

Common 2.0 XML RPC Interface

- 1 [Introduction](#)
- 2 [Terms and definitions](#)
- 3 [Conventions](#)
 - 3.1 [Data types formatting rules](#)
 - 3.2 [Methods Naming and Calling](#)
 - 3.3 [Method definition formatting rules](#)
- 4 [Connection and Authentication](#)
- 5 [Types](#)
 - 5.1 [Generic Types](#)
 - 5.2 [PagingList](#)
 - 5.3 [UID](#)
 - 5.4 [LicenseInfo](#)
 - 5.5 [ActivationInfo](#)
 - 5.6 [UserAccess](#)
 - 5.7 [AuthInfo](#)
 - 5.8 [NodeStatus](#)
 - 5.9 [LemmaCategory](#)
 - 5.10 [LemmaWord](#)
 - 5.11 [ProtocolFilter](#)
 - 5.12 [MimeCategory](#)
 - 5.13 [WBListVal](#)
 - 5.14 [WLRequest](#)
 - 5.15 [NodeDesc](#)
 - 5.16 [LoginHistoryRecord](#)
 - 5.17 [AdBlockInfo](#)
 - 5.18 [GroupInfo](#)
 - 5.19 [UserInfo](#)
 - 5.20 [LDAPUserInfo](#)
 - 5.21 [RadiusUserInfo](#)
 - 5.22 [TimeInterval](#)
 - 5.23 [TimeRule](#)
 - 5.24 [TrafficRule](#)
 - 5.25 [RuleCondition](#)
 - 5.25.1.1 [CategoryCondition](#)
 - 5.25.1.2 [MorphologyCondition](#)
 - 5.25.1.3 [TimeCondition](#)
 - 5.25.1.4 [TimeCondition](#)
 - 5.26 [CustomDnsInfo](#)
 - 5.27 [UrlCheckResult](#)
 - 5.28 [IMHistoryRecord](#)
 - 5.29 [IMHistoryCategory](#)
 - 5.30 [IMHistoryFilter](#)
 - 5.31 [IMExclusionRecord](#)
 - 5.32 [IMExclusionFilter](#)
 - 5.33 [WebStatInfo](#)
 - 5.34 [RegexpInfo](#)

- 5.35 [JournalEntryInfo](#)
- 5.36 [ADImportInfo](#)
- 5.37 [StorageUploadSessionInfo](#)
- 5.38 [StorageUploadSessionCheckInfo](#)
- 5.39 [StorageFileInfo](#)
- 5.40 [FilePreviewInfo](#)
- 5.41 [DlpUserInfo](#)
- 5.42 [DlpLevelInfo](#)
- 5.43 [DlpGroupInfo](#)
- 5.44 [DlpPolicyType](#)
- 5.45 [NamedListType](#)
- 5.46 [NamedList](#)
- 5.47 [SimpleNamedList](#)
- 5.48 [NListVal](#)
- 5.49 [CertInfo](#)
- 5.50 [SystemLogsInfo](#)
- 6 [Methods](#)
 - 6.1 [Component: Core](#)
 - 6.1.1 [v1.core.login](#)
 - 6.1.2 [v1.core.logout](#)
 - 6.1.3 [v1.core.login.as](#)
 - 6.1.4 [v1.core.password.recovery.start](#)
 - 6.1.5 [v1.core.password.recovery.finish](#)
 - 6.1.6 [v1.core.session.ping](#)
 - 6.1.7 [v1.core.license.info](#)
 - 6.1.8 [v1.core.license.activation](#)
 - 6.1.9 [v1.core.license.remove](#)
 - 6.1.10 [v1.core.register.master](#)
 - 6.1.11 [v1.core.register.slave](#)
 - 6.1.12 [v1.core.nodes.list](#)
 - 6.1.13 [v1.core.get.categories](#)
 - 6.1.14 [v1.core.check.url](#)
 - 6.1.15 [v1.core.public.check.url](#)
 - 6.1.16 [v1.core.temp.unblock](#)
 - 6.1.17 [v1.core.temp.unblock.get](#)
 - 6.1.18 [v1.core.user.passwd](#)
 - 6.1.19 [v1.core.nodes.list](#)
 - 6.1.20 [v1.core.node.remove](#)
 - 6.1.21 [v1.core.login.history.list](#)
 - 6.1.22 [v1.core.login.history.clear](#)
 - 6.1.23 [v1.core.public.check.urls](#)
 - 6.1.24 [v1.core.codepages.list](#)
 - 6.2 [Component: Settings](#)
 - 6.2.1 [v1.settings.get.string](#)
 - 6.2.2 [v1.settings.get.public.string](#)
 - 6.2.3 [v1.settings.set.string](#)
 - 6.2.4 [v1.settings.get.bool](#)
 - 6.2.5 [v1.settings.get.public.bool](#)
 - 6.2.6 [v1.settings.set.bool](#)

- 6.2.7 [v1.settings.set.filter.categories.level](#)
- 6.2.8 [v1.settings.get.filter.categories.level](#)
- 6.2.9 [v1.settings.get.filter.categories](#)
- 6.2.10 [v1.settings.get.filter.categories](#)
- 6.2.11 [v1.settings.suggest.categories](#)
- 6.2.12 [v1.settings.white.list.requests.add](#)
- 6.2.13 [v1.settings.white.list.requests.delete](#)
- 6.2.14 [v1.settings.white.list.requests.list](#)
- 6.2.15 [v1.settings.adblock.info](#)
- 6.2.16 [v1.settings.adblock.update](#)
- 6.2.17 [v1.settings.custom.dnses.list](#)
- 6.2.18 [v1.settings.custom.dns.add](#)
- 6.2.19 [v1.settings.custom.dns.update](#)
- 6.2.20 [v1.settings.custom.dns.delete](#)
- 6.2.21 [v1.settings.generate.certificate](#)
- 6.2.22 [v1.settings.systemlogs.status](#)
- 6.2.23 [v1.settings.systemlogs.set](#)
- 6.3 [Component: Accounts](#)
 - 6.3.1 [v2.accounts.groups.list](#)
 - 6.3.2 [v2.accounts.group.fetch](#)
 - 6.3.3 [v2.accounts.group.parents.list](#)
 - 6.3.4 [v2.accounts.group.users.list](#)
 - 6.3.5 [v2.accounts.group.add](#)
 - 6.3.6 [v2.accounts.group.update](#)
 - 6.3.7 [v2.accounts.group.delete](#)
 - 6.3.8 [v2.accounts.group.assign.rules](#)
 - 6.3.9 [v2.accounts.group.assign.nlists](#)
 - 6.3.10 [v2.accounts.users.list](#)
 - 6.3.11 [v2.accounts.user.fetch](#)
 - 6.3.12 [v2.accounts.user.add](#)
 - 6.3.13 [v2.accounts.user.update](#)
 - 6.3.14 [v2.accounts.user.delete](#)
 - 6.3.15 [v2.accounts.user.assign.rules](#)
 - 6.3.16 [v2.accounts.user.assign.nlists](#)
 - 6.3.17 [v2.accounts.ldap.import](#)
- 6.4 [Component: Named Lists](#)
 - 6.4.1 [Типы именованных списков](#)
 - 6.4.2 [v1.nlists.public.list](#)
 - 6.4.3 [v1.nlists.owned.list](#)
 - 6.4.4 [v1.nlists.list](#)
 - 6.4.5 [v1.nlists.add](#)
 - 6.4.6 [v1.nlists.update](#)
 - 6.4.7 [v1.nlists.delete](#)
 - 6.4.8 [v1.nlists.list.list](#)
 - 6.4.9 [v1.nlists.list.add](#)
 - 6.4.10 [v1.nlists.list.update](#)
 - 6.4.11 [v1.nlists.list.delete](#)
 - 6.4.12 [v1.nlists.list.import](#)
 - 6.4.13 [v1.nlists.list.export](#)

- 6.5 [Component: Rules](#)
 - 6.5.1 [v2.rules.list](#)
 - 6.5.2 [v2.rules.owned.list](#)
 - 6.5.3 [v2.rules.add](#)
 - 6.5.4 [v2.rules.delete](#)
 - 6.5.5 [v2.rules.update](#)
 - 6.5.6 [v2.rules.public.get.name](#)
- 6.6 [Component: Instant Messengers](#)
 - 6.6.1 [v1.im.message.history.list](#)
 - 6.6.2 [v1.im.message.history.clear](#)
 - 6.6.3 [v1.im.message.delete](#)
 - 6.6.4 [v1.im.exclusions.list](#)
 - 6.6.5 [v1.im.exclusion.add](#)
 - 6.6.6 [v1.im.exclusion.update](#)
 - 6.6.7 [v1.im.exclusion.delete](#)
- 6.7 [Component: Webstat](#)
 - 6.7.1 [v1.webstat.info](#)
 - 6.7.2 [v1.webstat.register](#)
 - 6.7.3 [v1.webstat.counters](#)
- 6.8 [Component: Regular Expressions](#)
 - 6.8.1 [v1.regexps.list](#)
 - 6.8.2 [v1.regexps.add](#)
 - 6.8.3 [v1.regexps.fetch](#)
 - 6.8.4 [v1.regexps.update](#)
 - 6.8.5 [v1.regexps.delete](#)
 - 6.8.6 [v1.regexps.get.protocol.filters](#)
 - 6.8.7 [v1.regexps.set.protocol.filters](#)
- 6.9 [Component: Storage](#)
 - 6.9.1 [v2.storage.init.upload.session](#)
 - 6.9.2 [v2.storage.check.upload.session](#)
 - 6.9.3 [v2.storage.file.fetchInfo](#)
 - 6.9.4 [v2.storage.file.previewInfo](#)
 - 6.9.5 [v2.storage.file.genExternalUrl](#)
 - 6.9.6 [v2.storage.file.delete](#)
 - 6.9.7 [v2.storage.files.list](#)
- 6.10 [Component: DLP](#)
 - 6.10.1 [v1.dlp.levels.list](#)
 - 6.10.2 [v1.dlp.level.add](#)
 - 6.10.3 [v1.dlp.level.update](#)
 - 6.10.4 [v1.dlp.level.delete](#)
 - 6.10.5 [v1.dlp.policy.set](#)
 - 6.10.6 [v1.dlp.policies.list](#)
 - 6.10.7 [v1.dlp.groups.list](#)
 - 6.10.8 [v1.dlp.group.add](#)
 - 6.10.9 [v1.dlp.group.update](#)
 - 6.10.10 [v1.dlp.group.delete](#)
 - 6.10.11 [v1.dlp.users.list](#)
 - 6.10.12 [v1.dlp.user.add](#)
 - 6.10.13 [v1.dlp.user.fetch](#)

- 6.10.14 [v1.dlp.user.update](#)
- 6.10.15 [v1.dlp.user.delete](#)
- 6.11 [Component: Mailsecurity](#)
 - 6.11.1 [LicenseUserEmail](#)
 - 6.11.2 [MailsecSettings](#)
 - 6.11.3 [v1.mailsec.license.users.list](#)
 - 6.11.4 [v1.mailsec.license.users.add](#)
 - 6.11.5 [v1.mailsec.license.user.fetch](#)
 - 6.11.6 [v1.mailsec.license.users.delete](#)
 - 6.11.7 [v1.mailsec.license.user.update](#)
 - 6.11.8 [v1.mailsec.settings.get](#)
 - 6.11.9 [v1.mailsec.settings.set](#)
- 7 [All Errors](#)

Introduction

Для RPC используется [XML-RPC](#).

В документе описываются XML-RPC запросы к KG/DNS серверу 2.x версий, реализованных на **Erlang**.

Terms and definitions

- *открытый метод* — метод, которые не требует предварительной аутентификации
- *защищённый метод* — метод, требующий предварительной аутентификации; первым параметром такого метода всегда передаётся токен, полученный при вызове метода аутентификации

Conventions

Data types formatting rules

Базовые типы данных обозначаются как в списке [базовых типов](#).

Новые типы (в основном это структуры, но также могут быть и списки констант) оформляются как новый раздел в блоке [Types](#). В заголовке для типа должно стоять *только* имя типа (в Camel-нотации). Затем это имя при использовании внутри документации заключается в «ссылку», например, `[#MailLetterInfo]`; такой подход используется при описании сигнатур методов, и вообще везде где только можно, нужно стараться использовать ссылки вместо просто указания типа.

Methods Naming and Calling

Для именованя методов используется следующая схема:

```

<method-name> ::= <proto-version> "." <component-namespace-part> "."
<component-method-name>
<proto-version> ::= "v1"
<component-namespace-part> ::= <alpha>+
<component-method-name> ::= <alpha> ( <ALPHA> | "." ) *
<alpha> ::= "a" .. "z"
<ALPHA> ::= "a" .. "z" "A" .. "Z"

```

Примеры валидных методов:

- v1.mail.folders.list
- v1.mail.folder.letters.list
- v1.calendar.events.list
- v1.calendar.event.create
- v1.mail.folder.markRead

Примеры невалидных методов:

- CalendarEventsList

Говоря неформально, название метода состоит из английской фразы, характеризующей действие, вместо пробелов ставится точка. В случае, если используется действие `list`, то к предыдущему слову добавляется `s`. Вот так:

- v1.mail.folders.list

а не:

- v1.mail.folder.list

Method definition formatting rules

Каждый метод оформляется по описанному ниже шаблону.

В первой строке горизонтальная черта для отделения описаний друг от друга (четыре минуса, ----). Затем в блоке `h3` идёт название метода. Затем в блоке `{panel}` идёт сигнатура метода. Сигнатура состоит из типа возвращаемого значения, собственно названия метода и списка формальных параметров с указанием типа. Не забываем про правила оформления типов данных (в примере ниже — это `[#CustomType]`). Имена формальных параметров пишутся полужирным шрифтом (в примере ниже — `*auth_token*`, `*arg*`).

h3. v1.COMPONENT.METHOD

```

{panel}List v1.COMPONENT.METHOD(String *auth_token*, [#CustomType]
*arg*) {panel}

```

Затем в тексте идёт короткое описание метода, пара строчек, описывать характер возвращаемого значения не нужно.

Затем идёт описание формальных параметров (в блоке `h6.`). Формат описания: список, один параметр — одна строка, имя параметра в звёздочках, затем три минуса и далее короткое описание параметра.

Если в параметре или возвращаемом значении идёт список, то собственно тип данных внутри списка указывается в описании (аргументов или возвращаемого значения) ниже.

`h6. Arguments`

```
* *auth_token* --- токен для аутентификации  
* *arg* --- ARGUMENT SHORT DESCRIPTION
```

Далее описывается возвращаемое значение, в блоке `h6.`

`h6. Return value`

Произвольный неформализованный текст. Не забываем указывать ссылки на нестандартные типы данных.



Метод может по логике не возвращать значение, однако в RPC такое недопустимо, поэтому действует соглашение, когда метод, не возвращающий значение по логике, в RPC возвращает Boolean true.

Далее идёт описание ошибок (в блоке `h6.`). Блок оформляется по шаблону, указанному ниже — таблица из двух колонок. В первой колонке код ошибки, во второй — текст сообщения по-английски. Сводная таблица всех ошибок в главном разделе с ошибками. В блоке ниже по сути копируются только первые две колонки из сводной таблицы.

`h6. Errors`

```
|| faultCode || description ||  
| *104* | Invalid auth token |  
| | |
```



Все ошибки должны репортироваться *только* посредством механизма ошибок RPC, не в возвращаемом значении функции. То есть если функция вызывает ошибку, это должно репортироваться посредством RPC-механизма репортирования ошибок.

Connection and Authentication

Аутентификация производится вызовом одного из методов аутентификации ([v1.core.login](#)). В случае успешного прохождения аутентификации метод возвращает токен (строку), которая должна

передаваться первым параметром в каждом защищённом методе. Вместе с токеном на стороне сервера генерируется сессия, которая устаревает за некоторое время неактивности.

Types

В описаниях составных типов и методов используются условные имена стандартных типов, для них есть соответствия в используемых RPC-протоколах.

Все строки (тип String) в кодировке UTF-8.

Generic Types

Стандартные базовые типы:

условное название типа	тип XML-RPC	описание
String	<string>	строковое значение, длина не лимитируется, единственная допустимая кодировка — UTF-8
Integer	<int>, <i4>	целое знаковое значение, четырёхбайтовое (ограничение из XML-RPC)
Integer64	<i8>	целое знаковое 64-битное целое, является расширением XML-RPC
Double	<double>	знаковое значение с плавающей точкой, двойной точности (63 бита)
Date	<dateTime.iso8601>	дата-время, например, 2010-03-22T19:25
Boolean	<boolean>	булевское значение, литералы <code>true</code> или <code>false</code>
Base64	<base64>	BLOB, закодированный в base64
List	<array>	плоский список элементов

PagingList

Структура, содержащая фрагмент списка.

key	type	description
count	Integer	общее количество элементов в списке (то есть общее количество элементов в списке, из которого возвращается фрагмент)


items List фрагмент списка, обычно соответствует некоторой странице

UID

Уникальный идентификатор объекта. GUID, UUID или что там ещё.

LicenseInfo

Структура с регистрационной информацией

key	type	description
pin_code	String	Пин-код
reg_name	String	Регистрационное имя
email	String	Электронная почта
expiry_date	Date	Дата истечения срока действия
license_type	String	одна из строк <code>normal demo trial not_registered</code>
license_error_code	Integer	последний код ошибки при проверке/активации лицензии, 0, если не было ошибок
version	String	Номер версии сервера
use_legal_person	 Boolean	Используется механизм создания юридических физических пользователей
modules	List	список информационных блоков о модулях, каждый блок — структура с полями module , expiry , license_error_code












только dnsfilter


ActivationInfo

Структура, отправляемая при активации продукта.

Обязательные поля обозначены 

key	type	description
pin_code	String 	Пин-код
reg_name	String 	Регистрационное имя
email	String 	Электронная почта
user_name	String 	Имя пользователя
last_name	String 	Фамилия пользователя
company	String 	Компания

country	String	 Страна
region	String	 Область для России
address	String	Адрес
phone	String	Телефон
fax	String	Факс
subscribe	Boolean	Подписка на новости
connections	Integer	 Число коннектов
product_name	String	название продукта, опциональная строка, указывающая, из какого именно продукта идёт запрос, сейчас допускается только одно значение — <code>mailsecurity</code>

 только для imdlp

UserAccess

Права доступа пользователя, возможные значения в виде строк:

val	description
admin	Глобальный администратор
group_admin	Администратор группы и ее подгрупп
user	Обычный пользователь системы
home_user	Обычный пользователь с упрощенным интерфейсом
dlp_user	юзер dlp, не должен вообще никак логиниться в систему

AuthInfo

Структура с результатом авторизации

key	type	description
auth_token	String	Уникальный идентификатор текущей сессии
user_id	UID	Уникальный идентификатор залогинившегося пользователя, 0 если глобальный админ
group_id	UID	Уникальный идентификатор группы залогинившегося пользователя, 0 если глобальный админ
user_access	UserAccess	Права доступа.
ui_language	String	Язык UI по дефолту для пользователя "en", "ru", "de"
node	String	Имя узла эрланг, на который залогинились

NodeStatus

Структура с информацией о ноде эрланга

key	type	description
name	String	Имя ноды
license	String	no_license/active
status	String	online/offline

LemmaCategory

Категория морфологического анализатора

key	type	description
id	Integer	Идентификатор категории
name	String	Имя категории
enabled	Boolean	Включена ли категория
weight	Int	Пороговый вес категории
words	List< LemmaWord >	Фразы, входящие в категорию
filters	List< ProtocolFilter >	Список протоколов, к которым применяется категория и действий при срабатывании

LemmaWord

Фраза категории морфологического анализатора с весом

key	type	description
id	Integer	Идентификатор
word	String	Слово
weight	Integer	Вес слова

ProtocolFilter

Действие, применяемое к протоколу

key	type	description
protocol	String	Протокол обмена или социальная сеть
action	String	Действие skip/block/spy

Remarks

Возможные значения для **protocol**

- icq
- xmpp
- vkontakte
- odnoklassniki

- facebook
- msn
- ymsg
- mrim

MimeCategory

Категория MIME-типов

key	type	description
id	Integer	Идентификатор категории
name	String	Имя категории
enabled	Boolean	Включена ли категория
types	List<String>	майм типы, напр. images/jpeg, audio/* и тд

WBListVal

Элемент черного или белого списка

key	type	description
id	Integer	Идентификатор
value	String	Значение элемента

WLRequest

Элемент списка запросов на добавление в белый список

key	type	description
link	String	Добавляемый хостнейм
node	String	узел с которого был запрос

NodeDesc

Описание узла сети эрланг

key	type	description
name	String	имя узла
license	String	статус лицензии active , no_license , expired
status	String	статус узла online , offline

LoginHistoryRecord

Запись для журнала истории входов в системе

key	type	description
date	Date	дата входа в систему
user_id	Integer	идентификатор пользователя
node	String	узел сети, с которого был выход в систему

AdBlockInfo

Информация о модуле adblock

key	type	description
enabled	Boolean	Включен или нет модуль
version	String	Версия списков
date	Date	дата последнего обновления списков

GroupInfo

Структура с информацией о группе пользователей

key	type	description
id	<u>UID</u>	Уникальный идентификатор
parent_id	<u>UID</u>	Идентификатор группы-родителя, 0 если корневая
name	String	Имя группы
enabled	Boolean	Включена ли группа пользователей
rules	List< <u>UID</u> >	Список идентификаторов правил, назначенных группе
parent_rules	List< <u>UID</u> >	Список идентификаторов правил, назначенных группе родительскими группами
















rules и **parent_rules** только для четния, возвращаются методом [v2.accounts.groups.list](#)



для назначения правил есть метод [v2.accounts.group.assign.rules](#)

UserInfo

Структура с информацией о пользователе

key	type	description
id	<u>UID</u>	Уникальный идентификатор
group_id	<u>UID</u>	Идентификатор группы, которой принадлежит пользователь
name	String	Имя пользователя

enabled		Boolean	Выключен ли пользователь
ip_address_list		List<String>	с версии 3.1 список IP-адресов или диапазонов для авторизации, каждый диапазон выглядит как $x.x.x.x-y.y.y.y$, где $x.x.x.x$ — первый адрес диапазона, $y.y.y.y$ — последний адрес, оба — включительно.
emails		List<String>	с версии 3.1 список E-mail адресов пользователя, в DLP адреса из этого списка используются для матчинга клиентов
login		String	Логин пользователя
password		String	MD5-хэш пароля пользователя (сервером не возвращается)
user_access		UserAccess	Права доступа.
ui_language		String	Идентификатор языка пользователя
ldap	 	Boolean	пользователь загружен из AD или нет
ldap_params		LDAPUserInfo	дополнительные параметры юзера из ldap-каталога
radius		Boolean	true, если пользователь из radius
radius_params		RadiusUserInfo	дополнительная информация о radius-пользователе
rules	 	List< UID >	Список идентификаторов правил, назначенных пользователю
parent_rules	 	List< UID >	Список идентификаторов правил, назначенных пользователю родительскими группами
nlists	 	List< UID >	Список идентификаторов списков(белых/черных), назначенных пользователю
parent_nlists	 	List< UID >	Список идентификаторов списков, назначенных пользователю родительскими группами
attributes		List	Список дополнительных атрибутов {name, value}
		String	не используется с версии 3.1 Значение для авторизации. IP-адрес

	String	не используется с версии 3.1 Конечное значение диапазона при авторизации ip_range. IP-адрес
	String	не используется с версии 3.1 E-mail пользователя
	String	Тип сети пользователя wan/lan
	UID	Идентификатор пользователя wan, к которому будет привязан lan пользователь
	String	Тип авторизации. ip, ip_range или dynamic_ip . В последнем случае IP-адрес не заполняется


 **wan_user** нужен только если тип пользователя type=**lan**

 поля только для чтения

rules, parent_rules, nlists и **parent_nlists** только для чтения, возвращаются методом [v2.accounts.users.list](#)

для назначения правил есть метод [v2.accounts.user.assign.rules](#)

и [v2.accounts.user.assign.nlists](#)

 **attributes** является необязательным полем.

- **limit_rate** - Integer > 0. используется для DNS фильтра с включенным механизмом юридических пользователей (см. [LicenseInfo](#)) . Ограничение скорости днс запросов в секунду.
- **safe_search** - true/false включен безопасный поиск для юзера или нет
- **ad_block** - true/false включена блокировка банеров для юзера или нет
- **avira_scan** - true/false включен avira антивирус или нет
- **blocking_post** - true/false включена блокировка POST запросов или нет

LDAPUserInfo

В структуре описываются дополнительные параметры пользователя, если он взят из AD/LDAP-каталога.

key	type	description
guid	String	GUID пользователя LDAP/AD
dn	String	DN
upn	String	userPrincipalName

RadiusUserInfo

В структуре описываются дополнительные параметры пользователя, если он взят из Radius-сервера.

key	type	description
server_ip	String	адрес RADIUS-сервера
shared_key	String	shared secret

TimeInterval

Структура, задающая интервал времени.

key	type	description
start	Integer	Число в пределах 0-86399(сек.), описывающее начало интервала в пределах суток
end	Integer	Число в пределах 0-86399(сек.), описывающее начало интервала в пределах суток

TimeRule

Все поля не обязательные

monday	List< TimeInterval >	Набор временных интервалов в пределах суток
tuesday	List< TimeInterval >	Набор временных интервалов в пределах суток
wednesday	List< TimeInterval >	Набор временных интервалов в пределах суток
thursday	List< TimeInterval >	Набор временных интервалов в пределах суток
friday	List< TimeInterval >	Набор временных интервалов в пределах суток
saturday	List< TimeInterval >	Набор временных интервалов в пределах суток
sunday	List< TimeInterval >	Набор временных интервалов в пределах суток

TrafficRule

Структура с информацией о правиле фильтрации

key	type	description
id	UID	Уникальный идентификатор
guid	UID	GUID

owner_id	UID	ID пользователя-владельца, 0 если администратор. Не меняется после создания.
name	String	Имя правила
enabled	Boolean	true, если правило включено.
logic	String	Тип логики условий правила, or или and
conditions	List< RuleCondition >	Список условий правила
action	String	block/pass/intercept
attributes	List	Список дополнительных атрибутов {name, value}

RuleCondition

В каждом правиле может быть одно или несколько условий, при срабатывании которых выполняется действие **action**
Условия определяются своим типом.

CategoryCondition

Проверка url на принадлежность к категории, за исключением некоторых адресов

key	type	description
type	String	category
categories	List< UID >	список идентификаторов категорий на которые необходимо проверить адрес
exclusions	List<String>	список адресов-исключений

MorphologyCondition

Проверка контента на принадлежность к категории морфологического анализатора, за исключением некоторых категорий классификатора

key	type	description
type	String	morphology
lemmas	List< UID >	список идентификаторов категорий морфологического анализатора на которые необходимо проверить контент
categories	List< UID >	список идентификаторов категорий-исключений классификатора, на которые необходимо проверить адрес

TimeCondition

Проверка на попадание по времени

key	type	description
time	String	time
time	TimeRule	Временные интервалы

TimeCondition

Проверка контента регулярными выражениями

key	type	description
type	String	regexp
regexps	List<String>	список регулярных выражений

CustomDnsInfo

Структура с информацией о dns сервере

key	type	description
id	UID	Уникальный идентификатор
dns	String	IP-адрес

UrlCheckResult

Структура с информацией доступен ли URL

key	type	description
url	String	URL
allow	Boolean	true/false

IMHistoryRecord

Структура с информацией о перехваченном сообщении какого либо мессенджера

key	type	description
id	String	Идентификатор сообщения (md5 hash)
datetime	<dateTime.iso8601>	Время сообщения
protocol	String	Имя протокола
from	String	идентификатор пользователя (зависит от протокола)
to	String	идентификатор пользователя (зависит от протокола)
direction	String	направление сообщения in/out
message	String	тело сообщения

category	List< IMHistoryCategory >	Категория блокировки
action	String	Действие которое было предпринято none/spy/block
client_ip	String	IP-адрес сообщения
node	String	узел, на котором произошло событие

IMHistoryCategory

Категория для истории инстант мессенджеров

key	type	description
module	String	имя модуля lemmatizer/charset_detector/black_list/regexp_filter/conn
value	String	Идентификатор лемматизера черного списка либо unrecognize случае regexp_filter - имя регехр фильтра, который сработал; в случае connection - disconnected

IMHistoryFilter

Структура с информацией о настройках фильтра мгновенных сообщений

key	type	description
begin_date	<dateTime.iso8601>	Время начала
end_date	<dateTime.iso8601>	Время окончания
protocol	String	Имя протокола либо all
user	String	идентификатор пользователя (зависит от протокола)
user2	String	идентификатор второго пользователя (зависит от протокола), если нужен
search_string	String	строка для поиска в в поле message

IMExclusionRecord

Структура с информацией об исключении для IM

key	type	description
id	Integer	Идентификатор
uin	String	UIN или имя пользователя, в зависимости от типа IM
type	String	white/black тип исключения (блокировать или всегда пропус
protocol	String	Протокол icq/xmpp/vkontakte/odnoklassniki/facebook/msn/ymsg

IMExclusionFilter

Структура с фильтром для поиска в исключениях IM

key	type	description
uin	String	Подсторка, которая будет искаться в uin
type	String	white/black в каком списке искать
protocol	String	Искать либо во всех протоколах all , либо в конкретном icq/xmpp/vkontakte/odnoklassniki/facebook/msn/ymsg/

WebStatInfo

информация о сервер статистики

key	type	description
status	String	off/linked/cant_connect/not_registered/pass_error
email	String	email, на который привязана статистика
auth_token	String	Токен для аутентификации на сервере статистики, который присылается на емейл

RegexpInfo

информация о регулярном выражении

key	type	description
id	UID	идентификатор
enabled	Boolean	Включен ли регэксп или нет
name	String	Имя
pattern	String	шаблон регулярного выражения
compare_string	String	Опциональная строка для сравнения
ignore_case	Boolean	Игнорировать регистр или нет

Дополнительные поля для продукта Internet DLP:

level_id	Integer	идентификатор уровня, к которому привязано регулярное выражение
-----------------	---------	---

JournalEntryInfo

информация о записи в журнале изменений

key	type	description
id	UID	идентификатор

user	UID	идентификатор пользователя, создавшего транзакцию
group	UID	идентификатор группы пользователя
desc	List<String>	описание транзакции
timestamp	<dateTime.iso8601>	время последнего действия с транзакцией (открытие, обновление, откат, закрытие)
status	String	текущее состояние транзакции (открыта, закончена, подтверждена, сделан откат) opened/finished/rolled
access	String	уровень доступа к транзакции admin/group_admin/user/home_user
modifiers	List<String>	предыдущие состояния полей базы данных, которые были изменены в данной транзакции

ADImportInfo

Информация для импорта пользователей из Active Directory

key	type	description
ip_address	String	IP-адрес сервера AD
login_ad	String	Логин для доступа к AD
password	String	Пароль для доступа к AD
domain	String	Имя домена, с которого будет импорт пользователей
group_id	UID	идентификатор группы, в которую будут добавлены импортированные пользователи
users_type	String	тип пользователей wan/lan
wan_user	UID	Идентификатор WAN-пользователя, к которому будут привязаны импортируемые пользователи (только для users_type= lan)

StorageUploadSessionInfo

Структура содержит информацию о сессии загрузки файла в хранилище на сервере.

key	type	description
upload_url	String	относительный URL (то есть без схемы, домена и порта), на который клиент должен отправлять POST-запрос с файлом, URL инвалидируется после начала загрузки

check_token	String	токен, который используется для запроса статуса закички файла на сервер
storage_file_uid	UID <Integer>	идентификатор, который получит файл после загрузки в хранилище

StorageUploadSessionCheckInfo

Структура содержит информацию о сессии закички файлов в хранилище.

key	type	description
total_length	Integer	общая длина закичиваемого файла
uploaded_length	Integer	общая длина уже принятой сервером части

StorageFileInfo

Структура содержит информацию о загруженном файле в доменное хранилище.

key	type	description
content_type	String	content-type файла в том виде, в каком его передаёт клиент в POST-запросе закички
original_filename	String	имя файла в том виде, в каком его передаёт клиент в POST-запросе закички
size	Integer	размер файла в байтах
storage_file_uid	UID <String>	идентификатор, который получил файл после загрузки в хранилище

FilePreviewInfo

Структура, в которой содержится информации о превью файла из хранилища. В будущем в структуре появятся новые аргументы, чтобы можно было показывать картинки, например.

key	type	description
storage_file_uid	UID <String>	идентификатор файла в хранилище
html	String	собственно текст превью: какой-то извлечённый html из документа

DlpUserInfo

Структура содержит информацию о DLP-пользователе.

key	type	description
name	String	имя пользователя

DlpLevelInfo

Структура содержит информацию об одном уровне DLP.

key	type	description
id	UID	идентификатор уровня
name	String	название уровня

DlpGroupInfo

Структура содержит информацию о DLP-группе.

key	type	description
id	UID	Уникальный идентификатор
name	String	Имя группы

DlpPolicyType

Строковой тип, содержит константы для типов политик, возможные значения перечислены ниже.

value description

SKIP

BLOCK

ALERT

NamedListType

Тип списка это фиксированная строка

Тип	attributes	description
white	нет	Белый список
black	нет	Черный список
intercept	нет	Список сайтов для перехвата SSL
wblist	нет	Псевдотип для выборки белых и черных списков одновременно
morphology	weight: integer(), low, normal, high	Список морфологических терминов
mime	нет	список MIME-типов

httpsb	нет	список для блокировки https webfilter, при изменении сохраняется в файл etc/https_black_list.txt
cat_black	нет	Список для блокирования по категории
cat_white	нет	Белый список категории сайтов
cat_intercept	нет	Список категорий для перехвата SSL

вспомогательные типы для получения списков

Тип	description
all	Все списки
wblist	Черные и белые списки

NamedList

Полная информация об именованном списке.

key	type	description
id	UID	идентификатор именованного списка
owner_id	UID	идентификатор пользователя-владельца
enabled	Boolean	Включен список или нет
name	String	Имя списка
type	String	NamedListType
editable	Boolean	Редактируемый список или нет
url	String	Url для автоматического обновления списка
version	Integer	версия списка
last_update	Date	Дата последнего обновления
attributes	List	список дополнительных атрибутов вида {name, value}

для списков морфологии в атрибутах будет weight со значением порогового веса всей категории

Дополнительные поля для продукта Internet DLP:

level_id	Integer	идентификатор уровня, к которому привязан именованный список
-----------------	---------	--

SimpleNamedList

Сокращённая информация об именованном списке, для публичного использования.

key	type	description
-----	------	-------------

id	UID	идентификатор именованного списка
guid	UID	GUID именованного списка
name	String	Имя списка
type	String	NamedListType
attributes	List	список дополнительных атрибутов вида {name, value}

NListVal

key	type	description
id	UID	идентификатор (при добавлении не указывается)
value	String	значение
weight	Integer	вес слова поле будет только для морфологического списка

CertInfo

key	type	description
cert_country	String	Страна(например RU или EN)
cert_state	String	Область
cert_city	String	Город
cert_company	String	Компания
cert_url	String	интернет адрес
cert_email	String	емейл

SystemLogsInfo

key	type	description
enabled	Boolean	статус логов, включены или выключены
level	String	уровень логирования, допустимые значения: <i>all, info, error, warning</i>

Methods

Component: Core

В данном разделе описываются RPC-методы для работы с ядром: общие для всех других компонентов действия, например, получение/изменение персональных данных.

v1.core.login

[AuthInfo](#) v1.core.login(String **username**, String **password**)

Вход в систему с использованием логина и пароля.

Arguments

- **username** — имя пользователя, например, `user1@example.com`
- **password** — MD5 строка от пароля пользователя

Return value

Структура с токеном и правами доступа для последующей аутентификации. Полученный токен должен передаваться первым параметром при каждом защищённом запросе.

Remarks



Пароль должен вычисляться по формуле MD5(**password**)

Errors

faultCode	description
100	Неверно указан логин или пароль при попытке аутентификации

v1.core.logout

Boolean v1.core.logout(String **auth_token**)

Выход из системы, на стороне сервера уничтожается сессия, а токен инвалидируется.

Arguments

- **auth_token** — токен для аутентификации

Return value

true

Errors

faultCode	description
104	Invalid auth token

v1.core.login.as

[AuthInfo](#) v1.core.login.as(String **auth_token**, String **loginas**)

Вход в систему с использованием логина и пароля.

Arguments

- **auth_token** — Аутентификационный токен
- **loginas** — логин пользователя под которым хотим логиниться

Return value

Структура с токеном и правами доступа для последующей аутентификации. Полученный токен должен передаваться первым параметром при каждом защищённом запросе.

Errors

faultCode	description
100	Неверно указан логин или пароль при попытке аутентификации
102	Доступ запрещен если не admin и не group_admin
104	Invalid auth token

v1.core.password.recovery.start

```
Boolean v1.core.password.recovery.start(String lang)
```

Инициализация процедуры восстановления пароля. Метод не требует авторизации.

Arguments

- **lang** — код языка, который нужно далее передать в скрипт восстановления пароля

Return value

true

Remarks

В процессе выполнения этого метода продукт должен обратиться к директории с запросом на урл https://usergate.ru/bitrix/lic_table/recover_password.php и передать нужные данные, сам метод и его параметры описаны в документе <http://wiki.entensys.com/display/licsystem/New+activation+script+for+Entensys+software+products> В ответе придет строка-хеш, которую нужно где-то сохранить в базе, она используется для валидации.

Errors

faultCode	description
???	восстановление пароля не поддерживается (демо-версия, например)

v1.core.password.recovery.finish

```
Boolean v1.core.password.recovery.finish(String key, String
```

Завершение процедуры восстановления пароля. Метод не требует авторизации.

Arguments

- **key** — ключ, который пользователь получил в емейле
- **password** — новый пароль

Return value

true

Remarks

В процессе выполнения этого метода продукт должен проверить введённые данные, для этого он берёт значение, полученное ранее методом `v1.core.password.recovery.start (es)`, и делает проверку: `es == SHA1(SHA1(hwid) + key)`.

Errors

faultCode	description
???	некорректный код восстановления

v1.core.session.ping

```
String v1.core.session.ping(String auth_token)
```

Проверка токена аутентификации и alive-запрос.

Arguments

- **auth_token** — токен для аутентификации

Return value

Возвращается **auth_token**.

Errors

faultCode	description
104	Invalid auth token

v1.core.license.info

[LicenseInfo](#) v1.core.license.info(String **auth_token**)

Получить информацию о лицензии

Arguments

- **auth_token** — токен для аутентификации

Return value

Возвращается структура с информацией о лицензии.

Errors

faultCode	description
104	Invalid auth token

v1.core.license.activation

[LicenseInfo](#) v1.core.license.activation(String **auth_token**, **info**)

Зарегистрировать продукт

Arguments

- **auth_token** — токен для аутентификации
- **info** — активационная информация

Return value

Возвращается структура с информацией о лицензии.

Errors

faultCode	description
104	Invalid auth token
301	Ошибка записи в базу данных
302	Пин-код не найден
303	Пин-код неактивен
304	Превышение количества активаций
305	Модуль отсутствует в поставке

306	Исходный пин-код не найден
307	Новый пин-код не найден
308	Новый пин-код уже активирован
309	Новый пин-код неактивен
310	Ошибка генерации лицензионного ключа
311	Не отправилось письмо
312	Несоответствие версии
313	Нет данных о пользователе
314	Исходный пин-код уже был преобразован в новый
315	Нет ключей СТ для активации
316	Старый и новый пин-коды относятся к разным продуктам
317	Активация пин-кода для обновления без старого пин-кода
318	Некорректные переменные
319	Несоответствие количества событий
320	Пин-код не соответствует программе
321	"transport_error" "Reply was not recieved"
322	"wrong_user_data" "Wrong user data"
323	"invalid_user_data" "There is no enought data"
324	"wrong_action" "Wrong action for this type of the license"
325	"validation_failed" "Server's response validation failed"
326	"modules_missed" "There are no modules to process"
327	"wrong_type" "Wrong license type while creating module request"
351	not supported method
352	license/key is expired
353	limit of cluster nodes is reached
354	License is not available for this client
355	Hardware id not found

v1.core.license.remove

```
Boolean v1.core.license.remove(String auth_token, String node)
```

Отозвать лицензию для узла эрланга

Arguments

- **auth_token** — токен для аутентификации
- **node** — имя ноды

Return value

true/false

Errors

faultCode	description
104	Invalid auth token
300+	Ошибки лицензии

v1.core.register.master

```
Bool v1.core.register.master(String AdminLogin, String AdminPass,  
String SecretCookie)
```

Регистрация нового узла

Arguments

- **AdminLogin** — логин глобального админа (роль admin)
- **AdminPass** — пароль глобального админа (роль admin)
- **SecretCookie** — секретный ключ для связи нескольких эрланг НОДОВ

Return value

true

Remarks

После этой команды к мастеру могут подключаться дополнительные узлы, используя тот же **SecretCookie**

Errors

faultCode	description
2	Registration fault

v1.core.register.slave

```
Bool v1.core.register.slave(String MasterNodeName, String  
SecretCookie)
```

Регистрация нового узла в эрланг сети

Arguments

- **MasterNodeName** — имя узла в сети например node1@machine2
- **SecretCookie** — секретный ключ для связи нескольких эрланг НОДОВ

Return value

true

Errors

faultCode	description
106	Не удалось установить связь с MasterNodeName

v1.core.nodes.list

```
List<NodeStatus> v1.core.node.status(String auth_token)
```

Информация о текущем состоянии узлов сети эрланг.

Arguments

- **auth_token** — токен для аутентификации

Return value

Возвращает список структур [NodeStatus](#) о текущем состоянии узла эрланг сети.

Errors

faultCode	description
104	Invalid auth token

v1.core.get.categories

```
List<{id, name}> v1.core.get.categories()
```

Получить полный список категорий

Return value

Возвращает список структур вида {int id, String name}

Errors

faultCode	description
0	

v1.core.check.url

```
List<id> v1.core.check.url(String auth_token, String url)
```

Получить список категорий, к которым относится URL

Arguments

- **auth_token** — токен для аутентификации
- **url** — адрес для проверки

Return value

Возвращает список идентификаторов категорий

Errors

faultCode	description
354	Нет лицензии
3	Неорректный урл
2	Внутренняя ошибка модуля classify

v1.core.public.check.url

```
List<id> v1.core.public.check.url(String login, String md5_pass,  
url)
```

Получить список категорий, к которым относится URL

Arguments

- **login** — логин пользователя
- **md5_pass** — хеш md5 от пароля пользователя
- **url** — адрес для проверки

Return value

Возвращает список идентификаторов категорий

Errors

faultCode	description
354	Нет лицензии
3	Неорректный урл
2	Внутренняя ошибка модуля classify
100	Неверный логин или пароль

v1.core.temp.unblock

```
Boolean v1.core.temp.unblock(String auth_token, Boolean set, Int  
time)
```

Временно разблокировать работу киндергейта

Arguments

- **auth_token** — токен для аутентификации
- **set** — true - разблокировать, false - вернуть блокировку
- **time** — время разблокировки в секундах

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.core.temp.unblock.get

```
Boolean v1.core.temp.unblock.get(String auth_token)
```

Получить статус временной разблокировки

Arguments

- **auth_token** — токен для аутентификации

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.core.user.passwd

```
Boolean v1.core.user.passwd(String auth_token, String oldpass,  
newpass)
```

Поменять пароль текущего пользователя

Arguments

- **auth_token** — токен для аутентификации
- **oldpass** — md5 старого пароля
- **newpass** — md5 нового пароля

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
100	Неверный старый пароль

v1.core.nodes.list

```
List<NodeDesc> v1.core.nodes.list(String auth_token)
```

Получить список узлов сети эрланг

Arguments

- **auth_token** — токен для аутентификации

Return value

Список структур [NodeDesc](#)

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.core.node.remove

```
Boolean v1.core.node.remove(String auth_token, String
```

Удаление узла из кластера

Arguments

- **auth_token** — токен для аутентификации
- **node_name** — Имя узла

Return value

true

Errors

faultCode	description
------------------	--------------------

- 102** Доступ запрещен
- 104** Неверный аутентификационный токен

v1.core.login.history.list

```
PagingList v1.core.login.history.list(String auth_token, Integer start, Integer limit)
```

Получить страницу истории входов в систему

Arguments

- **auth_token** — токен для аутентификации
- **start** — стартовый индекс
- **limit** — максимальное число возвращаемых записей

Return value

[PagingList](#) где items содержит структуры [LoginHistoryRecord](#)

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.core.login.history.clear

```
Boolean v1.core.login.history.clear(String auth_token)
```

Очистить журнал входов в систему

Arguments

- **auth_token** — токен для аутентификации

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.core.public.check.urls

```
List<UrlCheckResult>v1.core.public.check.urls(String login, String password, List<String>urls)
```

Проверить доступность списка URL для пользователя

Arguments

- **login** — логин пользователя
- **password** — пароль пользователя
- **urls** — список адресов для проверки

Return value

Список структур [UrlCheckResult](#), в котором адреса расположены в том же порядке что и в запросе

Errors

faultCode	description
100	Неверно указан логин или пароль при попытке аутентификации

v1.core.codepages.list

```
List<String>v1.core.codepages.list(String auth_token)
```

Возвращает список возможных кодовых страниц на сервере (**iconv -l**)

Arguments

- **auth_token** — токен для аутентификации

Return value

Список строк с кодовыми страницами

Errors

faultCode	description
104	Неверный аутентификационный токен

Component: Settings

Работа с настройками и распределенным конфигом



Необходимо помнить

Любое изменение базы данных теперь протоколируется. Для этого используется журнал транзакций, которые в последствии можно

просмотреть и понять что же произошло с настройками. При этом можно откатить (rollback) сделанные изменения (при наличии достаточных прав у данного пользователя), либо подтвердить транзакцию (удалить - изменения обратно можно будет сделать только "руками"; также, при наличии прав у пользователя). Исходя из этого, необходимо сначала открывать транзакцию и только после этого изменять базу данных (добавлять пользователей, например). После завершения изменений, необходимо закрыть транзакцию как описано в соответствующем разделе.

v1.settings.get.string

```
String v1.settings.get.string(String auth_token, String name)
```

Получить параметр из конфигурации по ключу

Arguments

- **auth_token** — токен для аутентификации
- **name** — имя параметра

Return value

Возвращает значение параметра

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Параметр не найден

v1.settings.get.public.string

```
String v1.settings.get.public.string(String name)
```

Получить параметр из конфигурации по ключу

Arguments

- **name** — имя параметра

Return value

Возвращает значение параметра

Errors

faultCode	description
105	Параметр не найден

v1.settings.set.string

```
Boolean v1.settings.set.string(String auth_token, String name, String value)
```

Установить новое значение для параметра

Arguments

- **auth_token** — токен для аутентификации
- **name** — имя параметра
- **value** — новое значение

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.settings.get.bool

```
Boolean v1.settings.get.bool(String auth_token, String name)
```

Получить булевый параметр из конфигурации по ключу

Arguments

- **auth_token** — токен для аутентификации
- **name** — имя параметра

Return value

Возвращает значение параметра

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Параметр не найден

v1.settings.get.public.bool

```
Boolean v1.settings.get.public.bool(String name)
```

Получить параметр из конфигурации по ключу

Arguments

- **name** — имя параметра

Return value

Возвращает значение параметра

Errors

faultCode	description
105	Параметр не найден

v1.settings.set.bool

```
Boolean v1.settings.set.bool(String auth_token, String name,  
value)
```

Установить новое значение для параметра

Arguments

- **auth_token** — токен для аутентификации
- **name** — имя параметра
- **value** — новое значение

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.settings.set.filter.categories.level

```
Boolean v1.settings.set.filter.categories.level(String auth_token,  
level)
```


Установить уровень фильтрации по категориям

Arguments

- **auth_token** — токен для аутентификации
- **level** — уровень фильтрации

Return value

true

Remarks

Возможные уровни

- **0** — фильтрация выключена
- **1** — [33, 31, 42, 61, 1]
- **2** — [33, 31, 42, 61, 1, 40]
- **3** — [33, 31, 42, 61, 1, 40, 23, 76, 65]
- **4** — [33, 31, 42, 61, 1, 40, 23, 76, 65, 18, 13, 26, 73, 77, 3, 29, 42, 2]
- **5** — режим ручного выбора категорий

Errors

faultCode	description
102	Доступ запрещен
103	Неверный параметр level
104	Неверный аутентификационный токен

v1.settings.get.filter.categories.level

```
Integer v1.settings.get.filter.categories.level(String auth_token)
```

Получить уровень фильтрации по категориям

Arguments

- **auth_token** — токен для аутентификации

Return value

Уровень фильтрации

Remarks

Возможные уровни

- **0** — фильтрация выключена
- **1** — [33, 31, 42, 61, 1]
- **2** — [33, 31, 42, 61, 1, 40]

- **3** — [33, 31, 42, 61, 1, 40, 23, 76, 65]
- **4** — [33, 31, 42, 61, 1, 40, 23, 76, 65, 18, 13, 26, 73, 77, 3, 29, 42, 2]
- **5** — режим ручного выбора категорий

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.settings.get.filter.categories

```
List v1.settings.get.filter.categories(String auth_token)
```

Получить список идентификаторов, фильтруемых категорий.

Arguments

- **auth_token** — токен для аутентификации

Return value

Список идентификаторов

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.settings.get.filter.categories

```
Boolean v1.settings.get.filter.categories(String auth_token, List<int> uids)
```

Получить список идентификаторов, фильтруемых категорий.

Arguments

- **auth_token** — токен для аутентификации
- **uids** — список идентификаторов категорий

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.settings.suggest.categories

```
String v1.settings.suggest.categories(String auth_token, String url,
List<Integer> cats)
```

Предложить категорию для сайта.

Arguments

- **auth_token** — токен для аутентификации
- **url** — сайт, для которого предлагается категория
- **cats** — список категорий для сайта

Return value

строка с xml ответом от коммтача

Remarks

код ошибки commtouch

100 — COMMTOUCH_XML_STATUS_NOT_UTF_8'
101 — COMMTOUCH_XML_STATUS_INVALID_URL'
102 — COMMTOUCH_XML_STATUS_INVALID_LICENSE'
103 — COMMTOUCH_XML_STATUS_INVALID_VERSION'
104 — COMMTOUCH_XML_STATUS_INVALID_EXTRA_CONTENT'
105 — COMMTOUCH_XML_STATUS_CORRUPTED_DATA'
999 — COMMTOUCH_XML_STATUS_GENERAL_SCHEMA_ERROR'

Errors

faultCode	description
102	Доступ запрещен
104	Неверный идентификационный токен
109	Ошибка связи с сервером Commtouch

v1.settings.white.list.requests.add

```
Boolean v1.settings.white.list.requests.add(String link)
```

Добавить адрес к таблице запросов на добавление в белый список

Arguments

- **auth_token** — токен для аутентификации

- **link** — хостнейм

Return value

true

Errors

faultCode	description
103	Хостнейм не может быть пустым
107	Элемент уже присутствует в запросах на добавление

v1.settings.white.list.requests.delete

```
Boolean v1.settings.white.list.requests.delete(String auth_token,  
link)
```

Удалить элемент из списка запросов на добавление.

Arguments

- **auth_token** — токен для аутентификации
- **link** — хостнейм

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Элемент не найден

v1.settings.white.list.requests.list

```
List<WLRequest> v1.settings.white.list.requests.list(String
```

Получить список запросов на добавление в белый список

Arguments

- **auth_token** — токен для аутентификации

Return value

Список структур [WLRequest](#)

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.settings.adblock.info

```
AdBlockInfo v1.settings.adblock.info(String auth_token)
```

Получить информацию о списках adblock

Arguments

- **auth_token** — токен для аутентификации

Return value

Структура [AdBlockInfo](#)

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.settings.adblock.update

```
Boolean v1.settings.adblock.update(String auth_token)
```

Обновить списки adblock с сервера

Arguments

- **auth_token** — токен для аутентификации

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.settings.custom.dnses.list

```
List v1.settings.custom.dnses.list(String auth_token)
```

Получить список пользовательских dns серверов

Arguments

- **auth_token** — токен для аутентификации

Return value

Список структур вида {id, Dns}

Errors

faultCode	description
104	Неверный аутентификационный токен

v1.settings.custom.dns.add

```
UID v1.settings.custom.dns.add(String auth_token, CustomDnsInfo dns)
```

Добавить новый dns-сервер в список

Arguments

- **auth_token** — токен для аутентификации
- **dns** — новый dns-сервер

Return value

[UID](#) нового элемента в списке

Errors

faultCode	description
104	Неверный аутентификационный токен

v1.settings.custom.dns.update

```
UID v1.settings.custom.dns.update(String auth_token, UID id, CustomDnsInfo dns)
```

Редактирование dns-сервера в списке пользовательских dns-серверов

Arguments

- **auth_token** — токен для аутентификации

- **id** — идентификатор редактируемого элемента в списке
- **dns** — новый dns-сервер

Return value

true

Errors

faultCode	description
104	Неверный аутентификационный токен
105	Элемент списка не найден

v1.settings.custom.dns.delete

```
UID v1.settings.custom.dns.delete(String auth_token, UID id)
```

Удаление dns-сервера из списка пользовательских dns-серверов

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор удаляемого элемента в списке

Return value

true

Errors

faultCode	description
104	Неверный аутентификационный токен
105	Элемент списка не найден

v1.settings.generate.certificate

```
Boolean v1.settings.generate.certificate(String auth_token, CertInfo info)
```

Генерация SSL сертификатов

Arguments

- **auth_token** — токен для аутентификации
- **info** — Информация для генерации сертификата

Return value

true

Errors

faultCode	description
104	Неверный аутентификационный токен

v1.settings.systemlogs.status

```
SystemLogsInfo v1.settings.systemlogs.status(String auth_token)
```

Получить параметры логирования

Arguments

- **auth_token** — токен для аутентификации

Return value

Возвращает параметры логов в виде структуры [SystemLogsInfo](#)

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.settings.systemlogs.set

```
Boolean v1.settings.systemlogs.set(String auth_token,  
logs)
```

Изменить параметры логирования

Arguments

- **auth_token** — токен для аутентификации
- **logs** — новые параметры логов

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

Component: Accounts

В разделе описываются методы для работы с группами и пользователями.



Важно

Всегда есть суперпользователь с ролью **admin**, он не принадлежит никакой группе (`GroupId = 0`). Остальные пользователи могут быть добавлены только в существующую группу. Пользователи с ролью **group_admin** могут администрировать всех пользователей текущей группы и всех ее подгрупп.

Пользователи с ролью **user** и **home_user** не имеют возможности влиять на структуру групп и пользователей. Они видят только свои настройки.

v2.accounts.groups.list

```
List<GroupInfo> v2.accounts.groups.list(String auth_token)
```

Получить список всех групп, доступных для текущего пользователя

Arguments

- **auth_token** — токен для аутентификации

Return value

Возвращается список структур типа [GroupInfo](#)

Errors

faultCode	description
104	Invalid auth token
102	Access Denied

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	Список всех подгрупп текущей группы
admin	Метод должен возвращать весь список групп

v2.accounts.group.fetch

```
GroupInfo v2.accounts.group.fetch(String auth_token, UID id)
```

Получить информацию о группе

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор группы

Return value

Структура [GroupInfo](#)

Errors

faultCode	description
104	Invalid auth token
102	Access Denied

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	Если есть доступ по иерархии групп
admin	Полный доступ

v2.accounts.group.parents.list

```
List<GroupInfo> v2.accounts.group.parents.list(String auth_token,  
group_id)
```

Получить список родительских групп для группы **group_id**, доступных для текущего пользователя. Этот список групп включает в себя все группы которые могут быть назначены родительской для запрашиваемой группы.

Arguments

- **auth_token** — токен для аутентификации
- **group_id** — идентификатор группы, для которой нужно получить родительские группы

Return value

Возвращается список структур типа [GroupInfo](#)

Errors

faultCode	description
104	Invalid auth token

102 Access Denied

v2.accounts.group.users.list

```
List<UserInfo> v2.accounts.group.users.list(String auth_token, UID group_id)
```

Получить список пользователей для группы.

Arguments

- **auth_token** — токен для аутентификации
- **group_id** — идентификатор группы, для которой нужно получить пользователей

Return value

Возвращается список структур типа [UserInfo](#)

Errors

faultCode	description
104	Invalid auth token
102	Access Denied

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	Список всех пользователей группы
admin	Список всех пользователей группы

v2.accounts.group.add

```
UID v2.accounts.group.add(String auth_token, GroupInfo group)
```

Добавление новой группы пользователей.

Arguments

- **auth_token** — токен для аутентификации
- **group** — данные о группе

Return value

Идентификатор только что добавленной группы.

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	Parent group not found

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	Доступ запрещен, если родительская группа не входит в список доступных
admin	Полный доступ

v2.accounts.group.update

```
Boolean v2.accounts.group.update(String auth_token, UID group\_id, GroupInfo group)
```

Обновление информации о группе.

Arguments

- **auth_token** — токен для аутентификации
- **group_id** — идентификатор группы
- **group** — структура типа [GroupInfo](#) с параметрами, которые нужно поменять. Изменяются только те параметры, которые в этой структуре передаются. Поле id игнорируется в любом случае.

Return value

true

Errors

faultCode	description
102	Access Denied
103	Recursive group detect
104	Invalid auth token
105	Group not found

Access

role	description
user	Доступ запрещен

home_user	Доступ запрещен
group_admin	Доступ запрещен, если группа не входит в список доступных
admin	Полный доступ

v2.accounts.group.delete

```
Boolean v2.accounts.group.delete(String auth_token, UID group_id)
```

Удаление группы, пользователей с ней связанных и всех правил созданных пользователями.

Arguments

- **auth_token** — токен для аутентификации
- **group_id** — идентификатор группы

Return value

true

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	Group not found

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	Доступ запрещен, если группа не входит в список доступных
admin	Полный доступ

v2.accounts.group.assign.rules

```
Boolean v2.accounts.group.assign.rules(String auth_token, UID group_id, List<UID> rids)
```

Назначение правил группе

Arguments

- **auth_token** — токен для аутентификации

- **group_id** — идентификатор пользователя
- **rids** — массив идентификаторов правил, назначаемых группе

Return value

true

Remarks

Чтобы убрать связь группы и правила, нужно идентификатор правила указать **со знаком минус**.

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	Group not found

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	доступ
admin	Полный доступ

v2.accounts.group.assign.nlists

```
Boolean v2.accounts.group.assign.nlists(String auth_token, UID group_id, List<UID> lids)
```

Назначение именованных списков группе

Arguments

- **auth_token** — токен для аутентификации
- **group_id** — идентификатор пользователя
- **lids** — массив идентификаторов именованных списков, назначаемых группе

Return value

true

Remarks

Чтобы убрать связь группы и списка, нужно идентификатор списка указать **со знаком минус**.

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	Group not found

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	доступ
admin	Полный доступ

v2.accounts.users.list

```
PagingList v2.accounts.users.list(String auth_token, Integer start, Integer limit, String filter)
```

Получить список всех пользователей, доступных для текущего пользователя

Arguments

- **auth_token** — токен для аутентификации
- **start** — стартовый индекс страницы
- **limit** — лимит записей на страницу
- **filter** — фильтр, поиск подстроки в имени, логине, емейле и ip-адресе

Return value

Возвращается [PagingList](#) где **items** список структур типа [UserInfo](#)

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	User not found

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен

group_admin Полный доступ
admin Полный доступ

v2.accounts.user.fetch

```
UserInfo v2.accounts.user.fetch(String auth_token, UID user_id)
```

Получить параметры одного пользователя

Arguments

- **auth_token** — токен для аутентификации
- **user_id** — идентификатор пользователя

Return value

Возвращается структура типа [UserInfo](#).

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	User not found

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	Полный доступ
admin	Полный доступ

v2.accounts.user.add

```
UID v2.accounts.user.add(String auth_token, UserInfo user)
```

Добавление нового пользователя в группу. Группа определяется полем **group_id** в структуре [UserInfo](#).

Arguments

- **auth_token** — токен для аутентификации
- **user** — данные о пользователе

Return value

Идентификатор только что добавленного пользователя.

Errors

faultCode	description
102	Access Denied
103	Full user info required
104	Invalid auth token

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	Полный доступ
admin	Полный доступ

v2.accounts.user.update

```
Boolean v2.accounts.user.update(String auth_token, UID user_id,  
UserInfo user)
```

Обновление информации о пользователе.

Arguments

- **auth_token** — токен для аутентификации
- **user_id** — идентификатор пользователя
- **user** — структура типа [UserInfo](#) с параметрами, которые нужно поменять. Изменяются только те параметры, которые в этой структуре передаются.

Return value

true

Remarks

метод не дает возможности менять **id**. Данные поля в структуре будут просто проигнорированы.

Errors

faultCode	description
3	Wrong method parameters
102	Access Denied
103	Conflict IPRange
104	Invalid auth token

105 User not found
108 Limit users reached

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	Доступ к редактированию только пользователей своей группы и ее подгрупп
admin	Полный доступ

v2.accounts.user.delete

```
Boolean v2.accounts.user.delete(String auth_token, UID user_id)
```

Удаление пользователя

Arguments

- **auth_token** — токен для аутентификации
- **user_id** — идентификатор пользователя

Return value

true

Errors

faultCode	description
102	Access Denied
103	Can't delete yourself
104	Invalid auth token
105	User not found

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	Доступ к удалению только пользователей своей группы и ее подгрупп
admin	Полный доступ

v2.accounts.user.assign.rules

```
Boolean v2.accounts.user.assign.rules(String auth_token, UID  
List<UID> rids)
```

Назначение правил пользователю

Arguments

- **auth_token** — токен для аутентификации
- **user_id** — идентификатор пользователя
- **rids** — массив идентификаторов правил, назначаемых пользователю

Return value

true

Remarks

Чтобы убрать связь пользователя и правила, нужно идентификатор правила указать **со знаком минус**.

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	User not found

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	доступ
admin	Полный доступ

v2.accounts.user.assign.nlists

```
Boolean v2.accounts.user.assign.nlists(String auth_token, UID  
List<UID> lids)
```

Назначение именованных списков пользователю

Arguments

- **auth_token** — токен для аутентификации
- **user_id** — идентификатор пользователя

- **lids** — массив идентификаторов списков, назначаемых пользователю

Return value

true

Remarks

Чтобы убрать связь пользователя и именованного списка, нужно идентификатор списка указать **со знаком минус**.

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	User not found

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	доступ
admin	Полный доступ

v2.accounts.ldap.import

```
Boolean v2.accounts.ldap.import(String auth_token, ADImportInfo)
```

Назначение правил пользователю

Arguments

- **auth_token** — токен для аутентификации
- **info** — информация для импорта

Return value

true

Errors

faultCode	description
100	Invalid login or password
102	Access Denied
104	Invalid auth token

109 Ошибка связи с AD

Access

role	description
user	Доступ запрещен
home_user	Доступ запрещен
group_admin	доступ
admin	Полный доступ

Component: Named Lists

Раздел содержит функции для работы с именованными списками

Типы именованных списков

[NamedListType](#)

v1.nlists.public.list

```
List<SimpleNamedList> v1.nlists.public.list(NamedListType type)
```

Получить список именованных списков определенного типа

Arguments

- **type** — тип именованного списка

Return value

СПИСОК [SimpleNamedList](#)

Errors

faultCode description

v1.nlists.owned.list

```
List<NamedList> v1.nlists.owned.list(String auth_token,  
type)
```

Получить список именованных списков для владельца

Arguments

- **auth_token** — токен для аутентификации
- **type** — тип требуемых записей [NamedListType](#)

Return value

СПИСОК [NamedList](#)

Errors

faultCode	description
104	Неверный аутентификационный токен

v1.nlists.list

```
PagingList v1.nlists.list(String auth_token, NamedListType type, start, Integer limit, String filter)
```

Получить страницу из списка именованных списков

Arguments

- **auth_token** — токен для аутентификации
- **type** — тип требуемых записей [NamedListType](#)
- **start** — стартовый индекс страницы
- **limit** — лимит записей на страницу
- **filter** — фильтр по имени списка

Return value

[PagingList](#) где items это структуры [NamedList](#)

Errors

faultCode	description
104	Неверный аутентификационный токен

v1.nlists.add

```
Integer v1.nlists.add(String auth_token, NamedList val)
```

Добавить новый именованный список

Arguments

- **auth_token** — токен для аутентификации
- **val** — структура с информацией о списке

Return value

Идентификатор нового элемента в списке

Errors

faultCode	description
104	Неверный аутентификационный токен

v1.nlists.update

```
Boolean v1.nlists.update(String auth_token, UID id, NamedList val)
```

Обновить именованный список

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор обновляемого именованного списка
- **val** — структура с информацией о списке

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Неизвестный список

v1.nlists.delete

```
Integer v1.nlists.delete(String auth_token, UID id)
```

Удалить именованный список

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор удаляемого именованного списка

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Неизвестный список

v1.nlists.list.list

```
PagingList v1.nlists.list.list(String auth_token, Integer lid, Integer Integer limit, String filter)
```

Получить страницу из именованного списка

Arguments

- **auth_token** — токен для аутентификации
- **lid** — идентификатор именованного списка
- **start** — стартовый индекс страницы
- **limit** — лимит записей на страницу
- **filter** — фильтр

Return value

[PagingList](#) где items это структуры [NListVal](#)

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Неизвестный список

v1.nlists.list.add

```
UID v1.nlists.list.add(String auth_token, Integer lid, NListVal value)
```

Добавить значение в именованный список

Arguments

- **auth_token** — токен для аутентификации
- **lid** — идентификатор именованного списка
- **value** — значение

Return value

Идентификатор нового значения

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Неизвестный список

v1.nlists.list.update

```
Boolean v1.nlists.list.update(String auth_token, Integer lid, Integer NListVal value)
```

Добавить значение в именованный список

Arguments

- **auth_token** — токен для аутентификации
- **lid** — идентификатор именованного списка
- **vid** — идентификатор изменяемого элемента в списке
- **value** — новое значение

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Неизвестный список

v1.nlists.list.delete

```
Boolean v1.nlists.list.delete(String auth_token, Integer lid, Integer
```

Удалить значение из именованного списка

Arguments

- **auth_token** — токен для аутентификации
- **lid** — идентификатор именованного списка
- **vid** — идентификатор удаляемого элемента в списке

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Неизвестный список

v1.nlists.list.import

```
Boolean v1.nlists.list.import(String auth_token, Integer lid, String filename)
```

Импортировать список из файла

Arguments

- **auth_token** — токен для аутентификации
- **lid** — идентификатор именованного списка
- **filename** — имя текстового файла, предварительно загруженного в storage, со списком значений в utf8

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Неизвестный список

v1.nlists.list.export

```
String v1.nlists.list.export(String auth_token, Integer lid)
```

Экспортировать список

Arguments

- **auth_token** — токен для аутентификации
- **lid** — идентификатор именованного списка

Return value

URL по которому можно скачать итоговый файл

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен
105	Неизвестный список
110	Список пустой

Component: Rules

Раздел содержит функции для работы с правилами фильтрации.



Важно

Каждое правило принадлежит пользователю его создавшему. Владелец правила не может быть изменен. Глобальный администратор(**admin**) или администратор группы(**group_admin**) не могут назначать группам или пользователям правила, владельцем которых являются подчиненные пользователи, но они могут их редактировать и удалять.



Необходимо помнить

Любое изменение базы данных теперь протоколируется. Для этого используется журнал транзакций, которые в последствии можно просмотреть и понять что же произошло с настройками. При этом можно откатить (rollback) сделанные изменения (при наличии достаточных прав у данного пользователя), либо подтвердить транзакцию (удалить - изменения обратно можно будет сделать только "руками"; также, при наличии прав у пользователя). Исходя из этого, необходимо сначала открывать транзакцию и только после этого изменять базу данных (добавлять пользователей, например). После завершения изменений, необходимо закрыть транзакцию как описано в соответствующем разделе.

v2.rules.list

```
List<TrafficRule> v2.rules.list(String auth_token)
```

Получить список правил

Arguments

- **auth_token** — токен для аутентификации.

Return value

Список структур [TrafficRule](#).

Errors

faultCode	description
104	Invalid auth token

Access

user	Возвращаются правила только текущего юзера
home_user	Возвращаются правила только текущего юзера

group_admin Возвращаются правила только текущей группы и правила юзеров, входящих в группу
admin Возвращается полный список правил

v2.rules.owned.list

```
List<TrafficRule> v2.rules.owned.list(String auth_token)
```

Получить список правил, владельцем которых является текущий пользователь

Arguments

- **auth_token** — токен для аутентификации.

Return value

Список структур [TrafficRule](#).

Errors

faultCode	description
104	Invalid auth token

Access

user	Возвращаются правила только текущего юзера
home_user	Возвращаются правила только текущего юзера
group_admin	Возвращаются правила только текущего юзера
admin	Возвращаются правила только текущего юзера

v2.rules.add

```
UID v2.rules.add(String auth_token, TrafficRule rule)
```

Создать новое правило для пользователя

Arguments

- **auth_token** — токен для аутентификации.
- **rule** — добавляемое правило

Return value

Идентификатор нового правила

Errors

faultCode	description
103	Не полная информация о правиле
104	Invalid auth token

Access

user	Полный доступ
home_user	Полный доступ
group_admin	Полный доступ
admin	Полный доступ

v2.rules.delete

```
Boolean v2.rules.delete(String auth_token, UID id)
```

Удалить правило

Arguments

- **auth_token** — токен для аутентификации.
- **id** — идентификатор удаляемого правила

Return value

true

Errors

faultCode	description
102	Access denied
104	Invalid auth token
105	Правило не найдено

v2.rules.update

```
Boolean v2.rules.update(String auth_token, UID id, TrafficRule rule)
```

Обновить информацию в правиле

Arguments

- **auth_token** — токен для аутентификации.
- **id** — идентификатор обновляемого правила
- **rule** — новое правило

Return value

true

Errors

faultCode	description
102	Access denied
104	Invalid auth token
105	Правило не найдено

v2.rules.public.get.name

```
String v2.rules.public.get.name(UID id)
```

Вернуть имя правила

Arguments

- **id** — идентификатор правила

Return value

имя правила

Errors

faultCode	description
105	Правило не найдено

Component: Instant Messengers

Раздел содержит методы для работы с перехваченными сообщениями различных мессенджеров.



Необходимо помнить

Любое изменение базы данных теперь протоколируется. Для этого используется журнал транзакций, которые в последствии можно просмотреть и понять что же произошло с настройками. При этом можно откатить (rollback) сделанные изменения (при наличии достаточных прав у данного пользователя), либо подтвердить транзакцию (удалить - изменения обратно можно будет сделать только "руками"; также, при наличии прав у пользователя). Исходя из этого, необходимо сначала открывать транзакцию и только после этого изменять базу данных (добавлять пользователей, например). После завершения изменений, необходимо закрыть транзакцию как описано в соответствующем разделе.

v1.im.message.history.list

```
PagingList v1.im.message.history.list(String auth_token,  
MessageHistoryFilter filter, Integer start, Integer limit)
```

Получить страницу лога сообщений мессенджеров.

Arguments

- **auth_token** — токен для аутентификации
- **filter** — фильтр
- **start** — стартовый индекс страницы
- **limit** — лимит записей на страницу

Return value

[PagingList](#) где items это структуры [MessageHistoryRecord](#)

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.im.message.history.clear

```
Boolean v1.im.message.history.clear(String auth_token, UID JEID)
```

Очистить историю сообщений

Arguments

- **auth_token** — токен для аутентификации
- **JEID** — идентификатор транзакции

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.im.message.delete

```
Boolean v1.im.message.delete(String auth_token, UID JEID, String
```

удалить 1 сообщение.

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор удаляемого сообщения
- **JEID** — идентификатор транзакции

Return value

true

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.im.exclusions.list

```
PagingList v1.im.exclusions.list(String auth_token, IMExclusionFilter filter, Integer start, Integer limit)
```

Получить список исключений для IM

Arguments

- **auth_token** — токен для аутентификации
- **filter** — фильтр
- **start** — стартовый индекс страницы
- **limit** — лимит записей на страницу

Return value

[PagingList](#) где items это структуры [IMExclusionRecord](#)

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.im.exclusion.add

```
UID v1.im.exclusion.add(String auth_token, UID JEID, IMExclusionRecord record)
```

Добавить новый элемент в список исключений

Arguments

- **auth_token** — токен для аутентификации
- **record** — запись для вставки в список
- **JEID** — идентификатор транзакции

Return value

[UID](#) нового элемента в списке

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.im.exclusion.update

```
Boolean v1.im.exclusion.update(String auth_token, UID JEID, UID IMExclusionRecord record)
```

Обновить элемент в списке исключений

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор элемента в списке
- **record** — запись для вставки в список
- **JEID** — идентификатор транзакции

Return value

true

Errors

faultCode	description
102	Доступ запрещен
103	Элемент не найден
104	Неверный аутентификационный токен

v1.im.exclusion.delete

```
Boolean v1.im.exclusion.delete(String auth_token, UID JEID, UID id)
```

Удалить элемент из списка исключений

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор элемента в списке
- **JEID** — идентификатор транзакции

Return value

true

Errors

faultCode	description
102	Доступ запрещен
103	Элемент не найден
104	Неверный аутентификационный токен

Component: Webstat

В разделе описаны методы для работы с Web-статистикой.

[WEB Statistics RPC Interface](#)

v1.webstat.info

```
WebStatInfo v1.webstat.info(String auth_token)
```

Получить информацию о связи с вебстатистикой

Arguments

- **auth_token** — токен для аутентификации

Return value

Возвращается структура [WebStatInfo](#)

Errors

faultCode	description
104	Invalid auth token
102	Access Denied

v1.webstat.register

```
String v1.webstat.register(String auth_token, String locale)
```

Зарегистрировать продукт на сервере веб-статистики

Arguments

- **auth_token** — токен для аутентификации
- **locale** — локаль(**en, ru...**), используется для письма, которое придет на зарегистрированный емейл при активации

Return value

Email-адрес, куда будет выслан ключ для сервера статистики

Errors

faultCode	description
2	Ошибка вызова метода на удаленном сервере
103	Неверные входные данные
104	Invalid auth token
109	Ошибка коммуникации с удаленным сервером статистики
354	Нет лицензии

v1.webstat.counters

```
Struct v1.webstat.counters(String auth_token)
```

Зарегистрировать продукт на сервере веб-статистики

Arguments

- **auth_token** — токен для аутентификации

Return value

Структура со счетчиками

field	description
requests	Число полученных DNS-реквестов
responded	число отвеченных DNS-запросов
drop	число отброшенных DNS-запросов
cache_hit	число попаданий в кеш DNS

Errors

faultCode	description
2	Ошибка вызова метода на удаленном сервере
103	Неверные входные данные
104	Invalid auth token

Component: Regular Expressions

Работа с регулярными выражениями



Необходимо помнить

Любое изменение базы данных теперь протоколируется. Для этого используется журнал транзакций, которые в последствии можно просмотреть и понять что же произошло с настройками. При этом можно откатить (rollback) сделанные изменения (при наличии достаточных прав у данного пользователя), либо подтвердить транзакцию (удалить - изменения обратно можно будет сделать только "руками"; также, при наличии прав у пользователя). Исходя из этого, необходимо сначала открывать транзакцию и только после этого изменять базу данных (добавлять пользователей, например). После завершения изменений, необходимо закрыть транзакцию как описано в соответствующем разделе.

v1.regexps.list

```
PagingList v1.regexps.list(String auth_token, Integer start, Integer limit, String filter)
```

Получить список регулярных выражений

Arguments

- **auth_token** — токен для аутентификации
- **start** — стартовый индекс страницы
- **limit** — лимит записей на страницу
- **filter** — фильтр, поиск подстроки в имени

Return value

[PagingList](#), где items список структур [RegexpInfo](#)

Errors

faultCode	description
102	Доступ запрещен
104	Неверный аутентификационный токен

v1.regexps.add

```
UID v1.regexps.add(String auth_token, UID JEID, RegexpInfo info)
```

Добавление нового регэкспа

Arguments

- **auth_token** — токен для аутентификации
- **info** — данные о регэкспе

- **JEID** — идентификатор транзакции

Return value

Идентификатор только что добавленного регулярного выражения.

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.regexps.fetch

```
Boolean v1.regexps.fetch(String auth_token, UID id)
```

Обновление информации о регулярном выражении.

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор регулярного выражения

Return value

[RegexpInfo](#)

Errors

faultCode	description
3	Wrong method parameters
102	Access Denied
104	Invalid auth token
105	Regexp not found

v1.regexps.update

```
Boolean v1.regexps.update(String auth_token, UID JEID, UID id,  
RegexpInfo info)
```

Обновление информации о регулярном выражении.

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор регулярного выражения

- **info** — структура типа [RegexpInfo](#) с параметрами, которые нужно поменять. Изменяются только те параметры, которые в этой структуре передаются.
- **JEID** — идентификатор транзакции

Return value

true

Remarks

метод не дает возможности менять **id**. Данные поля в структуре будут просто проигнорированы.

Errors

faultCode	description
3	Wrong method parameters
102	Access Denied
104	Invalid auth token
105	Regexp not found

v1.regexps.delete

```
Boolean v1.regexps.delete(String auth_token, UID JEID, UID id)
```

Удаление регулярного выражения

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор регулярного выражения
- **JEID** — идентификатор транзакции

Return value

true

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	Regexp not found

v1.regexps.get.protocol.filters

```
List<ProtocolFilter> v1.regexps.get.protocol.filters(String auth_token)
```

Получить настройки фильтрации протоколов для регулярных выражений

Arguments

- **auth_token** — токен для аутентификации

Return value

Список структур [ProtocolFilter](#)

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.regexps.set.protocol.filters

```
Boolean v1.regexps.set.protocol.filters(String auth_token, UID JEID,  
List<ProtocolFilter> list)
```

Получить настройки фильтрации протоколов для регулярных выражений

Arguments

- **auth_token** — токен для аутентификации
- **list** — список протоколов с настройками
- **JEID** — идентификатор транзакции

Return value

true

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

Component: Storage

В данном разделе описываются RPC-методы для работы с ядром: общие для всех других компонентов действия, например, получение/изменение персональных данных.

v2.storage.init.upload.session

```
StorageUploadSessionInfo v2.storage.init.upload.session(String  
auth_token)
```

Инициализация сессии для загрузки файла в файловое хранилище

Arguments

- **auth_token** — токен для аутентификации

Return value

Структура типа `StorageUploadSessionInfo`, в которой содержится необходимая информация для загрузки. В поле **upload_url** находится URL, на который клиент должен загружать файл. В поле **check_token** находится строка, по которой клиент может отслеживать статус загружаемого файла.

Errors

faultCode	description
104	Неверно указан логин или пароль при попытке аутентификации

v2.storage.check.upload.session

```
StorageUploadSessionCheckInfo  
auth_token, String check_token)
```

Запрос информации о прогрессе сессии загрузки файла

Arguments

- **auth_token** — Invalid auth token
- **check_token** — токен для сессии загрузки, полученный ранее вызовом метода `v2.storage.init.upload.session`

Return value

Структура с информацией о прогрессе загрузки файла. Если состояние загрузки неопределённое (то есть она либо не начиналась, либо уже закончилась), то в структуре **StorageUploadSessionCheckInfo** значения **total_length** и **uploaded_length** равны нулю.

Errors

faultCode	description
104	Invalid auth token
101	Access Denied

11001 Upload session not found

v2.storage.file.fetchInfo

```
StorageFileInfo v2.storage.file.fetchInfo(String auth_token,  
storage_file_uid)
```

Получение информации о файле из хранилища

Arguments

- **auth_token** — Invalid auth token
- **storage_file_uid** — идентификатор файла в хранилище

Return value

Структура с информацией о файле, например, типе (Content-type).

Errors

faultCode	description
104	Invalid auth token
101	Access Denied
11002	File not found in the storage

v2.storage.file.previewInfo

```
FilePreviewInfo v2.storage.file.previewInfo(String auth_token,  
Uid<String> storage_file_uid)
```

Попытаться получить текстовое представление файла из хранилища

Arguments

- **auth_token** — Invalid auth token
- **storage_file_uid** — идентификатор файла в хранилище

Return value

Структура FilePreviewInfo с текстом из файла

Errors

faultCode	description
104	Invalid auth token
101	Access Denied
11002	File not found in the storage

11003

11002 File not found in the storage

v2.storage.file.genExternalUrl

```
String v2.storage.file.genExternalUrl(String auth_token, Uid<String>  
storage_file_uid)
```

Генерация адреса для доступа к файлу из хранилища минуя PHP-бакенд.

Arguments

- **auth_token** — Invalid auth token
- **storage_file_uid** — идентификатор файла в хранилище

Return value

Адрес относительный, без схемы, сервера и порта.

Errors

faultCode	description
104	Invalid auth token
101	Access Denied
11002	File not found in the storage

v2.storage.file.delete

```
Boolean v2.storage.file.delete(String auth_token, Uid<String>  
storage_file_uid)
```

Удаление файла из хранилища

Arguments

- **auth_token** — Invalid auth token
- **storage_file_uid** — идентификатор файла в хранилище

Return value

true, если удаление прошло успешно

Errors

faultCode	description
104	Invalid auth token
101	Access Denied
11002	File not found in the storage

v2.storage.files.list

```
AuthInfo v2.storage.files.list(String auth_token, Filter, Integer Start, Integer Limit)
```

Получить список файлов в хранилище

Arguments

- **auth_token** — Invalid auth token
- **Start** — смещение от начала
- **Limit** — размер листа
- **Filter** — за какой период выбирать файлы

Return value

Лист структур с информацией о файлах

Errors

faultCode	description
104	Invalid auth token
101	Access Denied

Component: DLP

v1.dlp.levels.list

```
List<DlpLevelInfo> v1.dlp.levels.list(String auth_token)
```

Список DLP-уровней.

Arguments

- **auth_token** — токен для аутентификации

Return value

Список структур типа [DlpLevelInfo](#).

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.level.add

```
UID v1.dlp.level.add(String auth_token, DlpLevelInfo level)
```

available since version 3.0

Создание нового уровня DLP.

Arguments

- **auth_token** — токен для аутентификации
- **level** — структура с описанием уровня

Return value

UID созданного уровня.

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.level.update

```
Boolean v1.dlp.level.update(String auth_token, UID id, DlpLevelInfo level)
```

Редактирование уровня DLP.

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор уровня, который нужно изменить
- **level** — структура с описанием уровня

Return value

true

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.level.delete

```
Boolean v1.dlp.level.delete(String auth_token, UID id)
```

Удаление уровня DLP.

Arguments

- **auth_token** — токен для аутентификации
- **id** — идентификатор уровня, который нужно удалить

Return value

true

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.policy.set

```
Boolean v1.dlp.policy.set(String auth_token, UID level\_id, UID group\_id, DlpPolicyType policy)
```

Задание одной политики для DLP.

Arguments

- **auth_token** — токен для аутентификации
- **level_id** — идентификатор уровня
- **group_id** — идентификатор группы
- **policy** — тип политики, который нужно установить для сочетания (**level_id**, **group_id**)

Return value

true

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.policies.list

```
List v1.dlp.policies.list(String auth_token)
```

Запрос списка всех политик.

Arguments

- **auth_token** — токен для аутентификации

Return value

Возвращается список всех политик, каждая политика в виде списка `level_id, group_id, policy`. То есть если в системе 4 группы и 5 уровней, то в результате будет $4 \times 5 = 20$ элементов, для каждой пары (уровень, группа).

Remarks

Если для какой-то пары (`level_id`, `group_id`) не была явно выставлена политика методом [v1.dlp.policy.set](#), то политикой по умолчанию является `SKIP`, именно это и нужно вернуть.

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.groups.list

```
List<DlpGroupInfo> v1.dlp.groups.list(String auth_token)
```

Список DLP-групп. Фактически эти группы являются обычными webfilter-группами, однако не все webfilter-группы являются DLP-группами.

Arguments

- **auth_token** — токен для аутентификации

Return value

Список структур типа [DlpGroupInfo](#).

Errors

faultCode	description
102	Access Denied

104 Invalid auth token

v1.dlp.group.add

```
UID v1.dlp.group.add(String auth_token, DlpGroupInfo group)
```

Создание новой DLP-группы.

Arguments

- **auth_token** — токен для аутентификации
- **group** — параметры новой группы

Return value

Идентификатор созданной группы.

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.group.update

```
Boolean v1.dlp.group.update(String auth_token, UID uid, group)
```

Редактирование DLP-группы.

Arguments

- **auth_token** — токен для аутентификации
- **uid** — идентификатор группы
- **group** — новые параметры группы

Return value

true

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.group.delete

```
Boolean v1.dlp.group.delete(String auth_token, UID uid)
```

Удаление DLP-группы.

Arguments

- **auth_token** — токен для аутентификации
- **uid** — идентификатор группы

Return value

true

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.users.list

```
PagingList<DlpUserInfo> v1.dlp.users.list(String auth_token, Integer start, Integer limit)
```

Получение списка всех DLP-пользователей.

Arguments

- **auth_token** — токен для аутентификации
- **start** — стартовый индекс страницы
- **limit** — лимит записей на страницу

Return value

Список структур типа [DlpUserInfo](#).

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.user.add


```
UID v1.dlp.user.add(String auth_token, DlpUser user)
```

Создание нового DLP-пользователя.

Arguments

- **auth_token** — токен для аутентификации
- **user** — данные о пользователе

Return value

Идентификатор созданного пользователя.

Errors

faultCode	description
102	Access Denied
104	Invalid auth token

v1.dlp.user.fetch

```
DlpUserInfo v1.dlp.user.fetch(String auth_token, UID user_id)
```

Запрос существующего DLP-пользователя.

Arguments

- **auth_token** — токен для аутентификации
- **user_id** — идентификатор пользователя

Return value

Структура типа [DlpUserInfo](#).

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	User not found

v1.dlp.user.update

```
Boolean v1.dlp.user.update(String auth_token, UID user_id, DlpUser user)
```

Создание нового DLP-пользователя.

Arguments

- **auth_token** — токен для аутентификации
- **user_id** — идентификатор пользователя
- **user** — новые данные о пользователе

Return value

true

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	User not found

v1.dlp.user.delete

```
Boolean v1.dlp.user.delete(String auth_token, UID user_id)
```

Удаление DLP-пользователя.

Arguments

- **auth_token** — токен для аутентификации
- **user_id** — идентификатор пользователя

Return value

true

Errors

faultCode	description
102	Access Denied
104	Invalid auth token
105	User not found

Component: Mailsecurity

В этом разделе описываются методы для интеграции с mail security.

Mailsecurity работает как отдельный продукт и использует метод [v1.core.login](#) со специальным логино-паролем для входа в API

вебфилтра. Эта сессия используется для получения юзеров, по которым идёт фильтрация, для активации и т.п.

LicenseUserEmail

Структура описывает юзера, для которого лицензией разрешается использовать антиспам (только для Gateway Antispam).

key	type	description
uid	UID<Integer>	идентификатор записи о юзере
email	String	email юзера
enabled	Boolean	флаг, определяющий, разрешается ли для данного юзера фильтровать почту
send_report	Boolean	флаг, определяющий, нужно ли данному юзеру посылать письмо с отчётом о проверенной почте
auto_collected	Boolean	флаг; true, если данный адрес был получен каким-нибудь процессом сбора адресов и ещё не добавлен в список; при коммите изменений все такие адреса становятся обычными и флаг сбрасывается в false

MailsecSettings

Структура описывает настройки mailsec-подсистемы вебфилтра.

key	type	description
auto_add_license_users	Boolean	true, если нужно автоматически добавлять емейлы в список license users при обработке входящей почты

v1.mailsec.license.users.list

```
List<LicenseUserEmail> v1.mailsec.license.users.list(String
```

Запрос списка обслуживаемых пользователей.

Arguments

- **auth_token** — токен для аутентификации

Return value

Список структур типа [LicenseUserEmail](#).

Errors

faultCode	description
104	Invalid auth token

v1.mailsec.license.users.add

```
List<UID<Integer>> v1.mailsec.license.users.add((String
List<LicenseUserEmail> emails)
```

Добавление новых обслуживаемых пользователей.

Arguments

- **auth_token** — токен для аутентификации
- **emails** — адреса, которые нужно добавить, в виде списка записей типа [LicenseUserEmail](#)

Return value

Список идентификаторов добавленных пользователей.

Errors

faultCode	description
104	Invalid auth token

v1.mailsec.license.user.fetch

```
LicenseUserEmail v1.mailsec.license.user.fetch(String auth_token,
UID<Integer> uid)
```

Запрос одного пользователя.

Arguments

- **auth_token** — токен для аутентификации
- **uid** — идентификатор пользователя

Return value

true

Errors

faultCode	description
104	Invalid auth token
105	User not found

v1.mailsec.license.users.delete

```
List<UID<Integer>> v1.mailsec.license.users.delete(String  
List<UID<Integer>> email_uids)
```

Удаление пользователей из списка обслуживаемых адресов.

Arguments

- **auth_token** — токен для аутентификации
- **email_uids** — идентификаторы адресов, которые нужно удалить

Return value

Список идентификаторов успешно удалённых пользователей.

Errors

faultCode	description
104	Invalid auth token

v1.mailsec.license.user.update

```
Boolean v1.mailsec.license.user.update(String auth_token,  
UID<Integer> uid, LicenseUserEmail email)
```

Arguments

- **auth_token** — токен для аутентификации
- **uid** — идентификатор редактируемого пользователя
- **email** — обновлённая информация о пользователе в виде структуры типа [LicenseUserEmail](#) (обновляются только переданные в этой структуре поля)

Return value

true

Errors

faultCode	description
104	Invalid auth token
108	Достигнут лимит пользователей(ограничение лицензии)

v1.mailsec.settings.get

```
MailsecSettings v1.mailsec.settings.get(String auth_token)
```

Получение настроек mailsec-компонента.

h6 Arguments

- **auth_token** — токен для аутентификации

Return type

Структура типа [MailsecSettings](#).

Errors

faultCode	description
104	Invalid auth token

v1.mailsec.settings.set

```
Boolean v1.mailsec.settings.set(String auth_token, MailsecSettings settings)
```

Получение настроек mailsec-компонента.

h6 Arguments

- **auth_token** — токен для аутентификации
- **settings** — настройки, записываются только переданные параметры

Return type

true

Errors

faultCode	description
104	Invalid auth token

All Errors

Все ошибки, сведенные в одну таблицу

faultCode	description
1	Метод не найден на сервере
2	Метод вернул ошибку
3	Некорректный вызов метода
5	Не найден сервер
6	Нет данных от сервера
100	Неверный пользователь или пароль
101	Не пройдена процедура инсталляции

102	Доступ запрещен
103	Недопустимое значение параметра либо нет параметра
104	Invalid auth token
105	Параметр/запись в бд отсутствует
106	Невозможно связаться с узлом эрланга
107	Запись уже существует и не может быть создана повторно
108	Достигнут лимит пользователей(ограничение лицензии)
109	Ошибка связи с удаленным сервером
110	Список пустой

Коды ошибок FrontEnd

faultCode	description
201	Метод не объявлен в RpcPoint
202	Метод в RpcPoint имеет другое число параметров
403	Сессия истекла
404	Файл Action не найден
903	Валидация параметров для метода не пройдена

Коды ошибок, связанные с активацией

faultCode	description
301	Ошибка записи в базу данных
302	Пин-код не найден
303	Пин-код неактивен
304	Превышение количества активаций
305	Модуль отсутствует в поставке
306	Исходный пин-код не найден
307	Новый пин-код не найден
308	Новый пин-код уже активирован
309	Новый пин-код неактивен
310	Ошибка генерации лицензионного ключа
311	Не отправилось письмо
312	Несоответствие версии
313	Нет данных о пользователе
314	Исходный пин-код уже был преобразован в новый
315	Нет ключей СТ для активации
316	Старый и новый пин-коды относятся к разным продуктам
317	Активация пин-кода для обновления без старого пин-кода
318	Некорректные переменные
319	Несоответствие количества событий
320	Пин-код не соответствует программе

321 "transport_error" "Reply was not recieved"
322 "wrong_user_data" "Wrong user data"
323 "invalid_user_data" "There is no enough data"
324 "wrong_action" "Wrong action for this type of the license"
325 "validation_failed" "Server's response validation failed"
326 "modules_missed" "There are no modules to process"
327 "wrong_type" "Wrong license type while creating module request"
351 not supported method
352 license/key is expired
353 limit of cluster nodes is reached
354 License is not available for this client
355 Hardware id not found
356 Old and new pincodes are the same
357 Demo license is not active
358 Product/key is not supported
359 Product code is not supported
371 Method signature check failed
372 No such subscription
373 Incorrect parameters
374 Pincode is already activated
375 Not registered serial number (for appliances/devices only)
376 Serial number is already linked with another pincode
11001 Upload session not found
11002 File not found in the storage
11003 Preview cannot be created for this kind of file