



Шлюз программного комплекса OpenShow

Оглавление

Оглавление	1
SQL - плагин.....	3
Настройки SQL-плагина	3
Создание группы.....	3
Тип и текст запроса.	5
Параметры	5
Плагин «Шлюз DataSync».....	6
Настройки плагина «Шлюз DataSync»	6
Работа с плагином.....	6
Плагин «Шлюз интеграции с бэк-офисом».....	13
Работа с плагином.....	13
Плагин «WEB-отчеты».....	18
Настройки плагина «WEB-отчеты»	18
Работа с плагином.....	20
Плагин лояльности.	24
Технология работы с плагином.	24
Построение системы лояльности.	24
Построение интернет-магазина.	25
Работа с плагином лояльности	25
Описание общих параметров.....	25
Методы плагина.	26
Плагин интеграции с процессингом.	37
Работа с плагином интеграции с процессингом	37
Описание общих параметров.....	37
Методы плагина.	37

Назначение. Краткое описание.

Шлюз OpenShow предназначен для обеспечения максимально гибкой интеграции решений OpenShow с внешними системами.

Шлюз представляет собой HOST-сервис, управляемый с «Консоли управления» OpenShow, под управлением которого работает произвольное кол-во плагинов, каждый из которых является функционально независимым. В качестве внешнего канала связи каждый плагин может использовать как встроенный WEB-шлюз, так и любой произвольный канал (соединение TCP/IP, файловый обмен и т.д.). Стандартная поставка содержит ряд плагинов, предлагаемых разработчиком, однако допускается и создание пользовательских плагинов самостоятельно или с привлечением третьей стороны.

При необходимости масштабирования нагрузки возможен запуск нескольких экземпляров шлюза на разных физических серверах.

SQL - плагин

SQL - плагин предназначен для решения произвольных пользовательских задач по средствам WEB-шлюза.

Настройки SQL-плагина

Настройки плагина открываются по двойному щелчку мышки на самом плагине. Откроется окно настроек плагина «Настройки плагина» (Рис. 1).

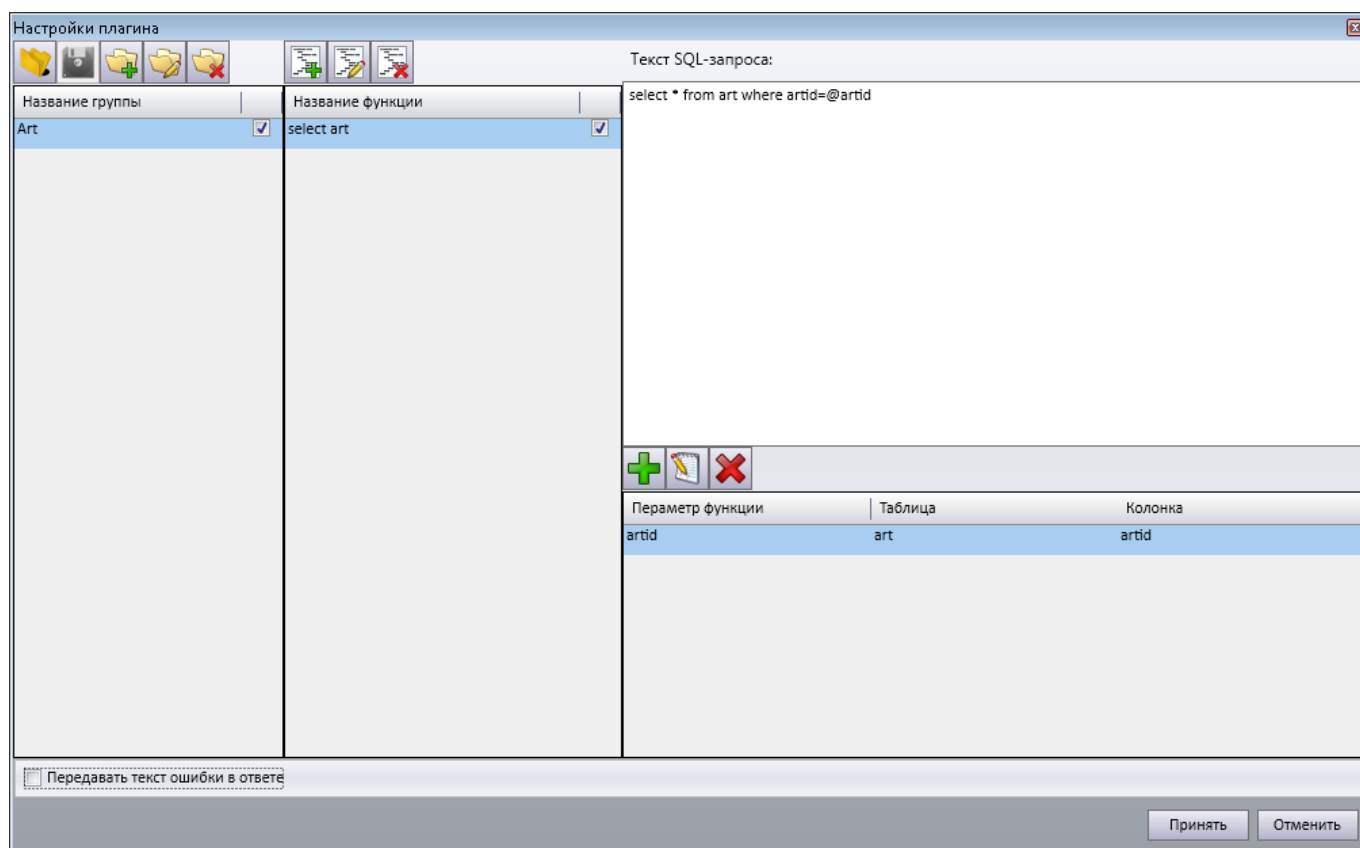


Рис. 1. Настройки плагина Sql сервиса WebGate

Окно настроек плагина Sql подразумевает заполнение ряда настроек и разделено на четыре области. В левой части окна отображается область настроек групп плагина. По центру отображается область настроек «Название функции», где формируется список функций, для выбранной группы. Справа отображается область настроек параметров каждой группы «Параметр», и вводится текст запроса.

Создание группы.

Область настроек «Группы» предназначена для добавления и редактирования информации о группах подключений плагина Sql. Информация о группах плагина представлена в виде таблицы: название группы и признак активности

Чтобы добавить новую группу в список групп нажмите кнопку «Добавить группу», расположенную в верхней области настроек. Откроется окно настроек параметров группы «Параметры группы». В данном окне следует заполнить следующие поля

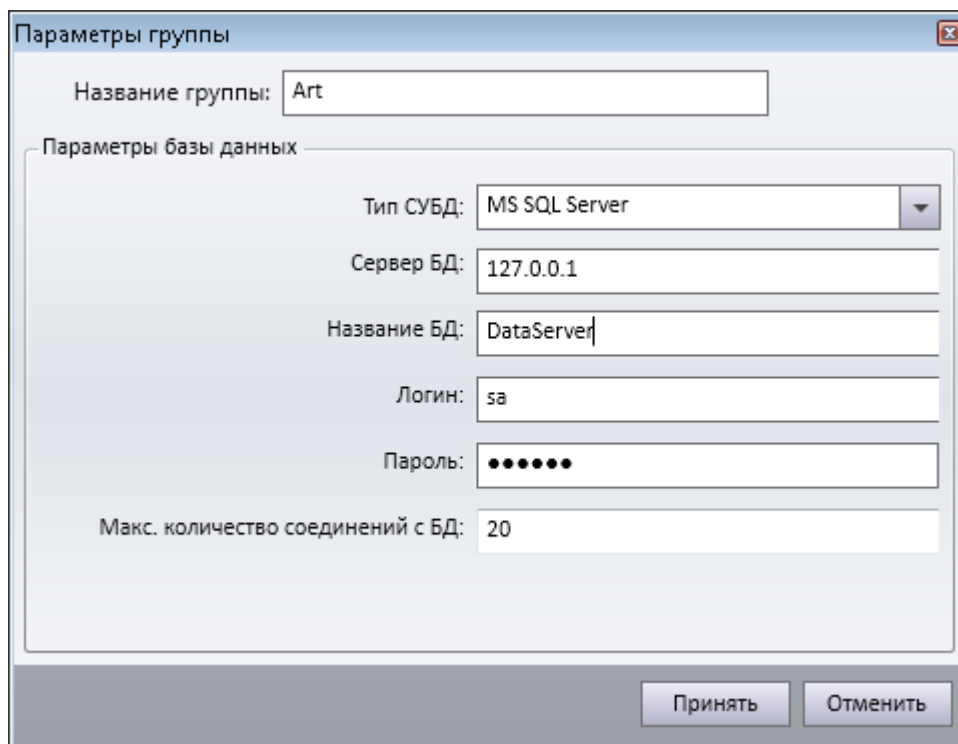


Рис. 2. Окно создания подключения

В поле «Название группы» укажите название группы функций

В поле «Тип СУБД» выберите тип базы данных, с которой работает данное подключение (пользователь). Настройки соединения с БД зависят от типа базы данных. В случае если в качестве БД используется SQL Server или Firebird необходимо задать следующие параметры: IP-адрес или наименование сервера баз данных, наименование БД «Сервера данных». Если используется Oracle необходимо задать наименование БД. Если в качестве БД используется IBM DB2 в поле «Название БД» необходимо указать путь к БД. Для БД (MS SQL Server, Oracle и Firebird) необходимо ввести логин и пароль соединения (пользователя), который подключается к БД.

В поле «Макс. количество соединений с БД» следует указать максимально допустимое количество соединений с базой данных. По умолчанию данное значение равно 20.

Создание новой функции группы

Область (или группа) настроек «Функции» предназначена для добавления и редактирования информации о функциях групп плагина SqlPlugin. Каждой группе привязывается ряд функций. Информация о функциях группы плагина представлена в виде списка функций.

Чтобы добавить новую функцию группы нажмите кнопку «Добавить функцию», расположенную в верхней области настроек. Откроется окно настроек функции «Параметры функции».

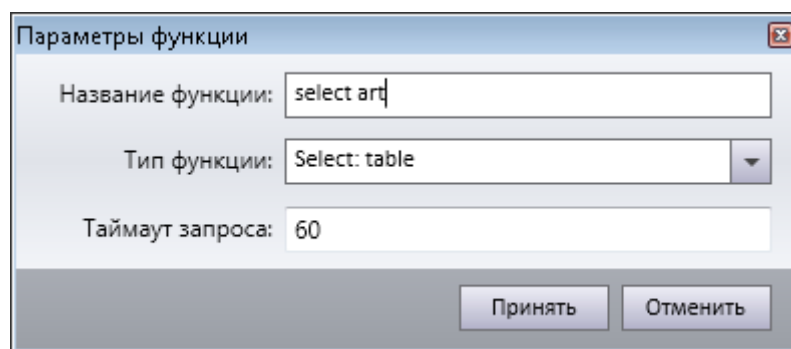


Рис. 3. Окно «Редактирование группы функций»

В поле «Название функции» укажите название функции, которое будет привязываться к группе подключений.

В поле «Тип функции» в зависимости от получения необходимой информации, из раскрывающегося списка выберите тип функции: «Select: table», «Select: scalar» или «Delete/Insert/Update».

Если необходимо получить всю информацию запроса. Выберите тип запроса «Select: table», а в поле «Текст запроса» введите текст запроса. Будет выведена таблица с необходимыми данными.

Если необходимо получить определенные данные какой-то таблицы запроса. Выберите тип запроса «Select: scalar», и в поле «Текст запроса» введите соответственно текст.

Если необходимо удалить, редактировать или обновить какие-то данные, то в этом случае выберите тип запроса «Delete/Insert/Update», и в поле «текст запроса» введите текст запроса.

Тип и текст запроса.

В области «Текст запроса» окна настроек плагина SqlPlugin отображается текст запроса, который вводится пользователем при создании запроса.

Параметры

Область настроек «Параметр» предназначена для добавления и редактирования информации о параметрах функции группы в окне настроек плагина SqlPlugin. О каждом параметре функции группы выводятся такие данные: название параметра, таблица параметра и поле параметра функции группы.

Создание нового параметра

Чтобы добавить новый параметр функции группы нажмите кнопку «Добавить параметр», расположенную в верхней области настроек «Параметр». Откроется окно настроек параметра функции группы «Параметр функции»:

- «Параметр функции» - в этом поле указывается название параметра функции группы;
- «Таблица» - в этом поле укажите название таблицы параметра функции группы;
- «Колонка» - в этом поле укажите название поля соответствующего параметра.

Плагин «Шлюз DataSync».

Плагин позволяет работать по DataSync-образному протоколу.

Настройки плагина «Шлюз DataSync»

Настройки плагина открываются по двойному щелчку мышки на самом плагине. Откроется окно настроек плагина «Настройки плагина».

В данном окне следует заполнить следующие поля:

- «Тип СУБД» - из раскрывающегося списка выберите тип базы данных, с которой работает данный плагин.
- «Сервер БД» - указывается IP-адрес базы данных к которой необходимо подключиться.
- В поле «Название БД» необходимо указать путь к БД. Для БД (MS SQL Server, Oracle и Firebird) необходимо ввести логин и пароль соединения (пользователя), который подключается к БД.
- Поля «Логин» и «Пароль» - необходимо ввести логин и пароль соответственно.
- «Максимальный размер пакета» - указывается максимальный размер пакета данных.
- «Время жизни сессии клиента», с – следует указать время жизни сессии клиента в секундах.

Работа с плагином

Работа плагином производится при помощи запросов к его методам через протокол HTTP. Все методы – GET. Запрос формируется в следующем виде:

[http://<Сервер>:<порт>/идентификатор/<метод>? \[&Параметры\]](http://<Сервер>:<порт>/идентификатор/<метод>? [&Параметры])

Здесь:

- <Сервер> - IP-адрес сервера,
- <порт> - порт, по которому работает шлюз (по умолчанию 8080),
- идентификатор – строковая константа (GUID), указывающая на конкретный плагин (шлюз поддерживает произвольное кол-во плагинов, каждый из которых должен иметь уникальный идентификатор). Для настоящего плагина идентификатор имеет вид:
04E02234-A87F-44FD-B7E7-3BA4756B8A53.

Методы плагина.

LoadDataSync – Предназначен для получения списка таблиц, которые можно получить при помощи шлюза и актуальных версий по каждой из этих таблиц (о версиях ниже).

Параметры запроса:

- *SystemId* - обязательный, строка, уникальный идентификатор системы, для которой выполняются запросы (если у вас работает несколько клиентов шлюза, то они должны иметь разные значения этого параметра).
- *CallID* - обязательный, строка, уникальный номер запроса.

Параметры ответа:

- *ErrorInfo* - код ошибки, 0 – если ошибок нет.
- *DirName* - имя таблицы,
- *UpdateNum* - текущая версия данных таблицы,
- *SyncPriority* - приоритет импорта таблицы, учитывающий целостность БД (следует принимать таблицы в порядке увеличения приоритета).

Пример запроса:

<http://127.0.0.1:8080/04E02234-A87F-44FD-B7E7-3BA4756B8A53/LoadDataSync?SystemID=qw123&CallID=12>

Пример ответа (фрагмент):

```
<?xml version="1.0" encoding="utf-8" ?>
- <Root>
- <ErrorInfo>
    <ErrorCode>0</ErrorCode>
</ErrorInfo>
- <CommandResults>
    - <DATASYNCDIRS>
        <DIRNAME>ACCOUNTTYPE</DIRNAME>
        <UPDATENUM>1</UPDATENUM>
        <SYNCPRIORITY>0</SYNCPRIORITY>
    </DATASYNCDIRS>
    - <DATASYNCDIRS>
        <DIRNAME>CLNTFORM</DIRNAME>
        <UPDATENUM>1</UPDATENUM>
        <SYNCPRIORITY>0</SYNCPRIORITY>
    </DATASYNCDIRS>
    ...
    - <DATASYNCDIRS>
        <DIRNAME>BACKACCOUNTTRANS</DIRNAME>
        <UPDATENUM>0</UPDATENUM>
        <SYNCPRIORITY>30</SYNCPRIORITY>
    </DATASYNCDIRS>
</CommandResults>
</Root>
```


OpenReadStream – Предназначен для открытия ридера по указанной в параметрах таблице. В случае выполнения повторно в рамках одного SystemId, предыдущий ридер автоматически закрывается.

Параметры запроса:

- SystemId - см. выше,
- CallID - см. выше,
- TableName - обязательный, строка, имя таблицы, для которой открывается ридер,
- UpdateNumFrom - версия данных клиента (максимальная версия с которой получены данные),
- UpdateNumTo - версия данных сервера (полученная из запроса LoadDataSync для данной таблицы)

Параметры ответа:

- ErrorInfo - см. выше.

Пример запроса:

<http://127.0.0.1:8080/04E02234-A87F-44FD-B7E7-3BA4756B8A53/OpenReadStream?SystemID=qwl23&CallID=12&TableName=UNIT&UpdateNumFrom=0&UpdateNumTo=10>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8" ?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

ReadRecords – Предназначен для вычитки пакета записей из открытого ридера.

Параметры запроса:

- SystemId - см. выше,
- CallID - см. выше,
- PacketSize - обязательный, целое, количество записей получаемых в ответе на запрос (при повторном запросе возвращается следующий пакет данных и т.д.)

Параметры ответа:

- ErrorInfo - см. выше.
- IsEndOfStream - признак того что из ридера вычитаны все данные (т.е. повторять запрос следует до тех пор, пока не будет получен ответ с IsEndOfStream=true, что будет соответствовать вычитке всех доступных данных)

В теге `CommandResults` содержатся записи, соответствующие структуре запрашиваемой таблицы.

Пример запроса:

<http://127.0.0.1:8090/04E02234-A87F-44FD-B7E7-3BA4756B8A53/ReadRecords?SystemID=qw123&CallID=12&PacketSize=1000>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8" ?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <StreamState>
    <IsEndOfStream>true</IsEndOfStream>
  </StreamState>
  <CommandResults>
    <UNIT>
      <UNITID>0</UNITID>
      <UNITNAME>0</UNITNAME>
      <UNITFULLNAME>0</UNITFULLNAME>
      <DEFAULTPACKDTYPE>0</DEFAULTPACKDTYPE>
      <DELFLAG>1</DELFLAG>
      <DEFAULTQUANTMASK>0</DEFAULTQUANTMASK>
      <UPDATENUM>1</UPDATENUM>
    </UNIT>
    <UNIT>
      <UNITID>1</UNITID>
      <UNITNAME>Ит</UNITNAME>
      <UNITFULLNAME />
      <DEFAULTPACKDTYPE>1</DEFAULTPACKDTYPE>
      <DELFLAG>0</DELFLAG>
      <DEFAULTQUANTMASK>0</DEFAULTQUANTMASK>
      <UPDATENUM>4</UPDATENUM>
    </UNIT>
    <UNIT>
      <UNITID>2</UNITID>
      <UNITNAME>кр</UNITNAME>
      <UNITFULLNAME>Kr</UNITFULLNAME>
      <DEFAULTPACKDTYPE>0</DEFAULTPACKDTYPE>
      <DELFLAG>0</DELFLAG>
      <DEFAULTQUANTMASK>0</DEFAULTQUANTMASK>
      <UPDATENUM>2</UPDATENUM>
    </UNIT>
  </CommandResults>
</Root>
```

```

</UNIT>
= <UNIT>
    <UNITID>4</UNITID>
    <UNITNAME>ящ</UNITNAME>
    <UNITFULLNAME>Ящик</UNITFULLNAME>
    <DEFAULTPACKDTYPE>1</DEFAULTPACKDTYPE>
    <DELFLAG>0</DELFLAG>
    <DEFAULTQUANTMASK>0</DEFAULTQUANTMASK>
    <UPDATENUM>3</UPDATENUM>
</UNIT>
</CommandResults>
</Root>

```

GetLastPacket – Предназначен для повторного получения последнего пакета (в случае если на запрос пакета пришел некорректный ответ).

Параметры запроса:

- SystemId - см. выше,
- CallID - см. выше.

Параметры ответа:

Соответствуют параметрам *ReadRecords*.

Пример запроса:

<http://127.0.0.1:8090/04E02234-A87F-44FD-B7E7-3BA4756B8A53/GetLastPacket?SystemID=qw123&CallID=12>

Пример ответа:

```

<?xml version="1.0" encoding="utf-8" ?>
= <Root>
= <ErrorInfo>
    <ErrorCode>0</ErrorCode>
</ErrorInfo>
= <StreamState>
    <IsEndOfStream>true</IsEndOfStream>
</StreamState>
= <CommandResults>
    = <UNIT>
        <UNITID>0</UNITID>
        <UNITNAME>0</UNITNAME>
        <UNITFULLNAME>0</UNITFULLNAME>
        <DEFAULTPACKDTYPE>0</DEFAULTPACKDTYPE>

```

```

        <DELFLAG>1</DELFLAG>
        <DEFAULTQUANTMASK>0</DEFAULTQUANTMASK>
        <UPDATENUM>1</UPDATENUM>
    </UNIT>
    - <UNIT>
        <UNITID>1</UNITID>
        <UNITNAME>Шт</UNITNAME>
        <UNITFULLNAME />
        <DEFAULTPACKDTYPE>1</DEFAULTPACKDTYPE>
        <DELFLAG>0</DELFLAG>
        <DEFAULTQUANTMASK>0</DEFAULTQUANTMASK>
        <UPDATENUM>4</UPDATENUM>
    </UNIT>
    - <UNIT>
        <UNITID>2</UNITID>
        <UNITNAME>кр</UNITNAME>
        <UNITFULLNAME>Кр</UNITFULLNAME>
        <DEFAULTPACKDTYPE>0</DEFAULTPACKDTYPE>
        <DELFLAG>0</DELFLAG>
        <DEFAULTQUANTMASK>0</DEFAULTQUANTMASK>
        <UPDATENUM>2</UPDATENUM>
    </UNIT>
    - <UNIT>
        <UNITID>4</UNITID>
        <UNITNAME>ящ</UNITNAME>
        <UNITFULLNAME>Ящик</UNITFULLNAME>
        <DEFAULTPACKDTYPE>1</DEFAULTPACKDTYPE>
        <DELFLAG>0</DELFLAG>
        <DEFAULTQUANTMASK>0</DEFAULTQUANTMASK>
        <UPDATENUM>3</UPDATENUM>
    </UNIT>
</CommandResults>
</Root>

```

CloseReadStream – Принудительно закрывает ридер (желательно закрывать ридер, если не собираетесь дочитывать его до конца и открывать следующий.).

Параметры запроса:

- SystemId - см. выше,
- CallID - см. выше.

Параметры ответа:

- ErrorInfo - см. выше.

Пример запроса:

<http://127.0.0.1:8090/04E02234-A87F-44FD-B7E7-3BA4756B8A53/CloseReadStream?SystemID=qw123&CallID=12>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8" ?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

Технология работы с плагином.

Для корректной работы с плагином клиентское приложение (далее - клиент) должно хранить у себя список импортируемых таблиц (не обязательно все, которые предоставляет шлюз) и последнюю версию данных по каждой из них.

Сеанс синхронизации начинается с вычитки списка предоставляемых шлюзом таблиц (метод *LoadDataSync*). Сопоставляя версию данных каждой из клиентских таблиц с версией соответствующей таблицы шлюза следует принимать решение о необходимости выполнения запроса для получения новых данных (если версии одинаковы – новых данных нет). Для получения новых данных по конкретной таблице следует открыть для нее ридер (*OpenReadStream*) указав в качестве параметра *UpdateNumFrom* значение версии данных хранящееся на клиенте, а в качестве *UpdateNumTo* – значение версии полученное для запрашиваемой таблицы из списка предоставляемых шлюзом таблиц (*LoadDataSync*). В случае получения положительного ответа на запрос открытия ридера (*ErrorCode=0*) следует последовательно, по пакетно вычитывать новые данные при помощи метода *ReadRecords*. Размер получаемого пакета определяется параметром *PacketSize* этого запроса. В случае получения некорректного пакета, его всегда можно повторно запросить при помощи *GetLastPacket*. Выполнять запрос *ReadRecords* следует до тех пор, пока очередной ответ не будет содержать *IsEndOfStream=true*, что соответствует полностью вычитанному ридеру.

Вычитывать новые данные по таблицам следует в соответствии с «приоритетом вычитки таблицы» (*SyncPriority*) полученном в запросе *LoadDataSync* что обеспечит выполнение условия целостности получаемых данных.

Каждый следующий цикл начинается с запроса *LoadDataSync*.

Плагин «Шлюз интеграции с бэк-офисом».

Шлюз может работать как на корневом сервере, так и на любом из подчиненных менеджеров (для работы обязательно требуется их серверная часть, с которой собственно и работает шлюз). Возможно использование нескольких шлюзов, работающих с одним и тем же или с разными серверными частями. При этом гарантируется уникальность идентификаторов создаваемых объектов вне зависимости от местоположения шлюза и распространение этих объектов по всей системе.

Работа с плагином

Работа плагином производится при помощи запросов к его методам через протокол HTTP. Запрос формируется в следующем виде:

[http://<Сервер>:<порт>/идентификатор/<метод>? \[&Параметры\]](http://<Сервер>:<порт>/идентификатор/<метод>? [&Параметры])

Здесь:

- <Сервер> - IP-адрес сервера,
- <порт> - порт, по которому работает шлюз (по умолчанию 8080),
- идентификатор – строковая константа (GUID), указывающая на конкретный плагин (шлюз поддерживает произвольное кол-во плагинов, каждый из которых должен иметь уникальный идентификатор). Для настоящего плагина идентификатор имеет вид:
5BC0F616-93ED-4E58-ACE5-37A1577C6A07.

Методы плагина.

ReservePK – Предназначен для получения пула зарезервированных идентификаторов, для создания новых сущностей в OpenStore.

Параметры запроса:

- SystemId - обязательный, строка, уникальный идентификатор системы, для которой выполняются запросы (если у вас работает несколько клиентов шлюза, то они должны иметь разные значения этого параметра).
- CallID - обязательный, строка, уникальный номер запроса.
- TableName - обязательный, строка, имя таблицы, для которой получаем идентификаторы.
- FieldName - обязательный, строка, поле-первичный ключ таблицы.
- PKCount - обязательный, число, количество идентификаторов в пуле.

Параметры ответа:

- ErrorInfo - код ошибки, 0 – если ошибок нет.

Пример запроса:

<http://127.0.0.1:8080/5BC0F616-93ED-4E58-ACE5-37A1577C6A07/ReservePK?CallID=635399125761610413&SystemID=TestClient&TableName=Unit&FieldName=UnitID&PKCount=2>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <Unit>
      <UnitID>2005</UnitID>
    </Unit>
    <Unit>
      <UnitID>2006</UnitID>
    </Unit>
  </CommandResults>
</Root>
```

LoadTableSchema – Предназначен для вычитки схемы таблицы по ее имени.

Параметры запроса:

- SystemId - обязательный, строка, уникальный идентификатор системы, для которой выполняются запросы (если у вас работает несколько клиентов шлюза, то они должны иметь разные значения этого параметра).
- CallID - обязательный, строка, уникальный номер запроса.
- TableName - обязательный, строка, имя таблицы, для которой получаем идентификаторы.

Параметры ответа:

- ErrorInfo - код ошибки, 0 – если ошибок нет.

Пример запроса:

<http://127.0.0.1:8080/5BC0F616-93ED-4E58-ACE5-37A1577C6A07/LoadTableSchema?CallID=635399182079031585&SystemID=TestClient&TableName=Unit>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
```

```

</ErrorInfo>
<CommandResults>
  <Unit>
    <ColumnName>UNITID</ColumnName>
    <DataType>1</DataType>
    <AllowNull>False</AllowNull>
  </Unit>
  <Unit>
    <ColumnName>UNITNAME</ColumnName>
    <DataType>2</DataType>
    <AllowNull>False</AllowNull>
  </Unit>
  <Unit>
    <ColumnName>UNITFULLNAME</ColumnName>
    <DataType>2</DataType>
    <AllowNull>True</AllowNull>
  </Unit>
  <Unit>
    <ColumnName>DEFAULTPACKDTYPE</ColumnName>
    <DataType>1</DataType>
    <AllowNull>False</AllowNull>
  </Unit>
  <Unit>
    <ColumnName>DELFLAG</ColumnName>
    <DataType>1</DataType>
    <AllowNull>False</AllowNull>
  </Unit>
  <Unit>
    <ColumnName>DEFAULTQUANTMASK</ColumnName>
    <DataType>1</DataType>
    <AllowNull>False</AllowNull>
  </Unit>
  <Unit>
    <ColumnName>UPDATENUM</ColumnName>
    <DataType>1</DataType>
    <AllowNull>False</AllowNull>
  </Unit>
</CommandResults>
</Root>

```

SaveTables – Предназначен для передачи данных по одной или нескольким таблицам. Если таблица содержит данные, первичные ключи которых уже содержатся в базе данных, то

автоматически выполняться изменение существующих данных, в противном случае будут добавлены новые записи. Является POST-запросом.

Параметры запроса:

- SystemId - обязательный, строка, уникальный идентификатор системы, для которой выполняются запросы (если у вас работает несколько клиентов шлюза, то они должны иметь разные значения этого параметра).
- CallID - обязательный, строка, уникальный номер запроса.

Параметры ответа:

- ErrorInfo - код ошибки, 0 – если ошибок нет.

Пример запроса:

- запрос:

<http://127.0.0.1:8080/5BC0F616-93ED-4E58-ACE5-37A1577C6A07/SaveTables?CallID=635399839254729885&SystemID=TestClient>

- содержание запроса:

```
<?xml version="1.0" encoding="utf-16"?>
<Tables>
  <Unit>
    <UNITID>2006</UNITID>
    <UNITNAME>шт_ш</UNITNAME>
    <UNITFULLNAME>шт_шлюз</UNITFULLNAME>
    <DEFAULTPACKDTYPE>1</DEFAULTPACKDTYPE>
    <DELFLAG>0</DELFLAG>
    <DEFAULTQUANTMASK>0</DEFAULTQUANTMASK>
    <UPDATENUM>0</UPDATENUM>
  </Unit>
</Tables>
```

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

Технология работы с плагином.

Шлюз может работать как на корневом сервере, так и на любом из подчиненных менеджеров (для работы обязательно требуется их серверная часть, с которой собственно и работает шлюз). Возможно использование нескольких шлюзов, работающих с одним и тем же или с разными серверными частями. При этом гарантируется уникальность идентификаторов создаваемых

объектов вне зависимости от местоположения шлюза и распространение этих объектов по всей системе.

Прежде чем создать новый объект необходимо получить для него уникальный идентификатор. Это делается при помощи метода *ReservePK*. Вы можете получать новый идентификатор каждый раз, при создании нового объекта или получив в одном запросе пул идентификаторов, минимизировать кол-во запросов к шлюзу (определяется параметром *PKCount*). При этом, если у вас несколько приложений собираются создавать элементы одного и того же справочника, вы можете быть уверены, что все они (приложения) получат уникальные идентификаторы.

Для получения XML-документа можно использовать схему интересующей вас таблицы полученную методом *LoadTableSchema*. Однако, при желании вы можете создавать XML-документ вручную. Созданные или измененные объекты передаются серверу при помощи метода *SaveTables*. Следует отметить, что при передаче нескольких таблиц или одной таблицы с несколькими записями, вся операция выполняется в транзакции.

Плагин «WEB-отчеты»

Плагин *WEB-отчетов* предназначен для решения задач получения различных отчетов посредством веб-приложений.

Настройки плагина «WEB-отчеты»

Настройки плагина открываются по двойному щелчку мышки на самом плагине. Откроется окно «Настройки плагина» содержащее такие пункты:

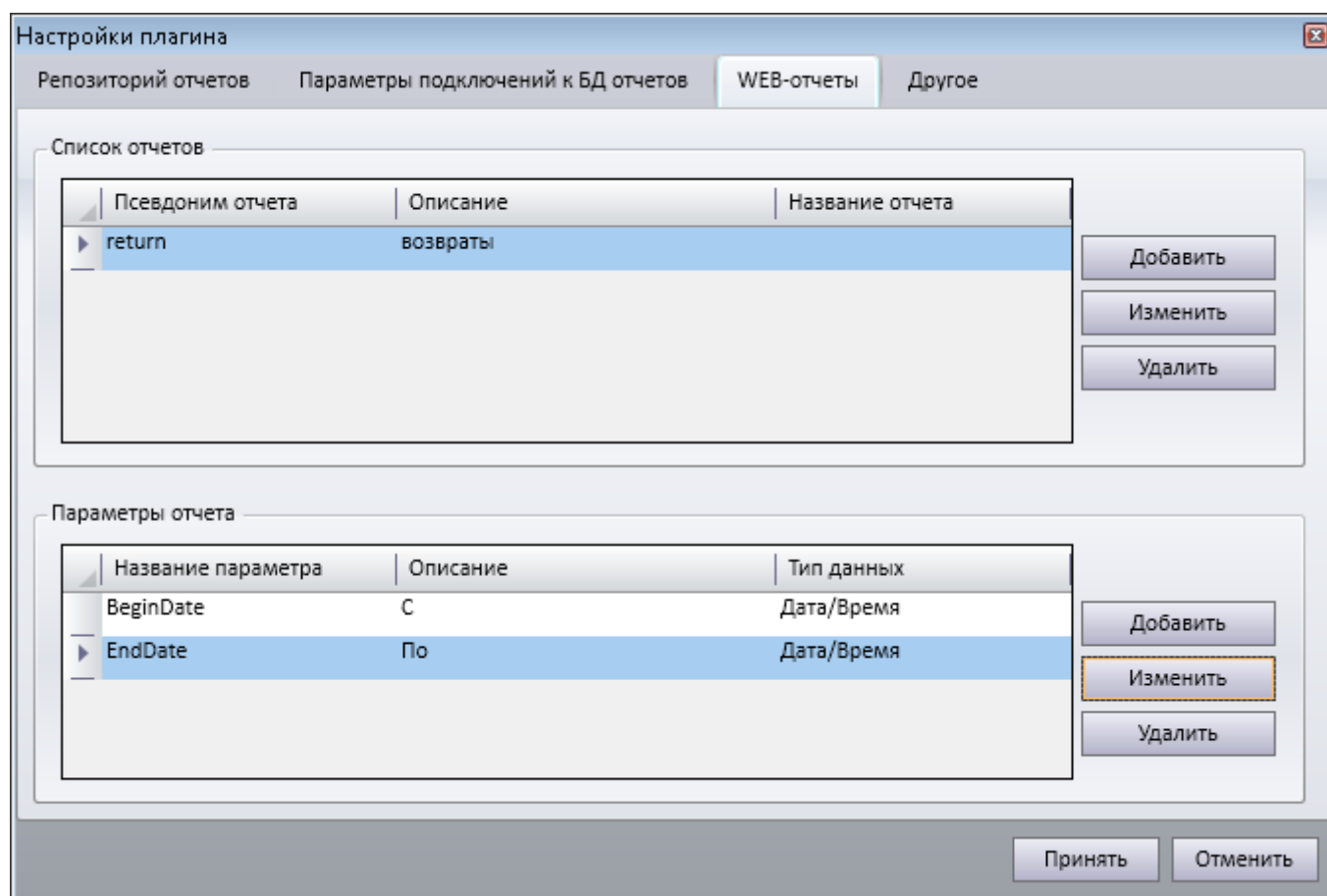


Рис. 4 Настройки плагина Web-отчетов, вкладка «Отчеты»

Вкладка «Репозиторий отчетов»

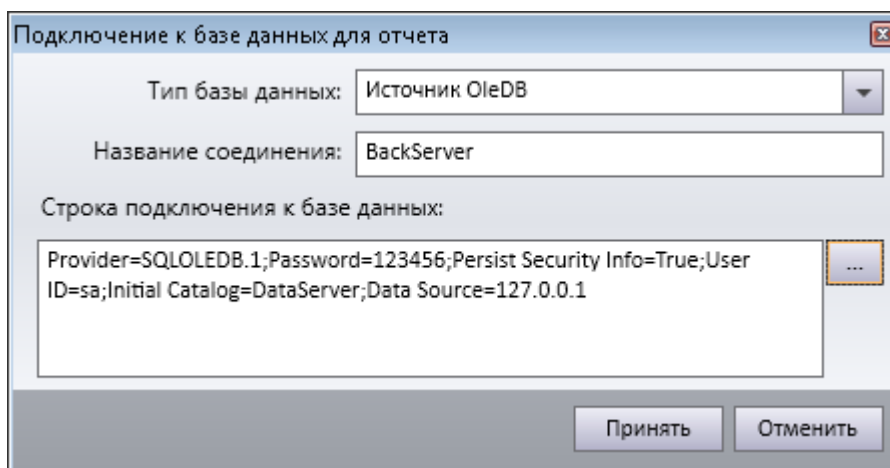
Для добавления нового отчета во вкладке «Отчеты», нажмите кнопку «Добавить» появится строка, состоящая из 2 пунктов:

«Название отчета» – Название отчета, которое будет понятно пользователю.

«Файл отчета» – указывается путь к файлу отчета в формате .mrt, который будет формироваться, и отображаться в веб приложении.

Вкладка «Параметры подключения к БД отчетов»

«Настроить соединение с базой данных» позволяет настроить соединение с базой данных, с которой будут взаимодействовать отчеты. Заданная в настройках строка соединения с базой данных передается глобальной переменной отчета `ConnectionString`, таким образом, в отчетах можно задавать данную переменную, чтобы получить строку соединения из настроек плагина.



Параметры подключения к базам отчетов» - добавляются алиасы и настраиваются подключения к базам данных, с которыми работают отчеты. Отчеты, поставляемые в рамках программного комплекса OpenStore, работают с одним алиасом, который имеет фиксированное имя - **BackServer**. При добавлении нового подключения, прежде всего, необходимо выбрать тип используемой базы, данный в одноименном поле, после чего в «Название соединения» указывается алиас. В зависимости от используемого типа СУБД во вкладки «Поставщик данных» необходимо выбрать провайдера. Для MS SQL Server используется «Microsoft OLE DB Provider for SQL Server», для PostgreSQL достаточно указать параметры подключения к БД. После выбора провайдера перейдите на вкладку «Подключение», где необходимо настроить подключение к базе данных, параметры подключения будут зависеть от выбранного провайдера

Вкладка «WEB-отчеты»

Для добавления WEB отчета в одноименной вкладке, нажмите кнопку «Добавить» появиться строка, состоящая из таких пунктов:

- Псевдоним – здесь отображено наименование, которое должно указываться в URL-строке HTTP-пакета при вызове конкретного запроса, в качестве вызываемого ресурса.
- Отчет – из раскрывающегося списка выберите необходимый отчет. Тут будут отображаться все отчеты, которые были добавлены во вкладке «Репозиторий отчетов».
- Описание – произвольное описание отчета, которое будет понятно пользователю.

В области «Параметры отчета» необходимо задать переменные, которые используются в отчете. Для этого при нажатии «Добавить» в поле «Имя параметра» необходимо ввести имя переменной, которое можно будет задавать при формировании отчета. В поле «Описание» указывается описание данной переменной.

Список отчетов

Псевдоним отчета	Описание	Название отчета
return	возвраты	

Добавить
Изменить
Удалить

Параметры отчета

Название параметра	Описание	Тип данных
BeginDate	С	Дата/Время
EndDate	По	Дата/Время

Добавить
Изменить
Удалить

Работа с плагином

Работа с плагином производится при помощи запросов к его методам через протокол HTTP.

Запрос формируется в следующем виде:

[http://<Сервер>:<порт>/идентификатор/<метод>? \[&Параметры\]](http://<Сервер>:<порт>/идентификатор/<метод>? [&Параметры])

Здесь:

- <Сервер> - IP-адрес сервера,
- <порт> - порт, по которому работает шлюз (по умолчанию 8080),
- идентификатор – строковая константа (GUID), указывающая на конкретный плагин (шлюз поддерживает произвольное кол-во плагинов, каждый из которых должен иметь уникальный идентификатор). Для настоящего плагина идентификатор имеет вид:
- A0DA34B9-524C-4333-BF3C-E277368C3C60.

Методы плагина.

GetReportsList – Предназначен для получения списка отчетов, предоставляемых шлюзом (используется порталами, предоставляющими доступ к отчетам).

Параметры запроса:

- SystemId - обязательный, строка, уникальный идентификатор системы, для которой выполняются запросы (если у вас работает несколько клиентов шлюза, то они должны иметь разные значения этого параметра).

Параметры ответа:

- ErrorInfo - код ошибки, 0 – если ошибок нет,
- ReportAlias - алиас конкретного отчета (необходим для получения отчета),
- Description - текстовое описание отчета (для отображения отчета в списке).

Пример запроса:

<http://127.0.0.1:8080/A0DA34B9-524C-4333-BF3C-E277368C3C60/GetReportsList?SystemID=qw123>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8" ?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <Reports>
      <ReportAlias>test1</ReportAlias>
      <Description>Первый тестовый отчет</Description>
    </Reports>
  </CommandResults>
</Root>
```

GetReportParameters – Предназначен для получения списка параметров, требуемых конкретным отчетом (используется порталами, предоставляющими доступ к отчетам).

Параметры запроса:

- SystemId - обязательный, строка, уникальный идентификатор системы, для которой выполняются запросы (если у вас работает несколько клиентов шлюза, то они должны иметь разные значения этого параметра).
- AliasName - обязательный, строка, алиас конкретного отчета (ReportAlias из ответа на метод *GetReportsList*)

Параметры ответа:

- ErrorInfo - код ошибки, 0 – если ошибок нет,
- ParameterName - имя параметра,
- Description - текстовое описание параметра (для отображения отчета в списке).
- ParameterTypeName - тип данных параметра (для построения диалога выбора на портале).

Пример запроса:

<http://127.0.0.1:8080/A0DA34B9-524C-4333-BF3C-E277368C3C60/GetReportParameters?SystemID=qw123&AliasName=test1>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8" ?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <ReportParameters>
      <ParameterName>Test</ParameterName>
      <Description>Тестовый параметр</Description>
      <ParameterTypeName>String</ParameterTypeName>
    </ReportParameters>
  </CommandResults>
</Root>
```

RunReport – Предназначен для получения HTML-страницы, представляющей собой собственно отчет.

Параметры запроса:

- *SystemId* - обязательный, строка, уникальный идентификатор системы, для которой выполняются запросы (если у вас работает несколько клиентов шлюза, то они должны иметь разные значения этого параметра).
- *AliasName* - обязательный, строка, алиас конкретного отчета (*ReportAlias* из ответа на метод *GetReportsList*)
- [Параметры отчета] параметры, требуемые отчетом (могут быть получены при помощи метода *GetReportParameters*).

Пример запроса:

<http://127.0.0.1:8080/A0DA34B9-524C-4333-BF3C-E277368C3C60/RunReport?AliasName=test1&SystemID=qw123&Test=someString>

Пример ответа:

Ответом будет HTML-страница, представляющая собой отчет.

Технология работы с плагином.

Для просмотра отчетов достаточно использовать метод *RunReport* с соответствующими параметрами. Такой подход приемлем для просмотра простых, по возможности не требующих или требующих постоянных параметров отчетов.

Для большего удобства пользователей целесообразно строить WEB-портал – просмотрщик отчетов. Все остальные методы данного шлюза предназначены именно для этого. Собственно, портал должен выглядеть следующим образом:

- первая страница – список доступных отчетов (получаемых методом *GetReportsList*);
- при выборе конкретного отчета – страница с параметрами этого отчета (список параметров и их типы получаем при помощи *GetReportParameters*);
- при выборе «показать» формируем отчет методом *RunReport*.

Плагин лояльности.

Технология работы с плагином.

Работа со шлюзом в самом упрощенном виде сводится к созданию и оплате / закрытию чека, что в свою очередь приводит к расчету скидок (для созданного чека) и расчету и генерации записей «начисления» бонусов, после закрытия чека или записей «списания» бонусов, в случае чека возврата. При использовании в качестве оплаты чека полностью (*ReceiptPaymentToClose*) или частично (*ReceiptPayment*) типа оплаты *PaymentType=2* (оплата бонусами) выполняется онлайн-операция списания средств со счетов процессинга в соответствии с указанной в настройках плагина схемой оплаты. При этом непосредственное обращение к процессингу осуществляет сам плагин с использованием любого из принятых в OpenStore каналов (Tcp / HTTP) в т.ч. и с использованием SSL, что позволяет строить безопасные системы без использования дополнительных сторонних средств.

Каждая работающая со шлюзом система должна обладать настройками номера торговой площадки (должна быть заведена в системе OpenStore в соответствующем справочнике) и номера системы (уникальный в пределах торговой площадки целочисленный идентификатор). При необходимости, каждая система может одновременно работать с несколькими чеками (определяется количеством слотов чека указываемых в настройках плагина).

Построение системы лояльности.

Основной задачей решаемой плагином является использование решения на основе OpenStore в качестве внешней системы лояльности для сторонних фронт-офисных систем.

Для работы с лояльностью OpenStore сторонний фронт-офис должен получить онлайн-связь каждого экземпляра фронт-офиса с плагином лояльности. При этом количество плагинов не ограничено и архитектурно плагин может быть развернут как один на магазин или на несколько магазинов, так и персонально для каждого экземпляра фронт-офиса.

В зависимости от особенностей стороннего фронт-офиса формирование чека в слоте может выполняться как одним запросом (*OpenSaleReceipt* или *OpenSaleReceiptWithLoad* с содержанием полной фактуры чека) так и последовательно, добавляя в фактуру чека товары по мере изменения исходного чека фронт-офиса (чек создается одним из методов *OpenSaleReceipt* или *OpenSaleReceiptWithLoad* с пустым содержанием и далее фактура формируется методами добавления элементов фактуры и изменения количества *AddReceiptItems*, *AddReceiptItemsWithLoad* и *EditReceiptItemsQuantity*), что в свою очередь позволит динамически пересчитывать скидки по чеку в процессе его формирования как это делает касса OpenStore. Для отображения рассчитанных скидок и их распределения по позициям в любой момент чек может быть получен методом *GetReceipt*.

Для полноценной работы системы лояльности и управления средствами OpenStore акциями в разрезе товаров, в БД OpenStore должна присутствовать номенклатура сторонней системы. Однако если это по каким-либо причинам является нежелательным, то возможна работа и без номенклатуры. В этом случае для создания чека следует использовать методы «без дозагрузки из БД» (*OpenSaleReceipt*, *AddReceiptItems*). В этом случае следует помнить, что для корректной работы с минимальными ценами они должны передаваться в фактуре чека (при «дозагрузке» минимальные цены загружаются из БД и ими можно управлять средствами OpenStore).

Построение интернет-магазина.

Плагин лояльности можно использовать также для построения полноценного интернет-магазина. Особенностью такого решения будет использование методов формирования чека с «дозагрузкой из БД» что позволит работать с ценами, актуальными для системы OpenStore для торговой площадки, по которой работает интернет-магазин.

Работа с плагином лояльности

Работа плагином производится при помощи запросов к его методам через протокол HTTP. Запрос формируется в следующем виде:

[http://<Сервер>:<порт>/идентификатор/<метод>? \[&Параметры\]](http://<Сервер>:<порт>/идентификатор/<метод>? [&Параметры])

Здесь:

- <Сервер> - IP-адрес сервера,
- <порт> - порт, по которому работает шлюз (по умолчанию 8080),
- идентификатор – строковая константа (GUID), указывающая на конкретный плагин (шлюз поддерживает произвольное кол-во плагинов, каждый из которых должен иметь уникальный идентификатор). Для настоящего плагина идентификатор имеет вид:
21138A37-6E3F-462B-9A62-126BA27646F0.

Описание общих параметров.

Общие параметры запросов:

- *CallID* - обязательный, строка, уникальный номер запроса.
- *SareaID* - обязательный, целое, идентификатор торговой площадки, на которой работает касса.
- *SystemID* - обязательный, целое, идентификатор системы (кассы), уникальный в пределах торговой площадки.
- *OperationID* - обязательный, строка, уникальный идентификатор операции (возможно использование GUID-а).

Параметры фактуры чека:

- *PackID* – целое, идентификатор упаковки товара;

- *ReceiptItemID* – уникально в пределах чека, целое, номер товарной позиции в чеке;
- *Quantity* – целое, количество товара (в единицах если QuantityType=1 или в тысячных, при остальных значениях QuantityType);
- *QuantityType* – целое, 0 – делимый, 1- не делимый, 2 – делимый по маске;
- *PartialType* – целое, 0 если QuantityType 0 или 1, иначе 1- делимый по 0.5, 2 – делимый по 0.25;
- *Price* – целое, цена в копейках;
- *MinPrice* - целое, минимальная цена в копейках;
- *Sum* - целое, сумма в позиции, в копейках;
- *ItemDiscount* – целое, сумма скидки на позицию, в копейках;
- *ReceiptDiscount* – целое, сумма скидки на чек, приходящаяся на позицию, в копейках;
- *MinBonusPrice* – целое, цена, используемая как минимальная при оплате бонусами, в копейках;
- *MinBonusChargePrice* – целое, цена используемая при начислении бонусов как минимальная, в копейках.

Общие параметры ответов:

- *ErrorCode* – код ошибки, 0 – если нет ошибок.
- *ErrorMessage* – текстовое описание ошибки (может быть отключено в настройках плагина шлюза.).

Методы плагина.

OpenSaleReceipt – Предназначен для открытия (создания нового) чека в слоте.

Параметры запроса:

- *SlotID* - обязательный, целое, номер слота в котором открывается чек для данного SystemID. Кол-во одновременных слотов ограничено настройками плагина, по умолчанию 3.

Содержание запроса:

Содержанием запроса является фактура чека, что позволяет открыть чек в слоте непосредственно с фактурой. Обязательными для фактуры являются параметры: PackId, ReceiptItemID, Quantity, QuantityType, PartialType, Price. В случае отсутствия содержания запроса, в указанном слоте будет открыт пустой чек (для дальнейшего добавления фактуры существует специальный метод).

Параметры ответа:

- *ReceiptID* – уникальный идентификатор чека, открытый в слоте (все дальнейшие операции с чеком будут выполняться с использованием этого идентификатора).

Пример запроса:

<http://127.0.0.1:8080/21138A37-6E3F-462B-9A62-126BA27646F0/OpenSaleReceipt?CallID=635851069686750650&SystemID=333&SAreaID=1&OperationID=124f16e6-19c9-418c-bdc8-7a198e54d290&SlotID=1>

Пример содержания (может быть пустым):

```
<?xml version="1.0" encoding="utf-16"?>
<Tables>
  <ReceiptItems>
    <PackID>3</PackID>
    <ReceiptItemID>4</ReceiptItemID>
    <Quantity>1</Quantity>
    <QuantityType>1</QuantityType>
    <PartialType>1</PartialType>
    <Price>1200</Price>
    <MinPrice>0</MinPrice>
    <Sum>1200</Sum>
    <ItemDiscount>0</ItemDiscount>
    <ReceiptDiscount>0</ReceiptDiscount>
    <MinBonusPrice>0</MinBonusPrice>
    <MinBonusChargePrice>0</MinBonusChargePrice>
  </ReceiptItems>
</Tables>
```

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <ReceiptID>F462CF9B-B2B3-4298-AF83-F1586162AF20</ReceiptID>
  </CommandResults>
</Root>
```

OpenSaleReceiptWithLoad – Предназначен для открытия (создания нового) чека в слоте с дозагрузкой данных о элементах фактуры из БД.

Параметры запроса:

- *SlotID* - обязательный, целое, номер слота в котором открывается чек для данного SystemID. Кол-во одновременных слотов ограничено настройками плагина, по умолчанию 3.

Содержание запроса:

Содержанием запроса является фактура чека, что позволяет открыть чек в слоте непосредственно с фактурой. Обязательными для фактуры являются параметры: PackId, ReceiptItemID, Quantity, QuantityType, PartialType. Если в фактуре будет определен параметр Price, то он не будет переопределены данными из БД при дозагрузке.

В случае отсутствия содержания запроса, в указанном слоте будет открыт пустой чек (для дальнейшего добавления фактуры существует специальный метод).

Параметры ответа:

- *ReceiptID* – уникальный идентификатор чека, открытый в слоте (все дальнейшие операции с чеком будут выполняться с использованием этого идентификатора).

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/OpenSaleReceiptWithLoad?CallID=635851086729435436&SystemID=333&SAreaID=1&OperationID=8bb510b6-c967-4a1b-ad99-5ee82c52bbf3&SlotID=1>

Пример содержания (может быть пустым):

```
<?xml version="1.0" encoding="utf-16"?>
<Tables>
  <ReceiptItems>
    <PackID>3</PackID>
    <ReceiptItemID>4</ReceiptItemID>
    <Quantity>1</Quantity>
    <QuantityType>1</QuantityType>
    <PartialType>1</PartialType>
    <Price>1200</Price>
    <MinPrice>0</MinPrice>
    <Sum>1200</Sum>
    <ItemDiscount>0</ItemDiscount>
    <ReceiptDiscount>0</ReceiptDiscount>
    <MinBonusPrice>0</MinBonusPrice>
    <MinBonusChargePrice>0</MinBonusChargePrice>
  </ReceiptItems>
</Tables>
```

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <ReceiptID>F462CF9B-B2B3-4298-AF83-F1586162AF20</ReceiptID>
  </CommandResults>
</Root>
```

AddReceiptItems – Предназначен для добавление в фактуру существующего чека товара.

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad* и *OpenSaleReceipt*).

Содержание запроса:

Содержанием запроса является фактура чека. Она может содержать как одну так и несколько товарных позиций. Обязательными для фактуры в этом методе являются параметры: *PackId*, *ReceiptItemId*, *Quantity*, *QuantityType*, *PartialType*, *Price*.

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/AddReceiptItems?CallID=635851727731518658&SystemID=333&SAreaID=1&OperationID=edf08770-263d-4cc4-baf6-8e4197a226e6&ReceiptID=F0755FF9-079D-46D5-ABA6-4FA99B94DFAC>

Пример содержания:

```
<?xml version="1.0" encoding="utf-16"?>
<Tables>
  <ReceiptItems>
    <PackID>3</PackID>
    <ReceiptItemID>4</ReceiptItemID>
    <Quantity>1</Quantity>
    <QuantityType>1</QuantityType>
    <PartialType>1</PartialType>
    <Price>1200</Price>
    <MinPrice>0</MinPrice>
    <Sum>1200</Sum>
    <ItemDiscount>0</ItemDiscount>
    <ReceiptDiscount>0</ReceiptDiscount>
    <MinBonusPrice>0</MinBonusPrice>
    <MinBonusChargePrice>0</MinBonusChargePrice>
  </ReceiptItems>
</Tables>
```

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

AddReceiptItemsWithLoad – Предназначен для добавление в фактуру существующего чека товара с дозагрузкой данных о элементах фактуры из БД.

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad* и *OpenSaleReceipt*).

Содержание запроса:

Содержанием запроса является фактура чека. Она может содержать как одну так и несколько товарных позиций. Обязательными для фактуры в этом методе являются параметры: *PackId*, *ReceiptItemID*, *Quantity*, *QuantityType*, *PartialType*. Если в фактуре будет определен параметр *Price*, то он не будут переопределены данными из БД при дозагрузке.

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/AddReceiptItemsWithLoad?CallID=635851736290228188&SystemID=333&SAreaID=1&OperationID=ca2228e8-15a6-41a3-9419-5a8228f619d6&ReceiptID=F0755FF9-079D-46D5-ABA6-4FA99B94DFAC>

Пример содержания:

```
<?xml version="1.0" encoding="utf-16"?>
<Tables>
  <ReceiptItems>
    <PackID>3</PackID>
    <ReceiptItemID>4</ReceiptItemID>
    <Quantity>1</Quantity>
    <QuantityType>1</QuantityType>
    <PartialType>1</PartialType>
  </ReceiptItems>
</Tables>
```

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

EditReceiptItemsQuantity – Предназначен для изменения количества в заданной товарной позиции.

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad* и *OpenSaleReceipt*).

Содержание запроса:

Содержанием запроса является фактура чека (в сочетаниях номер позиции в чеке + количество в позиции). Она может содержать как одну так и несколько товарных позиций. Обязательными для фактуры в этом методе являются параметры: *ReceiptItemID*, *Quantity*, *QuantityType*, *PartialType*.

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/EditReceiptItemsQuantity?CallID=635851740072164502&SystemID=333&SAreaID=1&OperationID=bb3dc7e3-689a-46d7-b232-4cb0a073d22c&ReceiptID=F0755FF9-079D-46D5-ABA6-4FA99B94DFAC>

Пример содержания:

```
<?xml version="1.0" encoding="utf-16"?>
<Tables>
  <ReceiptItems>
    <ReceiptItemID>4</ReceiptItemID>
    <Quantity>15</Quantity>
    <QuantityType>1</QuantityType>
    <PartialType>1</PartialType>
  </ReceiptItems>
</Tables>
```

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

RemoveReceiptItem – Предназначен для удаления позиции из чека.

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad* и *OpenSaleReceipt*).

Содержание запроса:

Содержанием запроса является фактура чека (номера позиций). Обязательным для фактуры в этом методе является параметр: *ReceiptItemID*.

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/RemoveReceiptItem?CallID=635851743786786967&SystemID=333&SAreaID=1&OperationID=c3bfl678-90a0-447e-a898-511a8d022732&ReceiptID=F0755FF9-079D-46D5-ABA6-4FA99B94DFAC>

Пример содержания:

```
<?xml version="1.0" encoding="utf-16"?>
<Tables>
  <ReceiptItems>
    <ReceiptItemID>4</ReceiptItemID>
  </ReceiptItems>
</Tables>
```

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

AddClientByDCard – Предназначен для добавление в чек клиента посредством его карты.

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad* и *OpenSaleReceipt*).
- *DCardCode* - обязательный, строка, код карты клиента.

Параметры ответа:

- *ClientName* – имя клиента на которого указывает использованная карта клиента.

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/AddClientByDCard?CallID=635851746320231871&SystemID=333&SAreaID=1&OperationID=819e7765-a56d-41d8-8808-a09dc02687f1&ReceiptID=F0755FF9-079D-46D5-ABA6-4FA99B94DFAC&DCardCode=293061459870>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <ClientName>Иванов Иван Иванович</ClientName>
  </CommandResults>
</Root>
```

AddClientById – Предназначен для добавление в чек клиента указания его идентификатора в БД.

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad* и *OpenSaleReceipt*).
- *ClientID* - обязательный, целое, идентификатор клиента.

Параметры ответа:

- *ClientName* – имя клиента на которого указывает использованная карта клиента.

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/AddClientById?CallID=635851753873803911&SystemID=333&SAreaID=1&OperationID=6e51564c-7874-43fb-950d-84c5de7915b7&ReceiptID=F0755FF9-079D-46D5-ABA6-4FA99B94DFAC&ClientID=10>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <ClientName>Иванов Иван Иванович</ClientName>
  </CommandResults>
</Root>
```

RemoveDCard – Предназначен для удаления клиента из чека.

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad* и *OpenSaleReceipt*).

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/RemoveDCard?CallID=635851756502914287&SystemID=333&SAreaID=1&OperationID=4b68f66f-e652-4a74-8533-2af5294083b3&ReceiptID=F0755FF9-079D-46D5-ABA6-4FA99B94DFAC>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

ReceiptPaymentToClose – Предназначен для полного закрытия чека конкретным видом оплаты.

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad* и *OpenSaleReceipt*).
- *PaymentType* - обязательный, число, 0 – наличными, 1 – банковской картой, 2 – бонусами (при оплате бонусами выполняется операция с процессингом).
- *DCardCode* – обязательный только для *PaymentType=2*, строка, код карты клиента, применяемой для оплаты бонусами

Параметры ответа:

- *ReceiptIDToRefund* – идентификатор чека, предназначенный для формирования чека возврата.
- *ProcessingTransactionID* – идентификатор транзакции оплаты через процессинг (при оплатах отличных от 2 этот параметр имеет пустое значение).

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/ReceiptPaymentToClose?CallID=635851871028484775&SystemID=333&SArealID=1&OperationID=be1d36b4-15e8-4e10-bf63-920c1b4bfa8e&ReceiptID=A4B10AC6-C51C-4D00-9135-2E1A0D72EABB&PaymentType=2&DCardCode=293061459870>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <ReceiptIDToRefund>0001014D0009002F</ReceiptIDToRefund>
    <ProcessingTransactionID>7df56e2f-6b18-4128-b244-2ec02e9b900c</ProcessingTransactionID>
  </CommandResults>
</Root>
```

ReceiptPayment – Предназначен для выполнения оплаты чека конкретным видом оплаты, в т.ч. и частичной, т.е. не на всю сумму чека (что не приводит к закрытию чека).

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad* и *OpenSaleReceipt*).
- *PaymentType* - обязательный, число, 0 – наличными, 1 – банковской картой, 2 – бонусами (при оплате бонусами выполняется операция с процессингом).
- *PaymentSum* - обязательный, число, сумма оплаты в копейках.
- *DCardCode* – обязательный только для *PaymentType=2*, строка, код карты клиента, применяемой для оплаты бонусами

Параметры ответа:

- *IsReceiptClosed* – признак того, что чек закрыт.
- *ReceiptIDToRefund* – идентификатор чека, предназначенный для формирования чека возврата (присутствует только если чек закрыт, т.е. *IsReceiptClosed=true*).
- *ProcessingTransactionID* – идентификатор транзакции оплаты через процессинг (при оплатах отличных от 2 этот параметр имеет пустое значение).

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/ReceiptPayment?CallID=635851878866503084&SystemID=333&SArealID=1&OperationID=fc079c94-375c-40d8-ac6e-e4dcdeccaccf&ReceiptID=C187AB30-0EDB-46C7-AE5F-71FD8CDFE452&PaymentType=0&PaymentSum=32700>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <IsReceiptClosed>True</IsReceiptClosed>
    <ReceiptIDToRefund>0001014D00090030</ReceiptIDToRefund>
    <ProcessingTransactionID />
  </CommandResults>
</Root>
```

```
</CommandResults>
</Root>
```

OpenRefundReceipt – Предназначен для открытия (создания нового) чека возврата в слоте на основании ранее созданного чека продажи.

Параметры запроса:

- *SlotID* - обязательный, целое, номер слота в котором открывается чек для данного SystemID. Кол-во одновременных слотов ограничено настройками плагина, по умолчанию 3.
- *ReceiptIDToRefund* - обязательный, строка, идентификатор оплаченного (закрытого) чека (возвращается методами *ReceiptPayment* и *ReceiptPaymentToClose*).

Содержание запроса:

Содержанием запроса является фактура чека, что позволяет открыть чек в слоте непосредственно с фактурой. Обязательными для фактуры являются параметры: PackId, Quantity, QuantityType, PartialType. В случае отсутствия содержания запроса, в указанном слоте будет открыт пустой чек возврата (для дальнейшего добавления фактуры используются методы *AddReceiptItemsWithLoad* и *AddReceiptItems* с обязательными параметрами PackId, Quantity, QuantityType, PartialType).

Следует помнить, что QuantityType и PartialType передаваемого товара должны соответствовать параметрам кол-ва ожидаемого чеком возврата.

Параметры ответа:

ReceiptID – уникальный идентификатор чека, открытый в слоте (все дальнейшие операции с чеком будут выполняться с использованием этого идентификатора).

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/OpenRefundReceipt?CallID=635851120910150461&SystemID=333&SAreaID=1&OperationID=429e872d-605f-4f6b-91c5-f195a51b1f65&SlotID=1&ReceiptIDToRefund=0001014D0009000A>

Пример содержания:

```
<?xml version="1.0" encoding="utf-16"?>
<Tables>
  <ReceiptItems>
    <PackID>777</PackID>
    <Quantity>500</Quantity>
    <QuantityType>2</QuantityType>
    <PartialType>1</PartialType>
  </ReceiptItems>
</Tables>
```

Пример ответа (фрагмент):

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <ReceiptID>5BEC26E4-5856-4FCF-B1AD-366F1DBBC679</ReceiptID>
  </CommandResults>
</Root>
```

CancelReceipt– Предназначен для отмены чека в слоте и освобождения слота. Отменен может быть, как чек продажи, так и чек возврата.

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad*, *OpenSaleReceipt* и *OpenRefundReceipt*).

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/CancelReceipt?CallID=635852005519367214&SystemID=333&SAreaID=1&OperationID=d35be9d8-2f57-4296-9da6-885caaf04212&ReceiptID=5C923A50-B734-46AF-862D-F0DF6F61C77E>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

GetReceipt – Предназначен для получения чека из слота. Может быть применен как к чеку продажи, так и к чеку возврата

Параметры запроса:

- *ReceiptID* - обязательный, строка, идентификатор чека в слоте, возвращаемый методами открытия чека (*OpenSaleReceiptWithLoad*, *OpenSaleReceipt* и *OpenRefundReceipt*).

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/GetReceipt?CallID=635852009292563028&SystemID=333&SAreaID=1&OperationID=4f5b84d6-c692-4a13-b6f3-79cb69dc4a75&ReceiptID=7E990B1C-E7F3-424D-B968-339F77B8D3B3>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <ReceiptData />
  <ReceiptItems>
    <ReceiptItems>
      <PackID>3</PackID>
      <ReceiptItemID>0</ReceiptItemID>
      <Quantity>1</Quantity>
      <QuantityType>1</QuantityType>
      <PartialType>0</PartialType>
      <Price>1200</Price>
      <MinPrice>0</MinPrice>
      <Sum>1100</Sum>
      <ItemDiscount>100</ItemDiscount>
      <ReceiptDiscount>0</ReceiptDiscount>
      <MinBonusPrice>0</MinBonusPrice>
      <MinBonusChargePrice>0</MinBonusChargePrice>
      <OfferID>0</OfferID>
    </ReceiptItems>
  </ReceiptItems>
</Root>
```

GetReceiptIdBySlotId – Предназначен для получения идентификатора чека, находящегося в слоте.

Параметры запроса:

- *SlotID* - обязательный, целое, номер слота в котором открыт чек для данного SystemID.

Кол-во одновременных слотов ограничено настройками плагина, по умолчанию 3

Параметры ответа:

- *ReceiptID* – уникальный идентификатор чека, открытый в слоте (все дальнейшие операции с чеком будут выполняться с использованием этого идентификатора).

Пример запроса:

<http://127.0.0.1:8090/21138A37-6E3F-462B-9A62-126BA27646F0/GetReceiptIdBySlotId?CallID=635852011505809619&SystemID=333&SAreaID=1&OperationID=cdb04b5a-f5dd-4d2a-93b6-ff9dc07bab31&SlotID=1>

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <ReceiptID>7E990B1C-E7F3-424D-B968-339F77B8D3B3</ReceiptID>
  </CommandResults>
</Root>
```

Плагин интеграции с процессингом.

Работа с плагином интеграции с процессингом

Работа с плагином производится при помощи запросов к его методам через протокол HTTP.

Запрос формируется в следующем виде:

[http://<Сервер>:<порт>/идентификатор/<метод>? \[&Параметры\]](http://<Сервер>:<порт>/идентификатор/<метод>? [&Параметры])

Здесь:

- <Сервер> - IP-адрес сервера,
- <порт> - порт, по которому работает шлюз (по умолчанию 8080),
- идентификатор – строковая константа (GUID), указывающая на конкретный плагин (шлюз поддерживает произвольное кол-во плагинов, каждый из которых должен иметь уникальный идентификатор). Для настоящего плагина идентификатор имеет вид:
3347027D-5209-4FC9-99A6-85D12EAED3B9.

Описание общих параметров.

Общие параметры запросов:

- *CallID* - обязательный, строка, уникальный номер запроса.
- *SareaID* - обязательный, целое, идентификатор торговой площадки, на которой работает касса.
- *SystemID* - обязательный, целое, идентификатор системы (кассы), уникальный в пределах торговой площадки.
- *OperationID* - обязательный, строка, уникальный идентификатор операции (возможно использование GUID-а).

Методы плагина.

PaymentDocument – Предназначен для выполнения операции оплаты посредством процессинга.

Параметры запроса:

- *PaymentSchemeID* - обязательный, целое, идентификатор схемы оплаты.
- *CurrencyID* - обязательный, целое, идентификатор валюты, в которой передается сумма.
- *DcardCode* – обязательный, строка, код карты клиента, на основании которой производится оплата.
- *Sum* – обязательный, целое, сумма оплаты в копейках и в указанной в параметре *CurrencyID* валюте.

Параметры ответа:

- *AccountDocID* – уникальный идентификатор транзакции оплаты.

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/PaymentDocument?SystemID=1Q2&operationID=12&CallID=120&paymentSchemeID=1001¤cyID=1&dcardCode=10010&sum=212&SAreaID=1>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <AccountDocID>a3ed5914-127e-49c4-9a9a-9f0113b5341a</AccountDocID>
  </CommandResults>
</Root>
```

PaymentRefundDocument – Предназначен для выполнения операции возврата части суммы оплаты на основании транзакции оплаты.

Параметры запроса:

- *DocumentID* - обязательный, строка, идентификатор транзакции оплаты, возвращаемый методом *PaymentDocument* в параметре *AccountDocID*.
- *Sum* – обязательный, целое, сумма возврата в копейках и в указанной в транзакции оплаты валюте.

Параметры ответа:

- *AccountDocID* – уникальный идентификатор транзакции возврата.

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/PaymentRefundDocument?SystemID=1Q2&operationID=12&CallID=120&documentID=b30a6397-fe39-47f5-9399-88df58a36d26&sum=123&SAreaID=1>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <AccountDocID>726287bd-cf99-4f1c-97b4-d01c37b89d15</AccountDocID>
  </CommandResults>
</Root>
```

OpenAccount – Предназначен для открытия счета клиента.

Параметры запроса:

- *CurrencyID* - обязательный, целое, идентификатор валюты, в которой передается сумма.
- *AccountTypeID* - обязательный, целое, идентификатор типа счета.
- *ClientID* - обязательный, целое, идентификатор клиента, для которого создается счет.
- *IsBlocked* - обязательный, целое, 0 – счет создается незаблокированный, 1 - счет создается заблокированным.
- *Comment* – не обязательный, строка, описание к счету.

Параметры ответа:

- *AccountID* – уникальный идентификатор счета.

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/OpenAccount?SystemID=1Q2&operationID=12212&CallID=12&accountTypeID=2003¤cyID=1&clientID=9&comment=Комментарий к счету&isBlocked=0>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <AccountID>f62ba06c-07ba-4f61-af5c-c0f7685593c9</AccountID>
  </CommandResults>
</Root>
```

OpenCreditAccount – Предназначен для открытия кредитного счета клиента.

Параметры запроса:

- *CurrencyID* - обязательный, целое, идентификатор валюты, в которой передается сумма.
- *AccountTypeID* - обязательный, целое, идентификатор типа счета.
- *ClientID* - обязательный, целое, идентификатор клиента, для которого создается счет.
- *MaxCredit* - обязательный, целое, максимальная глубина кредита в копейках.
- *IsBlocked* - обязательный, целое, 0 – счет создается незаблокированный, 1 - счет создается заблокированным.
- *Comment* – не обязательный, строка, описание к счету.

Параметры ответа:

- *AccountID* – уникальный идентификатор счета.

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/OpenCreditAccount?SystemID=1Q2&operationID=2232&CallID=12&accountTypeID=2004¤cyID=1&clientID=9&comment=qwertyComment&isBlocked=0&maxCredit=1000>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <AccountID>b603a0fa-a878-4cf0-bc98-914fe85ed268</AccountID>
  </CommandResults>
</Root>
```

SetMaxCredit – Предназначен для управления максимальной глубиной кредита.

Параметры запроса:

- *AccountID* - обязательный, строка, идентификатор счета, для которого устанавливается новая глубина кредита (это идентификатор, который возвращает сетод создания счета).

- *Sum* - обязательный, целое, новая глубина кредита в копейках.

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/SetMaxCredit?SystemID=1Q2&operationID=411&CallID=120&accountID=b603a0fa-a878-4cf0-bc98-914fe85ed268&sum=3000>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

LockAccount – Предназначен для блокировки счета.

Параметры запроса:

- *AccountID* - обязательный, строка, идентификатор счета, для которого выполняется блокировка (это идентификатор, который возвращает сетод создания счета).

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/LockAccount?SystemID=1Q2&operationID=5&CallID=120&accountID=f62ba06c-07ba-4f61-af5c-c0f7685593c9>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

UnLockAccount – Предназначен для отмены блокировки счета.

Параметры запроса:

- *AccountID* - обязательный, строка, идентификатор счета, для которого отменяется блокировка (это идентификатор, который возвращает метод создания счета).

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/UnLockAccount?SystemID=1Q2&operationID=511&CallID=120&accountID=f62ba06c-07ba-4f61-af5c-c0f7685593c9>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
</Root>
```

LoadAccounts – Предназначен для получения счетов клиента.

Параметры запроса:

- *ClientID* – обязательный, целое, идентификатор клиента, для которого получаем счета.
- *OnlyUnlocked* – обязательный, целое, 0 – все счета клиента, 1 – только не заблокированные счета клиента.

Параметры ответа:

- *AccountID* – уникальный идентификатор счета.
- *AccountTypeName* – наименование типа счета.
- *IsCredit* – признак того, что счет кредитный.
- *IsDeposit* – признак того, что счет депозитный.
- *CurrencyName* – наименование валюты.
- *AccountSum* – сумма на счету в копейках.
- *MaxCredit* – глубина кредита в копейках.
- *IsBlocked* – признак того, что счет заблокирован.
- *IsClosed* – признак того, что счет закрыт.
- *OpenTime* – дата / время открытия счета.
- *CloseTime* – дата / время закрытия счета.
- *Comment* – текстовый комментарий к счету (определяется при создании счета, может отсутствовать).
- *CurrencyID* – идентификатор валюты счета.
- *AccountTypeID* – идентификатор типа счета.
- *ClientID* – идентификатор клиента счета.

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/LoadAccounts?SystemID=1Q2&operationID=71&CallID=120&clientID=9&onlyUnlocked=0>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <Accounts>
    <Account>
      <AccountID>b603a0fa-a878-4cf0-bc98-914fe85ed268</AccountID>
      <AccountNumber>B603A0FA</AccountNumber>
      <AccountTypeName>Кредит евреев</AccountTypeName>
      <IsCredit>1</IsCredit>
      <IsDeposit>0</IsDeposit>
      <KindDescription>Безналичный</KindDescription>
      <IsCash>0</IsCash>
      <IsCashless>1</IsCashless>
      <CurrencyName>Грн</CurrencyName>
      <AccountSum>0</AccountSum>
      <MaxCredit>3000</MaxCredit>
      <IsBlocked>0</IsBlocked>
      <IsLocal>0</IsLocal>
      <IsClosed>0</IsClosed>
      <OpenTime>20151209130705</OpenTime>
      <CloseTime/>
      <Comment>qwertyComment</Comment>
      <CurrencyID>1</CurrencyID>
      <AccountTypeID>2004</AccountTypeID>
      <ClientID>9</ClientID>
    </Account>
  </Accounts>
</Root>
```

```

<Account>
  <AccountID>f62ba06c-07ba-4f61-af5c-c0f7685593c9</AccountID>
  <AccountNumber>F62BA06C</AccountNumber>
  <AccountTypeName>Питание евреев</AccountTypeName>
  <IsCredit>0</IsCredit>
  <IsDeposit>1</IsDeposit>
  <KindDescription>Безналичный</KindDescription>
  <IsCash>0</IsCash>
  <IsCashless>1</IsCashless>
  <CurrencyName>Грн</CurrencyName>
  <AccountSum>0</AccountSum>
  <MaxCredit>0</MaxCredit>
  <IsBlocked>1</IsBlocked>
  <IsLocal>0</IsLocal>
  <IsClosed>0</IsClosed>
  <OpenTime>20151209125332</OpenTime>
  <CloseTime/>
  <Comment>комментарий к счету</Comment>
  <CurrencyID>1</CurrencyID>
  <AccountTypeID>2003</AccountTypeID>
  <ClientID>9</ClientID>
</Account>
</Accounts>
</Root>

```

AddSumDocument – Предназначен для выполнения операции пополнения счета.

Параметры запроса:

- *AccountID* - обязательный, строка, идентификатор счета, для которого выполняется пополнение (это идентификатор, который возвращает метод создания счета).
- *Sum* - обязательный, целое, сумма пополнения в копейках.

Параметры ответа:

AccountDocID – уникальный идентификатор транзакции пополнения.

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/AddSumDocument?SystemID=1Q2&operationID=118&CallID=120&accountID=fdb8e66b-08eb-469d-be99-33568df7e4d2&sum=112>

Пример ответа:

```

<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <AccountDocID>852384b1-28d4-4989-bf77-930e1daa5931</AccountDocID>
  </CommandResults>
</Root>

```

SubSumDocument – Предназначен для выполнения операции списания со счета.

Параметры запроса:

- *AccountID* - обязательный, строка, идентификатор счета, по которому производится операция списания (это идентификатор, который возвращает метод создания счета).
- *Sum* - обязательный, целое, сумма списания в копейках.

Параметры ответа:

AccountDocID – уникальный идентификатор транзакции списания.

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/SubSumDocument?SystemID=1Q2&operationID=119&CallID=120&accountID=fdb8e66b-08eb-469d-be99-33568df7e4d2&sum=112>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <AccountDocID>6cfec342-98e9-48d8-a1ea-94e233ac673f</AccountDocID>
  </CommandResults>
</Root>
```

MoveDocument – Предназначен для выполнения операции перемещения средств с одного счета на другой (с той же валютой).

Параметры запроса:

- *AccountIDFrom* - обязательный, строка, идентификатор счета, с которого списывается сумма.
- *AccountIDTo* - обязательный, строка, идентификатор счета, на который зачисляется сумма.
- *Sum* - обязательный, целое, сумма перемещаемая со счета на счет в копейках.

Параметры ответа:

AccountDocID – уникальный идентификатор транзакции отмены.

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/MoveDocument?SystemID=1Q2&SAreaID=1&operationID=141111&CallID=120&accountIDFrom=fdb8e66b-08eb-469d-be99-33568df7e4d2&sum=1000&accountIDTo=d1749f87-7d72-4a05-a5ff-1c907d04b025>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <AccountDocID>198970d1-9ca8-4b8d-b7b2-d123db1072f1</AccountDocID>
  </CommandResults>
</Root>
```

CancelDocument – Предназначен для выполнения операции отмены транзакции.

Параметры запроса:

- *AccountID* - обязательный, строка, идентификатор счета, по которому отменяется транзакция.
- *DocumentID* - обязательный, строка, идентификатор транзакции которая подлежит отмене.

Параметры ответа:

AccountDocID – уникальный идентификатор транзакции отмены.

Пример запроса:

<http://127.0.0.1:8090/3347027D-5209-4FC9-99A6-85D12EAED3B9/CancelDocument?SystemID=1Q2&operationID=110&CallID=120&accountID=fdb8e66b-08eb-469d-be99-33568df7e4d2&DocumentID=6cfec342-98e9-48d8-a1ea-94e233ac673f&SAreaID=1>

Пример ответа:

```
<Root>
  <ErrorInfo>
    <ErrorCode>0</ErrorCode>
  </ErrorInfo>
  <CommandResults>
    <AccountDocID>b5de9a12-b88a-45d5-ba79-936142ad4b31</AccountDocID>
  </CommandResults>
</Root>
```