**Maindb**

**БАЗА ДАННЫХ**

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**Листов 133**

2023

**АННОТАЦИЯ**

В документе приведены сведения, необходимые пользователю (it-специалист) для работы с базой данных maindb.

Перечислены назначения и условия применения, описана подготовка к работе, приведены примеры работы с maindb. В приложениях к документу содержатся справочники API интерфейса, представлений и процедур.

Настоящий документ пересматривается на регулярной основе. Содержание данного документа может быть изменено без предварительного уведомления пользователей с последующим размещением на соответствующей странице сайта https://softdoc.taxsee.com/

Правообладатель: ООО «ТАКСТЕЛЕКОМ» (ИНН 4501170000).

**СОДЕРЖАНИЕ**

Введение 4

1. Назначение и условия применения 5

2. Подготовка к работе 6

2.1 Состав и содержание дистрибутивного носителя данных 6

2.2 Порядок загрузки данных и программ 6

2.3 Термины используемые в системе 6

2.4 Порядок проверки работоспособности 6

2.4.1 Подключения 7

2.4.2 Авторизация 9

2.4.3 Работа с представлениями 13

2.4.4 Работа с процедурами 18

3. Описание операций 21

3.1 Функциональные характеристики 21

3.2 Выполнение программы 23

3.2.1 Работа с географическими данными 24

3.2.2 Работа с автомобилями 31

3.2.3 Работа с телефонами клиентов 36

3.2.4 Работа с тарифами 40

3.2.5 Работа с лицевыми счетами 44

3.2.6 Работа с реестром записей телефонных переговоров 45

4. Аварийные ситуации 48

Приложение 1. API интерфейс к базе данных Maindb 49

**ВВЕДЕНИЕ**

Требования настоящего документа применяются при:

1. предварительных комплексных испытаниях;
2. опытной эксплуатации;
3. приемочных испытаниях;
4. промышленной эксплуатации.
   1. **НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ**

|  |  |
| --- | --- |
| Наименование базы данных: | Maindb |
| Назначение, область применения  и функциональные возможности базы данных: | средство для эффективного хранения большого объема информации, позволяющее посредством СУБД и иных программных средств автоматизировать работу операторов по приему и обработке заказов и иной информации, поступающей по сети Интернет, работу по оформлению и сопровождению заказов на услуги (включая, но не ограничиваясь) по перевозке пассажиров, в том числе по заказу, перевозке грузов, курьерской доставке, запуску двигателя автомобиля, по вызову эвакуатора, личного водителя, аварийного комиссара, по буксировке, перевозке животных, погрузочно-разгрузочным работам; сопровождения и изменения данных заказов, а также управлять работой компании, обеспечивая учет и обработку данных в сфере грузовых и легковых автоперевозок, доставки и прочих услуг |
| Содержит следующие самостоятельные материалы: | - географические точки территории Российской Федерации, Республики Болгария, Украины, Грузии, Республики Беларусь, Республики Казахстан (наименование городов, известных мест, улиц, домов;  - марки легковых автомобилей;  - номера телефонов, с который поступают заказы на перевозки;  - список тарифов по географическим точкам;  - список лицевых счетов перевозчиков и клиентов;  - реестр записей телефонных переговоров, зафиксированных в процессе формирования заказа |

Работа с maindb происходит через API интерфейс. Это позволяет скрыть от приложений внутреннее устройство источников данных, делает доступ к данным и сервисам единообразным, решает задачи контроля доступа и кеширования данных.

* 1. **ПОДГОТОВКА К РАБОТЕ**

**2.1 Состав и содержание дистрибутивного носителя данных**

Дистрибутивного носителя база данных не имеет.

**2.2 Порядок загрузки данных и программ**

maindb не требует загрузки на ЭВМ пользователя. Взаимодействие с приложением происходит через API интерфейс.

**2.3 Термины используемые в системе**

*Предтсавление (view)* - обьект используемый для получения данных. Например, для чтения справочников.

*Процедура (proc)* - объект использующийся для изменения данных и/или выполнения действий. Например, добавить запись в справочник или назначить водителя на заказ.

*Класс (class)* - это группа Представлений и Процедур объединенных общей задачей. Например класс T\_AUTO содержит представления и процедуры предназначеные для работы с автомобилями.

**2.4 Порядок проверки работоспособности**

Для проверки работоспособности нужно установить соединение с maindb сервером и запросить данные через API интерфейс. Убедившись, что данные получены, можно считать приложение maindb работоспособным.

API интерфейс работает на основе SOAP протокола. Поэтому, для взаимодействия с maindb, в качестве эмулятора предлагается использовать SoapUI. Приложение SoapUI можно скачать по ссылке <https://www.soapui.org/tools/soapui/>. Достаточно бесплатной Open Source версии.

**2.4.1 Подключение**

В приложении SoapUI создаем новый SOAP проект (рисунок 1)

Меню: File -> New SOAP Project

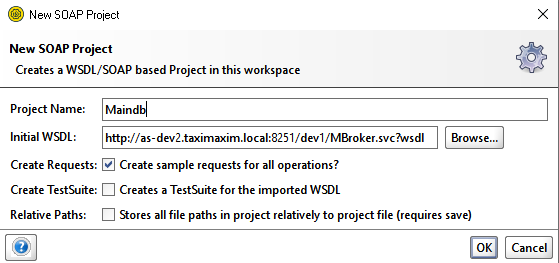


рисунок 1 - создание нового SOAP проекта

В поле Initial WSDL указываем адрес Maindb, в данном примере:

<http://as-dev2.taximaxim.local:8251/dev1/MBroker.svc?wsdl>.

Фактический адрес нужно запросить у администратора системы (здесь и далее см. «Связь пользователей с технической поддержкой» на сайте https://softdoc.taxsee.com/

В случае успешного подключения, SoapUI отобразит дерево доступных API интерфейсов предоставляемых Maindb (рисунок 2).



рисунок 2 - дерево доступных API интерфейсов

**2.4.2 Авторизация**

Для выполнения запросов к Maindb необходимо авторизоваться. Для этого используется пара логин-пароль и API интерфейс **GetContextV2**. Связка логин-пароль выдается администратором системы.

Вызываем GetContextV2, передав ему логин и пароль пользователя, в ответ получаем ID контекста (рисунок 3).

ID контекста используется при последующей работе с системой. Используя его система идентифицирует пользователя.

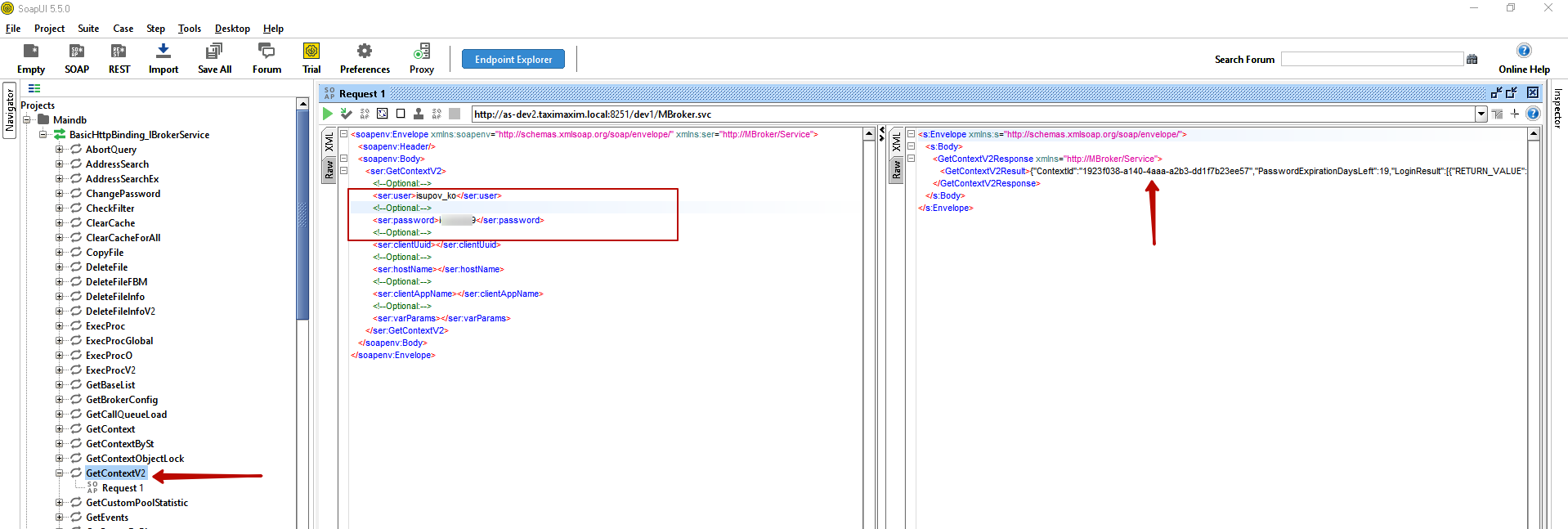


рисунок 3 - запрос на получение контекста

Запрос:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://MBroker/Service">

<soapenv:Header/>

<soapenv:Body>

<ser:GetContextV2>

**<ser:user>login</ser:user>**

**<ser:password>password</ser:password>**

<ser:clientUuid></ser:clientUuid>

<ser:hostName></ser:hostName>

<ser:clientAppName></ser:clientAppName>

<ser:varParams></ser:varParams>

</ser:GetContextV2>

</soapenv:Body>

</soapenv:Envelope>

Ответ:

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

<s:Body>

<GetContextV2Response xmlns="http://MBroker/Service">

<GetContextV2Result>{

**"ContextId": "1923f038-a140-4aaa-a2b3-dd1f7b23ee57",**

"PasswordExpirationDaysLeft": 19,

"LoginResult": [

{

"RETURN\_VALUE": 0

},

{

"LOGINRES": "LOGINOK"

},

{

"COMPNAME": "AS-DEV2"

},

{

"FIO": "Исупов Кирилл Олегович"

},

{

"IDBASE": 178

},

{

"IDWORKERGROUP": 27

},

{

"GROUPS": "|\_AccessManager|\_Admin|\_InterfaceMaxim|\_WebPanelView|\_ITDivision|\_TrainingSpecialistCC|\_SystemTrainingLMS|\_i18nTranslate|\_getListContexts"

},

{

"SESSIONID": 74844168

},

{

"LINE": "SIP/3461"

},

{

"ATSTYPE": 1

},

{

"WPTYPE": 2

},

{

"IDWORKER": 31916

},

{

"IDWORKPLACE": null

},

{

"ATSQUEREG": 0

},

{

"FREQ": null

},

{

"ATSADDRTAPI": "10.1.5.8:5042"

},

{

"WEBARMTAPI": "{\"stompTcp\":\"tcp://10.45.144.62:61612\", \"stompWs\": \"wss://aist.taxsee.com/tapi/office\"}"

},

{

"PERSONID": "1408d45c-c56c-11e9-80ef-0cc47a14e3f4"

},

{

"IDORGANIZATION": 312

},

{

"DOMAINLOGIN": "isupov\_ko"

},

{

"SIPADDR": "10.1.5.8"

},

{

"IDORGANIZATIONTYPE": 0

},

{

"IDSHARD": 2

},

{

"TIMEZONE": "Asia/Yekaterinburg"

},

{

"CNTCONNECT": 0

},

{

"WEBPHONE": "{\"realm\": \"pbx-office.taximaxim.ru\", \"serverURI\": \"wss://pbx.taxsee.com\"}"

}

]

}</GetContextV2Result>

</GetContextV2Response>

</s:Body>

</s:Envelope>

**2.4.3 Работа с представлениями**

Работа ведется с использования API интерфейса **GetViewV2**

В качестве входных параметров интерфейсу передается контекст, имя класса, имя представления и опционально параметры фильтра.

Для получения контекста используется метод **GetContextV2.**

Подробнее в разделе 2.4.2 Авторизация.

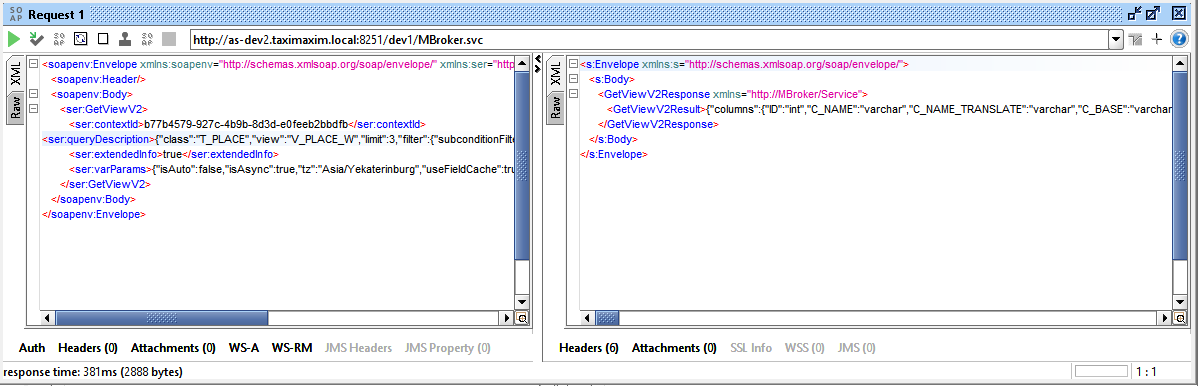


рисунок 5 - вызов представления V\_PLACE\_W (Населенные пункты)

Пример получения списка из трех населенных пунктов Курганской области.

Запрос:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://MBroker/Service">

<soapenv:Header/>

<soapenv:Body>

<ser:GetViewV2>

<ser:contextId>9f4914d6-7cf1-4609-945e-53a373e3e9c8</ser:contextId>

<ser:queryDescription>{

"class": "T\_PLACE",

"view": "V\_PLACE\_W",

"limit": 3,

"filter": {

"subconditionFilter": {

"C\_COUNTRY": {

"eq": "Россия"

},

"C\_REGION": {

"eq": "Курганская область"

}

}

}

}</ser:queryDescription>

<ser:extendedInfo>true</ser:extendedInfo>

<ser:varParams>{"isAuto":false,"isAsync":false,"tz":"Asia/Yekaterinburg","useFieldCache":true}</ser:varParams>

</ser:GetViewV2>

</soapenv:Body>

</soapenv:Envelope>

Ответ:

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

<s:Body>

<GetViewV2Response xmlns="http://MBroker/Service">

<GetViewV2Result>{

"columns": {

"ID": "int",

"C\_NAME": "varchar",

"C\_NAME\_TRANSLATE": "varchar",

"C\_BASE": "varchar",

"C\_PLACE\_TYPE": "varchar",

"C\_COUNTRY": "varchar",

"C\_REGION": "varchar",

"C\_SUBREGION": "varchar",

"C\_FEDERAL\_DISTRICT": "varchar",

"C\_LATITUDE": "double",

"C\_LONGITUDE": "double",

"C\_COORD": "varchar",

"C\_IS\_ADDR\_INFO": "varchar",

"C\_PLACE\_MAIN": "varchar",

"ID\_BASE": "int",

"C\_POPULATION": "int",

"C\_SYN\_COUNT": "bigint",

"ID\_SUBREGION": "int",

"ID\_COUNTRY": "int",

"C\_TIME\_ZONE": "varchar",

"ID\_TIME\_ZONE": "int",

"C\_PHONE\_CODE": "int"

},

"data": [

{

"ID": 219,

"C\_NAME": "Ялым",

"C\_NAME\_TRANSLATE": "Ялым",

"C\_BASE": null,

"C\_PLACE\_TYPE": "село",

"C\_COUNTRY": "Россия",

"C\_REGION": "Курганская область",

"C\_SUBREGION": "Притобольный район",

"C\_FEDERAL\_DISTRICT": "Уральский",

"C\_LATITUDE": 54.7686,

"C\_LONGITUDE": 65.08761,

"C\_COORD": "54.7686, 65.08761",

"C\_IS\_ADDR\_INFO": "Нет",

"C\_PLACE\_MAIN": null,

"ID\_BASE": null,

"C\_POPULATION": 500,

"C\_SYN\_COUNT": 0,

"ID\_SUBREGION": 1656,

"ID\_COUNTRY": 1,

"C\_TIME\_ZONE": "Asia/Yekaterinburg",

"ID\_TIME\_ZONE": 244,

"C\_PHONE\_CODE": null

},

{

"ID": 240,

"C\_NAME": "Яблочное",

"C\_NAME\_TRANSLATE": "Яблочное",

"C\_BASE": null,

"C\_PLACE\_TYPE": "село",

"C\_COUNTRY": "Россия",

"C\_REGION": "Курганская область",

"C\_SUBREGION": "Варгашинский район",

"C\_FEDERAL\_DISTRICT": "Уральский",

"C\_LATITUDE": 55.76293,

"C\_LONGITUDE": 66.16932,

"C\_COORD": "55.76293, 66.16932",

"C\_IS\_ADDR\_INFO": "Нет",

"C\_PLACE\_MAIN": null,

"ID\_BASE": null,

"C\_POPULATION": 300,

"C\_SYN\_COUNT": 0,

"ID\_SUBREGION": 1916,

"ID\_COUNTRY": 1,

"C\_TIME\_ZONE": "Asia/Yekaterinburg",

"ID\_TIME\_ZONE": 244,

"C\_PHONE\_CODE": null

},

{

"ID": 68693,

"C\_NAME": "Большое Щучье",

"C\_NAME\_TRANSLATE": "Большое Щучье",

"C\_BASE": "Ф Максим Мокроусово",

"C\_PLACE\_TYPE": "деревня",

"C\_COUNTRY": "Россия",

"C\_REGION": "Курганская область",

"C\_SUBREGION": "Мокроусовский район",

"C\_FEDERAL\_DISTRICT": "Уральский",

"C\_LATITUDE": 55.664629,

"C\_LONGITUDE": 67.426359,

"C\_COORD": "55.664629, 67.426359",

"C\_IS\_ADDR\_INFO": "Нет",

"C\_PLACE\_MAIN": null,

"ID\_BASE": 5992,

"C\_POPULATION": 75,

"C\_SYN\_COUNT": 1,

"ID\_SUBREGION": 818,

"ID\_COUNTRY": 1,

"C\_TIME\_ZONE": "Asia/Yekaterinburg",

"ID\_TIME\_ZONE": 244,

"C\_PHONE\_CODE": null

}

],

"sql": "SELECT ID,C\_NAME,C\_NAME\_TRANSLATE,C\_BASE,C\_PLACE\_TYPE,C\_COUNTRY,C\_REGION,C\_SUBREGION,C\_FEDERAL\_DISTRICT,C\_LATITUDE,C\_LONGITUDE,C\_COORD,C\_IS\_ADDR\_INFO,C\_PLACE\_MAIN,ID\_BASE,C\_POPULATION,C\_SYN\_COUNT,ID\_SUBREGION,ID\_COUNTRY,C\_TIME\_ZONE,ID\_TIME\_ZONE,C\_PHONE\_CODE FROM city.V\_PLACE\_W AS vw WHERE (((C\_COUNTRY=@Param0)) AND ((C\_REGION=@Param1))) limit 3 \n/\*\nParam0 = Россия\nParam1 = Курганская область\n\*/"

}

</GetViewV2Result>

</GetViewV2Response>

</s:Body>

</s:Envelope

**2.4.4 Работа с процедурами**

Работа ведется с использования API интерфейса **ExecProcV2.**

В качестве входных параметров интерфейсу передается контекст, имя класса, имя процедуры и список параметров.

Для получения контекста используется метод **GetContextV2.** Подробнее в разделе 2.4.2 Авторизация.

Список параметров индивидуален для каждой процедуры. Его можно получить, используя метод **GetProcParam**.

Пример получения списка параметров для процедуры autoClassSet класса T\_AUTO\_CLASS

Запрос параметров:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://MBroker/Service">

<soapenv:Header/>

<soapenv:Body>

<ser:**GetProcParam**>

<ser:contextId>893643ed-7145-41fe-9b70-2be4e8e2a0dd</ser:contextId>

<ser:classCode>**T\_AUTO\_CLASS**</ser:classCode>

<ser:procName>**autoClassSet**</ser:procName>

</ser:**GetProcParam**>

</soapenv:Body>

</soapenv:Envelope>

Ответ:

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

<s:Body>

<GetProcParamResponse xmlns="http://MBroker/Service">

<GetProcParamResult>{

"C\_CLASS":"T\_AUTO\_CLASS",

"C\_PROC":"autoClassSet",

"C\_PARAMS":[

{"ID":0,"C\_PROC":"autoClassSet","C\_CLASS":"T\_AUTO\_CLASS","C\_CODE":"@Name","C\_ORDER":259142,"C\_SIZE":200,"C\_DEFAULT":"**C\_NAME**","C\_TYPE":"STRING","C\_DIRECT":"","C\_DATE\_CREATE":"2020-08-19T10:08:47.6318097"},

{"ID":0,"C\_PROC":"autoClassSet","C\_CLASS":"T\_AUTO\_CLASS","C\_CODE":"@IDAutoCategory","C\_ORDER":259143,"C\_SIZE":0,"C\_DEFAULT":"**ID\_AUTO\_CATEGORY**","C\_TYPE":"INTEGER","C\_DIRECT":"","C\_DATE\_CREATE":"2020-08-19T10:08:47.6318097"},

{"ID":0,"C\_PROC":"autoClassSet","C\_CLASS":"T\_AUTO\_CLASS","C\_CODE":"@Description","C\_ORDER":259144,"C\_SIZE":0,"C\_DEFAULT":"**C\_DESCRIPTION**","C\_TYPE":"STRING","C\_DIRECT":"","C\_DATE\_CREATE":"2020-08-19T10:08:47.6318097"},

{"ID":0,"C\_PROC":"autoClassSet","C\_CLASS":"T\_AUTO\_CLASS","C\_CODE":"@Deleted","C\_ORDER":259145,"C\_SIZE":0,"C\_DEFAULT":"**C\_DELETED**","C\_TYPE":"INTEGER","C\_DIRECT":"","C\_DATE\_CREATE":"2020-08-19T10:08:47.6318097"}

]}

</GetProcParamResult>

</GetProcParamResponse>

</s:Body>

</s:Envelope>

В ответе видно, что для вызова процедуры GetProcParam ей следует передать четыре входных параметра:

**C\_NAME**

**ID\_AUTO\_CATEGORY**

**C\_DESCRIPTION**

**C\_DELETED**

Следует уточнить, что для всех процедур существует параметр ID, который не отображается в ответе GetProcParam, но его следует указывать при вызове.

Если указать ID=NULL - процедура создаст новую запись и вернет ее ID.

Если в ID передать номер существующей записи, то процедура отредактирует указанную запись.

Пример вызова процедуры **autoClassSet** класса **T\_AUTO\_CLASS** для создания нового класса автомобиля (ID=NULL)

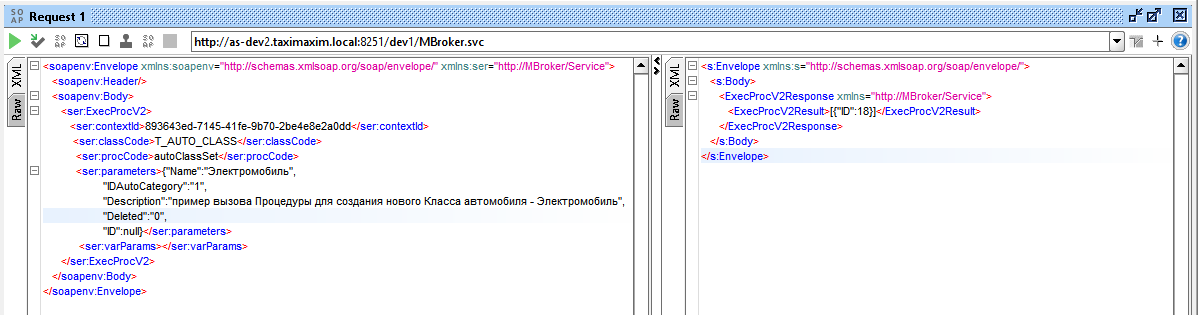


рисунок 4 - вызов процедуры autoClassSet

Запрос на добавление нового класса автомобиля в справочник Классы автомобилей).

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://MBroker/Service">

<soapenv:Header/>

<soapenv:Body>

<ser:ExecProcV2>

<ser:contextId>893643ed-7145-41fe-9b70-2be4e8e2a0dd</ser:contextId>

<ser:classCode>**T\_AUTO\_CLASS**</ser:classCode>

<ser:procCode>**autoClassSet**</ser:procCode>

<ser:parameters>{

"**Name**":"Электромобиль",

"**IDAutoCategory**":"1",

"**Description**":"новый класс автомобиля - Электромобиль",

"**Deleted**":"0",

"**ID**":null}</ser:parameters>

<ser:varParams></ser:varParams>

</ser:ExecProcV2>

</soapenv:Body>

</soapenv:Envelope>

Ответ на запрос (ID=18 говорит о том что новой записи справочника присвоен код 18)

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

<s:Body>

<ExecProcV2Response xmlns="http://MBroker/Service">

<ExecProcV2Result>[{"**ID**":18}]</ExecProcV2Result>

</ExecProcV2Response>

</s:Body>

</s:Envelope>

**3 ОПИСАНИЕ ОПЕРАЦИЙ**

**3.1 Функциональные характеристики**

Функциональным назначением maindb является хранение большого объема информации, позволяющее посредством СУБД и иных программных средств автоматизировать работу операторов по приему и обработке заказов и иной информации, поступающей по сети Интернет, работу по оформлению и сопровождению заказов на услуги (включая, но не ограничиваясь) по перевозке пассажиров, в том числе по заказу, перевозке грузов, курьерской доставке, запуску двигателя автомобиля, по вызову эвакуатора, личного водителя, аварийного комиссара, по буксировке, перевозке животных, погрузочно-разгрузочным работам; сопровождения и изменения данных заказов, а также управлять работой компании, обеспечивая учет и обработку данных в сфере грузовых и легковых автоперевозок, доставки и прочих услуг.

Функциональность реализована в формате Представлений и Процедур. Доступ к Представлениям и Процедурам осуществляется через собственный API.

Пдробное описание API приведено в Приложении 1.

Основные функциональные блоки maindb.

* Города и географические данные

работа со справочником стран

работа со справочником городов

работа со справочником улиц

работа со справочником домов

работа с границами районов и населенных пунктов

* Автомобили

работа со справочниками автомобилей

работа с подключениями автомобилей

работа с фотоконтролями автомобилей

* Клиенты

работа со справочником клиентов

работа с промокодами

работа с «черными списками»

рейтинги клиентов

* Тарифы

работа с тарифами для заказов на различные услуги

* Лицевые счета

работа с историей изменений ЛС

работа с фискальными документами

работа с задолжностями

* Телефония

работа с архивом телефонных переговоров

работа с архивом отправленных сообщений (push/sms)

Представления и Процедуры, реализующие функциональность каждого блока представлены в разделах 3.2.1 - 3.2.5.

**3.2 Выполнение программы**

Сценарий использования maindb

* Клиентское приложение, используя API интерфейс, формирует запрос к maindb. В зависимости от задачи, в запросе указывается необходимое представление или процедура с параметрами.
* Maindb обрабатывает запрос - проводит работу с данными или выполняет операцию.
* Maindb возвращает ответ клиентскому приложению.

**3.2.1 Работа с географическими данными**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Описание класса** | **Тип** | **Процедура | Представление** | **Описание** |
| T\_ADDRESS | Адреса | view | V\_ADDRESS | Полный список |
| T\_ADDRESS | Адреса | view | V\_ADDRESS\_HINT | Подсказки |
| T\_ADDRESS | Адреса | view | V\_ADDRESS\_W | Быстрые адреса |
| T\_HOUSE | Дома | view | V\_HOUSE | Полный список |
| T\_HOUSE | Дома | view | V\_HOUSE\_W | Рабочие |
| T\_ZONE\_GROUP | Зоны | view | V\_ZONE\_GROUP | Полный список |
| T\_ZONE\_GROUP | Зоны | view | V\_ZONE\_GROUP\_W | Рабочие |
| T\_ADDRESS\_TYPE | Категории адресов | view | V\_ADDRESS\_TYPE | Категории адресов |
| T\_ADDRESS\_MEET\_POINT | Места встречи | view | V\_ADDRESS\_MEET\_POINT | Полный список |
| T\_ADDRESS\_MEET\_POINT | Места встречи | view | V\_ADDRESS\_MEET\_POINT\_GEO | Для проверки в GH |
| T\_ADDRESS\_MEET\_POINT | Места встречи | view | V\_ADDRESS\_MEET\_POINT\_W | Рабочие |
| T\_PLACE | Населенные пункты | view | V\_PLACE | Полный список |
| T\_PLACE | Населенные пункты | view | V\_PLACE\_4ORD | Представление для урезанного списка населенных пунктов |
| T\_PLACE | Населенные пункты | view | V\_PLACE\_4TRANSFER | Города с районами и БА |
| T\_PLACE | Населенные пункты | view | V\_PLACE\_ARM | Для АРМ |
| T\_PLACE | Населенные пункты | view | V\_PLACE\_FOR\_DRIVER\_PLAN | Список городов для формы планирования поездок водителей |
| T\_PLACE | Населенные пункты | view | V\_PLACE\_FOR\_SEARCH | Служебная для поиска |
| T\_PLACE | Населенные пункты | view | V\_PLACE\_IS\_BASE | Города с подразделениями |
| T\_PLACE | Населенные пункты | view | V\_PLACE\_MBTILE | Для вытягивания городов с границей mbtile билдером |
| T\_PLACE | Населенные пункты | view | V\_PLACE\_W | Рабочие |
| T\_PLACE | Населенные пункты | view | V\_PLACE\_W\_ADDR | Города с адресами |
| T\_PLACE | Населенные пункты | view | V\_PLACE\_W\_ADDR\_CHECK\_BASE | Города с адресами с проверкой прав |
| T\_ADDRESS\_PLACES | Привязка быстрых адресов к населенным пунктам | view | V\_ADDRESS\_PLACES | Полный список |
| T\_ZONES | Районы | view | V\_ZONES | Полный список |
| T\_ZONES | Районы | view | V\_ZONES\_4ALL | Рабочие для всех |
| T\_ZONES | Районы | view | V\_ZONES\_OUT | Межгород |
| T\_ZONES | Районы | view | V\_ZONES\_W | Рабочие |
| T\_ZONES | Районы | view | V\_ZONES\_W0 | Полный список |
| T\_ZONES | Районы | view | V\_ZONES\_WITH\_ACCESS | Рабочие с ограничениями |
| T\_SUBREGION | Регион.Районы | view | V\_SUBREGION | Полный список |
| T\_SUBREGION | Регион.Районы | view | V\_SUBREGION\_ALL | Полный список |
| T\_SUBREGION | Регион.Районы | view | V\_SUBREGION\_TRANSLATE | Список для перевода |
| T\_REGION | Регионы | view | V\_REGION | Полный список |
| T\_REGION | Регионы | view | V\_REGION\_LIC | Регионы выдачи лицензии |
| T\_REGION | Регионы | view | V\_REGION\_TRANSLATE | Список для перевода |
| T\_SYNONYM | Синонимы | view | V\_SYNONYM | Полный список |
| T\_SYNONYM | Синонимы | view | V\_SYNONYM\_ADDRESS | Синонимы быстрых адресов |
| T\_SYNONYM | Синонимы | view | V\_SYNONYM\_PLACE | Синонимы городов |
| T\_SYNONYM | Синонимы | view | V\_SYNONYM\_STREET | Синонимы улиц |
| T\_SYNONYM\_ADDRESS | Синонимы адресов | view | V\_SYNONYM\_ADDRESS | Синонимы быстрых адресов |
| T\_SYNONYM\_PLACE | Синонимы городов | view | V\_SYNONYM\_PLACE | Синонимы городов |
| T\_SYNONYM\_STREET | Синонимы улиц | view | V\_SYNONYM\_STREET | Синонимы улиц |
| T\_SYSTEM | Системные представления и операции | view | V\_PLACE\_W\_4ALL | Города |
| T\_COUNTRY | Страны | view | V\_COUNTRY | Полный список |
| T\_COUNTRY | Страны | view | V\_COUNTRY\_4ALL | Полный список |
| T\_HOUSE\_TYPE | Типы домов | view | V\_HOUSE\_TYPE | Полный список |
| T\_ADDRESS\_MEET\_TYPE | Типы мест встречи | view | V\_ADDRESS\_MEET\_TYPE | Полный список |
| T\_PLACE\_TYPE | Типы населенных пунктов | view | V\_PLACE\_TYPE | Типы населенных пунктов |
| T\_STREET\_TYPE | Типы улиц | view | V\_STREET\_TYPE | Полный список |
| T\_STREETS | Улицы | view | V\_STREETS | Полный список |
| T\_STREETS | Улицы | view | V\_STREETS\_W | Рабочие |
| T\_FEDERAL\_DISTRICT | Федеральные округа | view | V\_FEDERAL\_DISTRICT | Полный список |
| T\_ADDRESS | Адреса | proc | Addrblock | Блокировать выбранные |
| T\_ADDRESS | Адреса | proc | AddrEditType | Изменить категорию |
| T\_ADDRESS | Адреса | proc | AddrSet | Добавить |
| T\_ADDRESS | Адреса | proc | AddrSet | Изменить |
| T\_ADDRESS | Адреса | proc | ImportCsv | Импортировать из файла |
| T\_ADDRESS | Адреса | proc | SetMapDisplay | Изменить отображение на карте при поиске |
| T\_COUNTRY\_GROUP | Группы стран | proc | CountryGroupDel | Удалить |
| T\_COUNTRY\_GROUP | Группы стран | proc | CountryGroupSet | Добавить |
| T\_COUNTRY\_GROUP | Группы стран | proc | CountryGroupSet | Изменить |
| T\_HOUSE | Дома | proc | house\_apply\_bindings\_to | Применить автопривязку выбранных домов к районам |
| T\_HOUSE | Дома | proc | HouseDel | Удалить |
| T\_HOUSE | Дома | proc | HouseSet | Добавить |
| T\_HOUSE | Дома | proc | HouseSet | Изменить |
| T\_ZONE\_GROUP | Зоны | proc | ZoneGroupDel | Удалить |
| T\_ZONE\_GROUP | Зоны | proc | ZoneGroupSet | Добавить |
| T\_ZONE\_GROUP | Зоны | proc | ZoneGroupSet | Изменить |
| T\_ADDRESS\_TYPE | Категории адресов | proc | AddressTypeDel | Удалить |
| T\_ADDRESS\_TYPE | Категории адресов | proc | AddressTypeSet | Добавить |
| T\_ADDRESS\_TYPE | Категории адресов | proc | AddressTypeSet | Изменить |
| T\_ADDRESS\_TYPE | Категории адресов | proc | AddressTypeSetSubst | Изменить подсказку |
| T\_ADDRESS\_MEET\_POINT | Места встречи | proc | AddrMeetPointDel | Удалить |
| T\_ADDRESS\_MEET\_POINT | Места встречи | proc | AddrMeetPointSet | Добавить |
| T\_ADDRESS\_MEET\_POINT | Места встречи | proc | AddrMeetPointSet | Изменить |
| T\_PLACE | Населенные пункты | proc | clear\_city\_border | Удалить границы города |
| T\_PLACE | Населенные пункты | proc | getIdPlaceByCoordinates | Функция для брокера по координатам возвращает id\_place, которому принадлежат координаты |
| T\_PLACE | Населенные пункты | proc | PlaceDel | Удалить |
| T\_PLACE | Населенные пункты | proc | PlacePolygon | Показать WKT полигон |
| T\_PLACE | Населенные пункты | proc | PlaceSet | Добавить |
| T\_PLACE | Населенные пункты | proc | PlaceSet | Изменить |
| T\_PLACE | Населенные пункты | proc | reset\_city\_border | Сбросить границы города |
| T\_ADDRESS\_NOT\_FOUND | Не найденные адреса по запросам | proc | AddressNotFoundSetReason | Установить причину |
| T\_ADDRESS\_NOT\_FOUND\_REASON | Не найденные адреса по запросам. Причины | proc | AddressNotFoundReasonBlock | Блокировать |
| T\_ADDRESS\_NOT\_FOUND\_REASON | Не найденные адреса по запросам. Причины | proc | AddressNotFoundReasonSet | Добавить |
| T\_ADDRESS\_NOT\_FOUND\_REASON | Не найденные адреса по запросам. Причины | proc | AddressNotFoundReasonSet | Изменить |
| T\_ZONE\_ACCESS | Ограничения доступности района | proc | zoneAccess | Добавить |
| T\_ZONE\_ACCESS | Ограничения доступности района | proc | zoneAccess | Изменить |
| T\_ZONE\_ACCESS | Ограничения доступности района | proc | zoneAccessDel | Удалить |
| T\_HOUSE\_ENTRANCE | Подъезды | proc | HouseEntranceSet | Изменить |
| T\_ADDRESS\_PLACES | Привязка быстрых адресов к населенным пунктам | proc | AddrPlacesDel | Удалить |
| T\_ADDRESS\_PLACES | Привязка быстрых адресов к населенным пунктам | proc | AddrPlacesSet | Добавить |
| T\_ZONES | Районы | proc | ZoneDel | Удалить |
| T\_ZONES | Районы | proc | zonegeoreset | Сбросить геометрию |
| T\_ZONES | Районы | proc | ZoneHomeClear | Удалить привязку домов |
| T\_ZONES | Районы | proc | ZoneSet | Добавить |
| T\_ZONES | Районы | proc | ZoneSet | Изменить |
| T\_SUBREGION | Регион.Районы | proc | SubregionDel | Удалить |
| T\_SUBREGION | Регион.Районы | proc | SubregionSet | Добавить |
| T\_SUBREGION | Регион.Районы | proc | SubregionSet | Изменить |
| T\_REGION | Регионы | proc | RegionSet | Добавить |
| T\_REGION | Регионы | proc | RegionSet | Изменить |
| T\_SYNONYM | Синонимы | proc | SynonymDel | Удалить |
| T\_SYNONYM | Синонимы | proc | SynonymSet | Добавить |
| T\_SYNONYM | Синонимы | proc | SynonymSet | Изменить |
| T\_SYNONYM\_ADDRESS | Синонимы адресов | proc | SynonymDel | Удалить |
| T\_SYNONYM\_ADDRESS | Синонимы адресов | proc | SynonymSet | Добавить |
| T\_SYNONYM\_ADDRESS | Синонимы адресов | proc | SynonymSet | Изменить |
| T\_SYNONYM\_PLACE | Синонимы городов | proc | SynonymDel | Удалить |
| T\_SYNONYM\_PLACE | Синонимы городов | proc | SynonymSet | Добавить |
| T\_SYNONYM\_PLACE | Синонимы городов | proc | SynonymSet | Изменить |
| T\_SYNONYM\_STREET | Синонимы улиц | proc | SynonymDel | Удалить |
| T\_SYNONYM\_STREET | Синонимы улиц | proc | SynonymSet | Добавить |
| T\_SYNONYM\_STREET | Синонимы улиц | proc | SynonymSet | Изменить |
| T\_ZONES\_JOIN | Соседние районы | proc | ZoneJoinDel | Удалить |
| T\_ZONES\_JOIN | Соседние районы | proc | ZoneJoinSet | Добавить |
| T\_ZONES\_JOIN | Соседние районы | proc | ZoneJoinSet | Изменить |
| T\_COUNTRY | Страны | proc | CountryDel | Удалить |
| T\_COUNTRY | Страны | proc | CountrySet | Добавить |
| T\_COUNTRY | Страны | proc | CountrySet | Изменить |
| T\_HOUSE\_TYPE | Типы домов | proc | HouseTypeDel | Удалить |
| T\_HOUSE\_TYPE | Типы домов | proc | HouseTypeSet | Добавить |
| T\_HOUSE\_TYPE | Типы домов | proc | HouseTypeSet | Изменить |
| T\_ADDRESS\_MEET\_TYPE | Типы мест встречи | proc | AddrMeetTypeDel | Удалить |
| T\_ADDRESS\_MEET\_TYPE | Типы мест встречи | proc | AddrMeetTypeSet | Добавить |
| T\_ADDRESS\_MEET\_TYPE | Типы мест встречи | proc | AddrMeetTypeSet | Изменить |
| T\_PLACE\_TYPE | Типы населенных пунктов | proc | PlaceTypeSet | Добавить |
| T\_PLACE\_TYPE | Типы населенных пунктов | proc | PlaceTypeSet | Изменить |
| T\_ZONE\_ACCESS\_TYPE | Типы ограничения доступности района | proc | zoneAccessType | Добавить |
| T\_ZONE\_ACCESS\_TYPE | Типы ограничения доступности района | proc | zoneAccessType | Изменить |
| T\_ZONE\_ACCESS\_TYPE | Типы ограничения доступности района | proc | zoneAccessTypeDel | Удалить |
| T\_HOUSE\_ENTRANCE\_TYPE | Типы подъездов | proc | HouseEntranceTypeSet | Добавить |
| T\_HOUSE\_ENTRANCE\_TYPE | Типы подъездов | proc | HouseEntranceTypeSet | Изменить |
| T\_STREET\_TYPE | Типы улиц | proc | StreetTypeDel | Удалить |
| T\_STREET\_TYPE | Типы улиц | proc | StreetTypeSet | Добавить |
| T\_STREET\_TYPE | Типы улиц | proc | StreetTypeSet | Изменить |
| T\_STREETS | Улицы | proc | ImportCsv | Импортировать из файла |
| T\_STREETS | Улицы | proc | StreetDel | Удалить |
| T\_STREETS | Улицы | proc | StreetSet | Добавить |
| T\_STREETS | Улицы | proc | StreetSet | Изменить |
| T\_STREETS | Улицы | proc | StreetVoiceSet | Настроить произношение |
| T\_FEDERAL\_DISTRICT | Федеральные округа | proc | FederalDistrictDel | Удалить |
| T\_FEDERAL\_DISTRICT | Федеральные округа | proc | FederalDistrictSet | Добавить |
| T\_FEDERAL\_DISTRICT | Федеральные округа | proc | FederalDistrictSet | Изменить |

**3.2.2 Работа с автомобилями**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Описание класса** | **Тип** | **Процедура | Представление** | **Описание** |
| T\_AUTO | Автомобили справочник | view | V\_AUTO | Полный список |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_4ALL | Полный список из проваливания |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_4FORMCHECK | Полный список |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_DBL | Дубли |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_DBL\_VIEW | Просмотр дублей |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_F | Уволены |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_FIND | Поиск автомобилей |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_MY | Созданные мной |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_MY\_TODAY | Созданные мной сегодня |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_REMIND | Осмотр в офисе |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_TRANSFER | Перевод из другого подразделения |
| T\_AUTO | Автомобили справочник | view | V\_AUTO\_W | Работают |
| T\_AUTO\_CLASS | Автомобили. Классы | view | V\_AUTO\_CLASS | Полный список |
| T\_AUTO\_CLASS | Автомобили. Классы | view | V\_AUTO\_CLASS\_W | Рабочие |
| T\_AUTO\_BRAND | Бренды автомобилей | view | V\_AUTO\_BRAND | Полный список |
| T\_DOP\_INFO\_SET | Доп. параметры (массив для авто) | view | V\_DOP\_INFO\_SET | Полный список |
| T\_DRIVER\_AUTO\_DOPINFO | Доп. параметры связки | view | V\_DRIVER\_AUTO\_DOPINFO | Полный список |
| T\_AUTO\_CATEGORY | Категории автомобилей | view | V\_AUTO\_CATEGORY | Полный список |
| T\_AUTO\_MARK | Марки автомобилей | view | V\_AUTO\_MARK | Полный список |
| T\_AUTO\_MARK | Марки автомобилей | view | V\_AUTO\_MARK\_AUDIO | Рабочие с озвучкой |
| T\_AUTO\_MARK | Марки автомобилей | view | V\_AUTO\_MARK\_W | Рабочие |
| T\_DRIVER\_CHECK\_QUEUE | Назначенные проверки | view | V\_DRIVER\_CHECK\_QUEUE | Полный список |
| T\_DRIVER\_CHECK\_QUEUE | Назначенные проверки | view | V\_DRIVER\_CHECK\_QUEUE\_W | В работе |
| T\_DRIVER\_CHECK\_RULES | Правила проверок | view | V\_DRIVER\_CHECK\_RULES | Полный список |
| T\_DRIVER\_CHECK\_RULES | Правила проверок | view | V\_DRIVER\_CHECK\_RULES\_W | Активные |
| T\_REGION\_AUTO\_CODE | Регионы. Коды автомобилей | view | V\_REGION\_AUTO\_CODE | Полный список |
| T\_RATING | Рейтинги | view | V\_DRIVERS\_RATING | Рейтинги водителей |
| T\_RATING | Рейтинги | view | V\_DRIVERS\_RATING\_POINTS | Рейтинги водителей (новые) |
| T\_RATING | Рейтинги | view | V\_DRIVERS\_RATING\_POINTS\_HISTORY | История изменения рейтинга водителей (новые) |
| T\_ADW\_STATUS | Состояния автомобилей | view | VA\_ADW\_STATUS | Архив состояния автомобиля |
| T\_ADW\_STATUS | Состояния автомобилей | view | VA\_ADW\_STATUS\_LIC | Архив состояния автомобиля (Таксопарк) |
| T\_ADW\_STATUS | Состояния автомобилей | view | VA\_ADW\_STATUS\_MAP | Архив состояния автомобиля (c правами для карты) |
| T\_SMENA\_TARIFF\_AUTO\_CLASS | Тариф смен. Классы автомобилей | view | V\_SMENA\_TARIFF\_AUTO\_CLASS | Полный список |
| T\_AUTO | Автомобили справочник | proc | AutoEdit | Изменить |
| T\_AUTO | Автомобили справочник | proc | AutoFind4Web | Найти авто из водительского кабинета |
| T\_AUTO | Автомобили справочник | proc | AutoSet | Добавить |
| T\_AUTO | Автомобили справочник | proc | AutoSet4Web | Добавить/изменить из водительского кабинета |
| T\_AUTO | Автомобили справочник | proc | AutoTransfer | Изменить подразделение |
| T\_AUTO | Автомобили справочник | proc | AutoTransferRent | Перевод в таксопарк на аренде |
| T\_AUTO | Автомобили справочник | proc | AutoUniqCheck | проверить авто |
| T\_AUTO | Автомобили справочник | proc | AutoUpdate | Назначить дату осмотра в офисе |
| T\_AUTO | Автомобили справочник | proc | GetAutoNumFromStr | Получить короткий номер(новая) |
| T\_AUTO | Автомобили справочник | proc | PhotoCheckLog | Назначить фотоконтроль |
| T\_AUTO\_CLASS | Автомобили. Классы | proc | autoClassDel | Удалить |
| T\_AUTO\_CLASS | Автомобили. Классы | proc | autoClassSet | Добавить |
| T\_AUTO\_CLASS | Автомобили. Классы | proc | autoClassSet | Изменить |
| T\_AUTO\_BRAND | Бренды автомобилей | proc | AutoBrandDel | Удалить |
| T\_AUTO\_BRAND | Бренды автомобилей | proc | AutoBrandSet | Добавить |
| T\_AUTO\_BRAND | Бренды автомобилей | proc | AutoBrandSet | Изменить |
| T\_DOP\_INFO\_SET | Доп. параметры (массив для авто) | proc | DopInfoSetAdd | Добавить доп.Инфо |
| T\_DOP\_INFO\_SET | Доп. параметры (массив для авто) | proc | DopInfoSetDel | Удалить доп.Инфо |
| T\_DRIVER\_AUTO\_DOPINFO | Доп. параметры связки | proc | InfoDispEdit | Доп.Инфо |
| T\_ORDERS\_ALL | Заказы (журнал) | proc | AutoExchangeBlockSet | Добавить автомобиль в черный список по банку |
| T\_ORDERS\_ALL | Заказы (журнал) | proc | DriverExchangeBlockSet | Добавить водителя в черный список по банку |
| T\_ORDERS\_CHECK | Заказы (проверка) | proc | AutoExchangeBlockSet | Добавить автомобиль в черный список по банку |
| T\_ORDERS\_CHECK | Заказы (проверка) | proc | DriverExchangeBlockSet | Добавить водителя в черный список по банку |
| T\_AUTO\_CATEGORY | Категории автомобилей | proc | AutoCategoryDel | Удалить |
| T\_AUTO\_CATEGORY | Категории автомобилей | proc | AutoCategorySet | Добавить |
| T\_AUTO\_CATEGORY | Категории автомобилей | proc | AutoCategorySet | Изменить |
| T\_AUTO\_MARK | Марки автомобилей | proc | AutoMarkSet | Добавить |
| T\_AUTO\_MARK | Марки автомобилей | proc | AutoMarkSet | Изменить |
| T\_DRIVER\_CHECK\_RULES | Правила проверок | proc | DriverCheckRules | Добавить |
| T\_DRIVER\_CHECK\_RULES | Правила проверок | proc | DriverCheckRules | Изменить |
| T\_DRIVER\_CHECK\_RULES | Правила проверок | proc | DriverCheckRulesDel | Удалить |
| T\_REGION\_AUTO\_CODE | Регионы. Коды автомобилей | proc | RegionAutoCodeDel | Удалить |
| T\_REGION\_AUTO\_CODE | Регионы. Коды автомобилей | proc | RegionAutoCodeSet | Добавить |
| T\_RATING | Рейтинги | proc | DriverRatingCalc | Расчёт рейтинга |
| T\_RATING | Рейтинги | proc | DriverRatingSave | Сохранение рейтинга водителей |
| T\_WORK\_NOTE | Служебные записки | proc | AutoExchangeBlockSet | Добавить автомобиль в черный список по банку |
| T\_WORK\_NOTE | Служебные записки | proc | DriverExchangeBlockSet | Добавить водителя в черный список по банку |
| T\_ORDERS\_STATUS | Состояния заказов | proc | ViolationSet | Добавить нарушение водителю |
| T\_SMENA\_TARIFF\_AUTO\_CLASS | Тариф смен. Классы автомобилей | proc | TariffAutoClassDel | Удалить |
| T\_SMENA\_TARIFF\_AUTO\_CLASS | Тариф смен. Классы автомобилей | proc | TariffAutoClassSet | Добавить |
| T\_SMENA\_TARIFF\_AUTO\_CLASS | Тариф смен. Классы автомобилей | proc | TariffAutoClassSet | Изменить |

**3.2.3 Работа с телефонами клиентов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Описание класса** | **Тип** | **Процедура | Представление** | **Описание** |
| T\_PROMO\_ACTIVATION | Активации промокодов | view | V\_PROMO\_ACTIVATION | Полный список |
| T\_PROMO\_ACTIVATION | Активации промокодов | view | V\_PROMO\_ACTIVATION\_P | Промоутеры |
| T\_PROMO\_ACTIVATION | Активации промокодов | view | V\_PROMO\_ACTIVATION\_REF | Полный список |
| T\_BLACK\_LIST\_LOG | Журнал добавления телефона в черный список | view | V\_BLACK\_LIST\_LOG | Полный список |
| T\_ADDRESS\_HISTORY\_OCCASIONAL | История адресов клиента | view | V\_ADDRESS\_HISTORY\_OCCASIONAL | Полный список |
| T\_CLIENT\_SEGMENT\_MONTHLY | История сегментов клиента | view | V\_CLIENT\_SEGMENT\_MONTHLY | Полный список |
| T\_CLPHONES | Клиент. Телефоны | view | V\_CLPHONES | Полный список |
| T\_CLIENT\_CHECK | Клиенты (проверка) | view | V\_CLIENT\_CHECK | Полный список |
| T\_CLIENT\_CHECK | Клиенты (проверка) | view | V\_CLIENT\_CHECK\_W | В работе |
| T\_CLFIS | Клиенты физ.лица | view | V\_CLFIS | Полный список |
| T\_CLFIS | Клиенты физ.лица | view | V\_CLFIS\_F | Блокированные |
| T\_CLFIS | Клиенты физ.лица | view | V\_CLFIS\_FIND\_FOR\_PERSON | Клиенты поиск по физ лицу |
| T\_CLFIS | Клиенты физ.лица | view | V\_CLFIS\_W | Рабочие |
| T\_CLUR | Клиенты юр.лица | view | V\_CLUR | Полный список |
| T\_CLUR | Клиенты юр.лица | view | V\_CLUR\_BASES | clur bases |
| T\_CLUR | Клиенты юр.лица | view | V\_CLUR\_F | Блокированные |
| T\_CLUR | Клиенты юр.лица | view | V\_CLUR\_HASH | Полный список с хэшем |
| T\_CLUR | Клиенты юр.лица | view | V\_CLUR\_LIC | Лицензиары |
| T\_CLUR | Клиенты юр.лица | view | V\_CLUR\_TRANSLATE | для перевода |
| T\_CLUR | Клиенты юр.лица | view | V\_CLUR\_W | Рабочие |
| T\_CAUSE\_BLACK\_LIST\_TUNE | Настройки причин добавления в черный список | view | V\_CAUSE\_BLACK\_LIST\_TUNE | Полный список |
| T\_PROMO\_CODE\_BASES | Подразделения действия промокода | view | V\_PROMO\_CODE\_BASES | Полный список |
| T\_CAUSE\_BLACK\_LIST | Причины добавления в черный список | view | V\_CAUSE\_BLACK\_LIST | Полный список |
| T\_PROMO\_CODE | Промокоды | view | V\_PROMO\_CODE | Полный список |
| T\_PROMO\_CODE | Промокоды | view | V\_PROMO\_CODE\_REF | Полный список |
| T\_PROMO\_CODE | Промокоды | view | V\_PROMO\_CODE\_W | Активные |
| T\_PROMO\_CODE | Промокоды | view | V\_PROMOCODE\_AUDIT | Аудит добавления кода |
| T\_RATING | Рейтинги | view | V\_CLIENT\_RATING | Рейтинги клиентов |
| T\_PROMO\_TYPE | Типы промокодов | view | V\_PROMO\_TYPE | Полный список |
| T\_PHONEDRV\_DISLIKE | Чёрный список | view | V\_PHONEDRV\_DISLIKE | Полный список |
| T\_ORDER\_PATTERN | Шаблоны заказов | view | V\_ORDER\_PATTERN | Полный список |
| T\_PROMO\_ACTIVATION | Активации промокодов | proc | PromoActCancel | Отменить активацию |
| T\_ADDRESS\_HISTORY\_OCCASIONAL | История адресов клиента | proc | AddressHistoryDel | Удалить |
| T\_CLPHONES | Клиент. Телефоны | proc | ClPhoneDelV2 | Удалить |
| T\_CLPHONES | Клиент. Телефоны | proc | ClPhoneSetV2 | Добавить |
| T\_CLPHONES | Клиент. Телефоны | proc | ClPhoneSetV2\_PG | ClPhoneSetV2\_PG |
| T\_CLPHONES | Клиент. Телефоны | proc | PhoneList | Изменить настройки |
| T\_CLIENT\_CHECK | Клиенты (проверка) | proc | ClientCheckClose | Закрыть |
| T\_CLIENT\_CHECK | Клиенты (проверка) | proc | ClientCheckSet | Изменить |
| T\_CLFIS | Клиенты физ.лица | proc | ClFisAddV3 | Добавить |
| T\_CLFIS | Клиенты физ.лица | proc | ClfisList | ClfisList |
| T\_CLFIS | Клиенты физ.лица | proc | ClFisSetV3 | Изменить |
| T\_CLFIS | Клиенты физ.лица | proc | ClFisTabDel | ClFisTabDel |
| T\_CLFIS | Клиенты физ.лица | proc | GeneratePromoterCode | Сгенерировать код промоутера |
| T\_CLFIS | Клиенты физ.лица | proc | up\_clfis\_jumporderset | up\_clfis\_jumporderset |
| T\_CLFIS | Клиенты физ.лица | proc | up\_clfis\_jumporderset2 | up\_clfis\_jumporderset2 |
| T\_CLFIS | Клиенты физ.лица | proc | up\_clfis\_jumporderstatusset | up\_clfis\_jumporderstatusset |
| T\_CLUR | Клиенты юр.лица | proc | ClUrSet | Добавить |
| T\_CLUR | Клиенты юр.лица | proc | ClUrSet | Изменить |
| T\_CAUSE\_BLACK\_LIST\_TUNE | Настройки причин добавления в черный список | proc | CauseBlackListTuneDel | Удалить |
| T\_CAUSE\_BLACK\_LIST\_TUNE | Настройки причин добавления в черный список | proc | CauseBlackListTuneSet | Добавить |
| T\_CAUSE\_BLACK\_LIST\_TUNE | Настройки причин добавления в черный список | proc | CauseBlackListTuneSet | Изменить |
| T\_CAUSE\_BLACK\_LIST | Причины добавления в черный список | proc | CauseBlackListDel | Удалить |
| T\_CAUSE\_BLACK\_LIST | Причины добавления в черный список | proc | CauseBlackListSet | Добавить |
| T\_CAUSE\_BLACK\_LIST | Причины добавления в черный список | proc | CauseBlackListSet | Изменить |
| T\_PROMO\_CODE | Промокоды | proc | PromoCode4Base | Промокоды для подразделения |
| T\_PROMO\_CODE | Промокоды | proc | PromoCodeActivateOper | Активаиция промокода оператором |
| T\_PROMO\_CODE | Промокоды | proc | PromoCodeBlocked | Заблокировать промокод |
| T\_PROMO\_CODE | Промокоды | proc | PromoCodeDel | Удалить |
| T\_PROMO\_CODE | Промокоды | proc | PromoCodeSet | Добавить |
| T\_PROMO\_CODE | Промокоды | proc | PromoCodeSet | Изменить |
| T\_RATING | Рейтинги | proc | ClientRatingSet\_PG | ClientRatingSet |
| T\_SMENAS | Смены | proc | ContractRegularPayRecalc | Повторный расчет биллинга |
| T\_PROMO\_TYPE | Типы промокодов | proc | PromoTypeDel | Удалить |
| T\_PROMO\_TYPE | Типы промокодов | proc | PromoTypeSet | Добавить |
| T\_PROMO\_TYPE | Типы промокодов | proc | PromoTypeSet | Изменить |
| T\_PHONEDRV\_DISLIKE | Чёрный список | proc | PhoneDrvDislikeDel | Удалить |
| T\_PHONEDRV\_DISLIKE | Чёрный список | proc | PhoneDrvDislikeSet | Добавить |
| T\_PHONEDRV\_DISLIKE | Чёрный список | proc | PhoneDrvDislikeSet | изменить |
| T\_ORDER\_PATTERN | Шаблоны заказов | proc | OrderPatternDel | Удалить |

**3.2.4 Работа с тарифами**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Описание класса** | **Тип** | **Процедура | Представление** | **Описание** |
| T\_TARIFF\_TUNE | Настройка тарифа | view | V\_TARIFF\_TUNE | Полный список |
| T\_TARIFF\_TIMETABLE | Периоды действия тарифа | view | V\_TARIFF\_TIMETABLE | Полный список |
| T\_TARIFF\_BASES\_SYNC | Синхронное изменение тарифов для подразделений | view | V\_TARIFF\_BASES\_SYNC | Полный список |
| T\_TARIFF\_DISABLE\_MARK | Тариф. Исключенные марки авто | view | V\_TARIFF\_DISABLE\_MARK | Полный список |
| T\_TARIFF\_AUTO\_CLASS | Тариф. Классы автомобилей | view | V\_TARIFF\_AUTO\_CLASS | Полный список |
| T\_TARIFF\_ZONES | Тариф. Маршруты | view | V\_TARIFF\_ZONES | Полный список |
| T\_SATELLIT\_OUTCITY\_ROUND | Тариф. Округлять до города | view | V\_SATELLIT\_OUTCITY\_ROUND | Полный список |
| T\_TARIFF | Тарифы | view | V\_TARIFF | Полный список |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_4DRV | Доступные водителям |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_4ORD | Рабочие без id\_base=0 |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_ARC | Архив тарифов |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_AUDIT | Аудит тарифов |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_BASIC | Базовые тарифы |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_BASIC\_RIVAL | Базовые тарифы(Перевозчики.Заказы.Пересчет цен) |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_INCREMENTS | Надбавки тарифов |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_TEMPLATE | Шаблоны тарифов |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_TEMPLATE\_ORDER | Шаблоны тарифов, для получения порядка |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_W | Рабочий |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_W\_4ALL | Тарифы для всех |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_W\_SEARCH | Поиск рабочего тарифа |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_W\_STAT | Статистика |
| T\_TARIFF | Тарифы | view | V\_TARIFF\_WCOMPARE | Сравнение тарифов |
| T\_SMENA\_TARIFF | Тарифы смен | view | V\_SMENA\_TARIFF | Полный список |
| T\_SMENA\_TARIFF | Тарифы смен | view | V\_SMENA\_TARIFF\_W | Рабочий |
| T\_TARIFF\_TAG | Теги тарифов | view | V\_TARIFF\_TAG | Теги тарифов |
| T\_TARIFF\_TYPE | Типы тарифов | view | V\_TARIFF\_TYPE | Полный список |
| T\_TARIFF\_TYPE | Типы тарифов | view | V\_TARIFF\_TYPE\_4ALL | Полный список для всех |
| T\_TARIFF\_TYPE\_DESCRIPTION | Типы тарифов. Описания | view | V\_TARIFF\_TYPE\_DESCRIPTION | Полный список |
| T\_TARIFF\_DESCR\_TEMPLATE | Шаблоны описаний тарифов | view | V\_TARIFF\_DESCR\_TEMPLATE | Полный список |
| T\_TARIFF\_TUNE | Настройка тарифа | proc | TariffTuneCopy | Копировать |
| T\_TARIFF\_TUNE | Настройка тарифа | proc | TariffTuneDel | Удалить |
| T\_TARIFF\_TUNE | Настройка тарифа | proc | TariffTunePriceDrv | Доплаты водителю |
| T\_TARIFF\_TUNE | Настройка тарифа | proc | TariffTuneSet | Добавить |
| T\_TARIFF\_TUNE | Настройка тарифа | proc | TariffTuneSet | Изменить |
| T\_TARIFF\_TIMETABLE | Периоды действия тарифа | proc | TariffTimeDel | Удалить |
| T\_TARIFF\_TIMETABLE | Периоды действия тарифа | proc | TariffTimeSet | Добавить |
| T\_TARIFF\_TIMETABLE | Периоды действия тарифа | proc | TariffTimeSet | Изменить |
| T\_TARIFF\_BASES\_SYNC | Синхронное изменение тарифов для подразделений | proc | TariffBasesSyncDel | Удалить |
| T\_TARIFF\_BASES\_SYNC | Синхронное изменение тарифов для подразделений | proc | TariffBasesSyncSet | Добавить |
| T\_TARIFF\_BASES\_SYNC | Синхронное изменение тарифов для подразделений | proc | TariffBasesSyncSet | Изменить |
| T\_TARIFF\_DISABLE\_MARK | Тариф.Исключенные марки авто | proc | TariffDisableMarkDel | Удалить |
| T\_TARIFF\_DISABLE\_MARK | Тариф. Исключенные марки авто | proc | TariffDisableMarkSet | Добавить |
| T\_TARIFF\_AUTO\_CLASS | Тариф. Классы автомобилей | proc | TariffAutoClassCopy | Копировать |
| T\_TARIFF\_AUTO\_CLASS | Тариф. Классы автомобилей | proc | TariffAutoClassDel | Удалить |
| T\_TARIFF\_AUTO\_CLASS | Тариф. Классы автомобилей | proc | TariffAutoClassSet | Добавить |
| T\_TARIFF\_AUTO\_CLASS | Тариф. Классы автомобилей | proc | TariffAutoClassSet | Изменить |
| T\_TARIFF\_ZONES | Тариф. Маршруты | proc | TariffZonesDel | Удалить |
| T\_TARIFF\_ZONES | Тариф. Маршруты | proc | TariffZonesDelMass | Удалить (массово) |
| T\_TARIFF\_ZONES | Тариф. Маршруты | proc | TariffZonesSet | Добавить |
| T\_TARIFF\_ZONES | Тариф. Маршруты | proc | TariffZonesSet | Изменить |
| T\_TARIFF\_ZONES | Тариф. Маршруты | proc | TariffZonesSetBack | Создать обратный маршрут (массово) |
| T\_TARIFF\_ZONES | Тариф. Маршруты | proc | TariffZonesSetMass | Изменение цены (массово) |
| T\_SATELLIT\_OUTCITY\_ROUND | Тариф. Округлять до города | proc | SatellitOutcityRoundDel | Удалить |
| T\_SATELLIT\_OUTCITY\_ROUND | Тариф. Округлять до города | proc | SatellitOutcityRoundSet | Добавить |
| T\_TARIFF | Тарифы | proc | GetRoundingIncrement | Округление цены заказа по настройке тарифа |
| T\_TARIFF | Тарифы | proc | tariffActiveToBasic | Отменить изменения |
| T\_TARIFF | Тарифы | proc | TariffBlock | Блокировать/Разблокировать |
| T\_TARIFF | Тарифы | proc | TariffCopy | Копировать |
| T\_TARIFF | Тарифы | proc | tariffMakeActiveVersion | Обновить активную версию |
| T\_TARIFF | Тарифы | proc | TariffParentSet | Изменить базовый тариф |
| T\_TARIFF | Тарифы | proc | TariffPriceFactorSet | Изменить коэффициент увеличения |
| T\_TARIFF | Тарифы | proc | tariffReasonSet | Указать причину обновления версии |
| T\_TARIFF | Тарифы | proc | TariffSet | Изменить |
| T\_TARIFF | Тарифы | proc | TariffSimpleSet | Добавить |
| T\_TARIFF | Тарифы | proc | TariffTemplateCountrySet | Изменить страну |
| T\_TARIFF | Тарифы | proc | TariffTimeSet | Расписание |
| T\_TARIFF | Тарифы | proc | TariffToArc | Добавить тариф в архив |
| T\_SMENA\_TARIFF | Тарифы смен | proc | SmenaTariffSet | Добавить |
| T\_SMENA\_TARIFF | Тарифы смен | proc | SmenaTariffSet | Изменить |
| T\_TARIFF\_TAG | Теги тарифов | proc | tariff\_tag\_set | Добавить |
| T\_TARIFF\_TYPE | Типы тарифов | proc | TariffTypeDel | Удалить |
| T\_TARIFF\_TYPE | Типы тарифов | proc | TariffTypeDopInfoSet | Изменить доп.инфо |
| T\_TARIFF\_TYPE | Типы тарифов | proc | TariffTypeSet | Добавить |
| T\_TARIFF\_TYPE | Типы тарифов | proc | TariffTypeSet | Изменить |
| T\_TARIFF\_TYPE\_DESCRIPTION | Типы тарифов. Описания | proc | TariffTypeDescriptionSet | Добавить |
| T\_TARIFF\_TYPE\_DESCRIPTION | Типы тарифов. Описания | proc | TariffTypeDescriptionSet | Изменить |
| T\_TARIFF\_DESCR\_TEMPLATE | Шаблоны описаний тарифов | proc | TariffDescriptionTemplateDel | Удалить |
| T\_TARIFF\_DESCR\_TEMPLATE | Шаблоны описаний тарифов | proc | TariffDescriptionTemplateSet | Добавить |
| T\_TARIFF\_DESCR\_TEMPLATE | Шаблоны описаний тарифов | proc | TariffDescriptionTemplateSet | Изменить |

**3.2.5 Работа с лицевыми счетами**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Описание класса** | **Тип** | **Процедура | Представление** | **Описание** |
| T\_ACCOUNT | Лицевые счета | view | V\_ACCOUNT\_DRV\_W | Водительские рабочие |
| T\_FISCALIZATIONS\_DOCS | Фискальные документы | view | V\_FISCALIZATIONS\_DOCS | Все фискальные документы |
| T\_FISCALIZATIONS\_DOCS | Фискальные документы | view | V\_FISCALIZATIONS\_DOCS\_CREATE | Документы к отправке |
| T\_FISCALIZATIONS\_DOCS | Фискальные документы | view | V\_FISCALIZATIONS\_DOCS\_SEND | Документы на проверке |
| T\_FISCALIZATIONS\_CASHBOX | Фискальные кассы | view | V\_FISCALIZATIONS\_CASHBOX | Все кассы |
| T\_FISCALIZATIONS\_CASHBOX | Фискальные кассы | view | V\_FISCALIZATIONS\_CASHBOX\_W | Только рабочие |
| T\_ACCOUNT\_DEBTOR | Задолженности клиентов | proc | AccountDebtorSet | Создать задолженность |
| T\_ORDERS\_ALL | Заказы (журнал) | proc | OrderDebtorSet | Создать задолженность на клиента |
| T\_ORDERS\_CHECK | Заказы (проверка) | proc | OrderDebtorSet | Закрепить задолженность |
| T\_FISCALIZATIONS\_CASHBOX | Фискальные кассы | proc | FiscalizationCashboxSet | Добавить |
| T\_FISCALIZATIONS\_CASHBOX | Фискальные кассы | proc | FiscalizationCashboxSet | Изменить |

**3.2.6 Работа с реестром записей телефонных переговоров**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Описание класса** | **Тип** | **Процедура | Представление** | **Описание** |
| T\_DRIVERS\_SIP | SIP учетки водителей | view | V\_DRIVERS\_SIP | Полный список |
| T\_DRIVERS\_SIP | SIP учетки водителей | view | V\_DRIVERS\_SIP\_LCR\_ONLY | LCR ONLY Полный список |
| T\_CDR\_CHECKS | Анализ записей разговоров | view | V\_CDR\_CHECKS | Полный список |
| T\_IVR\_RECORDS | Аудиофайлы IVR | view | GetObjectsText2Voice | GetObjectsText2Voice |
| T\_IVR\_RECORDS | Аудиофайлы IVR | view | V\_IVR\_FILES | Файлы озвучки |
| T\_IVR\_RECORDS | Аудиофайлы IVR | view | V\_IVR\_RECORDS | Все записи |
| T\_CDR | Запись разговоров | view | V\_CDR | Полный список |
| T\_CDR | Запись разговоров | view | V\_CDR\_4ALL | Полный список для всех |
| T\_CDR | Запись разговоров | view | V\_CDR\_4REG | Полный список |
| T\_CDR | Запись разговоров | view | V\_CDR\_4REP | Операторские звонки |
| T\_CDR | Запись разговоров | view | V\_CDR\_CALLEE2O | Звонки к оплате |
| T\_CDR | Запись разговоров | view | V\_CDR\_CHECKS | Анализ разговоров |
| T\_CDR | Запись разговоров | view | V\_CDR\_CHILD | Статистика очереди |
| T\_CDR | Запись разговоров | view | V\_CDR\_DONE | Принятые |
| T\_CDR | Запись разговоров | view | V\_CDR\_HIDDEN | Скрытые звонки |
| T\_CDR | Запись разговоров | view | V\_CDR\_INCOMING\_CALLS | Звонки к оплате |
| T\_CDR | Запись разговоров | view | V\_CDR\_LOCAL | Офис. внутренние |
| T\_CDR | Запись разговоров | view | V\_CDR\_LOGIST | Полный список (логисту) |
| T\_CDR | Запись разговоров | view | V\_CDR\_OFFICE | Офисные |
| T\_CDR | Запись разговоров | view | V\_CDR\_OFFICE\_MY | Мои офисные |
| T\_CDR | Запись разговоров | view | V\_CDR\_OFFICE\_PERSON | Для подбора кадров |
| T\_CDR | Запись разговоров | view | V\_CDR\_OUT | Офис. внешние |
| T\_CDR | Запись разговоров | view | V\_CDR\_OUT\_REF | Офис. внешние (пр.) |
| T\_CDR | Запись разговоров | view | V\_CDR\_OUTGOING\_CALLS | Звонки к оплате |
| T\_CDR | Запись разговоров | view | V\_CDR\_REF | Звонки по заказу |
| T\_CDR | Запись разговоров | view | V\_CDR\_TRANSFER | Трансферы между очередями |
| T\_SMS\_OUT | Исходящие SMS | view | VA\_SMS\_OUT | Обработанные |
| T\_SMS\_OUT | Исходящие SMS | view | VA\_SMS\_OUT\_HIDDEN | Скрытые обработанные |
| T\_CALLOUT\_SETTINGS | Настройки автоотзвона | view | V\_CALLOUT\_SETTINGS | Все настройки |
| T\_SMS\_MAILING | Рассылки SMS | view | V\_SMS\_MAILING | Полный список |
| T\_PHONES\_SMSLOG | Рекламные SMS | view | V\_PHONES\_SMSLOG | Полный список |
| T\_ACCOUNTCODE | Типы звонков | view | V\_ACCOUNTCODE | Полный список |
| T\_SMS\_TYPE | Типы СМС | view | V\_SMS\_TYPE | Полный список |
| T\_CDR\_CHECKS | Анализ записей разговоров | proc | CdrChecksSet | Добавить |
| T\_IVR\_RECORDS | Аудиофайлы IVR | proc | ivr\_get\_class\_dir | Путь к файлам озвучки |
| T\_IVR\_RECORDS | Аудиофайлы IVR | proc | ivr\_get\_file\_name | Имя файла озвучки |
| T\_IVR\_RECORDS | Аудиофайлы IVR | proc | IvrRecordDel | Удалить |
| T\_IVR\_RECORDS | Аудиофайлы IVR | proc | IvrRecordSet | Добавить |
| T\_IVR\_RECORDS | Аудиофайлы IVR | proc | IvrRecordSet | Изменить |
| T\_CDR | Запись разговоров | proc | CdrSetCommentsManager | Изменить комментарий |
| T\_SMS\_OUT | Исходящие SMS | proc | InsertArcSmsOut | InsertArcSmsOut |
| T\_CALLOUT\_SETTINGS | Настройки автоотзвона | proc | CalloutSettingDel | Удалить |
| T\_CALLOUT\_SETTINGS | Настройки автоотзвона | proc | CalloutSettingSet | Добавить |
| T\_CALLOUT\_SETTINGS | Настройки автоотзвона | proc | CalloutSettingSet | Изменить |
| T\_SMS\_MAILING | Рассылки SMS | proc | SmsMailingDel | Удалить |
| T\_SMS\_MAILING | Рассылки SMS | proc | SmsMailingSet | Добавить |
| T\_SMS\_MAILING | Рассылки SMS | proc | SmsMailingSet | Изменить |
| T\_ACCOUNTCODE | Типы звонков | proc | AccountCodeDel | Удалить |
| T\_ACCOUNTCODE | Типы звонков | proc | AccountCodeSet | Добавить |
| T\_ACCOUNTCODE | Типы звонков | proc | AccountCodeSet | Изменить |
| T\_SMS\_TYPE | Типы СМС | proc | SMSTypeDel | Удалить |
| T\_SMS\_TYPE | Типы СМС | proc | SMSTypeSet | Добавить |
| T\_SMS\_TYPE | Типы СМС | proc | SMSTypeSet | Изменить |

**4** **АВАРИЙНЫЕ СИТУАЦИИ**

В случае возникновения ошибок при работе с Maindb, не описанных в данном документе, необходимо обратиться к службе поддержки, или по адресу электронной почты [support@taxsee.ru](mailto:support@taxsee.ru).

При ошибках в работе аппаратных средств восстановление функции системы возлагается на ОС.

При ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

При неверных действиях пользователей, неверных форматах или недопустимых значениях входных данных, система выдает пользователю соответствующие сообщения, после чего возвращается в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

**Приложение 1. API интерфейс к базе Maindb**

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.AbortQuery** | ( | string | contextId, |
|  |  | string | queryId, |
|  |  | string | userContextId |
|  | ) |  |  |

Остановка выполнения запроса

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста. |
| **queryId** | Идентификатор запроса или guid потока, который нужно завершить. |
| **userContextId** | Идентификатор контекста пользователя, чей запрос нужно завершить. |

Для завершения запроса с текущего контекста userContextId передается равным текущему идентификатору контекста либо NULL.

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.AddressSearch** | ( | string | contextId, |
|  |  | int | idPlace, |
|  |  | int | idBase, |
|  |  | long? | currentAddress, |
|  |  | long? | currentPlace, |
|  |  | string | phoneNumber, |
|  |  | string | filter, |
|  |  | bool | hidePlaces |
|  | ) |  |  |

Поиск адреса

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **idPlace** | Идентификатор филиала |
| **idBase** | Идентификатор филиала |
| **currentPlace** |  |
| **phoneNumber** | Номер телефона |
| **filter** | Строка поиска |
| **hidePlaces** | Скрывать филиалы |
| **currentAddress** |  |

**Возвращает**

Возвращает список адресов

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.AddressSearchEx** | ( | string | contextId, |
|  |  | int | idPlace, |
|  |  | int | idBase, |
|  |  | long? | currentAddress, |
|  |  | long? | currentPlace, |
|  |  | string | phoneNumber, |
|  |  | string | filter, |
|  |  | bool | hidePlaces, |
|  |  | string | varParams |
|  | ) |  |  |

Расширенный метод адресного поиска

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **idPlace** | Идентификатор филиала |
| **idBase** | Идентификатор подразделения |
| **currentPlace** | идентификатор текущего филиала |
| **phoneNumber** | Номер телефона |
| **filter** | Строка поиска |
| **hidePlaces** | Скрывать филиалы |
| **currentAddress** | Идентификатор текущего адреса |
| **varParams** | Дополнительные параметры |

Офисание дополнительных параметров

{

"showHidden": value, Показывать скрытые адреса

}

Пример возвращаемого значения

[

{

"C\_TYPE":1,

"ID\_PLACE":1,

"C\_PLACE":"Курган",

"ID\_ADDR":null,

"C\_ADDR\_NAME":null,

"ID\_STREET":100599,

"C\_STREET\_NAME":"Радионова",

"C\_NUM\_HOUSE":null,

"ID\_ZONE":null,

"C\_ZONE\_NAME":null,

"C\_SYNONYM":null,

"C\_NOTE":null,

"ID\_BASE":null,

"C\_LATITUDE":null,

"C\_LONGITUDE":null,

"ID\_CITY":null,

"C\_CITY":null,

"ID\_TYPE\_ADDR":null,

"C\_NAME":null,

"C\_DESCRIPTION":null,

"ID\_COUNTRY":null,

"C\_POPULATION":null,

"ID\_REGION":null,

"C\_IS\_ADDR":null,

"C\_SELECTED":true,

"C\_MEET\_POINT":null,

"C\_MEET\_POINT\_EXT":null,

"C\_ADD\_DOP\_INFO":null,

"C\_STREET\_NAME\_SHORT":"Радионова",

"ID":0

},

{

"C\_TYPE":2,

"ID\_PLACE":1,

"C\_PLACE":"Курган",

"ID\_ADDR":4354239,

"C\_ADDR\_NAME":"Остановка Максим на Радионова",

"ID\_STREET":null,

"C\_STREET\_NAME":null,

"C\_NUM\_HOUSE":null,

"ID\_ZONE":null,

"C\_ZONE\_NAME":null,

"C\_SYNONYM":null,

"C\_NOTE":null,

"ID\_BASE":null,

"C\_LATITUDE":55.438853,

"C\_LONGITUDE":65.312828,

"ID\_CITY":null,

"C\_CITY":null,

"ID\_TYPE\_ADDR":6,

"C\_NAME":null,

"C\_DESCRIPTION":null,

"ID\_COUNTRY":null,

"C\_POPULATION":null,

"ID\_REGION":null,

"C\_IS\_ADDR":null,

"C\_SELECTED":false,

"C\_MEET\_POINT":null,

"C\_MEET\_POINT\_EXT":null,

"C\_ADD\_DOP\_INFO":null,

"C\_STREET\_NAME\_SHORT":null,

"ID":0

}

]

**Возвращает**

Возвращает список адресов

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.ChangePassword** | ( | string | user, |
|  |  | string | oldPassword, |
|  |  | string | newPassword |
|  | ) |  |  |

Смена пароля пользователя

**Аргументы**

|  |  |
| --- | --- |
| **user** | Имя пользователя |
| **oldPassword** | Старый пароль |
| **newPassword** | Новый пароль |

**Исключения**

|  |  |
| --- | --- |
| ChangePassword\_WrongPasswordException |  |
| TokenExpiredException |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.CheckFilter** | ( | string | contextId, |
|  |  | string | filter |
|  | ) |  |  |

Метод служит для теста разбора параметра - фильтра, при построении SQL - запроса в GetView. Тест проходит на Primary end point.

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | id контекста |
| **filter** | объект-фильтр |

**Возвращает**

сгенерированный SQL-запрос

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void MBroker.IBrokerService.ClearCache** | ( | string | contextId | ) |  |

Очистка кэша пользователя (права доступа, соединения с зеркалами и т.п.)

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void MBroker.IBrokerService.ClearCacheForAll** | ( | string | contextId | ) |  |

Очистка кэша для всех пользователей

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |

**Исключения**

|  |  |
| --- | --- |
| TokenExpiredException |  |
| LoginFailedException |  |
| [**AccessDeniedException**](https://docs.technology45.ru/mbroker/html/class_m_broker_1_1_access_denied_exception.html) |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.CopyFile** | ( | string | contextId, |
|  |  | string | fileId, |
|  |  | string | fileName, |
|  |  | string | type, |
|  |  | int | idRecord, |
|  |  | string | className, |
|  |  | string | mirror, |
|  |  | string | varParams |
|  | ) |  |  |

Делает копию файла в хранилище

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **fileId** | Строка с уникальным идентификатором файл |
| **fileName** | Имя файла |
| **type** | Ссылка на тип файла из T\_FILE\_TYPES |
| **idRecord** | ID записи внутри выбранного класса |
| **className** | Имя класса (например: T\_DRIVERS) |
| **mirror** | Зеркало, на котором хранится файл |
| **varParams** | JSON с дополнительными параметрами |

**Возвращает**

Строка с уникальным идентификатором нового файла

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.DeleteFile** | ( | string | contextId, |
|  |  | string | guid |
|  | ) |  |  |

Метод служит для удаления файла из хранилища

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Id контекста |
| **guid** | Идентификатор файла |

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.DeleteFileFBM** | ( | string | contextId, |
|  |  | string | fileId, |
|  |  | string | mirror |
|  | ) |  |  |

Удаление файла

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **fileId** | Строка с уникальным идентификатором файла |
| **mirror** | Зеркало, на котором хранится файл |

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.DeleteFileInfo** | ( | string | contextId, |
|  |  | string | guid |
|  | ) |  |  |

Метод служит для удаления записи о файле из хранилища

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Id контекста |
| **guid** | Идентификатор файла |

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.DeleteFileInfoV2** | ( | string | contextId, |
|  |  | string | mirror, |
|  |  | string | guid, |
|  |  | string | varParams |
|  | ) |  |  |

Метод служит для удаления записи о файле из хранилища с учетом зеркала

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Id контекста |
| **mirror** | Зеркало |
| **guid** | Идентификатор файла |
| **varParams** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **int MBroker.IBrokerService.ExecProc** | ( | string | contextId, |
|  |  | string | classCode, |
|  |  | string | procName, |
|  |  | string | parameters |
|  | ) |  |  |

Выполнение процедуры на сервере, с учетом MIRROR.

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **classCode** | Код класса |
| **procName** | Имя процедуры |
| **parameters** | JSON со списком параметров |

**Возвращает**

?

**Исключения**

|  |  |
| --- | --- |
| [**AccessDeniedException**](https://docs.technology45.ru/mbroker/html/class_m_broker_1_1_access_denied_exception.html) |  |
| BrokerSqlException |  |
| ServiceUserException |  |
| TokenExpiredException |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.ExecProcGlobal** | ( | string | contextId, |
|  |  | string | classCode, |
|  |  | string | procCode, |
|  |  | string | parameters, |
|  |  | string | varParams |
|  | ) |  |  |

Выполнение процедуры на всех известных брокеру шардах, с учетом MIRROR. Для varParams: опционально передаётся объект с полями: [bool] isAuto – показывает, является ли запрос автоматизированным, либо он инициирован пользователем интерактивно

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **classCode** | Класс процедуры |
| **procCode** | Код процедуры |
| **parameters** | JSON со списком параметров |
| **varParams** | JSON-объект с дополнительными параметрами для самого брокера |

**Возвращает**

?

**Исключения**

|  |  |
| --- | --- |
| [**AccessDeniedException**](https://docs.technology45.ru/mbroker/html/class_m_broker_1_1_access_denied_exception.html) |  |
| BrokerSqlException |  |
| ServiceUserException |  |
| TokenExpiredException |  |

Формат JSON-объекта с дополнительными параметрами для самого брокера

{

"isAuto": value // Определяет автоматический запрос или сгенерированный вручную. Если true - автоматический, false - вручную

"fetchResult": true/false// Если включено, то значения из процедур возвращаются из result set'ов, если выключено - из output параметров

"priority": 0/1/2/3 // Приоритет запроса. Если не указан, для всех запросов приоритет будет равен 1

"ttl": 1 // Время жизни запроса. Используется совместно с приоритетом. Если параметр не указан, время жизни для всех запросов будет равно 1 минуте.

"tz":"UTC"// Таймзона. Все параметры Даты/Времени будут приведены к данной таймзоне

"idShard": value // Идентификатор шарда, на котором будет выполняться запрос, если брокер сам не сможет этого определить

}

Пояснение к параметру "isAuto".

Если в течение интервала времени t от клиентского приложения поступают только автоматические запросы, закрываем сессию, что влечет за собой фиксацию времени закрытия в журнале сессий пользователя. Таким образом, после суммирования всех фиксаций и умножения на интервал t получим время, которое пользователь не работал. ///

Пример возвращаемого значения.

[

{

"IdShard":1,// Идентификатор шарда

"Status":"Success",// Статус выполнения

"Result":"[{\"GETDATE\":\"2017-10-23T07:05:56.404375\"}]",// Результат выполнения

"Ex":null// Эксепшен

},

{

"IdShard":2,

"Status":"Success",

"Result":"[{\"GETDATE\":\"2017-10-23T07:05:56.408813\"}]",

"Ex":null

}

]

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.ExecProcO** | ( | string | contextId, |
|  |  | string | classCode, |
|  |  | string | procName, |
|  |  | string | parameters |
|  | ) |  |  |

Выполнение процедуры на сервере, с учетом MIRROR.

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **classCode** | Код класса |
| **procName** | Имя процедуры |
| **idBase** | ID филиала |
| **parameters** | JSON со списком параметров |

**Возвращает**

JSON с результатами выполнения. Возвращаются параметры с типом Output или Input/Output

**Исключения**

|  |  |
| --- | --- |
| [**AccessDeniedException**](https://docs.technology45.ru/mbroker/html/class_m_broker_1_1_access_denied_exception.html) |  |
| BrokerSqlException |  |
| TokenExpiredException |  |
| ServiceUserException |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.ExecProcV2** | ( | string | contextId, |
|  |  | string | classCode, |
|  |  | string | procCode, |
|  |  | string | parameters, |
|  |  | string | varParams |
|  | ) |  |  |

Выполнение процедуры на сервере, с учетом MIRROR. Для varParams: опционально передаётся объект с полями: [bool] isAuto – показывает, является ли запрос автоматизированным, либо он инициирован пользователем интерактивно

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **classCode** | Класс процедуры |
| **procCode** | Код процедуры |
| **parameters** | JSON со списком параметров |
| **varParams** | JSON-объект с дополнительными параметрами для самого брокера |

**Возвращает**

?

**Исключения**

|  |  |
| --- | --- |
| [**AccessDeniedException**](https://docs.technology45.ru/mbroker/html/class_m_broker_1_1_access_denied_exception.html) |  |
| BrokerSqlException |  |
| ServiceUserException |  |
| TokenExpiredException |  |

Формат JSON-объекта с дополнительными параметрами для самого брокера

{

"isAuto": value // Определяет автоматический запрос или сгенерированный вручную. Если true - автоматический, false - вручную

"isAsync": value // Параметр определяет будет ли запрос выполняться асинхронно. Если true - запрос выполняется асинхронно, false - синхронно

"fetchResult": true/false// Если включено, то значения из процедур возвращаются из result set'ов, если выключено - из output параметров

"priority": 0/1/2/3 // Приоритет запроса. Если не указан, для всех запросов приоритет будет равен 1

"ttl": 1 // Время жизни запроса. Используется совместно с приоритетом. Если параметр не указан, время жизни для всех запросов будет равно 1 минуте.

"tz":"UTC"// Таймзона. Все параметры Даты/Времени будут приведены к данной таймзоне

"idShard": value // Идентификатор шарда, на котором будет выполняться запрос, если брокер сам не сможет этого определить

"language": "value"// Локализация результата запроса. Если данный параметр не задан, берется локаль из контекста

}

Пояснение к параметру "isAuto".

Если в течение интервала времени t от клиентского приложения поступают только автоматические запросы, закрываем сессию, что влечет за собой фиксацию времени закрытия в журнале сессий пользователя. Таким образом после суммирования всех фиксаций и умножения на интервал t получим время, которое пользователь не работал.

Пояснение к параметру "isAsync".

При использовании параметра isAsync учитывается время выполнения асинхронного запроса. В случае, если запрос выполняется быстро, возвращается JSON с результатом. Если запрос долгий, то возвращается строка с числом, по которому осуществляется получение результата запроса при помощи функции **[GetResultQuery](https://docs.technology45.ru/mbroker/html/interface_m_broker_1_1_i_broker_service.html" \l "aac730b2907c8231f3225c0ac004d7aa2)**/n.

В случае, если запрос асинхронный, статус его выполнения необходимо периодически запрашивать функцией **[GetQueryStatus](https://docs.technology45.ru/mbroker/html/interface_m_broker_1_1_i_broker_service.html" \l "a7326fc35b7438cf750dff2b50a495620)**. Если этого не делать, запрос будет отменет и удален из коллекции запросов контекста.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetBaseList** | ( | string | contextId | ) |  |

Получение полного списка незаблокированных филиалов

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |

**Возвращает**

JSON со списком филиалов

**Исключения**

|  |  |
| --- | --- |
| BrokerSqlException |  |
| TokenExpiredException |  |
| ServiceUserException |  |

Пример возвращаемого JSON:

[

{"ID": 1, "C\_NAME":"Курган", "C\_INFO":"...", ... },

{"ID": 2, "C\_NAME":"Шадринск", "C\_INFO":"...", ... },

...

]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetBrokerConfig** | ( | string | contextId | ) |  |

Метод служит для получения конфигурации брокера

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Контекст пользователя |

**Возвращает**

JSON с представлением

///

/// Пример возвращаемого JSON:

{

"brokerSection":

[

{

"brokerSection\_Id":0,"logfilter":"V\_CHAT\_W,V\_EVENTS","brokerId":"work1"

}

],

"cache":

[

{

"parameterTTL":"5","columnTTL":"5","smoTTL":"3","mirrorsTTL":"4","brokerSection\_Id":0

}

]

}

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetCallQueueLoad** | ( | string | contextId, |
|  |  | string[] | codes |
|  | ) |  |  |

Получение нагрузки по очередям ожидающих звонков

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **codes** | коды очередей, если их нет, вернутся все коды |

**Возвращает**

Список кода со значением цвета(нагрузки на очередь)

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetContext** | ( | string | user, |
|  |  | string | password, |
|  |  | string | clientUuid |
|  | ) |  |  |

Получение контекста пользователя

**Аргументы**

|  |  |
| --- | --- |
| **user** | Имя пользователя |
| **password** | Пароль |
| **clientUuid** | UUID клиента |

**Исключения**

|  |  |
| --- | --- |
| InvalidUserException |  |
| IPValidationFailedException |  |
| LoginFailedException |  |
| PasswordExpiredException |  |
| ServiceUserException |  |
| SqlOperationException |  |
| UnknownUserException |  |
| UserLockedOutException |  |

**Возвращает**

Идентификатор контекста пользователя

Вызов метода сбрасывает кэш пользователя и соединения с зеркалами.

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetContextBySt** | ( | string | service, |
|  |  | string | ticket |
|  | ) |  |  |

Получение контекста для приложений в которых уже пройдена авторизация через CAS

**Аргументы**

|  |  |
| --- | --- |
| **service** | Имя сервиса/приложения |
| **ticket** | Токен |

**Возвращает**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetContextObjectLock** | ( | string | contextId | ) |  |

Получение объектов, заблокированных контекстом

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |

**Возвращает**

Список зеркал с заблокированными объектами данным контекстом

Пример возвращаемого значения

{

"primary":

{

"ObjectLockList":

[

{

"ObjType":"T\_ORDERS",

"ObjID":342447379

}

]

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetContextV2** | ( | string | user, |
|  |  | string | password, |
|  |  | string | clientUuid, |
|  |  | string | hostName, |
|  |  | string | clientAppName, |
|  |  | string | varParams |
|  | ) |  |  |

Получение контекста пользователя

**Аргументы**

|  |  |
| --- | --- |
| **user** | Имя пользователя |
| **password** | Пароль |
| **clientUuid** | UUID клиента |
| **hostName** | Название рабочей станции клиента |
| **clientAppName** | Название клиентского приложения |
| **varParams** | Дополнительные параметры в формате JSON |

**Исключения**

|  |  |
| --- | --- |
| InvalidUserException |  |
| IPValidationFailedException |  |
| LoginFailedException |  |
| PasswordExpiredException |  |
| ServiceUserException |  |
| SqlOperationException |  |
| UnknownUserException |  |
| UserLockedOutException |  |

**Возвращает**

Идентификатор контекста пользователя.

Вызов метода сбрасывает кэш пользователя и соединения с зеркалами.

Формат varParams

{

"language": "value"// Определяет локаль подключившегося пользователя, если параметр не не задан, то автоматически русская локаль

}

Локаль задается текстовой строкой:

"ru"- русская локаль\n

"de"- немецкая локаль\n

"uk"- украинская локаль\n

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetCustomPoolStatistic** | ( | string | contextId | ) |  |

Получение статистики кастомного пула

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |

**Возвращает**

Словарь пулов со статистикой

Пример возвращаемого значения

{

"Имя пула":

{

"PoolSize":100,

"AverageTimeExecuteQuery":0.0,

"CountQuery":1,

"CountFreeConnection":100,

"QueueStatistic":

{

"AverageLifeTimeQuery":0.0,

"CountQueryInQueue":0

},

"DateUpdate":"2015-02-17T11:05:51.9398261"

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetEvents** | ( | string | contextId, |
|  |  | int? | idBase, |
|  |  | DateTime? | dateTime |
|  | ) |  |  |

Вернуть текущеее значение фильтра выделенного лога.

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetEventsByPlace** | ( | string | contextId, |
|  |  | int? | idPlace, |
|  |  | DateTime? | dateTime |
|  | ) |  |  |

Вернуть текущеее значение фильтра выделенного лога по idPlace.

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetFile** | ( | string | contextId, |
|  |  | string | guid |
|  | ) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetFileFBM** | ( | string | contextId, |
|  |  | string | fileId, |
|  |  | string | mirror |
|  | ) |  |  |

Метод получения файла с сервера

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **fileId** | Строка с уникальным идентификатором файла |
| **mirror** | Зеркало, на котором хранится файл |

**Возвращает**

Бинарные данные

* Пример строки с уникальным идентификатором файла "5360d23bffd0070994b03f47" Пример возвращаемого значения Аналог входного параметра "date" в методе "UploadFile"

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetFileInfo** | ( | string | contextId, |
|  |  | string | guid |
|  | ) |  |  |

Метод служит для получения изначального имени файла.

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | id контекста |
| **guid** | Идентификатор файла |

**Возвращает**

JSON-объект, поля: FileName - имя файла, FileType - MIME-тип

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetFileInfoFBM** | ( | string | contextId, |
|  |  | string | fileId |
|  | ) |  |  |

Метод служит для получения информации о файле

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **fileId** | Идентификатор файла |

**Возвращает**

JSON-объект.  
поля: FileName - имя файла,  
FileType - MIME-тип,  
Mirror - зеркало, на котором лежит файл

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetFileInfoV2** | ( | string | contextId, |
|  |  | string | mirror, |
|  |  | string | fileId, |
|  |  | string | varParams |
|  | ) |  |  |

Метод служит для получения информации о файле с учетом зеркала хранения

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **mirror** | Зеркало |
| **fileId** | Идентификатор файла |
| **varParams** | Дополнительные параметры |

**Возвращает**

JSON-объект.  
поля: FileName - имя файла,  
FileType - MIME-тип,  
Mirror - зеркало, на котором лежит файл

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetFilter** | ( | string | contextId | ) |  |

Вернуть текущеее значение фильтра выделенного лога.

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetGroupAccesses** | ( | string | contextId, |
|  |  | int | groupId |
|  | ) |  |  |

Возвращает объекты доступа группы

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **groupId** | Идентификатор группы доступа |

**Возвращает**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetLoginData** | ( | string | contextId | ) |  |

Получение данных созданной сессии.

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |

**Возвращает**

JSON с результатом выполнения UserLogin

**Исключения**

|  |  |
| --- | --- |
| [**AccessDeniedException**](https://docs.technology45.ru/mbroker/html/class_m_broker_1_1_access_denied_exception.html) |  |
| BrokerSqlException |  |
| ServiceUserException |  |
| TokenExpiredException |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetProcParam** | ( | string | contextId, |
|  |  | string | classCode, |
|  |  | string | procName |
|  | ) |  |  |

Получение параметров хранимой процедуры

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **classCode** | Класс процедуры |
| **procName** | Имя процедуры |

**Возвращает**

|  |  |  |  |
| --- | --- | --- | --- |
| **int MBroker.IBrokerService.GetQueryStatus** | ( | string | contextId, |
|  |  | int | id |
|  | ) |  |  |

Возвращает статус асинхронного запроса

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **id** | Идентификатор запроса |

**Возвращает**

Статус запроса

* Описание возвращающегося значения  
  1 - Запрос выполняется  
  0 - Запрос выполнен  
  -1 - Запроса не существует

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetResultQuery** | ( | string | contextId, |
|  |  | int | id |
|  | ) |  |  |

Получение результата запроса по идентификатору

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Контекст пользователя |
| **id** | Идентификатор |

**Возвращает**

Результат в формате JSON

* Описание  
  Если запрос или представление с идентификаторм id выполнились, то возвращается результат в формате JSON иначе возвращается null, если запроса не существует, возвращается null

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetServiceInfo** | ( | string | contextId | ) |  |

Возвращает статус активности сервиса и отдельных пользователей

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |

Офисание возвращаемого значения

{

{

"ServiceActived":false,

"UserActivityForbidden":{}

}// Сервис не активен

{

"ServiceActived":false,

"UserActivityForbidden":

{

"ivanov":false

}

}// Сервис не активен, но пользователю ivanov, разрешены операции с сервисом

}

{

"ServiceActived":true,

"UserActivityForbidden":

{

"ivanov":true

}

}// Сервис не активен, но пользователю ivanov, запрещены операции с сервисом

}

**Возвращает**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetServiceState** | ( | string | contextId | ) |  |

Получение статуса сервиса

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |

Офисание возвращаемого значения

{"service":"active"} - Сервис активный

{"service":"suspended"} - Сервис не активный

**Возвращает**

Состояние сервиса

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetStatistic** | ( | string | contextId | ) |  |

Получение статистики вызова процедур и использования вьюх

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Контекст пользователя |

**Возвращает**

JSON с результатом выполнения

Пример возвращаемого значения

{

{

"Procedure":

{

"getListContexts":1

},

"View":

{

"psGetClassesFX":2

}

}

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetThreadStatistic** | ( | string | contextId | ) |  |

Получение статистики потоков

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |

**Возвращает**

JSON со статистикой

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetThumbnail** | ( | string | contextId, |
|  |  | string | guid |
|  | ) |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetTzAccesses** | ( | string | contextId | ) |  |

Возвращает список объектов доступа

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |

**Возвращает**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetTzGroups** | ( | string | contextId | ) |  |

Возвращает список групп доступа

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |

**Возвращает**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetUnreadMessages** | ( | string | contextId | ) |  |

Получение непрочитанных сообщений

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Контекст пользователя |

**Возвращает**

Непрочитанные сообщения в формате JSON

Пример возвращаемого значения

{

"ID":227944,

"C\_DATE":"2014-05-23T05:00:19.623",

"C\_TEXT":"Тест",

"C\_USER":"novgorodov\_dv"

}

"null" - если сообщений нет

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetVersion** | ( |  | ) |  |

Получение версии брокера

**Возвращает**

Версия брокера

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetVersionEx** | ( |  | ) |  |

Расширенный метод получения версии брокера

///

Офисание возвращаемого значения

{

"Version":

{

"Major":2,

"Minor":0,

"Build":6282,

"Revision":21922,

"MajorRevision":0,

"MinorRevision":21922

},

"BuildDate":"03/14/2017 07:10:44"

}

**Возвращает**

Версия брокера

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetView** | ( | string | contextId, |
|  |  | string | queryDescription, |
|  |  | bool | extendedInfo |
|  | ) |  |  |

Получение представления

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **queryDescription** | JSON с описанием запрашиваемого представления |
| **extendedInfo** | Если установлено в true, возвращает расширенную информацию о запросе. |

**Возвращает**

JSON с представлением

**Исключения**

|  |  |
| --- | --- |
| BrokerSqlException |  |
| TokenExpiredException |  |
| ServiceUserException |  |

Формат JSON-запроса к представлению:

{

"base": BASEID, // ID филиала. Если 0 или отсутствует - все доступные филиалы

"class":"CLASS\_NAME", // Имя класса. Обязательный параметр

"view":"VIEW\_NAME", // Имя представления или функции. Обязательный параметр

"params": [ "PARAM1", "PARAM2"], // Список параметров для хранимой функции. Обязательный для вызовов функций.

"limit": ROWCOUNT, // Максимальное количество строк. Если не указано или равно 0, то возвращается полный список

"fields": [ "FIELD1", "FIELD2"], // Запрашиваемые поля. Если не указано, возвращаются все поля представления

"filter": ..., // Фильтр (тип фильтра задается в нижнем регистре)

"groupby": [ "FIELD1", "FIELD2"], // Группирование

"order": { "FIELD1": "DESC"} // Порядок сортировки

}

Описание поля групировки  
Операция группировки, поддерживает возможность группировать результаты выборки по датам с учетом выбранного клиентом метода сокращения даты Пример поля groupby:

{

"groupby":["C\_USER\_CREATE", {"C\_DATE\_CREATE":{"trunc":"year"}}]}

"groupby":["C\_USER\_CREATE", {"C\_DATE\_CREATE":{"part":"year"}}]}

}

Методы работы с датой:

{

"trunc"- Округление даты до недели, месяца, квартала, года

"part"- Получение полей времени

}

Возможные значения trunc/part:

{

"minute"

"hour"

"day"

"week"

"month"

"quarter"

"year"

}

Пример запроса с использованием groupby и сокращением даты

{

{

"view":"V\_ORDERS",

"class":"T\_ORDERS",

"fields":

[

"C\_USER\_CREATE",

{"count": "ID","name": "C\_SUM"}

],

"groupby":

[

"C\_USER\_CREATE",

{

"C\_DATE\_CREATE":

{

"part":"year"

}

}

]

}

}

Возвращает количество созданных заявок для каждого пользователя в течении года Транслируется в следующий SQL код:

SELECT DATEPART(year,C\_DATE\_CREATE) AS C\_DATE\_CREATE\_PART\_YEAR,C\_USER\_CREATE,COUNT(ID) AS C\_SUM FROM V\_ORDERS AS vw GROUP BY C\_USER\_CREATE,DATEPART(year,C\_DATE\_CREATE)

Описание фильтра  
Если фильтр представлен в виде массива, то элементы объединяются условием OR. Если фильтр представлен одним объектом, то свойства объединяются условием AND.  
Пример AND-фильтра:

{

"C\_FIELD1":"text1",

"C\_FIELD2":"text2"

}

// Данный JSON будет транслирован в SQL-условие WHERE C\_FIELD1='text1' AND C\_FIELD2='text2'

Пример OR-фильтра:

[

{"C\_FIELD1": "text1"},

{"C\_FIELD2": "text2"}

]

// Данный JSON будет транслирован в SQL-условие WHERE C\_FIELD1='text1' OR C\_FIELD2='text2'

Агрегатные функции  
В списке запрашиваемых полей поддерживаются агрегатные функции, такие как sum, avg, min, max, count. Общий формат:

{"[операция]":"FIELD", "name":"[имя\_результа]" }

Служебные модификаторы:

# - В начальной позиции имени поля заменяется на DISTINCT

Пример использования агрегатной функции в списке запрашиваемых полей:

"fields": [ "ID\_BASE", { "sum":"C\_SUM", "name":"C\_SUM\_TOTAL" } ]

// Данный запрос в SQL выглядит как SELECT ID\_BASE, SUM(C\_SUM) AS C\_SUM\_TOTAL [...]

Операции над полями  
Поддерживаются операции coalesce и cast в фильтре и списке запрашиваемых полей.  
Формат для использования в списке запрашиваемых полей:

{"coalesce": [ "C\_FIELD","DEFAULT\_VALUE"] }

// Результат - COALESCE(C\_FIELD, "DEFAULT\_VALUE")

{"cast": [ "C\_FIELD","SOME\_SQL\_TYPE" ] }

// Результат - CAST(C\_FIELD AS SOME\_SQL\_TYPE)

Формат операций в фильтре:

{"ID\_BASE": { "coalesce": 0, "cast": "SQL\_TYPE"...[условия]... }}

Например:

{"ID\_BASE": { "coalesce": 0, "gt": 0 } }

// Дает в SQL условие COALESCE(ID\_BASE, 0)>0

{"ID\_BASE": { "cast":"VARCHAR(10)", "like":"1%" } }

// Дает условие CAST(ID\_BASE AS VARCHAR(10)) LIKE '1%'

Формат условных операций  
Общий формат условия:

{"C\_FIELD": { "условие":"значение" } }

Возможные условия:

|  |  |  |
| --- | --- | --- |
| **Условие** | **Получаемый SQL** |  |
| {"C\_FIELD": { "eq": 10 } } | C\_FIELD=10 | |
| {"C\_FIELD": { "neq": 10 } } | C\_FIELD<>10 | |
| {"C\_FIELD": { "gt": 10 } } | C\_FIELD>10 | |
| {"C\_FIELD": { "lt": 10 } } | C\_FIELD<10 | |
| {"C\_FIELD": { "gte": 10 } } | C\_FIELD>=10 | |
| {"C\_FIELD": { "lte": 10 } } | C\_FIELD<=10 | |
| {"C\_FIELD": { "in": [10, 20, 30] } } | C\_FIELD IN (10, 20, 30) | |
| {"C\_FIELD": { "nin": [10, 20, 30] } } | C\_FIELD NOT IN (10, 20, 30) | |
| {"C\_FIELD": { "like":"%text%" } } | C\_FIELD LIKE '%text%' | |
| {"C\_FIELD": { "nlike":"%text%" } } | C\_FIELD NOT LIKE '%text%' | |

Условие может дополняться индексом, если в рамках одной конструкции несколько однотипных. Например:

{"ID\_BASE": { "neq1": null, "neq2": 0 } }

// JSON будет транслирован в SQL как "ID\_BASE IS NOT NULL AND ID\_BASE<>0"

Вложенные условия задаются с помощью ключевого слова "subcondition". Например:

{

"C\_INFO": { "neq": null }

"subcondition": [ { "ID\_BASE": null }, {"ID\_BASE": 0 } ]

}

// Фильтр задает SQL-условие C\_INFO IS NOT NULL AND (ID\_BASE IS NULL OR ID\_BASE=0)

Допускается нумерация вложенных условий, если их несколько в рамках одной конструкции ("subcondition1", "subcondition2", "subcondition3", etc.)

**Примеры фильтров:**

|  |  |  |
| --- | --- | --- |
| **JSON** | **SQL** |  |
| [ { "ID\_BASE": null }, {"ID\_BASE": 0 }] | ID\_BASE IS NULL OR ID\_BASE=0 | |
| {"ID": { "gt": 100, "lt": 200 },"C\_FIELD": { "neq": 900 } } | ID>100 AND ID<200 AND C\_FIELD<>900 | |
| [  {  "ID": { "in": [ 100, 101, 102 ] },  "C\_FIELD": { "like": "%something"}  },  {  "C\_FAIL": 0  }  ] | (ID IN (100, 101, 102) AND C\_FIELD LIKE '%something') OR (C\_FAIL=0) | |

**Примеры запросов:**

// Запрос списка классов

{"class": "T\_SYSTEM","view": "psGetClassesFX","params": [0,"username"] }

// Обновление списка событий

{

"class":"T\_SYSTEM",

"view":"V\_EVENTS",

"filter":

{

"C\_NAME": { "neq1": null, "neq2":"UNKNOWN" },

"C\_DATE\_UPDATE": { "gt":"\/Date(1350596695633+0600)\/"}

}

}

// Запрос представления

{

"class":"T\_AUTO",

"view":"V\_AUTO\_W",

"limit": 200,

"base": 0,

"order":

{

"C\_FIRE\_DISPL":"ASC",

"C\_TYPE\_NAME":"ASC",

"C\_MARK":"ASC"

}

}

Запрос представления с использованием массива order

{

"class":"T\_AUTO",

"view":"V\_AUTO\_W",

"limit": 200,

"base": 0,

"order":

[

{

"C\_FIRE\_DISPL":"ASC"

},

{

"C\_TYPE\_NAME":"ASC",

"C\_MARK":"ASC"

}

]

}

// Запрос обновления чата

{

"class":"T\_SYSTEM",

"view":"V\_CHAT\_W",

"order": {

"ID":"ASC"

},

"filter": {

"ID": {

"gt": 0

},

"subcondition": [{

"C\_USER":"impa"

}, {

"C\_TARGET\_SYS":"\_ALL\_FREQ"

}, {

"subcondition1": [{

"ID\_BASE": null

}, {

"ID\_BASE": 0

}, {

"ID\_BASE": 1

}],

"subcondition2": [{

"C\_TARGET\_SYS": null

}, {

"C\_TARGET\_SYS": {

"in": ["\_ALL\_", "\_ALL\_OTHER\_"]

}

}]

}]

}

}

Пример вложенного запроса:

{

"class":"T\_POLL\_LOG",

"view":"V\_POLL\_CALL",

"limit": 1,

"filter": {

"ID\_POLL": {

"in": {

"view":"V\_POLL\_EDIT\_ACTIVE",

"fields": ["ID"]

}

}

}

}

// ==> SELECT TOP 1 \* FROM V\_POLL\_CALL WHERE ID\_POLL IN (SELECT ID FROM V\_POLL\_EDIT\_ACTIVE)

Пример запроса с ref-фильтром:

{

"class":"T\_PUSH\_TOKENS",

"view":"V\_PUSH\_TOKENS",

"limit":100,

"fields": ["ID","ID\_DRIVER","ID\_PHONE","C\_DATE","C\_TOKEN\_SHOW","C\_UDID\_SHOW","C\_ACTIVE","C\_VERSION","C\_LOCALE","C\_APP\_NAME","C\_UDID"],

"filter":

{

"subcondition":

[

{"ID":10},

{"ID":11}

],

"ref":

{

"view":"V\_DRIVERS\_WORK\_4ALL",

"class":"T\_DRIVERS",

"column":"C\_TOKENS",

"data":

{

"ID":3235902

}

}

},

"order":

{

}

}

// ==> SELECT TOP 100 ID,ID\_DRIVER,ID\_PHONE,C\_DATE,C\_TOKEN\_SHOW,C\_UDID\_SHOW,C\_ACTIVE,C\_VERSION,C\_LOCALE,C\_APP\_NAME,C\_UDID FROM V\_PUSH\_TOKENS AS vw WHERE (((ID=@Param0)) OR ((ID=@Param1))) AND (ID\_DRIVER=@Param2)

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetViewColumns** | ( | string | contextId, |
|  |  | string | viewName, |
|  |  | string | classCode, |
|  |  | bool | loadInvisibleFields |
|  | ) |  |  |

Возвращает массив с полями представления

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **viewName** | Представление |
| **classCode** | Класс |
| **loadInvisibleFields** | Флаг, указывающий на необходимость загрузки скрытых полей представления |

**Возвращает**

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetViewColumnsV2** | ( | string | contextId, |
|  |  | string | viewName, |
|  |  | string | classCode, |
|  |  | bool | loadInvisibleFields, |
|  |  | string | varParams |
|  | ) |  |  |

Возвращает массив с полями представления

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **viewName** | Представление |
| **classCode** | Класс |
| **loadInvisibleFields** | Флаг, указывающий на необходимость загрузки скрытых полей представления |
| **varParams** | {"language":"value"} - вернутся колонки, переведенные на язык, который указали в value, если данный параметр не задан, берется локаль из контекста |

**Возвращает**

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetViewColumnTypes** | ( | string | contextId, |
|  |  | string | viewName, |
|  |  | string | classCode |
|  | ) |  |  |

Возвращает словарь типов колонок

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **viewName** | Представление |
| **classCode** | Класс |

Описание возвращаемого JSON

{

"ID":"int",

"ID\_BASE":

"int",

"C\_YEAR":"int",

"C\_NUM\_GAI":"varchar",

"C\_DATE\_TO":"datetime",

"C\_INFO":"varchar",

"C\_FIRE":"int",

"C\_FIRE\_DATE":"datetime",

"C\_TYPE":"int",

"ID\_MARK":"int",

"ID\_COLOR":"int",

"ID\_EXT":"int",

"C\_DATE\_START":"datetime",

"C\_USER\_CREATE":"varchar",

"C\_LIC\_NUM":"varchar",

"ID\_CLUR":"int",

"ID\_NUM\_TYPE":"int",

"C\_NUM\_SHORT":"varchar",

"C\_DATE\_REMIND":"date",

"C\_DATE\_CHECK":"date",

"C\_BASE":"varchar",

"C\_FIRE\_DISPL":"varchar",

"C\_MARK":"varchar",

"C\_COLOR":"varchar",

"C\_INFO\_DISP":"varchar",

"C\_INFO\_DISP\_SHOW":"varchar",

"C\_CHANGELOG":"varchar",

"C\_IMAGES":"bigint",

"C\_ASSIGN\_DRIVER":"varchar",

"C\_CLUR":"varchar",

"ID\_ORGANIZATION":"int",

"ID\_LIC\_REGION":"int",

"C\_LIC\_REGION":"varchar",

"ID\_AUTO\_CATEGORY":"int",

"C\_CONDITION":"int",

"C\_STATE":"int",

"C\_LENGTH":"decimal",

"C\_WIDTH":"decimal",

"C\_HEIGHT":"decimal",

"C\_CAPACITY":"decimal",

"C\_SEATS":"int",

"C\_AUTO\_CLASS":"varchar",

"C\_STATE\_SHOW":"varchar",

"ID\_LICENSE":"int",

"ID\_COUNTRY":"int",

"C\_LICENSE\_STATUS":"varchar"

}

**Возвращает**

Словарь типов колонок

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetViewData** | ( | string | contextId, |
|  |  | string | query |
|  | ) |  |  |

Выполнение запроса. Метод будет удален в будущем.

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **query** | SQL-запрос |

**Исключения**

|  |  |
| --- | --- |
| BrokerSqlException |  |
| TokenExpiredException |  |

**Возвращает**

JSON с результатами выполнения запроса

|  |  |  |  |
| --- | --- | --- | --- |
| **Stream MBroker.IBrokerService.GetViewStream** | ( | string | contextId, |
|  |  | string | queryDescription, |
|  |  | bool | extendedInfo, |
|  |  | bool | compressed |
|  | ) |  |  |

Получение потока данных представления. Параметры полностью аналогичны методу GetView

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **queryDescription** | JSON с описанием запрашиваемого представления |
| **extendedInfo** | Если установлено в true, возвращает расширенную информацию о запросе. |
| **compressed** | Если установлено в true, возвращает упакованый GZip поток |

**Возвращает**

JSON с представлением

|  |  |  |  |
| --- | --- | --- | --- |
| **Stream MBroker.IBrokerService.GetViewStreamV2** | ( | string | contextId, |
|  |  | string | queryDescription, |
|  |  | bool | extendedInfo, |
|  |  | bool | compressed, |
|  |  | string | varParams |
|  | ) |  |  |

Получение потока данных представления. Параметры полностью аналогичны методу GetView

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **queryDescription** | JSON с описанием запрашиваемого представления |
| **extendedInfo** | Если установлено в true, возвращает расширенную информацию о запросе. |
| **compressed** | Если установлено в true, возвращает упакованый GZip поток |
| **varParams** | JSON-объект с дополнительными параметрами для самого брокера |

**Возвращает**

JSON с представлением

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetViewV2** | ( | string | contextId, |
|  |  | string | queryDescription, |
|  |  | bool | extendedInfo, |
|  |  | string | varParams |
|  | ) |  |  |

Получение представления Для varParams: опционально передаётся объект с полями: [bool] isAuto – показывает, является ли запрос автоматизированным, либо он инициирован пользователем интерактивно

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **queryDescription** | JSON с описанием запрашиваемого представления |
| **extendedInfo** | Если установлено в true, возвращает расширенную информацию о запросе. |
| **varParams** | JSON-объект с дополнительными параметрами для самого брокера |

**Возвращает**

JSON с представлением

**Исключения**

|  |  |
| --- | --- |
| BrokerSqlException |  |
| TokenExpiredException |  |
| ServiceUserException |  |

Формат JSON-объекта с дополнительными параметрами для самого брокера

{

"isAuto": value // Определяет автоматический запрос или сгенерированный вручную. Если true - автоматический, false - вручную

"isAsync": value // Параметр определяет будет ли запрос выполняться асинхронно. Если true - запрос выполняется асинхронно, false - синхронно

"priority": 0/1/2/3 // Приоритет запроса. Если не указан, для всех запросов приоритет будет равен 1

"ttl": 1 // Время жизни запроса. Используется совместно с приоритетом. Если параметр не указан, время жизни для всех запросов будет равно 1 минуте.

"useFieldCache": value //

"tz":"UTC"// Таймзона. Все поля представления формата Даты/Времени будут приведены к данной таймзоне

"idShard": value // Идентификатор шарда, на котором будет выполняться запрос

"include\_block\_base":true// Включает заблокированные подразделения

"language": "value"// Локализация результата запроса. Если данный параметр не задан, берется локаль из контекста

}

Пояснение к параметру "isAuto".

Если в течение интервала времени t от клиентского приложения поступают только автоматические запросы, закрываем сессию, что влечет за собой фиксацию времени закрытия в журнале сессий пользователя. Таким образом, после суммирования всех фиксаций и умножения на интервал t получим время, которое пользователь не работал.

Пояснение к параметру "isAsync".

При использовании параметра isAsync учитывается время выполнения асинхронного запроса. В случае, если запрос выполняется быстро, возвращается JSON с результатом. Если запрос долгий, то возвращается строка с числом, по которому осуществляется получение результата запроса при помощи функции **[GetResultQuery](https://docs.technology45.ru/mbroker/html/interface_m_broker_1_1_i_broker_service.html" \l "aac730b2907c8231f3225c0ac004d7aa2)**/n.

В случае, если запрос асинхронный, статус его выполнения необходимо периодически запрашивать функцией **[GetQueryStatus](https://docs.technology45.ru/mbroker/html/interface_m_broker_1_1_i_broker_service.html" \l "a7326fc35b7438cf750dff2b50a495620)**. Если этого не делать, запрос будет отменет и удален из коллекции запросов контекста.

Пояснение к параметру "tz"  
Если параметр "tz" существует и не null, то в зависимости от окончания полей представления:  
С...UTCTZ - Оставляем как есть в UTC  
C...LOCALTZ - приводим к таймзоне, которая прописана для ID\_BASE записи, если ID\_BASE не нашлось, то приводим к переданной таймзоне.

Формат JSON-запроса к представлению:

{

"base": BASEID, // ID филиала. Если 0 или отсутствует - все доступные филиалы

"view":"VIEW\_NAME", // Имя представления или функции. Обязательный параметр

"params": [ "PARAM1", "PARAM2"], // Список параметров для хранимой функции. Обязательный для вызовов функций.

"limit": ROWCOUNT, // Максимальное количество строк. Если не указано или равно 0, то возвращается полный список

"fields": [ "FIELD1", "FIELD2"], // Запрашиваемые поля. Если не указано, возвращаются все поля представления

"filter": ..., // Фильтр (тип фильтра задается в нижнем регистре)

"groupby": [ "FIELD1", "FIELD2"], // Группирование

"order": { "FIELD1": "DESC"} // Порядок сортировки

}

Описание поля групировки  
Операция группировки, поддерживает возможность группировать результаты выборки по датам с учетом выбранного клиентом метода сокращения даты Пример поля groupby:

{

"groupby":["C\_USER\_CREATE", {"C\_DATE\_CREATE":{"trunc":"year"}}]}

"groupby":["C\_USER\_CREATE", {"C\_DATE\_CREATE":{"part":"year"}}]}

}

Методы работы с датой:

{

"trunc"- Округление даты до недели, месяца, квартала, года

"part"- Получение полей времени

}

Возможные значения trunc/part:

{

"minute"

"hour"

"day"

"week"

"month"

"quarter"

"year"

}

Пример запроса с использованием groupby и сокращением даты

{

{

"view":"V\_ORDERS",

"class":"T\_ORDERS",

"fields":

[

"C\_USER\_CREATE",

{"count": "ID","name": "C\_SUM"}

],

"groupby":

[

"C\_USER\_CREATE",

{

"C\_DATE\_CREATE":

{

"part":"year"

}

}

]

}

}

Возвращает количество созданных заявок для каждого пользователя в течении года Транслируется в следующий SQL код:

SELECT DATEPART(year,C\_DATE\_CREATE) AS C\_DATE\_CREATE\_PART\_YEAR,C\_USER\_CREATE,COUNT(ID) AS C\_SUM FROM V\_ORDERS AS vw GROUP BY C\_USER\_CREATE,DATEPART(year,C\_DATE\_CREATE)

Описание фильтра  
Если фильтр представлен в виде массива, то элементы объединяются условием OR. Если фильтр представлен одним объектом, то свойства объединяются условием AND.  
Пример AND-фильтра:

{

"C\_FIELD1":"text1",

"C\_FIELD2":"text2"

}

// Данный JSON будет транслирован в SQL-условие WHERE C\_FIELD1='text1' AND C\_FIELD2='text2'

Пример OR-фильтра:

[

{"C\_FIELD1": "text1"},

{"C\_FIELD2": "text2"}

]

// Данный JSON будет транслирован в SQL-условие WHERE C\_FIELD1='text1' OR C\_FIELD2='text2'

Агрегатные функции.

В списке запрашиваемых полей поддерживаются агрегатные функции, такие как sum, avg, min, max, count.

Общий формат:

{"[операция]":"FIELD", "name":"[имя\_результа]" }

Пример использования агрегатной функции в списке запрашиваемых полей:

"fields": [ "ID\_BASE", { "sum":"C\_SUM", "name":"C\_SUM\_TOTAL" } ]

// Данный запрос в SQL выглядит как SELECT ID\_BASE, SUM(C\_SUM) AS C\_SUM\_TOTAL [...]

Операции над полями.

Поддерживаются операции coalesce и cast в фильтре и списке запрашиваемых полей.  
Формат для использования в списке запрашиваемых полей:

{"coalesce": [ "C\_FIELD","DEFAULT\_VALUE"] }

// Результат - COALESCE(C\_FIELD, "DEFAULT\_VALUE")

{"cast": [ "C\_FIELD","SOME\_SQL\_TYPE" ] }

// Результат - CAST(C\_FIELD AS SOME\_SQL\_TYPE)

Формат операций в фильтре:

{"ID\_BASE": { "coalesce": 0, "cast": "SQL\_TYPE"...[условия]... }}

Например:

{"ID\_BASE": { "coalesce": 0, "gt": 0 } }

// Дает в SQL условие COALESCE(ID\_BASE, 0)>0

{"ID\_BASE": { "cast":"VARCHAR(10)", "like":"1%" } }

// Дает условие CAST(ID\_BASE AS VARCHAR(10)) LIKE '1%'

Формат условных операций  
Общий формат условия:

{"C\_FIELD": { "условие":"значение" } }

Возможные условия:

|  |  |  |
| --- | --- | --- |
| **Условие** | **Получаемый SQL** |  |
| {"C\_FIELD": { "eq": 10 } } | C\_FIELD=10 | |
| {"C\_FIELD": { "neq": 10 } } | C\_FIELD<>10 | |
| {"C\_FIELD": { "gt": 10 } } | C\_FIELD>10 | |
| {"C\_FIELD": { "lt": 10 } } | C\_FIELD<10 | |
| {"C\_FIELD": { "gte": 10 } } | C\_FIELD>=10 | |
| {"C\_FIELD": { "lte": 10 } } | C\_FIELD<=10 | |
| {"C\_FIELD": { "in": [10, 20, 30] } } | C\_FIELD IN (10, 20, 30) | |
| {"C\_FIELD": { "nin": [10, 20, 30] } } | C\_FIELD NOT IN (10, 20, 30) | |
| {"C\_FIELD": { "like":"%text%" } } | C\_FIELD LIKE '%text%' | |
| {"C\_FIELD": { "nlike":"%text%" } } | C\_FIELD NOT LIKE '%text%' | |

Условие может дополняться индексом, если в рамках одной конструкции несколько однотипных. Например:

{"ID\_BASE": { "neq1": null, "neq2": 0 } }

// JSON будет транслирован в SQL как "ID\_BASE IS NOT NULL AND ID\_BASE<>0"

Вложенные условия задаются с помощью ключевого слова "subcondition". Например:

{

"C\_INFO": { "neq": null }

"subcondition": [ { "ID\_BASE": null }, {"ID\_BASE": 0 } ]

}

// Фильтр задает SQL-условие C\_INFO IS NOT NULL AND (ID\_BASE IS NULL OR ID\_BASE=0)

Допускается нумерация вложенных условий, если их несколько в рамках одной конструкции ("subcondition1", "subcondition2", "subcondition3", etc.)

**Примеры фильтров:**

|  |  |  |
| --- | --- | --- |
| **JSON** | **SQL** |  |
| [ { "ID\_BASE": null }, {"ID\_BASE": 0 }] | ID\_BASE IS NULL OR ID\_BASE=0 | |
| {"ID": { "gt": 100, "lt": 200 },"C\_FIELD": { "neq": 900 } } | ID>100 AND ID<200 AND C\_FIELD<>900 | |
| [  {  "ID": { "in": [ 100, 101, 102 ] },  "C\_FIELD": { "like": "%something"}  },  {  "C\_FAIL": 0  }  ] | (ID IN (100, 101, 102) AND C\_FIELD LIKE '%something') OR (C\_FAIL=0) | |

**Примеры запросов:**

// Запрос списка классов

{"class": "T\_SYSTEM","view": "psGetClassesFX","params": [0,"username"] }

// Обновление списка событий

{

"class":"T\_SYSTEM",

"view":"V\_EVENTS",

"filter":

{

"C\_NAME": { "neq1": null, "neq2":"UNKNOWN" },

"C\_DATE\_UPDATE": { "gt":"\/Date(1350596695633+0600)\/"}

}

}

// Запрос представления

{

"class":"T\_AUTO",

"view":"V\_AUTO\_W",

"limit": 200,

"base": 0,

"order":

{

"C\_FIRE\_DISPL":"ASC",

"C\_TYPE\_NAME":"ASC",

"C\_MARK":"ASC"

}

}

// Запрос обновления чата

{

"class":"T\_SYSTEM",

"view":"V\_CHAT\_W",

"order": {

"ID":"ASC"

},

"filter": {

"ID": {

"gt": 0

},

"subcondition": [{

"C\_USER":"impa"

}, {

"C\_TARGET\_SYS":"\_ALL\_FREQ"

}, {

"subcondition1": [{

"ID\_BASE": null

}, {

"ID\_BASE": 0

}, {

"ID\_BASE": 1

}],

"subcondition2": [{

"C\_TARGET\_SYS": null

}, {

"C\_TARGET\_SYS": {

"in": ["\_ALL\_", "\_ALL\_OTHER\_"]

}

}]

}]

}

}

Пример вложенного запроса:

{

"class":"T\_POLL\_LOG",

"view":"V\_POLL\_CALL",

"limit": 1,

"filter": {

"ID\_POLL": {

"in": {

"view":"V\_POLL\_EDIT\_ACTIVE",

"fields": ["ID"]

}

}

}

}

// ==> SELECT TOP 1 \* FROM V\_POLL\_CALL WHERE ID\_POLL IN (SELECT ID FROM V\_POLL\_EDIT\_ACTIVE)

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetWebContext** | ( | string | user, |
|  |  | string | password, |
|  |  | string | cookie, |
|  |  | string | confirmCode |
|  | ) |  |  |

Получение контекста пользователя

**Аргументы**

|  |  |
| --- | --- |
| **user** | Имя пользователя |
| **password** | Пароль |
| **cookie** | Cookie web-клиента |
| **confirmCode** | Код SMS-подтверждения |

**Исключения**

|  |  |
| --- | --- |
| InvalidUserException |  |
| LoginFailedException |  |
| PasswordExpiredException |  |
| ServiceUserException |  |
| SqlOperationException |  |
| TokenLimitedException |  |
| UnknownUserException |  |
| UserLockedOutException |  |

**Возвращает**

Идентификатор контекста пользователя

Вызов метода сбрасывает кэш пользователя и соединения с зеркалами.

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetWebContextBySt** | ( | string | clientAppUrl, |
|  |  | string | ticket, |
|  |  | string | clientHostName, |
|  |  | string | clientAppName, |
|  |  | string | varParams |
|  | ) |  |  |

Получение контекста для приложений в которых уже пройдена авторизация через CAS

**Аргументы**

|  |  |
| --- | --- |
| **clientAppUrl** | Url клиентского приложения |
| **clientHostName** | Имя хоста |
| **ticket** | Токен |
| **clientAppName** | Имя клиентского приложения |
| **varParams** | Дополнительные параметры |

Формат возвращаемого JSON-объекта

{

"ContextId":"9473277e-b491-45a2-979e-1b1efdf0c8be",

"PasswordExpirationDaysLeft":16,

"LoginResult":

[

{

"COMPNAME":"",

"LOGINRES":"LOGINOK",

"FIO":"Ядрышников Андрей Евгеньевич",

"IDBASE":178,

"IDWORKERGROUP":27,

"GROUPS":"|\_Admin|\_AESAdmin|\_Auditview|\_BrokerAdmin|\_DBAgentView|\_DriverManager|\_FilesView|\_InterfaceMaxim|\_ITDivision|\_Messtabview|\_Ordersarcview|\_SDKAdmin|\_Sessionview|\_Systeminterface|\_TimeSheet|\_TimeSheetLog|\_WebPanelView",

"LINE":"SIP/1371",

"ATSTYPE":1,

"WPTYPE":2,

"IDWORKER":7307,

"IDWORKPLACE":null,

"ATSQUEREG":0,

"FREQ":"1371",

"ATSADDRTAPI":"10.1.5.8:5042",

"WEBARMTAPI":"{\"stompTcp\":\"tcp://voip-pbx-sbc.infranet.co:61612\", \"stompWs\": \"wss://arm.taxsee.ru/tapi\"}",

"PERSONID":"d204a888-9ea3-11e3-80f4-0025902d5b9f",

"IDORGANIZATION":312,

"DOMAINLOGIN":"yadryshnikov\_ae",

"SIPADDR":"10.1.5.8",

"IDORGANIZATIONTYPE":0,

"IDSHARD":2,

"TIMEZONE":"Asia/Yekaterinburg"

}

]

}

**Возвращает**

JSON-объект

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetWebContextV2** | ( | string | user, |
|  |  | string | password, |
|  |  | string | cookie, |
|  |  | string | confirmCode, |
|  |  | string | hostName, |
|  |  | string | clientAppName, |
|  |  | string | varParams |
|  | ) |  |  |

Получение контекста пользователя

**Аргументы**

|  |  |
| --- | --- |
| **user** | Имя пользователя |
| **password** | Пароль |
| **cookie** | Cookie web-клиента |
| **confirmCode** | Код SMS-подтверждения |
| **hostName** | Название рабочей станции клиента |
| **clientAppName** | Название клиентского приложения |
| **varParams** | Дополнительные параметры в формате JSON |

**Исключения**

|  |  |
| --- | --- |
| InvalidUserException |  |
| LoginFailedException |  |
| PasswordExpiredException |  |
| ServiceUserException |  |
| SqlOperationException |  |
| TokenLimitedException |  |
| UnknownUserException |  |
| UserLockedOutException |  |

**Возвращает**

Идентификатор контекста пользователя, признак ограниченности контекста (IsLimited), количество дней до окончания срока действия пароля (PasswordExpirationDaysLeft)

Вызов метода сбрасывает кэш пользователя и соединения с зеркалами.

Формат varParams

{

"language": "value"// Определяет локаль подключившегося пользователя, если параметр не не задан, то автоматически русская локаль

}

Локаль задается текстовой стройкой:  
"ru" - русская локаль  
"de" - немецкая локаль  
"uk" - украинская локаль

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.GetWorkerGroups** | ( | string | contextId, |
|  |  | int | workerId |
|  | ) |  |  |

Возвращает группы доступа пользователя

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **workerId** | Идентификатор сотрудника |

**Возвращает**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void MBroker.IBrokerService.KillContext** | ( | string | contextId | ) |  |

Уничтожает контекст пользователя из кэша брокера. При уничтожении контекста закрываются все соединения с базой данных и зеркалами.

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |

**Исключения**

|  |  |
| --- | --- |
| TokenExpiredException |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.KillContextV2** | ( | string | contextId, |
|  |  | int | exitCode |
|  | ) |  |  |

Уничтожает контекст пользователя из кэша брокера. При уничтожении контекста закрываются все соединения с базой данных и зеркалами.

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **exitCode** | Код возврата |

**Исключения**

|  |  |
| --- | --- |
| TokenExpiredException |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.ListContexts** | ( | string | contextId | ) |  |

Вернуть список активных контекстов.

**Возвращает**

Превьюха

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.MakeActivityForbidden** | ( | string | contextId, |
|  |  | string | varParams |
|  | ) |  |  |

Сделать сервис неактивным

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **varParams** | Дополнительные параметры |

Офисание дополнительных параметров

{

"userForbidden": {"user":isForbidden} // Словарь "Имя пользователя - Флаг активности" используется, для задания параметра Активности сервиса для конкретного пользователя

}

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.MakeActivityPermitted** | ( | string | contextId, |
|  |  | string | varParams |
|  | ) |  |  |

Сделать сервис активным

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **varParams** | Дополнительные параметры |

Офисание дополнительных параметров

{

"userForbidden": {"user":isForbidden} // Словарь "Имя пользователя - Флаг активности" используется, для задания параметра Активности сервиса для конкретного пользователя

}

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.MarkMessageAsRead** | ( | int | messageId, |
|  |  | string | contextId |
|  | ) |  |  |

Метод отмечает сообщение как прочитанное

**Аргументы**

|  |  |
| --- | --- |
| **messageId** | Идентификатор сообщения |
| **contextId** | Контекст пользователя |

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.ObjLock** | ( | string | contextId, |
|  |  | string | className, |
|  |  | Int64 | id |
|  | ) |  |  |

Блокировка объекта

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **className** | Класс объекта, например T\_ORDERS |
| **id** | Идентификатор объекта |

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.ObjTouchLock** | ( | string | contextId, |
|  |  | string | className, |
|  |  | Int64 | id |
|  | ) |  |  |

Продление блокировки объекта

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **className** | Класс объекта, например T\_ORDERS |
| **id** | Идентификатор объекта |

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.ObjUnLock** | ( | string | contextId, |
|  |  | string | className, |
|  |  | Int64 | id |
|  | ) |  |  |

Разблокировка объекта

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **className** | Класс объекта, например T\_ORDERS |
| **id** | Идентификатор объекта |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void MBroker.IBrokerService.ObjUnLockAll** | ( | string | contextId | ) |  |

Стяние всех блокировок

**Аргументы**

|  |  |
| --- | --- |
| **id** | Идентификатор объекта |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void MBroker.IBrokerService.Ping** | ( | string | contextId | ) |  |

Пинг брокера. Требуется для поддержания соединения, если пользователь не выполняет никаких операций.

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |

**Исключения**

|  |  |
| --- | --- |
| TokenExpiredException |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void MBroker.IBrokerService.PingSession** | ( | string | contextId | ) |  |

Сброс времени простоя сессии

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.PsGetObjectsDesc** | ( | string | contextId, |
|  |  | string | filter, |
|  |  | string | varParams |
|  | ) |  |  |

Получение описания объектов с учетом фильтра

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **filter** | Фильтр |
| **varParams** | Дополнительные параметры |

**Возвращает**

Описание объектов

Описание параметра *filter*

{

"object\_types":[],//Список типов объектов

"classes":[],//Список классов

"condition"://Обект условия выборки

[

{

"field":

{

"condition":"value"

}

}

]

}

**Варианты всех возможных условий "condition" для фильтрации выборки**

|  |  |  |
| --- | --- | --- |
| **Условие** | **Описание** |  |
| {"C\_FIELD": { "eq": 10 } } | Сравнение равенства | |
| {"C\_FIELD": { "gt": 10 } } | Представляет числовое сравнение объектов "больше, чем" | |
| {"C\_FIELD": { "gte": 10 } } | Представляет числовое сравнение объектов "больше или равно" | |
| {"C\_FIELD": { "lte": 10 } } | Представляет числовое сравнение объектов "меньше или равно" | |
| {"C\_FIELD": { "lt": 10 } } | Представляет числовое сравнение объектов "меньше, чем" | |
| {"C\_FIELD": { "ne": 10 } } | Предствляет сравнение неравенства | |
| {"C\_FIELD": { "in":"text" } } | Предствляет проверку условия "содержится" | |
| {"C\_FIELD": { "sw":"text" } } | Предствляет проверку условия "Начинается с" | |
| {"C\_FIELD": { "ew":"text" } } | Предствляет проверку условия "Заканчивается на" | |

Условия фильтра могут быть связаны логическим "И" либо логическим "ИЛИ". Одновременно оба варианта в данной версии не поддерживаются.  
**Пример связывания условий логическим "ИЛИ"**

{

"object\_types":[],//Список типов объектов

"classes":[],//Список классов

"condition"://Обект условия выборки

[

{

"field":

{

"condition":"value"

}

},

{

"field2":

{

"condition2":"value2"

}

}

]

}

**Пример связывания условий логическим "И"**

{

"object\_types":[],//Список типов объектов

"classes":[],//Список классов

"condition"://Обект условия выборки

{

"field":

{

"condition":"value"

},

"field2":

{

"condition2":"value2"

}

}

}

**Варианты всех возможных значений элементов списка "object\_types"**

|  |  |  |
| --- | --- | --- |
| **Значение** | **Описание** |  |
| "VIEW" | Значение для указания необходимости выборки представлений | |
| "PROC" | Значение для указания необходимости выборки хранимых процедур | |
| "CLASS" | Значение для указания необходимости выборки классов | |

Примеры возвращаемых значений

{"object\_types":["VIEW"],"classes":["T\_AUTO"],"condition":[{"C\_CODE":{"in":"V\_AUTO"}}]}

{

"C\_VIEWS":

[

{

"IDSQL":"taxi\_driver.V\_AUTO",

"C\_CODE":"V\_AUTO\_FIND",

"C\_NAME":"Поиск автомобилей",

"C\_CLASS":"T\_AUTO",

"C\_ORDER":12,

"C\_ACCESS\_FLAG":"CHECK\_BASE",

"C\_PROPERTIES":"MIRROR=maindb\_ro|VISIBLE=NO"

},

...

]

}

{"object\_types":["PROC"],"classes":["T\_AUTO"]}

{

"C\_PROCS":

[

{

"IDSQL":"taxi\_driver.autodopcheck",

"C\_CODE":"AutoDopCheck",

"C\_NAME":"Проверка наличия доп параметра у автомобиля",

"C\_CLASS":"T\_AUTO",

"C\_ACCESS\_FLAG":"CHECK\_BASE",

"C\_PROPERTIES":"VISIBLE=NO|MIRROR=maindb\_ro",

"C\_TYPE":"DUMMY"

},

...

]

}

{"object\_types":["CLASS"],"classes":["T\_AUTO"]}

{

"C\_CLASSES":

[

{

"ID":8806,

"C\_CODE":"T\_AUTO",

"C\_NAME":"Автомобили справочник",

"C\_PROPERTIES":"GROUPOPER=YES",

"C\_ACCESS\_FLAG":"CHECK\_BASE",

"C\_TYPE":"TAXI"

}

]

}

{"object\_types":["CLASS","VIEW", "PROC"],"classes":["T\_AUTO"]}

{

"C\_CLASSES":

[

{

}

],

"C\_VIEWS":

[

{

}

],

"C\_PROCS":

[

{

}

]

}

*varParams* может принимать следующие значения

{

"language":"ru"- Язык локализации описания объектов. Данный параметр является приоритетнее языка контекста

"worker": {"id\_worker":N,"id\_worker\_group":C, "login": L} - пользователь для которого необходимо вернуть права

}

/// Описание дополнительных параметров

Объект "worker" параметра varParams используется для получения прав пользователя отличного от текущего. Если поля id\_worker, id\_worker\_group объекта "worker" имеют значения отличные от null, то приоритет их использования больше поля "login". </description?

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.PsGetObjectsFx** | ( | string | contextId, |
|  |  | string | objectType, |
|  |  | string | classCode, |
|  |  | string | varParams |
|  | ) |  |  |

Получение описания объектов базы данных

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **objectType** | Тип объекта |
| **classCode** | Код класса |
| **varParams** | Дополнительные параметры |

**Возвращает**

Описание объектов

Описание входных параметров  
*objectType* может принимать следующие значения

{

"VIEW"- Получение описания представлений (обязательно задание значения параметра <paramref name="classCode"/>) \n

"PROC"- Получение описания хранимых процедур (обязательно задание значения параметра <paramref name="classCode"/>)

}

*varParams* может принимать следующие значения

{

"include\_desc":"true/false"- флаг, позволяющий включить/исключить описание объектов в выборку (по умолчанию описание включено)

"language":"ru"- Язык локализации описания объектов. Данный параметр является приоритетнее языка контекста

"worker": {"id\_worker":N,"id\_worker\_group":C,"login":L} - пользователь для которого необходимо вернуть права

}

Описание дополнительных параметров

Объект "worker" параметра varParams используется для получения прав пользователя отличного от текущего. Если поля id\_worker, id\_worker\_group объекта "worker" имеют значения отличные от null, то приоритет их использования больше поля "login". </description? Формат возвращаемого значения при значение параметров *objectType* ="VIEW", *classCode* ="T\_AUTO", *varParams* "include\_desc"="false"

[

{

"Key":

{

"C\_CLASS":"T\_AUTO",

"C\_CODE":"V\_AUTO\_FIND"

},

"Value":

{

"C\_DESC":null,

"C\_ACR":

{

"C\_BASES":[0]

}

}

},

{

"Key":

{

"C\_CLASS":"T\_AUTO",

"C\_CODE":"V\_AUTO\_MY\_TODAY"

},

"Value":

{

"C\_DESC":null,

"C\_ACR":

{

"C\_BASES":[0]

}

}

},

...

]

Формат возвращаемого значения при значение параметров *objectType* ="VIEW", *classCode* ="T\_AUTO", *varParams* "include\_desc"="true"

[

{

"Key":

{

"C\_CLASS":"T\_AUTO",

"C\_CODE":"V\_AUTO\_FIND"

},

"Value":

{

"C\_DESC":

{

"C\_CUSTOM":0,

"IDSQL":"taxi\_driver.V\_AUTO",

"ViewCode":"V\_AUTO\_FIND",

"C\_CODE":"V\_AUTO\_FIND",

"C\_NAME":"Поиск автомобилей",

"C\_CLASS":"T\_AUTO",

"C\_ORDER":12,

"C\_ACCESS\_FLAG":"CHECK\_BASE",

"C\_FILTER":null,

"C\_PROPERTIES":"MIRROR=maindb\_ro|VISIBLE=NO"

},

"C\_ACR":

{

"C\_BASES":[0]

}

}

},

...

]

Формат возвращаемого значения при значение параметров *objectType* ="PROC", *classCode* ="T\_AUTO", *varParams* "include\_desc"="false"

[

{

"Key":

{

"C\_TYPE":"EDIT",

"C\_CLASS":"T\_AUTO",

"C\_CODE":"AutoSet4Web"

},

"Value":

{

"C\_DESC":null,

"C\_ACR":

{

"C\_BASES":[0]

}

}

},

{

"Key":

{

"C\_TYPE":"DUMMY",

"C\_CLASS":"T\_AUTO",

"C\_CODE":"AutoDopCheck"

},

"Value":

{

"C\_DESC":null,

"C\_ACR":

{

"C\_BASES":[0]

}

}

},

...

]

Формат возвращаемого значения при значение параметров *objectType* =NULL, *classCode* ="T\_AUTO", *varParams* "include\_desc"="true"

{

"C\_PROCS\_FX":

[

{

"Key":

{

"C\_TYPE":"EDIT",

"C\_CLASS":"T\_AUTO",

"C\_CODE":"AutoSet4Web"

},

"Value":

{

"C\_DESC":

{

"IDSQL":

"taxi\_driver.autoset\_4web",

"C\_CODE":"AutoSet4Web",

"C\_NAME":"Добавить/изменить из водительского кабинета",

"C\_CLASS":"T\_AUTO",

"C\_ACCESS\_FLAG":"CHECK\_BASE",

"C\_PROPERTIES":"WEBONLY=YES|VISIBLE=NO|MIRROR=maindb",

"C\_TYPE":"EDIT"

},

"C\_ACR":

{

"C\_BASES":[0]

}

}

},

...,

],

"C\_VIEWS\_FX":

[

{

"Key":

{

"C\_CLASS":"T\_AUTO",

"C\_CODE":"V\_AUTO\_FIND"

},

"Value":

{

"C\_DESC":

{

"C\_CUSTOM":0,

"IDSQL":"taxi\_driver.V\_AUTO",

"ViewCode":"V\_AUTO\_FIND",

"C\_CODE":"V\_AUTO\_FIND",

"C\_NAME":"Поиск автомобилей",

"C\_CLASS":"T\_AUTO",

"C\_ORDER":12,

"C\_ACCESS\_FLAG":"CHECK\_BASE",

"C\_FILTER":null,

"C\_PROPERTIES":"MIRROR=maindb\_ro|VISIBLE=NO"

},

"C\_ACR":

{

"C\_BASES":[0]

}

}

},

...

]

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void MBroker.IBrokerService.RebuildMirrorList** | ( | string | contextId | ) |  |

Обновить список алиасов из конфига.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void MBroker.IBrokerService.ResetAddressCache** | ( | string | contextId | ) |  |

Сброс адресного кэша

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.RetrieveAddressTemplates** | ( | string | contextId, |
|  |  | int[] | idPlaces, |
|  |  | string | varParams |
|  | ) |  |  |

Получение списка всех адресов для заданных филиалов

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **idPlaces** | Идентификаторы филиалов |
| **varParams** | Дополнительные параметры |

///

Пример возвращаемого значения

[

{

"C\_TYPE":1,

"ID\_PLACE":101,

"C\_PLACE":"Советск",

"ID\_ADDR":null,

"C\_ADDR\_NAME":null,

"ID\_STREET":3165689,

"C\_STREET\_NAME":"Академика Шишкина",

"C\_NUM\_HOUSE":null,"ID\_ZONE":null,

"C\_ZONE\_NAME":null,

"C\_SYNONYM":null,

"C\_NOTE":null,

"ID\_BASE":null,

"C\_LATITUDE":null,

"C\_LONGITUDE":null,

"ID\_CITY":null,

"C\_CITY":null,

"ID\_TYPE\_ADDR":null,

"C\_NAME":null,

"C\_DESCRIPTION":null,

"ID\_COUNTRY":null,

"C\_POPULATION":null,

"ID\_REGION":null,

"C\_IS\_ADDR":null,

"C\_SELECTED":false,

"C\_MEET\_POINT":null,

"ID":0

},

{

"C\_TYPE":1,

"ID\_PLACE":101,

"C\_PLACE":"Советск",

"ID\_ADDR":null,

"C\_ADDR\_NAME":null,

"ID\_STREET":3165690,

"C\_STREET\_NAME":"Водопроводная",

"C\_NUM\_HOUSE":null,

"ID\_ZONE":null,

"C\_ZONE\_NAME":null,

"C\_SYNONYM":null,

"C\_NOTE":null,

"ID\_BASE":null,

"C\_LATITUDE":null,

"C\_LONGITUDE":null,

"ID\_CITY":null,

"C\_CITY":null,

"ID\_TYPE\_ADDR":null,

"C\_NAME":null,

"C\_DESCRIPTION":null,"

ID\_COUNTRY":null,

"C\_POPULATION":null,

"ID\_REGION":null,

"C\_IS\_ADDR":null,

"C\_SELECTED":false,

"C\_MEET\_POINT":null,

"ID":0

}

]

Формат varParams

{

"language": "value"// Если данный параметр не задан, берется локаль из контекста

}

Локаль задается текстовой строкой:

"ru"- русская локаль\n

"de"- немецкая локаль\n

"uk"- украинская локаль\n

**Возвращает**

Список адресов

|  |  |  |  |
| --- | --- | --- | --- |
| **Stream MBroker.IBrokerService.RetrieveAddressTemplatesStream** | ( | string | contextId, |
|  |  | int[] | idPlaces, |
|  |  | bool | compressed, |
|  |  | string | varParams |
|  | ) |  |  |

Получение списка всех адресов для заданных филиалов

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **idPlaces** | Идентификаторы филиалов |
| **compressed** | Если установлено в true, возвращает упакованый GZip поток |
| **varParams** | Дополнительные параметры |

**Возвращает**

Поток данных адресного кэша для заданных филиалов.  
Пример возвращаемого значения описан в документации к методу **[RetrieveAddressTemplates](https://docs.technology45.ru/mbroker/html/interface_m_broker_1_1_i_broker_service.html" \l "a20257f7d81000b40026c544c6f1c56ff)**

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.SaveFileInfo** | ( | string | contextId, |
|  |  | string | filename, |
|  |  | string | className, |
|  |  | int | idRecord, |
|  |  | string | type, |
|  |  | string | fileType, |
|  |  | string | guid, |
|  |  | int | idBd |
|  | ) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.SendDepsEvent** | ( | string | user, |
|  |  | string | password, |
|  |  | string | phoneNumber, |
|  |  | string @ | event, |
|  |  | int | id |
|  | ) |  |  |

**Аргументы**

|  |  |
| --- | --- |
| **user** |  |
| **password** |  |
| **phoneNumber** |  |
| **event** |  |
| **id** |  |

**Возвращает**

Замещается в **[MBroker.WorkerBroker](https://docs.technology45.ru/mbroker/html/class_m_broker_1_1_worker_broker.html" \l "a038641378eef575f516c2754aa43fe05)** и **[MBroker.WorkerBroker](https://docs.technology45.ru/mbroker/html/class_m_broker_1_1_worker_broker.html" \l "a038641378eef575f516c2754aa43fe05)**.

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.SetLocale** | ( | string | contextId, |
|  |  | string | locale |
|  | ) |  |  |

Смена локали пользователя

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | ID контекста пользователя |
| **locale** | Локаль |

Пример задания локали

Локаль задается текстовой стройкой:\n

"ru"- русская локаль\n

"de"- немецкая локаль\n

"uk"- украинская локаль\n

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void MBroker.IBrokerService.SmsReset** | ( | string | contextId | ) |  |

Сброс состояния SMS [SMSDaemon]

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |

|  |  |  |  |
| --- | --- | --- | --- |
| **void MBroker.IBrokerService.Subscribe** | ( | string | contextId, |
|  |  | string | filter |
|  | ) |  |  |

Начать вести выделенный лог, состоящий из записей, содержащих строку, переданную в filter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void MBroker.IBrokerService.Unsubscribe** | ( | string | contextId | ) |  |

Остановить запись выделенного лога.

|  |  |  |  |
| --- | --- | --- | --- |
| **int MBroker.IBrokerService.UploadFile** | ( | string | contextId, |
|  |  | string | filename, |
|  |  | string | file, |
|  |  | string | className, |
|  |  | int | idRecord, |
|  |  | string | type, |
|  |  | string | fileType |
|  | ) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.UploadFileEx** | ( | string | contextId, |
|  |  | string | date, |
|  |  | string | fileName, |
|  |  | string | className, |
|  |  | int | idRecord, |
|  |  | string | type, |
|  |  | string | fileType, |
|  |  | string | mirror, |
|  |  | string | varParams |
|  | ) |  |  |

Метод загрузки файла на сервер

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **date** | Бинарные данные |
| **fileName** | Оригинальное имя файла |
| **className** | Имя класса (например: T\_DRIVERS) |
| **idRecord** | ID записи внутри выбранного класса |
| **type** | Ссылка на тип файла из T\_FILE\_TYPES |
| **fileType** | MIME-тип |
| **mirror** | Зеркало для загрузки файла |
| **varParams** | JSON-объект с дополнительными параметрами |

**Возвращает**

Строка с уникальным идентификатором файла

Формат JSON-объекта с дополнительными параметрами

{

"user":"value"- пользователь, загружающий файл

"replaceFileGuid":"value"- guid файла, который необходимо заменить

"idLang":"value"- идентификатор локали

"idFile":"value"- идентификатор файла из T\_FILES, если передан, обновится запись в T\_FILES, иначе создастся новая

"fileStatus":"value"- статус файла. Tсли не передан, статус файла изменится на ACTIVE

}

* Пример бинарных данных  
  "YWJj" Уникальный идентификатор файла  
  Используется для загрузки файла с сервера

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.UploadFileFBM** | ( | string | contextId, |
|  |  | string | date, |
|  |  | string | fileName, |
|  |  | string | className, |
|  |  | int | idRecord, |
|  |  | string | type, |
|  |  | string | fileType, |
|  |  | string | mirror |
|  | ) |  |  |

Метод загрузки файла на сервер

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста |
| **date** | Бинарные данные |
| **fileName** | Оригинальное имя файла |
| **className** | Имя класса (например: T\_DRIVERS) |
| **idRecord** | ID записи внутри выбранного класса |
| **type** | Ссылка на тип файла из T\_FILE\_TYPES |
| **fileType** | MIME-тип |
| **mirror** | Зеркало для загрузки файла |

**Возвращает**

Строка с уникальным идентификатором файла

* Пример бинарных данных  
  "YWJj" Уникальный идентификатор файла  
  Используется для загрузки файла с сервера

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **int MBroker.IBrokerService.UserCount** | ( |  | ) |  |

Вернуть количество залогиненных пользователей.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **string MBroker.IBrokerService.UserCountByLogin** | ( | string | login | ) |  |

Возвращает общее число залогиненных пользователей и число залогиненных с данным логином

**Аргументы**

|  |  |
| --- | --- |
| **login** | Логин пользователя |

///

* Офисание дополнительных параметров
* {
* "UserCount":3,//общее число пользователей
* "UserCountByLogin":1//число пользователелей c переданным логином
* }

**Возвращает**

|  |  |  |  |
| --- | --- | --- | --- |
| **string MBroker.IBrokerService.UserLogin** | ( | string | contextId, |
|  |  | string | hostName, |
|  |  | string | clientAppName |
|  | ) |  |  |

Подключение пользователя к сервису.

**Аргументы**

|  |  |
| --- | --- |
| **contextId** | Идентификатор контекста пользователя |
| **hostName** | Имя компьютера в сети |
| **clientAppName** | Имя клиента (приложения) |

**Возвращает**

JSON с результатом выполнения

**Исключения**

|  |  |
| --- | --- |
| [**AccessDeniedException**](https://docs.technology45.ru/mbroker/html/class_m_broker_1_1_access_denied_exception.html) |  |
| BrokerSqlException |  |
| ServiceUserException |  |
| TokenExpiredException |  |