

## Описание протокола взаимодействия сервисов IPTV ПОРТАЛ с внешними системами

### Протокол IPTVPORTAL JSON-SQL ver.0.3 (07.07.2013)

Основные сущности системы (см. Рисунок 1)

subscriber	абонент iptv-телевидения
terminal	абонентское устройство для просмотра iptv-телевидения (стб, pc-плеер)
media	медиа ресурс, доступный для просмотра на абонентском устройстве (tv-канал, видео по запросу, ...)
service	медиа сервис (youtube, vidimax, ivi, ...)
package	пакет медиа ресурсов (пакет каналов, ...)
key	ключ условного доступа для закрытого медиа ресурса
playlist	список медиа ресурсов, отображаемый на абонентском устройстве (рубрикатор каналов, видео ...)
profile	профиль абонента (родительский контроль, ...)
pprofile	профиль плеера, позволяет разграничивать медиа ресурсы в зависимости от типа терминала (pc, stb)
sprofile	системный профиль, позволяет разграничивать медиа ресурсы в зависимости от терминала или абонента
user	администратор системы
domain	настройки системы

### Описание алгоритма работы терминала (STB, PC-плеер)

При включении терминал производит запрос аутентификации (рисунок 2), в качестве аргументов посылаются макадрес и ip-адрес терминала (при наличии также регистрационный идентификатор), в случае успеха возвращается идентификатор сессии, в случае ошибки поведение терминала определяется настройками системы.

В случае разрешения регистрации терминала (allow\_terminal\_reg), терминал посылает запрос регистрации.

Количество терминалов для одного абонента может быть ограничено на уровне системы (max\_terminal\_per\_subscriber), а также на уровне абонента (max\_terminal)

В настройках системы также предусмотрена возможность deregистрации терминала (allow\_terminal\_unreg), позволяющая пользователю самостоятельно производить смену оборудования.

Существует возможность разрешения в системе анонимного пользователя (allow\_anonymous), при этом анонимному пользователю будут доступны бесплатные медиа ресурсы (paid<>true).

Авторизованный терминал получает доступ к бесплатным медиа ресурсам (paid<>true), а также к платным ресурсам (paid=true), основываясь на данных приведенных на рисунке диаграммы классов.

Платный ресурс может иметь срок действия (expired\_on).

Контроль срока действия может быть реализован выбором нужного периода обновления ключа условного доступа (update\_period).



### Протокол IPTVPORTAL JSON-SQL

В качестве транспорта для протокола используется связка **HTTPS** и **JSON-RPC**  
( <http://json-rpc.org/wiki/specification> ).

Функционально протокол реализует команды манипуляции данными **DML** языка SQL, точнее PostgreSQL диалекта ( <http://www.postgresql.org/docs/8.4/interactive/index.html> )

Реализованы основные команды **SELECT, INSERT, UPDATE, DELETE**.  
Название команды указывается в поле **method JSON-RPC**

```
{  
  "jsonrpc" : "2.0",  
  "id"      : 1,  
  "method"  : "select" | "insert" | "update" | "delete",  
  "params"  : {...}  
}
```

**SELECT**

select - запрос выборки данных  
формат команды

```
{
  "distinct" : distinct_clause,
  "data"      : data_clause,
  "from"      : from_clause,
  "where"     : where_clause,
  "group_by" : group_by_clause,
  "order_by" : order_by_clause,
  "limit"     : limit_clause,
  "offset"    : offset_clause
}
```

Пример. Выборка каналов, отсортированных по названию.

```
{
  "jsonrpc" : "2.0",
  "id"      : 1,
  "method"  : "select",
  "params"  : {
    "data"      : ["id", "name", "protocol", "inet_addr", "port"],
    "from"      : "media",
    "where"     : {"is": ["is_tv", true]},
    "order_by" : "name"
  }
}
```

Пример. Выборка абонентов, отсортированных по имени логину.

```
{
  "jsonrpc" : "2.0",
  "id"      : 2,
  "method"  : "select",
  "params"  : {
    "data"      : ["id", "username", "password"],
    "from"      : "subscriber",
    "order_by" : "username"
  }
}
```

## INSERT

**insert** - добавление новых данных

формат команды

```
{
    "into"      : table_clause,
    "columns"   : columns_clause,
    "values"    : values_clause,
    "returning" : returning_clause
}
```

Пример. Добавление пакетов "movie", "sports".

```
{
"jsonrpc" : "2.0",
"id"      : 3,
"method"  : "insert",
"params"  : {
    "into"      : "package",
    "columns"   : ["name", "paid"],
    "values"    : [{"movie", true}, {"sports", true}],
    "returning" : "id"
}
}}
```

Пример. Добавление пакетов "movie", "sports" абоненту с именем акаунта "12345".

```
{
"jsonrpc" : "2.0",
"id"      : 4,
"method"  : "insert",
"params"  : {
    "into"      : "subscriber_package",
    "columns": ["subscriber_id", "package_id", "enabled"],
    "select" : {
        "data": [{"s": "id"}, {"p": "id"}, true],
        "from": [{
            "table": "subscriber", "as": "s"
        }, {
            "table": "package", "as": "p"
        }],
        "where": {
            "and": [{
                "eq": [{"s": "username"}, "12345"]
            }, {
                "in": [{"p": "name"}, "movie", "sports"]
            }
        ]
    }
},
"returning": "package_id"
}}
```

## UPDATE

**update** - обновление существующих данных

формат команды

```
{
  "table"      : table_clause,
  "set"        : set_clause,
  "from"       : from_clause,
  "where"      : where_clause,
  "returning" : returning_clause
}
```

Пример. Отключение абонента с именем акаунта "12345".

```
{
  "jsonrpc" : "2.0",
  "id"      : 2,
  "method"  : "update",
  "params"  : {
    "table" : "subscriber",
    "set"   : {
      "disabled": true
    },
    "where": {"eq": ["username", "12345"]},
    "returning": "id"
  }
}
```

## DELETE

**delete** - удаление существующих данных

формат команды

```
{
    "from"      : from_clause,
    "using"     : using_clause,
    "where"     : where_clause,
    "returning" : returning_clause
}
```

Пример. Удаление абонентских устройств абонента с именем акаунта "test".

```
{
"jsonrpc" : "2.0",
"id"      : 2,
"method"  : "delete",
"params"  : {
    "from" : "terminal",
    "where" : {"in": ["subscriber_id", {
        "select" : {
            "data" : "id",
            "from" : "subscriber",
            "where" : {"eq": ["username", "test"]}
        }
    }]}},
    "returning": "id"
}}
```

Пример. Удаление пакетов абонента с именем акаунта "test".

```
{
"jsonrpc" : "2.0",
"id"      : 2,
"method"  : "delete",
"params"  : {
    "from": "subscriber_package",
    "where": {"in": ["subscriber_id", {
        "select": {
            "data": "id",
            "from": "subscriber",
            "where": {"eq": ["username", "test"]}
        }
    }]}},
    "returning": "id"
}}
```

## Функции и операции

### Логические Операции

`{"and": [op1,op2...]}` - логическое умножение (И)  
`{"or": [op1,op2...]}` - логическое сложение (ИЛИ)  
`{"not": [op1]}` - логическое отрицание (НЕ)

### Операции Сравнения

`{"is": [op1,op2]}`  
`{"is_not": [op1,op2]}`  
`{"eq": [op1,op2]}`  
`{"neq": [op1,op2]}`  
`{"lt": [op1,op2]}`  
`{"gt": [op1,op2]}`  
`{"lte": [op1,op2]}`  
`{"gte": [op1,op2]}`

### Математические Операции

`{"add": [op1,op2...]}` - сложение  
`{"sub": [op1,op2]}` - вычитание  
`{"mul": [op1,op2...]}` - умножение  
`{"div": [op1,op2]}` - деление  
`{"mod": [op1,op2]}` - деление по модулю

### Операторы соответствия

`{"like": [op1,op2...]}`  
`{"ilike": [op1,op2...]}`

### Выражения для подзапросов

`{"in"}`  
`{"not_in"}`



Адрес API: <https://admin.domain.iptvportal.ru/api/jsonsql/>  
где domain – выделенный оператору поддомен в системе IPTV ПОРТАЛ

для работы по протоколу требуется добавлять HTTP-заголовок:  
Iptvportal-Authorization: sessionid=sid  
где sid - идентификатор сессии, полученный при авторизации

### **Авторизация администратора**

Адрес авторизации: <https://admin.domain.iptvportal.ru/api/jsonrpc/>  
где domain – выделенный оператору поддомен в системе IPTV ПОРТАЛ

Запрос авторизации.

```
{
  "jsonrpc" : "2.0",
  "id"      : 1,
  "method"  : "authorize_user",
  "params"  : {
    "username" : "admin",
    "password" : "admin-password"
  }
}
```

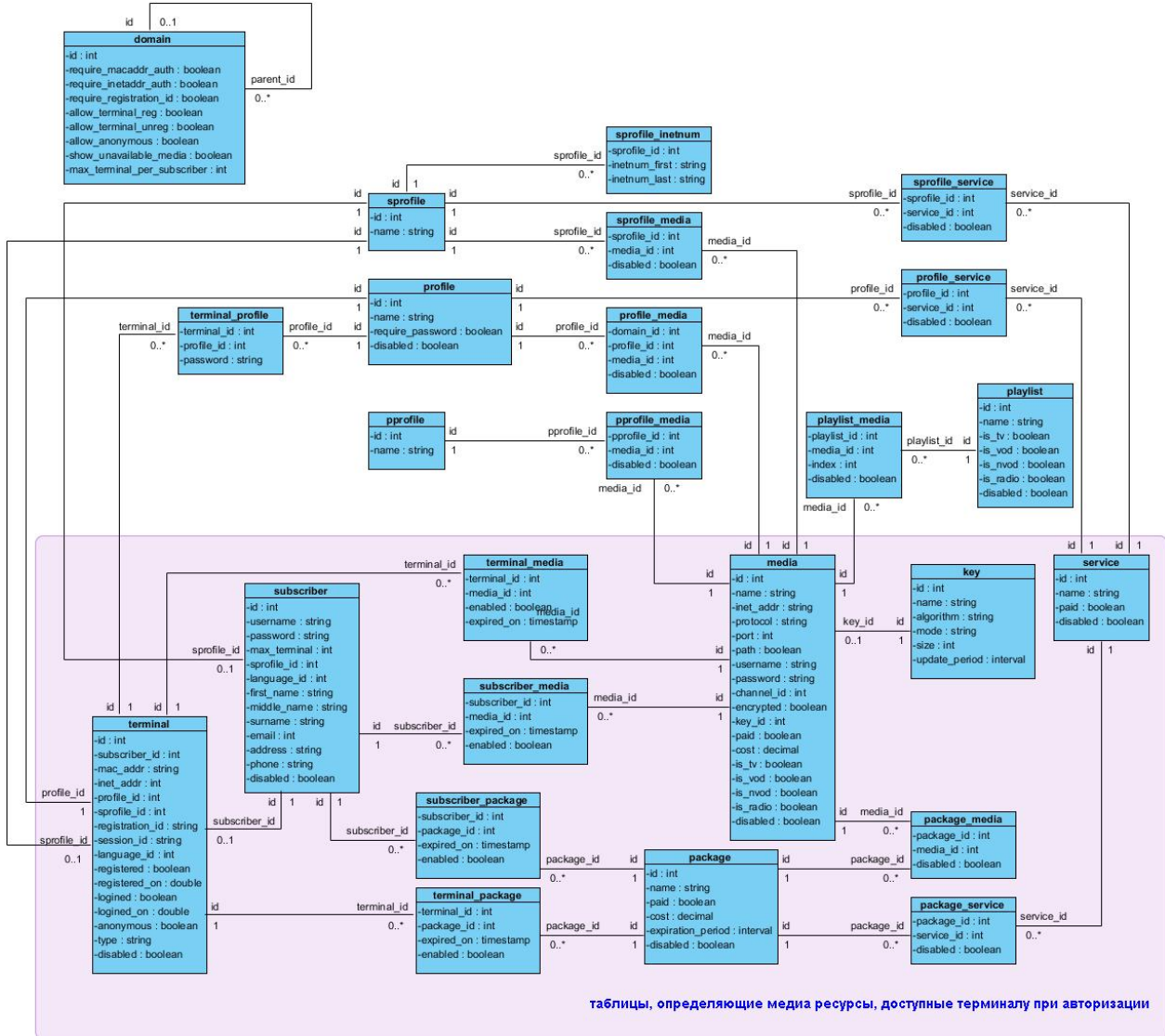
В случае положительной авторизации

```
{
  "jsonrpc" : "2.0",
  "id"      : 1,
  "result"  : {
    "session_id" : "sid"
  }
}
```

В случае ошибки

```
{
  "jsonrpc" : "2.0",
  "id"      : 1,
  "error"   : {
    "message" : "... "
  }
}
```

Рисунок 1: Диаграмма классов



**Рисунок 2:** Диаграмма взаимодействия (терминала с порталом)

