

# Содержание

<b>1</b>	<b>Описание сервиса</b>	<b>2</b>
1.1	Назначение . . . . .	2
1.2	Схема работы . . . . .	2
<b>2</b>	<b>Авторизация и аутентификация</b>	<b>2</b>
<b>3</b>	<b>Перечень методов сервиса</b>	<b>2</b>
<b>4</b>	<b>Коды ответов</b>	<b>3</b>
<b>5</b>	<b>Описание методов</b>	<b>3</b>
5.1	invoice . . . . .	3
5.2	order-info . . . . .	7
5.3	payment . . . . .	11
5.4	order-status . . . . .	13
5.5	order-cancel . . . . .	14
5.6	invoice/data . . . . .	15
<b>6</b>	<b>Доступность методов</b>	<b>20</b>
<b>7</b>	<b>Перечисления</b>	<b>20</b>
<b>8</b>	<b>Обработка ошибок</b>	<b>20</b>
<b>9</b>	<b>Фискализация продаж</b>	<b>21</b>
<b>A</b>	<b>ПРИЛОЖЕНИЕ</b>	<b>22</b>

# 1 Описание сервиса

## 1.1 Назначение

Сервис платежей представляет собой отдельную информационную базу, выполненную на платформе 1С. Предназначен для использования в качестве промежуточного звена при проведении платежных транзакций для удаленных продаж товаров.

Возможна работа сервиса с несколькими юридическими лицами, несколькими ККМ (одна ККМ для одного юридического лица, одна ККМ для каждого подразделения одного юридического лица).

Возможна работа сервиса с несколькими платежными страницами (одна платежная страница для одного юридического лица, одна платежная страница для каждого подразделения одного юридического лица).

## 1.2 Схема работы

Предполагается следующая схема работы:

1. Менеджер, получив от клиента согласие на оплату услуг/товаров, экспортирует в сервис счет на оплату.
2. Сервис формирует и отдает менеджеру короткую ссылку на страницу оплаты.
3. Менеджер произвольным транспортом доставляет полученную короткую ссылку клиенту.
4. Клиент переходит по полученной ссылке на страницу оплаты, нажимает на кнопку оплатить и переходит на страницу интерфейса платежной системы.
5. Сервис, получив данные об успешной оплате клиентом счета, проводит фискализацию операции, автоматически пробивая чек на ККМ.

Промежуточная информационная база требуется по следующим причинам:

1. Безопасность. В случае получения злоумышленником несанкционированного доступа к платежному сервису основная информационная база остается не скомпрометированной.
2. Отказоустойчивость. При отказе платежного сервиса основная информационная база сохраняет полную работоспособность.

# 2 Авторизация и аутентификация

Предполагается к использованию базовый тип аутентификации пользователей сервиса. Логин и пароль для доступа выдает администратор ИБ сервиса.

# 3 Перечень методов сервиса

Методы сервиса платежей:

1. invoice — записывает счет на оплату в ИБ.
2. order-info — выдает полную информацию о запрошенном счете на оплату.
3. payment — устанавливает информацию об оплате счета, полученную от платежного шлюза банка.
4. order-status — выдает статус запрошенного счета.

- 5. order-cancel – отменяет счет на оплату, если это возможно.
- 6. invoice/data – позволяет записать/прочитать файл счета на оплату (печатную форму)

## 4 Коды ответов

Сервис возвращает следующие коды ответов:

- 200 - во всех случаях, когда запрос может быть обработан системой.
- 503 - обработка данных отключена.
- Иные коды, предусмотренные стандартом HTTP в прочих случаях.

## 5 Описание методов

### 5.1 invoice

POST

Производит импорт счета на оплату в ИБ сервиса.

#### Входные данные

В теле запроса метод принимает структуру, описывающую счет на оплату, в JSON.

```

1  {
2  "id" : "",
3  "order_date" : "",
4  "order_number" : "",
5  "incoming_date" : "2020.03.04 12:00",
6  "incoming_number" : "ТМТ0000015",
7  "company" : "ООО ТОП МОТО",
8  "company_uid" : "идентификатор_организации",
9  "departament": "HD Москва",
10 "departament_uid": "идентификатор_подразделения",
11 "customer" : "John Doe",
12 "customer_phone" : "",
13 "customer_email" : "",
14 "amount" : 2200.00,
15 "amount_of_payment" : 2200.00,
16 "amount_of_payment_without_VAT" : 2200,
17 "calculation_object": "ТОВАР",
18 "calculation_method": "АВАНС",
19 "VAT_RATE": "VAT_120",
20 "VAT": 200.00,
21 "payment_basis": "что-то для основания платежа",
22 "currency_code" : "643",
23 "order_status" : "",
24 "payment_deadline" : "2020-03-05T15:40:00Z",
25 "order_printed_form": [],
26 "items": [
27     {
28         "item": "КРЫЛО ЛЕВОЕ",
29         "is_service": 0,
30         "is_comission_item": 1,
31         "artile": "",
32         "count": 1.000,
33         "cost": 1200.00,
34         "sum": 1200.00,
35         "VAT_rate": "VAT_NONE",
36         "VAT": 0.00,
37         "sum_with_VAT": 1200.00,
38         "supplier": {
39             "supplier_TIN": "7716536701",
40             "supplier_name": "",
41             "supplier_phone": "723-58-95"
42         }
43     }
44 ]
45 }

```

Listing 1: Входной пакет метода invoice.

### Поля входной структуры

- id - идентификатор счета на оплату в ИБ сервиса, строка.
- date - дата счета на оплату в ИБ сервиса, строка.
- number - номер счета на оплату в ИБ сервиса, строка.

- incoming\_date - дата счета на оплату, передаваемая из внешней системы, строка, обязательный.
- incoming\_number - номер счета на оплату, передаваемый из внешней системы, строка, обязательный.
- company - наименование организации, строка.
- company\_uid - идентификатор организации, строка, обязательный.
- departament - наименование подразделения, строка.
- departament\_uid - идентификатор подразделения, строка.
- customer - плательщик, строка, обязательный.
- customer\_phone - телефон плательщика, строка.
- customer\_email - адрес электронной почты плательщика, строка.
- amount - сумма счета на оплату, число.
- amount\_of\_payment - сумма к оплате, число, обязательный.
- amount\_of\_payment\_without\_VAT - сумма к оплате без НДС, число, обязательный.
- calculation\_object - признак предмета расчета, строка, обязательный.
- calculation\_method - признак метода расчета, строка, обязательный.
- VAT\_RATE - ставка НДС, перечисления, обязательный.
- VAT - сумма НДС, число.
- payment\_basis - основание выполнения платежа, строка, обязательный.
- currency\_code - код валюты, строка, обязательный.
- currency - наименование валюты, сокращенное, строка.
- order\_status - статус счета на оплату, перечисление.
- payment\_deadline - дата, до которой необходимо оплатить счет, дата, обязательный.
- items - массив товарных позиций, обязательный.
- order\_printed\_form - структура, описывающая прикрепленные к счету на оплату файлы.

#### **Поля структуры–элемента массива номенклатуры**

- item - наименование номенклатуры, строка, обязательный.
- is\_service — признак услуги, 0 — товар, 1 — услуга. По умолчанию всегда товар.
- is\_comission\_item — признак комиссионного товара, пока не обрабатывается, т. к. нет ясности как это делать. По умолчанию товар наш.
- article - артикул номенклатуры, строка.
- count - количество номенклатуры, число, обязательный.
- cost - стоимость номенклатуры, число, обязательный.
- sum - сумма номенклатуры, число, обязательный.

- VAT\_rate - ставка НДС, перечисление, обязательный.
- VAT - сумма НДС, число.
- sum\_with\_VAT - сумма с НДС, число, обязательный.
- supplier - данные поставщика номенклатуры. передавать обязательно при is\_comission\_item = 1
- supplier\_TIN - ИНН поставщика номенклатуры, строка.
- supplier\_name - наименование поставщика номенклатуры, строка.
- supplier\_phone - телефон поставщика номенклатуры, строка.

Для плательщика необходимо передавать адрес электронной почты или телефон, для передачи данных ОФД.

Нельзя передавать недействительный ИНН поставщика, т. к. ККМ может быть настроена на его проверку. В этом случае фискализация платежа не будет проведена.

Сумма к оплате может не совпадать с суммой с НДС товаров/услуг в массиве номенклатуры, т.к. номенклатура передается только для информирования клиента.

Идентификация счетов производится по входящему номеру и организации. Если они повторились, то система не создаст счет повторно, а выведет сообщение с ошибкой. Если необходимо выставить клиенту счет повторно — создайте его заново.

## Выходные данные

Результат работы метода представляет собой структуру следующего вида.

```

1 {
2     "id": "35059545-758a-11ea-8119-0050568175b3",
3     "order_number": "TMT0000015",
4     "order_status": "NEW",
5     "order_shortlink": "h-dmoscow.shop/AsBScrQC"
6 }
```

Listing 2: Выходной пакет метода invoice.

## Поля результирующей структуры

- id - идентификатор созданного в ИБ сервиса счета на оплату, строка.
- order\_number - исходный номер счета на оплату, строка.
- order\_date - исходная дата счета на оплату, строка.
- order\_status — текущий статус в платежной системе, перечисление.
- order\_shortlink — короткая ссылка, подлежащая передаче клиенту.

Метод осуществляет формирование короткой ссылки на загруженный документ. Предусмотреть обработку случая, когда в поле короткой ссылки будет возвращена пустая строка (сформировать короткую ссылку не удалось, но запись документа произведена корректно).

## Коды ошибок метода

Код	Сообщение	Описание
3	parameter 'id' is not valid	Значение параметра не валидно
4	invoice already exist	Счет с указанными входными реквизитами уже существует
5	invoice is overdue	Дата оплаты счета меньше текущей
5	data source not found	Источник данных не найден
6	payments are not accepted	Неверный идентификатор организации, либо установлена блокировка платежей по организации
7	parameter 'item' is not valid	Значение поля табличной части не валидно
9	error saving data	Ошибка создания счета на оплату
10	error saving data	Ошибка добавления данных о счете

Таблица 1: Коды ошибок метода invoice.

## 5.2 order-info

### POST

Выдает полную информацию о счете на оплату, необходимую для показа клиенту и передаче платежному шлюзу.

### Входные данные

В теле запроса метод принимает структуру с короткой ссылкой на счет на оплату, в JSON.

```
1  {
2    "order_shortlink": "VSzVMHCq"
3  }
```

Listing 3: Входной пакет метода order-info.

### Поля входной структуры

- order\_shortlink - сформированная методом invoice короткая ссылка на счет на оплату, строка, обязательный.

### Выходные данные

Если счет возможно оплатить, то отдается структура с полной информацией о счете на оплату.

```

1  {
2  "id": "6ad02b7e-7592-11ea-8119-0050568175b3",
3  "order_date": "2020-04-03T10:03:51Z",
4  "order_number": "000000019",
5  "incoming_date": "04.03.2020 12:00",
6  "incoming_number": "TMT0000019",
7  "company": "ООО ТОН МОТО",
8  "company_uid": "",
9  "departament": "HD Москва",
10 "departament_uid": "",
11 "customer": "John Doe",
12 "customer_phone": "",
13 "customer_email": "",
14 "amount": 2200.00,
15 "amount_of_payment": 2200.00,
16 "VAT_rate": "VAT_120",
17 "VAT": 200.00,
18 "currency_code": "643",
19 "currency": "RUB",
20 "order_status": "NEW",
21 "items": [
22     {
23         "item": "КРЫЛО ЛЕБОВОЕ",
24         "is_service": 0,
25         "is_comission_item": 0,
26         "supplier": {
27             "supplier_phone": "",
28             "supplier_TIN": "",
29             "supplier_name": ""
30         },
31         "article": "",
32         "count": 1.000,
33         "cost": 1000.00,
34         "sum": 1000.00,
35         "VAT_rate": "VAT_NONE",
36         "VAT": 0.00,
37         "sum_with_VAT": 1000.00
38     }
39 ],
40 "payment_deadline": "2020-03-05T12:40:00Z",
41 "order_printed_form": []
42 }

```

Listing 4: Выходной пакет метода order-info в случае возможности оплатить счет.

Если счет уже оплачен или отменен, то отдается структура с информацией о статусе счета на оплату.



```

1      {
2          "id": "6ad02b7e-7592-11ea-8119-0050568175b3",
3          "order_number": "TMT0000019",
4          "order_date": "04.03.2020 12:00",
5          "order_status": "CANCELED",
6          "amount": 2200.00,
7          "payment_system": "VISA",
8          "payment_date": "0001-01-01T00:00:00Z",
9          "fiscal_date": "0001-01-01T00:00:00Z"
10     }

```

Listing 5: Выходной пакет метода order-info в случае отмененного счета на оплату.

В реквизите order\_printed\_form, если он заполнен отдается массив, элементы которого описывают присоединенные к счету на оплату файлы. Элементом массива является структура следующего вида:

```

1      {
2          "id_file": "1a50cde1-7d9a-11ea-811a-0050568175b3",
3          "description": "Счет на оплату TMT000019 от 12.04.2020 г.",
4          "type": "invoice",
5          "format": "png",
6          "size": 4.92
7      }

```

Listing 6: Структура, описывающая файл, прикрепленный к счету на оплату.

- id - идентификатор счета на оплату в ИБ сервиса, строка.
- date - дата счета на оплату в ИБ сервиса, строка.
- number - номер счета на оплату в ИБ сервиса, строка.
- incoming\_date - дата счета на оплату, передаваемая из внешней системы, строка, обязательный.
- incoming\_number - номер счета на оплату, передаваемый из внешней системы, строка, обязательный.
- company\_uid - идентификатор организации, строка, обязательный.
- customer - плательщик, строка, обязательный.
- customer\_phone - телефон плательщика, строка.
- customer\_email - адрес электронной почты плательщика, строка.
- amount - сумма счета на оплату, число.
- amount\_of\_payment - сумма к оплате, число, обязательный.
- amount\_of\_payment\_without\_VAT - сумма к оплате без НДС, число, обязательный.
- calculation\_object - признак предмета расчета, строка, обязательный.
- calculation\_method - признак метода расчета, строка, обязательный.
- VAT\_RATE - ставка НДС, перечисления, обязательный.

- VAT - сумма НДС, число.
- payment\_basis - основание выполнения платежа, строка, обязательный.
- currency\_code - код валюты, строка, обязательный.
- currency - наименование валюты, сокращенное, строка.
- order\_status - статус счета на оплату, перечисление.
- payment\_deadline - дата, до которой необходимо оплатить счет, дата, обязательный.
- items - массив товарных позиций, обязательный.
- order\_printed\_form - структура с описанием прикрепленных файлов, массив.
- id\_file - идентификатор прикрепленного файла, строка.
- description - описание прикрепленного файла, строка.
- size - размер прикрепленного файла в килобайтах, число.
- format - расширение прикрепленного файла, без точки, строка.
- type - тип прикрепленного файла, перечисление.

#### **Поля структуры-элемента массива номенклатуры**

- item - наименование номенклатуры, строка, обязательный.
- is\_service — признак услуги, 0 — товар, 1 — услуга. По умолчанию всегда товар.
- is\_comission\_item — признак комиссионного товара, пока не обрабатывается, т. к. нет ясности как это делать. По умолчанию товар наш.
- article - артикул номенклатуры, строка.
- count - количество номенклатуры, число, обязательный.
- cost - стоимость номенклатуры, число, обязательный.
- sum - сумма номенклатуры, число, обязательный.
- VAT\_rate - ставка НДС, перечисление, обязательный.
- VAT - сумма НДС, число.
- sum\_with\_VAT - сумма с НДС, число, обязательный.
- supplier - данные поставщика номенклатуры. передавать обязательно при is\_comission\_item = 1 (требуется фискальному регистратору)
- supplier\_TIN - ИНН поставщика номенклатуры, строка.
- supplier\_name - наименование поставщика номенклатуры, строка.
- supplier\_phone - телефон поставщика номенклатуры, строка.

## Поля структуры при невозможности оплатить счет

- id - идентификатор счета на оплату, строка.
- order\_number - номер счета на оплату исходный, строка.
- order\_date - дата счета на оплату исходная, строка.
- order\_status - статус счета на оплату текущий, перечисление.
- amount - сумма счета, число.
- payment\_system - платежная система плательщика, перечисление.
- payment\_date - дата платежа по счету, дата.
- fiscal\_date - дата фискализации платежа.

## Коды ошибок метода

Код	Сообщение	Описание
2	parameter 'order_shortlink' not found	Отсутствует обязательный параметр
3	parameter 'order_shortlink' is not valid	Значение параметра не валидно

Таблица 2: Коды ошибок метода order-info.

## 5.3 payment

### POST

Производит запись информации об оплате счета с указанным идентификатором в ИБ сервиса.

### Входные данные

Входная информация ожидается в виде структуры, в формате JSON.

```
1      {
2        "id": "806775a8-5edd-11ea-8118-0050568175b3",
3        "orderNumber": "TMT0000013",
4        "orderStatus": 2,
5        "actionCode": 0,
6        "actionCodeDescription": "",
7        "amount": 400,
8        "currency": "643",
9        "date": "2020-04-09T08:09:22Z",
10       "orderDescription": "Платеж по счету TMT000000013 от 08.04.2020",
11       "ip": "77.108.110.33",
12       "merchantOrderParams": [],
13       "cardAuthInfo": {},
14       "paymentAmountInfo": {},
15       "bankInfo": {}
16     }
```

Listing 7: Входной пакет метода payment.

## Поля входной структуры

- id - идентификатор счета, для которого устанавливается оплата, строка, обязательный.
- orderNumber -номер заказа, строка, обязательный.
- orderStatus - статус заказа, строка.
- actionCode - код результата, число, обязательный.
- actionCodeDescription -описание результата, обязательный.
- amount - сумма в копейках, число, обязательный.
- currency - код валюты, строка.
- date - дата транзакции, дата, обязательный.
- orderDescription - описание платежа, строка.
- ip - IP-адрес плательщика, строка, обязательный.
- merchantOrderParams - дополнительные параметры заказа, массив.
- cardAuthInfo - структура, описывающая карту, обязательный.
- paymentAmountInfo - структура, описывающая платеж, обязательный.
- bankInfo - структура, описывающая банк-эмитент карты.

Помимо указанных реквизитов, ответ платежного шлюза может содержать дополнительную информацию.

## Выходные данные

Выходные данные - структура в JSON, описывающая текущее состояние счета на оплату. Статус счета устанавливается в Оплачен.

```
1      {
2      "id": "6ad02b7e-7592-11ea-8119-0050568175b3",
3      "order_number": "TMT0000019",
4      "order_date": "04.03.2020 12:00",
5      "order_status": "PAID",
6      "amount": 4.00,
7      "payment_system": "VISA",
8      "payment_date": "2020-04-09T08:09:22Z",
9      "fiscal_date": "0001-01-01T00:00:00Z"
10     }
11
```

Listing 8: Выходной пакет метода payment.

## Поля результирующей структуры

- id - идентификатор счета на оплату, строка.
- order\_number - номер счета на оплату исходный, строка.
- order\_date - дата счета на оплату исходная, строка.
- order\_status - статус счета на оплату текущий, перечисление.

- amount - сумма счета, число.
- payment\_system - платежная система плательщика, перечисление.
- payment\_date - дата платежа по счету, дата.
- fiscal\_date - дата фискализации платежа.

#### Коды ошибок метода

Код	Сообщение	Описание
2	parameter 'id' not found	Отсутствует обязательный параметр
3	parameter 'parameter 'id' is not valid	Значение параметра не валидно
11	payment cannot be accepted	Статус счета не позволяет принять оплату (не Новый)
12	field amount not found	Ошибка разбора данных транзакции
13	error saving data	Ошибка сохранения данных платежа
14	error saving data	Ошибка изменения статуса счета после оплаты

Таблица 3: Коды ошибок метода payment.

## 5.4 order-status

### POST

Выдает краткую информацию о текущем статусе счета на оплату.

#### Входные данные

В теле запроса метод принимает структуру с идентификатором счета на оплату, в JSON.

```

1      {
2      "id": "35059545-758a-11ea-8119-0050568175b3"
3      }
```

Listing 9: Входной пакет метода order-status.

#### Поля входной структуры

- id - идентификатор счета на оплату, строка, обязательный.

#### Выходные данные

```

1      {
2      "id": "806775a8-5edd-11ea-8118-0050568175b3",
3      "order_number": "TMT0000009",
4      "order_date": "04.03.2020 12:00",
5      "order_status": "PAID",
6      "amount": 2200.00,
7      "payment_system": "VISA",
8      "payment_date": "2020-03-05T15:23:14Z",
9      "fiscal_date": "2020-03-04T21:00:00Z"
10     }
```

Listing 10: Выходной пакет метода order-status.

## Поля результирующей структуры

- id - идентификатор счета на оплату, строка.
- order\_number - номер счета на оплату исходный, строка.
- order\_date - дата счета на оплату исходная, строка.
- order\_status - статус счета на оплату текущий, перечисление.
- amount - сумма счета, число.
- payment\_system - платежная система плательщика, перечисление.
- payment\_date - дата платежа по счету, дата.
- fiscal\_date - дата фискализации платежа.

## Коды ошибок метода

Код	Сообщение	Описание
2	parameter 'id' not found	Обязательный параметр не найден
3	parameter 'id' is not valid	Значение параметра не валидно

Таблица 4: Коды ошибок метода order-status.

## 5.5 order-cancel

### POST

Производит отмену счета на оплату. Отмена возможна только для счетов, которые были не оплачены и не отменены ранее.

### Входные данные

В теле запроса метод принимает структуру с идентификатором счета на оплату, в JSON.

```
1  {
2    "id": "35059545-758a-11ea-8119-0050568175b3"
3  }
```

Listing 11: Входной пакет метода order-cancel.

## Поля входной структуры

- id - идентификатор счета на оплату, строка, обязательный.

## Выходные данные

Формат выходных данных соответствует методу order-status. Статус счета устанавливается в Отменен.

```

1  {
2  "id": "806775a8-5edd-11ea-8118-0050568175b3",
3  "order_number": "TMT0000009",
4  "order_date": "04.03.2020 12:00",
5  "order_status": "PAID",
6  "amount": 2200.00,
7  "payment_system": "VISA",
8  "payment_date": "2020-03-05T15:23:14Z",
9  "fiscal_date": "2020-03-04T21:00:00Z"
10 }

```

Listing 12: Выходной пакет метода order-cancel.

### Поля результирующей структуры

- id - идентификатор счета на оплату, строка.
- order\_number - номер счета на оплату исходный, строка.
- order\_date - дата счета на оплату исходная, строка.
- order\_status - статус счета на оплату текущий, перечисление.
- amount - сумма счета, число.
- payment\_system - платежная система плательщика, перечисление.
- payment\_date - дата платежа по счету, дата.
- fiscal\_date - дата фискализации платежа.

### Коды ошибок метода

Код	Сообщение	Описание
2	parameter 'id' not found	Обязательный параметр не найден
3	parameter 'id' is not valid	Значение параметра не валидно
4	invoice not found	Идентификатор счета не соответствует ни одному счету на оплату
5	incorrect status of invoice	Статус счета не позволяет выполнить отмену
5	invoice already canceled	Счет уже отменен
5	invoice already payment	Счет оплачен, отмена не возможна
14	error saving data	Ошибка операции отмены счета на оплату

Таблица 5: Коды ошибок метода order-cancel.

## 5.6 invoice/data

### GET, POST, DELETE

Методы позволяют импортировать документы различных видов и форматов для указанного счета на оплату, а также получать эти документы и удалять их.

### Входные данные

## Добавление файла

### Метод POST

Добавление файла производится отправлением на url вида **invoice/data/{id}** составного сообщения, первой частью которого является структура с именем **metadata**, описывающая передаваемые файлы. Второй и последующими частями (при их наличии) следуют файлы в двоичном виде. Имя каждой части ожидается как **name{i}**, где **i** - порядковый номер файла в теле запроса. Между собой все части разделены пустыми строками и разделителем.

Запрос должен иметь заголовок, указывающий на тип содержимого файла: **Content-Type : multipart/form-data**. Обязательно должно передаваться значение разделителя (boundary).

Каждая часть должна начинаться заголовками, указывающими на ее тип. Часть с метаданными предваряется заголовком **Content-Disposition: form-data; name=metadata**. Части с файлами предваряются заголовками **Content-Disposition: form-data; name=name{i}**.

Пример сформированного составного сообщения приведен в листинге ниже.

```
1      <Пустая строка>
2      ==Asrf456BGe4h
3      Content-Disposition: form-data; name=metadata
4      <Пустая строка>
5      {
6      "id": "806775a8-5edd-11ea-8118-0050568175b3",
7      "items": [
8          {
9              "description": "Счет на оплату ТМТ000019 от 12.04.2020 г.",
10             "type": "invoice",
11             "format": "png"
12         },
13         {
14             "description": "Рекламная листовка HD STREET ROD 2020 г.",
15             "type": "flyer",
16             "format": "jpeg"
17         }
18     ]
19 }
20 <Пустая строка>
21 ==Asrf456BGe4h
22 Content-Disposition: form-data; name=name1; filename=v8_20E4_1a;
23 <Пустая строка>
24 <Двоичные данные первого файла>
25 <Пустая строка>
26 ==Asrf456BGe4h
27 Content-Disposition: form-data; name=name2; filename=v8_20E4_2a;
28 <Пустая строка>
29 <Двоичные данные второго файла>
30 ==Asrf456BGe4h==
```

Listing 13: Пример составного HTTP-сообщения на добавление файла/файлов.

## Получение файла

### Метод GET

Получение всех файлов, ранее добавленных к счету на оплату, производится обращением на url вида **invoice/data/{id}**.

Получение всех файлов указанного типа, ранее добавленных к счету на оплату, производится обращением на url вида **invoice/data/{id}/{type}**.



Получение конкретного файла, ранее добавленного к счету на оплату, производится обращением на url вида **invoice/data/{id}/{type}/{id\_file}**.

### Удаление файла

Метод DELETE

Удаление всех файлов, ранее добавленных к счету на оплату, производится обращением на url вида **invoice/data/{id}**.

Удаление всех файлов указанного типа, ранее добавленных к счету на оплату, производится обращением на url вида **invoice/data/{id}/{type}**.

Удаление конкретного файла, ранее добавленного к счету на оплату, производится обращением на url вида **invoice/data/{id}/{type}/{id\_file}**.

### Параметры url

Во всех случаях:

- id - идентификатор счета на оплату, полученный методом invoice, строка.
- type - тип файла, перечисление.
- format - расширение файла, строка.
- id\_file - идентификатор файла, строка.

### Выходные данные

#### Добавление файла

В теле ответа возвращается структура с массивом, описывающая добавленные файлы, в JSON.

```
1      {
2          "id": "идентификатор_счета_на_оплату",
3          "items": [
4              {
5                  "id_file": "1a50cde1-7d9a-11ea-811a-0050568175b3",
6                  "description": "Счет на оплату TMT000019 от 12.04.2020 г.",
7                  "type": "invoice",
8                  "format": "png",
9                  "size": 4.92
10             }
11         ]
12     }
```

Listing 14: Выходной пакет операции добавления файла

### Получение файла/файлов

Результатом работы метода является составное HTTP-сообщение, содержащее внутри себя несколько вложенных сообщений.

Первой частью сообщения с именем **metadata** является структура в JSON, описывающая файлы, соответствующие запросу.

Второй и последующими частями (при наличии) являются файлы в двоичном виде. Каждая часть имеет имя вида **name{i}**, где i - порядковый номер файла. Также каждая из данных частей имеет атрибут **id\_file**.

```

1  {
2  "id": "806775a8-5edd-11ea-8118-0050568175b3",
3  "items": [
4      {
5          "id_file": "d29bb323-7d9f-11ea-811a-0050568175b3",
6          "description": "",
7          "type": "invoice",
8          "format": "png",
9          "size": 5042
10     },
11     {
12         "id_file": "d29bb322-7d9f-11ea-811a-0050568175b3",
13         "description": "",
14         "type": "invoice",
15         "format": "png",
16         "size": 5042
17     }
18 ]
19 }

```

Listing 15: Первая часть составного HTTP-сообщения – результата операции чтения файла/файлов

### Удаление файла/файлов

В теле метода возвращается структура с описанием удаляемых файлов, в JSON.

```

1  {
2  "id": "806775a8-5edd-11ea-8118-0050568175b3",
3  "items": [
4      {
5          "id_file": "f7ea203e-7d99-11ea-811a-0050568175b3",
6          "description": "Счет на оплату ТМТ000019 от 12.04.2020 г.",
7          "type": "invoice",
8          "format": "png",
9          "size": 5042
10     },
11     {
12         "id_file": "1a50cde1-7d9a-11ea-811a-0050568175b3",
13         "description": "",
14         "type": "invoice",
15         "format": "png",
16         "size": 5042
17     }
18 ]
19 }

```

Listing 16: Выходной пакет операции удаления файла/файлов

### Поля результирующих структур

- id - идентификатор счета на оплату, строка.
- items - массив структур, описывающих файлы.

- id\_file - идентификатор файла, строка.
- description - описание файла, строка.
- type - тип файла, перечисление.
- format - формат файла, его расширение.
- size - размер файла, в килобайтах.

## Коды ошибок метода

### Ошибки при сохранении файла

Код	Сообщение	Описание
2	parameter 'id' not found	Отсутствует обязательный параметр
3	parameter 'id' is not valid	Значение параметра не валидно
4	missing expected header or its value is incorrect	Обязательные заголовки отсутствуют, либо некорректны)
5	content size exceeds maximum	Превышение максимального размера файла
6	message parsing error	Ошибка разбора сообщения
7	file saving error	Ошибка сохранения файла

Таблица 6: Коды ошибок метода invoice/data при сохранении файла.

### Ошибки при получении файла

Код	Сообщение	Описание
2	parameter 'id' not found	Отсутствует обязательный параметр
3	parameter 'id_file' is not valid	Значение параметра не валидно
4	error receiving file	Ошибка при получении файла. Например, файл отсутствует.

Таблица 7: Коды ошибок метода invoice/data при получении файла.

### Ошибки при удалении файла

Код	Сообщение	Описание
2	parameter 'id' not found	Отсутствует обязательный параметр
3	parameter 'type' has an invalid value	Значение параметра не валидно
4	missing expected header or its value is incorrect	Обязательные заголовки отсутствуют, либо некорректны
5	content size exceeds maximum	Превышение максимального размера файла
6	content is missing	В запросе нет тела файла
7	file saving error	Ошибка сохранения файла

Таблица 8: Коды ошибок метода invoice/data при удалении файла.

## 6 Доступность методов

N	Метод	URL	Для платежной страницы	Для базы источника
1	invoice	/invoice	—	+
2	order-info	/order-info	+	—
3	payment	/payment	+	—
4	order-status	/order-status	—	+
5	order-cancel	/order-cancel	—	+
6	invoice/data, POST	/invoice/data/{id}	—	+
7	invoice/data, GET	/invoice/data/{id}	+	+
8	invoice/data, GET	/invoice/data/{id}/{type}	+	+
9	invoice/data, GET	/invoice/data/{id}/{type}/{id_file}	+	+
10	invoice/data, DELETE	/invoice/data/{id}	-	+
11	invoice/data, DELETE	/invoice/data/{id}/{type}	-	+
12	invoice/data, DELETE	/invoice/data/{id}/{type}/{id_file}	-	+

Таблица 9: Доступность методов сервиса

## 7 Перечисления

В сервисе используются следующие перечисления:

1. VAT\_RATE - ставка НДС; принимает следующие значения: VAT\_NONE, VAT\_10, VAT\_20, VAT\_120 (Без НДС, НДС 10%, НДС 20%, НДС 20/120% соответственно).
2. status\_order - статус счета на оплату; принимает следующие значения: NEW, PAID, CANCEL (Новый, Оплачен, Отменен соответственно).
3. calculation\_object - признак предмета расчета; принимает следующие значения: Товар, Работа, Услуга, Платеж, Агентское Вознаграждение, Иной Предмет Расчета, Внереализационный Доход.
4. calculation\_method - признак способа расчета; принимает следующие значения: Полная Предварительная Оплата, Частичная Предварительная Оплата, Аванс, Полный Расчет, Оплата Предмета Расчета После Передачи В Кредит.
5. type - типы документов, поддерживаемые сервисом; принимает следующие значения: invoice.
6. payment\_system - наименование платежной системы, принимает следующие значения: VISA, МИР, Diners Club, JCB International, American Express, Maestro, MasterCard, Discover, China UnionPay, УЭК.

## 8 Обработка ошибок

Сообщения об ошибках для всех методов имеют вид следующей структуры:

```
1 {
2     "code": 3,
3     "description": "parameter 'id' is not valid"
4 }
```

Listing 17: Выходной пакет любого метода в случае ошибки

Каждый метод имеет свою нумерацию кодов ошибок. Смотрите перечень кодов ошибок и их значение в описаниях соответствующих методов.

## 9 Фискализация продаж

Реализован процесс автоматической фискализации продаж. Для его выполнения сервис автоматически определяет на основе переданного при загрузке счета на оплату идентификатора организации (и подразделения - если для организации ведется учет по подразделениям) на какой ККМ необходимо осуществлять пробитие чека.

ККМ должна быть постоянно включенной в сеть питания, быть подключенной к компьютерной сети холдинга.

Выполнение фискализации продаж осуществляется регламентным заданием с установленным периодом в 5 минут.

Рассылка электронных чеков покупателям осуществляется оператором фискальных данных. Данные для отправки загружаются вместе со счетом на оплату.

Необходимо обращать особое внимание на корректность форматов телефона и электронного адреса. Телефон принимается в формате 7XXXXXXXXX, электронный адрес принимается в формате <строка>@<строка>. При указании неверных данных повторная рассылка электронного чека невозможна.

При указании и номера телефона и электронного адреса приоритет отдается электронному адресу.

## А ПРИЛОЖЕНИЕ

Данное приложение содержит проектную структуру информационной базы, кратко описывающую реквизиты всех используемых объектов.

[illegible]

Рис. 1: Структура таблиц информационной базы.



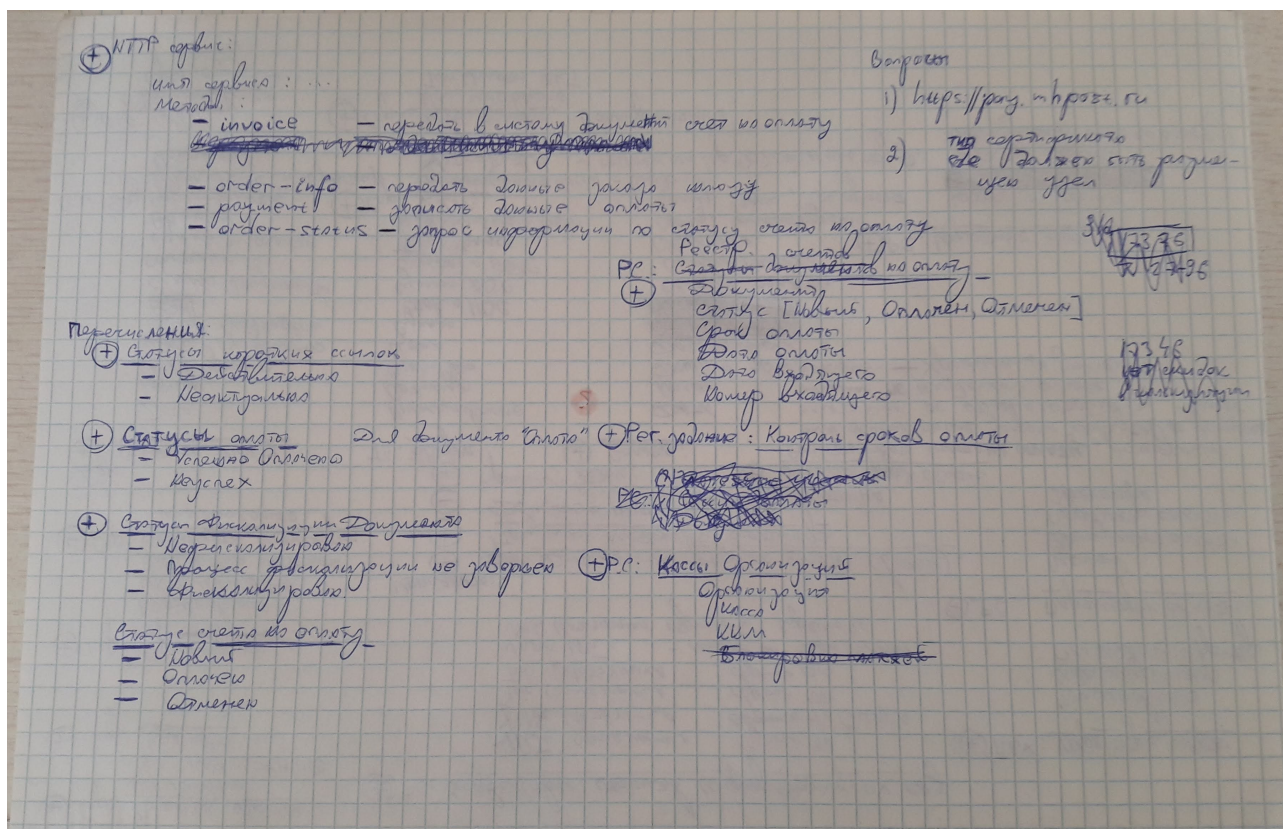


Рис. 2: Структура таблиц информационной базы.