

Netarium.TV v.4

Описание API сетевого взаимодействия

19.04.24	Изменен метод отправки сообщений. Добавлено описание объектов Media
15.06.24	Добавлены описания метода управления модулями
02.09.24	Исправлено описание вызова Locations:Book
14.09.24	Ребрендинг

Оглавление

Концепция.....	4
Вызов API.....	4
Общие методы доступа к объектам.....	5
Метод GET.....	5
Метод POST.....	6
Метод DELETE.....	7
Ссылки.....	7
Вызов GET.....	7
Вызов POST.....	8
Вызов DELETE.....	8
Предопределенные типы объектов и их структуры.....	8
Класс объекта.....	8
Пользователь.....	8
Роль пользователя.....	9
Приложения Android.....	10
Действия.....	11
Лента новостей.....	11
Новость.....	12
Тип размещение.....	13
Размещение.....	13
Состояние размещения.....	15
Источник EPG.....	15
Источник медиаконтента (плейлист).....	16
Гость.....	19
Конфигурация устройств.....	19
Устройство.....	19
Предопределенные действия.....	20
Работа с Размещением.....	21
Locations:AssignTV.....	21
Locations:Release.....	21
Locations:Maintain.....	21
Locations:Book.....	22
Отправка сообщений (Mailbox:Send).....	22
Действия для ТВ (устройств).....	22
TV:WoL.....	22
TV:SetMode.....	22
TV:Command.....	22
TV:SetConfig.....	22
TV:Screenshot.....	22
TV:Logcat.....	22
TV:ReleaseLocation.....	23
Управление модулями, запущенными на ТВ.....	23

Концепция.

В системе Netarium.TV все объекты имеют общего предка RObject, который в свою очередь имеет следующие поля:

id — идентификатор объекта. Обязательное строковое поле, длиной 72 символа максимум уникальное в рамках системы.

classId — идентификатор класса (типа) объекта. Обязательное строковое поле, длиной 72 символа максимум. Является уникальным идентификатором типа объекта, предопределенные типы описаны ниже. Не запрещается создавать объекты своего собственного типа.

name — наименование объекта. Строковое поле, максимальной длиной 32768 символов, может быть уникальным в рамках класса (для некоторых классов).

Все объекты, передаваемые в вызовах API представлены как JSON. Таким образом, минимальное представление объекта выглядит следующим образом:

```
{
  "id": "00026D892FFC1A7D1DF284",
  "class": "c3bea332-821f-4934-98dc-824aa2c239dc",
  "name": "SaluteTV SaluteTV"
}
```

В случае работы с конкретным объектом его JSON представление может содержать другие поля, которые будут пояснены в описании соответствующего объекта.

Так же, в рамках системы поддерживается механизм «общих» ссылок. Этот механизм позволяет связать два произвольных объекта. Механизм предназначен для формирования пользовательских представлений (например для просмотра устройств в разрезе этажа гостиницы).

Некоторые операции изменяют/создают неявно большое количество объектов. Например «заселение гостя» создает объекты «гость», «заселение», и изменяет объекты «размещение» и «ТВ». Что бы не исключить ошибку в выдаче прав на объекты пользователю предусмотрен механизм *действий*. Действие — это объект, который вызывается через специальный метод API, и выполняет какую-то конкретную операцию.

Вызов API

Вызов методов API осуществляется по протоколу HTTP. Путь к методу выглядит как **/api/метод**, допускаются методы вызова POST, GET, DELETE. При вызове метода в заголовке запроса должен передаваться параметр **X-API-Key**, значение которого должно соответствовать значению **api-key** конфигурации службы **web-api** сервера.

В случае POST вызова параметры передаются в виде JSON в теле запроса.

В случае успешного выполнения запроса, сервер возвращает HTTP-код 200 и результат обработки запроса в виде JSON. Если была ошибка, то сервер может вернуть один из следующих кодов

400 — переданы неверные параметры запроса

403 — у вас нет прав, для доступа к объекту/операции

404 — запрашиваемый объект не найден

500 — ошибка обработки данных на сервере.

В случае, если HTTP код не равен 200, то в теле передается JSON следующей структуры

```
{
  "reason": "Описание ошибки",
  "code": код ошибки (соответствует HTTP-коду)
}
```

Общие методы доступа к объектам

Для доступа/модификации/удаления любого объекта системы предназначен метод **object**.

Метод GET

При GET вызове в параметрах запроса может быть переданы следующие параметры

class — идентификатор класса, список объектов которого вы хотите получить. Возвращает JSON-массив объектов.

Пример: `/api/object?class=3058802b-1e9d-4bd1-a806-edc60abdc717`

Результат:

```
[
  {
    "id": "4d3045cb-bf14-4990-90f0-a212fb4418a8",
    "class": "3058802b-1e9d-4bd1-a806-edc60abdc717",
    "name": "admin",
    "userName": "Super administrator",
    "pincode": "303464",
    "password": "*****",
    "enabled": true,
    "roles": [
      {
        "id": "6b513ca4-0d16-4647-b30f-bce5c5a3e616",
        "class": "3058802b-1e9d-4bd1-a806-63bcfc4ad497",
        "name": "Super administrator"
      }
    ]
  },
  {
    "id": "da59ccbd-3bc7-448b-822d-429e582603bb",
    "class": "3058802b-1e9d-4bd1-a806-edc60abdc717",
    "name": "system",
    "userName": "system",
    "pincode": "",
    "password": "",
    "enabled": true,
    "roles": [
      {
        "id": "6b513ca4-0d16-4647-b30f-bce5c5a3e616",
        "class": "3058802b-1e9d-4bd1-a806-63bcfc4ad497",
        "name": "Super administrator"
      }
    ]
  }
]
```

Данный запрос вернет список пользователей (класс объекта 3058802b-1e9d-4bd1-a806-edc60abdc717).

id — идентификатор конкретного объекта который вы хотите получить. Возвращает JSON-объект.

Пример: **/api/object?id=da59ccbd-3bc7-448b-822d-429e582603bb**

Результат:

```
{
  "id": "da59ccbd-3bc7-448b-822d-429e582603bb",
  "class": "3058802b-1e9d-4bd1-a806-edc60abdc717",
  "name": "system",
  "userName": "system",
  "pincode": "",
  "password": "",
  "enabled": true,
  "roles": [
    {
      "id": "6b513ca4-0d16-4647-b30f-bce5c5a3e616",
      "class": "3058802b-1e9d-4bd1-a806-63bcfc4ad497",
      "name": "Super administrator"
    }
  ]
}
```

Обратите внимание, что при указании и **class** и **id** параметров, параметр **class** игнорируется. Так же, в запрос может быть передан дополнительный параметр **asObject=true**, при наличии которого возвращается минимальная информация об объекте (идентификатор, наименование и идентификатор класса). Этот параметр может быть использован вместе с любым другим параметром.

Пример: **/api/object?class=3058802b-1e9d-4bd1-a806-edc60abdc717&asObject=true**

Результат:

```
[
  {
    "id": "4d3045cb-bf14-4990-90f0-a212fb4418a8",
    "class": "3058802b-1e9d-4bd1-a806-edc60abdc717",
    "name": "admin"
  },
  {
    "id": "da59ccbd-3bc7-448b-822d-429e582603bb",
    "class": "3058802b-1e9d-4bd1-a806-edc60abdc717",
    "name": "system"
  }
]
```

Метод POST.

Этот метод создает/модифицирует объект. Признаком того, что объект нужно создать является отсутствие в JSON запросе поля **id** или параметр запроса **createNew=true**. Если этого поля в запросе нет, то будет создан новый объект, если оно присутствует, то будет попытка модификации существующего объекта. Если указан **id** объекта и параметр **createNew**, то будет попытка создать объект с указанным в запросе **id** при его отсутствии, а при наличии — модификация существующего объекта.

Тело запроса зависит от класса объекта. В теле запроса и при создании и при модификации должно присутствовать поле **class** с указанием класса объекта.

В случае успеха метод возвращает полное описание объекта (с присвоенным новым идентификатором, если было создание объекта).

Пример (создание нового пользователя): **/api/object**

Запрос:

```
{
  "class": "3058802b-1e9d-4bd1-a806-edc60abdc717",
  "name": "user_login",
  "userName": "Новый пользователь",
  "pincode": "",
  "password": "",
  "enabled": true
}
```

```

    "enabled": true
  }
  Ответ:
  {
    "id": "a5eddf9e-5caf-8db0-fd51-711d1be10951",
    "class": "3058802b-1e9d-4bd1-a806-edc60abdc717",
    "name": "user_login",
    "userName": "Новый пользователь",
    "pincode": "332333",
    "password": "",
    "enabled": true,
    "roles": []
  }

```

Метод DELETE

Вызов DELETE предназначен для удаления объекта. Идентификатор удаляемого объекта передается в параметре **id**. Так же может быть передан параметр **links=true**, при наличии этого параметра будет удален объект, и все **объекты**, которые на него ссылаются (см. раздел [ссылки](#)). Если параметр links не указан, то будут удалены только **ссылки** на этот объект (но не сылающиеся объекты). В случае успеха возвращается JSON содержащий id удаленного объекта.

Пример: **/api/object?id=a5eddf9e-5caf-8db0-fd51-711d1be10951**

Результат:

```

{
  "id": "a5eddf9e-5caf-8db0-fd51-711d1be10951"
}

```

Ссылки

Для управления ссылками предназначен метод API link. Метод допускает вызовы GET, POST, DELETE. Обязательным параметром любого запроса является параметр **fromId** — идентификатор объекта от которого будет построена ссылка.

Вызов GET.

Позволяет получить список объектов, которые ссылаются на заданный в поле fromId объект. Допускает дополнительный параметр asObject=true аналогичный такому же параметру в вызове метода [object](#). Если будет передан дополнительный параметр **class**, указывающий класс объекта, то будут возвращены только объекты указанного класса, ссылающиеся на заданный. Возвращает JSON-массив, объектов.

Пример использования: **/api/link?fromId=0002039DA04187BE40224C**

Результат:

```

[
  {
    "id": "033cc671-8016-dd35-dbff-b7c72d954a49",
    "class": "61746224-5eac-4663-a7db-396befeec01c",
    "name": "109",
    "type": 4,
    "parentId": "407a630e-36a5-9d2c-07f2-dc6e0994c520",
    "final": true,
    "state": null,
    "vars": []
  }
]

```

```

    },
    {
      "id": "288b3ecc-90b9-4054-a46f-e73d6fc92cdc",
      "class": "61746224-5eac-4663-a7db-396beffec01c",
      "name": "2-й корпус",
      "type": 1,
      "parentId": null,
      "final": false,
      "state": null,
      "vars": []
    }
  ]
}
]

```

Вызов POST.

Создает ссылки на указанный объект. В теле запроса передается JSON-массив строк, идентификаторы объектов которые должны ссылаться на данный. Возвращает JSON-объект, ключами которого являются Id привязываемых объектов, а значением HTTP коды успеха операции (см. [коды ошибок](#)). Код 200 означает успешно выполненную операцию.

Пример использования: `/api/link?fromId=0002039DA04187BE40224C`

Запрос :

```
["033cc671-8016-dd35-dbff-b7c72d954a49", "288b3ecc-90b9-4054-a46f-e73d6fc92cdc"]
```

Результат :

```

{
  "033cc671-8016-dd35-dbff-b7c72d954a49": 200,
  "288b3ecc-90b9-4054-a46f-e73d6fc92cdc": 200
}

```

Вызов DELETE.

Удаляет ссылку на объекты. Формат запроса и ответа аналогичен вызову POST. Кроме того, в статусах операций может возвращать код 412, который означает что указанные в запросе объекты не связаны.

Пример использования: `/api/link?fromId=0002039DA04187BE40224C`

Запрос:

```
["033cc671-8016-dd35-dbff-b7c72d954a49", "000b3ecc-90b9-4054-a46f-e73d6fc945c", "31a25e66-1818-68c5-237e-7faefae5668"]
```

Результат :

```

{
  "033cc671-8016-dd35-dbff-b7c72d954a49": 200,
  "000b3ecc-90b9-4054-a46f-e73d6fc945c": 404,
  "31a25e66-1818-68c5-237e-7faefae5668": 412
}

```

Предопределенные типы объектов и их структуры

Класс объекта.

Идентификатор класса **311e30e6-82c1-4c99-950e-ea7318704402**

Этот объект описывает класс объекта, имеет минимальный набор полей. Поле name является уникальным.

Пользователь.

Идентификатор класса **3058802b-1e9d-4bd1-a806-edc60abdc717**. Содержит следующие поля

name - строка, логин пользователя. Уникальна в рамках класса.

userName — строка, произвольное наименование пользователя

pincode — строка, ПИН пользователя для входа с мобильных устройств. Только для чтения, в запросах создания/модификации игнорируется.

password — строка, пароль пользователя. Только для записи. При чтении передается строка ********* если пароль задан, или пустая строка если нет. Если при создании/записи передать в этом поле значение отличное от пустого, то пользователю будет назначен указанный пароль.

enabled — логическое, признак активен ли пользователь (разрешен вход через WebUI и мобильные устройства).

roles — JSON-массив, список ролей назначенных пользователю. Только чтение. В системе управление отношением пользователь-роль реализовано со «стороны» ролей (см. ниже). Каждая запись массива — элементарный объект класса «роль» (id 3058802b-1e9d-4bd1-a806-63bcfc4ad497).

Обратите внимание, что в системе существуют два пользователя с id 4d3045cb-bf14-4990-90f0-a212fb4418a8 (Супер администратор) и da59ccbd-3bc7-448b-822d-429e582603bb (системный пользователь), которые не могут быть отключены (значение параметра enabled при модификации пользователя не учитывается). Так же, для системного пользователя нельзя задать пароль.

Пример структуры «пользователь»

```
{
  "id": "710c7c3b-b2a1-464a-9925-47948f6f1f6b",
  "class": "3058802b-1e9d-4bd1-a806-edc60abdc717",
  "name": "animator",
  "userName": "Ведущий рубрики \"Развлечения\"",
  "pincode": "162044",
  "password": "*****",
  "enabled": true,
  "roles": [
    {
      "id": "05c8b64a-4ed9-f38b-c03c-1ab2f84c02a9",
      "class": "3058802b-1e9d-4bd1-a806-63bcfc4ad497",
      "name": "Аниматор"
    }
  ]
}
```

Роль пользователя

Реализует набор прав, доступный пользователям, включенным в эту роль.

Идентификатор класса **3058802b-1e9d-4bd1-a806-63bcfc4ad497**. Содержит следующие поля

allow — JSON-массив, содержащий список объектов, доступ к которым **разрешен** данной роли. Каждый элемент массива является объектом со следующими полями:

id — идентификатор конкретного объекта или специальное значение **dbe589bb-0dc8-4696-a9d4-58c330665439** «все объекты данного класса»

class — идентификатор класса объекта или специальное значение **311e30e6-82c1-4c99-950e-ea7318704402** «все классы»

name — строка, наименование объекта, только чтение

access — число, битовая маска прав доступа. 1 — чтение, 2 — изменение, 4 — удаление.

deny — JSON-массив, содержащий список объектов, доступ к которым **запрещен** данной роли. Структура объектов аналогична объектам массива allow.

users - JSON-массив пользователей, включенных в роль. Содержит минимальные поля id, class, name, userName. При записи все поля кроме id игнорируются.

Обратите внимание, что поля **allow** и **deny** взаимоисключающие и реализуют правила «запрещено все, кроме...» (заполнено allow) и «разрешено все, кроме...» (заполнено deny).

Пример роли:

```
{
  "id": "05c8b64a-4ed9-f38b-c03c-1ab2f84c02a9",
  "class": "3058802b-1e9d-4bd1-a806-63bcfc4ad497",
  "name": "Аниматор",
  "allow": [
    {
      "id": "dbe589bb-0dc8-4696-a9d4-58c330665439",
      "class": "61746224-5eac-4663-a7db-396beffec01c",
      "name": "AllObjects",
      "access": 1
    },
    {
      "id": "dbe589bb-0dc8-4696-a9d4-58c330665439",
      "class": "61ecb0c0-93a2-44b2-9871-d9887b30f2a6",
      "name": "AllObjects",
      "access": 1
    },
    {
      "id": "dbe589bb-0dc8-4696-a9d4-58c330665439",
      "class": "91ccb8b2-eb7c-471d-98db-64759f1e6410",
      "name": "AllObjects",
      "access": 1
    },
    {
      "id": "dbe589bb-0dc8-4696-a9d4-58c330665439",
      "class": "91ccb8b2-eb7c-471d-98db-a3e9c34d054a",
      "name": "AllObjects",
      "access": 3
    }
  ],
  "deny": [],
  "users": [
    {
      "id": "710c7c3b-b2a1-464a-9925-47948f6f1f6b",
      "class": "3058802b-1e9d-4bd1-a806-edc60abdc717",
      "name": "аниматор",
      "userName": "Ведущий рубрики \"Развлечения\""
    }
  ]
}
```

Приложения Android

Описывает приложения Android для TV зарегистрированные в системе. Не может быть создан/модифицирован через вызов POST object.

Идентификатор класса **d26eeb3b-276f-4738-aced-fcb682465463**.

Объект имеет следующие поля:

id,name — всегда имеют значение наименования android пакета приложения

verName — строковое наименование версии

verInt — числовое значение версии

size — размер приложения в байтах.

Пример объекта

```
{
  "id": "cs.Netarium.TV3",
  "class": "d26eeb3b-276f-4738-aced-fcb682465463",
  "name": "cs.Netarium.TV3",
  "verName": "3.11",
  "verInt": 311,
  "size": 24720067
}
```

Для загрузки приложение используется метод API **/apk**

Этот метод поддерживает два вызова POST и GET. При GET в параметре id передается имя пакета, который надо получить. Пакет возвращается в виде application/octet-stream.

В методе POST в теле запроса передается пакет в виде multipart/form-data. При успешном разборе и анализе возвращается JSON-структура описанная выше. При наличии в системе такого же пакета, и при условии, что версия загружаемого больше чем версия загруженного, загруженный заменяется на новую версию.

Действия

Этот класс описывает действия, которые можно совершать над различными объектами.

Идентификатор класса **b80eb51e-9cf0-4fd0-89b9-6e865b5f8f51**.

Класс содержит следующие поля:

class-allow — идентификатор класса объекта, для которого применимо данное действие.

params — строка, описание параметров, которые передаются в данное действие

```
{
  "id": "TV:SetMode",
  "class": "b80eb51e-9cf0-4fd0-89b9-6e865b5f8f51",
  "name": "TV: Switch TV to desired mode",
  "class-allow": "c3bea332-821f-4934-98dc-824aa2c239dc",
  "params": "TV mode as number"
}
```

Для вызова см. раздел «Вызов действий».

Лента новостей

Лента новостей — объект, описывающий подборку новостей определенной тематики

Идентификатор класса **b80eb51e-9cf0-4fd0-89b9-6e865b5f8f51**.

Класс содержит следующие поля:

actualTime — число, время жизни новости в этой ленте в часах. 0 — новость всегда актуальна

```
{
  "id": "7a251355-4652-c0e2-893c-761bd27b2ad8",
  "class": "91ccb8b2-eb7c-471d-98db-64759f1e6410",
  "name": "Новости отеля",
  "actualTime": 24
}
```

Новость

Этот объект описывает статью, размещенную в «Ленте новостей».

Идентификатор класса **91ccb8b2-eb7c-471d-98db-a3e9c34d054a**.

Класс содержит следующие поля:

creator_id — идентификатор пользователя, который создал запись.

creator_name — имя пользователя, который создал запись, только для чтения

published — дата создания новости в UnixTime. Автоматически формируется при создании новости. При передаче в POST вызове при редактировании объекта игнорируется.

title — заголовок новости

link — ссылка на веб-ресурс или другой объект идентифицируемый URI связанный с НОВОСТЬЮ

content — содержимое новости в виде текста или HTML

images — массив строк изображений. Каждая строка содержит изображение в base64 с использованием схемы [data:image](#)

```
{
  "id": "2d7b77f3-7915-f38f-a4f9-9a08061ad532",
  "class": "91ccb8b2-eb7c-471d-98db-a3e9c34d054a",
  "name": "Бассейн",
  "creator_id": "4d3045cb-bf14-4990-90f0-a212fb4418a8",
  "creator_name": "admin",
  "published": 1712640437,
  "title": "Бассейн в нашем отеле",
  "link": "",
  "content": "<p>\n\t<span style=\"color: rgb(33, 37, 41); font-family: Rubik, -apple-system, BlinkMacSystemFont, &quot;Segoe UI&quot;, Roboto, &quot;Helvetica Neue&quot;, Arial, sans-serif, &quot;Apple Color Emoji&quot;, &quot;Segoe UI Emoji&quot;, &quot;Segoe UI Symbol&quot;; font-size: 16px; font-weight: 300; text-align: left; background-color: rgb(255, 255, 255); text-decoration-color: initial; display: inline !important;>В отеле «Джамайка» для гостей оборудован роскошный вместительный бассейн.&nbsp;</span><strong style=\"box-sizing: border-box; font-weight: bolder; color: rgb(33, 37, 41); font-family: Rubik, -apple-system, BlinkMacSystemFont, &quot;Segoe UI&quot;, Roboto, &quot;Helvetica Neue&quot;, Arial, sans-serif, &quot;Apple Color Emoji&quot;, &quot;Segoe UI Emoji&quot;, &quot;Segoe UI Symbol&quot;; font-size: 16px; text-align: left; background-color: rgb(255, 255, 255); text-decoration-color: initial;>350 квадратных</strong><span style=\"color: rgb(33, 37, 41); font-family: Rubik, -apple-system, BlinkMacSystemFont, &quot;Segoe UI&quot;, Roboto, &quot;Helvetica Neue&quot;, Arial, sans-serif, &quot;Apple Color Emoji&quot;, &quot;Segoe UI Emoji&quot;, &quot;Segoe UI Symbol&quot;; font-size: 16px; font-weight: 300; text-align: left; background-color: rgb(255, 255, 255); text-decoration-color: initial; display: inline !important;>&nbsp;</span><strong style=\"box-sizing: border-box; font-weight: bolder; color: rgb(33, 37, 41); font-family: Rubik, -apple-system, BlinkMacSystemFont, &quot;Segoe UI&quot;, Roboto, &quot;Helvetica Neue&quot;, Arial, sans-serif, &quot;Apple Color Emoji&quot;, &quot;Segoe UI Emoji&quot;, &quot;Segoe UI Symbol&quot;; font-size: 16px; text-align: left; background-color: rgb(255, 255, 255); text-decoration-color:
```

```
initial;">метров</strong><span style="color: rgb(33, 37, 41); font-family:
Rubik, -apple-system, BlinkMacSystemFont, &quot;Segoe UI&quot;;, Roboto,
&quot;Helvetica Neue&quot;;, Arial, sans-serif, &quot;Apple Color Emoji&quot;;,
&quot;Segoe UI Emoji&quot;;, &quot;Segoe UI Symbol&quot;; font-size: 16px; font-
weight: 300; text-align: left; background-color: rgb(255, 255, 255); text-
decoration-color: initial; display: inline !important;">&nbsp;лазурной
кристально-чистой поверхности. На прилежащей территории расположена
функциональная зона отдыха. Еду и прохладительные напитки из ресторана можно
заказать прямо к лежакам и в полной мере насладиться курортным вариантом
завтрака/обеда или даже ужина.</span>\n</p>",
  "images": [
    "..."
  ]
}
```

Тип размещение

Объект, описывающий тип размещения (корпус, этаж, номер, комната, континент, планета и т. д.). Не содержит дополнительных полей

Идентификатор класса **5c3300a9-2f11-4a97-8472-134e5b1fc0db**.

```
{
  "id": "6f55c1c4-924f-02c9-4f43-c5031850f0cd",
  "class": "5c3300a9-2f11-4a97-8472-134e5b1fc0db",
  "name": "Стандартный номер без балкона"
}
```

Размещение

Объект описывает размещение — точку, в которую осуществляется заселение и к которой могут быть привязаны устройства. Объект имеет внутреннюю иерархическую структуру, существует метод для отображения дерева объектов.

Идентификатор класса **61746224-5eac-4663-a7db-396beffec01c**.

Объект имеет следующие поля

type — JSON объект класса *Тип размещения*. При изменении/создания объекта *Размещение* требуется передавать только поля **class** и **id** этого объекта

parentId — идентификатор родительского объекта или **null** если объект является корневым

final — логическое. Если установлен в **true** то этот объект не может являться родителем для другого объекта

state — объект класса «Состояние размещения» если данное размещение занято, или **null** если свободно. Только для чтения.

vars — массив объектов переменных подстановки для устройств. Содержит объекты имеющие пару полей **key** и **value**, соответствующие наименованию переменной и его значению.

```
{
  "id": "009acc9a-ab26-490d-0692-6475ca56e6c9",
  "class": "61746224-5eac-4663-a7db-396beffec01c",
  "name": "3116",
  "type": {
    "id": "ff8294e9-a2ec-0826-8922-2c529fdc9594",
    "class": "5c3300a9-2f11-4a97-8472-134e5b1fc0db",
    "name": "2К Люкс"
  },
}
```

```

"parentId": "f3946d1c-cc60-adf2-cbe7-14fcbfa087c8",
"final": true,
"state": {
  "id": "5098e190-f291-435c-8d82-115914b6775a",
  "class": "b2053fd8-ece3-4dff-8f45-7f32bfbdafa34",
  "name": "",
  "date_in": "1712649907",
  "date_out": "0",
  "state": 1,
  "guest_name": "Иванов Иван",
  "call_name": "Иванов Иван",
  "guests": [
    {
      "id": "51aca011-c843-d76b-673e-b07263cad3e2",
      "class": "27312207-9b2c-4549-99f6-d2ffda89820c",
      "name": "Иванов Иван",
      "fullname": "Иванов Иван",
      "callname": "",
      "document": {
        "serie": "",
        "number": "",
        "issueInfo": ""
      }
    }
  ]
},
"vars": [
  {
    "key": "",
    "value": "f4dbd087-cc9b-45fe-8c57-b39c131be9da"
  }
]
}

```

Для доступа к иерархической структуре предназначен метод **location**. Этот метод может принимать параметр **parent** в котором передается идентификатор начального узла, с которого отображается дерево. Если параметр не указан, то отображение начинается с самого верхнего уровня. Возвращается массив объектов «Размещение», однако в каждом объекте присутствует поле **children**, которое является массивом дочерних объектов.

```

[
  {
    "id": "8eed2a91-4829-a72e-923e-6e7cae712799",
    "class": "61746224-5eac-4663-a7db-396beffec01c",
    "name": "0",
    "type": {
      "id": "9e099d5a-e4eb-45b9-b9dd-a5a105d3fe0d",
      "class": "5c3300a9-2f11-4a97-8472-134e5b1fc0db",
      "name": "ЭТАЖ"
    },
    "parentId": "07c705de-1720-b14b-1d81-974c1a805386",
    "final": false,
    "state": null,
    "vars": [],
    "children": [
      {
        "id": "0d61ff53-23d1-0814-077d-42d5f133ccfd",
        "class": "61746224-5eac-4663-a7db-396beffec01c",
        "name": "4",
        "type": {
          "id": "2a6a4aa6-1071-0e49-bcbf-501668e97032",
          "class": "5c3300a9-2f11-4a97-8472-134e5b1fc0db",
          "name": "Семейный Домик"
        }
      }
    ]
  }
]

```

```

    },
    "parentId": "8eed2a91-4829-a72e-923e-6e7cae712799",
    "final": true,
    "state": null,
    "vars": [],
    "children": []
  }
}
]

```

Состояние размещения

Этот объект описывает заселение/перевод в состояния *служебное* или *ремонт* размещения.

Объект этого класса не может быть создан с помощью вызова POST метода object, он может быть только изменен. Удаление данного объекта невозможно.

Идентификатор класса **b2053fd8-ece3-4dff-8f45-7f32bfbdafa34**.

Объект имеет следующие поля

date_in — дата заселения в UnixTime

date_out — дата выезда в UnixTime, может быть 0 если дата выезда не определена.

state — тип состояния. 1 — заселение гостя, 2 — служебное помещение, 3 — помещение в состоянии «обслуживание/ремонт»

guest_name — строка, имя гостя если состояние имеет тип 1. Только для чтения

call_name — строка, обращение к гостю, если состояние имеет тип 1. Только для чтения

guests — JSON массив, список гостей. Должен передаваться при изменении состояния.

Должен содержать хотя бы одну запись, если тип состояния 1. Каждым элементом массива является объект типа «Гость».

```

{
  "id": "5098e190-f291-435c-8d82-115914b6775a",
  "class": "b2053fd8-ece3-4dff-8f45-7f32bfbdafa34",
  "name": "",
  "date_in": "1712649907",
  "date_out": "0",
  "state": 1,
  "guest_name": "Иванов Иван",
  "call_name": "Иванов Иван",
  "guests": [
    {
      "id": "51aca011-c843-d76b-673e-b07263cad3e2",
      "class": "27312207-9b2c-4549-99f6-d2ffda89820c",
      "name": "Иванов Иван",
      "fullname": "Иванов Иван",
      "callname": "",
      "document": {
        "serie": "",
        "number": "",
        "issueInfo": ""
      }
    }
  ]
}

```

Источник EPG.

Объект описывает автообновляемый источник ТВ программ.

Идентификатор класса **e550e446-fb22-4719-8bbe-668687db50c4**.

Объект содержит следующие поля

channels — число разобранных каналов, только чтение

source — адрес источника программ

nextUpdate — дата следующего обновления в UnixTime, только чтение

Для получения списка каналов используется метод **/media/epg**. В этот метод передается параметр **id** — идентификатор объекта класса EPG, возвращается массив, каждый элемент которого имеет следующие поля

chId — идентификатор канала в EPG

name — наименование канала (используется первое найденное в записях)

icon — логотип канала

Источник медиаконтента (плейлист)

Объект описывает источник медиаконтента (список треков).

Идентификатор класса **61ecb0c0-93a2-44b2-9871-d9887b30f2a6**.

Объект содержит следующие поля

source — адрес внешнего источника. Может быть пустым, если плейлист редактируется пользователем.

refresh — период обновления в часах. Должен быть 0 если плейлист редактируется пользователем

epgId — идентификатор объекта «Источник EPG» из которого будет браться программа и логотипы ТВ каналов. Может быть пустым

channels — количество каналов. Только чтение.

external — признак того, что список получается с внешнего источника. Если true то должны быть указаны source и refresh

channels-list — массив, список каналов. При external true только чтение. Каждый элемент массива содержит следующие поля

name — наименование канала.

order — порядок в списке (может быть 0, тогда список сортируется по наименованию)

hash — хешсумма наименования. Используется для привязки к EPG

group — наименование группы

source — адрес источника

args — JSON объект параметров воспроизведения. Может содержать следующие поля

duration — время воспроизведения в секундах, если источник статическое изображение

loop — логическое, признак «зацикливания» трека

muted — логическое, признак воспроизведения без звука

vlc — параметры для плеера VLC, строка, разделены точкой с запятой

Для расчета хеш суммы при добавлении используется метод **/hash** которому передается параметр val со строкой, хеш которой надо посчитать, ответ в виде text/plain — рассчитанная хеш-сумма. Строка передается в URLEncoded виде.

Пример пользовательского плейлиста

```
{
```



```

    "id": "84501245-2a02-b421-6ac1-ce953d3b78fd",
    "class": "61ecb0c0-93a2-44b2-9871-d9887b30f2a6",
    "name": "Реклама",
    "source": "",
    "refresh": 0,
    "epgId": "",
    "channels": 3,
    "external": false,
    "channels-list": [
      {
        "name": "Tempus0",
        "order": 0,
        "hash": "9711893794342463231",
        "group": "No group",
        "source": "http://roomtv.local/Video/video-0.mp4",
        "args": {
          "loop": "true",
          "muted": "true"
        }
      },
      {
        "name": "Tempus1",
        "order": 0,
        "hash": "1108717470601005479",
        "group": "No group",
        "source": "http://roomtv.local/Video/video-1.mp4",
        "args": {}
      },
      {
        "name": "Tempus2",
        "order": 0,
        "hash": "4791119347026700974",
        "group": "No group",
        "source": "http://roomtv.local/Video/video-2.mp4",
        "args": {}
      }
    ]
  }
}

```

Пример «внешнего» плейлиста

```

{
  "id": "9c3d2b1c-19f7-5d54-ec0e-71e15f567b9e",
  "class": "61ecb0c0-93a2-44b2-9871-d9887b30f2a6",
  "name": "IPTV",
  "source":
"http://27e0fa96dabb.goodstreem.org/playlists/uplist/0476eeb2012ae5b3255c98350a8e0cb1/playlist.m3u8",
  "refresh": 24,
  "epgId": "09a5abc9-a28b-a5e7-896b-ccd478769c0c",
  "channels": 126,
  "external": true,
  "channels-list": [
    {
      "name": "360 HD",
      "order": 65535,
      "hash": "662789737",
      "group": "HD",
      "source":
"http://swu0hqz6.megogo.xyz/iptv/QZNCG8X97PWYF6/2065/index.m3u8",
      "args": {}
    },
    {

```

```

        "name": "Viju TV1000 Action HD",
        "order": 65535,
        "hash": "2997376907",
        "group": "HD",
        "source":
"http://swu0hqz6.megogo.xyz/iptv/QZNCG8X97PWYF6/6386/index.m3u8",
        "args": {}
    },
    {
        "name": "Viju TV1000 HD",
        "order": 65535,
        "hash": "1153259050",
        "group": "HD",
        "source":
"http://swu0hqz6.megogo.xyz/iptv/QZNCG8X97PWYF6/6387/index.m3u8",
        "args": {}
    },
    {
        "name": "Viju TV1000 Русское HD",
        "order": 65535,
        "hash": "2368314403",
        "group": "HD",
        "source":
"http://swu0hqz6.megogo.xyz/iptv/QZNCG8X97PWYF6/6388/index.m3u8",
        "args": {}
    },
    {
        "name": "Блокбастер HD",
        "order": 65535,
        "hash": "1045688904",
        "group": "HD",
        "source":
"http://swu0hqz6.megogo.xyz/iptv/QZNCG8X97PWYF6/2436/index.m3u8",
        "args": {}
    },
    {
        "name": "Дом Кино Премиум HD",
        "order": 65535,
        "hash": "799991058",
        "group": "HD",
        "source":
"http://swu0hqz6.megogo.xyz/iptv/QZNCG8X97PWYF6/911/index.m3u8",
        "args": {}
    },
    {
        "name": "Душевное кино HD",
        "order": 65535,
        "hash": "1374918334",
        "group": "HD",
        "source":
"http://swu0hqz6.megogo.xyz/iptv/QZNCG8X97PWYF6/914/index.m3u8",
        "args": {}
    },
    {
        "name": "Жара HD",
        "order": 65535,
        "hash": "4111591569",
        "group": "HD",
        "source":
"http://swu0hqz6.megogo.xyz/iptv/QZNCG8X97PWYF6/2129/index.m3u8",
        "args": {}
    }
}
]
}

```

Гость

Объект описывает персону, которая была заселена в *Размещение*. Объект не может быть удален.

Идентификатор класса **27312207-9b2c-4549-99f6-d2ffda89820c**.

Объект имеет следующие поля

fullname — полное имя гостя

callname — обращение к гостю

document — JSON объект, описывающий документ удостоверяющий личность гостя. Имеет следующие поля:

serie — серия документа

number — номер документа

issueInfo — дополнительная информация (кем выдан и т. д.)

Поля **document** должны присутствовать в запросе создания/изменения, но не обязательно должны быть заполнены при создании.

```
{
  "id": "51aca011-c843-d76b-673e-b07263cad3e2",
  "class": "27312207-9b2c-4549-99f6-d2ffda89820c",
  "name": "Иванов Иван",
  "fullname": "Иванов Иван",
  "callname": "",
  "document": {
    "serie": "XXXX",
    "number": "111111",
    "issueInfo": "Нет информации"
  }
}
```

Конфигурация устройств

Данный объект описывает конфигурацию устройства. **Создание или модификация этого объекта сторонними система не предусматривается, соответственно объект не документирован.**

Идентификатор класса **c3bea332-821f-4934-98dc-f06675ac0fec**.

Устройство

Этот объект описывает подключенное устройство. Создание или модификация объекта средствами API не допускается. Устройство автоматически создается при первом подключении ТВ или приставки к серверу.

Идентификатор класса **c3bea332-821f-4934-98dc-824aa2c239dc**.

Объект имеет следующие поля:

name — производитель и модель устройства

location — размещение устройства. Если устройство привязано к Размещению, то возвращается объект «Размещение», иначе null.

config — объект «Конфигурация устройства».

mode — назначенный режим работы устройства. 0 — выключено, 1 — онлайн, 2 — режим сервисного обслуживания

inMode — текущий режим работы устройства, если устройство в сети

ip — IP адрес, с которого последний раз было подключено устройство

mac — MAC адрес устройства

online -логическое, признак что устройство в сети

vars — JSON объект переменных устройства.

```
{
  "id": "0002039DA04187BE40224C",
  "class": "c3bea332-821f-4934-98dc-824aa2c239dc",
  "name": "DEXP H24H8000C",
  "location": {
    "id": "2f45a96e-0ef7-6e5e-60ac-232d9a35b253",
    "class": "61746224-5eac-4663-a7db-396beffec01c",
    "name": "3316",
    "type": {
      "id": "323ee046-4d90-ed90-ade9-33c29d9a859c",
      "class": "5c3300a9-2f11-4a97-8472-134e5b1fc0db",
      "name": "ЗК СТАНДАРТ 2 спальная кровать б/балкона"
    },
    "parentId": "63822988-8a62-cfd0-7ab8-d3aa1398e8e0",
    "final": true,
    "state": null,
    "vars": []
  },
  "config": {
    "id": "c3bea332-821f-4934-98dc-dcfd9d7d4c07",
    "name": "Default configuration"
  },
  "mode": 2,
  "inMode": 0,
  "ip": "192.168.0.55",
  "mac": "04:05:dd:77:34:fe",
  "online": false,
  "vars": {
    "isNewGuest": "0"
  }
}
```

Предопределенные действия

Для некоторых объектов существуют предопределенные действия, которые выполняют операции над этими объектами. Для вызова действия существует метод **action**. Этот метод допускает только POST вызов, и требует передачу обязательного параметра **id** — идентификатора действия. В отличие от других объектов, для действий **Id** формируется так, что бы можно было понять назначение действия (например «Locations:AssignTV», «TV:SetMode» и т. д.).

В теле запроса передается JSON объект, в котором должно присутствовать обязательное поле **objects** — массив строк, идентификаторов объектов над которыми выполняется данное действие. В массиве должен быть хотя бы один элемент. Так же может присутствовать поле **args** — строка, аргументы передаваемые в действие. Значение и тип аргументов можно получить прочитав описание действия, полученное через метод **object**. Все JSON объекты должны быть сериализованы в строку.

Метод возвращает JSON объект, в котором для каждого id переданного в objects возвращается объект имеющий два поля: success — результат выполнения действия, и result — зависящий от действия.

Примеры вызова

/api/action?id=TV:SetMode

Запрос

```
{
  "objects" : [ "0002039DA04187BE40224C", "000000000" ],
  "args" : "1"
}
```

Результат:

```
{
  "0002039DA04187BE40224C": {
    "success": 200,
    "result": "success"
  },
  "000000000": {
    "success": 404,
    "result": "Object not found"
  }
}
```

Вызов **/api/action?id=TV:Logcat**

Запрос

```
{
  "objects" : [ "0002039DA04187BE40224C" ]
}
```

Результат

```
{
  "0002039DA04187BE40224C": {
    "success": 200,
    "result": "----- beginning of crash\n04-05 18:40:34.632  9018
9083....."
  }
}
```

Ниже описаны зарегистрированные в системе методы.

Работа с Размещением

Locations:AssignTV

Назначает размещение для устройства. В качестве параметра передается идентификатор устройства, которому надо назначить размещение. Если в параметре objects передано несколько размещений, то будет назначено последнее.

Locations:Release

Переводит размещение в состояние «Свободно» (выселение гостя). Не требует дополнительных параметров

Locations:Maintain

Переводит размещение в состояние «Обслуживание/ремонт». В параметрах передается JSON объект в **виде строки**, который должен содержать следующие поля

date_in — дата начала события в UnixTime

date_out — дата окончания события в UnixTime (может быть 0 если не определена)

Locations:Book

Переводит размещение в состояние «Заселено». В параметрах передается JSON объект в **виде строки**, который должен содержать следующие поля

state — число, состояние номера. 1 — заселен, 2 — служебный, 3 — ремонт.

date_in — дата начала события в UnixTime

date_out — дата окончания события в UnixTime (может быть 0 если не определена)

guests — JSON-массив объектов типа «Гость». Объекты могут не существовать в системе (id не указано), при выполнении действия они будут созданы. Должен быть указан минимум один элемент (ответственный гость). Для ответственного гостя в описании должно присутствовать поле isMain: true.

При значении **state** 2 или 3 гостей указывать не обязательно

Отправка сообщений (Mailbox:Send)

Отправляет сообщение. Метод применим к объектам классов «Размещение» или «ТВ». При передаче объекта класса «Размещение» сообщение будет отправлено всем ТВ в этом размещении и в дочерних. Параметром передается текст сообщения, текст может быть оформлен в виде HTML.

Действия для ТВ (устройств)

TV:WoL

Разбудить устройство, если оно поддерживает Wake on Lan. Не требует параметров

TV:SetMode

Переключить ТВ в желаемый режим работы. В параметрах переводится номер режима (0 - выключено, 1 — онлайн, 2 — обслуживание) в виде **строки**

TV:Command

Отправить на ТВ команду. В качестве параметров передается номер команды в виде **строки**. Применим только для устройств в режиме «онлайн». Допустимы следующие команды:

1 — перезапустить Netarium.TV

2 — Выключить экран ТВ (аналогично нажатию кнопки «Питание» на пульте)

16 — перезагрузить ТВ

17 — выключить ТВ

TV:SetConfig

Назначить конфигурацию для устройства. В качестве параметра передается идентификатор конфигурации.

TV:Screenshot

Получить скриншот экрана ТВ. Параметры не требуются, в случае успеха в поле result возвращается PNG изображение в схеме [data:image](#)

TV:Logcat

Получить журнал устройства. Параметры не требуются, в случае успеха возвращается журнал устройства в виде текста.

TV:ReleaseLocation

Удалить ТВ из Размещения. Параметры не требуются.

Управление модулями, запущенными на ТВ

Для управления модулями, запущенными на ТВ предназначен метод /tv. Доступ к нему осуществляется через POST, в параметрах запроса передается поле id — идентификатор устройства, на котором вы будете управлять модулем. В теле запроса передается json со следующими обязательными полями:

moduleId — строка, идентификатор модуля

cmd — число, команда передаваемая модулю. Зависит от того, каким модулем вы хотите управлять

param — необязательная строка, параметры, передаваемые модулю.

Возвращает пустой json, успех передается в виде HTTP кода ответа

Пример вызова /api/tv?id=0002039DA04187BE40224C

Запрос

```
{  
  "moduleId": "WebBrowser",  
  "cmd": 1,  
  "param": "https://rbc.ru"  
}
```

Ответ

```
{}
```