

Руководство разработчика АСР LANBilling 1.9. Описание API.

Платформы:

«Интернет»

(агенты «Ethernet», «NetFlow», «SFlow», «RADIUS DialIN»)

«Телефония»

(агенты «PABX», «PCDR», «RADIUS VoIP»)

«ТВ/Телематика»

(агент «USBox»)

Информация для контактов:

Адрес в интернет: <http://www.lanbilling.ru>

Общие вопросы: itdep@lanbilling.ru

Техническая поддержка: support@lanbilling.ru

Телефоны: +7 (495) 780-4420, +7 (495) 795-0677
с 10-00 до 19-00

Оглавление

РУКОВОДСТВО РАЗРАБОТЧИКА АСР LANBILLING 1.9. ОПИСАНИЕ API	1
ВВЕДЕНИЕ. ДЛЯ КОГО ПРЕДНАЗНАЧЕН ЭТОТ ДОКУМЕНТ	6
ОБЩАЯ ИНФОРМАЦИЯ	6
ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ, СПОСОБСТВУЮЩАЯ ПОНИМАНИЮ ТЕХНОЛОГИИ SOAP	6
RНР БИБЛИОТЕКА (КЛАСС LANBILLING)	7
СТРУКТУРА БИБЛИОТЕКИ	7
АВТОРИЗАЦИЯ	8
БАЗИС ФУНКЦИЙ	8
get()	8
save()	9
delete()	9
WSDL – ОПИСАНИЕ СЕРВИСА LANBILLING И ЕГО ПРОГРАММНОГО ИНТЕРФЕЙСА	10
ОБЩАЯ ИНФОРМАЦИЯ	10
ТИПОВЫЕ ОБЪЕКТЫ LANBILLING 1.9 SERVER	10
Объект soapFilter	10
ПРИКЛАДНЫЕ ФУНКЦИИ	12
Функция getAccounts	12
Функции getStat и getClientStat	18
Функция getAccount	21
Функция getVgroup	27
Функция delAccount	35
Функция delVgroup	36
Функция insupdVgroup	37
Функция insupdAccount	38
Функция getDocuments	40
Функция getPayments	42
Функция insupdDocument	45
Функция insupdTarfsRasp	47
Функция delTarfsRasp	50
Функция getTarfsRasp	51
Функция getCurrencies	52

Функция getOptions	54
Функция getLicense	55
Функция getRates	57
Функция getCardSets	59
Функция insupdCardSet	61
Функция delCardSet	62
Функция genOrders	63
Функция genOrderGetFile	65
Функция insupdCurrency	67
Функция delCurrency	68
Функция updCurrencyRate	70
Функция delOrder	71
Функция getVgroups	71
Функция getAddressCountries	74
Функция getAddressRegions	76
Функция getAddressAreas	78
Функция getAddressCities	79
Функция getAddressSettles	81
Функция getAddressStreets	82
Функция getAddressBuildings	84
Функция getAddressFlats	86
Функция getAddressMeanings	87
Функция insupdAddressCountry	89
Функция insupdAddressRegion	90
Функция insupdAddressArea	91
Функция insupdAddressCitiy	92
Функция insupdAddressSettle	93
Функция insupdAddressStreet	94
Функция insupdAddressBuilding	95
Функция insupdAddressFlat	96
Функция delAddressCountry	97
Функция delAddressRegion	98
Функция delAddressArea	99
Функция delAddressCitiy	100
Функция delAddressSettle	100
Функция delAddressStreet	101
Функция delAddressBuilding	102
Функция delAddressFlat	103
Функция Payment	104
Функция ExternCheckPayment	105
Функция ExternPayment	106
Функция ExternCancelPayment	107
Функция delVgroup	108
Функция getPayments	109
Функция getAccountsAddonsStaff	111

Функция getAccountsAddonsSet	113
Функция insupdAccountsAddonsSet	114
Функция getAgreementsAddonsStaff	116
Функция getAgreementsAddonsSet	117
Функция insupdAgreementsAddonsSet.....	119
Функция getVgroupsAddonsStaff	120
Функция getVgroupsAddonsSet	121
Функция insupdVgroupsAddonsSet.....	123
Функция blkClientVgroup	124
Функция blkVgroup	125
Функция insupdUsboxService.....	126
Функция getTarCategories.....	128

Введение. Для кого предназначен этот документ

Основная цель данного документа – предоставить внешним разработчикам возможность управления системой из собственных приложений, разработки дополнений и расширений функционала ACP, поставляемого в базовом комплекте, а так же снабдить пользователей ACP LANBilling 1.9 прозрачным механизмом интеграции со смежными решениями сторонних компаний.

В основе API лежит сокетный сервер ACP LANBilling 1.9, который реализует в себе алгоритм обслуживания запросов от программных клиентов (как внешних, так и включенных в комплект поставки ACP, в частности Web интерфейса) на основе технологии SOAP (Simple Objects Access Protocol).

Данный документ постоянно дополняется по мере расширения функционального базиса API и содержит описания используемых для работы с API библиотек. В первую очередь (первая редакция данного документа) описана библиотека для языка PHP. Во второй редакции будет описана внутренняя C++ библиотека.

Общая информация

Сервер ACP LANBilling 1.9 функционирует на TCP порту 34012 и обслуживает запросы только от клиентов, которые расположены на IP адресах, включенных в список доверенных. По умолчанию, при первичной инсталляции ACP только локальный хост (адрес 127.0.0.1) уполномочен соединяться с сервером через порт 34012. Для того, что бы использовать API, необходимо добавить IP адрес клиента в список доверенных. Этот процесс детально описан в документе «Руководство по эксплуатации ACP LANBilling», раздел «Настройка общих параметров (опций) ACP».

Обращение к API сервера происходит под полномочиями одного из пользователей ACP: администратора, менеджера или конечного пользователя. Объем информации, который будет передан пользователю или принят от него зависит от настроек данного пользователя в ACP. Настройками определяется набор данных, с которыми пользователю дозволено работать. Администратор не имеет ограничений по данным. Полномочия конечного пользователя (абонента) так же фиксированы и не настраиваются. Полномочия менеджеров ACP определяются в соответствии с описанием, приведенном в документе «Руководство по эксплуатации ACP LANBilling», раздел «Менеджеры».

Дополнительная информация, способствующая пониманию технологии SOAP

В основе реализованной в сервере ACP LANBilling 1.9 SOAP технологии обмена XML сообщениями между клиентом и сервером лежит спецификация на XML

(Extensible Markup Language – расширяемый язык разметки), найти которую можно по ссылке: <http://www.w3.org/TR/2000/REC-xml-20001006> и которая определяет общую структуру данных, обмен которыми происходит в рамках SOAP. Специфика управления сервисами, реализованными в ACP LANBilling 1.9 описана при помощи стандартного формата описания Web сервисов WSDL (Web Services Description Language), для понимания сути которого имеет смысл обратиться к ресурсам: <http://www.itpedia.ru/index.php/WSDL> и <http://www.w3.org/TR/wsdl>. Последний окажет неоценимую помощь в продвижении к цели написания приложений к ACP LANBilling начинающим использование SOAP разработчикам.

Кроме вышеприведенных источников мы рекомендуем ознакомиться с ресурсами, на которых приводятся практические примеры и советы по использованию SOAP и WSDL:

- <http://webservices.xml.com/pub/a/ws/2003/05/27/wsdl.html> (перевод на русский язык: (<http://xmlhack.ru/texts/03/wsdl.tales/wsdlintralook1.html>))
- <http://webservices.xml.com/pub/a/ws/2003/06/24/wsdl.html> (перевод на русский язык: (<http://xmlhack.ru/texts/03/wsdl.tales/wsdlintralook2.html>))
- <http://www.citforum.ru/database/cache/web-services/>

Кроме описанных выше источников и еще массы ресурсов, которые можно найти при помощи известных поисковых сервисов, крайне полезным для разработчиков на PHP5 будет ресурс: <http://phpclub.ru/detail/article/soap> на котором детально рассмотрен пример написания первого SOAP приложения.

PHP библиотека (класс LANBilling)

Структура библиотеки

Реализация функций библиотеки для работы с сервером ACP LANBilling 1.9 содержится в файлах `soap.class.php` и `main.class.php` административного Web интерфейса системы, который находится в каталоге `/admin/` относительно директории, в которую установлен управляющий Web клиент ACP.

Функции оформлены методами базового для всего Web интерфейса класса LANBilling (определение содержится в файле `main.class.php`), который расширяет (`extends`) класс SOAP, определенный в файле `soap.class.php`

Для использования функций библиотеки в коде приложения требуется создать экземпляр класса LANBilling, в результате чего обеспечивается возможность использования базовых методов класса. Пример создания экземпляра класса LANBilling (объекта LANBilling) имеется в файле-диспетчере Web интерфейса `config.php`:

```
$lanbilling = new LANBilling();
```

Конструктор класса не требует определения аргументов. Все ключевые конфигурационные данные считываются из настроечных файлов.

Авторизация

В функции библиотеки для работы с API встроен механизм разграничения полномочий доступа к данным и управлению АСР. В зависимости от полномочий менеджера, под атрибутами которого осуществлен вход в АСР, сервер обеспечивает контроль над использованием функции API.

Метод `Authorize()` базового класса `LANBilling` запускается из его конструктора при создании объекта. Для корректной аутентификации и успешного создания объекта класса `LANBilling` метод ожидает существование POST переменных `'login'` и `'password'` в адресном пространстве Web приложения. При авторизации содержание этих переменных проверяется на не пустое значение. В случае, если переменные не определены, сессия не создается. Так же завершается ошибкой создание объекта `LANBilling`. В случае успешного создания объекта `LANBilling` приложение пользователя получает доступ к методам класса – базису API функций: `get()`, `save()`, `delete()`.

Базис функций

В большинстве случаев, с точки зрения разработчика внешнего, по отношению к АСР приложения, достаточно использовать 3 базовые функции API библиотеки:

- `get()` – обеспечивает получение данных от сервера АСР `LANBilling`
- `save()` – обеспечивает сохранение или изменение данных АСР
- `delete()` – обеспечивает удаление данных АСР

Каждая из перечисленных функций-методов ожидает передачу ей в качестве входных параметров своего набора объектов, перечисленных в определении функции.

get()

В коде класса SOAP функция получения данных от сервера АСР определяется следующим образом:

```
public function get( $functionName = "", $options = array(), $revalidate = false )
```

Параметр `$functionName` содержит символьное название API функции, доступ к которой запрашивается.

Параметр `$options` содержит массив опций (переменных), который индивидуален для каждой функции API, и влияет, в общем случае, на результирующий набор данных, возвращаемых функцией.

Параметр `$revalidate` булев и содержит флаг необходимости перепроверки данных, полученных при предыдущем запросе, при помощи этой функции.

save()

В коде класса SOAP функция сохранения/изменения данных сервера ACP определяется следующим образом:

```
public function save( $functionName, $options = array(), $insert = false, $flush = array(), $ft = array() )
```

Параметр *\$functionName* содержит символьное название API функции, доступ к которой запрашивается.

Параметр *\$options* содержит массив опций (переменных), который индивидуален для каждой функции API, и влияет, в общем случае, на набор данных, изменяемых функцией.

Параметр *\$insert* булев и содержит признак того, что передаваемые этой функцией API данные новые. Другими словами – признак вставки данных.

Параметр *\$flush* содержит массив названий функций API SOAP кэш которых необходимо обнулить для того что бы эти функции возвращали или изменяли актуальные значения, которые были модифицированы вызовом save(). В частности, если вызов save() влияет на данные объекта «Пользователь» ACP, то целесообразно в качестве параметра *\$flush* передать значение 'getAccount' с целью исключения возврата функцией getAccount не актуальных данных в течение периода SOAP кэширования.

Параметр *\$ft* определяет фильтр (ограничения) на набор записей, подлежащих изменению при помощи данной функции

delete()

В коде класса SOAP функция удаления данных ACP определяется следующим образом:

```
public function delete( $functionName, $options = array(), $flush = array() )
```

Параметр *\$functionName* содержит символьное название API функции, доступ к которой запрашивается.

Параметр *\$options* содержит массив опций (переменных), который индивидуален для каждой функции API, и влияет, в общем случае, на набор данных, удаляемых функцией.

Параметр *\$flush* содержит массив названий функций API SOAP кэш которых необходимо обнулить для того что бы эти функции возвращали или изменяли актуальные значения, которые были модифицированы вызовом delete(). Смысл параметра аналогичен одноименному параметру в функции save(). См. описание параметра функции save() выше.

WSDL – описание сервиса LANBilling и его программного интерфейса

Общая информация

В данном разделе приведено описание функций SOAP API безотносительно средств доступа к ним (под средствами доступа понимается в данном контексте средства библиотеки). Прикладные функции API могут быть использованы из любой среды разработки, который предполагает наличие SOAP клиента.

Структура описания функции содержит:

- раздел описания входных и выходных параметров
- раздел описания определения функции
- пример использования прикладной функции в коде приложения, с использованием поддерживаемых библиотек (PHP и C++)

Большая часть функции в качестве входных параметров принимает стандартные объекты API, которые описаны вне связи с конкретной функцией. Эти объекты описаны в разделе «Типовые объекты SOAP API LANBilling 1.9 Server»

Типовые объекты LANBilling 1.9 Server

Объект soapFilter

Объект используется как входной параметр для функций получения списков объектов. Все его поля являются необязательными, назначение тех или иных полей будет дано при описании функций, использующих soapFilter.

```
<complexType name="soapFilter">
  <sequence>
    <element name="agrmid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="userid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="vgid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="tarid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="taridprev" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="operid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="agentid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="recordid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="istemplate" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="ugroups" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="groups" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="notgroups" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    <element name="direction" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
  </sequence>
</complexType>
```

```
<element name="pgsize" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="pgnum" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="setid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="activated" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="personid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="archive" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="blocked" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="catid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="nodata" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="port" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="proto" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="asnum" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="repnum" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="repdetail" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="mgrid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="onfly" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="docid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="orderid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="common" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="deviceid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
<element name="receipt" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="serno" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="cardkey" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="dtcreated" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="dtactivated" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="dtfrom" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="dto" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="numfrom" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="numto" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="name" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="login" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="vglogin" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="descr" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="tardescr" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="agrmnum" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="email" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="phone" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="inn" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="ip" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="dstip" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="telstaff" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="code" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="telnum" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
<element name="namesound" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
</sequence>
```

```
</complexType>
```

В тех функциях, в которых реализован постраничный вывод, алгоритм обработки элементов реализован так, как показано ниже:

```
if(pgnum)
{
  if (!pgsize) pgsize = 50;
  get *pgsize* records for page *pgnum*;
}
else
{
  get all records;
}
```

Прикладные функции

Функция getAccounts

Получение списка пользователей.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAccountBrief

Объект представляет сокращенный вариант записи из таблицы Accounts (Пользователи АСР). Используется при получении списка пользователей

```
<complexType name="soapAccountBrief">
<sequence>
//id пользователя
  <element name="uid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//количество у.з. пользователя
  <element name="vgcnt" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Тип: 1-юр. лицо, 2-физ. лицо
  <element name="type" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Логин для статистики клиента
  <element name="login" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//ФИО/Название компании
  <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Комментарий
```

```
<element name="descr" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Адрес электронной почты
<element name="email" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Номер телефона
<element name="phone" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Объект soapUsergroup

Объект представляет запись из таблицы Usergroups (Группы пользователей)

```
<complexType name="soapUsergroup">
<sequence>
//id группы
<element name="groupid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Разрешить обещанный платеж
<element name="promiseallow" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Сумма платежа равна АП
<element name="promiserent" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Срок погашения платежа, дн
<element name="promisetill" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//max сумма платежа
<element name="promisemax" type="xsd:double" minOccurs="1" maxOccurs="1"/>
//min сумма платежа
<element name="promisemin" type="xsd:double" minOccurs="1" maxOccurs="1"/>
//Кол-во однорв. платежей
<element name="promiselimit" type="xsd:double" minOccurs="1" maxOccurs="1"/>
//Название группы
<element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Описание
<element name="description" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Объект soapUsergroupFull

Расширяет запись из таблицы Usergroups (Группы пользователей), используется при получении списков.

```
<complexType name="soapUsergroupFull">
<sequence>
//группа пользователей
<element name="usergroup" type="lbaapi:soapUsergroup" minOccurs="1" maxOccurs="1"/>
//кол-во пользователей в группе
```

```
<element name="usercnt" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//у менеджера есть права на чтение пользователей данной группы
<element name="fread" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//у менеджера есть права на запись пользователей данной группы
<element name="fwrite" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Объект soapAgreementBrief

Объект представляет сокращенный вариант записи из таблицы Agreements (Договора или Лицевые счета). Используется при получении списков.

```
<complexType name="soapAgreementBrief">
<sequence>
//id договора
<element name="agrmid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//признак «валюта по умолчанию»
<element name="defcurr" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Текущий баланс лицевого счета
<element name="balance" type="xsd:double" minOccurs="1" maxOccurs="1"/>
//Номер договора
<element name="number" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Символ валюты договора
<element name="symbol" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Код оплаты для внешней системы
<element name="code" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Объект soapAddressBrief

Объект представляет почтовый адрес пользователя или оператора.

```
<complexType name="soapAddressBrief">
<sequence>
//для пользователя 1-физический, 2-юридический или прописки, 3-доставки счета
<element name="type" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//код по КЛАДР, через ',': страна, регион, район, город, нас.пункт, улица, квартира
<element name="code" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//текстовое представление адреса
<element name="address" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Объект soapAccountsListItem

Используется для получения списка пользователей

```
<complexType name="soapAccountsListItem">
  <sequence>
    //пользователь
    <element name="account" type="lbapi:soapAccountBrief" minOccurs="1" maxOccurs="1"/>
    //список групп, в которые он входит
    <element name="usergroups" type="lbapi:soapUsergroupFull" minOccurs="0" maxOccurs="unbounded"/>
    //список договоров
    <element name="agreements" type="lbapi:soapAgreementBrief" minOccurs="0" maxOccurs="unbounded"/>
    //список адресов
    <element name="addresses" type="lbapi:soapAddressBrief" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Определение функции

Получение списка пользователей. На вход подается soapFilter, на выходе принимается список soapAccountsListItem или exception (исключение).

```
<!-- operation request element -->
<element name="getAccounts">
  <complexType>
    <sequence>
      <element name="flt" type="lbapi:soapFilter" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
```

```
</element>
<!-- operation response element -->
<element name="getAccountsResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapAccountsListItem" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```


Проверки полей фильтра

istemplate	0 для пользователей, 1 для шаблонов пользователей
archive	0 для активных, 1 для удаленных
if (ugroups)	пользователи входят в группу с id = ugroups
if (agrmid)	у пользователя есть договор с id = agrmid
if (!name.empty())	название пользователя like '%name.value%'
if (!login.empty())	логин пользователя like '%login.value%'
if (!vglogin.empty())	логин учетной записи пользователя like '%vglogin.value%'
if (!inn.empty())	ИНН пользователя like '%inn.value%'
if (!phone.empty())	телефон пользователя like '%phone.value%' (проверяется именно телефон в таблице Accounts, а не телефоны, привязанные к у.з.)
if (!email.empty())	email пользователя like '%email.value%'
if (!agrmnum.empty())	номер договора пользователя like '%agrmnum.value%'
if (!code.empty())	код договора для связи с внешними системами пользователя like '%code.value%'
if (!namesound.empty())	consonant (название пользователя, 0) regexr 'namesound.value'.
Функция consonant входит в поставку БД АСР	

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAccounts.req.xml](#), [api3.getAccounts.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
// Входные параметры фильтра: ключ и значение
// name - имя абонента
// agrmnum - номер договора
// login - логин абонента
// vglogin - логин учетной записи, которая принадлежит абоненту
// email - email абонента
// phone - телефон абонента
// pgsz - количество строк в странице
// pgnum - номер страницы

// Подготовка фильтра
$filter = array( ... )
// Контрольная сумма массива. Нужна если включен кэш soap ответов
$filter = $lanbilling->controlSum($filter);

// Входные параметры
// Вызываемая функция
// Фильтр записей:
// id - идентификатор пользователя

if (false != ($result = $lanbilling->get("getAccounts", array("flt" => $filter, "md5" => $md5))) )
{
    // Приведение объекта массиву значений
    if(!is_array($result)) {
        $result-> = array($result);
    }

    // Массив объектов
    print_r($result->account);
}
```

Функции getStat и getClientStat

Получение статистики использования тарифицируемых услуг

Для получения статистики (матрицы текстовых значений) от сервера используются функции, определенные как:

```
getStat (soapFilter flt, vector<soapOrderby>ord, vector<soapStat>& ret);
getClientStat (soapFilter flt, vector <soapOrderby>ord, vector<soapStat>& ret);
```

Отличие между функциями состоит только в том, что в клиентской статистике в объект фильтра автоматически подставляется **<userid>** пользователя, создавшего сессию.

Обязательным параметром фильтра является поле **<repnum>**, которое определяет имя процедур и таблиц, используемых при получении статистики. Кроме того, возможно получить только данные о размерности результата и названиях полей, установив параметр фильтра **<nodata>** в 1.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Входной параметр **ord** содержит массив структур:

```
soapOrderby
{
    std::string    name;    // имя поля
    long          ascdesc; // направление сортировки, 0 — asc, 1 — desc
}
```

Согласно этому набору выполняется окончательная сортировка массива записей перед выдачей его клиенту.

Выходной параметр **ret** содержит структуру

```
soapStat
{
    long rows;           // кол-во строк в таблице
    long cols;          // кол-во столбцов в таблице
    soapVectorString    names; // вектор названий столбцов
    vector<soapVectorString> data; // матрица результата
}
```

Где

```
soapVectorString
{
    vector<string> val;
}
```

размерности cols.

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getStat.req.xml](#), [api3.getStat.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
// Входные параметры фильтра: ключ и значение
// Общие
// agentid - идентификатор агента
// username - поиск по имени абонента
// vglogin - поиск по логину учетной записи
// agrmnum - поиск по номеру договора
// vgid - поиск по идентификатору учетной записи
// agrmid - поиск по идентификатору договора
// dtfrom - начало периода
// dtto - окончание периода
// pgnum - номер страницы

// Кабельные агенты
// srcip - поиск по адресу (источнику)
// dstip - поиск по адресу (назначение)
// port - фильтровать по порту
// proto - фильтровать по протоколу
// repdetail = 1 (группировать записи по часам и валюте)
// repdetail = 2 (группировать записи по дням и валюте)
// repdetail = 0 (группировать записи по учетным записям и валюте)
// repnum = 7 (статистика RADIUS)
// repnum = 5 (статистика кабельного агента)

// Телефония
// amountfrom - величина списания "От"
// numfrom - величина списания "До"
// numto - вызываемый номер
// durfrom - длительность "От" (сек)
// durto - длительность "До" (сек)
// direction - направление звонка
// showdefault - присоединить к списку неучтенный трафик
// defaultonly - исключить из списка учетный трафик
// operid - идентификатор оператора (-1 - все операторы)
// repnum = 9

// Услуги
// repnum = 6
// repdetail = 1 (группировать записи по дням)
// repdetail = 2 (группировать записи по услугам)
```

```
// repdetail = 3 (группировать записи по учетным записям)

// Подготовка фильтра
$filter = array( ... )
// Контрольная сумма массива. Нужна если включен кэш soap ответов
$filter = $lanbilling->controlSum($filter);

$result = $lb->get("getStat", array("flt" => $filter, "ord" => "amount", "md5" => $md5));

if(!is_array($result->data)) {
    $result->data = array($result->data);
}

// Группировка ключей и значений в объекте
$tmp = $lanbilling->dataCombine($result->names->val, $result->data);

print_r($tmp);
```

Функция getAccount

Получение полной информации об одном пользователе.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объекты soapAddressBrief и soapUsergroupFull документированы в описании функции getAccounts.

Объект soapAccount

Объект представляет информацию о пользователе (запись из таблицы Accounts).

```
<complexType name="soapAccount">
  <sequence>
    //id пользователя
    <element name="uid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //Доступ в статистику клиента только ip адресов его у.з.
    <element name="ipaccess" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //Способ доставки счета: 0-курьер, 1-почтой, 2-самостоятельно, 3-другой
    <element name="billdelivery" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //Категория абонента: 0-Бюджет, 1-хозрасчет, 2-население, 3-ПБОЮЛ
    <element name="category" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //Тип пользователя: 1-юр. лицо, 2-физ. лицо
```

```
<element name="type" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Код страны абонента по ОКСМ
<element name="oksm" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Запись является шаблоном пользователя
<element name="templ" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Кол-во неудачных попыток активации карты оплаты из кабинета клиента
<element name="wrongactive" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Запись удалена
<element name="archive" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Логин для входа на страничку клиента
<element name="login" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Пароль для входа на страничку клиента
<element name="pass" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Комментарий
<element name="descr" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Имя пользователя: ФИО/Название компании
<element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Номер телефона
<element name="phone" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Номер факса
<element name="fax" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Адрес электронной почты
<element name="email" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Наименование банка
<element name="bankname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Наименование отделения банка
<element name="branchbankname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Наименование казначейства
<element name="treasuryname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Номер лицевого счета в казначействе
<element name="treasuryaccount" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//БИК
<element name="bik" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Расчетный счет
<element name="settl" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Корр. счет
<element name="corr" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//КПП
<element name="kpp" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//ИНН
<element name="inn" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//ОГРН
<element name="ogrn" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//ОКПО
<element name="okpo" type="xsd:string" minOccurs="1" maxOccurs="1"/>
```

```
//ОКВЕД
  <element name="okved" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Ген. Директор
  <element name="gendiru" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Главный бухгалтер
  <element name="glbuhgu" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Контактное лицо
  <element name="kontperson" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Действует на основании
  <element name="actonwhat" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Серия паспорта
  <element name="passsernum" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//№ паспорта
  <element name="passno" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Дата выдачи паспорта
  <element name="passissuedate" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Кем выдан паспорт
  <element name="passissuedep" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Место выдачи паспорта
  <element name="passissueplace" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Дата рождения
  <element name="birthdate" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Место рождения
  <element name="birthplace" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Метка времени последней модификации записи (только чтение)
  <element name="lastmoddate" type="xsd:string" minOccurs="0" maxOccurs="1"/>
//Дата блокировки возможности активации карт оплаты (только чтение)
  <element name="wrongdate" type="xsd:string" minOccurs="0" maxOccurs="1"/>
//ОКАТО
  <element name="okato" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Объект soapAgreement

Объект представляет собой Договора (Лицевой счета) пользователя с оператором. Используется при получении, изменении, добавлении одного пользователя.

```
<complexType name="soapAgreement">
  <sequence>
//id договора
    <element name="agrmid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//id пользователя
    <element name="uid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
  
```

```
//id оператора, с которым заключен договор
  <element name="operid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//id валюты договора
  <element name="curid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Уведомлять о балансе
  <element name="bnotify" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Договор удален
  <element name="archive" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Количество у.з. на договоре, только чтение
  <element name="vgroups" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Текущий баланс лицевого счета
  <element name="balance" type="xsd:double" minOccurs="1" maxOccurs="1"
//Размер кредита
  <element name="credit" type="xsd:double" minOccurs="1" maxOccurs="1"/>
//Пороговое значение баланса для уведомления
  <element name="blimit" type="xsd:double" minOccurs="1" maxOccurs="1"/>
//Номер договора
  <element name="number" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Код оплаты для внешней системы
  <element name="code" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Дата договора
  <element name="date" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Дата последней отправки уведомления о состоянии баланса
  <element name="bcheck" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Символ валюты договора
  <element name="symbol" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
</sequence>
</complexType>
```


Объект soapAccountFull

Объект предоставляет полную информацию о пользователе АСР, включая списки договоров, адресов и групп пользователей. Используется при получении, добавлении и изменении одного объекта.

```
<complexType name="soapAccountFull">
  <sequence>
//Пользователь
    <element name="account" type="Ibapi:soapAccount" minOccurs="1" maxOccurs="1"/>
//Список групп, которым он принадлежит
    <element name="usergroups" type="Ibapi:soapUsergroupFull" minOccurs="0" maxOccurs="unbounded"/>
//Список адресов
    <element name="addresses" type="Ibapi:soapAddressBrief" minOccurs="0" maxOccurs="unbounded"/>
//Список договоров
    <element name="agreements" type="Ibapi:soapAgreement" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Определение функции

Получение одного пользователя. На вход подается id пользователя, на выходе принимается объект soapAccountFull или exception (исключение).

```
<!-- operation request element -->
<element name="getAccount">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getAccountResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapAccountFull" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAccount.req.xml](#), [api3.getAccount.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
// Входные параметры
// Вызываемая функция
// Фильтр записей:
// id - идентификатор пользователя

if( false != ($result = $lanbilling->get("getAccount", array("id" => (integer)$_POST['uid']))) )
{
    // Приведение объекта с договорами пользователя к массиву значений, если не пустой
    if(!empty($result->agreements)) {
        if(!is_array($result->agreements)) {
            $result->agreements = array($result->agreements);
        }
    }

    // Приведение объекта с адресами пользователя к массиву значений, если не пустой
    if(!empty($result->addresses)) {
        if(!is_array($result->addresses)) {
            $result->addresses = array($result->addresses);
        }
    }
}
```

```
}  
    // Приведение объекта с группами в которые входит пользователь к массиву значений,  
    //если не пустой  
    if(!empty($result->usergroups)) {  
        if(!is_array($result->usergroups)) {  
            $result->usergroups = array($result->usergroups);  
        }  
    }  
}  
  
// Все основные статические данные есть в объекте account  
print_r($result->account);  
}
```

Функция getVgroup

Получение учетной записи

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapVgroup

Объект предоставляет информацию об учетной записи (запись из таблицы vgroups).

```
<complexType name="soapVgroup">  
  <sequence>  
    //id записи  
    <element name="vgid" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
    //id родительской у.з.  
    <element name="parentvgid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>  
    //id валюты  
    <element name="cuid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>  
    //id тарифа  
    <element name="tarid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>  
    //id договора  
    <element name="agrmid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>  
    //id пользователя  
    <element name="uid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>  
    //id агента  
    <element name="id" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>  
  //Ограничение полосы пропускания
```

```
<element name="shape" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Полоса пропускания текущая
<element name="currentshape" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Максимальное количество одновременных RADIUS сессий
<element name="maxsessions" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//запрос блокировки
<element name="blkreq" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//блокировка
<element name="blocked" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//признак изменения
<element name="changed" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//шаблон
<element name="templ" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Детализировать по ресурсам
<element name="ipdet" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Детализировать по сервисам
<element name="portdet" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ifindex
<element name="ifindex" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Удалена
<element name="archive" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//категория пользователя
<element name="usercategory" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//проверять телефонный номер, привязанный к у.з., на уникальность (флаг)
<element name="checkduplicate" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//у.з. ожидает перерасчета (флаг)
<element name="dirty" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
// Идентификатор пользователя через которого произведено подключение
<element name="connectedfrom" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Счетчик потребленной услуги для реализации ограничения трафика за период
<element name="dlimit" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//величина списаний
<element name="amount" type="xsd:double" minOccurs="0" maxOccurs="1"
default="0.000000"/>
//Дата первого подключения у.з.
<element name="accodatefirst" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Метка времени создания учетной записи
<element name="creationdate" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Время запланированного подключения уч. записи. obsoleted from 22.09.2011
<element name="accodate" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Время запланированного отключения уч. записи. obsoleted from 22.09.2011
<element name="accoffdate" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Дата последнего изменения состояния блокировки
<element name="blockdate" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Дата начала текущего отчетного периода
```

```
<element name="cdate" type="xsd:string" minOccurs="0" maxOccurs="1" default="" />
//Дата начала текущего отчетного периода
<element name="dclear" type="xsd:string" minOccurs="0" maxOccurs="1" default="" />
//Описание учетной записи
<element name="descr" type="xsd:string" minOccurs="0" maxOccurs="1" default="" />
//Логин для доступа к услуге
<element name="login" type="xsd:string" minOccurs="0" maxOccurs="1" default="" />
//Пароль для доступа к услуге
<element name="pass" type="xsd:string" minOccurs="0" maxOccurs="1" default="" />
//логин родительской у.з.
<element name="parentvglogin" type="xsd:string" minOccurs="0" maxOccurs="1" default="" />
</sequence>
</complexType>
```

Объект soapTarifsRasp

Объект предоставляет информацию о запланированной смене тарифа (запись из иаблицы tariffs_rasp).

```
<complexType name="soapTarifsRasp">
<sequence>
//ID записи
<element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID учетной записи, для которой запланирована смена тарифа (опционально)
<element name="vgid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID договора
<element name="agrmid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID объединения, для которого запланирована смена тарифа (опционально)
<element name="groupid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID учетной записи
<element name="uid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID агента, для учетных записей которого запланирована смена тарифа (обязательный
параметр)
<element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID Новый тариф
<element name="taridnew" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID объединения по тарифу, для которого запланирована смена тарифа (опционально)
<element name="taridold" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Тип агента
<element name="agenttype" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID валюты нового тарифа
```

```
<element name="tarnewcurid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID валюты старого тарифа
<element name="taroldcurid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID менеджера, создавшего правило смены, или NULL, если смена тарифа заказана абонентом
<element name="requestby" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Запланированное время смены тарифа
<element name="changetime" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Логин у.з
<element name="vglogin" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Номер договора
<element name="agrnnum" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Код договора
<element name="code" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Имя пользователя
<element name="acname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Название агента
<element name="agentname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Название нового тарифа
<element name="tarnewname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Название старого тарифа
<element name="taroldname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Менеджер
<element name="mgrname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Символ валюты нового тарифа
<element name="tarnewsymbol" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
// Символ валюты старого тарифа
<element name="taroldsymbol" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
</sequence>
</complexType>
```

Объект soapIPMask

Объект предоставляет пару IP/mask, определяющую сетевой сегмент.

```
<complexType name="soapIPMask">
<sequence>
//IP
<element name="ip" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//маска (1..32)
<element name="mask" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
</sequence>
```

Объект soapStaff

Объект предоставляет сетевой сегмент, назначенный учетной записи (запись из таблицы staff)

```
<complexType name="soapStaff">
  <sequence>
//ID записи
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID учетной записи
    <element name="vgid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Тип сегмента: 0-IP подсеть, 1-номер AS, 2-номер порта
    <element name="type" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Подсеть
    <element name="ipmask" type="lbapi:soapIPMask" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

Объект soapTelStaff

Объект предоставляет телефон/транк, назначенный учетной записи (запись из таблицы tel_staff)

```
<complexType name="soapTelStaff">
  <sequence>
//ID записи
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID учетной записи
    <element name="vgid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Тип устройства: 0-телефон, 1-МТА, 2-транк
    <element name="device" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Выход на МГ/МН направления (флаг)
    <element name="ldservice" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Шаблон телефонного номера/транка
    <element name="phonenumber" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Комментарий
    <element name="comment" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

Объект soapMacStaff

Объект предоставляет транспортный адрес, назначенный учетной записи (запись из таблицы mac_staff)

```
<complexType name="soapMacStaff">
  <sequence>
//ID записи
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID учетной записи
    <element name="vgid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Сегмент
    <element name="segment" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Транспортный адрес
    <element name="mac" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

Объект soapVgroupAddon

Объект предоставляет дополнительное поле учетной записи (запись из таблицы vgroups_addons_vals)

```
<complexType name="soapVgroupAddon">
  <sequence>
//id учетной записи
    <element name="vgid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Тип поля: 0 - текстовое, 1 - список значений
    <element name="type" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//id агента
    <element name="agentid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Внутренний номер записи
    <element name="idx" type="xsd:long" minOccurs="1" maxOccurs="1"/>
// Уникальное имя поля
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Описание поля
    <element name="descr" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Значение поля
    <element name="strvalue" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```


Объект soapCustomService

Объект предоставляет индивидуальную подписку на сервисы/категории в тарифе (запись из таблицы `custom_services`)

```
<complexType name="soapCustomService">
  <sequence>
    //id учетной записи
    <element name="vgid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    //id тарифа
    <element name="tarid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    //id категории
    <element name="catid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
  </sequence>
</complexType>
```

Объект soapBlockRasp

Объект предоставляет информации о запланированной блокировке. (запись из таблицы `vg_blocks_rasp`)

```
<complexType name="soapBlockRasp">
  <sequence>
    //id учетной записи
    <element name="vgid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    //id объединения (не используется)
    <element name="groupid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    //тип блокировки
    <element name="blkreq" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    //id агента
    <element name="id" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    //id менеджера
    <element name="requestby" type="xsd:long" minOccurs="0" maxOccurs="1" default="-1"/>
    //дата блокировки
    <element name="changetime" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
    //фио менеджера (readonly)
    <element name="mgrname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
    //описание менеджера (readonly)
    <element name="mgrdescr" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
  </sequence>
</complexType>
```

Объект soapVgroupFull

Объект предоставляет полную информацию об учетной записи, включая расписание смеи тарифов, списки сегментов, телефонов и транспортных адресов. Используется при получении, добавлении и изменении одного объекта.

```
<complexType name="soapVgroupFull">
  <sequence>
    //учетная запись
    <element name="vgroup" type="lbapi:soapVgroup" minOccurs="1" maxOccurs="1"/>
    //расписание смены тарифов
    <element name="tarrasp" type="lbapi:soapTarifsRasp" minOccurs="0" maxOccurs="unbounded"/>
    //расписание блокировок
    <element name="blockrasp" type="lbapi:soapBlockRasp" minOccurs="0"
maxOccurs="unbounded"/>
    //сетевые сегменты
    <element name="staff" type="lbapi:soapStaff" minOccurs="0" maxOccurs="unbounded"/>
    //номера / транки
    <element name="telstaff" type="lbapi:soapTelStaff" minOccurs="0" maxOccurs="unbounded"/>
    //транспортные адреса
    <element name="macstaff" type="lbapi:soapMacStaff" minOccurs="0" maxOccurs="unbounded"/>
    //доп. поля учетной записи
    <element name="addons" type="lbapi:soapVgroupAddon" minOccurs="0"
maxOccurs="unbounded"/>
    //индивидуальные подписки на сервисы
    <element name="services" type="lbapi:soapCustomService" minOccurs="0"
maxOccurs="unbounded"/>
    <element name="addresses" type="lbapi:soapAddressBrief" minOccurs="0"
maxOccurs="unbounded"/> //адрес оказания услуги
  </sequence>
</complexType>
```

Определение функции

Получение одной учетной записи. На вход подается id , на выходе принимается объект soapVgroupFull или exception (исключение).

```
<!-- operation request element -->
<element name="getVgroup">
<complexType>
<sequence>
```

```
<element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getVgroupResponse">
<complexType>
<sequence>
<element name="ret" type="lbapi:soapVgroupFull" minOccurs="1" maxOccurs="unbounded"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: api3.getVgroup.req.xml, api3.getVgroup.res.xml

Функция delAccount

Перемещение информации о пользователе в архив.

Определение функции

Перемещение в архив информации о пользователе, его договорах и учетных записях. На вход подается id пользователя, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delAccount">
<complexType>
<sequence>
<element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="delAccountResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

</element>

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delAccount.req.xml](#), [api3.delAccount.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
// Входные параметры
// Вызываемая функция
// Фильтр
// id - идентификатор пользователя, который должен быть отмечен как "удален"
// Массив функций, результаты которых должны быть удалены из кэша сессии после успешного
выполнения
if( false == $lanbilling->delete("delAccount", array("id" => (integer)$_POST['deluid']), array("getAccounts",
"getAccount")) ) {
    echo 'There was an error while deleting item. Look server logs for details';
}
}
```

Функция delVgroup

Перемещение информации об учетной записи в архив.

Определение функции

Перемещение в архив информации об учетной записи пользователя. Данная функция оставляет в базе данных метку на то, что запись помечена на удаление, а окончательную обработку производит соответствующий сетевой агент. После помещения записи в архив все связанные с ней списки (сети, номера, MAC, расписания) удаляются. На вход подается id пользователя, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delVgroup">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="delVgroupResponse">
  <complexType>
    <sequence>
```

```
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delVgroup.req.xml](#), [api3.delVgroup.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
// Входные параметры
// Вызываемая функция
// Фильтр
// id - идентификатор учетной записи, которая должна быть отмечена как "удалена"
// Массив функций, результаты которых должны быть удалены из кэша сессии после успешного
//выполнения

if( false == $lanbilling->delete("delVgroup", array("id" => (integer)$_POST['delvgroup']), array("getAccount",
"getVgroup", "getTarifsRasp", "getTarifsHistory", "getVgroups"))) )
{
    echo 'There was an error while deleting item. Look server logs for details';
}
}
```

Функция insupdVgroup

Добавление или изменение информации об учетной записи.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapVgroupFull

Объект soapVgroupFull документирован в описании функции getVgroup.

Определение функции

Добавление новой или изменение существующей учетной записи. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapVgroupFull, на выходе принимается id записи в таблице vgroups (vg_id) или exception.

При добавлении новой учетной записи ее тариф должен быть занесен в расписание (список `tarrasp`). При добавлении у.з. необходимо указание даты включения (список `blockrasp`)

Список дополнительных полей учетной записи удаляется из БД и заполняется новыми значениями из `addons`. Список назначенных блокировок удаляется из БД и заполняется новыми значениями из `blockrasp`.

Из списков расписаний смен тарифа (`tarrasp`), сегментов (`staff`), MAC-адресов (`macstaff`) и телефонных номеров (`telstaff`), обновляются записи, у которых определен `recordid`, добавляются записи с `recordid=0`, остальные удаляются из соответствующих таблиц (`tariffs_rasp`, `staff`, `mac_staff`, `tel_staff`).

```
<!-- operation request element -->
<element name="insupdVgroup">
  <complexType>
    <sequence>
      <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="val" type="lbapi:soapVgroupFull" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="insupdVgroupResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: `api3.insupdVgroup.req.xml`, `api3.insupdVgroup.res.xml`

Функция `insupdAccount`

Добавление или изменение информации о пользователе.

Описание объектов, поступающих в качестве параметров и возвращаемых

функцией

Объект soapAccountFull

Объект soapAccountFull документирован в описании функции getAccount..

Определение функции

Добавление нового или изменение существующего пользователя. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapAccountFull, на выходе принимается id записи в таблице accounts (uid) или exception. В списке договоров пользователя должны приходить все текущие договора, причем soapAgreement::agrmid должен быть нулевым для вновь добавленного договора и ненулевым для измененного. Договора, которых нет в пришедшем списке помечаются как архивные.

Списки предыдущих адресов пользователя и групп, в которые он входит, удаляются из БД, затем записываются новые значения.

```
<!-- operation request element -->
<element name="insupdAccount">
  <complexType>
    <sequence>
      <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="val" type="lbapi:soapAccountFull" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="insupdAccountResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdAccount.req.xml](#), [api3.insupdAccount.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования функции представлен в файле php скрипта административного интерфейса (./admin/user.php в теле функции saveUserData()).

Функция getDocuments

Получение списка документов

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapDocument

Объект представляет информацию о документе (запись из таблицы Documents).

```
<complexType name="soapDocument">
  <sequence>
//ID записи
    <element name="docid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//В сумму документа включен НДС (флаг)
    <element name="ndsabove" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Идентификатор группы пользователей для автоматической генерации документов
    <element name="usergroupid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Идентификатор объединения для автоматической генерации документов
    <element name="groupid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Документ оплачивается (флаг)
    <element name="payable" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Сгенерированные документы доступны из кабинета клиента
    <element name="clientallowed" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Скрыть шаблон из списков при генерации
    <element name="hidden" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//тип документа (0 — счет, 1 — договор, 2 — отчет, 3 — квитанция, )
    <element name="onfly" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//резерв
    <element name="detail" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Имя шаблона
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Имя файла шаблона на диске
    <element name="doctemplate" type="xsd:string" minOccurs="1" maxOccurs="1"/>
```



```
//Путь к папке для сохранения сгенерированных по шаблону документов
<element name="savepath" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Тип MIME для сгенерированных файлов
<element name="uploadext" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Имя группы автоматической генерации ( справочно )
<element name="usergroupname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Имя объединения автоматической генерации ( справочно )
<element name="groupname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
</sequence>
</complexType>
```

Определение функции

Получение списка документов. На вход подается фильтр, на выходе принимается вектор объектов soapDocument или exception (исключение).

В фильтре проверяются следующие поля:

if (flt.docid) //ID записи в таблице

else if (onfly >= 0) //тип документа. Если onfly == -1 , то все записи.

```
<!-- operation request element -->
<element name="getDocuments">
<complexType>
<sequence>
<element name="flt" type="lbapi:soapFilter" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getDocumentsResponse">
<complexType>
<sequence>
<element name="ret" type="lbapi:soapDocument" minOccurs="1" maxOccurs="unbounded"/>
</sequence>
</complexType>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getDocuments.req.xml](#), [api3.getDocuments.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
// Входные параметры
// Вызываемая функция
// Фильтр записей:
// onfly - тип шаблона: -1 - все, 0 - платежный документ, 1 - пользовательский документ, 2 - отчет,
//3 - Квитанция

// Подготовка фильтра и контрольная сумма если используется кэш
$filter = array('docid' => (integer)$_POST['db_document']);
$md5 = $lanbilling->controlSum($filter);

if( false != ($result = $lanbilling->get('getDocuments', array('flt' => $filter, 'md5' => $md5))) )
{
    if(!is_array($result)) {
        $result = array($result);
    }

    print_r($result);
}
```

Функция getPayments

Получение списка платежей

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapPayment

Объект представляет информацию о платеже (запись из таблицы Payments).

```
<complexType name="soapPayment">
  <sequence>
    //Идентификатор записи
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1" default="0"/>
    //Идентификатор лицевого счета, на который поступил платеж
    <element name="agrmid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //Идентификатор менеджера, проводившего платеж
    <element name="modperson" type="xsd:long" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

```
//Идентификатор валюты платежа
<element name="currid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Идентификатор счета в `orders`, если проводилась оплата счета
<element name="orderid" type="xsd:long" minOccurs="1" maxOccurs="1" default="0"/>
//Статус платежа: 0-проведен, 1-проведен и подтвержден сверкой (при загрузке фала сверки),
//2-платеж аннулирован
<element name="status" type="xsd:long" minOccurs="1" maxOccurs="1" default="0"/>
//Величина платежа в валюте лицевого счета
<element name="amount" type="xsd:double" minOccurs="1" maxOccurs="1"/>
//Дата платежа
<element name="paydate" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
//Дата фактического проведения платежа по часам АСР
<element name="localdate" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
//Дата аннулирования платежа (только для status=2)
<element name="canceledate" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
//Идентификатор платежа/номер платежного поручения
<element name="receipt" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
//Комментарий к платежу
<element name="comment" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
</sequence>
</complexType>
```

Объект soapPaymentFull

Объект представляет расширенную информацию о платеже.

```
<complexType name="soapPaymentFull">
<sequence>
//Запись из таблицы Payments
<element name="pay" type="lbapi:soapPayment" minOccurs="1" maxOccurs="1"/>
//Сумма платежа в валюте
<element name="amountcurr" type="xsd:double" minOccurs="1" maxOccurs="1"/>
//Номер счета, если это оплата счета
<element name="ordernum" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID пользователя
<element name="uid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID оператора
<element name="operid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Символ валюты
<element name="curr symb" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Имя пользователя
<element name="uname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Номер договора
```

```
<element name="agrm" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Имя менеджера
<element name="mgr" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Имя оператора
<element name="opername" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Определение функции

Получение списка платежей. На вход подается фильтр, на выходе принимается вектор объектов `soapPaymentFull` или `exception` (исключение).

В фильтре проверяются следующие поля:

<code>if (flt.agrmid)</code>	<code>//ID договора</code>
<code>if (flt.userid)</code>	<code>//ID пользователя</code>
<code>if (!flt.receipt.empty())</code>	<code>//Идентификатор платежа, в данном случае проверяется</code>
	<code>// также flt.mgrid</code>
<code>if (!flt.dtfrom.empty())</code>	<code>//Дата платежа больше или равна заданной</code>
<code>if (!flt.dtto.empty())</code>	<code>//Дата платежа меньше заданной</code>

```
<!-- operation request element -->
<element name="getPayments">
<complexType>
<sequence>
<element name="flt" type="lbapi:soapFilter" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getPaymentsResponse">
<complexType>
<sequence>
<element name="ret" type="lbapi:soapPaymentFull" minOccurs="1" maxOccurs="unbounded"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getPayments.req.xml](#), [api3.getPayments.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
// Входные параметры
// Вызываемая функция
// Фильтр записей:
// agrmid - идентификатор договора
// userid - идентификатор пользователя
// receipt - номер платежного поручения
// mgrid - идентификатор менеджера
// dtfrom - дата платежа больше или равна заданной
// dtto - окончание периода

// Подготовка фильтра и контрольная сумма если используется кэш
$filter = array( ... );
$md5 = $lanbilling->controlSum($filter);
if( false != ($result = $lanbilling->get('getPayments', array('ft' => $filter, 'md5' => $md5))) )
{
    if(!is_array($result)) {
        $result = array($result);
    }

    print_r($result);
}
```

Функция insupdDocument

Добавление или изменение информации о документе.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapDocument

Объект soapDocument документирован в описании функции [getDocuments](#).

Определение функции

Добавление нового или изменение существующего документа. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapDocument, на выходе принимается id записи в таблице documents (doc_id) или exception.

```
<!-- operation request element -->
<element name="insupdDocument">
  <complexType>
    <sequence>
      <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="val" type="lbapi:soapDocument" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="insupdDocumentResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdDocument.req.xml](#), [api3.insupdDocument.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
$struct = array(
  "docid" => ... //идентификатор редактируемой записи или 0 для создания новой
  "ndsabove" =>... // (integer) НДС
  "usergroupid" => ... //генерить для группы пользователей
  "groupid" =>... // генерить для объединения
  "payable" => ... //1 - документ оплачивается
  "clientallowed" => ...// 1 - разрешить просмотр документов созданных по этому шаблону в
  // кабинете клиента
  "hidden" =>... // 1 - скрыть в списках
  "onfly" =>... // 0 - счет, 1 - договор, 2 - отчет, 3 - квитанция
  "name" =>... // Название,
  "doctemplate" =>... // файл шаблона,
  "savepath" => ... //куда сохранять созданный файлы по шаблону,
  "uploadext" =>... // заголовок отправляемый интерфейсом при выгрузке файла созданного
```

```
        //по шаблону
    );

    if( false != $lanbilling->save("insupdDocument", $struct, (((integer)$struct['docid'] == 0) ? true : false),
    array("getDocuments")) ) {

        return true;
    }
}
```

Функция insupdTarifsRasp

Добавление или изменение записи в расписание смены тарифов.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapTarifsRasp

Объект представляет информацию о запланированной смене тарифа (запись из таблицы Tarifs_rasp а так же необязательная текстовая информация о валюте, названии тарифа и т.д.).

```
<complexType name="soapTarifsRasp">
  <sequence>
    //ID записи в таблице
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //ID учетной записи, для которой запланирована смена тарифа
    <element name="vgid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //ID договора
    <element name="agrmid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    //ID объединения, для которой запланирована смена тарифа
    <element name="groupid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //ID пользователя
    <element name="uid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    //ID агента
    <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //ID нового тарифа
    <element name="taridnew" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //ID объединения по тарифу, для которого запланирована смена тарифа
    <element name="taridold" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //Тип агента
    <element name="agenttype" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
  </sequence>
</complexType>
```

```
//ID валюты нового тарифа
  <element name="tarnewcurid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID валюты текущего тарифа
  <element name="taroldcurid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID менеджера, создавшего правило смены (NULL – абонентом)
  <element name="requestby" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Запланированное время смены тарифа
  <element name="changetime" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//логин учетной записи
  <element name="vglogin" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Номер договора
  <element name="agrnum" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Код оплаты из договора
  <element name="code" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Имя пользователя
  <element name="accname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Название агента
  <element name="agentname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Название нового тарифа
  <element name="tarnewname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Название предыдущего тарифа
  <element name="taroldname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Имя менеджера
  <element name="mgrname" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Символ валюты нового тарифа
  <element name="tarnewsymbol" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//Символ валюты предыдущего тарифа
  <element name="taroldsymbol" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
</sequence>
</complexType>
```

Определение функции

Добавление или изменение записи о запланированной смене тарифа. На вход подается флаг insert, (1 – добавление, 0 – изменение) и объект soapTarifsRasp, на выходе принимается id записи в таблице tariffs_rasp (doc_id) или exception. При вызове необходимо указать val.id (ID агента) а также (опционально) один из параметров val.vgid, val.groupid, val.taridold соответственно для назначения смены тарифа по учетной записи, объединению учетных записей или объединению по тарифу. В случае, если опциональные параметры не заданы, смена тарифа будет назначена для всех учетных записей агента.

```
<!-- operation request element -->
<element name="insupdTarifsRasp">
```



```
<complexType>
<sequence>
<element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
<element name="val" type="lbapi:soapTarifsRasp" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="insupdTarifsRaspResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdTarifsRasp.req.xml](#), [api3.insupdTarifsRasp.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования функции представлен в файле php скрипта административного интерфейса (./admin/groups.php в теле функции saveRasp()).

```
$struct = array(
    'recordid' => ... //Номер записи,
    'id' => ... //Номер агента,
    'vgid' => ... //назначить смену для учетной записи,
    'taridold' =>... // назначить смену для объединения по тарифу,
    'groupid' =>... // назначить смену для объединения по учетным записям,
    'taridnew' => ... //Номер нового тарифа,
    'requestby' => ... //кем выполняется запрос: -1 - из клиентского интерфейса, или
    //идентификатор менеджера,
    'changetime' => ... //дата и время для смены тарифа
);

if( false == $lanbilling->save("insupdTarifsRasp", $struct, (((integer)$struct['recordid'] == 0) ? true :
false), "getTarifsRasp")) {
```

```
return false;  
}
```

Функция delTarifsRasp

Удаление записи о назначенной смене тарифа.

Определение функции

На вход подается id записи в таблице tariffs_rasp, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->  
<element name="delTarifsRasp">  
  <complexType>  
    <sequence>  
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
    </sequence>  
  </complexType>  
</element>  
<!-- operation response element -->  
<element name="delTarifsRaspResponse">  
  <complexType>  
    <sequence>  
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
    </sequence>  
  </complexType>  
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delTarifsRasp.req.xml](#), [api3.delTarifsRasp.res.xml](#)

Пример использования функции в коде (PHP библиотека)

На вход подается id записи в таблице tariffs_rasp

// "id" - идентификатор тарифа

```
if( false == $lanbilling->delete("delTarifsRasp", array("id" => ..., "getTarifsRasp")) ) {  
    return false;  
}
```

Функция getTarifsRasp

Получение списка запланированных смен тарифов

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapTarifsRasp

Объект soapTarifsRasp документирован в описании функции insupdTarifsRasp.

Определение функции

Получение списка документов. На вход подается фильтр, на выходе принимается вектор объектов soapTarifsRasp или exception (исключение).

В фильтре проверяются следующие поля:

if (flt.taridprev)	//ID предыдущего тарифа
if (flt.groups)	//ID объединения
if (flt.vgid)	//ID учетной записи
if (flt.agentid)	//ID агента
if (flt.tarid)	//ID нового тарифа
if (flt.recordid)	//ID записи в таблице

```
<!-- operation request element -->
<element name="getTarifsRasp">
  <complexType>
    <sequence>
      <element name="flt" type="lbapi:soapFilter" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getTarifsRaspResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapTarifsRasp" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getTarifsRasp.req.xml](#), [api3.getTarifsRasp.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Параметры в фильтре определяются на выбор в зависимости от того, какое расписание нужно получить

```
$_filter = array();
$_filter['groups'] // Идентификатор объединения учетных записей
$_filter['taridprev'] //Идентификатор тарифа
$_filter['agentid'] //Идентификатор агента

$_md5 = $lanbilling->controlSum($_filter);
if( false != ($result = $lanbilling->get("getTarifsRasp", array("flt" => $_filter, "md5" => $_md5))) ) {

    if(!is_array($result)) {
        $result = array($result);
    }

    print_r($result);
}
```

Функция getCurrencies

Получение списка валют.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapCurrency

Объект представляет информацию о валюте (запись из таблицы Currency).

```
<complexType name="soapCurrency">
  <sequence>
//ID записи
    <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Флаг «Валюта по умолчанию»
    <element name="def" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Символ валюты
    <element name="symbol" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

```
//Наименование валюты
  <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Определение функции

Получение списка валют. На выходе принимается вектор объектов soapCurrency или exception (исключение).

```
<!-- operation request element -->
<element name="getCurrencies">
<complexType>
  <sequence>
  </sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getCurrenciesResponse">
<complexType>
  <sequence>
    <element name="ret" type="lbapi:soapCurrency" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getCurrencies.req.xml](#), [api3.getCurrencies.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
if( false != ($result = $lanbilling->get("getCurrencies")) )
{
  if(!is_array($result)) {
    $result = array($result);
  }

  print_r($result);
}
```

Функция getOptions

Получение списка опций.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapOption

Объект представляет информацию о настройках системы (запись из таблицы Options).

```
<complexType name="soapOption">
  <sequence>
//Название опции
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Описание
    <element name="descr" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Значение
    <element name="value" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

Определение функции

Получение списка настроек. На выходе принимается вектор объектов soapOption или exception (исключение).

```
<!-- operation request element -->
<element name="getOptions">
  <complexType>
    <sequence>
  </sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getOptionsResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapOption" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
```

```
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getOptions.req.xml](#), [api3.getOptions.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
if( false != ($result = $lanbilling->get("getOptions")) )
{
    if(!is_array($result) ) {
        $result = array($result);
    }

    print_r($result);
}
```

Функция getLicense

Получение лицензии.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapLicense

Объект представляет информацию о выданной лицензии.

```
<complexType name="soapLicense">
  <sequence>
    //Название программы
    <element name="progname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    //CD-KEY
    <element name="cdkey" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    //Оператор, на которого выписана лицензия
    <element name="progowner" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    //Дата генерации лицензии
```

```
<element name="gendate" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Действительно до даты
<element name="expire" type="xsd:unsignedLong" minOccurs="0" maxOccurs="1" default="0"/>
//Максимальное количество пользователей в системе
<element name="userlimit" type="xsd:unsignedLong" minOccurs="0" maxOccurs="1" default="0"/>
//Использовать операторскую схему
<element name="useoperators" type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"
default="0"/>
//интеграция с FIDELIO
<element name="usefidelio" type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"
default="0"/>
//модуль E-mail connector
<element name="useec" type="xsd:unsignedLong" minOccurs="0" maxOccurs="1" default="0"/>
//модуль Inventory E-mail connector
<element name="useinventory" type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"
default="0"/>
//интеграция с AV-Desk
<element name="useavdesk" type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"
default="0"/>
//интеграция с CerberCrypt
<element name="usecerber" type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"
default="0"/>
//интеграция с внешними платежными системами
<element name="payments" type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"
default="0"/>
</sequence>
</complexType>
```

Определение функции

Получение информации о действующей лицензии. На выходе принимается вектор объект soapLicense или exception (исключение).

```
<!-- operation request element -->
<element name="getLicense">
<complexType>
<sequence>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getLicenseResponse">
<complexType>
<sequence>
<element name="ret" type="lbapi:soapLicense" minOccurs="1" maxOccurs="unbounded"/>
```



```
</sequence>  
</complexType>  
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getLicense.req.xml](#), [api3.getLicense.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
if( false != ($result = $lanbilling->get("getLicense")) )  
{  
    if(!is_array($result)) {  
        $result = array($result);  
    }  
  
    print_r($result);  
}
```

Функция getRates

Получение курсов валют на дату (период).

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapRate

Объект представляет информацию о валюте и ее курсе на определенную дату.

```
<complexType name="soapRate">  
    <sequence>  
//ID валюты  
        <element name="curid" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
//Флаг «валюта по умолчанию»  
        <element name="def" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
        <element name="modperson" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>  
//курс  
        <element name="rate" type="xsd:double" minOccurs="1" maxOccurs="1"/>  
//дата  
        <element name="date" type="xsd:string" minOccurs="1" maxOccurs="1"/>
```

```
//наименование валюты
  <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//символ валюты
  <element name="symbol" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//дата, по которую устанавливается курс
<element name="date" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Определение функции

Получение списка валют. На вход подается фильтр, на выходе принимается вектор объектов soapRate или exception (исключение). В фильтре проверяются следующие поля:

```
if (flt.dtfrom.empty())           //Курс на текущую дату
else if (!flt.dtto.empty())      //Курс за период
    else                          //Курс на заданную дату
if (flt.curid > -1)              //Фильтр по ID валюты
```

Если курс на текущую или указанную дату не задан, то в объекте на выходе rate = 0, date пустая. Если не задан какой-либо из курсов за период, он не попадает в выходной массив.

```
<!-- operation request element -->
<element name="getRates">
<complexType>
<sequence>
  <element name="flt" type="lbapi:soapFilter" minOccurs="0" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getRatesResponse">
<complexType>
<sequence>
  <element name="ret" type="lbapi:soapRate" minOccurs="1" maxOccurs="unbounded"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getRates.req.xml](#), [api3.getRates.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
$date //Текущая дата
if( false !== ($result = $this->get("getRates", array('flt' => array('dtfrom' => $_date))))){
    if(!is_array($result)) {
        $result = array($result);
    }
    print_r($result);
}
```

Функция getCardSets

Получение списка наборов карт.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapCardset

Объект представляет информацию о наборе карт (запись из таблицы Card_set).

```
<complexType name="soapCardset">
  <sequence>
//ID набора
    <element name="setid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID менеджера
    <element name="createdby" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID валюты
    <element name="curid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID шаблона пользователя
    <element name="acctpl" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Период действия
    <element name="expireperiod" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//кол-во карт в наборе
    <element name="cardscnt" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Описание набора
    <element name="setdescr" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Имя менеджера
    <element name="createdbyname" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
```

```
//Имя валюты
  <element name="currname" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
//Имя пользователя из шаблона
  <element name="accname" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
</sequence>
```

Определение функции

Получение списка наборов карт. На выходе принимается вектор объектов soapCardset или exception (исключение).

```
<!-- operation request element -->
<element name="getCardSets">
  <complexType>
    <sequence>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getCardSetsResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapCardset" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getCardSets.req.xml](#), [api3.getCardSets.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
if( false != ($result = $lanbilling->get("getCardSets")) ) {
    if(!is_array($result)) {
        $result = array($result);
    }
    print_r($result);
}
```

Функция insupdCardSet

Добавление или изменение информации о наборе карт.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapCardsetFull

Объект soapCardsetFull документирован в описании функции getCardSet.

Определение функции

Добавление нового или изменение существующего набора карт. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapCardsetFull, на выходе принимается id записи в таблице card_set (set_id) или exception.

```
<!-- operation request element -->
<element name="insupdCardSet">
  <complexType>
    <sequence>
      <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="val" type="lbapi:soapCardsetFull" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="insupdCardSetResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdCardSet.req.xml](#), [api3.insupdCardSet.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции saveCardset() в файле cardgroups.php административного интерфейса.

Функция delCardSet

Удаление набора карт.

Определение функции

Удаление набора карт. На вход подается id набора, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delCardSet">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="delCardSetResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delCardSet.req.xml](#), [api3.delCardSet.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
(integer)$_POST['cardgrpdel'] //id удаляемого набора карт

if( false == ($lanbilling->delete("delCardSet", array("id" => (integer)$_POST['cardgrpdel'],
    array("getCardSet", "getCardSets")))) ) {
    define("SAVEERROR", true);
}
}
```

Функция genOrders

Функция запускает процесс генерации счетов

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapGenOrderData

Объект представляет набор параметров, используемых при генерации счетов.

```
<complexType name="soapGenOrderData">
<sequence>
//ID оператора. Значение 0xFFFF соответствует «всем операторам»
<element name="oper" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID пользователя, для которого генерируется документ
<element name="uid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID объединения, для которого генерируется документ
<element name="grp" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID группы пользователей, для которой генерируется документ
<element name="ugrp" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID типа документа
<element name="docid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Период, в формате YYYYMM
<element name="period" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Начальный номер
<element name="num" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Дата выставления
<element name="date" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID договора, для которого генерируется документ
<element name="agrmid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Сумма документа (для документов на предоплату)
<element name="summ" type="xsd:double" minOccurs="1" maxOccurs="1"/>
</sequence>
```

```
</complexType>
```

Объект soapFilter

Объект представляет необязательный набор параметров, используемых при генерации счетов. Описание структуры данных приведено выше. Поля из фильтра сохраняются в рамках сессии MySQL согласно примеру:

```
sprintf(cmd, "set @agrmid = '%ld'", ft.agrmid);    into_db(cmd);  
sprintf(cmd, "set @agrmid = '%ld'", ft.agrmid);    into_db(cmd);  
и т.д.
```

Эти переменные могут быть использованы в шаблоне документа.

Определение функции

Генерация списка документов согласно входным параметрам. На вход подаются объекты soapGenOrderData, и soapFilter, на выходе принимается id записи последнего сгенерированного документа в таблице orders (order_id) или exception.

Для документов на предоплату или типа «Квитанция» сумма определяется как val.summ если значение больше нуля или как сумма абонентской платы по всем учетным записям договора

Для документов типа «Бухгалтерский документ» запись в таблице orders создается, если сумма списаний больше нуля. Кроме того, оплаченный документ, подразумевающий оплату, не может быть сгенерирован заново.

Для документов типов «Отчетный документ» и «Пользовательский документ» создается единственная запись в таблице orders, при этом agrm_id в таблице имеет смысл только для «Пользовательского документа», в противном случае он фиктивный.

```
<!-- operation request element -->  
<element name="genOrders">  
  <complexType>  
    <sequence>  
      <element name="val" type="lbapi:soapGenOrderData" minOccurs="1" maxOccurs="1"/>  
      <element name="ft" type="lbapi:soapFilter" minOccurs="0" maxOccurs="1"/>  
    </sequence>  
  </complexType>  
</element>  
<!-- operation response element -->  
<element name="genOrdersResponse">  
  <complexType>  
    <sequence>
```



```
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.genOrders.req.xml](#), [api3.genOrders.res.xml](#)

Пример использования функции в коде (PHP библиотека)

```
$struct = array(
    "period" => date("Ym"),
    "docid" => (integer)$_POST['docid'],
    "num" => 0,
    "date" => 0,
    "summ" => (float)$_POST['orsum'],
    "grp" => 0,
    "ugrp" => 0,
    "uid" => (integer)$_POST['userid'],
    "agrmid" => (integer)$_POST['agrmid'],
    "oper" => 0
);

if( false != ($lanbilling->save("genOrders", $struct)) ) {
    if($lanbilling->saveReturns->ret == 0) {
        echo "{{ success: false, errors: { reason: 'Server returns empty
document ID. Look server logs for detail' } }}";
    }
    else echo "{{ success: true, fileid: " . $lanbilling->saveReturns->ret . " }}";
}
else echo "{{ success: false, errors: { reason: 'There was an error to start generation. Look
server logs for detail' } }}";
} // end createInvoice()
```

Функция genOrderGetFile

Функция запускает процесс генерации счета и отгружает сгенерированный документ клиенту

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapGenOrderData

Объект soapFilter

Объекты и их использование описаны в документации на функцию genOrders.

Объект soapbase64Binary

Объект представляет содержимое сгенерированного файла, подготовленного в соответствии с MIME Base64. Более подробно об использовании можно узнать из документа http://www.cs.fsu.edu/~engelen/soapdoc2.html#tth_sEc10.12

```
<simpleType name="soapbase64Binary">  
  <restriction base="xsd:base64Binary">  
  </restriction>  
</simpleType>
```

Определение функции

Генерация документа согласно входным параметрам. На вход подаются объекты soapGenOrderData, и soapFilter, на выходе принимается содержимое сгенерированного документа или exception.

В целом, алгоритм генерации документа схож с алгоритмом функции soapGenorders, а важным отличием от нее является то, что генерация документа производится только для одного договора, определяемого параметром val.agrmid. ID пользователя, группы или объединения игнорируются.

После успешной генерации данные из файла копируются в выходной объект, а сам файл и запись о нем в таблице orders удаляются.

```
<!-- operation request element -->  
<element name="genOrderGetFile">  
  <complexType>  
    <sequence>  
      <element name="val" type="lbapi:soapGenOrderData" minOccurs="1" maxOccurs="1"/>  
      <element name="flt" type="lbapi:soapFilter" minOccurs="0" maxOccurs="1"/>  
    </sequence>  
  </complexType>  
</element>
```

```
<!-- operation response element -->
<element name="genOrderGetFileResponse">
<complexType>
<sequence>
  <element name="ret" type="lbapi:soapbase64Binary" minOccurs="1" maxOccurs="unbounded"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.genOrderGetFile.req.xml](#), [api3.genOrderGetFile.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции printReceipt() в файле payments.class.php административного интерфейса.

Функция insupdCurrency

Добавление или изменение валюты

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapCurrency

Объект описан в документации на функцию getCurrencies.

Определение функции

Добавление нового или изменение существующей валюты. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapCurrency, на выходе принимается id записи в таблице currency (id) или exception. Т.к. «валюта по умолчанию» в системе может быть только одна, то при установке val.def != 0 этот флаг будет снят с остальных валют.

```
<!-- operation request element -->
<element name="insupdCurrency">
<complexType>
<sequence>
  <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

```
<element name="val" type="lbapi:soapCurrency" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="insupdCurrencyResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdCurrency.req.xml](#), [api3.insupdCurrency.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции saveCurrencyValue() в файле rate_control.php административного интерфейса.

Функция delCurrency

Удаление валюты.

Определение функции

На вход подается id записи в таблице currency, на выходе «1» в случае успеха или exception (исключение). Валюту, которая используется в договорах или тарифах система удалить не даст.

```
<!-- operation request element -->
<element name="delCurrency">
<complexType>
<sequence>
<element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

```
<!-- operation response element -->
  <element name="delCurrencyResponse">
<complexType>
<sequence>
  <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delCurrency.req.xml](#), [api3.delCurrency.res.xml](#)

Пример использования функции в коде (РНР библиотека)

Пример использования приведен в функции `removeCurrency()` в файле `rate_control.php` административного интерфейса.

Функция updCurrencyRate

Установка курса валюты на дату (период).

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapRate

Объект описан в документации на функцию getRates.

Определение функции

Установка курса валюты на дату или на период. На вход подается объект soapRate, на выходе «1» в случае успеха или exception (исключение). Если val.datetill пустая, то курс устанавливается на дату val.date, иначе на период

```
<!-- operation request element -->
<element name="updCurrencyRate">
  <complexType>
    <sequence>
      <element name="val" type="lbapi:soapRate" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="updCurrencyRateResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.updCurrencyRate.req.xml](#), [api3.updCurrencyRate.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции `saveCurrencyRate()` в файле `rate_control.php` административного интерфейса.

Функция `delOrder`

Удаление счета.

Определение функции

Удаление записи из таблицы `orders` и связанного с ней файла на диске. На вход подается `id` набора, на выходе «1» в случае успеха или `exception` (исключение). Если счет оплачен, его удаление невозможно.

```
<!-- operation request element -->
<element name="delOrder">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="delOrderResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delOrder.req.xml](#), [api3.delOrder.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции `RemoveDocument()` в файле `orders.php` административного интерфейса.

Функция `getVgroups`

Получение списка учетных записей.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAddressBrief

Объект представляет информацию об адресе пользователя, описан выше.

Объект soapRate

Объект представляет информацию об учетной записи.

```
<complexType name="soapVgroupBrief">
  <sequence>
//ID учетной записи
    <element name="vgid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID агента
    <element name="id" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID тарифа
    <element name="tarid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID договора
    <element name="agrmd" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//состояние запроса на блокировку
    <element name="blkreq" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//состояние блокировки
    <element name="blocked" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//??
    <element name="canmodify" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID пользователя
    <element name="uid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//тип агента
    <element name="agenttype" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//категория пользователя
    <element name="usercategory" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//баланс договора
    <element name="balance" type="xsd:double" minOccurs="0" maxOccurs="1" default="0.000000"/>
//логин
    <element name="login" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//описание
    <element name="descr" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//№ договора
    <element name="agrnum" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//код оплаты договора
    <element name="code" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//имя пользователя
```



```
<element name="username" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//дата создания
<element name="creationdate" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//дата подключения
<element name="accondate" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//дата отключения
<element name="accoffdate" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//дата блокировки
<element name="blockdate" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//описание агента
<element name="agentdescr" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//описание тарифа
<element name="tarifdescr" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//символ валюты
<element name="symbol" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//физический адрес пользователя
<element name="address" type="lbapi:soapAddressBrief" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
</complexType>
```

Определение функции

Получение списка валют. На вход подается фильтр, на выходе принимается вектор объектов `soapVgroupBrief` или `exception` (исключение). В фильтре проверяются следующие поля:

<code>flt.istemplate</code>	//шаблон/учетка
<code>flt.archive</code>	//удаленные/живые
<code>if (flt.vgid)</code>	//id ү.з
<code>if (flt.tarid)</code>	//id тарифа
<code>if (flt.agentid)</code>	//id агента
<code>if (flt.agrmid)</code>	//id договора
<code>if (flt.userid)</code>	//id пользователя
<code>if (flt.blocked)</code>	//блокировка
<code>if (flt.groups)</code>	//входит в объединение
<code>if (flt.notgroups)</code>	//не входит в объединение
<code>if (flt.category >= 0)</code>	//категория пользователя
<code>if (!flt.name.empty())</code>	//имя пользователя
<code>if (!flt.login.empty())</code>	//логин
<code>if (!flt.descr.empty())</code>	//описание
<code>if (!flt.agrmnum.empty())</code>	//номер договора
<code>if (!flt.code.empty())</code>	//код договора
<code>if (!flt.tardescr.empty())</code>	//описание тарифа

```
<!-- operation request element -->
<element name="getVgroups">
  <complexType>
    <sequence>
      <element name="flt" type="lbapi:soapFilter" minOccurs="0" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getVgroupsResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapVgroupBrief" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getVgroups.req.xml](#), [api3.getVgroups.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции `getVgroups()` в файле `groups.php` административного интерфейса.

Функция `getAddressCountries`

Получение списка стран.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект `soapAddressBrief`

Объект представляет фильтр для работы с адресами.

```
<complexType name="soapAddressFilter">
  <sequence>
    //ID записи
    <element name="recordid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    //ID страны
```

```
<element name="country" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID региона региона
<element name="region" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID района
<element name="area" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID города
<element name="city" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID нас. пункта
<element name="settl" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID улицы
<element name="street" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/> //ID дома

<element name="building" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Уровень привязки (1..7)
<element name="level" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Размер страницы
<element name="pagesize" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Номер страницы
<element name="pgnum" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Флаг «Без данных», для получения кол-ва записей
<element name="nodata" type="xsd:long" minOccurs="0" maxOccurs="1"
default="0"/> //Название
<element name="name" type="xsd:string" minOccurs="0" maxOccurs="1" default="" />
</sequence>
</complexType>
```

Объект soapAddressCountry

Объект представляет информацию о стране (таблица address_country).

```
<complexType name="soapAddressCountry">
<sequence>
//ID записи
<element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Название
<element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Определение функции

Получение списка стран. На вход подается фильтр, на выходе принимается вектор объектов soapAddressCountry или exception (исключение). В фильтре проверяются следующие поля:

if (!flt.name.empty()) //название

```
<!-- operation request element -->
<element name="getAddressCountries">
  <complexType>
    <sequence>
      <element name="flt" type="lbapi:soapAddressFilter" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getAddressCountriesResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapAddressCountry" minOccurs="1"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAddressCountries.req.xml](#), [api3.getAddressCountries.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции `getCountries()` в файле `address.php` административного интерфейса.

Функция `getAddressRegions`

Получение списка регионов.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект `soapAddressFilter` описан в документации (функция `getAddressCountries`)

Объект `soapAddressRegion`

Объект представляет информацию о регионе (таблица `address_region`).

```
<complexType name="soapAddressRegion">
  <sequence>
//ID записи
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID страны
    <element name="country" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Название
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Принятое сокращение
    <element name="shortname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

Определение функции

Получение списка регионов. На вход подается фильтр, на выходе принимается вектор объектов soapAddressRegion или exception (исключение). В фильтре проверяются следующие поля:

```
flt.country          //ID страны, обязательное
if (!flt.name.empty()) //название
```

```
<!-- operation request element -->
<element name="getAddressRegions">
  <complexType>
  <sequence>
    <element name="flt" type="lbapi:soapAddressFilter" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getAddressRegionsResponse">
  <complexType>
  <sequence>
    <element name="ret" type="lbapi:soapAddressRegion" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAddressRegions.req.xml](#), [api3.getAddressRegions.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции `getRegions()` в файле `address.php` административного интерфейса.

Функция `getAddressAreas`

Получение списка районов.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект `soapAddressFilter` описан в документации на функцию `getAddressCountries`

Объект `soapAddressArea`

Объект представляет информацию о районе (таблица `address_area`).

```
<complexType name="soapAddressArea">
  <sequence>
    //ID записи
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //ID региона
    <element name=" region" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //Название
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    //Принятое сокращение
    <element name="shortname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

Определение функции

Получение списка районов. На вход подается фильтр, на выходе принимается вектор объектов `soapAddressArea` или `exception` (исключение). В фильтре проверяются следующие поля:

```
flt.region           //ID региона, обязательное
if (!flt.name.empty()) //название
```

```
<!-- operation request element -->
<element name="getAddressAreas">
  <complexType>
    <sequence>
      <element name="flt" type="lbapi:soapAddressFilter" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getAddressAreasResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapAddressArea" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAddressAreas.req.xml](#), [api3.getAddressAreas.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции `getDistricts()` в файле `address.php` административного интерфейса.

Функция `getAddressCities`

Получение списка городов.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект `soapAddressFilter` описан в документации на функцию `getAddressCountries`

Объект `soapAddressCity`

Объект представляет информацию о городе (таблица `address_city`).

```
<complexType name="soapAddressCity">
```

```
<sequence>
//ID записи
  <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID региона
  <element name="region" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID района
  <element name="area" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Название
  <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Принятое сокращение
  <element name="shortname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Определение функции

Получение списка городов. На вход подается фильтр, на выходе принимается вектор объектов `soapAddressArea` или `exception` (исключение). В фильтре проверяются следующие поля:

```
flt.region          //ID региона, обязательное
if (flt.area)       //ID района
if (!flt.name.empty()) //название
```

```
<!-- operation request element -->
<element name="getAddressCities">
<complexType>
  <sequence>
    <element name="flt" type="lbapi:soapAddressFilter" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getAddressCitiesResponse">
<complexType>
  <sequence>
    <element name="ret" type="lbapi:soapAddressCity" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAddressCities.req.xml](#),

[api3.getAddressCities.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции `getCities()` в файле `address.php` административного интерфейса.

Функция `getAddressSettles`

Получение списка населенных пунктов.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект `soapAddressFilter` описан в документации на функцию `getAddressCountries`

Объект `soapAddressSettle`

Объект представляет информацию о населенном пункте (таблица `address_settl`).

```
<complexType name="soapAddressSettle">
  <sequence>
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/> //ID записи
    <element name="region" type="xsd:long" minOccurs="1" maxOccurs="1"/> //ID региона
    <element name="area" type="xsd:long" minOccurs="1" maxOccurs="1"/> //ID района
    <element name="city" type="xsd:long" minOccurs="1" maxOccurs="1"/> //ID города
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/> //Название
    <element name="shortname" type="xsd:string" minOccurs="1" maxOccurs="1"/> //Принятое
    сокращение
  </sequence>
</complexType>
```

Определение функции

Получение списка населенных пунктов. На вход подается фильтр, на выходе принимается вектор объектов `soapAddressSettle` или `exception` (исключение). В фильтре проверяются следующие поля:

```
flt.region //ID региона, обязательное
if (flt.area) //ID района
if (flt.city) //ID города
```

```
if (!flt.name.empty()) //название
```

```
<!-- operation request element -->
<element name="getAddressSettles">
  <complexType>
    <sequence>
      <element name="flt" type="lbpapi:soapAddressFilter" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getAddressSettlesResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbpapi:soapAddressSettle" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAddressSettles.req.xml](#), [api3.getAddressSettles.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции getSettlements() в файле address.php административного интерфейса.

Функция getAddressStreets

Получение списка улиц.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAddressFilter описан в документации на функцию getAddressCountries

Объект soapAddressStreet

Объект представляет информацию об улице (таблица address_street).

```
<complexType name="soapAddressStreet">
```

```
<sequence>
//ID записи
  <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID региона
  <element name="region" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID района
  <element name="area" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID города
  <element name="city" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID населенного пункта
  <element name="settl" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Почтовый индекс
  <element name="idx" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Название
  <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Принятое сокращение
  <element name="shortname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Определение функции

Получение списка улиц. На вход подается фильтр, на выходе принимается вектор объектов soapAddressStreet или exception (исключение). В фильтре проверяются следующие поля:

```
flt.region          //ID региона, обязательное
if (flt.city)       //ID города
if (flt.settl)      //ID населенного пункта
if (!flt.name.empty()) //название
```

```
<!-- operation request element -->
<element name="getAddressStreets">
<complexType>
  <sequence>
    <element name="flt" type="lbapi:soapAddressFilter" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getAddressStreetsResponse">
<complexType>
  <sequence>
    <element name="ret" type="lbapi:soapAddressStreet" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

```
</sequence>  
</complexType>  
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAddressStreets.req.xml](#), [api3.getAddressStreets.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции `getStreets()` в файле `address.php` административного интерфейса.

Функция `getAddressBuildings`

Получение списка зданий.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект `soapAddressFilter` описан в документации на функцию `getAddressCountries`

Объект `soapAddressBuilding`

Объект представляет информацию о здании (таблица `address_build`).

```
<complexType name="soapAddressBuilding">  
  <sequence>  
    //ID записи  
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
    //ID региона  
    <element name="region" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
    //ID района  
    <element name="area" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
    //ID города  
    <element name="city" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
    //ID населенного пункта  
    <element name="settl" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
    //ID улицы  
    <element name="street" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
    //Почтовый индекс
```

```
<element name="idx" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Название
<element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Принятое сокращение
<element name="shortname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Определение функции

Получение списка зданий. На вход подается фильтр, на выходе принимается вектор объектов `soapAddressBuilding` или `exception` (исключение). В фильтре проверяются следующие поля:

<code>flt.region</code>	//ID региона, обязательное
<code>if (flt.city)</code>	//ID города
<code>if (flt.settl)</code>	//ID населенного пункта
<code>if (flt.street)</code>	//ID улицы
<code>if (!flt.name.empty())</code>	//название

```
<!-- operation request element -->
<element name="getAddressBuildings">
<complexType>
<sequence>
<element name="flt" type="lbapi:soapAddressFilter" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getAddressBuildingsResponse">
<complexType>
<sequence>
<element name="ret" type="lbapi:soapAddressBuilding" minOccurs="1"
maxOccurs="unbounded"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAddressBuildings.req.xml](#), [api3.getAddressBuildings.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции `getBuildings()` в файле `address.php` административного интерфейса.

Функция `getAddressFlats`

Получение списка квартир.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект `soapAddressFilter` описан в документации на функцию `getAddressCountries`

Объект `soapAddressFlat`

Объект представляет информацию о квартире (таблица `address_flat`).

```
<complexType name="soapAddressBuilding">
  <sequence>
//ID записи
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID региона
    <element name="region" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID здания
    <element name="building" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Название
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Принятое сокращение
    <element name="shortname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

Определение функции

Получение списка квартир. На вход подается фильтр, на выходе принимается вектор объектов `soapAddressBuilding` или `exception` (исключение). В фильтре проверяются следующие поля:

```
flt.region          //ID региона, обязательное
if (flt.street)     //ID улицы
if (!flt.name.empty()) //название
```

```
<!-- operation request element -->
<element name="getAddressFlats">
  <complexType>
    <sequence>
      <element name="flt" type="lbapi:soapAddressFilter" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getAddressFlatsResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapAddressFlat" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства:

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции `getFlats()` в файле `address.php` административного интерфейса.

Функция `getAddressMeanings`

Получение списка наименований объектов.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект `soapAddressFilter` описан в документации на функцию `getAddressCountries`

Объект `soapAddressMeaning`

Объект представляет информацию о сокращенном наименовании объекта (таблица `address_meaning`).

```
<complexType name="soapAddressMeaning">
  <sequence>
```

```
//ID записи
  <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Сокращение для региона (флаг)
  <element name="level1" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Сокращение для района (флаг)
  <element name="level2" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Сокращение для города (флаг)
  <element name="level3" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Сокращение для населенного пункта (флаг)
  <element name="level4" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Сокращение для улицы (флаг)
  <element name="level5" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Сокращение для дома (флаг)
  <element name="level6" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Сокращение для квартиры (флаг)
  <element name="level7" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Полное название
  <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Сокращение
  <element name="shortname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Определение функции

Получение списка наименований объектов. На вход подается фильтр, на выходе принимается вектор объектов `soapAddressMeaning` или `exception` (исключение). В фильтре проверяются следующие поля:

`fit.level` //уровень, [0..7]

```
<!-- operation request element -->
<element name="getAddressMeanings">
<complexType>
<sequence>
  <element name="fit" type="lbapi:soapAddressFilter" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getAddressMeaningsResponse">
<complexType>
<sequence>
  <element name="ret" type="lbapi:soapAddressMeaning" minOccurs="1"
```



```
maxOccurs="unbounded"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAddressMeanings.req.xml](#), [api3.getAddressMeanings.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции getAddressMeaning() в файле address.php административного интерфейса.

Функция insupdAddressCountry

Добавление или изменение записи в таблице стран.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAddressCountry

Объект soapAddressCountry документирован в описании функции getAddressCountries

Определение функции

Добавление новой или изменение существующей страны. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapAddressCountry, на выходе принимается id записи в таблице address_country (record_id) или exception.

```
<!-- operation request element -->
<element name="insupdAddressCountry">
<complexType>
<sequence>
<element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
<element name="val" type="lbase:soapAddressCountry" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
```

```
<element name="insupdAddressCountryResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdAddressCountry.req.xml](#), [api3.insupdAddressCountry.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции saveAddressItems() в файле address.php административного интерфейса.

Функция insupdAddressRegion

Добавление или изменение записи в таблице регионов.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAddressRegion

Объект soapAddressRegion документирован в описании функции getAddressregions

Определение функции

Добавление нового или изменение существующего региона. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapAddressRegion, на выходе принимается id записи в таблице address_region (record_id) или exception.

```
<!-- operation request element -->
<element name="insupdAddressRegion">
<complexType>
<sequence>
<element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
<element name="val" type="lbpapi:soapAddressRegion" minOccurs="1" maxOccurs="1"/>
```

```
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="insupdAddressRegionResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdAddressRegion.req.xml](#), [api3.insupdAddressRegion.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции saveAddressItems() в файле address.php административного интерфейса.

Функция insupdAddressArea

Добавление или изменение записи в таблице районов.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAddressArea

Объект soapAddressArea документирован в описании функции getAddressAreas.

Определение функции

Добавление нового или изменение существующего региона. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapAddressArea, на выходе принимается id записи в таблице address_area (record_id) или exception.

```
<!-- operation request element -->
<element name="insupdAddressArea">
  <complexType>
    <sequence>
```

```
<element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
<element name="val" type="lbapi:soapAddressArea" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="insupdAddressAreaResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdAddressArea.req.xml](#), [api3.insupdAddressArea.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции saveAddressItems() в файле address.php административного интерфейса.

Функция insupdAddressCity

Добавление или изменение записи в таблице городов.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAddressCity

Объект soapAddressCity документирован в описании функции getAddressCities

Определение функции

Добавление нового или изменение существующего города. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapAddressCity, на выходе принимается id записи в таблице address_city (record_id) или exception.

```
<!-- operation request element -->
<element name="insupdAddressCity">
```

```
<complexType>
<sequence>
<element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
<element name="val" type="lbase:soapAddressCity" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="insupdAddressCityResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdAddressCitiy.req.xml](#), [api3.insupdAddressCitiy.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции saveAddressItems() в файле address.php административного интерфейса.

Функция insupdAddressSettle

Добавление или изменение записи в таблице населенных пунктов.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAddressSettle

Объект soapAddressSettle документирован в описании функции getAddressSettles

Определение функции

Добавление нового или изменение существующего населенного пункта. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapAddressSettle, на выходе принимается id записи в таблице address_settl (record_id) или exception.

```
<!-- operation request element -->
<element name="insupdAddressSettles">
<!-- operation request element -->
<element name="insupdAddressSettle">
<complexType>
<sequence>
<element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
<element name="val" type="lbapi:soapAddressSettle" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="insupdAddressSettleResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdAddressSettle.req.xml](#), [api3.insupdAddressSettle.res.xml](#)

Функция insupdAddressStreet

Добавление или изменение записи в таблице улиц.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAddressStreet

Объект soapAddressStreet документирован в описании функции getAddressStreets

Определение функции

Добавление новой или изменение существующей улицы. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapAddressStreet, на выходе принимается id записи в таблице address_street (record_id) или exception.

```
<!-- operation request element -->
<element name="insupdAddressStreet">
  <complexType>
    <sequence>
      <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="val" type="lbapi:soapAddressStreet" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="insupdAddressStreetResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: api3.insupdAddressStreets.req.xml, api3.insupdAddressStreets.res.xml

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции saveAddressItems() в файле address.php административного интерфейса.

Функция insupdAddressBuilding

Добавление или изменение записи в таблице зданий.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAddressBuilding

Объект soapAddressBuilding документирован в описании функции getAddressBuildings

Определение функции

Добавление нового или изменение существующего здания. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapAddressBuilding, на выходе принимается id записи в таблице address_building (record_id) или exception.

```
<!-- operation request element -->
<element name="insupdAddressBuilding">
  <complexType>
    <sequence>
      <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="val" type="lbapi:soapAddressBuilding" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="insupdAddressBuildingResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdAddressBuilding.req.xml](#), [api3.insupdAddressBuilding.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции saveAddressItems() в файле address.php административного интерфейса.

Функция insupdAddressFlat

Добавление или изменение записи в таблице квартир.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAddressFlat

Объект soapAddressFlat документирован в описании функции getAddressFlats

Определение функции

Добавление новой или изменение существующей квартиры. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapAddressFlat, на выходе принимается id записи в таблице address_flat (record_id) или exception.

```
<!-- operation request element -->
<element name="insupdAddressFlat">
  <complexType>
    <sequence>
      <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="val" type="lba:soapAddressFlat" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="insupdAddressFlatResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdAddressFlat.req.xml](#), [api3.insupdAddressFlat.res.xml](#)

Пример использования функции в коде (PHP библиотека)

Пример использования приведен в функции saveAddressItems() в файле address.php административного интерфейса.

Функция delAddressCountry

Удаление записи в таблице стран.

Определение функции

Удаление информации о стране. На вход подается id записи, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delAddressCountry">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="delAddressCountryResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delAddressCountry.req.xml](#), [api3.delAddressCountry.res.xml](#)

Функция delAddressRegion

Удаление записи в таблице регионов.

Определение функции

Удаление информации о регионе. На вход подается id записи, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delAddressRegion">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
```

```
</element>
<!-- operation response element -->
<element name="delAddressRegionResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delAddressRegion.req.xml](#), [api3.delAddressRegion.res.xml](#)

Функция delAddressArea

Удаление записи в таблице районов.

Определение функции

Удаление информации о районе. На вход подается id записи, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delAddressArea">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="delAddressAreaResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delAddressArea.req.xml](#), [api3.delAddressArea.res.xml](#)

Функция delAddressCitiy

Удаление записи в таблице городов.

Определение функции

Удаление информации о городе. На вход подается id записи, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delAddressCity">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="delAddressCityResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delAddressCitiy.req.xml](#), [api3.delAddressCitiy.res.xml](#)

Функция delAddressSettle

Удаление записи в таблице населенных пунктов.

Определение функции

Удаление информации о населенном пункте. На вход подается id записи, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delAddressSettle">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="delAddressSettleResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delAddressSettle.req.xml](#), [api3.delAddressSettle.res.xml](#)

Функция delAddressStreet

Удаление записи в таблице улиц.

Определение функции

Удаление информации об улице. На вход подается id записи, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delAddressStreet">
  <complexType>
    <sequence>
```

```
<element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="delAddressStreetResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delAddressStreet.req.xml](#), [api3.delAddressStreet.res.xml](#)

Функция delAddressBuilding

Удаление записи в таблице зданий.

Определение функции

Удаление информации о здании. На вход подается id записи, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delAddressBuilding">
<complexType>
<sequence>
<element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="delAddressBuildingResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

```
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delAddressBuilding.req.xml](#), [api3.delAddressBuilding.res.xml](#)

Функция delAddressFlat

Удаление записи в таблице квартир.

Определение функции

Удаление информации о квартире. На вход подается id записи, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delAddressFlat">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="delAddressFlatResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delAddressFlat.req.xml](#), [api3.delAddressFlat.res.xml](#)

Функция Payment

Проведение платежа

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapPayment

Объект описан в документации на функцию getPayments.

Определение функции

Проведение платежа (вызов функции `PAYMENT`). Используются следующие поля объекта soapPayment:

Receipt

Paydate – если пустая, то используем текущую

Ordered

Agrmid

Modperson

Amount

Comment

На выходе принимается 1 в случае успеха или exception.

```
<!-- operation request element -->
<element name="Payment">
  <complexType>
    <sequence>
      <element name="val" type="lbapi:soapPayment" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="PaymentResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```


Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.Payment.req.xml](#), [api3.Payment.res.xml](#)

Функция ExternCheckPayment

Проверка наличия платежа в системе от имени внешней платежной системы.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapPaymentFull

Объект описан в документации на функцию getPayments.

Определение функции

На вход подается номер платежа, на выходе принимается объект soapPaymentFull или exception.

```
<!-- operation request element -->
<element name="ExternCheckPayment">
  <complexType>
    <sequence>
      <element name="receipt" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="ExternCheckPaymentResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapPaymentFull" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.ExternCheckPayment.req.xml](#), [api3.ExternCheckPayment.res.xml](#)

Функция ExternPayment

Проведение платежа от имени внешней платежной системы.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapPayment

Объект описан в документации на функцию getPayments.

Определение функции

На вход подается id (тип поля, по которому происходит поиск договора), str (значение поля, по которому происходит поиск договора), potexists (флаг «генерировать ошибку в случае, если платеж с таким же номером уже существует») и объект soapPayment val, используемые поля описаны в документации на функцию Payment.

В качестве id могут быть использованы следующие константы:

0	Логин учетной записи
1	Логин пользователя
2	№ телефона
3	IP
4	Название пользователя
5	Номер договора
6	Код договора
7	E-mail пользователя
8	Номер счета
9	ID учетной записи
10	ID пользователя
11	ID договора

На выходе принимается 1 в случае успеха или exception.

```
<!-- operation request element -->  
<element name="ExternPayment">
```

```
<complexType>
  <sequence>
    <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    <element name="str" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <element name="val" type="lbapi:soapPayment" minOccurs="1" maxOccurs="1"/>
    <element name="notexists" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
  </sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="ExternPaymentResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.ExternPayment.req.xml](#), [api3.ExternPayment.res.xml](#)

Функция ExternCancelPayment

Проверка наличия платежа в системе от имени внешней платежной системы.

Определение функции

На вход подается номер платежа, дата отмены платежа и флаг «генерировать ошибку в случае, если платеж с таким же номером уже отменен». На выходе принимается id отмененного платежа или exception.

```
<!-- operation request element -->
<element name="ExternCancelPayment">
  <complexType>
    <sequence>
      <element name="receipt" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <element name="canceldate" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
    </sequence>
  </complexType>
</element>
```

```
<element name="notexists" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="ExternCancelPaymentResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.ExternCancelPayment.req.xml](#), [api3.ExternCancelPayment.res.xml](#)

Функция delVgroup

Перемещение информации об учетной записи в архив.

Определение функции

Перемещение в архив информации об учетной записи пользователя. Данная функция оставляет в базе данных метку на то, что запись помечена на удаление, а окончательную обработку производит соответствующий сетевой агент. После помещения записи в архив все связанные с ней списки (сети, номера, MAC, расписания) удаляются. На вход подается id пользователя, на выходе «1» в случае успеха или exception (исключение).

```
<!-- operation request element -->
<element name="delVgroup">
<complexType>
<sequence>
<element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

```
<!-- operation response element -->
<element name="delVgroupResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.delVgroup.req.xml](#), [api3.delVgroup.res.xml](#)

Функция getPayments

Получение списка платежей

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapPayment

Объект представляет информацию о платеже (запись из таблицы Payments).

```
<complexType name="soapPayment">
  <sequence>
    <element name="recordid" type="xsd:long" minOccurs="1" maxOccurs="1" default="0"/>
      //Идентификатор записи
    <element name="agrmid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      //Идентификатор лицевого счета, на который поступил платеж
    <element name="modperson" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      //Идентификатор менеджера, проводившего платеж
    <element name="currid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      //Идентификатор валюты платежа
    <element name="orderid" type="xsd:long" minOccurs="1" maxOccurs="1" default="0"/>
      //Идентификатор счета в `orders`, если проводилась оплата счета
    <element name="status" type="xsd:long" minOccurs="1" maxOccurs="1" default="0"/>
      //Статус платежа: 0-проведен, 1-проведен и подтвержден сверкой (при загрузке фала
сверки), 2-платеж аннулирован
    <element name="amount" type="xsd:double" minOccurs="1" maxOccurs="1"/> //Величина
платежа в валюте лицевого счета
    <element name="paydate" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
```

```
//Дата платежа
<element name="localdate" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
//Дата фактического проведения платежа по часам АСР
<element name="canceledate" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
//Дата аннулирования платежа (только для status=2)
<element name="receipt" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
//Идентификатор платежа/номер платежного поручения
<element name="comment" type="xsd:string" minOccurs="1" maxOccurs="1" default=""/>
//Комментарий к платежу
</sequence>
</complexType>
```

Объект soapPaymentFull

Объект представляет расширенную информацию о платеже.

```
<complexType name="soapPaymentFull">
<sequence>
//Запись из таблицы Payments
<element name="pay" type="xmlapi:soapPayment" minOccurs="1" maxOccurs="1"/>
//Сумма платежа в валюте
<element name="amountcurr" type="xsd:double" minOccurs="1" maxOccurs="1"/>
//Номер счета, если это оплата счета
<element name="ordernum" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID пользователя
<element name="uid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID оператора
<element name="operid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Символ валюты
<element name="curr symb" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Имя пользователя
<element name="uname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Номер договора
<element name="agrm" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Имя менеджера
<element name="mgr" type="xsd:string" minOccurs="1" maxOccurs="1"/>
//Имя оператора
<element name="opername" type="xsd:string" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
```

Определение функции

Получение списка платежей. На вход подается фильтр, на выходе принимается вектор объектов `soapPaymentFull` или `exception` (исключение).

В фильтре проверяются следующие поля:

```
if (flt.agrmid)           //ID договора
if (flt.userid)          //ID пользователя
if (!flt.receipt.empty()) //Идентификатор платежа, в данном случае
    проверяется также flt.mgrid
if (!flt.dtfrom.empty()) //Дата платежа больше или равна заданной
if (!flt.dtto.empty())   //Дата платежа меньше заданной
```

```
<!-- operation request element -->
<element name="getPayments">
  <complexType>
    <sequence>
      <element name="flt" type="xmlapi:soapFilter" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getPaymentsResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xmlapi:soapPaymentFull" minOccurs="1"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: `api3.getPayments.req.xml`, `api3.getPayments.res.xml`

Функция `getAccountsAddonsStaff`

Получение списка допустимых значений для доп. полей пользователя типа "список".

Описание объектов, поступающих в качестве параметров и возвращаемых функций

Объект soapAccountsAddonsStaff

Объект представляет информацию о Значениях для доп. полей пользователя типа "список" (таблица accounts_addons_staff).

```
<complexType name="soapAccountsAddonsStaff">
  <sequence>
    // Внутренний номер записи
    <element name="idx" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    // Уникальное имя поля
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    // Значение
    <element name="value" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

Определение функции

Получение списка Значений для доп. полей пользователя типа "список". На вход подается фильтр, на выходе принимается вектор объектов soapAccountsAddonsStaff или exception (исключение). В фильтре проверяются следующие поля:

```
if (!flt.name.empty()) //имя поля
```

```
<!-- operation request element -->
<element name="getAccountsAddonsStaff">
  <complexType>
    <sequence>
      <element name="flt" type="lbapi:soapFilter" minOccurs="0" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getAccountsAddonsStaffResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapAccountsAddonsStaff" minOccurs="1"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```



```
</sequence>  
</complexType>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства [api3.getAccountsAddonsStaff.req.xml](#), [api3.getAccountsAddonsStaff.res.xml](#)

Функция getAccountsAddonsSet

Получение списка Дополнительных полей пользователей.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAccountsAddonsSet

Объект представляет информацию о Дополнительных полях пользователей (таблица `accounts_addons_set`).

```
<complexType name="soapAccountsAddonsSet">  
  <sequence>  
    // Уникальное имя поля. Допустимые символы: a-zA-Z0-9_  
    <element name="type" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
    // Описание поля  
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>  
    // Тип поля: 0 - текстовое, 1 - список значений  
    <element name="descr" type="xsd:string" minOccurs="1" maxOccurs="1"/>  
  </sequence>  
</complexType>
```

Объект soapAccountsAddonsSetFull

Объект представляет расширенную информацию о Дополнительных полях пользователя.

```
<complexType name="soapAccountsAddonsSetFull">  
  <sequence>  
    <element name="addon" type="lbapi:soapAccountsAddonsSet" minOccurs="1" maxOccurs="1"/>  
      //Доп. поле  
    <element name="staff" type="lbapi:soapAccountsAddonsStaff" minOccurs="0"  
maxOccurs="unbounded"/>    //Возможные значения для типа поля 1  
  </sequence>
```

```
</complexType>
```

Определение функции

Получение Дополнительных полей пользователя. На вход подается фильтр, на выходе принимается вектор объектов soapAccountsAddonsSetFull или exception (исключение). В фильтре проверяются следующие поля:

```
if (!flt.name.empty()) // имя поля
```

```
<!-- operation request element -->
<element name="getAccountsAddonsSet">
  <complexType>
    <sequence>
      <element name="flt" type="lbapi:soapFilter" minOccurs="0" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getAccountsAddonsSetResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapAccountsAddonsSetFull" minOccurs="1"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства [api3.getAccountsAddonsSet.req.xml](#), [api3.getAccountsAddonsSet.res.xml](#)

Функция insupdAccountsAddonsSet

Добавление или изменение записи из списка дополнительных полей пользователей.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAccountsAddonsSetFull

Объект описан в документации на функцию getAccountsAddonsSet.

Определение функции

Добавление нового или изменение существующего дополнительного поля пользователя. На вход подается флаг isInsert, (1 – добавление, 0 – изменение) и объект soapAccountsAddonsSetFull, на выходе принимается id записи в таблице vgroups_addons_set (name) или exception.

```
<!-- operation request element -->
<element name="insupdAccountsAddonsSet">
  <complexType>
    <sequence>
      <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="val" type="lbapi:soapAccountsAddonsSetFull" minOccurs="1"
maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="insupdAccountsAddonsSetResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства [api3.insupdAccountsAddonsSet.req.xml](#), [api3.insupdAccountsAddonsSet.res.xml](#)

Функция getAgreementsAddonsStaff

Получение списка допустимых значений для доп. полей договора типа "список".

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAgreementsAddonsStaff

Объект представляет информацию о Значениях для доп. полей договора типа "список" (таблица agreements_addons_staff).

```
<complexType name="soapAgreementsAddonsStaff">
  <sequence>
    // Внутренний номер записи
    <element name="idx" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    // Уникальное имя поля
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    // Значение
    <element name="value" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

Определение функции

Получение списка Значениях для доп. полей договора типа "список". На вход подается фильтр, на выходе принимается вектор объектов soapAgreementsAddonsStaff или exception (исключение). В фильтре проверяются следующие поля:

```
if (!flt.name.empty()) //имя поля
<!-- operation request element -->
<element name="getAgreementsAddonsStaff">
  <complexType>
    <sequence>
      <element name="flt" type="lbapi:soapFilter" minOccurs="0" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
```

```
<element name="getAgreementsAddonsStaffResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapAgreementsAddonsStaff" minOccurs="1"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAgreementsAddonsStaff.req.xml](#), [api3.getAgreementsAddonsStaff.res.xml](#)

Функция getAgreementsAddonsSet

Получение списка Дополнительных полей договора.

Описание объектов, поступающих в качестве параметров и возвращаемых функций

Объект soapAgreementsAddonsSet

Объект представляет информацию о Дополнительных полях пользователей (таблица agreements_addons_set).

```
<complexType name="soapAgreementsAddonsSet">
  <sequence>
    // Уникальное имя поля. Допустимые символы: a-zA-Z0-9-_-
    <element name="type" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    // Описание поля
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    // Тип поля: 0 - текстовое, 1 - список значений
    <element name="descr" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

Объект soapAgreementsAddonsSetFull

Объект представляет расширенную информацию о Дополнительных полях пользователя.

```
<complexType name="soapAgreementsAddonsSetFull">
  <sequence>
    <element name="addon" type="Ibapi:soapAgreementsAddonsSet" minOccurs="1"
maxOccurs="1"/> //Доп. поле
    <element name="staff" type="Ibapi:soapAgreementsAddonsStaff" minOccurs="0"
maxOccurs="unbounded"/> //Возможные значения для типа поля 1
  </sequence>
</complexType>
```

Определение функции

Получение Дополнительных полей договора. На вход подается фильтр, на выходе принимается вектор объектов soapAgreementsAddonsSetFull или exception (исключение). В фильтре проверяются следующие поля:

```
if (!flt.name.empty()) // имя поля
```

```
<!-- operation request element -->
<element name="getAgreementsAddonsSet">
  <complexType>
    <sequence>
      <element name="flt" type="Ibapi:soapFilter" minOccurs="0" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getAgreementsAddonsSetResponse">
  <complexType>
    <sequence>
      <element name="ret" type="Ibapi:soapAgreementsAddonsSetFull" minOccurs="1"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getAgreementsAddonsSet.req.xml](#),
[api3.getAgreementsAddonsSet.res.xml](#)

Функция insupdAgreementsAddonsSet

Добавление или изменение записи из списка дополнительных полей договора.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapAgreementsAddonsSetFull

Объект описан в документации на функцию `getAgreementsAddonsSet`.

Определение функции

Добавление нового или изменение существующего дополнительного поля договора. На вход подается флаг `isInsert`, (1 – добавление, 0 – изменение) и объект `soapAgreementsAddonsSetFull`, на выходе принимается `id` записи в таблице `agreements_addons_set` (name) или exception.

```
<!-- operation request element -->
<element name="insupdAgreementsAddonsSet">
  <complexType>
    <sequence>
      <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="val" type="lbapi:soapAgreementsAddonsSetFull" minOccurs="1"
maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="insupdAgreementsAddonsSetResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdAgreementsAddonsSet.req.xml](#), [api3.insupdAgreementsAddonsSet.res.xml](#)

Функция `getVgroupsAddonsStaff`

Получение списка допустимых значений для доп. полей учетной записи типа "список".

Описание объектов, поступающих в качестве параметров и возвращаемых функций

Объект `soapVgroupsAddonsStaff`

Объект представляет информацию о Значениях для дополнительных полей учетной записи типа "список" (таблица `vgroups_addons_staff`).

```
<complexType name="soapVgroupsAddonsStaff">
  <sequence>
    <element name="idx" type="xsd:long" minOccurs="1" maxOccurs="1"/> // Внутренний номер
    записи
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/> //
    Уникальное имя поля
    <element name="value" type="xsd:string" minOccurs="1" maxOccurs="1"/> // Значение
  </sequence>
</complexType>
```

Определение функции

Получение списка Значений для доп. полей пользователя типа "список". На вход подается фильтр, на выходе принимается вектор объектов `soapVgroupsAddonsStaff` или `exception` (исключение). В фильтре проверяются следующие поля:

```
if (!flt.name.empty()) //имя поля
```

```
<!-- operation request element -->
<element name="getVgroupsAddonsStaff">
```



```
<complexType>
  <sequence>
    <element name="fit" type="lbapi:soapFilter" minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="getVgroupsAddonsStaffResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapVgroupsAddonsStaff" minOccurs="1"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getVgroupsAddonsStaff.req.xml](#), [api3.getVgroupsAddonsStaff.res.xml](#)

Функция getVgroupsAddonsSet

Получение списка Дополнительных полей пользователей.

Описание объектов, поступающих в качестве параметров и возвращаемых функций

Объект soapVgroupsAddonsSet

Объект представляет информацию о Дополнительных полях учетной записи (таблица `vgroups_addons_set`).

```
<complexType name="soapVgroupsAddonsSet">
  <sequence>
    <element name="type" type="xsd:long" minOccurs="1" maxOccurs="1"/> // Уникальное имя
поля. Допустимые символы: a-zA-Z0-9_
    <element name="agentid" type="xsd:long" minOccurs="1" maxOccurs="1"/> //ID агента
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/> // Описание
поля
    <element name="descr" type="xsd:string" minOccurs="1" maxOccurs="1"/> // Тип поля:
0 - текстовое, 1 - список значений
```

```
</sequence>  
</complexType>
```

Объект soapVgroupsAddonsSetFull

Объект представляет расширенную информацию о дополнительных полях учетной записи.

```
<complexType name="soapVgroupsAddonsSetFull">  
<sequence>  
  <element name="addon" type="lbapi:soapVgroupsAddonsSet" minOccurs="1" maxOccurs="1"/>  
    //Доп. поле  
  <element name="staff" type="lbapi:soapVgroupsAddonsStaff" minOccurs="0"  
maxOccurs="unbounded"/>    //Возможные значения для типа поля 1  
</sequence>  
</complexType>
```

Определение функции

Получение Дополнительных полей учетной записи. На вход подается фильтр, на выходе принимается вектор объектов soapVgroupsAddonsSetFull или exception (исключение). В фильтре проверяются следующие поля:

```
if (!flt.name.empty()) // имя поля
```

```
<!-- operation request element -->  
<element name="getVgroupsAddonsSet">  
<complexType>  
<sequence>  
  <element name="flt" type="lbapi:soapFilter" minOccurs="0" maxOccurs="1"/>  
</sequence>  
</complexType>  
</element>  
<!-- operation response element -->  
<element name="getVgroupsAddonsSetResponse">  
<complexType>  
<sequence>  
  <element name="ret" type="lbapi:soapVgroupsAddonsSetFull" minOccurs="1"  
maxOccurs="unbounded"/>  
</sequence>
```

```
</complexType>  
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getVgroupsAddonsSet.req.xml](#)[|](#)[api3.getVgroupsAddonsSet.res.xml](#)

Функция insupdVgroupsAddonsSet

Добавление или изменение записи из списка дополнительных полей пользователей.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapVgroupsAddonsSetFull

Объект описан в документации на функцию `getVgroupsAddonsSet`.

Определение функции

Добавление нового или изменение существующего дополнительного поля пользователя. На вход подается флаг `isInsert`, (1 – добавление, 0 – изменение) и объект `soapVgroupsAddonsSetFull`, на выходе принимается `id` записи в таблице `vgroups_addons_set` (`name`) или `exception`.

```
<!-- operation request element -->  
<element name="insupdVgroupsAddonsSet">  
  <complexType>  
    <sequence>  
      <element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>  
      <element name="val" type="l1bapi:soapVgroupsAddonsSetFull" minOccurs="1" maxOccurs="1"/>  
    </sequence>  
  </complexType>  
</element>  
<!-- operation response element -->  
<element name="insupdVgroupsAddonsSetResponse">  
  <complexType>
```

```
<sequence>  
  <element name="ret" type="xsd:string" minOccurs="1" maxOccurs="1"/>  
</sequence>  
</complexType>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.insupdVgroupsAddonsSet.req.xml](#), [api3.insupdVgroupsAddonsSet.res.xml](#)

Функция blkClientVgroup

Блокировка учетной записи клиентом.

Определение функции

Отправка запроса на блокировку учетной записи клиентом. Окончательная обработка запроса на блокировку производится агентом асинхронно в соответствии с настройкой, заданной в settings.timer.

На вход подаются id учетной записи, тип блокировки blk (разрешены значения 2-пользователем, 3-администратором) и доп. Параметр для управления состоянием state, который может принимать значения «on», «off» и «» (пустая строка)

На выходе получаем «1» в случае успеха или исключение.

Исключения генерируются в случае, если учетная запись не найдена или если учетная запись находится в следующих состояниях блокировки: (1-заблокирована по балансу, 3-администратором, 4-по балансу (активная блокировка), 5-лимит трафика, 10-уч. запись отключена)

При blk = 2 параметр state определяет, в каком состоянии блокировки учетной записи будет находиться после обработки агентом: если «on», то 2-пользователем, если «off», то 0-уч. запись активна, иначе блокировка срабатывает как триггер 0 <->2

При blk = 3 параметр state считается «on», т.е. перевести себя в состояние административной блокировки клиент сможет только один раз и разблокировать его должен будет администратор.

```
<!-- operation request element -->  
<element name="blkClientVgroup">  
  <complexType>  
    <sequence>
```

```
<element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
<element name="blk" type="xsd:long" minOccurs="1" maxOccurs="1"/>
<element name="state" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="blkClientVgroupResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.blkClientVgroup.req.xml](#), [api3.blkClientVgroup.res.xml](#)

Функция blkVgroup

Блокировка учетной записи администратором.

Определение функции

Отправка запроса на блокировку учетной записи администратором. Окончательная обработка запроса на блокировку производится агентом асинхронно в соответствии с настройкой, заданной в settings.timer.

На вход подаются id учетной записи, тип блокировки blk (разрешены значения 2-пользователем, 3-администратором, 5-лимит трафика, 10-уч. запись отключена) и доп. Параметр для управления состоянием state, который может принимать значения «on», «off» и «» (пустая строка)

На выходе получаем «1» в случае успеха или исключение.

Исключения генерируются в случае, если учетная запись не найдена. Параметр state определяет, в каком состоянии блокировки учетная запись будет находиться после обработки агентом: если «on», то в состоянии, заданном blk, если «off», то 0-уч. запись активна, иначе блокировка срабатывает как триггер 0 <-> blk

```
<!-- operation request element -->
<element name="blkVgroup">
  <complexType>
    <sequence>
      <element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="blk" type="xsd:long" minOccurs="1" maxOccurs="1"/>
      <element name="state" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="blkVgroupResponse">
  <complexType>
    <sequence>
      <element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: api3.blkVgroup.req.xml, api3.blkVgroup.res.xml

Функция insupdUsboxService

Добавление или изменение услуги.

Описание объектов, поступающих в качестве параметров и возвращаемых функций

Объект soapUsboxService

```
<complexType name="soapUsboxService">
  <sequence>
    //ID услуги
    <element name="servid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
    //ID учетной записи
    <element name="vgid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
    //ID тарифа
    <element name="tarid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
```

```
//ID категории тарифа
  <element name="catidx" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//Флаг необходимости тарификации записи »
  <element name="needcalc" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//Флаг проверки на то, что услуга не может быть назначена дважды к одной учетке
  <element name="noduplicate" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
// В каком количестве оказана услуга (множитель для стоимости услуги из тарифа)
  <element name="mul" type="xsd:double" minOccurs="1" maxOccurs="1"/>
//ID менеджера, создавшего услугу, если услуга назначена клиентом – пустая строка
  <element name="personid" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//комментарий к услуге
  <element name="comment" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
//дата создания услуги
  <element name="created" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
  <element name="timefrom" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
// Дата начала оказания услуги
  <element name="timeto" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
</sequence>
</complexType>
```

Определение функции

Добавление нового или изменение существующей услуги. На вход подается флаг `isInsert`, (1 – добавление, 0 – изменение) и объект `soapUsboxService`, на выходе принимается `id` записи в таблице `usbox_services` (`serv_id`) или `exception`.

При вызове проверяются следующие условия:

- услуга должна принадлежать тарифу, который назначен или запланирован для учетной записи.
- период действия услуги не должен попадать в «заблокированный период» (`mysql>select value from options where name = 'lock_period'`)
- если на входе стоит флаг `noduplicate`, то проверяется, что у клиента нет действующей услуги с теми же `tarid` и `catidx`

На добавление услуги необходимо заполнить следующие поля в объекте: `vgid`, `tarid`, `catidx`, `personid`, `mul`, `timefrom`, `timeto`

На изменение услуги требуются `personid`, `mul`, `timefrom`, `timeto`, `servid`

```
<!-- operation request element -->
<element name="insupdUsboxService">
  <complexType>
  <sequence>
```

```
<element name="isInsert" type="xsd:long" minOccurs="1" maxOccurs="1"/>
<element name="val" type="lbapi:soapUsboxService" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
<element name="insupdUsboxServiceResponse">
<complexType>
<sequence>
<element name="ret" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: api3.insupdUsboxService.req.xml, api3.insupdUsboxService.res.xml

Функция getTarCategories

Получение списка категорий тарифа.

Описание объектов, поступающих в качестве параметров и возвращаемых функцией

Объект soapTarCategory

Объект представляет информацию о валюте и ее курсе на определенную дату.

```
<complexType name="soapTarCategory">
<sequence>
//ID тарифа
<element name="tarid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
//ID каталога
<element name="catid" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
//ID категории
<element name="catidx" type="xsd:long" minOccurs="1" maxOccurs="1"/>
// Величина услуги, включенной в абон. плату (Мб или мин)
<element name="includes" type="xsd:long" minOccurs="1" maxOccurs="1"/>
// Трафик по категории включен в "общий счетчик" (флаг). Для USBox - тип услуги: 0-разовая,1-
периодич. с ежемесяч. списанием,2-ежеднев.,3-ежеднев. равными долями
```



```
<element name="common" type="xsd:long" minOccurs="1" maxOccurs="1"/>
// Приоритет применения скидок: 0-мин стоимость, 1-по объему, 2-по времени, 3-мак стоимость
<element name="disprior" type="xsd:long" minOccurs="1" maxOccurs="1"/>
// Категория удалена (флаг)
<element name="archive" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
// Идентификатор оператора
<element name="operid" type="xsd:long" minOccurs="1" maxOccurs="1"/>
// Порог нетарифицируемого звонка (сек)
<element name="freeseconds" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
// Минимальная длительность звонка после округления (сек)
<element name="minchargedur" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
// Округлять с точностью до... (сек)
<element name="roundseconds" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
// Сервис, описываемый категорией, включен в подписку по умолчанию
<element name="enabled" type="xsd:long" minOccurs="0" maxOccurs="1" default="1"/>
// Возможность назначать/отменять подписку на сервис из клиентского интерфейса
<element name="available" type="xsd:long" minOccurs="0" maxOccurs="1" default="0"/>
// Цена за факт звонка. Для USBox - цена услуги для уч. записи в заблокированном состоянии.
<element name="permaabove" type="xsd:double" minOccurs="0" maxOccurs="1"
default="0.000000"/>
// Стоимость услуги в данной категории (в валюте тарифа)
<element name="above" type="xsd:double" minOccurs="1" maxOccurs="1"/>
// Описание категории
<element name="descr" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
// Код для связи с внешней системой
<element name="uuid" type="xsd:string" minOccurs="0" maxOccurs="1" default=""/>
</sequence>
</complexType>
```

Определение функции

Получение списка категорий тарифа. На вход подается id тарифа, на выходе принимается вектор объектов `soapTarCategory` или `exception` (исключение).

```
<!-- operation request element -->
<element name="getTarCategories">
<complexType>
<sequence>
<element name="id" type="xsd:long" minOccurs="1" maxOccurs="1"/>
</sequence>
</complexType>
</element>
<!-- operation response element -->
```

```
<element name="getTarCategoriesResponse">
  <complexType>
    <sequence>
      <element name="ret" type="lbapi:soapTarCategory" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Пример использования функции в приведено в следующих документах, являющихся приложением данного руководства: [api3.getTarCategories.req.xml](#), [api3.getTarCategories.res.xml](#)

ПРИЛОЖЕНИЕ №1

Примеры использования функций API

В примерах описаний функций API ПРИЛОЖЕНИЯ №1, дано содержание тега <SOAP-ENV:Body>, который входит в стандартный для всех функций тэг <SOAP-ENV:Envelope>, так как это представлено ниже:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:lbapi="urn:api3">
  <SOAP-ENV:Body>
    ...
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

api3.getAccounts.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getAccounts>
    <flt>
      <agrmid>0</agrmid>
      <userid>0</userid>
      <vgid>0</vgid>
      <tarid>0</tarid>
      <taridprev>0</taridprev>
      <operid>0</operid>
      <agentid>0</agentid>
      <recordid>0</recordid>
      <istemplate>0</istemplate>
      <ugroups>0</ugroups>
      <groups>0</groups>
      <notgroups>0</notgroups>
      <direction>0</direction>
      <pgsize>0</pgsize>
```

```
<pgnum>0</pgnum>
<setid>0</setid>
<activated>0</activated>
<personid>0</personid>
<archive>0</archive>
<blocked>0</blocked>
<catid>0</catid>
<nodata>0</nodata>
<port>0</port>
<proto>0</proto>
<asnum>0</asnum>
<repnum>0</repnum>
<repdetail>0</repdetail>
<mgrid>0</mgrid>
<onfly>0</onfly>
<docid>0</docid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<common>0</common>
<deviceid>0</deviceid>
<groupid>0</groupid>
<showdefault>0</showdefault>
<defaultonly>0</defaultonly>
<durfrom>0</durfrom>
<durto>0</durto>
<payed>-1</payed>
<addresstype>1</addresstype>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<receipt></receipt>
<serno></serno>
<cardkey></cardkey>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<numfrom></numfrom>
<numto></numto>
<name></name>
<login></login>
<vglogin></vglogin>
<descr></descr>
<tardescr></tardescr>
<agrmnum></agrmnum>
<email></email>
```

```
<phone></phone>
<inn></inn>
<ip></ip>
<dstip></dstip>
<telstaff></telstaff>
<code></code>
<telnum></telnum>
<namesound></namesound>
<addresscode></addresscode>
</flt>
</lbapi:getAccounts>
</SOAP-ENV:Body>
```

api3.getAccounts.res.xml

```
<SOAP-ENV:Body>
<lbapi:getAccountsResponse>
<ret>
<account>
<uid>0</uid>
<vgcnt>0</vgcnt>
<type>0</type>
<login></login>
<name></name>
<descr></descr>
<email></email>
<phone></phone>
</account>
<usergroups>
<usergroup>
<groupid>0</groupid>
<promiseallow>0</promiseallow>
<promiserent>0</promiserent>
<promisetill>0</promisetill>
<promisemax>0.000000</promisemax>
<promisemin>0.000000</promisemin>
<promiselimit>0.000000</promiselimit>
<name></name>
<description></description>
</usergroup>
<uids>
<val>0</val>
</uids>
<usercnt>0</usercnt>
```

```
<fread>0</fread>
<fwrite>0</fwrite>
</usergroups>
<agreements>
  <agrmid>0</agrmid>
  <defcurr>0</defcurr>
  <balance>0.0</balance>
  <number></number>
  <symbol></symbol>
  <code></code>
</agreements>
<addresses>
  <type>0</type>
  <code></code>
  <address></address>
</addresses>
</ret>
</lbapi:getAccountsResponse>
</SOAP-ENV:Body>
```

api3.getStat.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getStat>
    <flt>
      <agrmid>0</agrmid>
      <userid>0</userid>
      <vgid>0</vgid>
      <tarid>0</tarid>
      <taridprev>0</taridprev>
      <operid>0</operid>
      <agentid>0</agentid>
      <recordid>0</recordid>
      <istemplate>0</istemplate>
      <ugroups>0</ugroups>
      <groups>0</groups>
      <notgroups>0</notgroups>
      <direction>0</direction>
      <pgsize>0</pgsize>
      <pgnum>0</pgnum>
      <setid>0</setid>
      <activated>0</activated>
      <personid>0</personid>
      <archive>0</archive>
```

```
<blocked>0</blocked>
<catid>0</catid>
<nodata>0</nodata>
<port>0</port>
<proto>0</proto>
<asnum>0</asnum>
<repnum>0</repnum>
<repdetail>0</repdetail>
<mgrid>0</mgrid>
<onfly>0</onfly>
<docid>0</docid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<common>0</common>
<deviceid>0</deviceid>
<groupid>0</groupid>
<showdefault>0</showdefault>
<defaultonly>0</defaultonly>
<durfrom>0</durfrom>
<durto>0</durto>
<payed>-1</payed>
<addresstype>1</addresstype>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<receipt></receipt>
<serno></serno>
<cardkey></cardkey>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<numfrom></numfrom>
<numto></numto>
<name></name>
<login></login>
<vglogin></vglogin>
<descr></descr>
<tardescr></tardescr>
<agrmnum></agrmnum>
<email></email>
<phone></phone>
<inn></inn>
<ip></ip>
<dstip></dstip>
<telstaff></telstaff>
```

```
<code></code>
<telnum></telnum>
<namesound></namesound>
<addresscode></addresscode>
</flt>
<ord>
  <name></name>
  <ascdesc>0</ascdesc>
</ord>
</lbapi:getStat>
</SOAP-ENV:Body>
```

api3.getStat.res.xml

```
<SOAP-ENV:Body>
  <lbapi:getStatResponse>
    <ret>
      <rows>0</rows>
      <cols>0</cols>
      <total>0</total>
      <names>
        <val></val>
      </names>
      <data>
        <val></val>
      </data>
    </ret>
  </lbapi:getStatResponse>
</SOAP-ENV:Body>
```

api3.getAccount.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getAccount>
    <id>0</id>
  </lbapi:getAccount>
</SOAP-ENV:Body>
```

api3.getAccount.res.xml

```
<SOAP-ENV:Body>
  <lbapi:getAccountResponse>
```



```
<ret>
<account>
  <uid>0</uid>
  <ipaccess>0</ipaccess>
  <billdelivery>0</billdelivery>
  <category>0</category>
  <type>0</type>
  <oksm>0</oksm>
  <templ>0</templ>
  <wrongactive>0</wrongactive>
  <archive>0</archive>
  <login></login>
  <pass></pass>
  <descr></descr>
  <name></name>
  <phone></phone>
  <fax></fax>
  <email></email>
  <bankname></bankname>
  <branchbankname></branchbankname>
  <treasuryname></treasuryname>
  <treasuryaccount></treasuryaccount>
  <bik></bik>
  <settl></settl>
  <corr></corr>
  <kpp></kpp>
  <inn></inn>
  <ogrn></ogrn>
  <okpo></okpo>
  <okved></okved>
  <gendiru></gendiru>
  <glbuhgu></glbuhgu>
  <kontperson></kontperson>
  <actonwhat></actonwhat>
  <passsernum></passsernum>
  <passno></passno>
  <passissuedate></passissuedate>
  <passissuedep></passissuedep>
  <passissueplace></passissueplace>
  <birthdate></birthdate>
  <birthplace></birthplace>
  <lastmoddate></lastmoddate>
  <wrongdate></wrongdate>
  <okato></okato>
  <uuid></uuid>
```

```
<swift></swift>
<kio></kio>
<bicbei></bicbei>
<iban></iban>
<bankcorr></bankcorr>
<bankcorrcode></bankcorrcode>
<bankcorraccount></bankcorraccount>
<currency></currency>
<resident>0</resident>
</account>
<usergroups>
<usergroup>
<groupid>0</groupid>
<promiseallow>0</promiseallow>
<promiserent>0</promiserent>
<promisetill>0</promisetill>
<promisemax>0.000000</promisemax>
<promisemin>0.000000</promisemin>
<promiselimit>0.000000</promiselimit>
<name></name>
<description></description>
</usergroup>
<uids>
<val>0</val>
</uids>
<usercnt>0</usercnt>
<fread>0</fread>
<fwrite>0</fwrite>
</usergroups>
<addresses>
<type>0</type>
<code></code>
<address></address>
</addresses>
<agreements>
<agrmid>0</agrmid>
<uid>0</uid>
<operid>0</operid>
<curid>0</curid>
<bnotify>0</bnotify>
<archive>0</archive>
<vgroups>0</vgroups>
<balance>0.0</balance>
<credit>0.0</credit>
<blimit>0.0</blimit>
```

```
<number></number>
<code></code>
<date></date>
<bcheck></bcheck>
<symbol></symbol>
</agreements>
<addons>
<uid>0</uid>
<type>0</type>
<idx>0</idx>
<name></name>
<descr></descr>
<strvalue></strvalue>
</addons>
</ret>
</lbapi:getAccountResponse>
</SOAP-ENV:Body>
```

api3.getVgroup.req.xml

```
<SOAP-ENV:Body>
<lbapi:getVgroup>
<id>0</id>
</lbapi:getVgroup>
</SOAP-ENV:Body>
```

api3.getVgroup.res.xml

```
<SOAP-ENV:Body>
<lbapi:getVgroupResponse>
<ret>
<vgroup>
<vgid>0</vgid>
<parentvgid>0</parentvgid>
<cuid>0</cuid>
<tarid>0</tarid>
<agrmid>0</agrmid>
<id>0</id>
<uid>0</uid>
<shape>0</shape>
<currentshape>0</currentshape>
<maxsessions>0</maxsessions>
```

```
<blkreq>0</blkreq>
<blocked>0</blocked>
<changed>0</changed>
<templ>0</templ>
<ipdet>0</ipdet>
<portdet>0</portdet>
<ifindex>0</ifindex>
<archive>0</archive>
<usercategory>0</usercategory>
<checkduplicate>0</checkduplicate>
<dirty>0</dirty>
<connectedfrom>0</connectedfrom>
<dlimit>0</dlimit>
<amount>0.000000</amount>
<acondatfirst></acondatfirst>
<creationdate></creationdate>
<acondat></acondat>
<accaffdate></accaffdate>
<blockdate></blockdate>
<cdate></cdate>
<dclear></dclear>
<descr></descr>
<login></login>
<pass></pass>
<parentvglogin></parentvglogin>
</vgroup>
<tarrasp>
<recordid>0</recordid>
<vgid>0</vgid>
<agrmid>0</agrmid>
<groupid>0</groupid>
<uid>0</uid>
<id>0</id>
<taridnew>0</taridnew>
<taridold>0</taridold>
<agenttype>0</agenttype>
<tarnewcurid>0</tarnewcurid>
<taroldcurid>0</taroldcurid>
<requestby></requestby>
<changetime></changetime>
<vglogin></vglogin>
<agrmnum></agrmnum>
<code></code>
<accname></accname>
<agentname></agentname>
```

```
<tarnewname></tarnewname>
<taroldname></taroldname>
<mgrname></mgrname>
<tarnewsymbol></tarnewsymbol>
<taroldsymbol></taroldsymbol>
</tarrasp>
<blockrasp>
<vgid>0</vgid>
<groupid>0</groupid>
<blkreq>0</blkreq>
<id>0</id>
<requestby>-1</requestby>
<changetime></changetime>
<mgrname></mgrname>
<mgrdescr></mgrdescr>
</blockrasp>
<staff>
<recordid>0</recordid>
<vgid>0</vgid>
<type>0</type>
<as>0</as>
<ipmask>
<ip></ip>
<mask>0</mask>
</ipmask>
</staff>
<telstaff>
<recordid>0</recordid>
<vgid>0</vgid>
<device>0</device>
<ldservice>1</ldservice>
<phonenumber></phonenumber>
<comment></comment>
</telstaff>
<macstaff>
<recordid>0</recordid>
<vgid>0</vgid>
<segment></segment>
<mac></mac>
</macstaff>
<addons>
<vgid>0</vgid>
<type>0</type>
<agentid>0</agentid>
<idx>0</idx>
```

```
<name></name>
<descr></descr>
<strvalue></strvalue>
</addons>
<services>
  <vgid>0</vgid>
  <tarid>0</tarid>
  <catid>0</catid>
</services>
<addresses>
  <type>0</type>
  <code></code>
  <address></address>
</addresses>
</ret>
</lbapi:getVgroupResponse>
</SOAP-ENV:Body>
```

api3.delAccount.req.xml

```
<SOAP-ENV:Body>
  <lbapi:delAccount>
    <id>0</id>
  </lbapi:delAccount>
</SOAP-ENV:Body>
```

api3.delAccount.res.xml

```
<SOAP-ENV:Body>
  <lbapi:delAccountResponse>
    <ret>0</ret>
  </lbapi:delAccountResponse>
</SOAP-ENV:Body>
```

api3.insupdVgroup.req.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdVgroup>
    <isInsert>0</isInsert>
    <val>
```

```
<vgroup>
<vgid>0</vgid>
<parentvgid>0</parentvgid>
<cuid>0</cuid>
<tarid>0</tarid>
<agrmid>0</agrmid>
<id>0</id>
<uid>0</uid>
<shape>0</shape>
<currentshape>0</currentshape>
<maxsessions>0</maxsessions>
<blkreq>0</blkreq>
<blocked>0</blocked>
<changed>0</changed>
<templ>0</templ>
<ipdet>0</ipdet>
<portdet>0</portdet>
<archive>0</archive>
<usercategory>0</usercategory>
<checkduplicate>0</checkduplicate>
<dirty>0</dirty>
<connectedfrom>0</connectedfrom>
<dlimit>0</dlimit>
<amount>0.000000</amount>
<acondatefirst></acondatefirst>
<creationdate></creationdate>
<acondate></acondate>
<accaffdate></accaffdate>
<blockdate></blockdate>
<cdate></cdate>
<dclear></dclear>
<descr></descr>
<login></login>
<pass></pass>
<parentvglogin></parentvglogin>
</vgroup>
<tarrasp>
<recordid>0</recordid>
<vgid>0</vgid>
<agrmid>0</agrmid>
<groupid>0</groupid>
<uid>0</uid>
<id>0</id>
<taridnew>0</taridnew>
<taridold>0</taridold>
```

```
<agenttype>0</agenttype>
<tarnewcurid>0</tarnewcurid>
<taroldcurid>0</taroldcurid>
<requestby></requestby>
<changetime></changetime>
<vglogin></vglogin>
<agrmnum></agrmnum>
<code></code>
<accname></accname>
<agentname></agentname>
<tarnewname></tarnewname>
<taroldname></taroldname>
<mgrname></mgrname>
<tarnewsymbol></tarnewsymbol>
<taroldsymbol></taroldsymbol>
</tarrasp>
<blockrasp>
<vgid>0</vgid>
<groupid>0</groupid>
<blkreq>0</blkreq>
<id>0</id>
<requestby>-1</requestby>
<changetime></changetime>
<mgrname></mgrname>
<mgrdescr></mgrdescr>
</blockrasp>
<staff>
<recordid>0</recordid>
<vgid>0</vgid>
<type>0</type>
<as>0</as>
<ipmask>
<ip></ip>
<mask>0</mask>
</ipmask>
</staff>
<telstaff>
<recordid>0</recordid>
<vgid>0</vgid>
<device>0</device>
<ldservice>1</ldservice>
<phonenumber></phonenumber>
<comment></comment>
</telstaff>
<macstaff>
```



```
<recordid>0</recordid>
<vgid>0</vgid>
<segment></segment>
<mac></mac>
</macstaff>
<addons>
<vgid>0</vgid>
<type>0</type>
<agentid>0</agentid>
<idx>0</idx>
<name></name>
<descr></descr>
<strvalue></strvalue>
</addons>
<services>
<vgid>0</vgid>
<tarid>0</tarid>
<catid>0</catid>
</services>
<addresses>
<type>0</type>
<code></code>
<address></address>
</addresses>
</val>
</lbapi:insupdVgroup>
</SOAP-ENV:Body>
```

api3.insupdVgroup.res.xml

```
<SOAP-ENV:Body>
<lbapi:insupdVgroupResponse>
<ret>0</ret>
</lbapi:insupdVgroupResponse>
</SOAP-ENV:Body>
```

api3.delVgroup.req.xml

```
<SOAP-ENV:Body>
<lbapi:delVgroup>
<id>0</id>
</lbapi:delVgroup>
```

```
</SOAP-ENV:Body>
```

api3.delVgroup.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:delVgroupResponse>  
    <ret>0</ret>  
  </lbapi:delVgroupResponse>  
</SOAP-ENV:Body>
```

api3.insupdAccount.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:insupdAccount>  
    <isInsert>0</isInsert>  
    <val>  
      <account>  
        <uid>0</uid>  
        <ipaccess>0</ipaccess>  
        <billdelivery>0</billdelivery>  
        <category>0</category>  
        <type>0</type>  
        <oksm>0</oksm>  
        <templ>0</templ>  
        <wrongactive>0</wrongactive>  
        <archive>0</archive>  
        <login></login>  
        <pass></pass>  
        <descr></descr>  
        <name></name>  
        <phone></phone>  
        <fax></fax>  
        <email></email>  
        <bankname></bankname>  
        <branchbankname></branchbankname>  
        <treasuryname></treasuryname>  
        <treasuryaccount></treasuryaccount>  
        <bik></bik>  
        <settl></settl>  
        <corr></corr>  
        <kpp></kpp>  
        <inn></inn>  
        <ogrn></ogrn>
```

```
<okpo></okpo>
<okved></okved>
<gendiru></gendiru>
<glbuhgu></glbuhgu>
<kontperson></kontperson>
<actonwhat></actonwhat>
<passsernum></passsernum>
<passno></passno>
<passissuedate></passissuedate>
<passissuedep></passissuedep>
<passissueplace></passissueplace>
<birthdate></birthdate>
<birthplace></birthplace>
<lastmoddate></lastmoddate>
<wrongdate></wrongdate>
<okato></okato>
<uuid></uuid>
<swift></swift>
<kio></kio>
<bicbei></bicbei>
<iban></iban>
<bankcorr></bankcorr>
<bankcorrcode></bankcorrcode>
<bankcorraccount></bankcorraccount>
<currency></currency>
<resident>0</resident>
</account>
<usergroups>
<usergroup>
<groupid>0</groupid>
<promiseallow>0</promiseallow>
<promiserent>0</promiserent>
<promisetill>0</promisetill>
<promisemax>0.000000</promisemax>
<promisemin>0.000000</promisemin>
<promiselimit>0.000000</promiselimit>
<name></name>
<description></description>
</usergroup>
<uids>
<val>0</val>
</uids>
<usercnt>0</usercnt>
<fread>0</fread>
<fwrite>0</fwrite>
```

```
</usergroups>
<addresses>
  <type>0</type>
  <code></code>
  <address></address>
</addresses>
<agreements>
  <agrmid>0</agrmid>
  <uid>0</uid>
  <operid>0</operid>
  <curid>0</curid>
  <bnotify>0</bnotify>
  <archive>0</archive>
  <vgroups>0</vgroups>
  <balance>0.0</balance>
  <credit>0.0</credit>
  <blimit>0.0</blimit>
  <number></number>
  <code></code>
  <date></date>
  <bcheck></bcheck>
  <symbol></symbol>
</agreements>
<addons>
  <uid>0</uid>
  <type>0</type>
  <idx>0</idx>
  <name></name>
  <descr></descr>
  <strvalue></strvalue>
</addons>
</val>
</lbapi:insupdAccount>
</SOAP-ENV:Body>
```

api3.insupdAccount.res.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdAccountResponse>
    <ret>0</ret>
  </lbapi:insupdAccountResponse>
</SOAP-ENV:Body>
```

api3.getDocuments.req.xml

```
<SOAP-ENV:Body>
<lbapi:getDocuments>
  <flt>
    <agrmid>0</agrmid>
    <userid>0</userid>
    <vgid>0</vgid>
    <tarid>0</tarid>
    <taridprev>0</taridprev>
    <operid>0</operid>
    <agentid>0</agentid>
    <recordid>0</recordid>
    <istemplate>0</istemplate>
    <ugroups>0</ugroups>
    <groups>0</groups>
    <notgroups>0</notgroups>
    <direction>0</direction>
    <pgsize>0</pgsize>
    <pgnum>0</pgnum>
    <setid>0</setid>
    <activated>0</activated>
    <personid>0</personid>
    <archive>0</archive>
    <blocked>0</blocked>
    <catid>0</catid>
    <nodata>0</nodata>
    <port>0</port>
    <proto>0</proto>
    <asnum>0</asnum>
    <repnum>0</repnum>
    <repdetail>0</repdetail>
    <mgrid>0</mgrid>
    <onfly>0</onfly>
    <docid>0</docid>
    <orderid>0</orderid>
    <ordernum>0</ordernum>
    <common>0</common>
    <deviceid>0</deviceid>
    <groupid>0</groupid>
    <showdefault>0</showdefault>
    <defaultonly>0</defaultonly>
    <durfrom>0</durfrom>
    <durto>0</durto>
    <payed>-1</payed>
    <addresstype>1</addresstype>
```

```
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<receipt></receipt>
<serno></serno>
<cardkey></cardkey>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dto></dto>
<numfrom></numfrom>
<numto></numto>
<name></name>
<login></login>
<vglogin></vglogin>
<descr></descr>
<tardescr></tardescr>
<agrmnum></agrmnum>
<email></email>
<phone></phone>
<inn></inn>
<ip></ip>
<dstip></dstip>
<telstaff></telstaff>
<code></code>
<telnum></telnum>
<namesound></namesound>
<addresscode></addresscode>
</flt>
</lbapi:getDocuments>
</SOAP-ENV:Body>
```

api3.getDocuments.res.xml

```
<SOAP-ENV:Body>
<lbapi:getDocumentsResponse>
<ret>
<docid>0</docid>
<ndsabove>0</ndsabove>
<usergroupid>0</usergroupid>
<groupid>0</groupid>
<payable>0</payable>
<clientallowed>0</clientallowed>
<hidden>0</hidden>
<onfly>0</onfly>
```

```
<detail>0</detail>
<name></name>
<doctype></doctype>
<savepath></savepath>
<uploadext></uploadext>
<usergroupname></usergroupname>
<groupname></groupname>
</ret>
</lbapi:getDocumentsResponse>
</SOAP-ENV:Body>
```

api3.getPayments.req.xml

```
<SOAP-ENV:Body>
<lbapi:getPayments>
<flt>
<agrmid>0</agrmid>
<userid>0</userid>
<vgid>0</vgid>
<tarid>0</tarid>
<taridprev>0</taridprev>
<operid>0</operid>
<agentid>0</agentid>
<recordid>0</recordid>
<istemplate>0</istemplate>
<ugroups>0</ugroups>
<groups>0</groups>
<notgroups>0</notgroups>
<direction>0</direction>
<pgsize>0</pgsize>
<pgnum>0</pgnum>
<setid>0</setid>
<activated>0</activated>
<personid>0</personid>
<archive>0</archive>
<blocked>0</blocked>
<catid>0</catid>
<nodata>0</nodata>
<port>0</port>
<proto>0</proto>
<asnum>0</asnum>
<repnum>0</repnum>
<repdetail>0</repdetail>
<mgrid>0</mgrid>
```

```
<onfly>0</onfly>
<docid>0</docid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<common>0</common>
<deviceid>0</deviceid>
<groupid>0</groupid>
<showdefault>0</showdefault>
<defaultonly>0</defaultonly>
<durfrom>0</durfrom>
<durto>0</durto>
<payed>-1</payed>
<addresstype>1</addresstype>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<receipt></receipt>
<serno></serno>
<cardkey></cardkey>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<numfrom></numfrom>
<numto></numto>
<name></name>
<login></login>
<vglogin></vglogin>
<descr></descr>
<tardescr></tardescr>
<agrmnum></agrmnum>
<email></email>
<phone></phone>
<inn></inn>
<ip></ip>
<dstip></dstip>
<telstaff></telstaff>
<code></code>
<telnum></telnum>
<namesound></namesound>
<addresscode></addresscode>
</flt>
</lbapi:getPayments>
</SOAP-ENV:Body>
```


api3.getPayments.res.xml

```
<SOAP-ENV:Body>
  <lbapi:getPaymentsResponse>
    <ret>
      <pay>
        <recordid>0</recordid>
        <agrmid>0</agrmid>
        <modperson>0</modperson>
        <currid>0</currid>
        <orderid>0</orderid>
        <status>0</status>
        <amount>0.0</amount>
        <paydate></paydate>
        <localdate></localdate>
        <canceldate></canceldate>
        <receipt></receipt>
        <comment></comment>
      </pay>
      <amountcurr>0.0</amountcurr>
      <ordernum>0</ordernum>
      <uid>0</uid>
      <operid>0</operid>
      <curr symb></curr symb>
      <uname></uname>
      <agrm></agrm>
      <mgr></mgr>
      <opername></opername>
    </ret>
  </lbapi:getPaymentsResponse>
</SOAP-ENV:Body>
```

api3.insupdDocument.req.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdDocument>
    <isInsert>0</isInsert>
    <val>
      <docid>0</docid>
      <ndsabove>0</ndsabove>
      <usergroupid>0</usergroupid>
      <groupid>0</groupid>
      <payable>0</payable>
      <clientallowed>0</clientallowed>
    </val>
  </lbapi:insupdDocument>
</SOAP-ENV:Body>
```

```
<hidden>0</hidden>
<onfly>0</onfly>
<detail>0</detail>
<name></name>
<doctemplate></doctemplate>
<savepath></savepath>
<uploadext></uploadext>
<usergroupname></usergroupname>
<groupname></groupname>
</val>
</lbapi:insupdDocument>
</SOAP-ENV:Body>
```

api3.insupdDocument.res.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdDocumentResponse>
    <ret>0</ret>
  </lbapi:insupdDocumentResponse>
</SOAP-ENV:Body>
```

api3.insupdTarifsRasp.req.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdTarifsRasp>
    <isInsert>0</isInsert>
    <val>
      <recordid>0</recordid>
      <vgid>0</vgid>
      <agrmid>0</agrmid>
      <groupid>0</groupid>
      <uid>0</uid>
      <id>0</id>
      <taridnew>0</taridnew>
      <taridold>0</taridold>
      <agenttype>0</agenttype>
      <tarnewcurid>0</tarnewcurid>
      <taroldcurid>0</taroldcurid>
      <requestby></requestby>
      <changetime></changetime>
      <vglogin></vglogin>
      <agrmnum></agrmnum>
    </val>
  </lbapi:insupdTarifsRasp>
</SOAP-ENV:Body>
```

```
<accname></accname>
<agentname></agentname>
<tarnewname></tarnewname>
<taroldname></taroldname>
<mgrname></mgrname>
<tarnewsymbol></tarnewsymbol>
<taroldsymbol></taroldsymbol>
</val>
</lbapi:insupdTarifsRasp>
</SOAP-ENV:Body>
```

api3.insupdTarifsRasp.res.xml

```
<SOAP-ENV:Body>
<lbapi:insupdTarifsRaspResponse>
<ret>0</ret>
</lbapi:insupdTarifsRaspResponse>
</SOAP-ENV:Body>
```

api3.delTarifsRasp.req.xml

```
<SOAP-ENV:Body>
<lbapi:delTarifsRasp>
<id>0</id>
</lbapi:delTarifsRasp>
</SOAP-ENV:Body>
```

api3.delTarifsRasp.res.xml

```
<SOAP-ENV:Body>
<lbapi:delTarifsRaspResponse>
<ret>0</ret>
</lbapi:delTarifsRaspResponse>
</SOAP-ENV:Body>
```

api3.getTarifsRasp.req.xml

```
<SOAP-ENV:Body>
<lbapi:getTarifsRasp>
```

```
<flt>
<agrmid>0</agrmid>
<userid>0</userid>
<vgid>0</vgid>
<tarid>0</tarid>
<taridprev>0</taridprev>
<operid>0</operid>
<agentid>0</agentid>
<recordid>0</recordid>
<istemplate>0</istemplate>
<ugroups>0</ugroups>
<groups>0</groups>
<notgroups>0</notgroups>
<direction>0</direction>
<pgsize>0</pgsize>
<pgnum>0</pgnum>
<setid>0</setid>
<activated>0</activated>
<personid>0</personid>
<archive>0</archive>
<blocked>0</blocked>
<catid>0</catid>
<nodata>0</nodata>
<port>0</port>
<proto>0</proto>
<asnum>0</asnum>
<repnum>0</repnum>
<repdetail>0</repdetail>
<mgrid>0</mgrid>
<onfly>0</onfly>
<docid>0</docid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<common>0</common>
<deviceid>0</deviceid>
<groupid>0</groupid>
<showdefault>0</showdefault>
<defaultonly>0</defaultonly>
<durfrom>0</durfrom>
<durto>0</durto>
<payed>-1</payed>
<addresstype>1</addresstype>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<receipt></receipt>
```

```
<serno></serno>
<cardkey></cardkey>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<numfrom></numfrom>
<numto></numto>
<name></name>
<login></login>
<vglogin></vglogin>
<descr></descr>
<tardescr></tardescr>
<agrmnum></agrmnum>
<email></email>
<phone></phone>
<inn></inn>
<ip></ip>
<dstip></dstip>
<telstaff></telstaff>
<code></code>
<telnum></telnum>
<namesound></namesound>
<addresscode></addresscode>
</flt>
</lbapi:getTarifsRasp>
</SOAP-ENV:Body>
```

api3.getTarifsRasp.res.xml

```
<SOAP-ENV:Body>
<lbapi:getTarifsRaspResponse>
<ret>
<recordid>0</recordid>
<vgid>0</vgid>
<agrmid>0</agrmid>
<groupid>0</groupid>
<uid>0</uid>
<id>0</id>
<taridnew>0</taridnew>
<taridold>0</taridold>
<agenttype>0</agenttype>
<tarnewcurid>0</tarnewcurid>
<taroldcurid>0</taroldcurid>
```

```
<requestby></requestby>
<changetime></changetime>
<vglogin></vglogin>
<agrmnum></agrmnum>
<code></code>
<accname></accname>
<agentname></agentname>
<tarnewname></tarnewname>
<taroldname></taroldname>
<mgrname></mgrname>
<tarnewsymbol></tarnewsymbol>
<taroldsymbol></taroldsymbol>
</ret>
</lbapi:getTarifsRaspResponse>
</SOAP-ENV:Body>
```

api3.getCurrencies.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getCurrencies>
</lbapi:getCurrencies>
</SOAP-ENV:Body>
```

api3.getCurrencies.res.xml

```
<SOAP-ENV:Body>
  <lbapi:getCurrenciesResponse>
    <ret>
      <id>0</id>
      <def>0</def>
      <symbol></symbol>
      <name></name>
    </ret>
  </lbapi:getCurrenciesResponse>
</SOAP-ENV:Body>
```

api3.getOptions.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getOptions>
```

```
</lbapi:getOptions>  
</SOAP-ENV:Body>
```

api3.getOptions.res.xml

```
<SOAP-ENV:Body>  
<lbapi:getOptionsResponse>  
<ret>  
<name></name>  
<descr></descr>  
<value></value>  
</ret>  
</lbapi:getOptionsResponse>  
</SOAP-ENV:Body>
```

api3.getLicense.req.xml

```
<SOAP-ENV:Body>  
<lbapi:getLicense>  
</lbapi:getLicense>  
</SOAP-ENV:Body>
```

api3.getLicense.res.xml

```
<SOAP-ENV:Body>  
<lbapi:getLicenseResponse>  
<ret>  
<progrname></progrname>  
<cdkey></cdkey>  
<progowner></progowner>  
<gendate></gendate>  
<expire>0</expire>  
<userlimit>0</userlimit>  
<useoperators>0</useoperators>  
<usefidelio>0</usefidelio>  
<useec>0</useec>  
<useinventory>0</useinventory>  
<useavdesk>0</useavdesk>  
<usecerber>0</usecerber>  
<payments>0</payments>  
</ret>
```

```
</lbapi:getLicenseResponse>  
</SOAP-ENV:Body>
```

api3.getRates.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:getRates>  
    <date></date>  
  </lbapi:getRates>  
</SOAP-ENV:Body>
```

api3.getRates.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:getRatesResponse>  
    <ret>  
      <curid>0</curid>  
      <def>0</def>  
      <rate>0.0</rate>  
      <date></date>  
      <name></name>  
      <symbol></symbol>  
    </ret>  
  </lbapi:getRatesResponse>  
</SOAP-ENV:Body>
```

api3.getCardSets.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:getCardSets>  
  </lbapi:getCardSets>  
</SOAP-ENV:Body>
```

api3.getCardSets.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:getCardSetsResponse>  
  <ret>
```



```
<setid>0</setid>
<createdby>0</createdby>
<curid>0</curid>
<acctpl>0</acctpl>
<expireperiod>0</expireperiod>
<cardscnt>0</cardscnt>
<setdescr></setdescr>
<createdbyname></createdbyname>
<currname></currname>
<accname></accname>
</ret>
</lbapi:getCardSetsResponse>
</SOAP-ENV:Body>
```

api3.insupdCardSet.req.xml

```
<SOAP-ENV:Body>
<lbapi:insupdCardSet>
<isInsert>0</isInsert>
<val>
<cardset>
<setid>0</setid>
<createdby>0</createdby>
<curid>0</curid>
<acctpl>0</acctpl>
<expireperiod>0</expireperiod>
<cardscnt>0</cardscnt>
<setdescr></setdescr>
<createdbyname></createdbyname>
<currname></currname>
<accname></accname>
</cardset>
<list>
<setid>0</setid>
<vgtpl>0</vgtpl>
<common>0</common>
<vgtplid>0</vgtplid>
<vglogin></vglogin>
</list>
</val>
</lbapi:insupdCardSet>
</SOAP-ENV:Body>
```

api3.insupdCardSet.res.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdCardSetResponse>
    <ret>0</ret>
  </lbapi:insupdCardSetResponse>
</SOAP-ENV:Body>
```

api3.delCardSet.req.xml

```
<SOAP-ENV:Body>
  <lbapi:delCardSet>
    <id>0</id>
  </lbapi:delCardSet>
</SOAP-ENV:Body>
```

api3.delCardSet.res.xml

```
<SOAP-ENV:Body>
  <lbapi:delCardSetResponse>
    <ret>0</ret>
  </lbapi:delCardSetResponse>
</SOAP-ENV:Body>
```

api3.genOrders.req.xml

```
<SOAP-ENV:Body>
  <lbapi:genOrders>
    <val>
      <oper>0</oper>
      <uid>0</uid>
      <grp>0</grp>
      <ugrp>0</ugrp>
      <docid>0</docid>
      <period>0</period>
      <num>0</num>
      <date>0</date>
      <agrmid>0</agrmid>
      <summ>0.0</summ>
    </val>
    <flt>
      <agrmid>0</agrmid>
```

```
<userid>0</userid>
<vgid>0</vgid>
<tarid>0</tarid>
<taridprev>0</taridprev>
<operid>0</operid>
<agentid>0</agentid>
<recordid>0</recordid>
<istemplate>0</istemplate>
<ugroups>0</ugroups>
<groups>0</groups>
<notgroups>0</notgroups>
<direction>0</direction>
<pgsize>0</pgsize>
<pgnum>0</pgnum>
<setid>0</setid>
<activated>0</activated>
<personid>0</personid>
<archive>0</archive>
<blocked>0</blocked>
<catid>0</catid>
<nodata>0</nodata>
<port>0</port>
<proto>0</proto>
<asnum>0</asnum>
<repnum>0</repnum>
<repdetail>0</repdetail>
<mgrid>0</mgrid>
<onfly>0</onfly>
<docid>0</docid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<common>0</common>
<deviceid>0</deviceid>
<groupid>0</groupid>
<showdefault>0</showdefault>
<defaultonly>0</defaultonly>
<durfrom>0</durfrom>
<durto>0</durto>
<payed>-1</payed>
<addresstype>1</addresstype>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<receipt></receipt>
<serno></serno>
<cardkey></cardkey>
```

```
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<numfrom></numfrom>
<numto></numto>
<name></name>
<login></login>
<vglogin></vglogin>
<descr></descr>
<tardescr></tardescr>
<agrmnum></agrmnum>
<email></email>
<phone></phone>
<inn></inn>
<ip></ip>
<dstip></dstip>
<telstaff></telstaff>
<code></code>
<telnum></telnum>
<namesound></namesound>
<addresscode></addresscode>
</flt>
</lbapi:genOrders>
</SOAP-ENV:Body>
```

api3.genOrders.res.xml

```
<SOAP-ENV:Body>
  <lbapi:genOrdersResponse>
    <ret>0</ret>
  </lbapi:genOrdersResponse>
</SOAP-ENV:Body>
```

api3.genOrderGetFile.req.xml

```
<SOAP-ENV:Body>
  <lbapi:genOrderGetFile>
    <val>
      <oper>0</oper>
      <uid>0</uid>
      <grp>0</grp>
    </val>
  </lbapi:genOrderGetFile>
</SOAP-ENV:Body>
```

```
<ugrp>0</ugrp>
<docid>0</docid>
<period>0</period>
<num>0</num>
<date>0</date>
<agrmid>0</agrmid>
<summ>0.0</summ>
</val>
<flt>
<agrmid>0</agrmid>
<userid>0</userid>
<vgid>0</vgid>
<tarid>0</tarid>
<taridprev>0</taridprev>
<operid>0</operid>
<agentid>0</agentid>
<recordid>0</recordid>
<istemplate>0</istemplate>
<ugroups>0</ugroups>
<groups>0</groups>
<notgroups>0</notgroups>
<direction>0</direction>
<pgsize>0</pgsize>
<pgnum>0</pgnum>
<setid>0</setid>
<activated>0</activated>
<personid>0</personid>
<archive>0</archive>
<blocked>0</blocked>
<catid>0</catid>
<nodata>0</nodata>
<port>0</port>
<proto>0</proto>
<asnum>0</asnum>
<repnum>0</repnum>
<repdetail>0</repdetail>
<mgrid>0</mgrid>
<onfly>0</onfly>
<docid>0</docid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<common>0</common>
<deviceid>0</deviceid>
<groupid>0</groupid>
<showdefault>0</showdefault>
```

```
<defaultonly>0</defaultonly>
<durfrom>0</durfrom>
<durto>0</durto>
<payed>-1</payed>
<addresstype>1</addresstype>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<receipt></receipt>
<serno></serno>
<cardkey></cardkey>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<numfrom></numfrom>
<numto></numto>
<name></name>
<login></login>
<vglogin></vglogin>
<descr></descr>
<tardescr></tardescr>
<agrmnum></agrmnum>
<email></email>
<phone></phone>
<inn></inn>
<ip></ip>
<dstip></dstip>
<telstaff></telstaff>
<code></code>
<telnum></telnum>
<namesound></namesound>
<addresscode></addresscode>
</flt>
</lbapi:genOrderGetFile>
</SOAP-ENV:Body>
```

api3.genOrderGetFile.res.xml

```
<SOAP-ENV:Body>
  <lbapi:genOrderGetFileResponse>
    <ret></ret>
  </lbapi:genOrderGetFileResponse>
</SOAP-ENV:Body>
```

api3.insupdCurrency.req.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdCurrency>
    <isInsert>0</isInsert>
    <val>
      <id>0</id>
      <def>0</def>
      <symbol></symbol>
      <name></name>
    </val>
  </lbapi:insupdCurrency>
</SOAP-ENV:Body>
```

api3.insupdCurrency.res.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdCurrencyResponse>
    <ret>0</ret>
  </lbapi:insupdCurrencyResponse>
</SOAP-ENV:Body>
```

api3.delCurrency.req.xml

```
<SOAP-ENV:Body>
  <lbapi:delCurrency>
    <id>0</id>
  </lbapi:delCurrency>
</SOAP-ENV:Body>
```

api3.delCurrency.res.xml

```
<SOAP-ENV:Body>
  <lbapi:delCurrencyResponse>
    <ret>0</ret>
  </lbapi:delCurrencyResponse>
</SOAP-ENV:Body>
```

api3.updCurrencyRate.req.xml

```
<SOAP-ENV:Body>
  <lbapi:updCurrencyRate>
    <val>
      <curid>0</curid>
      <def>0</def>
      <modperson>0</modperson>
      <rate>0.0</rate>
      <date></date>
      <datetill></datetill>
      <name></name>
      <symbol></symbol>
    </val>
  </lbapi:updCurrencyRate>
</SOAP-ENV:Body>
```

api3.updCurrencyRate.res.xml

```
<SOAP-ENV:Body>
  <lbapi:updCurrencyRateResponse>
    <ret>0</ret>
  </lbapi:updCurrencyRateResponse>
</SOAP-ENV:Body>
```

api3.delOrder.req.xml

```
<SOAP-ENV:Body>
  <lbapi:delOrder>
    <id>0</id>
  </lbapi:delOrder>
</SOAP-ENV:Body>
```

api3.delOrder.res.xml

```
<SOAP-ENV:Body>
  <lbapi:delOrderResponse>
    <ret>0</ret>
  </lbapi:delOrderResponse>
</SOAP-ENV:Body>
```


api3.getVgroups.req.xml

```
<SOAP-ENV:Body>
<lbapi:getVgroups>
  <flt>
    <agrmid>0</agrmid>
    <userid>0</userid>
    <vgid>0</vgid>
    <tarid>0</tarid>
    <taridprev>0</taridprev>
    <operid>0</operid>
    <agentid>0</agentid>
    <recordid>0</recordid>
    <istemplate>0</istemplate>
    <ugroups>0</ugroups>
    <groups>0</groups>
    <notgroups>0</notgroups>
    <direction>0</direction>
    <pgsize>0</pgsize>
    <pgnum>0</pgnum>
    <setid>0</setid>
    <activated>0</activated>
    <personid>0</personid>
    <archive>0</archive>
    <blocked>0</blocked>
    <catid>0</catid>
    <nodata>0</nodata>
    <port>0</port>
    <proto>0</proto>
    <asnum>0</asnum>
    <repnum>0</repnum>
    <repdetail>0</repdetail>
    <mgrid>0</mgrid>
    <onfly>0</onfly>
    <docid>0</docid>
    <orderid>0</orderid>
    <ordernum>0</ordernum>
    <common>0</common>
    <deviceid>0</deviceid>
    <groupid>0</groupid>
    <showdefault>0</showdefault>
    <defaultonly>0</defaultonly>
    <durfrom>0</durfrom>
    <durto>0</durto>
    <payed>-1</payed>
    <addresstype>1</addresstype>
```

```
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<receipt></receipt>
<serno></serno>
<cardkey></cardkey>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dto></dto>
<numfrom></numfrom>
<numto></numto>
<name></name>
<login></login>
<vglogin></vglogin>
<descr></descr>
<tardescr></tardescr>
<agrmnum></agrmnum>
<email></email>
<phone></phone>
<inn></inn>
<ip></ip>
<dstip></dstip>
<telstaff></telstaff>
<code></code>
<telnum></telnum>
<namesound></namesound>
<addresscode></addresscode>
</flt>
</lbapi:getVgroups>
</SOAP-ENV:Body>
```

api3.getVgroups.res.xml

```
<SOAP-ENV:Body>
<lbapi:getVgroupsResponse>
<ret>
<vgid>0</vgid>
<id>0</id>
<tarid>0</tarid>
<agrmd>0</agrmd>
<blkreq>0</blkreq>
<blocked>0</blocked>
<canmodify>0</canmodify>
<uid>0</uid>
```

```
<balance>0.0</balance>
<login></login>
<descr></descr>
<agrmnum></agrmnum>
<code></code>
<username></username>
<creationdate></creationdate>
<accondate></accondate>
<accoffdate></accoffdate>
<blockdate></blockdate>
<agentdescr></agentdescr>
<tarifdescr></tarifdescr>
<symbol></symbol>
<address>
  <type>0</type>
  <code></code>
  <address></address>
</address>
</ret>
</lbapi:getVgroupsResponse>
</SOAP-ENV:Body>
```

api3.getAddressCountries.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getAddressCountries>
    <flt>
      <recordid>0</recordid>
      <country>0</country>
      <region>0</region>
      <area>0</area>
      <city>0</city>
      <settl>0</settl>
      <street>0</street>
      <building>0</building>
      <level>0</level>
      <pgsize>0</pgsize>
      <pgnum>0</pgnum>
      <nodata>0</nodata>
      <name></name>
    </flt>
  </lbapi:getAddressCountries>
```

```
</SOAP-ENV:Body>
```

api3.getAddressCountries.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:getAddressCountriesResponse>  
    <ret>  
      <recordid>0</recordid>  
      <name></name>  
    </ret>  
  </lbapi:getAddressCountriesResponse>  
</SOAP-ENV:Body>
```

api3.getAddressRegions.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:getAddressRegions>  
    <flt>  
      <recordid>0</recordid>  
      <country>0</country>  
      <region>0</region>  
      <area>0</area>  
      <city>0</city>  
      <settl>0</settl>  
      <street>0</street>  
      <building>0</building>  
      <level>0</level>  
      <pgsize>0</pgsize>  
      <pgnum>0</pgnum>  
      <nodata>0</nodata>  
      <name></name>  
    </flt>  
  </lbapi:getAddressRegions>  
</SOAP-ENV:Body>
```

api3.getAddressRegions.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:getAddressRegionsResponse>  
    <ret>
```

```
<recordid>0</recordid>
<country>0</country>
<name></name>
<shortname></shortname>
</ret>
</lbapi:getAddressRegionsResponse>
</SOAP-ENV:Body>
```

api3.getAddressAreas.req.xml

```
<SOAP-ENV:Body>
<lbapi:getAddressAreas>
<flt>
<recordid>0</recordid>
<country>0</country>
<region>0</region>
<area>0</area>
<city>0</city>
<settl>0</settl>
<street>0</street>
<building>0</building>
<level>0</level>
<pgsize>0</pgsize>
<pgnum>0</pgnum>
<nodata>0</nodata>
<name></name>
</flt>
</lbapi:getAddressAreas>
</SOAP-ENV:Body>
```

api3.getAddressAreas.res.xml

```
<SOAP-ENV:Body>
<lbapi:getAddressAreasResponse>
<ret>
<recordid>0</recordid>
<region>0</region>
<name></name>
<shortname></shortname>
</ret>
</lbapi:getAddressAreasResponse>
```

```
</SOAP-ENV:Body>
```

api3.getAddressCities.req.xml

```
<SOAP-ENV:Body>  
<lbapi:getAddressCities>  
  <flt>  
    <recordid>0</recordid>  
    <country>0</country>  
    <region>0</region>  
    <area>0</area>  
    <city>0</city>  
    <settl>0</settl>  
    <street>0</street>  
    <building>0</building>  
    <level>0</level>  
    <pgsize>0</pgsize>  
    <pgnum>0</pgnum>  
    <nodata>0</nodata>  
  </name></name>  
</flt>  
</lbapi:getAddressCities>  
</SOAP-ENV:Body>
```

api3.getAddressCities.res.xml

```
<SOAP-ENV:Body>  
<lbapi:getAddressCitiesResponse>  
  <ret>  
    <recordid>0</recordid>  
    <region>0</region>  
    <area>0</area>  
    <name></name>  
    <shortname></shortname>  
  </ret>  
</lbapi:getAddressCitiesResponse>  
</SOAP-ENV:Body>
```

api3.getAddressSettles.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getAddressSettles>
    <flt>
      <recordid>0</recordid>
      <country>0</country>
      <region>0</region>
      <area>0</area>
      <city>0</city>
      <settl>0</settl>
      <street>0</street>
      <building>0</building>
      <level>0</level>
      <pgsize>0</pgsize>
      <pgnum>0</pgnum>
      <nodata>0</nodata>
      <name></name>
    </flt>
  </lbapi:getAddressSettles>
</SOAP-ENV:Body>
```

api3.getAddressSettles.res.xml

```
<SOAP-ENV:Body>
  <lbapi:getAddressSettlesResponse>
    <ret>
      <recordid>0</recordid>
      <region>0</region>
      <area>0</area>
      <city>0</city>
      <name></name>
      <shortname></shortname>
    </ret>
  </lbapi:getAddressSettlesResponse>
</SOAP-ENV:Body>
```

api3.getAddressStreets.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getAddressStreets>
    <flt>
      <recordid>0</recordid>
```

```
<country>0</country>
<region>0</region>
<area>0</area>
<city>0</city>
<settl>0</settl>
<street>0</street>
<building>0</building>
<level>0</level>
<pgsize>0</pgsize>
<pgnum>0</pgnum>
<nodata>0</nodata>
<name></name>
</flt>
</lbapi:getAddressStreets>
</SOAP-ENV:Body>
```

api3.getAddressStreets.res.xml

```
<SOAP-ENV:Body>
<lbapi:getAddressStreetsResponse>
<ret>
<recordid>0</recordid>
<region>0</region>
<city>0</city>
<settl>0</settl>
<idx>0</idx>
<name></name>
<shortname></shortname>
</ret>
</lbapi:getAddressStreetsResponse>
</SOAP-ENV:Body>
```

api3.getAddressBuildings.req.xml

```
<SOAP-ENV:Body>
<lbapi:getAddressBuildings>
<flt>
<recordid>0</recordid>
<country>0</country>
<region>0</region>
<area>0</area>
<city>0</city>
<settl>0</settl>
```



```
<street>0</street>
<building>0</building>
<level>0</level>
<pgsize>0</pgsize>
<pgnum>0</pgnum>
<nodata>0</nodata>
<name></name>
</flt>
</lbapi:getAddressBuildings>
</SOAP-ENV:Body>
```

api3.getAddressBuildings.res.xml

```
<SOAP-ENV:Body>
<lbapi:getAddressStreetsResponse>
<ret>
<recordid>0</recordid>
<region>0</region>
<city>0</city>
<settl>0</settl>
<idx>0</idx>
<name></name>
<shortname></shortname>
</ret>
</lbapi:getAddressStreetsResponse>
</SOAP-ENV:Body>
```

api3.getAddressMeanings.req.xml

```
<SOAP-ENV:Body>
<lbapi:getAddressMeanings>
<flt>
<recordid>0</recordid>
<country>0</country>
<region>0</region>
<area>0</area>
<city>0</city>
<settl>0</settl>
<street>0</street>
<building>0</building>
<level>0</level>
<pgsize>0</pgsize>
```

```
<pgnum>0</pgnum>
<nodata>0</nodata>
<name></name>
</flt>
</lbapi:getAddressMeanings>
</SOAP-ENV:Body>
```

api3.getAddressMeanings.res.xml

```
<SOAP-ENV:Body>
<lbapi:getAddressMeaningsResponse>
<ret>
<recordid>0</recordid>
<level1>0</level1>
<level2>0</level2>
<level3>0</level3>
<level4>0</level4>
<level5>0</level5>
<level6>0</level6>
<level7>0</level7>
<name></name>
<shortname></shortname>
</ret>
</lbapi:getAddressMeaningsResponse>
</SOAP-ENV:Body>
```

api3.insupdAddressCountry.req.xml

```
<SOAP-ENV:Body>
<lbapi:insupdAddressCountry>
<isInsert>0</isInsert>
<val>
<recordid>0</recordid>
<name></name>
</val>
</lbapi:insupdAddressCountry>
</SOAP-ENV:Body>
```

api3.insupdAddressCountry.res.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdAddressCountryResponse>
    <ret>0</ret>
  </lbapi:insupdAddressCountryResponse>
</SOAP-ENV:Body>
```

api3.insupdAddressRegion.req.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdAddressRegion>
    <isInsert>0</isInsert>
    <val>
      <recordid>0</recordid>
      <country>0</country>
      <name></name>
      <shortname></shortname>
    </val>
  </lbapi:insupdAddressRegion>
</SOAP-ENV:Body>
```

api3.insupdAddressRegion.res.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdAddressRegionResponse>
    <ret>0</ret>
  </lbapi:insupdAddressRegionResponse>
</SOAP-ENV:Body>
```

api3.insupdAddressArea.req.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdAddressArea>
    <isInsert>0</isInsert>
    <val>
      <recordid>0</recordid>
      <region>0</region>
      <name></name>
      <shortname></shortname>
    </val>
```

```
</lbapi:insupdAddressArea>  
</SOAP-ENV:Body>
```

api3.insupdAddressArea.res.xml

```
<SOAP-ENV:Body>  
<lbapi:insupdAddressAreaResponse>  
<ret>0</ret>  
</lbapi:insupdAddressAreaResponse>  
</SOAP-ENV:Body>
```

api3.insupdAddressCitiy.req.xml

```
<SOAP-ENV:Body>  
<lbapi:insupdAddressCity>  
<isInsert>0</isInsert>  
<val>  
<recordid>0</recordid>  
<region>0</region>  
<area>0</area>  
<name></name>  
<shortname></shortname>  
</val>  
</lbapi:insupdAddressCity>  
</SOAP-ENV:Body>
```

api3.insupdAddressCitiy.res.xml

```
<SOAP-ENV:Body>  
<lbapi:insupdAddressCityResponse>  
<ret>0</ret>  
</lbapi:insupdAddressCityResponse>  
</SOAP-ENV:Body>
```

api3.insupdAddressSettle.req.xml

```
<SOAP-ENV:Body>  
<lbapi:insupdAddressSettle>  
<isInsert>0</isInsert>  
<val>
```

```
<recordid>0</recordid>
<region>0</region>
<area>0</area>
<city>0</city>
<name></name>
<shortname></shortname>
</val>
</lbapi:insupdAddressSettle>
</SOAP-ENV:Body>
```

api3.insupdAddressSettle.res.xml

```
<SOAP-ENV:Body>
<lbapi:insupdAddressSettleResponse>
<ret>0</ret>
</lbapi:insupdAddressSettleResponse>
</SOAP-ENV:Body>
```

api3.insupdAddressStreet.req.xml

```
<SOAP-ENV:Body>
<lbapi:insupdAddressStreet>
<isInsert>0</isInsert>
<val>
<recordid>0</recordid>
<region>0</region>
<city>0</city>
<settl>0</settl>
<idx>0</idx>
<name></name>
<shortname></shortname>
</val>
</lbapi:insupdAddressStreet>
</SOAP-ENV:Body>
```

api3.insupdAddressStreet.res.xml

```
<SOAP-ENV:Body>
<lbapi:insupdAddressStreetResponse>
<ret>0</ret>
</lbapi:insupdAddressStreetResponse>
</SOAP-ENV:Body>
```

api3.insupdAddressBuilding.req.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdAddressBuilding>
    <isInsert>0</isInsert>
    <val>
      <recordid>0</recordid>
      <region>0</region>
      <city>0</city>
      <settl>0</settl>
      <street>0</street>
      <idx>0</idx>
      <name></name>
      <shortname></shortname>
      <block></block>
    </val>
  </lbapi:insupdAddressBuilding>
</SOAP-ENV:Body>
```

api3.insupdAddressBuilding.res.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdAddressBuildingResponse>
    <ret>0</ret>
  </lbapi:insupdAddressBuildingResponse>
</SOAP-ENV:Body>
```

api3.insupdAddressFlat.req.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdAddressFlat>
    <isInsert>0</isInsert>
    <val>
      <recordid>0</recordid>
      <region>0</region>
      <building>0</building>
      <name></name>
      <shortname></shortname>
    </val>
  </lbapi:insupdAddressFlat>
</SOAP-ENV:Body>
```

api3.insupdAddressFlat.res.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdAddressFlatResponse>
    <ret>0</ret>
  </lbapi:insupdAddressFlatResponse>
</SOAP-ENV:Body>
```

api3.delAddressCountry.req.xml

```
<SOAP-ENV:Body>
  <lbapi:delAddressCountry>
    <id>0</id>
  </lbapi:delAddressCountry>
</SOAP-ENV:Body>
```

api3.delAddressCountry.res.xml

```
<SOAP-ENV:Body>
  <lbapi:delAddressCountryResponse>
    <ret>0</ret>
  </lbapi:delAddressCountryResponse>
</SOAP-ENV:Body>
```

api3.delAddressRegion.req.xml

```
<SOAP-ENV:Body>
  <lbapi:delAddressRegion>
    <id>0</id>
  </lbapi:delAddressRegion>
</SOAP-ENV:Body>
```

api3.delAddressRegion.res.xml

```
<SOAP-ENV:Body>
  <lbapi:delAddressRegionResponse>
    <ret>0</ret>
  </lbapi:delAddressRegionResponse>
</SOAP-ENV:Body>
```

api3.delAddressArea.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressArea>  
    <id>0</id>  
  </lbapi:delAddressArea>  
</SOAP-ENV:Body>
```

api3.delAddressArea.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressAreaResponse>  
    <ret>0</ret>  
  </lbapi:delAddressAreaResponse>  
</SOAP-ENV:Body>
```

api3.delAddressCitiy.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressCity>  
    <id>0</id>  
  </lbapi:delAddressCity>  
</SOAP-ENV:Body>
```

api3.delAddressCitiy.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressCityResponse>  
    <ret>0</ret>  
  </lbapi:delAddressCityResponse>  
</SOAP-ENV:Body>
```

api3.delAddressSettle.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressSettle>  
    <id>0</id>  
  </lbapi:delAddressSettle>  
</SOAP-ENV:Body>
```


api3.delAddressSettle.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressSettleResponse>  
    <ret>0</ret>  
  </lbapi:delAddressSettleResponse>  
</SOAP-ENV:Body>
```

api3.delAddressStreet.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressStreet>  
    <id>0</id>  
  </lbapi:delAddressStreet>  
</SOAP-ENV:Body>
```

api3.delAddressStreet.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressStreetResponse>  
    <ret>0</ret>  
  </lbapi:delAddressStreetResponse>  
</SOAP-ENV:Body>
```

api3.delAddressBuilding.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressBuilding>  
    <id>0</id>  
  </lbapi:delAddressBuilding>  
</SOAP-ENV:Body>
```

api3.delAddressBuilding.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressBuildingResponse>  
    <ret>0</ret>  
  </lbapi:delAddressBuildingResponse>  
</SOAP-ENV:Body>
```

api3.delAddressFlat.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressFlat>  
    <id>0</id>  
  </lbapi:delAddressFlat>  
</SOAP-ENV:Body>
```

api3.delAddressFlat.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:delAddressFlatResponse>  
    <ret>0</ret>  
  </lbapi:delAddressFlatResponse>  
</SOAP-ENV:Body>
```

api3.Payment.req.xml

```
<SOAP-ENV:Body>  
  <xmlapi:Payments>  
    <flt>  
      <agrmid>0</agrmid>  
      <userid>0</userid>  
      <vgid>0</vgid>  
      <tarid>0</tarid>  
      <taridprev>0</taridprev>  
      <operid>0</operid>  
      <agentid>0</agentid>  
      <recordid>0</recordid>  
      <istemplate>0</istemplate>  
      <ugroups>0</ugroups>  
      <groups>0</groups>  
      <notgroups>0</notgroups>  
      <direction>0</direction>  
      <pgsize>0</pgsize>  
      <pgnum>0</pgnum>  
      <setid>0</setid>  
      <activated>0</activated>  
      <personid>0</personid>  
      <archive>0</archive>  
      <blocked>0</blocked>  
      <catid>0</catid>
```

```
<nodata>0</nodata>
<port>0</port>
<proto>0</proto>
<asnum>0</asnum>
<repnum>0</repnum>
<repdetail>0</repdetail>
<mgrid>0</mgrid>
<onfly>0</onfly>
<docid>0</docid>
<orderid>0</orderid>
<common>0</common>
<receipt></receipt>
<serno></serno>
<cardkey></cardkey>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<numfrom></numfrom>
<numto></numto>
<name></name>
<login></login>
<vglogin></vglogin>
<descr></descr>
<tardescr></tardescr>
<agrmnum></agrmnum>
<email></email>
<phone></phone>
<inn></inn>
<ip></ip>
<dstip></dstip>
<telstaff></telstaff>
<code></code>
<telnum></telnum>
<namesound></namesound>
</flt>
</xmlapi:Payments>
</SOAP-ENV:Body>
```

api3.Payment.res.xml

```
<SOAP-ENV:Body>
  <lbapi:PaymentResponse>
    <ret>0</ret>
  </lbapi:PaymentResponse>
</SOAP-ENV:Body>
```

api3.ExternCheckPayment.req.xml

```
<SOAP-ENV:Body>
  <lbapi:ExternCheckPayment>
    <receipt></receipt>
  </lbapi:ExternCheckPayment>
</SOAP-ENV:Body>
```

api3.ExternCheckPayment.res.xml

```
<SOAP-ENV:Body>
  <lbapi:ExternCheckPaymentResponse>
    <ret>
      <pay>
        <recordid>0</recordid>
        <agrmid>0</agrmid>
        <modperson>0</modperson>
        <currid>0</currid>
        <orderid>0</orderid>
        <status>0</status>
        <amount>0.0</amount>
        <paydate></paydate>
        <localdate></localdate>
        <canceldate></canceldate>
        <receipt></receipt>
        <comment></comment>
      </pay>
      <amountcurr>0.0</amountcurr>
      <ordernum>0</ordernum>
      <uid>0</uid>
      <operid>0</operid>
      <curr symb></curr symb>
      <uname></uname>
      <agrm></agrm>
      <mgr></mgr>
    </ret>
  </lbapi:ExternCheckPaymentResponse>
</SOAP-ENV:Body>
```

```
<opername></opername>  
</ret>  
</lbapi:ExternCheckPaymentResponse>  
</SOAP-ENV:Body>
```

api3. ExternPayment.req.xml

```
<SOAP-ENV:Body>  
<lbapi:ExternPayment>  
<id>0</id>  
<str></str>  
<val>  
<recordid>0</recordid>  
<agrmid>0</agrmid>  
<modperson>0</modperson>  
<currid>0</currid>  
<orderid>0</orderid>  
<status>0</status>  
<amount>0.0</amount>  
<paydate></paydate>  
<localdate></localdate>  
<canceldate></canceldate>  
<receipt></receipt>  
<comment></comment>  
</val>  
</lbapi:ExternPayment>  
</SOAP-ENV:Body>
```

api3.ExternPayment.res.xml

```
<SOAP-ENV:Body>  
<lbapi:ExternPaymentResponse>  
<ret>0</ret>  
</lbapi:ExternPaymentResponse>  
</SOAP-ENV:Body>
```

api3. ExternCancelPayment.req.xml

```
<SOAP-ENV:Body>  
<lbapi:ExternCancelPayment>  
<receipt></receipt>
```

```
</lbapi:ExternCancelPayment>  
</SOAP-ENV:Body>
```

api3.ExternCancelPayment.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:ExternCancelPaymentResponse>  
    <ret>0</ret>  
  </lbapi:ExternCancelPaymentResponse>  
</SOAP-ENV:Body>
```

api3.delVgroup.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:delVgroup>  
    <id>0</id>  
  </lbapi:delVgroup>  
</SOAP-ENV:Body>
```

api3.delVgroup.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:delVgroupResponse>  
    <ret>0</ret>  
  </lbapi:delVgroupResponse>  
</SOAP-ENV:Body>
```

api3.getPayments.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:getPayments>  
    <flt>  
      <activated>0</activated>  
      <addresstype>1</addresstype>  
      <agentid>0</agentid>  
      <agrmid>0</agrmid>  
      <archive>0</archive>  
      <appid>0</appid>  
      <asnum>0</asnum>
```

```
<blocked>0</blocked>
<catid>0</catid>
<category>-1</category>
<curid>-1</curid>
<common>0</common>
<defaultonly>0</defaultonly>
<deviceid>0</deviceid>
<direction>0</direction>
<docid>0</docid>
<durfrom>0</durfrom>
<durto>0</durto>
<groupid>0</groupid>
<groups>0</groups>
<istemplate>0</istemplate>
<mgrid>-1</mgrid>
<needcalc>-1</needcalc>
<nodata>0</nodata>
<notgroups>0</notgroups>
<onfly>0</onfly>
<operid>0</operid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<payed>-1</payed>
<parentid>-1</parentid>
<personid>0</personid>
<pgnum>0</pgnum>
<pgsize>0</pgsize>
<port>0</port>
<postmanid>0</postmanid>
<proto>0</proto>
<recordid>0</recordid>
<repdetail>0</repdetail>
<repnum>0</repnum>
<setid>0</setid>
<shape>0</shape>
<showdefault>0</showdefault>
<tarid>0</tarid>
<parenttarid>-1</parenttarid>
<taridprev>0</taridprev>
<tartype>0</tartype>
<type>0</type>
<ugroups>0</ugroups>
<unavail>-1</unavail>
<userid>0</userid>
<vgid>0</vgid>
```

```
<payhistory>0</payhistory>
<hasregistry>-1</hasregistry>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<address></address>
<addresscode></addresscode>
<agrmnum></agrmnum>
<ani></ani>
<cardkey></cardkey>
<code></code>
<descr></descr>
<dstip></dstip>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<email></email>
<inn></inn>
<kpp></kpp>
<ip></ip>
<login></login>
<name></name>
<namesound></namesound>
<numfrom></numfrom>
<numto></numto>
<phone></phone>
<receipt></receipt>
<responsible></responsible>
<serno></serno>
<status></status>
<tardescr></tardescr>
<telstaff></telstaff>
<telnum></telnum>
<vglogin></vglogin>
<uuid></uuid>
<searchtempl>
<tplname></tplname>
<tablename></tablename>
<field></field>
<condition></condition>
<value></value>
<logic></logic>
</searchtempl>
</flt>
</lbapi:getPayments>
```



```
</SOAP-ENV:Body>
```

api3.getPayments.res.xml

```
<SOAP-ENV:Body>
  <lbapi:getPaymentsResponse>
    <ret>
      <pay>
        <recordid>0</recordid>
        <agrmid>0</agrmid>
        <modperson>0</modperson>
        <currid>0</currid>
        <orderid>0</orderid>
        <status>0</status>
        <classid>0</classid>
        <amount>0.0</amount>
        <fromagrmid>0</fromagrmid>
        <revno>0</revno>
        <revisions>0</revisions>
        <classname></classname>
        <paydate></paydate>
        <localdate></localdate>
        <canceldate></canceldate>
        <perioddate></perioddate>
        <receipt></receipt>
        <comment></comment>
      </pay>
      <bsodoc>0</bsodoc>
      <timestamp>0</timestamp>
      <local-timestamp>0</local-timestamp>
      <amountcurr>0.000000</amountcurr>
      <ordernum>0</ordernum>
      <uid>0</uid>
      <operid>0</operid>
      <curr symb></curr symb>
      <uname></uname>
      <agrm></agrm>
      <mgr></mgr>
      <mgrdescr></mgrdescr>
      <opername></opername>
      <login></login>
    </ret>
  </lbapi:getPaymentsResponse>
```

```
</SOAP-ENV:Body>
```

api3.getAccountsAddonsStaff.req.xml

```
<SOAP-ENV:Body>
<lbapi:getAccountsAddonsStaff>
  <flt>
    <activated>0</activated>
    <addresstype>1</addresstype>
    <agentid>0</agentid>
    <agrmid>0</agrmid>
    <archive>0</archive>
    <appid>0</appid>
    <asnum>0</asnum>
    <blocked>0</blocked>
    <catid>0</catid>
    <category>-1</category>
    <curid>-1</curid>
    <common>0</common>
    <defaultonly>0</defaultonly>
    <deviceid>0</deviceid>
    <direction>0</direction>
    <docid>0</docid>
    <durfrom>0</durfrom>
    <durto>0</durto>
    <groupid>0</groupid>
    <groups>0</groups>
    <istemplate>0</istemplate>
    <mgrid>-1</mgrid>
    <needcalc>-1</needcalc>
    <nodata>0</nodata>
    <notgroups>0</notgroups>
    <onfly>0</onfly>
    <operid>0</operid>
    <orderid>0</orderid>
    <ordernum>0</ordernum>
    <payed>-1</payed>
    <parentid>-1</parentid>
    <personid>0</personid>
    <pgnum>0</pgnum>
    <pgsize>0</pgsize>
    <port>0</port>
    <postmanid>0</postmanid>
    <proto>0</proto>
```

```
<recordid>0</recordid>
<repdetail>0</repdetail>
<repnum>0</repnum>
<setid>0</setid>
<shape>0</shape>
<showdefault>0</showdefault>
<tarid>0</tarid>
<parenttarid>-1</parenttarid>
<taridprev>0</taridprev>
<tartype>0</tartype>
<type>0</type>
<ugroups>0</ugroups>
<unavail>-1</unavail>
<userid>0</userid>
<vgid>0</vgid>
<payhistory>0</payhistory>
<hasregistry>-1</hasregistry>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<address></address>
<addresscode></addresscode>
<agrmnum></agrmnum>
<ani></ani>
<cardkey></cardkey>
<code></code>
<descr></descr>
<dstip></dstip>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<email></email>
<inn></inn>
<kpp></kpp>
<ip></ip>
<login></login>
<name></name>
<namesound></namesound>
<numfrom></numfrom>
<numto></numto>
<phone></phone>
<receipt></receipt>
<responsible></responsible>
<serno></serno>
<status></status>
```

```
<tardescr></tardescr>
<telstaff></telstaff>
<telnum></telnum>
<vglogin></vglogin>
<uuid></uuid>
<searchtempl>
  <tplname></tplname>
  <tablename></tablename>
  <field></field>
  <condition></condition>
  <value></value>
  <logic></logic>
</searchtempl>
</flt>
</lbapi:getAccountsAddonsStaff>
</SOAP-ENV:Body>
```

api3.getAccountsAddonsStaff.res.xml

```
<SOAP-ENV:Body>
  <lbapi:getAccountsAddonsStaffResponse>
    <ret>
      <idx>0</idx>
      <name></name>
      <value></value>
    </ret>
  </lbapi:getAccountsAddonsStaffResponse>
</SOAP-ENV:Body>
```

api3.getAccountsAddonsSet.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getAccountsAddonsSet>
    <flt>
      <activated>0</activated>
      <addresstype>1</addresstype>
      <agentid>0</agentid>
      <agrmid>0</agrmid>
      <archive>0</archive>
      <appid>0</appid>
      <asnum>0</asnum>
```

```
<blocked>0</blocked>
<catid>0</catid>
<category>-1</category>
<curid>-1</curid>
<common>0</common>
<defaultonly>0</defaultonly>
<deviceid>0</deviceid>
<direction>0</direction>
<docid>0</docid>
<durfrom>0</durfrom>
<durto>0</durto>
<groupid>0</groupid>
<groups>0</groups>
<istemplate>0</istemplate>
<mgrid>-1</mgrid>
<needcalc>-1</needcalc>
<nodata>0</nodata>
<notgroups>0</notgroups>
<onfly>0</onfly>
<operid>0</operid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<payed>-1</payed>
<parentid>-1</parentid>
<personid>0</personid>
<pgnum>0</pgnum>
<pgsize>0</pgsize>
<port>0</port>
<postmanid>0</postmanid>
<proto>0</proto>
<recordid>0</recordid>
<repdetail>0</repdetail>
<repnum>0</repnum>
<setid>0</setid>
<shape>0</shape>
<showdefault>0</showdefault>
<tarid>0</tarid>
<parenttarid>-1</parenttarid>
<taridprev>0</taridprev>
<tartype>0</tartype>
<type>0</type>
<ugroups>0</ugroups>
<unavail>-1</unavail>
<userid>0</userid>
<vgid>0</vgid>
```

```
<payhistory>0</payhistory>
<hasregistry>-1</hasregistry>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<address></address>
<addresscode></addresscode>
<agrmnum></agrmnum>
<ani></ani>
<cardkey></cardkey>
<code></code>
<descr></descr>
<dstip></dstip>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<email></email>
<inn></inn>
<kpp></kpp>
<ip></ip>
<login></login>
<name></name>
<namesound></namesound>
<numfrom></numfrom>
<numto></numto>
<phone></phone>
<receipt></receipt>
<responsible></responsible>
<serno></serno>
<status></status>
<tardescr></tardescr>
<telstaff></telstaff>
<telnum></telnum>
<vglogin></vglogin>
<uuid></uuid>
<searchtempl>
  <tplname></tplname>
  <tablename></tablename>
  <field></field>
  <condition></condition>
  <value></value>
  <logic></logic>
</searchtempl>
</flt>
</lbapi:getAccountsAddonsSet>
```

```
</SOAP-ENV:Body>
```

api3.getAccountsAddonsSet.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:getAccountsAddonsSetResponse>  
    <ret>  
      <addon>  
        <type>0</type>  
        <name></name>  
        <descr></descr>  
      </addon>  
      <staff>  
        <idx>0</idx>  
        <name></name>  
        <value></value>  
      </staff>  
    </ret>  
  </lbapi:getAccountsAddonsSetResponse>  
</SOAP-ENV:Body>
```

api3.insupdAccountsAddonsSet.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:insupdAccountsAddonsSet>  
    <isInsert>0</isInsert>  
    <val>  
      <addon>  
        <type>0</type>  
        <name></name>  
        <descr></descr>  
      </addon>  
      <staff>  
        <idx>0</idx>  
        <name></name>  
        <value></value>  
      </staff>  
    </val>  
  </lbapi:insupdAccountsAddonsSet>  
</SOAP-ENV:Body>
```

api3.insupdAccountsAddonsSet.res.xml

```
<SOAP-ENV:Body>  
  <lbapi:insupdAccountsAddonsSetResponse>  
    <ret></ret>  
  </lbapi:insupdAccountsAddonsSetResponse>  
</SOAP-ENV:Body>
```

api3.getAgreementsAddonsStaff.req.xml

```
<SOAP-ENV:Body>  
  <lbapi:getAgreementsAddonsStaff>  
    <flt>  
      <activated>0</activated>  
      <addresstype>1</addresstype>  
      <agentid>0</agentid>  
      <agrmid>0</agrmid>  
      <archive>0</archive>  
      <appid>0</appid>  
      <asnum>0</asnum>  
      <blocked>0</blocked>  
      <catid>0</catid>  
      <category>-1</category>  
      <curid>-1</curid>  
      <common>0</common>  
      <defaultonly>0</defaultonly>  
      <deviceid>0</deviceid>  
      <direction>0</direction>  
      <docid>0</docid>  
      <durfrom>0</durfrom>  
      <durto>0</durto>  
      <groupid>0</groupid>  
      <groups>0</groups>  
      <istemplate>0</istemplate>  
      <mgrid>-1</mgrid>  
      <needcalc>-1</needcalc>  
      <nodata>0</nodata>  
      <notgroups>0</notgroups>  
      <onfly>0</onfly>  
      <operid>0</operid>  
      <orderid>0</orderid>  
      <ordernum>0</ordernum>  
      <payed>-1</payed>  
      <parentid>-1</parentid>
```



```
<personid>0</personid>
<pgnum>0</pgnum>
<pgsize>0</pgsize>
<port>0</port>
<postmanid>0</postmanid>
<proto>0</proto>
<recordid>0</recordid>
<repdetail>0</repdetail>
<repnum>0</repnum>
<setid>0</setid>
<shape>0</shape>
<showdefault>0</showdefault>
<tarid>0</tarid>
<parenttarid>-1</parenttarid>
<taridprev>0</taridprev>
<tartype>0</tartype>
<type>0</type>
<ugroups>0</ugroups>
<unavail>-1</unavail>
<userid>0</userid>
<vgid>0</vgid>
<payhistory>0</payhistory>
<hasregistry>-1</hasregistry>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<address></address>
<addresscode></addresscode>
<agrmnum></agrmnum>
<ani></ani>
<cardkey></cardkey>
<code></code>
<descr></descr>
<dstip></dstip>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<email></email>
<inn></inn>
<kpp></kpp>
<ip></ip>
<login></login>
<name></name>
<namesound></namesound>
<numfrom></numfrom>
```

```
<numto></numto>
<phone></phone>
<receipt></receipt>
<responsible></responsible>
<serno></serno>
<status></status>
<tardescr></tardescr>
<telstaff></telstaff>
<telnum></telnum>
<vglogin></vglogin>
<uuid></uuid>
<searchtempl>
  <tplname></tplname>
  <tablename></tablename>
  <field></field>
  <condition></condition>
  <value></value>
  <logic></logic>
</searchtempl>
</flt>
</lbapi:getAgreementsAddonsStaff>
</SOAP-ENV:Body>
```

api3.getAgreementsAddonsStaff.res.xml

```
<SOAP-ENV:Body>
  <lbapi:getAgreementsAddonsStaffResponse>
    <ret>
      <idx>0</idx>
      <name></name>
      <value></value>
    </ret>
  </lbapi:getAgreementsAddonsStaffResponse>
</SOAP-ENV:Body>
```

api3.getAgreementsAddonsSet.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getAgreementsAddonsSet>
    <flt>
      <activated>0</activated>
      <addresstype>1</addresstype>
      <agentid>0</agentid>
    </flt>
  </lbapi:getAgreementsAddonsSet>
</SOAP-ENV:Body>
```

```
<agrmid>0</agrmid>
<archive>0</archive>
<appid>0</appid>
<asnum>0</asnum>
<blocked>0</blocked>
<catid>0</catid>
<category>-1</category>
<curid>-1</curid>
<common>0</common>
<defaultonly>0</defaultonly>
<deviceid>0</deviceid>
<direction>0</direction>
<docid>0</docid>
<durfrom>0</durfrom>
<durto>0</durto>
<groupid>0</groupid>
<groups>0</groups>
<istemplate>0</istemplate>
<mgrid>-1</mgrid>
<needcalc>-1</needcalc>
<nodata>0</nodata>
<notgroups>0</notgroups>
<onfly>0</onfly>
<operid>0</operid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<payed>-1</payed>
<parentid>-1</parentid>
<personid>0</personid>
<pgnum>0</pgnum>
<pgsize>0</pgsize>
<port>0</port>
<postmanid>0</postmanid>
<proto>0</proto>
<recordid>0</recordid>
<repdetail>0</repdetail>
<replibnum>0</replibnum>
<setid>0</setid>
<shape>0</shape>
<showdefault>0</showdefault>
<tarid>0</tarid>
<parenttarid>-1</parenttarid>
<taridprev>0</taridprev>
<tartype>0</tartype>
<type>0</type>
```

```
<ugroups>0</ugroups>
<unavail>-1</unavail>
<userid>0</userid>
<vgid>0</vgid>
<payhistory>0</payhistory>
<hasregistry>-1</hasregistry>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<address></address>
<addresscode></addresscode>
<agrmnum></agrmnum>
<ani></ani>
<cardkey></cardkey>
<code></code>
<descr></descr>
<dstip></dstip>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<email></email>
<inn></inn>
<kpp></kpp>
<ip></ip>
<login></login>
<name></name>
<namesound></namesound>
<numfrom></numfrom>
<numto></numto>
<phone></phone>
<receipt></receipt>
<responsible></responsible>
<serno></serno>
<status></status>
<tardescr></tardescr>
<telstaff></telstaff>
<telnum></telnum>
<vglogin></vglogin>
<uuid></uuid>
<searchtempl>
<tplname></tplname>
<tablename></tablename>
<field></field>
<condition></condition>
<value></value>
```

```
<logic></logic>
</searchtempl>
</flt>
</lbapi:getAgreementsAddonsSet>
</SOAP-ENV:Body>
```

api3.getAgreementsAddonsSet.res.xml

```
<SOAP-ENV:Body>
<lbapi:getAgreementsAddonsSetResponse>
<ret>
<addon>
<type>0</type>
<name></name>
<descr></descr>
</addon>
<staff>
<idx>0</idx>
<name></name>
<value></value>
</staff>
</ret>
</lbapi:getAgreementsAddonsSetResponse>
</SOAP-ENV:Body>
```

api3.insupdAgreementsAddonsSet.req.xml

```
<SOAP-ENV:Body>
<lbapi:insupdAgreementsAddonsSet>
<isInsert>0</isInsert>
<val>
<addon>
<type>0</type>
<name></name>
<descr></descr>
</addon>
<staff>
<idx>0</idx>
<name></name>
<value></value>
</staff>
</val>
```

```
</lbapi:insupdAgreementsAddonsSet>  
</SOAP-ENV:Body>
```

api3.insupdAgreementsAddonsSet.res.xml

```
<SOAP-ENV:Body>  
<lbapi:insupdAgreementsAddonsSetResponse>  
<ret></ret>  
</lbapi:insupdAgreementsAddonsSetResponse>  
</SOAP-ENV:Body>
```

api3.getVgroupsAddonsStaff.req.xml

```
<SOAP-ENV:Body>  
<lbapi:getVgroupsAddonsStaff>  
<flt>  
<activated>0</activated>  
<addresstype>1</addresstype>  
<agentid>0</agentid>  
<agrmid>0</agrmid>  
<archive>0</archive>  
<appid>0</appid>  
<asnum>0</asnum>  
<blocked>0</blocked>  
<catid>0</catid>  
<category>-1</category>  
<curid>-1</curid>  
<common>0</common>  
<defaultonly>0</defaultonly>  
<deviceid>0</deviceid>  
<direction>0</direction>  
<docid>0</docid>  
<durfrom>0</durfrom>  
<durto>0</durto>  
<groupid>0</groupid>  
<groups>0</groups>  
<istemplate>0</istemplate>  
<mgrid>-1</mgrid>  
<needcalc>-1</needcalc>  
<nodata>0</nodata>  
<notgroups>0</notgroups>  
<onfly>0</onfly>  
<operid>0</operid>
```

```
<orderid>0</orderid>
<ordernum>0</ordernum>
<payed>-1</payed>
<parentid>-1</parentid>
<personid>0</personid>
<pgnum>0</pgnum>
<pgsize>0</pgsize>
<port>0</port>
<postmanid>0</postmanid>
<proto>0</proto>
<recordid>0</recordid>
<repdetail>0</repdetail>
<repnum>0</repnum>
<setid>0</setid>
<shape>0</shape>
<showdefault>0</showdefault>
<tarid>0</tarid>
<parenttarid>-1</parenttarid>
<taridprev>0</taridprev>
<tartype>0</tartype>
<type>0</type>
<ugroups>0</ugroups>
<unavail>-1</unavail>
<userid>0</userid>
<vgid>0</vgid>
<payhistory>0</payhistory>
<hasregistry>-1</hasregistry>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<address></address>
<addresscode></addresscode>
<agrmnum></agrmnum>
<ani></ani>
<cardkey></cardkey>
<code></code>
<descr></descr>
<dstip></dstip>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<email></email>
<inn></inn>
<kpp></kpp>
<ip></ip>
```

```
<login></login>
<name></name>
<namesound></namesound>
<numfrom></numfrom>
<numto></numto>
<phone></phone>
<receipt></receipt>
<responsible></responsible>
<serno></serno>
<status></status>
<tardescr></tardescr>
<telstaff></telstaff>
<telnum></telnum>
<vglogin></vglogin>
<uuid></uuid>
<searchtempl>
  <tplname></tplname>
  <tablename></tablename>
  <field></field>
  <condition></condition>
  <value></value>
  <logic></logic>
</searchtempl>
</flt>
</lbapi:getVgroupsAddonsStaff>
</SOAP-ENV:Body>
```

api3.getVgroupsAddonsStaff.res.xml

```
<SOAP-ENV:Body>
  <lbapi:getVgroupsAddonsStaffResponse>
    <ret>
      <idx>0</idx>
      <name></name>
      <value></value>
    </ret>
  </lbapi:getVgroupsAddonsStaffResponse>
</SOAP-ENV:Body>
```

api3.getVgroupsAddonsSet.req.xml

```
<SOAP-ENV:Body>
  <lbapi:getVgroupsAddonsSet>
```



```
<flt>
<activated>0</activated>
<addresstype>1</addresstype>
<agentid>0</agentid>
<agrmid>0</agrmid>
<archive>0</archive>
<appid>0</appid>
<asnum>0</asnum>
<blocked>0</blocked>
<catid>0</catid>
<category>-1</category>
<curid>-1</curid>
<common>0</common>
<defaultonly>0</defaultonly>
<deviceid>0</deviceid>
<direction>0</direction>
<docid>0</docid>
<durfrom>0</durfrom>
<durto>0</durto>
<groupid>0</groupid>
<groups>0</groups>
<istemplate>0</istemplate>
<mgrid>-1</mgrid>
<needcalc>-1</needcalc>
<nodata>0</nodata>
<notgroups>0</notgroups>
<onfly>0</onfly>
<operid>0</operid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<payed>-1</payed>
<parentid>-1</parentid>
<personid>0</personid>
<pgnum>0</pgnum>
<pgsize>0</pgsize>
<port>0</port>
<postmanid>0</postmanid>
<proto>0</proto>
<recordid>0</recordid>
<repdetail>0</repdetail>
<repnum>0</repnum>
<setid>0</setid>
<shape>0</shape>
<showdefault>0</showdefault>
<tarid>0</tarid>
```

```
<parenttarid>-1</parenttarid>
<taridprev>0</taridprev>
<tartype>0</tartype>
<type>0</type>
<ugroups>0</ugroups>
<unavail>-1</unavail>
<userid>0</userid>
<vgid>0</vgid>
<payhistory>0</payhistory>
<hasregistry>-1</hasregistry>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<address></address>
<addresscode></addresscode>
<agrmnum></agrmnum>
<ani></ani>
<cardkey></cardkey>
<code></code>
<descr></descr>
<dstip></dstip>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<email></email>
<inn></inn>
<kpp></kpp>
<ip></ip>
<login></login>
<name></name>
<namesound></namesound>
<numfrom></numfrom>
<numto></numto>
<phone></phone>
<receipt></receipt>
<responsible></responsible>
<serno></serno>
<status></status>
<tardescr></tardescr>
<telstaff></telstaff>
<telnum></telnum>
<vglogin></vglogin>
<uuid></uuid>
<searchtempl>
<tplname></tplname>
```

```
<tablename></tablename>  
<field></field>  
<condition></condition>  
<value></value>  
<logic></logic>  
</searchtempl>  
</flt>  
</lbapi:getVgroupsAddonsSet>  
</SOAP-ENV:Body>
```

api3.getVgroupsAddonsSet.res.xml

```
<SOAP-ENV:Body>  
<lbapi:getVgroupsAddonsSetResponse>  
<ret>  
<addon>  
<type>0</type>  
<agentid>0</agentid>  
<name></name>  
<descr></descr>  
<agent></agent>  
</addon>  
<staff>  
<idx>0</idx>  
<name></name>  
<value></value>  
</staff>  
</ret>  
</lbapi:getVgroupsAddonsSetResponse>  
</SOAP-ENV:Body>
```

api3.insupdVgroupsAddonsSet.req.xml

```
<SOAP-ENV:Body>  
<lbapi:insupdVgroupsAddonsSet>  
<isInsert>0</isInsert>  
<val>  
<addon>  
<type>0</type>  
<agentid>0</agentid>  
<name></name>  
<descr></descr>  
<agent></agent>
```

```
</addon>
<staff>
  <idx>0</idx>
  <name></name>
  <value></value>
</staff>
</val>
</lbapi:insupdVgroupsAddonsSet>
</SOAP-ENV:Body>
```

api3.insupdVgroupsAddonsSet.res.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdVgroupsAddonsSetResponse>
    <ret></ret>
  </lbapi:insupdVgroupsAddonsSetResponse>
</SOAP-ENV:Body>
```

api3.blkClientVgroup.req.xml

```
<SOAP-ENV:Body>
  <lbapi:blkClientVgroup>
    <id>0</id>
    <blk>0</blk>
    <state></state>
  </lbapi:blkClientVgroup>
</SOAP-ENV:Body>
```

api3.blkClientVgroup.res.xml

```
<SOAP-ENV:Body>
  <lbapi:blkClientVgroupResponse>
    <ret>0</ret>
  </lbapi:blkClientVgroupResponse>
</SOAP-ENV:Body>
```

api3.blkVgroup.req.xml

```
<SOAP-ENV:Body>
  <lbapi:blkVgroup>
```

```
<id>0</id>
<blk>0</blk>
<state></state>
</lbapi:blkVgroup>
</SOAP-ENV:Body>
```

api3.blkVgroup.res.xml

```
<SOAP-ENV:Body>
  <lbapi:blkVgroupResponse>
    <ret>0</ret>
  </lbapi:blkVgroupResponse>
</SOAP-ENV:Body>
```

api3.insupdUsboxService.req.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdUsboxService>
    <isInsert>0</isInsert>
    <val>
      <servid>0</servid>
      <vgid>0</vgid>
      <tarid>0</tarid>
      <catid>0</catid>
      <needcalc>0</needcalc>
      <noduplicate>0</noduplicate>
      <mul>0.0</mul>
      <personid></personid>
      <comment></comment>
      <created></created>
      <timefrom></timefrom>
      <timeto></timeto>
    </val>
  </lbapi:insupdUsboxService>
</SOAP-ENV:Body>
```

api3.insupdUsboxService.res.xml

```
<SOAP-ENV:Body>
  <lbapi:insupdUsboxServiceResponse>
    <ret>0</ret>
  </lbapi:insupdUsboxServiceResponse>
```

```
</SOAP-ENV:Body>
```

api3.getTarCategories.req.xml

```
<SOAP-ENV:Body>
<lbapi:getTarCategories>
<id>0</id>
<flt>
<activated>0</activated>
<addresstype>1</addresstype>
<agentid>0</agentid>
<agrmid>0</agrmid>
<archive>0</archive>
<appid>0</appid>
<asnum>0</asnum>
<blocked>0</blocked>
<catid>0</catid>
<category>-1</category>
<curid>-1</curid>
<common>0</common>
<defaultonly>0</defaultonly>
<deviceid>0</deviceid>
<direction>0</direction>
<docid>0</docid>
<durfrom>0</durfrom>
<durto>0</durto>
<groupid>0</groupid>
<groups>0</groups>
<istemplate>0</istemplate>
<mgrid>-1</mgrid>
<needcalc>-1</needcalc>
<nodata>0</nodata>
<notgroups>0</notgroups>
<onfly>0</onfly>
<operid>0</operid>
<orderid>0</orderid>
<ordernum>0</ordernum>
<payed>-1</payed>
<parentid>-1</parentid>
<personid>0</personid>
<pgnum>0</pgnum>
<pgsize>0</pgsize>
<port>0</port>
<postmanid>0</postmanid>
```

```
<proto>0</proto>
<recordid>0</recordid>
<repdetail>0</repdetail>
<repnum>0</repnum>
<setid>0</setid>
<shape>0</shape>
<showdefault>0</showdefault>
<tarid>0</tarid>
<parenttarid>-1</parenttarid>
<taridprev>0</taridprev>
<tartype>0</tartype>
<type>0</type>
<ugroups>0</ugroups>
<unavail>-1</unavail>
<userid>0</userid>
<vgid>0</vgid>
<payhistory>0</payhistory>
<hasregistry>-1</hasregistry>
<amountfrom>0.000000</amountfrom>
<amountto>0.000000</amountto>
<address></address>
<addresscode></addresscode>
<agrmnum></agrmnum>
<ani></ani>
<cardkey></cardkey>
<code></code>
<descr></descr>
<dstip></dstip>
<dtcreated></dtcreated>
<dtactivated></dtactivated>
<dtfrom></dtfrom>
<dtto></dtto>
<email></email>
<inn></inn>
<kpp></kpp>
<ip></ip>
<login></login>
<name></name>
<namesound></namesound>
<numfrom></numfrom>
<numto></numto>
<phone></phone>
<receipt></receipt>
<responsible></responsible>
<serno></serno>
```

```
<status></status>
<tardescr></tardescr>
<telstaff></telstaff>
<telnum></telnum>
<vglogin></vglogin>
<uuid></uuid>
<searchtempl>
<tplname></tplname>
<tablename></tablename>
<field></field>
<condition></condition>
<value></value>
<logic></logic>
</searchtempl>
</flt>
</lbapi:getTarCategories>
</SOAP-ENV:Body>
```

api3.getTarCategories.res.xml

```
<SOAP-ENV:Body>
<lbapi:getTarCategoriesResponse>
<ret>
<tarid>0</tarid>
<catid>0</catid>
<catidx>0</catidx>
<catidxmaster>0</catidxmaster>
<includes>0</includes>
<common>0</common>
<disprior>0</disprior>
<archive>0</archive>
<operid>0</operid>
<freeseconds>0</freeseconds>
<minchargedur>0</minchargedur>
<roundseconds>0</roundseconds>
<enabled>1</enabled>
<available>0</available>
<permabove>0.000000</permabove>
<above>0.0</above>
<descr></descr>
<uuid></uuid>
<script></script>
</ret>
```



```
</lbapi:getTarCategoriesResponse>  
</SOAP-ENV:Body>
```