

Протокол информационного взаимодействия в рамках проекта сопряжения Системы «Город» и платежной системы Банка



БАРНАУЛ

Оглавление

Введение	3
Интерфейс создания и проведения платежей	3
1. Разграничение обязанностей.....	3
2. Этапы обмена данными между Банком и Системой «Город», их назначение	3
3. Последовательность оплаты.....	3
4. Этапы жизненного цикла платежа	4
5. Принцип информационного взаимодействия	5
6. Авторизация API.....	5
7. Этап I. Получение/Пересчет формы	5
8. Сторнирование платежа	6
9. Получение списка дополнительных услуг	6
10. Корзина платежей	6
11. Получение статуса платежа	7
Интерфейс получения отчетов	8
1. Получение OAuth токена.....	8
2. Выполнение отчета.....	8
3. Загрузка платежных поручений через REST API	9
Приложение №1 Схема взаимодействия	10
Приложение №2 Примеры запросов	11
Приложение №3 Формат файл загрузки платежных поручения.....	18
Список изменений	19

Введение

Протокол реализуется двумя интерфейсами:
Интерфейс создания и проведения платежей (SOAP-API)
Интерфейс выполнения отчетов (REST-API)

Интерфейс создания и проведения платежей

1. Разграничение обязанностей

Система «Город» реализует программный интерфейс (API) при вызове которого Банк может производить поиск абонентов, получение и пересчет форм платежей, сохранение и проведение платежей. Интерфейс для взаимодействия с пользователем реализует Банк. При этом Банк обязуется реализовать протокол в полном объеме.

2. Этапы обмена данными между Банком и Системой «Город», их назначение

Этап I. Получение/Пересчет формы.	<p>На данном этапе, Банк отсылает текущее состояние формы в Систему «Город» (список полей и их значений), в ответ получает новые поля для отображения.</p> <p>Банк отображает форму платежа содержащую только те поля, которые были указаны в ответе на запрос, поля должны быть инициализированными значениями, указанными в ответе.</p> <p>Банк предоставляет возможность пользователю изменить значения полей формы переданных в ответе на запрос, перейти к следующему шагу. Этап I заканчивается сохранением платежа в Системе «Город».</p>
Этап II. Проведение платежа.	<p>Данный этап предполагает, что платеж сохранен и ожидает запрос на проведение.</p> <p>Осуществляется проверка соответствия сохраненного платежа информации указанной в запросе, проводится платеж.</p> <p>При отправке запросов на проведение платежа, необходимо учитывать, что платеж может отсутствовать в системе (этапы жизненного цикла платежа описаны в п.4.), в этом случае, в ответ на такой запрос будет возвращаться ошибка, Банк должен корректно обработать эту ситуацию.</p>
Сторнирование платежа	<p>Сторнироваться могут только проведенные платежи, текущего операционного дня.</p> <p>Сторнирование может быть запрещено поставщиком услуги.</p>

3. Последовательность оплаты

На каждом шаге, Банк должен передавать идентификатор сессии и поля заполненные пользователем. В ответ Банк, получает новые поля для отображения.

На любом шаге может быть возвращено поле «error_val» с описанием ошибки. Описание ошибки отображается плательщику и «шага не происходит», т.е. форма не изменяет состояния и плательщик имеет возможность отредактировать данные в полях повторно.

Этап I реализуется шагами 1-5, Этап II – шагом - 6.

3.1. Шаг 1 – поиск услуг

На этом шаге осуществляется поиск услуг.

Для поиска по лицевому счету необходимо заполнить поле find_account_val

Для поиска по адресу набор полей find_address_city_val, find_address_street_val, find_address_house_val, find_address_apartment_val (поле опционально) или find_fias_street_val, find_address_house_val, find_address_apartment_val (поле опционально)

Для повтора последней поиска передать поле find_again, значение поле игнорируется

Ожидаемые поля:

- find_account_val
- find_address_city_val

- find_address_street_val
- find_address_house_val
- find_address_apartment_val
- find_fias_street_val
- find_again
- session_id (заполняется при повторно поиске)

Ответ:

- session_id (при повторном поиске отсутствует)
- select_service_val (элемент типа list)

При оплате дополнительных услуг шаг 1 пропускается.

3.2. Шаг 2 – получение календаря

На этом шаге система возвращает список для выбора периода, за который осуществляется оплата.

Ожидаемые поля:

- session_id (не заполняется, в случае если шаг 1 пропущен)
- select_service_val

Ответ:

- address
- find_account_val (в случае поиска по адресу)
- calendar (элемент типа list)

3.3. Шаг 3 – получение формы оплаты услуги

На этом шаге система возвращает поля формы оплаты.

Ожидаемые поля:

- session_id
- calendar

Ответ:

- Поля формы, наименование и типы полей указываются в ответе

3.4. Шаг 4 – пересчет формы оплаты

На этом шаге система пересчитывает форму оплаты.

Ожидаемые поля:

- session_id
- все поля формы полученные на предыдущем шаге

Ответ:

- Пересчитанные поля формы (все поля не редактируемые)
- paysum (сумма платежа, включая комиссию с абонента)
- commission (сумма комиссии, с абонента)
- need_identification (необходимость идентификации плательщика: 0 – идентификация не требуется; 1 – необходима идентификация)

3.5. Шаг 5 – сохранение платежа в корзину, получение чека

На этом шаге система пересчитывает форму оплаты.

Ожидаемые поля:

- session_id
- email_phone – телефон или адрес электронной почты, для фискализации платежа получателем, поле необязательно

Ответ:

- upr (уникальный номер корзины)
- paysum (сумма платежа, включая комиссию с абонента)
- commission (сумма комиссии, с абонента)
- Поля содержащие строки чека

3.6. Шаг 6 – проведение оплаты

Для осуществления этого шага, необходимо воспользоваться функцией **invoicePay**.

Ожидаемые поля:

- upr
- paysum (сумма платежа, включая комиссию с абонента)
- session_id

Ответ:

- ОК (платеж проведен), либо ERROR (платеж не проведен, ошибка)

4. Этапы жизненного цикла платежа

- I. Сохранение платежа в системе происходит после обработки «Шаг 5»;

- II. В течении одного часа с момента сохранения, платеж в системе имеет активное состояние и может быть проведен - «Шаг 6»;
- III. По истечении часа и до восьми суток с момента сохранения платежа, платеж в системе может быть удален, проведение платежа в этом случае будет невозможно;
- IV. По истечении восьми суток с момента сохранения платежа, платеж в системе автоматически удаляется, проведение платежа будет невозможно.

После проводки платеж относится к операционному дню соответствующему календарному времени проведения платежа (по Барнаульскому времени GMT+7), сторнирование проведенного платежа возможно только из операционного дня соответствующему календарному времени сторнирования.

5. Принцип информационного взаимодействия

Информационное взаимодействие происходит между банковской системой и Системой «Город»

Целью взаимодействия является предоставление возможности абонентам оплачивать жилищно-коммунальные и другие виды услуг через Систему «Город».

Информационное взаимодействие происходит при помощи технологии SOAP API

Используется SOAP 1.2

Набор используемых функций в SOAP API динамический и зависит от потребностей Организации.

Взаимодействие осуществляется с помощью протокола HTTP

Кодировка передаваемых данных: UTF-8

Максимальное количество одновременно обрабатываемых запросов любых типов: 50

6. Авторизация API

Доступ к API осуществляется по закрытым каналам связи, организация каналов связи обязанность Банка.

Ко всем функциям API применяется дополнительная защита в виде логина и пароля, согласно спецификации WS-Security 1.0. Имя пользователя и пароль могут быть сформированы по запросу Банка.

7. Этап I. Получение/Пересчет формы

Для получения или пересчета формы необходимо воспользоваться функцией **formCalc**.

Функции необходимо передать массив данных:

Переменная	Формат	Описание
messageDate	dateTime	дата создания этого сообщения
payType	string(50)	тип платежа, необходимо согласовать справочник значений
payPoint	string(50)	пункт приема платежа, необходимо согласовать справочник значений
payChannel	string(50)	канал приема платежа, необходимо согласовать справочник значений
fields		список полей формы
vame	string(100)	название поля (максимальная длина поля 100 символов)
value	string(100)	значение поля (максимальная длина поля 100 символов)

Результатом работы функции будет также массив данных:

Переменная	Формат	Описание
messageDate	dateTime	дата создания этого сообщения
fields		список полей формы
label	string(100)	подпись поля
name	string(100)	название поля
value	string(100)	значение поля (максимальная длина поля 100 символов)
listVals	string(100)	одно из значений для выбора из списка (элементов listVals может быть несколько)
type	string	тип поля, одно из значение HIDDEN – скрытое DISABLE – недоступное к редактированию LIST – поле типа список TEXT – текстовое поле NUMBER – поле возможностью вводить число (включая дробные) MONEY - поле с возможностью вводить число с двумя знаками после запятой

optional	boolean	Является ли поле не обязательным к заполнению true – не обязательно false – поле обязательно к заполнению
----------	---------	---

Значение переменной «payType» в рамках одного сеанса оплаты изменяться не может, значения переданные в указанном поле могут использоваться, для разделения отчетности по типам платежей, по согласованию с Банком.

8. Сторнирование платежа

Для сторнирования платежа необходимо воспользоваться функцией stornoPay. В форме должны присутствовать поля session_id, pausum и unpr.

Функции необходимо передать массив данных:

Переменная	Формат	Описание
messageDate	dateTime	дата создания этого сообщения
fields		список полей формы
Name	string(100)	название поля
Value	string(100)	значение поля

Результатом работы функции будет также массив данных:

Переменная	Формат	Описание
messageDate	dateTime	дата создания этого сообщения
status	string (20)	статус (OK, ERROR, DENY) OK – платеж успешно сторнирован DENY – сторнирование запрещено, платеж из закрытого дня или сторнирование запрещено поставщиком услуги ERROR – при сторнировании произошла ошибка

9. Получение списка дополнительных услуг

Список дополнительных услуг возвращается при вызове метода getAdditionalServices.

Входящие параметры:

Переменная	Формат	Описание
messageDate	dateTime	дата создания этого сообщения

Результатом работы функции будет также массив данных:

Переменная	Формат	Описание
messageDate	dateTime	дата создания этого сообщения
serviceList		список услуг
name	string (100)	наименование услуги

10. Корзина платежей

Для сохранения нескольких платежей в корзину, после завершения шага 5 необходимо начать сохранения платежа заново (Шаг 1) используя значение session_id полученное ранее, при каждом последующем выполнении шага 5 будет возвращаться новый идентификатор корзины, поля pausum и commission будут содержать сумму всех платежей добавленных в рамках этой сессии.

Получить текущее содержимое корзины и чек можно с помощью метода getCart, для удаления платежа из корзины payDelete

Получение текущего состояния корзины платежей getCart

Входящие параметры:

Переменная	Формат	Описание
messageDate	dateTime	дата создания этого сообщения
sessionId	string(40)	Идентификатор сессии

Результатом работы функции:

Переменная	Формат	Описание
messageDate	dateTime	дата создания этого сообщения
unpr	string(20)	Уникальный номер корзины
status	string(20)	Текущий статус

paysum	string (100)	общая сумма к оплате
commission	string (100)	общая сумма комиссии
chequeList	string(100)	Список строк содержащий чек
cartList		список услуг корзины
payId	string (20)	идентификатор платежа в корзине
paysum	string (100)	сумма платежа включая комиссию
commission	string (100)	сумма комиссии

Удаление платежа из корзины payDelete

Входящие параметры:

Переменная	Формат	Описание
messageDate	dateTime	дата создания этого сообщения
payId	string(20)	идентификатор платежа в корзине

Результатом работы функции:

Переменная	Формат	Описание
unp	string(20)	Уникальный номер корзины
status	string(20)	Текущий статус
paysum	string (100)	общая сумма к оплате
commission	string (100)	общая сумма комиссии
chequeList	string(100)	Список строк содержащий чек

11. Получение статуса платежа

Для получения текущего состояния платежа можно воспользоваться функцией getStatus.

Входящие параметры:

Переменная	Формат	Описание
unp	string(20)	Номер платежа полученный на шаге 5

Результатом работы функции:

Переменная	Формат	Описание
status	string(20)	Текущий статус платежа
authDate	dateTime	Дата проводки платежа (успешного выполнения шага 6)

Возможные значения статусов

NOT_FOUND	Платеж не найден
SAVED	Платеж сохранен, ожидает проводки
DELETED	Платеж удален, проводка платежа невозможна
AUTH	Платеж проведен
STORNED	Платеж сторнирован

Диаграмма изменений статусов



Интерфейс получения отчетов

Взаимодействие осуществляется посредством REST API

1. Получение OAuth токена

Для доступа к сервису необходимо пройти процесс авторизации. Получить токен можно выполнив POST запрос по адресу

```
http://<адрес сервера>/<приложение>/rest/v2/oauth/token
```

Значение заголовка **Content-Type** должно быть **application/x-www-form-urlencoded**, кодировка **UTF-8**.

Запрос должен содержать следующие параметры:

- grant_type - всегда значение password.
- username - логин пользователя приложения.
- password - пароль пользователя приложения.

Метод возвращает JSON объект:

```
{
  "access_token": "29bc6b45-83cd-4050-8c7a-2a8a60adf251",
  "token_type": "bearer",
  "refresh_token": "e765446f-d49e-4634-a6d3-2d0583a0e7ea",
  "expires_in": 43198,
  "scope": "rest-api"
}
```

Значение токена содержится в поле access_token

Чтобы использовать токен, его нужно передать в заголовке запроса **Authorization** с типом **Bearer**, например:

```
Authorization: Bearer 29bc6b45-83cd-4050-8c7a-2a8a60adf251
```

2. Выполнение отчета

Последовательность действия для выполнения отчета:

- Получение списка отчётов.
- Получение информации об отдельном отчёте.
- Запуск отчёта и скачивание результата.

2.1 Получение списка отчётов

Чтобы получить список всех существующих отчётов, выполните GET запрос по адресу:

```
http://<адрес сервера>/<приложение>/rest/reports/v1/report
```

Ответ будет содержать краткую информацию обо всех доступных отчётах

```
{
  "id": "2dd27fbf-8830-416a-899f-339543f8f27a",
  "name": "АКБРАУ_РЕП_1"
}
```

2.2 Получение информации об отчёте

Для получения детальной информации об отчёте выполните GET запрос по адресу:

```
http://<адрес сервера>/<приложение>/rest/reports/v1/report/{id}
```

Последняя часть запроса здесь – это идентификатор нужного отчёта.

Возвращаемый объект JSON будет содержать следующую информацию об отчёте:

- ID
- имя отчёта
- код отчёта (если задан)
- параметры и их типы

2.3 Запуск отчёта

Чтобы выполнить отчёт, выполните POST запрос по адресу:

```
http://<адрес сервера>/<приложение>/rest/reports/v1/run/{id}
```


Последняя часть запроса здесь – это идентификатор нужного отчёта, например:

```
POST http://<адрес сервера>/<приложение>/rest/reports/v1/run/2dd27fbf-8830-416a-899f-339543f8f27a HTTP/1.1
```

Значение параметров необходимо передать в теле запроса:

```
{ "parameters":  
  [  
    { "name": "agentId", "value": 1 },  
    { "name": "dbeg", "value": "2020-10-03 00:00:00.000" },  
    { "name": "dend", "value": "2020-10-03 23:59:59.999" }  
  ]  
}
```

Если в параметрах отчета присутствует дата, то она должна быть в формате "уууу-ММ-дд НН:мм:сс.SSS"

Пример запуска отчета с помощью утилиты curl:

```
curl -H "Content-Type: application/json" -H "Authorization: Bearer 23305557-2f54-4ffa-927d-a9e125e0d913" -d "{parameters: [{name:'organizationId3', value:1}, {name:'dbeg', value:'2020-10-03 00:00:00.000'}, {name:'dend', value:'2020-10-03 00:00:00.000'}]}" -X POST "http://<адрес сервера>/<приложение>/rest/reports/v1/run/6a8634e9-677d-2ff8-e7f5-d7e9ead4cb8e" >1.txt
```

3. Загрузка платежных поручений через REST API

Для загрузки платежных поручений необходимо выполнить **POST** запрос по адресу:

```
http://<адрес сервера>/<приложение>/rest/uploadPp/sendPp
```

Содержимое файла необходимо передать в параметра **file**. В ответ на запрос вернется идентификатор файла.

Формат файла описан в приложении 3.

Пример передачи файла с помощью утилиты curl:

```
CURL -H "Authorization: Bearer 29bc6b45-83cd-4050-8c7a-2a8a60adf251" -F file=@plat.txt http://<адрес сервера>/<приложение>/rest/uploadPp/sendPp
```

Для отслеживания статуса файла необходимо выполнить **GET** запрос и передать в параметрах идентификатор файла (fileId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx):

```
/rest/uploadPp/getFilePpInfo
```

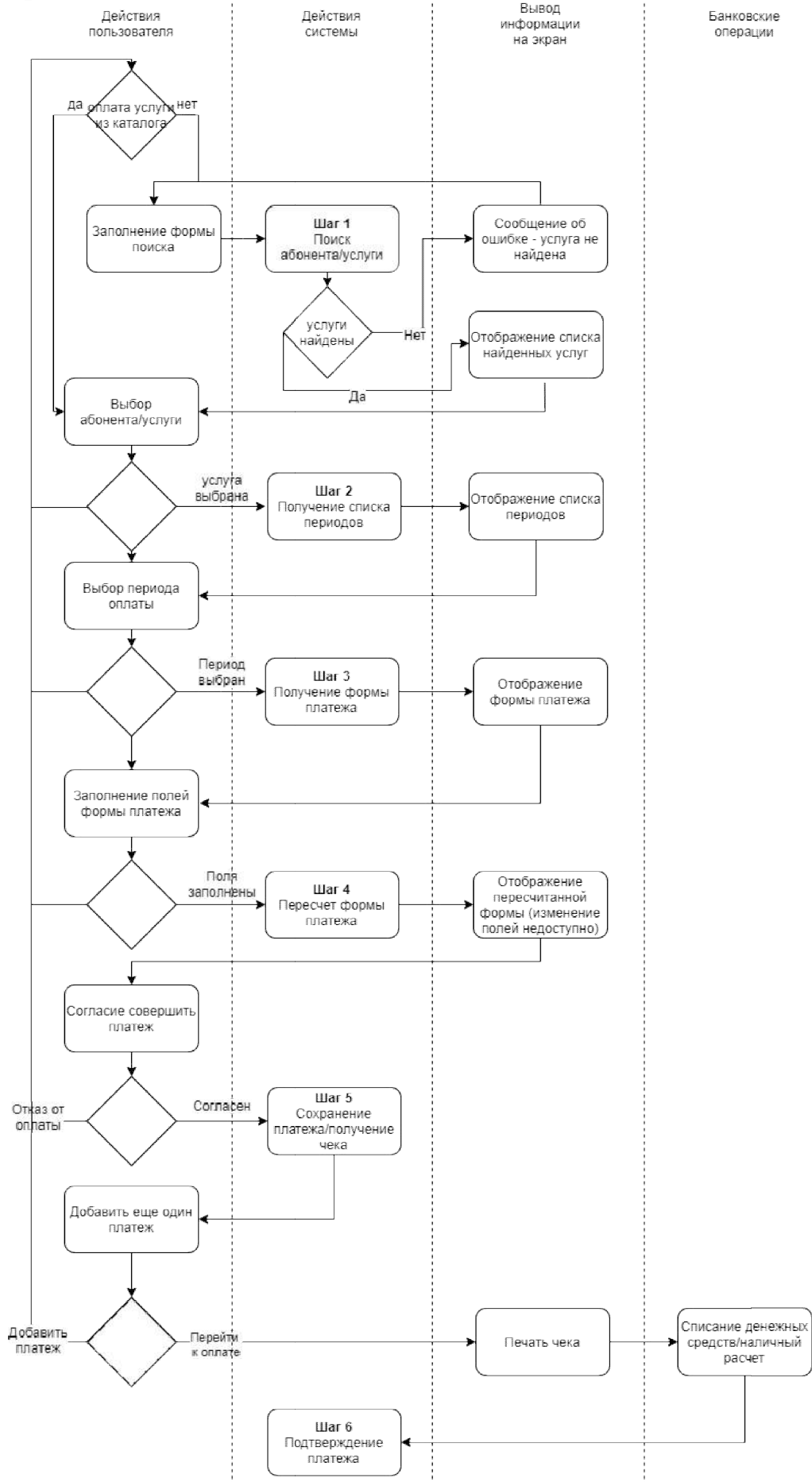
Пример отслеживания статуса файла с помощью утилиты curl:

```
CURL -H "Authorization: Bearer e35929ba-ca2b-46cb-a056-b93d778ecda5" http://<адрес сервера>/<приложение>/rest/uploadPp/getFilePpInfo?fileId=3a098260-6ae3-1d13-e012-b6334cd29b3b
```

В ответ вернется строка в формате JSON:

```
{"Файлы": [{"Сообщение об ошибке": "2 записей не загружено", "Название": "plat.txt", "Статус": "Загружен с ошибками"}]}
```

Приложение №1 Схема взаимодействия



Приложение №2 Примеры запросов

Поиск по лицевому счету	
<pre><env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"> <env:Header> <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" env:mustUnderstand="true"> <wsse:UsernameToken wsu:Id="UsernameToken-55B8B2E522D1579F1214030733348831"> <wsse:Username>FFFFFFF-6896-4DC1-ABCF-AD7C56D1A713</wsse:Username> <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">zaqwsxcderfv</wsse:Password> </wsse:UsernameToken> </wsse:Security> </env:Header> <env:Body> <ns3:formCalc xmlns:ns3="http://pay.sb.sg/" xmlns=""> <messageDate>2020-06-29T12:00:16.747+07:00</messageDate> <payType>CASH</payType> <payPoint>998</payPoint> <payChannel>KASSA</payChannel> <fields> <name>find_account_val</name> <value>999951906</value> </fields> </ns3:formCalc> </env:Body> </env:Envelope></pre>	<pre><soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"> <soap:Body> <ns2:formCalcResponse xmlns:ns2="http://pay.sb.sg/"> <messageDate>2020-06-29T12:05:29.119+07:00</messageDate> <fields> <name>session_id</name> <value>8758127068878908</value> <type>HIDDEN</type> <optional>true</optional> </fields> <fields> <name>select_service_val</name> <value>-</value> <type>LIST</type> <optional>true</optional> <listVals>-</listVals> <listVals>ЕДИНЬЙ ПЛАТЕЖЬЙ ДОКУМЕНТ ЖКУ (1-999951906)</listVals> <listVals>ПЕНЯ (3-999951906)</listVals> </fields> </ns2:formCalcResponse> </soap:Body> </soap:Envelope></pre>
<p>Поиск по адресу</p> <pre><env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"> <env:Header> <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" env:mustUnderstand="true"> <wsse:UsernameToken wsu:Id="UsernameToken-55B8B2E522D1579F1214030733348831"> <wsse:Username>FFFFFFF-6896-4DC1-ABCF-AD7C56D1A713</wsse:Username> <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">xlkco2409ancb</wsse:Password> </wsse:UsernameToken> </wsse:Security> </env:Header> <env:Body></pre>	<pre><soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"> <soap:Body> <ns2:formCalcResponse xmlns:ns2="http://pay.sb.sg/"> <messageDate>2020-08-07T17:51:46.040+07:00</messageDate> <fields> <name>session_id</name> <value>7609191846952637</value> <type>HIDDEN</type> <optional>true</optional> </fields> <fields> <name>select_service_val</name> <value>-</value> <type>LIST</type> <optional>true</optional> <listVals>-</listVals> <listVals>ТСТ ЭЛ. ЭНЕРГИЯ - БГЭС (1201-75280100)</listVals> </fields> </ns2:formCalcResponse> </soap:Body> </soap:Envelope></pre>

<pre> <ns3:formCalc xmlns:ns3="http://pay.sb.sg/" xmlns=""> <messageDate>2020-06-29T12:00:16.747+07:00</messageDate> <payType>CASH</payType> <payPoint>998</payPoint> <payChannel>KASSA</payChannel> <fields> <name>find_address_city_val</name> <value>Барнаул</value> </fields> <fields> <name>find_address_street_val</name> <value>Ленина</value> </fields> <fields> <name>find_address_house_val</name> <value>54</value> </fields> <fields> <name>find_address_apartment_val</name> <value>10</value> </fields> </ns3:formCalc> </env:Body> </env:Envelope> </pre>	<pre> <listVals>ЕДИНЫЙ ПЛАТЕЖНЫЙ ДОКУМЕНТ ЖКУ (1-999951906)</listVals> </fields> </ns2:formCalcResponse> </soap:Body> </soap:Envelope> </pre>
--	---

Получение календаря

<pre> <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"> <env:Header> <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" env:mustUnderstand="true"> <wsse:UsernameToken wsu:Id="UsernameToken-55B8B2E522D1579F1214030733348831"> <wsse:Username>FFFFFFF-6896-4DC1-ABCF-AD7C56D1A713</wsse:Username> <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">zaqwsxcderfv</wsse:Password> </wsse:UsernameToken> </wsse:Security> </env:Header> <env:Body> <ns3:formCalc xmlns:ns3="http://pay.sb.sg/" xmlns=""> <messageDate>2020-06-29T12:00:16.747+07:00</messageDate> <fields> <name>session_id</name> <value>8758127068878908</value> </fields> <fields> <name>select_service_val</name> <value>ЕДИНЫЙ ПЛАТЕЖНЫЙ ДОКУМЕНТ ЖКУ (1-999951906)</value> </fields> </ns3:formCalc> </env:Body> </env:Envelope> </pre>	<pre> <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"> <soap:Body> <ns2:formCalcResponse xmlns:ns2="http://pay.sb.sg/"> <messageDate>2020-06-29T12:08:12.617+07:00</messageDate> <fields> <name>address</name> <value>БАРНАУЛ, ЛЕНИНА ПР-КТ, 54, 10</value> <type>text</type> <optional>true</optional> </fields> <fields> <name>calendar</name> <value>-</value> <type>LIST</type> <optional>true</optional> <listVals>-</listVals> <listVals>Июнь 2020</listVals> <listVals>Май 2020</listVals> </fields> </ns2:formCalcResponse> </soap:Body> </soap:Envelope> </pre>
---	--

<pre> </fields> </ns3:formCalc> </env:Body> </env:Envelope> </pre>	
--	--

Получение календаря по дополнительной услуге

<pre> <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"> <env:Header> <wsse:Security xmlns:wsse="http://docs.oasis- open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity- utility-1.0.xsd" env:mustUnderstand="true"> <wsse:UsernameToken wsu:Id="UsernameToken- 55B8B2E522D1579F1214030733348831"> <wsse:Username>FFFFFFF-6896-4DC1-ABCF- AD7C56D1A713</wsse:Username> <wsse:Password Type="http://docs.oasis- open.org/wss/2004/01/oasis-200401-wss-username-token-profile- 1.0#PasswordText">xlkco2409anch</wsse:Password> </wsse:UsernameToken> </wsse:Security> </env:Header> <env:Body> <ns3:formCalc xmlns:ns3="http://pay.sb.sg/" xmlns=""> <messageDate>2020-06-29T12:00:16.747+07:00</messageDate> <fields> <name>select_service_val</name> <value>Школьное питание - КШП ИП Савина Е.С. (Аксиома) (2999)</value> </fields> </ns3:formCalc> </env:Body> </env:Envelope> </pre>	<pre> <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"> <soap:Body> <ns2:formCalcResponse xmlns:ns2="http://pay.sb.sg/"> <messageDate>2020-06-29T12:08:12.617+07:00</messageDate> <fields> <name>calendar</name> <value>-</value> <type>LIST</type> <optional>true</optional> <listVals>-</listVals> </fields> </ns2:formCalcResponse> </soap:Body> </soap:Envelope> </pre>
--	--

Получение форму оплаты услуги

<pre> <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"> <env:Header> <wsse:Security xmlns:wsse="http://docs.oasis- open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity- utility-1.0.xsd" env:mustUnderstand="true"> <wsse:UsernameToken wsu:Id="UsernameToken- 55B8B2E522D1579F1214030733348831"> <wsse:Username>FFFFFFF-6896-4DC1-ABCF- AD7C56D1A713</wsse:Username> <wsse:Password Type="http://docs.oasis- open.org/wss/2004/01/oasis-200401-wss-username-token-profile- 1.0#PasswordText">zaqwsxcderfv</wsse:Password> </wsse:UsernameToken> </wsse:Security> </env:Header> <env:Body> <ns3:formCalc xmlns:ns3="http://pay.sb.sg/" xmlns=""> <messageDate>2020-06-29T12:00:16.747+07:00</messageDate> </pre>	<pre> <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"> <soap:Body> <ns2:formCalcResponse xmlns:ns2="http://pay.sb.sg/"> <messageDate>2020-06-29T12:09:34.206+07:00</messageDate> <fields> <name>f_mon_1</name> <value>556.58</value> <label>Содержание помещения - ТСЖ "Проспект Ленина-54"</label> <type>MONEY</type> <optional>>false</optional> </fields> <fields> <name>f_mon_2</name> <value>417.6</value> <label>Газоснабжение - ОАО "Алтайкрайга</label> <type>MONEY</type> <optional>>false</optional> </fields> </ns2:formCalcResponse> </soap:Body> </soap:Envelope> </pre>
--	---

<pre> <fields> <name>session_id</name> <value>8758127068878908</value> </fields> <fields> <name>calendar</name> <value>Июнь 2020</value> </fields> </ns3:formCalc> </env:Body> </env:Envelope> </pre>	<pre> <name>f_mon_3</name> <value>45</value> <label>Домофон - 000 "Домофон-Центр"</label> <type>MONEY</type> <optional>>false</optional> </fields> <fields> <name>f_mon_4</name> <value>0.00</value> <label>Электрoэнергия СОИ - АО "Барнаулская горэлектросе</label> <type>MONEY</type> <optional>>false</optional> </fields> </ns2:formCalcResponse> </soap:Body> </soap:Envelope> </pre>
---	---

Пересчет формы

<pre> <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"> <env:Header> <wsse:Security xmlns:wsse="http://docs.oasis- open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity- utility-1.0.xsd" env:mustUnderstand="true"> <wsse:UsernameToken wsu:Id="UsernameToken- 55B8B2E522D1579F1214030733348831"> <wsse:Username>FFFFFFF-6896-4DC1-ABCF- AD7C56D1A713</wsse:Username> <wsse:Password Type="http://docs.oasis- open.org/wss/2004/01/oasis-200401-wss-username-token-profile- 1.0#PasswordText">zaqwsxcderfv</wsse:Password> </wsse:UsernameToken> </wsse:Security> </env:Header> <env:Body> <ns3:formCalc xmlns:ns3="http://pay.sb.sg/" xmlns=""> <messageDate>2020-06-29T12:00:16.747+07:00</messageDate> <fields> <name>session_id</name> <value>8758127068878908</value> </fields> <fields> <name>f_mon_1</name> <value>334.56</value> </fields> <fields> <name>f_mon_2</name> <value>12.56</value> </fields> <fields> <name>f_mon_3</name> <value>45</value> </pre>	<pre> <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"> <soap:Body> <ns2:formCalcResponse xmlns:ns2="http://pay.sb.sg/"> <messageDate>2020-06-29T12:12:49.766+07:00</messageDate> <fields> <name>paysum</name> <value>515.57</value> <type>DISABLE</type> <optional>>false</optional> </fields> <fields> <name>commission</name> <value>0</value> <type>DISABLE</type> <optional>>false</optional> </fields> <fields> <name>need_identification</name> <value>0</value> <type>DISABLE</type> <optional>>false</optional> </fields> </ns2:formCalcResponse> </soap:Body> </soap:Envelope> </pre>
--	---

<pre> </fields> <fields> <name>f_mon_4</name> <value>123.45</value> </fields> </ns3:formCalc> </env:Body> </env:Envelope> </pre>	
--	--

Сохранение платежа, получение чека

<pre> <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"> <env:Header> <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" env:mustUnderstand="true"> <wsse:UsernameToken wsu:Id="UsernameToken-55B8B2E522D1579F1214030733348831"> <wsse:Username>FFFFFFF-6896-4DC1-ABCF-AD7C56D1A713</wsse:Username> <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">zaqwsxcderfv</wsse:Password> </wsse:UsernameToken> </wsse:Security> </env:Header> <env:Body> <ns3:formCalc xmlns:ns3="http://pay.sb.sg/" xmlns=""> <messageDate>2020-06-29T12:00:16.747+07:00</messageDate> <fields> <name>session_id</name> <value>8758127068878908</value> </fields> </ns3:formCalc> </env:Body> </env:Envelope> </pre>	<pre> <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"> <soap:Body> <ns2:formCalcResponse xmlns:ns2="http://pay.sb.sg/"> <messageDate>2020-06-29T12:16:15.313+07:00</messageDate> <fields> <name>unp</name> <value>8688199648</value> <type>DISABLE</type> <optional>>false</optional> </fields> <fields> <name>paysum</name> <value>515.57</value> <type>DISABLE</type> <optional>>false</optional> </fields> <fields> <name>f_str_1</name> <value></value> <type>DISABLE</type> <optional>>true</optional> </fields> <fields> <name>f_str_2</name> <value>ЧЕК-ОРДЕР ИЗВЕЩЕНИЕ</value> <type>DISABLE</type> <optional>>true</optional> </fields> <fields> <name>f_str_3</name> <value>Ф-Л № 5440 Ф-Л № 5440</value> <type>DISABLE</type> <optional>>true</optional> </fields> <fields> <name>f_str_72</name> <value>Платеж проведен с использованием Платеж проведен с использованием</value> <type>DISABLE</type> <optional>>true</optional> </fields> </pre>
---	--

	<pre> <fields> <name>f_str_73</name> <value>Системы "Город" Системы "Город"</value> <type>DISABLE</type> <optional>true</optional> </fields> <fields> <name>f_str_74</name> <value>www.sistemagorod.ru www.sistemagorod.ru</value> <type>DISABLE</type> <optional>true</optional> </fields> <fields> <name>f_str_75</name> <value></value> <type>DISABLE</type> <optional>true</optional> </fields> <fields> <name>f_str_76</name> <value></value> <type>DISABLE</type> <optional>true</optional> </fields> </ns2:formCalcResponse> </soap:Body> </soap:Envelope> </pre>
--	--

<p>Проводка платежа</p> <pre> <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"> <env:Header> <wsse:Security xmlns:wsse="http://docs.oasis- open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity- utility-1.0.xsd" env:mustUnderstand="true"> <wsse:UsernameToken wsu:Id="UsernameToken- 55B8B2E522D1579F1214030733348831"> <wsse:Username>FFFFFFF-6896-4DC1-ABCF- AD7C56D1A713</wsse:Username> <wsse:Password Type="http://docs.oasis- open.org/wss/2004/01/oasis-200401-wss-username-token-profile- 1.0#PasswordText">zaqwsxcderfv</wsse:Password> </wsse:UsernameToken> </wsse:Security> </env:Header> <env:Body> <ns3:invoicePay xmlns:ns3="http://pay.sb.sg/" xmlns=""> <messageDate>2020-06-29T12:00:16.747+07:00</messageDate> <fields> <name>session_id</name> </pre>	<pre> <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"> <soap:Body> <ns2:invoicePayResponse xmlns:ns2="http://pay.sb.sg/"> <messageDate>2020-06-29T12:25:22.796+07:00</messageDate> <status>OK</status> </ns2:invoicePayResponse> </soap:Body> </soap:Envelope> </pre>
--	---


```
<value>8758127068878908</value>
</fields>
<fields>
  <name>unp</name>
  <value>8688199648</value>
</fields>
<fields>
  <name>paysum</name>
  <value>515.57</value>
</fields>
</ns3:invoicePay>
</env:Body>
</env:Envelope>
```

Приложение №3 Формат файл загрузки платежных поручения

Файл для загрузки информации о платежных поручениях формируется в текстовом формате в кодировке Windows-1251. Каждая строка содержит одно платежное поручение, поля в строке разделяются символом «;».

Перечень полей:

Номер поля по порядку	Описание поля	Значение поля	Макс. размер поля
1	Код платежного поручения	Уникальный код сформированного платежного поручения в Системе Город. Поле из отчета для формирования платежных поручений	255
2	Номер документа	Номер платежного поручения	10
3	Дата документа	Дата платежного поручения в формате dd/mm/yyyy	10
4	БИК банка плательщика	БИК банка плательщика	12
5	Корр.счет банка плательщика	Корр.счет банка плательщика	20
6	Расчетный счет плательщика	Расчетный счет плательщика	20
7	БИК банка получателя	БИК банка получателя	12
8	Корр.счет банка получателя	Корр.счет банка получателя	20
9	Расчетный счет получателя	Расчетный счет получателя	20
10	Сумма документа	Сумма документа в копейках	10
11	Наименование плательщика	Наименование плательщика	70
12	ИНН плательщика	ИНН плательщика	15
13	Банк плательщика	Банк плательщика	70
14	Наименование получателя	Наименование получателя	70
15	ИНН получателя	ИНН получателя	15
16	Банк получателя	Банк получателя	70
17	Назначение платежа	Назначение платежа	2000
18	КПП получателя	КПП получателя	9
19	КБК	КБК	20
20	ОКАТО	ОКАТО	10

Список изменений

Версия	Дата	Изменение
1.1	18.09.2017	Изменена схема взаимодействия (приложения №1)
1.2	15.05.2018	На шаге 4 добавлен признак необходимости идентификации плательщика
1.3	23.09.2019	Добавлено Приложение №2 Примеры запросов
1.4	13.03.2020	Добавлена функция получения статуса платежа
2.0	11.04.2020	Добавлен интерфейс получения отчетов,
2.1	29.06.2020	Актуализированы примеры
3.0	06.08.2020	Добавлена корзина платежей
3.1	06.08.2020	Исправлена схема взаимодействия
3.2	07.08.2020	Добавлены примеры
3.3	16.07.2021	Исправлен список параметров функции getCart
3.4	09.11.2021	Добавлено поле authDate, в ответ на метод getStatus Отслеживание статуса загруженного списка платежных поручений