращаемых данных

При успешном выполнении команды возвращаются доступные общие параметры сервиса. Пример ответа при запросе с использованием параметра `/all` показан ниже.



### helper : text : ru

Это поле может быть возвращено как "-empty-" (пустое) и тогда оно не будет показываться (использоваться) на страничке загрузки карты. Если поле установлено, то оно будет показываться вверху странички, перед основным текстом, при выборе русской странички.

### helper : text : en

Это поле может быть возвращено как "-empty-" (пустое) и тогда оно не будет показываться (использоваться) на страничке загрузки карты. Если поле установлено, то оно будет показываться вверху странички, перед основным текстом, при выборе английской странички.

```
{
  "helper" : {
    "text" : {
      "ru" : "<p>это дополнительный текст</p>",
      "en" : "<p>this is an additional text</p>"
    }
  },
  "email" : {
    "attachmentEncoding" : "base64",
    "subject" : "Карта в Passbook",
    "body" : "<html><body><p>Ваша электронная карта готова.
      </p><p>Если вы открыли это письмо на своем мобильном телефоне,
      то вы можете установить карту в приложение Passbook на iPhone
      или PassWallet на Android, нажав на иконку аттачмента,
      или загрузить ее пройдя по ссылке {link}</p></body></html>",
    "secure" : "tls",
    "attachmentName" : "pass.pkpass",
    "from" : "no-reply@osmicards.com",
    "fromName" : "OSMICards",
    "host" : "mail.osmicards.com",
    "attachmentType" : "application/vnd.apple.pkpass",
    "pass" : "*****",
    "enabled" : true,
    "smtpAuth" : true,
    "user" : "no-reply@osmicards.com",
    "port" : 587
  },
  "sms" : {
    "sender" : "OSMICARDS",
    "pass" : "*****",
    "apiKey" : "-empty-",
    "enabled" : true,
    "provider" : "Devino Telecom",
    "text" : "Your card is ready {link}",
    "user" : "Osmicards"
  }
}
```

### sms

Эта секция возвращает параметры отправки смс сообщений, установленные при настройке учетной записи пользователя. Поле text может быть динамически задано во время отправки сообщения. Последовательность {link} заменится во время отправки смс на актуальную ссылку на карту.

### email

Эта секция возвращает параметры отправки email сообщений, установленные при настройке учетной записи пользователя. Поля

subject, body, from, fromName и наличие или отсутствие аттачмента карты в письме могут быть динамически заданы во время отправки сообщения. Последовательность {link} заменится во время отправки email на актуальную ссылку на карту.

Если использован один из необязательных параметров /sms, /email или /help, то возвращаемые данные будут включать только соответствующую секцию данных.

## Изменить общие параметры сервиса

Метод: PUT  
Ресурс: /defaults  
Параметры: нет

Команда позволяет изменить некоторые параметры сервиса, установленные при создании учетной записи. В этой версии API возможно изменить шаблон текста для всех смс рассылок и добавить текст к страничке загрузки карты. Пример команды представлен ниже.

```
curl --digest -u {apild}:{apiKey} -X PUT "https://api.osmicards.com/v2/defaults" --data ' {
  "helper" : { "text" : { "ru" : "<p>Это дополнительный текст</p>", "en" : "<p>This is an
additional text</p>" } }, "sms" : { "text" : "Ваша карта готова. Кликните на линк {} для
загрузки карты на телефон." } }' --header "Content-Type: application/json"
```

Если не требуется обновлять какую-либо секцию, то ее можно не включать в тело передаваемого запроса. Например, если требуется только убрать дополнительный текст с русской странички загрузки и оставить все остальное без изменений, то можно использовать команду

```
curl --digest -u {apild}:{apiKey} -X PUT "https://api.osmicards.com/v2/defaults" --data ' {
  "helper" : { "text" : { "ru" : "-empty-" } } }' --header "Content-Type: application/json"
```

В секции helper:text разрешены только HTML тэги <p>, <br>, <strong>. Все остальные тэги будут отфильтрованы.

## Проверить подключение к сервису

Метод: GET  
Ресурс: /ping  
Параметры: нет

Команда позволяет проверить подключение к сервису и узнать текущую версию API. Пример команды представлен ниже.

```
curl --digest -u {apild}:{apiKey} -X GET "https://api.osmicards.com/v2/ping"
--header "Content-Type: application/json"
```

## Формат возвращаемых данных

При успешном выполнении команды возвращаются доступные общие параметры сервиса.

```
{
  "company" : "000 Ромашка",
  "api" : "1.7"
}
```

## Текстовый поиск по содержимому полей карт

Метод: POST  
Ресурс: /search/passes  
Параметры: нет

Команда позволяет осуществить полноценный поиск заданного текста по содержимому полей всех существующих карт. Возможно ограничить поиск по заданной стороне карты, списку шаблонов карт и полей карт. Пример команды представлен ниже.

```
curl --digest -u {apiId}:{apiKey} -X PUT "https://api.osmicards.com/v2/search/passes" --data '{
  "text" : "Иван С", "front" : true, "back" : false, "fields": [ "Владелец" , "Имя" ], "templates" : [
    "Стандартная карта", "Корпоративный" ] }' --header "Content-Type: application/json"
```

### text

Обязательный параметр. Содержит текст для поиска в полях карты. При задании текста «иван» поиск вернет поля со словами «Иванов», «Диван», «диван», «дИван» и тп. Возвращается полное содержимое поля, содержащее искомый текст. Если искомый текст найден более чем в одном поле одной и той-же карты, вернется только одно найденное поле.

### front

Необязательный параметр. Позволяет включить или отключить поиск в полях лицевой стороны карты. Если не задан в запросе, то подразумевается true.

### back

Необязательный параметр. Позволяет включить или отключить поиск в полях оборотной стороны карты.

### fields

Необязательный параметр. Формат array[]. Позволяет задать поля карты, в которых искать требуемый текст. Если не задан или пустой список, то поиск осуществляется во всех полях карты с учетом остальных фильтров.

## templates

Необязательный параметр. Формат `array[]`. Позволяет задать список имен шаблонов карт. Только в картах, выданных на шаблонах из списка заданных шаблонов, будет осуществляться поиск заданного текста. Если не задан или пустой список, то поиск осуществляется во всех картах с учетом остальных фильтров.

## Формат возвращаемых данных

При успешном выполнении команды возвращается список номеров карт и содержимых полей.

### serial

содержит номер карты, в одном из полей которой был найден требуемый текст.

### content

содержит содержимое поля, в котором был найден требуемый текст.

Пример ответа при поиске текста «4%»

```
[{
  "serial" : "000000",
  "content" : "Есть акция 4%"
},
{
  "serial" : "256",
  "content" : "Есть акция 4%"
},
{
  "serial" : "tb005",
  "content" : "Есть акция 4%"
},
{
  "serial" : "tb004",
  "content" : "Есть акция 4%"
}]
```

Если не задан текст для поиска, или оба фильтра `back` и `front` эквивалентны `false`, то поиск вернет ошибку.

## Работа с графической библиотекой

Внешний вид карты определяется выбранным типом карты и заданными графическими и цветовыми параметрами. Тип карты, графика и цвета элементов задаются в шаблоне карты. Шаблон карты контролируется из личного кабинета.

В большинстве случаев шаблон карты создается в соответствии с требованиями компании-эмитента к внешнему виду и стилю и его графический дизайн не меняется с течением времени. Все карты, создаваемые компанией с использованием выбранного шаблона, легко идентифицируются как карты компании, так как используют унифицированный визуальный стиль и графику.

В некоторых случаях может потребоваться изменить индивидуальный дизайн карты без изменения основного корпоративного шаблона компании. Одним из примеров может быть ситуация с использованием карты в качестве пропуска или клубной карты, несущей на себе фотографию пользователя. В таком случае уже невозможно просто использовать унифицированный шаблон компании для создания таких карт, так как в этом случае графика на всех картах будет одинаковой, задаваемой шаблоном карты.

В таких ситуациях более подходит изменение индивидуальных параметров карты, таких как отображаемая графика и цветовая палитра, без изменения основного корпоративного шаблона и без изменения принадлежности карты к корпоративному шаблону. Если карте были присвоены свои индивидуальные визуальные параметры, то они будут использоваться при отображении карты вместо параметров шаблона карты.

Присвоение карте индивидуальных графических параметров (`images`, `colors`) производится с использованием стандартной API команды `"PUT /passes/serial"` и рассмотрено в соответствующем разделе API.

Для того, чтобы карте можно было присвоить индивидуальный элемент графики, соответствующий графический файл должен быть доступен в библиотеке графических элементов. Функции API, предназначенные для работы с библиотекой графики, описаны ниже.

## Запросить список доступных графических файлов

Метод: GET  
Ресурс: /images  
Параметры: нет

Эта команда запрашивает список всех доступных графических файлов из библиотеки.

Формат команды с использованием cURL:

```
curl --digest -u {apiId}:{apiKey} -X GET "https://api.osmicards.com/v2/images"
```

Если команда выполнена без ошибок, то она вернет список доступных графических файлов. Пример ответа приведен ниже.

```
{
  "images" : [
    {
      "imgType" : "logo",
      "imgDescription" : "Совместный логотип",
      "imgId" : "3KB1NKUMSXXM",
      "usageCount" : 5
    },
    {
      "imgType" : "strip",
      "imgDescription" : "Акция продажи 1427",
      "imgId" : "3SYFUXNUP0K6",
      "usageCount" : 240
    },
    {
      "imgType" : "strip",
      "imgDescription" : "Альтернативная акция 1234",
      "imgId" : "3T9CB03WU26U",
      "usageCount" : 0
    }
  ],
  "totalSize" : 512817
}
```

## Формат возвращаемых данных

### images

Список всех доступных для использования графических файлов

### totalSize

Суммарный размер всех доступных графических файлов

### imgType

Тип графического файла. Задается при добавлении файла в библиотеку. Используется как средство контроля при обновлении параметров карты и привязке индивидуального графического файла к карте.

### imgId

Уникальный идентификатор файла в библиотеке. Создается системой при добавлении файла в библиотеку. Используется при обновлении параметров карты и привязке индивидуального графического файла к карте.

### imgDescription

Необязательное описание графического файла в библиотеке. Задается при добавлении файла в библиотеку или при изменении файла.

### usageCount

Справочное значение. Показывает количество карт, использующих данный графический файл.

## Запросить информацию о графическом файле из библиотеки

Метод: GET  
Ресурс: /images/{imgId}  
Параметры: необязательные /cards, /data

Эта команда запрашивает информацию о графическом файле из библиотеки.

Формат команды с использованием cURL:

```
curl --digest -u {apiId}:{apiKey} -X GET "https://api.osmicards.com/v2/images/3KB1NKUMSXXM"
```

/cards

Этот параметр необязателен. Если он присутствует, то дополнительно возвращается в поле "cards" список серийных номеров карт, использующих данный графический файл.

/data

Этот параметр не обязателен. Если он присутствует, то дополнительно возвращается поле "imgData" с содержимым файла из библиотеки и закодированным в формате base-64.

### Формат возвращаемых данных

Пример ответа приведен ниже. Следует учесть, что только поле "general" возвращается при использовании вызова без дополнительных параметров. Поля "cards" и "imgData" будут возвращены только при использовании соответственных параметров как описано выше.

```
{
  "cards" : [
    "123456",
    "334455"
  ],
  "general" : {
    "imgDTS" : "2014-05-24 09:31:30",
    "imgType" : "strip",
    "imgDescription" : "Акция 2014/05 спец. 1234",
    "imgId" : "3KB1NKUMSXXM",
    "imgSize" : 130517
  },
  "imgData" : "iVBORw0KGgoAAAANSUheUgAAAAoAAAAK...6AAAAAAE1FTkSuQmCC"
}
```

### cards

Список серийных номеров карт, использующих данный графический файл библиотеки.

### imgData

Графический файл из библиотеки, закодированный в формате base-64. Для просмотра графического файла эти данные нужно декодировать и сохранить как двоичный (binary) файл с расширением ".png".

Для прямого отображение файла на вэб страничке можно не декодировать и не сохранять это поле, а просто добавить в начало строки 'data:img/png;base64,' и затем использовать полученную строку в CSS через javascript DOM:

```
background-image: url('data:img/png;base64,iVB0Rw...')
```

### general

Секция общей информации о графическом файле в библиотеке.

### imgDTS

Время добавления файла в библиотеку или время его последнего изменения. Задается системой.

### imgType

Тип графического файла в терминах Пассбук. Поддерживаемые типы зависят от выбранного стиля карты. Возможные типы:

*strip, logo, icon, background, thumbnail*

Для стандартных электронных карт и купонов используются типы *strip, logo* и *icon*

#### Замечания:

- Желаемый тип задается пользователем при сохранении нового графического файла в библиотеке.
- При обновлении карты через PUT /passes/{serial} указывается параметр тип файла (imgType). Этот тип должен совпадать с типом файла, хранимом в библиотеке с указанным imgId.
- По запросу с ключом /data возвращается примерно на 30% больше данных, чем фактическая длина графического файла. Это связано с особенностями base-64 кодировки.

### imgDescription

Необязательная произвольная текстовая информация о графическом файле, хранящемся в библиотеке. Задается при добавлении файла в библиотеку или при использовании функции изменения библиотечного файла.

### imgId

Уникальный идентификатор файла в библиотеке. Создается системой при добавлении нового файла в библиотеку. Используется в вызовах



изменения содержимого карты и в функциях работы с библиотекой файлов.

### imgSize

Справочный параметр физического размера файла. Создается системой при добавлении нового файла в библиотеку и автоматически изменяется при использовании функции изменения библиотечного файла.

## Добавить новый графический файл в библиотеку

Метод: POST  
Ресурс: /images  
Параметры: нет

Эта команда добавляет новый графический файл в библиотеку для дальнейшего его использования в командах работы с картами.

Формат команды с использованием cURL:

```
curl --digest -u {apiId}:{apiKey} -X POST "https://api.osmicards.com/v2/images"  
--data '{"imgType": "strip", "imgDescription": "Моя картинка",  
"imgData": "iVBORw0KGgoAAAANSUhE... kSuQmCC"}'  
--header "Content-type: application/json"
```

### Возвращаемые данные

Формат возвращаемых данных совпадает с форматом данных, получаемых при использовании команды **"Запросить информацию о графическом файле из библиотеки"**.

### Параметры

#### imgDescription

Необязательная произвольная текстовая информация о графическом файле.

#### imgType

Тип графического файла. Возможные типы:

*strip, logo, icon, background, thumbnail*

Стандартные электронные карты и купоны используют типы *strip*, *logo* и *icon*

#### imgData

Двоичное содержимое графического файла, закодированное в формате base-64.

## Замечания

- Поддерживаются только графические файлы формата ".png"
- Проверяется только тип файла, но не его актуальное двоичное содержимое и размер в пикселях. Хотя система позволяет в последующем "подключить" сохраненный файл к индивидуальной карте, при фактическом отображении карты телефоном графика может отобразиться не так, как ожидалось. Одной из причин такого эффекта может быть нарушение требований формата Пассбук к размеру графики в пикселях для выбранного типа файла.
- При добавлении файла в библиотеку, система может отказать в добавлении файла, если его физический размер превышает установленный лимит.
- При добавлении файла в библиотеку, система может отказать в добавлении файла, если суммарный размер файлов пользователя в библиотеке превышает разрешенную квоту.

## Изменить существующий графический файл в библиотеке

Метод: PUT  
Ресурс: /images/{imgId}  
Параметры: необязательный /push

Эта команда изменяет содержимое и (опционально) описание графического файла в библиотеке. Она не изменяет ни установленный тип графического файла, ни его идентификатор, а только актуальное содержимое файла.

Формат команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X PUT "https://api.osmicards.com/v2/images/ HSJ3TMOFOWOK" -  
-data '{ "imgDescription" : "Моя картинка", "imgData" : "iVBORw0KGgoAAAANSUHE... SuQmCC"}'  
--header "Content-type: application/json"
```

/push

Этот параметр не обязателен. Если он присутствует, то все карты, использующие графический файл с указанным `imgId`, получают push-нотификацию об изменении содержимого карты.

## Возвращаемые данные

Формат возвращаемых данных совпадает с форматом данных, получаемых при использовании команды **"Запросить информацию о графическом файле из библиотеки"**.

Параметры

### **imgData**

Обязательный параметр. Двоичное содержимое графического файла, закодированное в формате base-64.

### **imgDescription**

Необязательный параметр.

## **Удалить существующий графический файл из библиотеки**

Метод: DELETE  
Ресурс: /images/{imgId}  
Параметры: нет

Эта команда удаляет существующий графический файл из библиотеки при условии, что данным файлом не пользуется ни одна карта.

Формат команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X DELETE https://api.osmicards.com/v2/images/HSJ3TMOFOWOK
```

### **Возвращаемые данные**

При успешном удалении файла возвращает HTTP статус 200, или сообщение об ошибке.

## **Безусловно удалить существующий графический файл из библиотеки и обновить карты**

Метод: DELETE  
Ресурс: /images/{imgId}/force  
Параметры: необязательный /push

Эта команда удаляет существующий графический файл из библиотеки даже если этот графический файл используется картами.

Формат команды с использованием cURL:

```
curl --digest -u {apild}:{apiKey} -X DELETE  
https://api.osmicards.com/v2/images/HSJ3TMOFOWOK/force/push
```

```
/push
```

Этот параметр не обязателен. Если он присутствует, то все карты, использующие графический файл с указанным `imgId`, получат push-нотификацию об изменении содержимого карты.

Данная команда находит все карты, использующие графический файл с указанным `imgId`, и затем удаляет связь между найденными картами и этим графическим файлом, фактически переключая карты на отображение графики шаблона карты. После этого графический файл физически удаляется из библиотеки.

Рекомендуется пользоваться этой командой с осторожностью, так как она может привести к одновременным нежелательным изменениям внешнего вида большого количества карт.

### Возвращаемые данные

При успешном удалении файла возвращает HTTP статус 200, или сообщение об ошибке.

## Автоматическая выдача карт

Система позволяет создавать так называемые кампании по автоматическому созданию и распространению электронных карт с заранее заданными параметрами. Каждая такая кампания обладает конфигурируемым набором свойств и параметров:

- Базовый шаблон для создания карт
- Два типа кампаний: Последовательная нумерация и Список
- Универсальная ссылка на загрузку карты
- Схема нумерации создаваемых карт
- Предварительно загружаемый список номеров карт для выдачи
- Срок действия
- Ограничение на количество выданных карт
- Настройка вэб страницы выдачи карт
- Настройка вэб страниц ошибок и специальных ситуаций
- Специальные метасимволы в полях данных карт

При осуществлении перехода по ссылке создается новая карта с последовательным серийным номером согласно схеме нумерации (тип auto) или из заранее заданного списка номеров (тип pool). При достижении заданного количества карт, использования всех карт из списка или окончания срока действия кампании выдача карт прекращается. Все созданные карты доступны для обновлений и редактирований используя стандартные методы API работы с картами и из Личного Кабинета (ЛК).

В поля данных шаблона, участвующего в маркетинговой кампании, можно прописывать специальные метасимволы, которые при автоматическом создании карты с такого шаблона заменятся на актуальные значения. Список поддерживаемых метасимволов дальше в этом разделе.

Автоматическая выдача карт может являться базой для создания систем он-лайн регистрации пользователей.

Ниже приведен набор методов API для работы с кампаниями – создание, редактирование, статистика.

*Приведенные ниже методы являются более современной альтернативой ранее использованным методам `/campaigns`, которые тоже поддерживаются для обратной совместимости. В новых разработках рекомендуется использовать методы `/campaigns2`. Старые методы `/campaigns` в дальнейшем могут быть убраны из числа поддерживаемых. Документация по старым методам `/campaigns` доступна в предыдущих версиях документации по REST API (1.10).*

## Метасимволы

### **{pin}**

при создании карты заменяется на случайный 4-х значный номер. Может использоваться как средство верификации карты.

### **{serial}**

при автоматическом создании карты заменяется на серийный номер карты.

### **{osmibase}**

заменяется на базовый адрес сервера, обслуживающего карту. Используется в процессе регистрации карт.

### **{register}**

заменяется на виртуальный адрес директории с кодом регистрационной анкеты. Используется в процессе регистрации.

Типичное использование переменных в процессе регистрации карт это прописать в шаблоне карты на оборотной стороне ссылку на страничку регистрации:

Для регистрации Вашей карты пройдите [по этой ссылке]({osmibase}{register}) и заполните анкету.

Для регистрации Вашей карты пройдите [по этой ссылке](http://site.com/mylink{register}) и заполните анкету.

## Запросить список кампаний

Метод: GET  
Ресурс: /campaigns2  
Параметры: необязательный `activeOnly=true`

Эта команда запрашивает список всех кампаний. Формат команды с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X GET "https://api.osmicards.com/v2/campaigns2"
```

```
{
  "campaigns": [
    {
      "name": "8 Марта",
      "type": "-auto-"
    },
    {
      "name": "Скидочная карта",
      "type": "-pool-"
    },
    {
      "name": "Welcom",
      "type": "-auto-"
    }
  ]
}
```

Если команда выполнена без ошибок, то она вернет список активных кампаний.

Если был использован параметр `activeOnly=true` то в список попадут только активные кампании.

## Запросить параметры кампании

Метод: GET  
Ресурс: /campaigns2/{имя кампании}  
Параметры: нет

Эта команда возвращает параметры определенной кампании. Формат команды с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X GET "https://api.osmicards.com/v2/campaigns2/Welcome9"
```

Если команда выполнена без ошибок, то она вернет параметры кампании.

Структура возвращаемых данных одинакова для типов кампаний `auto` и `pool`. Наполнение конкретных данных в некоторых случаях будет зависеть от типа кампании `auto` или `pool`.

В данном примере показан возвращаемый результат для кампании типа `pool`.

```

"cards": {
  "numLength": "-empty-",
  "lastSerial": "777771",
  "prefix": "-empty-",
  "issued": 2,
  "limit": 5
},
"active": true,
"linkActive": true,
"pages": {
  "inactive": {
    "type": "-notused-",
    "content": "-empty-"
  },
  "expired": {
    "type": "-default-",
    "content": "-empty-"
  },
  "introduction": {
    "type": "-notused-",
    "content": "-empty-"
  },
  "wrongPhone": {
    "type": "-default-",
    "content": "-empty-"
  },
  "download": {
    "type": "-notused-",
    "content": "-empty-"
  },
  "stdHeaders": {
    "headerRU": "-empty-",
    "footerRU": "-empty-",
    "headerEN": "-empty-",
    "footerEN": "-empty-"
  }
},
"smsText": "Ваша карта готова. Для установки нажмите на ссылку {}",
"link": "https://vm-pa.osmicards.com/XQN45RS7D2Q5E846ZELA",
"poolcards": [
  { "serial": "777771", "used": false },
  { "serial": "777772", "used": true },
  { "serial": "777773", "used": true },
  { "serial": "777774", "used": false },
  { "serial": "777775", "used": false }
],
"type": "-pool-",
"dates": {
  "start": "2015-08-27 00:00:00",
  "end": "2016-08-27 00:00:00",
  "DTS": "2016-03-06 07:40:31"
},
"name": "Welcome9",
"template": "Welcom OSMI"
}

```



## Формат возвращаемых данных

Формат возвращаемых в большей части совпадает с передаваемыми данными в запросах создания и редактирования кампаний.

- (ro) Параметр игнорируется при использовании в вызовах создания и обновления кампании
- (rw) параметр используется при создании и редактировании кампании

### active

(rw) Глобальный признак того что кампания активна и может создавать новые карты при условии, что другие требуемые параметры кампании позволяют это делать.

### linkActive

(rw) Ссылка на загрузку карты активна.

### cards

Секция определяет параметры нумерации карт, лимиты выдачи и текущую статистику выдачи карт.

Параметр	-auto-	-pool-
numLength	Количество символов в номере, не считая префикса (r/w)	Не используется
lastSerial	Последний автоматически созданный серийный номер (ro)	Следующий доступный номер из списка (ro)
prefix	Префикс номера карты (r/w)	Не используется
issued	Количество созданных карт (ro)	Количество созданных карт (ro)
limit	Лимит карт (r/w)	Количество карт в списке (ro)

### pages

(rw) Секция определяет параметры и свойства различных страниц, на которые попадает конечный пользователь, при прохождении по ссылке загрузки карты. Каждая страница может быть определенного типа. Тип страницы определяет ее поведение и внешний вид.

Тип страницы	Поведение	Примечание
-default-	Используется стандартная страница	
-notused-	Страница не будет показана	
-custom-	Страница со своим собственным HTML кодом и версткой	Содержание = HTML код
-redirect-	Будет осуществлен переход на указанный ресурс	Содержание = URL ссылка

### stdHeaders

Секция задает необязательный HTML код для показа в качестве первого и последнего параграфов соответственно на загрузочной странице.

Обычно используется с типом страницы `-default-` в качестве заголовка и текста oferty.

#### link

(ro) Ссылка на загрузку карт, выдаваемых в рамках данной кампании.

#### type

(rw) Тип кампании. Принимает значения `-auto-` или `-pool-`.

Тип кампании можно задавать только при создании новой кампании. Изменить тип кампании после ее создания нельзя.

#### dates

(rw) Секция дат начала и окончания сроков действия кампании и время создания кампании. Если клиент проходит по ссылке загрузки карты но кампания либо еще не началась или уже закончилась, то показывается соответственно страничка `inactive` или `expired`.

#### name

(ro) Имя кампании.

#### template

(rw) Имя шаблона, с которого выдаются карты. Так как карты создаются автоматически из шаблона, то следует заполнить все поля данных по умолчанию шаблона. Если поле данных шаблона оставить не заполненным, то на телефоне это поле на выданной карте не отобразится.

#### poolcards

(rw) Используется только для кампаний типа `-pool-`. Представляет собой список серийных номеров, которые будут использованы для создания карт, и их текущий статус. Для кампаний типа `-auto-` всегда возвращается пустой массив.

Параметр	-auto-	-pool-
serial	Не используется	Номер карты в списке доступных номеров карт
used	Не используется	Номер уже был использован или нет для создания карты

## Создать новую кампанию

Метод: POST  
Ресурс: /campaigns2/{имя кампании}  
Параметры: нет

Эта команда позволяет создать новую кампанию. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X POST "https://api.osmicards.com/v2/campaigns2/8%20mapra" --data
{"dates":{"start":"2015-06-14 12:00:00", "end":"2016-06-14 12:00:00"}, "type":"-auto-", "cards":
{"numLength": 5, "prefix": "ZZ1", "limit": 500 }} --header 'Content-type: application/json'
```

Если команда выполнена без ошибок, то она вернет параметры вновь созданной кампании как при вызове «Запросить параметры кампании».

Формат входных данных полностью повторяет формат возвращаемых данных вызова «Запросить параметры кампании», за исключением того, что информационные поля (ro) передавать не надо.

Обязательными для заполнения являются поля:

dates, type, cards (для -auto-), poolcards (для -pool-)

Если какие-либо необязательные параметры не были переданы, они будут заполнены значениями по умолчанию.

При создании кампании типа "-auto-" необходимо заполнить параметр "cards". Параметр "poolcards" будет проигнорирован.

При создании кампании типа "-pool-" необходимо заполнить параметр "poolcards". В нем передается просто массив серийных номеров карт без статуса, например так ["1234", "1235", "1236", ...].

## Редактировать параметры кампании

Метод: PUT  
Ресурс: /campaigns2/{имя кампании}  
Параметры: нет

Эта команда позволяет редактировать параметры кампании. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X PUT "https://api.osmicards.com/v2/campaigns2/8%20mapra" --data
{"dates":{"end":"2015-06-14 12:00:00"}, "pages": {"inactive": {"content": "http://www.romashka.ru/info"
:"type": "-redirect-" }} } --header 'Content-type: application/json'
```

Если команда выполнена без ошибок, то она вернет параметры кампании как при вызове «Запросить параметры кампании».

Формат входных данных полностью повторяет формат возвращаемых данных вызова «Запросить параметры кампании», за исключением того, что информационные поля передавать не надо.

Те поля, которые не переданы в запросе, останутся в кампании без изменений.

В кампаниях типа `-pool-` можно изменять список номеров карт. Для манипулирования списком карт в кампании можно использовать следующие дополнительные под-команды, передаваемые в теле запроса:

команда	назначение	примечание
<code>-add-</code>	Добавить новые карты к существующему списку карт	Переданные в запросе номера добавятся в список доступных карт. Дубликаты номеров не создаются и будут проигнорированы.
<code>-delete-</code>	Удалить карты из списка	Переданный список карт будет удален из списка доступных номеров для кампании.
<code>-deleteall-</code>	Удалить все карты из списка	Весь список номеров для кампании будет удален, невзирая на содержимое параметра <code>poolcards</code> . Но <code>poolcards</code> должен быть передан в запросе, даже если пустой массив []

Под-команды передаются в теле запроса, например как `"poolcmd": "-add"`. При использовании под-команды необходимо в запросе так-же передать параметр `"poolcards": [...]`

#### Пример:

```
curl --digest -u {apild}:{apiKey} -X PUT "https://api.osmicards.com/v2/campaigns2/8%20mapra" --data '{"dates": {"end": "2015-06-14 12:00:00"}, "poolcmd": "-add-", "poolcards": ["12", "13", "33"]}' --header 'Content-type: application/json'
```

#### Примечания:

1. При удалении карт из списка кампании (`-pool-`) обновляется только список доступных номеров для создания карт. Сами карты, если они уже были выданы из кампании, ЛК или через другие API – не удаляются.
2. При добавлении номеров в список доступных номеров карт для кампании (`-pool-`), номера глобально не резервируются. Это означает, что через ЛК или API можно создать карту с номером из списка вне рамок процесса создания карты через кампанию. Если карта была создана таким образом вне рамок кампании, то ее серийный номер перестанет быть доступен в списке номеров карт для кампании.

## Удалить кампанию

Метод: DELETE  
Ресурс: /campaigns2/{имя кампании}  
Параметры: нет

Эта команда позволяет удалить кампанию. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X DELETE 'https://api.osmicards.com/v2/campaigns2/8%20марта'
```

Если команда выполнилась без ошибок, то она вернет HTTP статус 204 и пустое тело ответа.

При удалении кампании удаляется только возможность автоматически выдавать карты при прохождении по линку кампании а не сами ранее выданные карты.

## Регистрация автоматически выданных карт

Очень часто автоматическая выдача карт с использованием кампаний подразумевает дополнительный шаг, заключающийся в приглашении пользователя, установившего автоматическую анонимную карту или купон, пройти он-лайн регистрацию. При прохождении такой регистрации пользователем, полученная информация и контакты пользователя становятся доступны бизнесу и появляется возможность конвертировать обезличенную карту в персональную и одновременно создать в бизнес-системе нового пользователя с проверенными контактными данными.

Для поддержания такого бизнес процесса предоставляются API для проверки и обновления статуса регистрации карты. Так-же предоставляются методы для сохранения полученной регистрационной информации и ее выгрузки в учетные системы бизнеса. Для работы с несколькими процессами регистрации, требующими разный набор вопросов к пользователю, предлагается механизм работы с регистрационными группами.

Типично на карте публикуется ссылка с приглашением пройти регистрацию он-лайн. Ссылка включает в себя номер карты и дополнительный случайный параметр защиты `{link}`, генерирующийся автоматически при создании карты и предотвращающий возможность автоматического подбора номеров карт.

## Получить статус регистрации карты

Метод: GET  
Ресурс: `/registration/status/{link}/{serial}`  
Параметры: нет

Эта команда позволяет запросить текущий статус регистрации карты. Статус возвращается только для автоматически созданных через кампанию карт, участвующих в процессе регистрации. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X GET 'https://api.osmicards.com/v2/registration/status/ABCDE/12345'
```

Если команда выполнена без ошибок, то она вернет статус регистрации карты:

```
{
  "status": 1
}
```

Возможные статусы регистрации карт показаны в таблице:

Статус	Значение
0	Регистрация данной карты запрещена
1	Карта еще не прошла регистрацию
2	Карта уже была зарегистрирована

Если такая карта не существует или карта не участвует в схеме регистрации, то вернется HTTP статус 404.

## Обновить статус регистрации карты

Метод: PUT  
Ресурс: /registration/status/{serial}  
Параметры: нет

Эта команда позволяет обновить статус регистрации карты. Статус обновляется только для карт, участвующих в схеме регистрации. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X PUT 'https://api.osmicards.com/v2/registration/status/12345' -data '{
"status": 2 }' -header 'Content-Type: application/json'
```

Если команда выполнилась без ошибок, то она вернет HTTP статус 204 и пустое тело ответа. Если такая карта не существует или карта не участвует в схеме регистрации, то вернется HTTP статус 404.

## Параметры

### status

Обязательный параметр для установки статуса карты. Возможные значения показаны в таблице

Параметр	Значение
-remove-	Удалить карту из схемы регистрации. Сама карта не удаляется.
-disabled-	Запретить регистрацию данной карты
-active-	Установить статус в «еще не прошла регистрацию»
-registered-	Установить статус в «уже была зарегистрирована»

При установке значения в -remove-, дополнительно удаляются ранее сохраненные данные регистрации для этой карты (см ниже).

## Обновить статус регистрации карты и сохранить данные он-лайн регистрации

Метод: PUT  
Ресурс: /registration/status/{serial}  
Параметры: нет

Эта команда является расширением предыдущей команды. Она позволяет сохранить информацию, полученную от пользователя при прохождении он-лайн регистрации, на сервере ОСМИ для последующего импорта в бизнес системы клиента. Статус и данные обновляются только для карт, участвующих в схеме регистрации. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apiId}:{apiKey} -X PUT 'https://api.osmicards.com/v2/registration/status/12345'  
-data '{ "status": 2, "group": "демо", "saveData": true, "values": [ { "field": "Имя", "value": "Иван" },  
{ "field": "phone", "value": "79164442211" } ] }' -header 'Content-Type: application/json'
```

Если команда выполнялась без ошибок, то она вернет HTTP статус 204 и пустое тело ответа. Если такая карта не существует или карта не участвует в схеме регистрации, то вернется HTTP статус 404. Если были переданы значения для сохранения, то значения проверяются. В случае передачи неправильных значений возвращается соответствующая ошибка.

## Параметры

### status

Обязательный параметр статуса карты. Принимает значения как в не расширенной версии команды.

### saveData

Необязательный параметр. Если установлен в `true` то при обновлении статуса регистрационной карты будут сохранены данные, переданные в параметре `values`. Ожидаются поля данных, принадлежащие группе с названием, указанным в параметре `group`.

### group

Необязательный параметр. Задет имя группы, описывающей название полей данных из регистрационной формы. Является обязательным параметром при использовании `saveData:true`

### values

Необязательный параметр. Массив значений, задающий названия регистрационных полей (`field`) и их значений (`value`). Является обязательным параметром при использовании `saveData:true`

## Примечания

- Новые данные сохраняются только при установке статуса в `-registered-`. При этом все ранее сохраненные регистрационные данные для выбранной карты удаляются перед сохранением новых данных.
- При установке статуса в любое другое значение и независимо от того, были ли переданы новые регистрационные данные или нет, все ранее сохраненные регистрационные данные для выбранной карты удаляются.
- Необязательно заполнять все поля данных, доступных в выбранной регистрационной группе. Если какое-либо поле не передано то оно автоматически сохраниться как пустая строка.



## Получить список регистрационных групп

Метод: GET  
Ресурс: /registration/groups  
Параметры: нет

Эта команда позволяет получить список доступных регистрационных групп. Каждая группа задает свой набор полей данных для запоминания регистрационных параметров, полученных в процессе он-лайн регистрации от пользователя. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X GET 'https://api.osmicards.com/v2/registration/groups'
```

Если команда выполнялась без ошибок, то она вернет список доступных групп:

```
{  
  "groups": [ "демо", "mygroup" ]  
}
```

## Получить поля данных регистрационной группы

Метод: GET  
Ресурс: /registration/groups/{group}  
Параметры: нет

Эта команда позволяет получить для группы список доступных полей для сохранения регистрационных данных. Каждая группа может иметь свой список произвольных полей. При сохранении регистрационных данных система проверяет переданные данные (имена полей `fields`) на их наличие в выбранной группе. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X GET 'https://api.osmicards.com/v2/registration/groups/демо'
```

Если команда выполнялась без ошибок, то она вернет список полей для запрошенной группы:

```
{  
  "fields": [  
    "Name",  
    "Surname",  
    "Middlename",  
    "Phone",  
    "PhoneCheck",  
    "OfertaCheck",  
    "Sex",  
    "Email"  
  ]  
}
```

Имена групп и доступные поля для сохранения данных для он-лайн регистрации создаются по запросу в службу технической поддержки. После регистрации группы готовы к использованию для хранения данных.

## Получить регистрационные данные для карт

Метод: GET  
Ресурс: /registration/data/{group} [/array]  
Параметры: необязательный [/array]

Эта команда позволяет получить ранее сохраненные регистрационные данные для карт, которые использовали параметры полей из заданной группы. Данные возвращаются только для карт со статусом -registered-. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X GET 'https://api.osmicards.com/v2/registration/data/демо'
```

Если команда выполнялась без ошибок, то она вернет список регистрационных данных по каждой карте, входящих в запрошенную регистрационную группу:

```
{
  "registrations": [
    {
      "serialNo": "W000004",
      "Middlename": "-empty-",
      "Sex": "-empty-",
      "Phone": "-empty-",
      "OfertaCheck": "-empty-",
      "Name": "Петя",
      "Surname": "Иванов",
      "PhoneCheck": "-empty-",
      "Email": "a@b.com"
    },
    {
      "serialNo": "W000008",
      "Middlename": "-empty-",
      "Sex": "-empty-",
      "Phone": "-empty-",
      "OfertaCheck": "-empty-",
      "Name": "Петя",
      "Surname": "Иванов",
      "PhoneCheck": "-empty-",
      "Email": "a@b.com"
    }
  ]
}
```

```
{
  "registrations": [
    {
      "field": "Name", "value": "Петя" },
      { "field": "Surname", "value": "Иванов" },
      { "field": "Middlename", "value": "-empty-" },
      { "field": "Phone", "value": "-empty-" },
      { "field": "PhoneCheck", "value": "-empty-" },
      { "field": "OfertaCheck", "value": "-empty-" },
      { "field": "Sex", "value": "-empty-" },
      { "field": "Email", "value": "a@b.com" },
      { "field": "serialNo", "value": "W000004" }
    },
    {
      "field": "Name", "value": "Петя" },
      { "field": "Surname", "value": "Иванов" },
      { "field": "Middlename", "value": "-empty-" },
      { "field": "Phone", "value": "-empty-" },
      { "field": "PhoneCheck", "value": "-empty-" },
      { "field": "OfertaCheck", "value": "-empty-" },
      { "field": "Sex", "value": "-empty-" },
      { "field": "Email", "value": "a@b.com" },
      { "field": "serialNo", "value": "W000008" }
    }
  ]
}
```

Первый пример показывает возвращаемые данные в стандартном формате динамического объекта. Второй пример показывает формат данных при использовании параметра /array.

Данные возвращаются ровно в том-же виде как они были сохранены. Все данные имеют строковое представление. Никакого разбора и преобразования данных не производится.

## Удалить регистрационные данные карт

Метод: POST  
Ресурс: /registration/deletedata  
Параметры: нет

Эта команда удаляет ранее сохраненные регистрационные данные карт. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X POST 'https://api.osmicards.com/v2/registration/deletedata' --data '{ "registrations": [ "A0001", "A1251", "0034567" ] }' --header 'Content-Type: application/json'
```

Если команда выполнялась без ошибок, то она вернет HTTP статус 204 и пустое тело ответа. Если в списке были переданы номера карт, не участвующих в схеме регистрации, они игнорируются при обработке запроса. В случае передачи неправильных значений возвращается соответствующая ошибка.

### Параметры

#### registrations

Обязательный параметр. Список серийных номеров карт.

## Общие замечания по регистрации карт

Успешное использование регистрации автоматически выдаваемых карт требует подготовки, такой как создание требуемых шаблонов карт со ссылками на регистрационную страничку, размещение на вэб сервере регистрационных страничек и исполняемого регистрационного кода, шаблоны зарегистрированных карт и тп.

Если предполагается временное хранение полученных регистрационных данных на серверах ОСМИ, то потребуется создать регистрационные группы с нужными полями данных. Поля данных должны соответствовать дизайну регистрационной странички с формами сбора данных и регистрационному коду.

ОСМИ предоставляет полностью функциональный пример он-лайн регистрации для размещения на серверах клиента и требующий минимальной подстройки. Пример включает в себя HTML странички с формами сбора данных в процессе регистрации, javascript код и серверный PHP код с подробным описанием. Поддержан опциональный режим проверки номера мобильного телефона клиента через использование проверочного sms кода, отправляемого клиенту в процессе регистрации. Примеры кода полностью оптимизированы для исполнения на мобильных устройствах.

Примеры доступны по запросу в службу технической поддержки ОСМИ.

## Резервирование карт

Подход с автоматической выдачей карт и (опциональной) последующей регистрацией пользователей эффективно работает в ситуациях, когда источник прихода посетителей заранее неизвестен и нумерация автоматически создаваемых карт не имеет существенного значения для бизнес процесса. По факту регистрации пользователя и проверке введенной им при регистрации персональной информации, его автоматически созданная электронная карта может быть привязана к уже существующей учетной записи пользователя в бизнес системе, или новый пользователь создан на базе полученных данных.

В ситуациях, когда требуется выдавать электронные карты уже существующим клиентам, у которых на руках находятся пластиковые карты с определенной нумерацией и создание электронных карт с новой нумерацией и последующей привязкой к учетной записи пользователя нежелательно, доступен альтернативный метод реализации требуемой задачи, т.н. Резервирование Карт.

В этом методе электронные карты заранее не создаются, как и не создается механизм автоматической генерации карт. Вместо этого берется существующая база номеров пластиковых карт, и на ее основе резервируется пул таких-же номеров электронных карт. Сами карты не создаются, а создается уникальная ссылка-приглашение на прохождение регистрации. Уникальные ссылки рассылаются по существующей базе клиентов. Те клиенты, кто решил пройти регистрацию, дополнительно идентифицируются в процессе регистрации по заранее сконфигурированному параметру (например, номер телефона или номер существующей пластиковой карты). При успешном завершении регистрации клиентом, создается и выдается персонализированная электронная карта с номером существующей пластиковой карты. Ссылка на эту регистрацию объявляется недействительной и более не может быть использована для создания персонализированной карты.

Такой подход позволяет:

- Обеспечить безопасный метод конвертации пластика в индивидуальную электронную карту.
- Задавать требуемые параметры проверки и идентификации пользователя при регистрации.
- Упростить интеграцию с бизнес системой, так как нумерация карт сохраняется и номер электронной карты соответствует номеру существующей пластиковой карты.
- Избежать массового создания большого количества электронных карт для всей существующей клиентской базы. Фактические электронные карты создаются только для тех клиентов, кто решил сконвертировать традиционную карту в электронную.
- Актуализировать базу клиентов.

## Получить информацию о ссылке резервирования карты

Метод: GET  
Ресурс: /registration/match/{link}  
Параметры: нет

Эта команда возвращает информацию, ассоциированную со ссылкой на резервирование. Пример команды с использованием cURL показан ниже.

```
curl --digest -u {apild}:{apiKey} -X GET 'https://api.osmicards.com/v2/registration/match/XGL34F'
```

### Формат возвращаемых данных

Если команда выполнена без ошибок, то она вернет данные, относящиеся к резервационной ссылке:

```
{
  "status": 0,
  "serial": "1356544",
  "param1": "11061977",
  "param2": "-empty-",
  "param3": "-empty-",
  "registered": false,
  "maxTry": 5,
  "link": "https://vm-cabinet.osmicards.com/ecard/XGL34F",
  "attempts": 3,
  "dts": "-empty-",
  "template": "Именная GOLD"
}
```

#### status

Статус карты. Имеет смысл только если карта уже создана. Если карта еще не создана, то будет равен 0. Если создана, то статус соответствует статусу карты, полученный из команды «Запросить информацию о карте».

#### serial

Серийный номер карты, зарезервированный за ссылкой. Следует иметь ввиду, что резервирование серийного номера имеет только ознакомительное а не обязательное значение. Другими словами, можно создать карту с указанным номером через API или личный кабинет, и наличие ссылки резервирования не запретит создание такой карты.

#### param1,2,3

вспомогательные общие переменные для хранения параметров, относящихся к ссылке резервирования. Принимает значение произвольной строки. Типичное использование этих переменных – участие в идентификации пользователя при прохождении по ссылке. Например, в переменную можно записать год рождения или телефон человека, и регистрационный код сравнит это(эти) значения со значениями, введенными человеком в форму регистрации.

Сам API относится к этим полям просто как к string переменным для стороннего кода и никаких логических действий с этим значениями не предпринимает.

### registered

Статус регистрации карты, относящейся к ссылке. Этот параметр формируется системой динамически и отражает реальное состояние карты – то есть была ли создана карта с серийным номером, зарезервированным за ссылкой. Если карта была создана, то статус будет равен `true`.

### maxTry

вспомогательная общая переменная, которая может использоваться внешним кодом проверки. Типично используется как максимальное число попыток проверки информации, введенной пользователем при проходе по ссылке. Сам API относится к этому полю просто как к `int` переменной для стороннего кода и никаких логических действий с этим значением не предпринимает.

### attemptst

вспомогательная общая переменная, которая может использоваться внешним кодом проверки. Типично используется как текущее число попыток проверки информации, введенной пользователем при проходе по ссылке. Сам API относится к этому полю просто как к `int` переменной для стороннего кода и никаких логических действий с этим значением не предпринимает.

### dts

поле в формате W3C, в котором может находиться дата и время, когда была создана карта с серийным номером, относящемуся к ссылке. Заполняется системой автоматически в момент создания карты любым доступным способом. Если карта еще не создана, то примет значение `"-empty-"`. Может использоваться в фильтрах проверки статуса ссылок и ассоциированными с ними карт.

### link

ссылка на URL процесса регистрации, по окончании которого будет сформирована карта с заданным серийным номером. Ссылка формируется системой автоматически и не может быть изменена. Обычно по ссылке должен располагаться код регистрации, предусматривающий проверку параметров, принятие клиентом оферты, создание новой карты и тд. Простой пример регистрационного PHP кода доступен по запросу.

### template

Имя шаблона, на котором нужно создавать карту после завершения процесса регистрации. Так как процесс регистрации обрабатывается внешним кодом (см выше), то это поле является индикатором для процесса на каком шаблоне создавать карту.

## Получить информацию о группе ссылок резервирования карт

Метод: GET  
Ресурс: /registration/match  
Параметры: необязательный ?registeredOnly=true  
              необязательный ?start=[W3C datetime]

Эта команда возвращает информацию об имеющихся ссылках резервирования карт. Необязательные параметры используются как фильтры параметров ссылок.

```
curl --digest -u {apild}:{apiKey} -X GET  
'https://api.osmicards.com/v2/registration/match?registeredOnly=true'
```

### Формат возвращаемых данных

Если команда выполнялась без ошибок, то она вернет список параметров ссылок. Каждый объект в возвращенном массиве полностью соответствует формату выходных данных команды «получить информацию о ссылке резервирования карты»:

```
{  
  "cards": [  
    {  
      "status": 0,  
      "serial": "000000",  
      "param1": "-empty-",  
      "param2": "-empty-",  
      "param3": "-empty-",  
      "registered": false,  
      "maxTry": 5,  
      "link": "https://vm-cabinet.osmicards.com/ecard/aaa",  
      "attempts": 0,  
      "dts": "-empty-",  
      "template": "CreaConcept"  
    },  
    {  
      "status": 1,  
      "serial": "1qa2ws",  
      "param1": "-empty-",  
      "param2": "-empty-",  
      "param3": "-empty-",  
      "registered": true,  
      "maxTry": 5,  
      "link": "https://vm-cabinet.osmicards.com/ecard/ccc",  
      "attempts": 2,  
      "dts": "2016-07-03T13:47:22Z",  
      "template": "Andrei test"  
    },  
    ...  
  ]  
}
```

### Параметры

#### registeredOnly

Необязательный параметр. Если используется и установлен в `true`, то в выходной список попадут только те объекты, для которых карта уже создана. Может быть скомбинирован с параметром `start`.

## start

Необязательный параметр. Если используется и установлен в дату и время в формате W3C (например 2016-07-07T18:00:00Z), то выходной список попадут только те объекты, у которых переменная `dto` установлена в дату и время равную или позже заданной. Так как `dto` устанавливается автоматически при создании карты, то использование этого параметра автоматически подразумевает установку `registeredOnly` в `true`. Дата и время должны быть urlencoded.

## Создать ссылки на резервирование карт

Метод: POST  
Ресурс: /registration/match  
Параметры: нет

Эта команда создает ссылки резервирования для создания карт. Ссылки можно создавать с необязательными параметрами проверки, которые позже можно использовать в коде регистрации пользователя и выдачи карты.

```
curl --digest -u {apiId}:{apiKey} -X POST 'https://api.osmicards.com/v2/registration/match'  
-data '{ "cards": [ { "serial": "aaa1", "template": "VIP Card", "param1": "12345", "maxTry": 5 }, ... ] }'  
-header 'Content-Type: application/json'
```

Если команда выполнена без ошибок, то она вернет список результатов операции и (при отсутствии ошибок) ссылку для каждого серийного номера карты.

```
{  
  "cards": [  
    {  
      "serial": "zzqqa17",  
      "RCODE": 200,  
      "link": "https://vm-cabinet.osmicards.com/ecard/1IDVIT8T"  
    },  
    {  
      "serial": "zzqqa2",  
      "RCODE": 619,  
      "link": "https://vm-cabinet.osmicards.com/ecard/0NAFH7YL"  
    },  
    {  
      "serial": "zzqqa22",  
      "RCODE": 311,  
      "link": "-empty-"  
    }  
  ]  
}
```



## Параметры

В качестве входных параметров принимает набор значений, описанных параметрами **param1**, **param2**, **param3**, **template**, **maxTry**, **serial**.

Эти параметры описаны в команде «Получить информацию о ссылке резервирования карты».

## Формат возвращаемых данных

### serial

Серийный номер карты. Сама карта не создается, а просто запоминается номер карты против созданного линка, как индикатор с каким номером нужно позже создать карту в коде регистрации пользователя.

### RCODE

Код операции. Его нужно проверять всегда для каждого переданного в запрос серийного номера. Только при условии **RCODE=200** можно считать что операция выполнена успешно и новая ссылка создана.

### link

Полная ссылка резервирования для серийного номера карты. Принимает значение только при **RCODE=200** и **RCODE=619**. При других кодах результата операции принимает значение `"-empty-"`.

## Обновить ссылки на резервирование карт

Метод: PUT  
Ресурс: /registration/match  
Параметры: нет

Эта команда обновляет существующие ссылки резервирования для создания карт. Позволяет обновить и необязательными параметрами проверки, которые позже можно использовать в коде регистрации пользователя и выдачи карты.

```
curl --digest -u {apild}:{apiKey} -X PUT 'https://api.osmicards.com/v2/registration/match'  
-data '{ "cards": [ { "serial": "aaa1", "template": "VIP Card", "param1": "12345", "attempt": "-add-",  
"maxTry": 5 }, ... ] }' -header 'Content-Type: application/json'
```

Если команда выполнена без ошибок, то она вернет список результатов операции и (при отсутствии ошибок) ссылку для каждого серийного номера карты в том-же формате, что и при вызове команды «Создать ссылки на резервирование карт».

```
{
  "cards": [
    {
      "serial": "zzqqa17",
      "RCODE": 200,
      "link": "https://vm-cabinet.osmicards.com/ecard/1IDVIT8T"
    },
    {
      "serial": "zzqqa2",
      "RCODE": 619,
      "link": "https://vm-cabinet.osmicards.com/ecard/0NAFH7YL"
    },
    {
      "serial": "zzqqa22",
      "RCODE": 311,
      "link": "-empty-"
    }
  ]
}
```

## Параметры

Параметры полностью соответствуют формату параметров команды «Создать ссылки на резервирование карт» с некоторыми дополнениями:

### attempt

Необязательный параметр. Позволяет обновить количество попыток проверки при регистрации карты. Может принимать следующие значения:

- «-add-» Увеличивает число попыток на 1
- «-delete-» Уменьшает число попыток на 1
- «-empty-» Сбрасывает число попыток в 0

Следует учитывать, что система только запоминает число попыток, но не пытается провести внутреннее сравнение с максимальным числом попыток `maxTry`. Такое сравнение и принятие (или не принятие) решения по результатам сравнения возлагается на внешний код регистрации пользователя.

### link

Необязательный параметр. Может принимать только одно значение «-delete-». Если использован, то ссылка на резервирование удаляется. На саму карту (если таковая была ранее создана) эта команда не влияет.

## Маркетинг

Типичными задачами маркетинга являются отправление пуш сообщений пользователям с произвольным текстом и отправка уведомлений о текущих акциях и предложениях. Такие сообщения и акции могут быть отправлены активным картам как мгновенно, так и с регулируемой задержкой по времени. Дополнительно, созданные таким образом акции могут иметь срок действия, по истечении которого они исчезнут с карты.

Данные функции являются прямой и бесплатной заменой маркетинговым смс-рассылкам и предоставляют дополнительные возможности по графическому наполнению информационного сообщения.

Динамические пуш сообщения не привязаны к какому-либо полю карты.

Динамические акции всегда отображаются на оборотной стороне карты, в самой верхней позиции. Содержимое поля динамической акции не доступно в блоке информации о карте, полученном через API, и не отображается в блоках информации шаблонов карт.

Проверить доставку отправленного пуш сообщения иди динамической акции можно через использование метода `/stats/delivery`

## Отправить пуш сообщение

Метод: POST  
Ресурс: `/marketing/pushmessage`  
Параметры: нет

Эта команда позволяет отправить пуш сообщение на карту. Сообщение отобразится на активных картах. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X POST 'https://api.osmicards.com/v2/marketing/pushmessage' -data '{
  "serials": [ "123", "456" ], "start": "2015-11-25T15:20:00Z", "message": "У вас сегодня запись к
  мастеру на 12 часов!" }' -header 'Content-Type: application/json'
```

```
curl --digest -u {apild}:{apiKey} -X POST 'https://api.osmicards.com/v2/marketing/pushmessage' -data '{
  "templates": [ "VIP карта", "Silver карта" ], "start": "2015-11-25T15:20:00Z", "message": "У нас новые
  поступления в магазин!" }' -header 'Content-Type: application/json'
```

```
curl --digest -u {apild}:{apiKey} -X POST 'https://api.osmicards.com/v2/marketing/pushmessage' -data '{
  "allCards": true, "start": "2015-11-25T15:20:00Z", "message": "У нас вышел новый каталог!" }' -
  header 'Content-Type: application/json'
```

Если команда выполнена без ошибок, то она вернет HTTP статус 204 и пустое тело ответа.

## Параметры

### serials

Необязательный список номеров карт, на которые оправить сообщение.

### templates

Необязательный список имен шаблонов, картам которых оправить сообщение.

### allCards

Необязательный параметр `true` или `false`. Если `true`, то сообщение отправится всем активным картам учетной записи. Если `false` или другое значение, то параметр игнорируется.

### start

Необязательное дата и время отправки сообщения, в формате W3C. Если не указан, то сообщение отправится сразу.

### message

Обязательный текст сообщения. Если задан как **“-empty-“**, то отложенное сообщение будет удалено.

## Примечания

- `Serials`, `templates` и `allCards` — взаимоисключающие параметры. При задании в запросе более одного параметра, будет использован только один параметр, в следующем порядке: `serials`, `templates`, `allCards`
- Если карте уже присвоено отложенное пуш сообщение, то при задании нового сообщения для этой-же карты, предыдущее сообщение будет заменено в системе на новое с новой датой и временем отсылки.

## Отправить динамическую акцию

Метод: POST  
Ресурс: /marketing/pushaction  
Параметры: нет

Эта команда позволяет создать динамическую акцию для показа на карте, со сроком действия и опциональным пуш сообщением. Акция и сообщение отобразятся на активных картах. Параметры списка карт такие-же как и в команде отправки пуш сообщения. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X POST 'https://api.osmicards.com/v2/marketing/pushaction' --data '{
  "serials": [ "123", "456" ], "start": "2015-11-25T15:20:00Z", "end": "2015-12-30T15:20:00Z", "message":
  "Скоро начнется распродажа...", "body": "Текст акции...", "label": "Распродажа" }' --header
'Content-Type: application/json'
```

Если команда выполнена без ошибок, то она вернет HTTP статус 204 и пустое тело ответа.

## Параметры

### **serials**

Необязательный список номеров карт, на которых установить динамическую акцию.

### **templates**

Необязательный список имен шаблонов, картам которых установить динамическую акцию.

### **allCards**

Необязательный параметр `true` или `false`. Если `true`, то динамическая акция установится всем активным картам учетной записи. Если `false` или другое значение, то параметр игнорируется.

### **start**

Необязательное дата и время появления динамической акции, в формате W3C. Если не указан, то акция отправится сразу.

### **end**

Необязательное дата и время окончания динамической акции, в формате W3C. Если не указан, то акция будет показываться на телефоне пока не будет удалена.

### **label**

Необязательный заголовок для поля акции.

### **message**

Необязательный текст пуш сообщения.

### **body**

Обязательный текст акции для отображения в поле карты. Если задан как “-empty-”, то отложенная акция будет удалена.

## Примечания

- `Serials`, `templates` и `allCards` — взаимоисключающие параметры. При задании в запросе более одного параметра, будет использован только один параметр, в следующем порядке: `serials`, `templates`, `allCards`
- Если карте уже присвоена отложенная акция, то при задании новой акции для этой-же карты, предыдущая акция будет заменена в системе на новую, с новой датой и временем отсылки и новым сроком окончания.

## Работа со статистикой

Система позволяет получать статистическую информацию о состоянии и количестве карт и устройств, доставке пуш сообщений на устройства, собирать информацию как в реальном времени, так и исторические данные. Некоторые аспекты данной информации можно было получать и ранее, используя уже документированные методы API. В этом разделе собраны специализированные методы, делающие сбор и анализ данных более эффективным.

Методы этой группы используют ресурс `/stats`

Поддержаны следующие функции:

- Общая статистика по всем картам в реальном времени
- Историческая статистика по динамике карт
- Отчет по доставке пуш сообщений на карты

### Запросить общую статистику

Метод: GET  
Ресурс: `/stats/general`  
Параметры: нет

Эта команда возвращает информацию по картам и устройствам для всей учетной записи. Данные актуальны на момент вызова метода, что позволяет наблюдать динамику состояния карт и устройств в реальном времени. Пример с использованием cURL приведен ниже.

```
curl --digest -u {apild}:{apiKey} -X GET 'https://api.osmicards.com/v2/stats/general'
```

Если команда выполнена без ошибок, то она вернёт HTTP статус 200 и статистические данные по учетной записи в теле ответа:

```
{
  "cards": {
    "markedDeleted": 0,
    "created": 559,
    "active": 1271,
    "inactive": 155,
    "finalDeleted": 1,
    "total": 1986
  },
  "devices": {
    "total": 1608,
    "ios": 1332,
    "android": 276
  }
}
```

## Формат возвращаемых данных

### **markedDeleted**

количество карт, удаленных оператором учетной.

### **created**

количество карт еще ни разу не установленных на телефоны.

### **active**

количество карт, установленных на телефоны.

### **inactive**

количество карт, которые после установки на телефон либо были удалены пользователем с телефона, либо пользователь отключил получение автоматических уведомлений об изменении карты.

### **finalDeleted**

количество карт, удаленных пользователями с телефонов после получения сообщения о том, что карта была удалена оператором учетной записи.

### **total (cards)**

Общее количество карт в учетной записи.

### **total (devices)**

Общее количество устройств, на которых установлены карты.

### **ios**

Общее количество устройств с iOS (iPhones) на которых установлены карты.

### **android**

Общее количество устройств с Android (iPhones) на которых установлены карты.

## Запросить историческую статистику

Метод:	GET
Ресурс:	/stats/history
Параметры:	необязательный start=startdate необязательный end=enddate

Эта команда возвращает историческую статистическую информацию по учетной записи. Данные рассчитываются один раз в день каждый день на конец дня. Если необходимо получить данные в реальном времени на текущий день, воспользуйтесь командой /stats/general

```
curl --digest -u {apild}:{apiKey} -X GET 'https://api.osmicards.com/v2/stats/history?start=2016-05-31&end=2016-06-12'
```

## Параметры

### start

Необязательный параметр начала периода статистики. Передается в формате ГГГГ-ММ-ДД. Если не указан, то подразумевается дата начала использования учетной записи.

### end

Необязательный параметр окончания периода статистики. Передается в формате ГГГГ-ММ-ДД. Если не указан, то подразумевается вчерашний день.

Если команда выполнена без ошибок, то она вернёт HTTP статус 200 и статистические данные по учетной записи в теле ответа:

```
{
  "stats": [
    {
      "total": {
        "android": 247,
        "inactive": 156,
        "active": 1271,
        "cards": 1986,
        "devices": 156,
        "ios": 1175
      },
      "date": "2016-06-08",
      "change": {
        "created": 2,
        "devices": 1,
        "deactivated": 1,
        "updated": 8,
        "activated": 2
      }
    },
    {
      "total": {
        "android": 246,
        "inactive": 155,
        "active": 1270,
        "cards": 1984,
        "devices": 155,
        "ios": 1175
      },
      "date": "2016-06-07",
      "change": {
        "created": 1,
        "devices": 2,
        "deactivated": 0,
        "updated": 3,
        "activated": 1
      }
    },
    ... ]
}
```

На каждый день в ответе присутствуют три секции:

**total** - статистика за этот день

**change** - изменение статистики по сравнению с предыдущим днем

**date** – день, за который отображается статистика



## Запросить статус доставки пуш сообщения

Метод: GET  
Ресурс: /stats/delivery  
Параметры: необязательный start=startdate  
              необязательный serialNo=номер карты

Эта команда возвращает статус доставки пуш сообщения для одной карты или для всех активных карт учетной записи. Факт доставки пуш сообщения проверяется по дате и времени последнего обращения телефона с картой к серверу для загрузки обновленной карты.

```
curl --digest -u {apild}:{apiKey} -X GET 'https://api.osmicards.com/v2/stats/delivery?start=2016-04-06T17%3A15%3A01Z&serialNo=1234'
```

### Параметры

#### start

Необязательный параметр начала периода с которого проверять была ли доставка пуш сообщения на карту(карты). Передается в формате W3C и должен быть urlencoded. Например дата и время 2016-04-06T17:15:01Z должны быть переданы как 2016-04-06T17%3A15%3A01Z. Использование формата W3C позволяет обойти проблему работы с часовыми поясами и учитывать локальные дату и время.

Если этот параметр не указан, то в получаемых данных отображаются все карты с последней датой доставки пуш сообщения.

#### serialNo

Необязательный параметр номер карты. Если указан, то проверяется и возвращается статус доставки только заданной карте. При использовании должен быть urlencoded.

### Формат возвращаемых данных

#### updated

Дата и время последнего обновления карты.

#### delivered

время доставки пуш сообщения на телефон. Если использовался параметр start и пуш сообщение на карту было доставлено раньше, чем заданное в параметре время, или ни разу еще не было доставлено, то карта не попадет в выводимый список.

#### serialNo

серийный номер карты

```

{
  "delivery": [
    {
      "updated": "2016-06-08T05:42:35Z",
      "delivered": "2016-06-08T05:43:04Z",
      "serialNo": "880002"
    },
    {
      "updated": "2016-04-06T17:15:01Z",
      "delivered": "2016-04-06T17:15:04Z",
      "serialNo": "880003"
    }, ... ]
  }
}

```

## Работа с уведомлениями о событиях

Помимо прямого запроса данных, Система поддерживает режим уведомлений третьей стороны о различных событиях, происходящих с картами, устройствами и регистрацией пользователей в реальном времени. В некоторых случаях это позволяет избежать необходимости регулярного опроса Системы на предмет изменения данных и запрашивать фактические измененные данные только тогда, когда наступило соответствующее событие.

Классическим примером такого подхода, при котором уведомление по событию значительно упрощает процесс сбора данных регистрации, является авторегистрация конечных пользователей. Система настраивается посылать уведомление на вэб сервис третьей стороны о факте успешной регистрации нового пользователя после заполнения онлайн анкеты. При получении такого уведомления, бизнес система через API запрашивает регистрационные данные нового пользователя и обновляет свои ERP или CRM модули с полученными данными и затем переходит в режим ожидания нового события регистрации.

Другим примером такого события может быть установка ранее выданной карты на телефон пользователя. Бизнес процесс может задать правила для максимального количества устройств, на которых может стоять данная карта и, при превышении лимита устройств, бизнес система может отправить на карту предупреждение о превышении лимита или заблокировать карту.

Уведомления о событиях отправляются в реальном времени на заданные для каждого типа событий настраиваемые URL в режимах GET или POST. В любой момент к Системе можно подключить новое уведомление, изменить параметры ранее созданного уведомления или вообще отключить одно или более уведомлений.

В передаваемом запросе в качестве параметров всегда передается тип наступившего события (список как в таблице) и номер карты, для которого произошло данное событие.

Поддерживаемые типы уведомлений приведены в таблице

Уведомление	Событие в системе
cardupdate	Содержимое карты изменилось
cardautocreate	Автоматически создана новая карта из маркетинговой кампании
userregister	Пользователь успешно заполнил и отправил он-лайн анкету в процессе саморегистрации карты
cardactive	Пользователь установил карту на телефон или разрешил ранее запрещенное автообновления для установленной карты
cardinactive	Пользователь удалил карту с телефона или выключил ранее разрешенное автообновления для установленной карты
carddownload	Телефон загрузил с сервера обновленную карту

## Получить список настроенных уведомлений

Метод: GET  
Ресурс: /events  
Параметры: нет

Эта команда возвращает список уведомлений, настроенных на определенные события. Для каждого события система будет вызывать заданный URL с параметрами события. Поддерживается опциональная авторизация. Пример команды с использованием cURL показан ниже.

```
curl --digest -u {apiId}:{apiKey} -X GET 'https://api.osmicards.com/v2/events'
```

## Формат возвращаемых данных

```
{
  "events": [
    {
      "info": "Card updated",
      "password": "*****",
      "method": "GET",
      "url": "http://my.site.com/get",
      "event": "cardupdate",
      "port": 4344,
      "user": "andrei"
    },
    {
      "info": "Card added to a device",
      "password": "-empty-",
      "method": "POST",
      "url": "http://my.com/get33",
      "event": "cardactive",
      "port": "-empty-",
      "user": "-empty-"
    }
  ]
}
```

### info

Краткое описание сути события.

### method

Каким способом отправлять уведомление о событии на сервер бизнеса. Поддерживаются запросы типа GET и POST.

### url

Ресурс на сервере бизнеса, на который будет отправлено уведомление о событии.

### event

Идентификатор наступившего события. Будет отправлен как параметр в уведомлении, отправляемом на сервер бизнеса.

### port

Номер порта в запросе. Если не задан, то будет использоваться стандартный порт для выбранной схемы, например 80 для HTTP и 443 для HTTPS.

### user & password

Эти два параметра тесно связаны. Если оба значения установлены и отличны от `"-empty-"`, то при запросе будет использоваться базовая авторизация, HTTP Basic Authentication. Если хотя бы один из этих двух параметров не установлен или равен `"-empty-"`, то авторизация при запросе использоваться не будет.

## Настроить список уведомлений

Метод: POST  
Ресурс: /events/{event}

Эта команда создает новое уведомление с заданными параметрами, или изменяет параметры ранее созданного уведомления. Пример команды с использованием cURL показан ниже.

```
curl --digest -u {apild}:{apiKey} -X POST 'https://api.osmicards.com/v2/events/cardupdate' --data '{
"method": "post", "url": "https://mysite.com/myevent" }' --header 'Content-Type: application/json'
```

## Параметры

### method

Обязательный параметр. Может принимать значение GET или POST. При использовании метода GET уведомление будет отправлено на сервер в формате:

```
url?serial=xxxx&event=yyyy
```

При использовании метода `POST` уведомление будет отправлено на сервер с параметрами `serial` и `event` в теле запроса, закодированными по методу `application/x-www-form-urlencoded`

- Параметру `serial` будет присвоен номер карты, для которого произошло событие.
- Параметру `event` будет присвоен тип уведомления из таблицы выше

#### **url**

Обязательный параметр. Должен быть равен полному адресу серверного ресурса, на который требуется передавать информацию о событии. Поддерживаются как `http://` так и `https://` схемы адресации. При использовании `https://` схемы, на принимающей стороне должен стоять действительный SSL сертификат для домена.

#### **Примечание**

Если в этом параметре передать значение `"-empty-"`, то уведомление для данного события удалится из списка активных уведомлений.

#### **event**

Обязательный параметр. Задаёт тип события, для которого будет отправляться уведомление. Типы событий должны соответствовать типам, перечисленным в таблице выше.

#### **port**

Номер порта в запросе. Если не задан, то будет использоваться стандартный порт для выбранной схемы, например 80 для HTTP и 443 для HTTPS.

#### **user & password**

Эти два параметра тесно связаны. Если оба значения установлены и отличны от `"-empty-"`, то при запросе будет использоваться базовая авторизация, HTTP Basic Authentication. Если хотя бы один из этих двух параметров не установлен или равен `"-empty-"`, то авторизация при запросе использоваться не будет.

### **Формат возвращаемых данных**

При успешном исполнении команды вернется список активных уведомлений как при исполнении команды «Получить список настроенных уведомлений».

## Групповые операции

Основная цель групповых операций – упростить обработку карт на стороне клиента и существенно сократить время на выдачу и обновление больших объемов карт.

Стандартный механизм создания карты и выдачи ее клиенту обычно состоит из трех шагов:

1. Создать новую карту на требуемом шаблоне
2. Обновить созданную карту с требуемыми значениями
3. Запросить ссылку на загрузку карты для последующей отправки

Данный механизм вполне адекватен при выдаче единичных карт по требованию из системы управления. Для выдачи относительно небольших объемов карт можно запустить несколько потоков исполнения, что существенно увеличивает общую производительность системы. Но при выдаче действительно больших объемов карт, такой процесс может занять от нескольких минут до нескольких часов, в зависимости от конкретной реализации бизнес-системы.

Другим типичным примером групповой обработки является обновление большого количества карт с отправкой пуш-уведомления по результатам дневного процессинга транзакций в организации и/или при синхронизации нескольких баз торговых точек в конце дня.

Для таких операций больше подходит групповой принцип обработки, когда формируется требуемый массив карт на создание или обновление, и одной операцией он передается на сервер где и обрабатывается.

Описанные ниже методы позволяют обрабатывать до 1000 карт за одну транзакцию при времени обработки 2-8 секунд.

## Групповое создание карт

Метод: POST  
Ресурс: /bulk/passes  
Параметры: необязательный ?withValues=true

Эта команда создает группу карт из переданного списка, опционально обновляет выборочные поля и параметры карты с требуемыми значениями и возвращает ссылку на загрузку карт и статусом операции. Список карт и параметры создания передаются в теле запроса. Пример команды с использованием cURL показан ниже.

```
curl --digest -u {apild}:{apiKey} -X POST 'https://api.osmicards.com/v2/bulk/passes?withValues=true'  
-data '{ "cards": [  
  { "serial": "123", "template": "Card", "data": { "values": [ { "value": "22", "label": "Bonus" } ] } },  
  { "serial": "345", "template": "VIP", "data": { "values": [ { "value": "33", "label": "Bonus" } ] } }, ...  
] }' -header 'Content-Type: application/json'
```

## Параметры

В запросе передается массив объектов. Каждый объект описывает одну карту. Можно передавать до 1000 объектов в одном запросе. Время исполнения запроса: от 2 - 8 секунд, в зависимости от наполнения и количества переданных объектов.

### **serial**

Обязательный параметр. Серийный номер карты для создания. Если отсутствует, то соответствующий объект пропускается при обработке. Если карта с таким номером уже существует, то новая карта не создается и не обновляется с переданными значениями (если было дополнительно запрошено обновление карты).

### **template**

Обязательный параметр. Имя шаблона на котором создать карту. Если отсутствует или указан несуществующий шаблон, то для данного объекта вернется ошибка и объект пропускается при обработке.

### **data**

Необязательный или обязательный параметр, в зависимости от `withValues`. Описывает данные полей карты и другие соответствующие значения, которые требуется применить к вновь созданной карте в дополнение и после того, как поля карты заполнятся значениями по умолчанию из шаблона при создании новой карты.

Формат параметра – это объект, полностью соответствующий по структуре и правилам обработки объекту, используемому в команде “Обновить карту”:

- Если в параметре запроса был указан `withValues=true`, то параметр `data` обязателен, и он будет дополнительно проверен на соответствие формату объекта команды “Обновить карту”.
- Если `withValues` не был задан или не был установлен в `true`, то карта создается со значениями по умолчанию из шаблона и параметр `data` будет проигнорирован.

## Формат возвращаемых данных

При успешном исполнении команды вернется список объектов, каждый из которых - это результат исполнения запроса для каждой карты.

Для каждого объекта в списке необходимо всегда проверять статус `RCODE`. Только значение `RCODE=200` означает успешное исполнение команды, то есть создание и возможное обновление полей и других параметров карты. Любое другое значение сигнализирует об ошибке. Расшифровка значений ошибок дается в Приложении 1.

```
{
  "opresults": [
    {
      "serial": "aa2301111",
      "RCODE": 200,
      "link": "https://vm-pd.osmicards.com/43PEHGBQ6SK"
    },
    {
      "serial": "aa23122001",
      "RCODE": 311,
      "link": "-empty-"
    },
    {
      "serial": "aa2323002",
      "RCODE": 200,
      "link": "https://vm-pd.osmicards.com/IYGCQD55LJXY"
    },
    {
      "serial": "aa2334003",
      "RCODE": 200,
      "link": "https://vm-pd.osmicards.com/BVTJEMU0C3G4"
    },
    ... ]
  }
}
```

### serial

Серийный номер карты. Если номер был передан во входных параметрах но отсутствует в списке выходных параметров, то это означает что карта не создалась скорее всего по причине ошибки во входных параметрах.

### RCODE

Результат операции. Равен 200 при успешном исполнении команды. Любое другое значение сигнализирует об ошибке.

### link

Ссылка на загрузку карты. Заполняется реальной ссылкой только при RCODE=200 и 319. Если добавить к ссылке строку суффикса ".png", то ссылка будет указывать на он-лайн картинку с QR кодом для прямого сканирования телефоном.

## Групповое обновление карт

Метод: PUT  
 Ресурс: /bulk/passes  
 Параметры: нет

Эта команда обновляет группу карт из переданного списка, опционально отсылая пуш-уведомление на выбранные карты. Список карт и параметры обновления передаются в теле запроса. Пример команды с использованием cURL показан ниже.

```
curl --digest -u {apild}:{apiKey} -X PUT 'https://api.osmicards.com/v2/bulk/passes'
--data '{ "cards": [
  { "serial": "123", "push": true, "data": { "values": [ { "value": "22", "label": "Bonus" } ] } },
  { "serial": "345", "push": false, "data": { "values": [ { "value": "33", "label": "Bonus" } ] } }, ...
] }' --header 'Content-Type: application/json'
```



## Параметры

В запросе передается массив объектов. Каждый объект описывает одну карту. Можно передавать до 1000 объектов в одном запросе. Время исполнения запроса: от 2 - 8 секунд, в зависимости от наполнения и количества переданных объектов.

### serial

Обязательный параметр. Серийный номер карты для обновления. Если отсутствует, то соответствующий объект пропускается при обработке.

### push

Необязательный параметр. Если присутствует и установлен в `true`, то после обновления будет отправлен пуш если карта активна, то есть установлена хотя бы на одно устройство.

### data

Обязательный параметр. Описывает данные для обновления полей карты и другие соответствующие значения.

Формат параметра – это объект, полностью соответствующий по структуре объекту, используемому в команде “Обновить карту”.

## Формат возвращаемых данных

При успешном исполнении команды вернется список объектов, каждый из которых - это результат исполнения запроса для каждой карты.

Для каждого объекта в списке необходимо всегда проверять статус `RCODE`. Только значение `RCODE=200`, означает успешное исполнение команды, то есть обновление карты. Любое другое значение сигнализирует об ошибке. Расшифровка значений ошибок дается в Приложении 1.

```
{
  "opresults": [
    {
      "serial": "aa2301111",
      "RCODE": 200,
    },
    {
      "serial": "aa23122001",
      "RCODE": 311,
    },
    {
      "serial": "aa2323002",
      "RCODE": 200,
    },
    {
      "serial": "aa2334003",
      "RCODE": 200,
    },
    ... ]
}
```

### serial

Серийный номер карты. Если номер был передан во входных параметрах но отсутствует в списке полученных данных, то это означает что карта не обновилась скорее всего по причине ошибки во входных параметрах.

### RCODE

Результат операции. Равен 200 при успешном исполнении команды. Любое другое значение сигнализирует об ошибке.

## ПРИЛОЖЕНИЕ 1. Расширенные сообщения об ошибках

При исполнении большинства API команд и получении HTTP статуса об ошибке 400, рекомендуется дополнительно проверить тело ответа на наличие расширенного кода ошибки и текстового сообщения ошибки. Ответ сервера в случае получения этих ошибок будет содержать заголовок "Content-Type: application/json" и данные в формате json

```
{"RCODE":3xx, "RMESSAGE":"расширенное сообщение об ошибке"}
```

Список расширенных ошибок:

- 500 Internal server error
- 300 Invalid API Key / API Secret
- 301 Invalid Pass Number
- 302 Invalid Template Number
- 303 Invalid Value
- 304 Account Disabled
- 305 Requires POST method
- 306 Requires HTTPS
- 307 Data error
- 309 Invalid command
- 310 Invalid Serial Number
- 311 Invalid Template
- 312 Update failed
- 313 Invalid location data
- 314 Missing data
- 315 Invalid label
- 316 Too many fields
- 317 Date/Time must be in W3C format
- 318 Push notification failed
- 319 Card already exists
- 320 Error creating card
- 321 SMS send failed
- 322 Invalid mobile number
- 323 Move failed
- 324 Failed to get card download link
- 325 Invalid card link type request
- 326 Unknown target for change
- 327 Update defaults failed
- 328 Invalid SMS template. Must include {}
- 329 Invalid email address
- 330 Error accessing current barcode values
- 331 Invalid barcode format
- 332 Invalid barcode message/signature show parameter
- 333 Invalid barcode message/signature type parameter
- 334 Unsupported barcode encoding

335 Barcode message has symbols illegal for configured encoding  
340 ACL - Access denied  
341 Invalid or missing image data  
342 Requested image type is not supported  
343 Failed to delete requested resource  
344 Invalid Image Id  
345 Image is used by card(s)  
346 Invalid image data  
347 Invalid image type  
348 Image type is not allowed for current card style  
349 Invalid combination of images for current card style  
350 Invalid color data  
357 Invalid PIN  
358 Email send failed  
359 Invalid protection option  
360 Invalid sms text, missing {link}  
370 Invalid campaign  
371 Invalid page content  
372 Invalid/missing card issuing parameters  
373 Invalid/too long card prefix  
374 Invalid/too long possible serial number  
375 Too many cards requested  
376 Resulting serial number is too long (max 20 chars)  
377 Campaign already exists  
378 Invalid date/time  
379 SMS provider error  
390 Invalid card style  
391 Invalid appStore data  
392 Template not empty  
400 Invalid link/serial  
415 Invalid registration group  
420 Send pass via email functionality disabled  
421 No recipient address  
422 Error creating attachment  
423 Card was sent via email to  
424 language not supported  
425 Database insert failure  
426 Language is not enabled for organization  
427 Language is not enabled for account  
610 Too many cards in the batch  
619 Link already exists

## ПРИЛОЖЕНИЕ 2. Примеры использования

### Выдача новой карты

При выдаче новой карты оптимальной является следующая последовательность действий (подразумевается, что шаблон уже был создан на портале):

1. Создать карту с новым серийным номером
2. Установить персонализированные значения полей, таких как индивидуальная скидка и/или имя владельца и, возможно, срок действия карты
3. отправить карту на телефон пользователя

### Автоматическое изменение поля

Обновить поле, например с меткой "Бонусы", используя команду "Обновить значение карты", с установленным параметром /push. Это приведет к посылке нотификации на карту сразу после обновления и, как следствие, автоматическому обновлению карты на телефоне клиента.

### Задание индивидуального внешнего вида карт

Если требуется изменить внешний вид индивидуальной карты, например показать фото пользователя или индивидуальное сообщение пользователю в графическом формате, то оптимальной является следующая последовательность действий:

1. Добавить требуемый графический файл в библиотеку и получить новый `imgId`, или воспользоваться уже имеющимся `imgId`.
2. Обновить карту, присвоив ей полученный `imgId` и новую цветовую палитру.
3. Если использование индивидуальной графики картой было разовое или привязано ко времени, то обновить карту, удалив ранее присвоенный ей `imgId`.
4. Не обязательно удалять файл с `imgId` из библиотеки так как он может быть повторно использован в дальнейшем.

## ПРИЛОЖЕНИЕ 3. Использование библиотек

При работе с API в среде Microsoft Windows и Linux для быстрой интеграции с существующими управляющими и CRM системами удобно использовать уже существующие REST библиотеки.

Мы рекомендуем библиотеку [RestSharp](#) как самую, на наш взгляд, удобную и поддерживающую .Net не только в среде MS Windows, но и MONO в среде Linux.

Для среды java рекомендуется библиотеки:

RESty: <http://beders.github.io/Resty/Resty/Overview.html> или

AsyncHttpClient <https://github.com/AsyncHttpClient/async-http-client>

*ООО ОСМИ может предоставить готовые примеры интеграционного кода для работы с порталом OSMICards через REST API. В частности ООО ОСМИ предлагает готовую библиотеку .Net для интеграции с API и примеры кода ее использования.*

### cURL

При тестировании интеграционных API вручную с командной строки можно использовать программу cURL <http://curl.haxx.se/download.html>

При выборе версии программы необходимо загружать версию cURL с SSL.

Более удобным является использование GUI REST клиента. Очень удобным является бесплатный клиент от компании WizTools, который можно загрузить по ссылке

<http://code.fosshub.com/WizToolsorg-RESTClient/downloads>

### Среда 1С

В среде 1С до версии 8.3 включительно при использовании встроенного объекта "HTTPСоединение" наблюдаются проблемы с авторизацией при доступе к API серверу. Проблема заключается в неправильной работе механизма HTTP digest authentication у встроенного объекта "HTTPСоединение".

Рекомендуется использовать внешний HTTP механизм (разумеется с правильно заданными параметрами Header и Open) как описано в <http://infostart.ru/public/166276/> с добавлением дополнительной команды `WinHTTP.setCredentials("API ID", "API KEY", 0);` после исполнения команды `WinHTTP.Open(...)`.

ОСМИ предлагает готовый модуль для 1С, встраиваемый в любую типовую конфигурацию. Модуль сертифицирован компанией 1С. В модуле реализованы все необходимые функции и задачи по работе с электронными картами, рабочим местом кассира и маркетингом.

## ПРИЛОЖЕНИЕ 4. Работа с баркодами

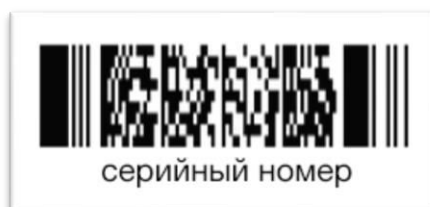
### Поддерживаемые типы баркодов

Электронные карты поддерживают баркоды четырех видов:

1. QR
2. PDF417
3. AZTEC
4. Code-128 (iOS 9 и Android)



QR Формат



PDF417 формат



AZTEC формат

### Общие замечания

В большинстве случаев достаточно установить в шаблоне поля **messageType** и **signatureType** в `'-serial-'`, **show** и **showSignature** в `'true'`, **encoding** в `'iso-8859-1'`.

При использовании API для создания новой карты, ее поля **show** и **showSignature** будут автоматически установлены в `'true'`.

При таком сценарии не будет нужды записывать индивидуальные серийные номера в поля **message** и **signature** каждой карты. В результате на новых картах будут присутствовать баркод с закодированной графикой серийного номера и текст серийного номера под ним.

### Структура баркода

При обновлении полей баркода в шаблоне или карте можно не передавать весь пакет полей. Если индивидуальное поле не присутствует в пакете обновления баркода, то его текущее значение не изменится при обновлении.

#### **format**

Определяет тип баркода и может принимать значения **QR**, **PDF417**, **AZTEC**, **CODE128**. Задается в шаблоне карты. Не может быть изменен для индивидуальной карты.

#### **protect**

Автоматически шифрует содержимое баркода. Обычно используется в схемах защиты карт от клонирования. Рекомендуется использовать только с типом содержимого баркода **messageType** `-text-`.

## encoding

Задаёт кодировку символов для поля **message**. Может принимать значения '**iso-8859-1**' и '**iso-8859-5**'. Задаётся в шаблоне карты. Не может быть изменён для индивидуальной карты.

**iso-8859-1** разрешает использование только латинского набора символов, но обеспечивает полную совместимость со всеми сканерами. **iso-8859-5** позволяет использовать латинские и русские символы, но может быть несовместим с некоторыми моделями сканеров и потребует дополнительной конфигурации сканера.

## messageType

Определяет что будет закодировано в графику баркода. Может принимать значения '**-serial-**', '**-text-**'. Задаётся в шаблоне карты. Не может быть изменён для индивидуальной карты.

Если это поле установлено в '**-text-**', то текст из поля **message** для этой карты будет использован и закодирован в графику баркода.

Если установлено в '**-serial-**', то текст из поля **message** для этой карты будет проигнорирован и серийный номер карты будет закодирован в графику баркода вместо него.

## message

Набор символов, который может быть закодирован в графику баркода. Устанавливается индивидуально для каждой карты.

Символы, разрешённые для использования при кодировании в графику баркода, должны быть из набора разрешённой кодировки. Кодировка задаётся в шаблоне, а именно значением поля **encoding** шаблона.

Если **messageType** шаблона установлено в '**-text-**', но поле **message** карты не содержит никаких символов (пустое), баркод на телефоне не отобразится.

## signature

Набор символов для отображения под графикой баркода. Устанавливается индивидуально для каждой карты.

Если **signatureType** шаблона установлено в '**-text-**', но поле **signature** карты не содержит никаких символов (пустое), подпись под баркод на телефоне не отобразится.

## signatureType

Определяет что будет отображено под графикой баркода. Может принимать значения '**-serial-**', '**-text-**'. Задаётся в шаблоне карты. Не может быть изменён для индивидуальной карты.

Если это поле установлено в '**-text-**', то текст из поля **signature** для этой карты будет использован и отображён под графикой баркода.



Если установлено в **'-serial-'**, то текст из поля **signature** для этой карты будет проигнорирован и серийный номер карты будет отображен под графикой баркода вместо него.

Если в результате комбинации параметров шаблона и карты графика баркода не отображается, то и подпись под графикой тоже не отобразится.

#### **show (шаблон)**

Глобально запрещает или разрешает показ баркода для всех карт данного шаблона. Может принимать значения **true**, **false**. Задается в шаблоне карты. Не может быть изменен для индивидуальной карты.

Если установлен в **true**, то показ баркода на индивидуальной карте будет определяться соответствующим параметром **show** этой карты.

Если установлен в **false**, то однозначно запрещает показ баркодов и подписей под ними на всех картах данного шаблона, независимо от установленного значения параметра **show** на индивидуальных картах.

#### **show (карта)**

Индивидуально запрещает или разрешает показ баркода для заданной карты. Может принимать значения **true**, **false**. Устанавливается индивидуально для каждой карты.

Если установлен в **true** и при условии, что в шаблоне карты параметр **show** установлен в **true**, разрешает показ баркода для заданной карты.

Если установлен в **false** и при условии, что в шаблоне карты параметр **show** установлен в **true**, запрещает показ баркода для заданной карты.

#### **showSignature (шаблон)**

Глобально запрещает или разрешает показ подписи под графикой баркода для всех карт данного шаблона. Может принимать значения **true**, **false**. Задается в шаблоне карты. Не может быть изменен для индивидуальной карты.

Если установлен в **true**, то показ подписи под графикой баркода на индивидуальной карте будет определяться соответствующим параметром **showSignature** этой карты.

Если установлен в **false**, то однозначно запрещает показ подписи под графикой баркода на всех картах данного шаблона, независимо от установленного значения параметра **showSignature** на индивидуальных картах.

#### **showSignature (карта)**

Индивидуально запрещает или разрешает показ подписи под графикой баркода для заданной карты. Может принимать значения **true**, **false**. Устанавливается индивидуально для каждой карты.

Если установлен в **true** и при условии, что в шаблоне карты параметр **showSignature** установлен в **true**, разрешает показ подписи под графикой баркода для заданной карты.



Если установлен в **false** и при условии, что в шаблоне карты параметр **showSignature** установлен в **true**, запрещает показ подписи под графикой баркода для заданной карты.

## Примеры

### Расширенная информация о баркоде

GET /passes/{serial}?extendedInfo=true

При запросе расширенной информации о карте возвращается 3 поля, относящихся к баркоду:

- Параметры карты
- Параметры шаблона
- Эффективная комбинация параметров карты и шаблона

Эффективная комбинация является наиболее точным представлением того, как баркод отобразится на электронной карте. В данном примере поля карты **message** и **signature** были заполнены индивидуальной информацией. Но поскольку в шаблоне поле **signature** и поле **signatureType** были установлены в **'-serial-'**, то в результате эффективное представление баркода на карте использовало серийный номер карты для содержимого баркода и подписи под ним вместо индивидуальных значений карты.

```
"barcode" : {
  "card" : {
    "showSignature" : true,
    "show" : true,
    "message" : "My barcode text",
    "signature" : "5555",
    "protect" : false
  },
  "effective" : {
    "showSignature" : true,
    "show" : true,
    "message" : "-serial-",
    "signature" : "-serial-",
    "protect" : false
  },
  "template" : {
    "format" : "QR",
    "signatureType" : "-serial-",
    "showSignature" : true,
    "show" : true,
    "protect" : false,
    "messageType" : "-serial-",
    "encoding" : "iso-8859-1"
  }
}
```

Если позднее карта эта карта была перемещена на шаблон, у которого поле **signature** и **signatureType** установлены в **'-text-'**, то баркод на карте обновится и покажет **"5555"** под графикой баркода, а содержимое графики самого баркода изменится на закодированную строку **"my barcode text"**.

В данном примере в шаблоне карты установлено значение **protect=false**. При создании карты это значение скопируется в поле баркода карты. Далее при редактировании карты это поле можно изменять независимо от значения соответствующего поля в шаблоне. Включать защиту баркода следует только в случае когда тип содержимого баркода **messageType** установлено в **«-text-»**

## Стандартная информация о баркоде

GET /passes/{serial}

При запросе информации о карте возвращается только одно поле, относящееся к баркоду:

```
"barcode" : {  
  "format" : "QR",  
  "signatureType" : "-serial-",  
  "showSignature" : true,  
  "show" : true,  
  "protect" : false,  
  "messageType" : "-serial-",  
  "encoding" : "iso-8859-1"  
}
```

Это поле является эффективной комбинацией параметров и данных карты и шаблона и наиболее точно представляет то, как баркод отобразится на электронной карте.

## Баркод шаблона

GET /templates/{имя шаблона}

PUT /templates/{имя шаблона}

При запросе информации о шаблоне возвращается одно поле, относящееся к баркоду.

```
"barcode" : {  
  "format" : "QR",  
  "signatureType" : "-serial-",  
  "showSignature" : true,  
  "show" : true,  
  "protect" : false,  
  "messageType" : "-serial-",  
  "encoding" : "iso-8859-1"  
}
```

В этом поле отображается информация о параметрах баркода данного шаблона.

При обновлении параметров шаблона приведенный формат следует использовать для обновления всех или индивидуальных полей баркода шаблона.

## Обновление информации для баркода карты

PUT /passes/{serial}

```
"barcode" : {  
  "showSignature" : true,  
  "show" : true,  
  "message" : "My barcode text",  
  "signature" : "5555",  
  "protect" : false  
}
```

При обновлении информации о баркоде карты приведенный формат следует использовать для обновления всех или индивидуальных полей баркода карты. Эффективное отображение баркода на карте будет также зависеть от параметров баркода, установленных в шаблоне карты. Параметр protect изначально копируется в карту из шаблона, но потом может индивидуально настраиваться для каждой карты.