# 流媒体服务器接入管理需求与概要设计

## 运营现状

* 运营部通过配置DNS服务器，根据观众的地区和运营商情况，把相同运营商的就近边缘结点IP列表交给用户，用户一般用返回的第一个ip访问流服务器。
* 当要下架一台流服务器时，先把ip从DNS中撤掉，等出口流量为 0后再做下架工作。
* 新的流服务器准备好时，把ip加入到DNS。

## 遇到的问题

* 有些用户的出口IP所在地区和真正用户所在地区不一致，以至于调度到的服务器不是就近的，网络情况差时只能人工干预修改host。
* 由于某些节点如汕尾电信的性能比较好，很多用户都调度到这个节点，有时会超出节点的出口流量，而另外一些节点如鄂州就会饿死，虽然它的网络情况也不差。
* 由于DNS缓冲的存在，当把ip从DNS中撤掉后，往往会有新的用户调度到这个服务器。

## 定义

流服务器：提供推流或拉流的流服务器。

流服务器组：同一个机房同一运营商下所有流服务器组合。

测试服务器：一个服务器组提供的一个用于测试网速的服务器。

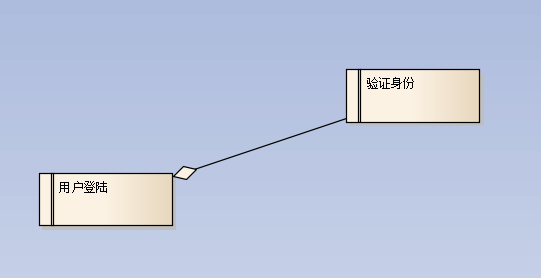
接入点：同一省同一运营商IP总和。

平台管理：用PHP或JAVA实现，用来管理流服务器，流服务器组，接入关系等系统。

## 功能性需求

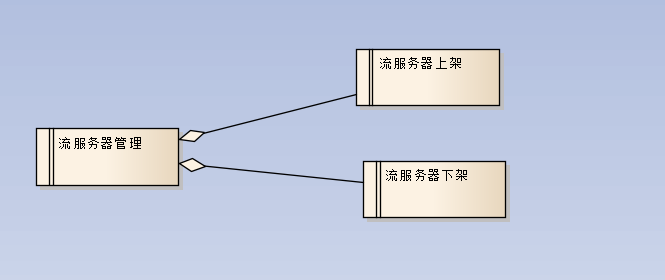
### 运营管理员登陆

管理员通过账号密码登陆流媒体服务器接入系统。



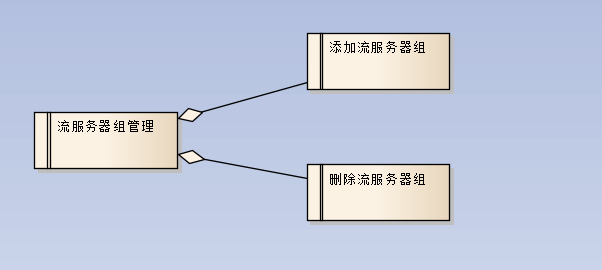
### 流服务器管理

管理员可以上架或者下架某个流服务器，当下架时所有新的接入不能访问此流服务器，当添加一个流服务器时要绑定到相应的流服务器组。这个流服务器的属性参考（5.1数据模型）。



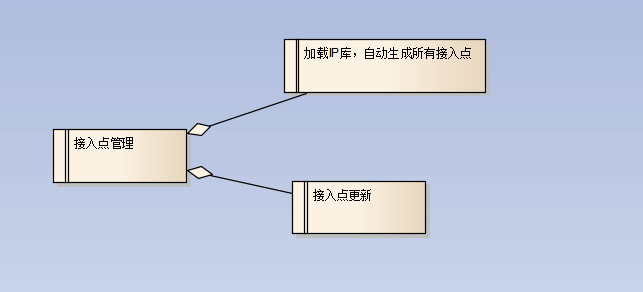
### 流服务器组管理

管理员可添加、删除流服务器组，流服务器组下挂多个流服务器。这个流服务器组的属性参考（5数据模型）。服务器组要根据运营成本分优先级。



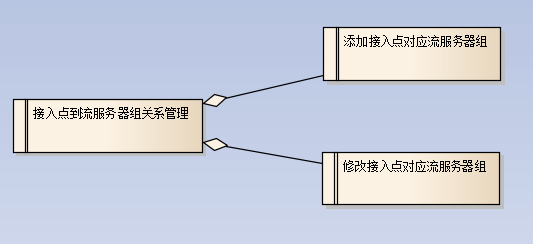
### 接入点管理

管理员修改好IP库后，在系统启动时能自动加载IP库，并生成所有接入点。并能通过简单的接口，在运行过程中更新接入点。这个接入点的属性参考（5数据模型）。



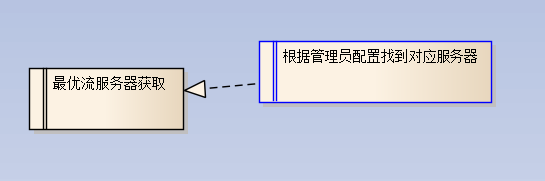
### 接入点到流服务器组关系管理

当系统启动加载ip库后，所有的接入点已经产生。接入点接入到那个流服务器组由管理员指定，把流服务器组分配到不同的接入点。一个接入点配置两个服务服务器组。



### 最优流服务器获取

前期这个功能先做成和原来DNS一样，就是根据管理员分配的接入点对应服务器组情况，在对应的组中选择一个流服务器，两个流服务器组分别提供一个流服务器。



### 接入点到流服器组网速的收集

相关数据收集以分析log为主。记录接入点到服务器组网速情况，同时它记录当前使用的流服务器组，把不是使用网络较好的用户记录下来。

### 流服务器连接数上报

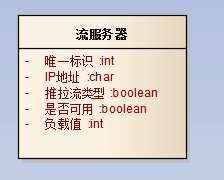
流服务器当前的连接数或网卡流量情况，要上报给接入服务器，以方便调度选择。（这一期不做）

### 定制功能

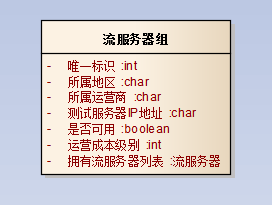
把那些到指定流服务器组速度跟接入配置不一致的用户，存入一个定制库里，以后每个新的用户接入时优先把属于这个定制库的用户指定流服务器组。（这一期不做）

## 数据模型

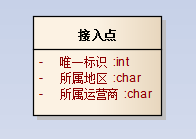
### 流服务器数据模型



### 流服务器组数据模型



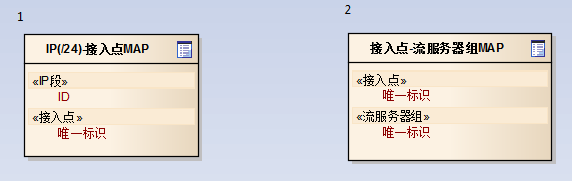
### 接入点数据模型



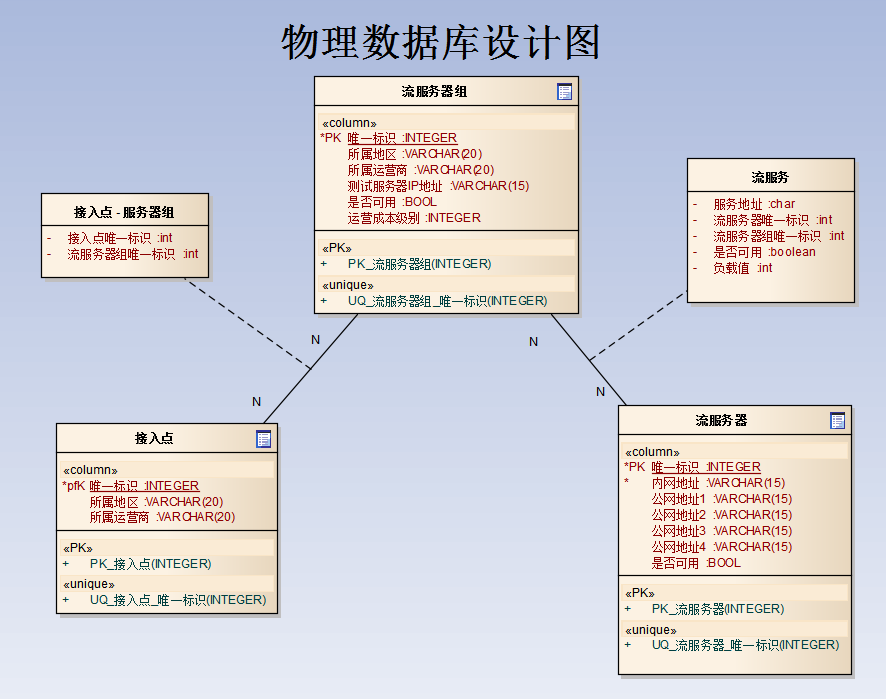
### 相关映射MAP

第一个MAP是每个网段（最终以24位掩码对应的值）到接入点的映射。

第二个MAP是每个接入点到服务器组的映射。

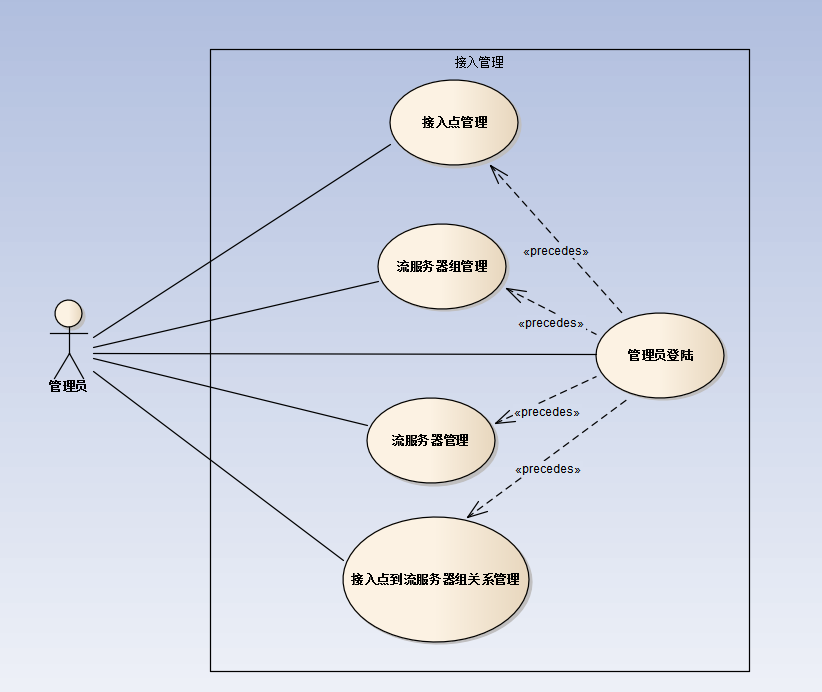


### 数据库物理逻辑

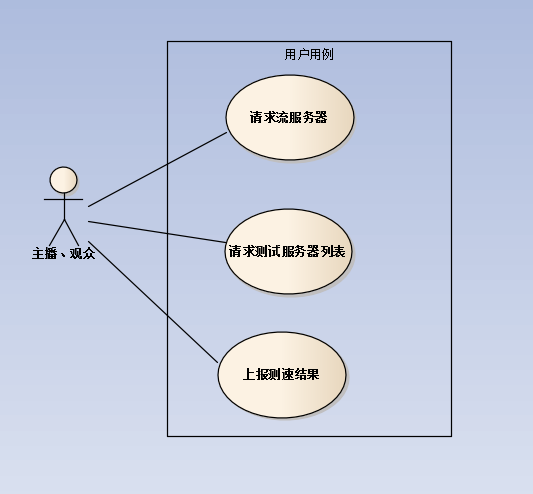


## 用例

### 管理员用例

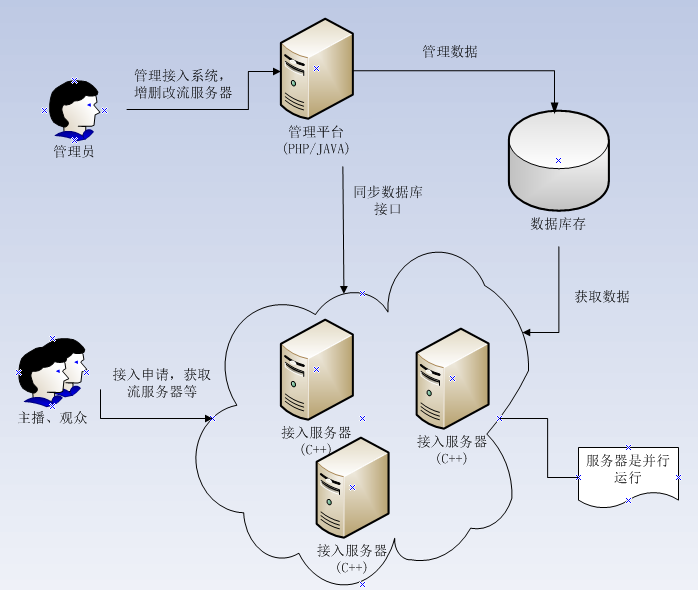


### 主播、观众用例



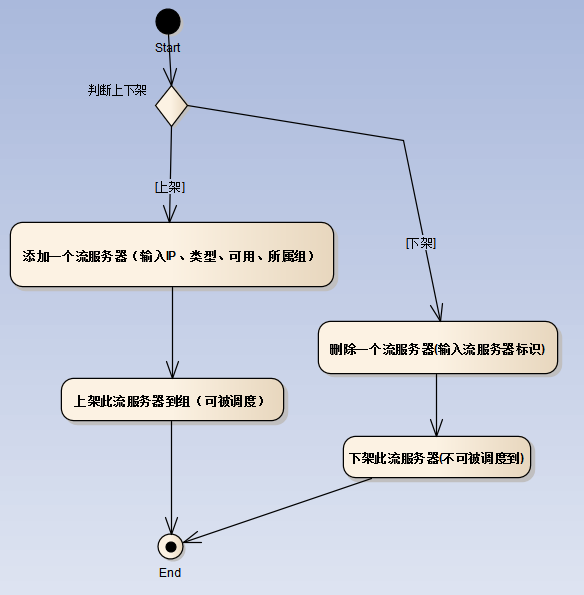
## 整体运营

为了方便管理员管理接入系统，对流服务器，流服务器组和接入关系，这里引入了平台管理服务器，这用PHP或JAVA实现，主要操作数据库，并用HTTP+JSON的简单接口通知接入服务器同步数据库信息。

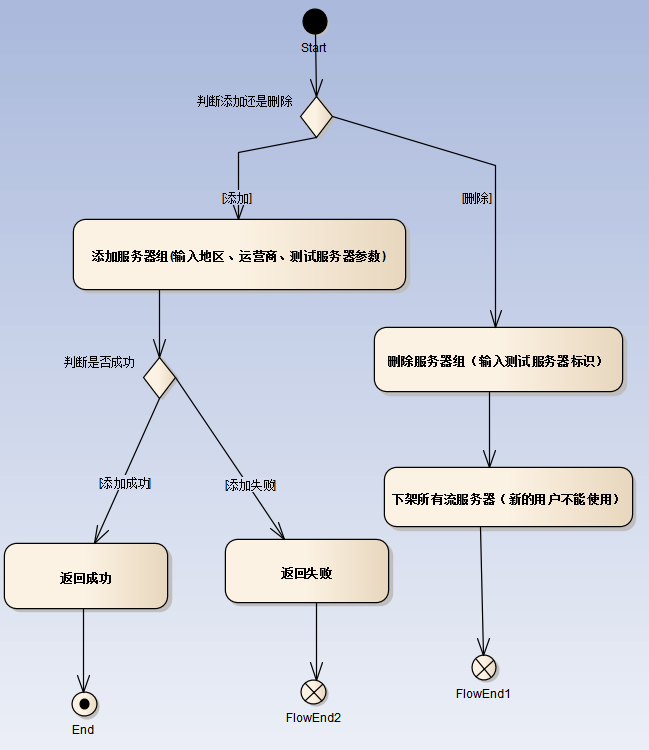


## 详细设计

### 流服务器上架、下架

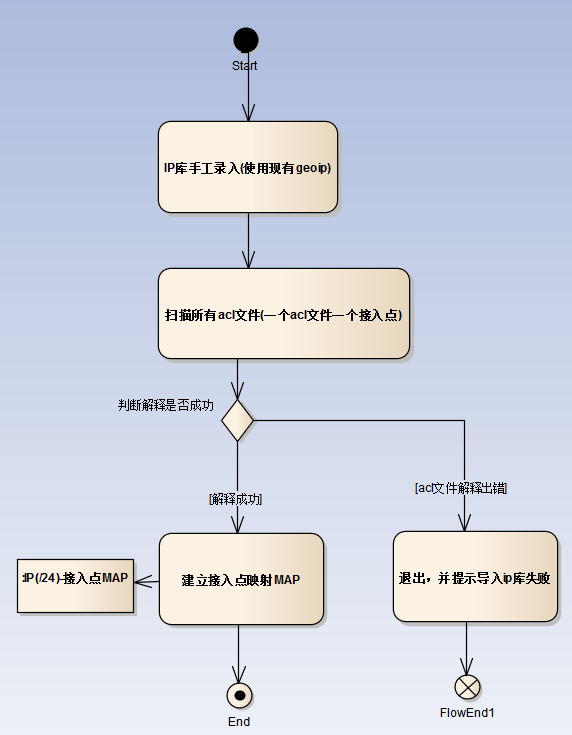


### 流服务器组添加、删除

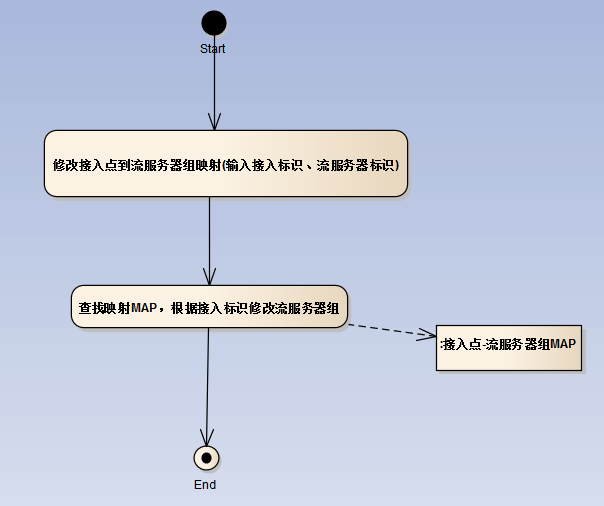


### 建立接入点映射表

沿用原来geoip库的规则，生成接入点映射表，这个表的数据模型见5.4的**IP(/24)-接入点MAP**。

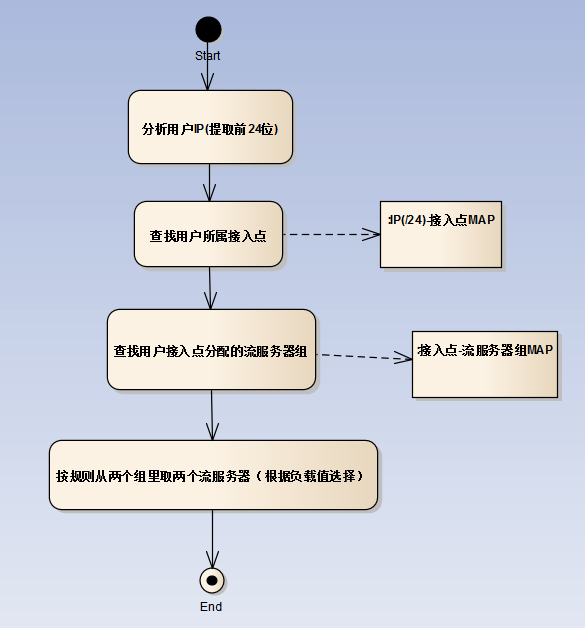


### 接入点到流服务器组关系修改



### 主播、观众请求流服务器

前期这个功能先做成和原来DNS一样，就是根据管理员分配的接入点对应服务器组，在这两个组中选择两个流服务器。流服务器的选择，根据在这个组里负载最轻的。同时根据运营成本把最低的排在前面。



### 收集接入点到流服务器组的网速

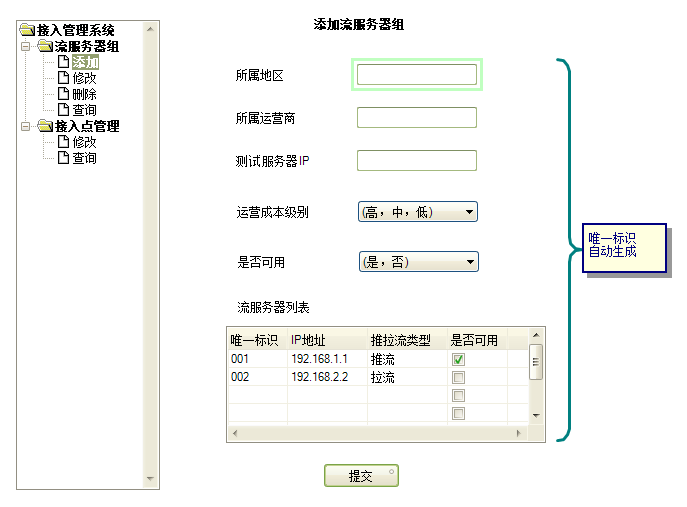
每次用户请求测速时，都把目前所有服务器组返回。同时以log的方式把接入点到所有流服务器组的速度打印出来，后期通过python等工具分析log，得出配置的接入点到流服务器组是否是最优的。

## 管理平台设计

### 登陆



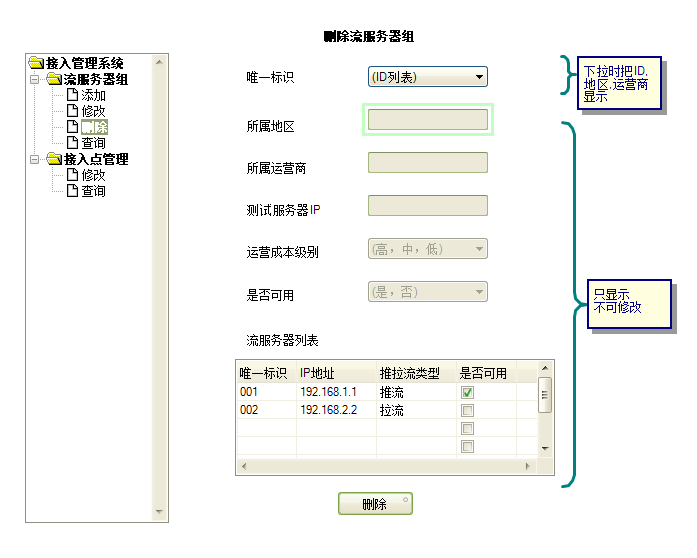
### 添加流服务器组



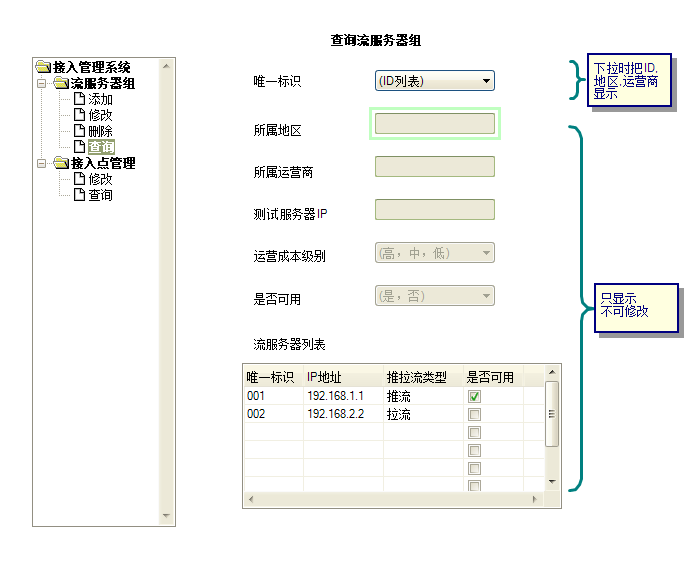
### 修改流服务器组



### 删除流服务器组



### 查询流服务器组



### 修改接入点



### 查询接入点



## 接口

### JS获取直播流服务器列表

* **URL：**http://[ip:port]/api/access/getAddressList.json
* **请求方式：**POST/GET
* **支持格式：**JSON
* **编码方式：**UTF8
* **是否需要用户认证：**否（暂时不做认证加密）
* **请求参数：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 参数说明 | 是否必选 | 数据类型 | 多选 |
| UsageType | 期望回复的网络列表是用于发布(public)或者是播放(play)媒体流。 | 是 | int | 1:发布(publish) 2:播放(play) |
| UserType | 用户类型 | 是 | int | 0:网站游客  1:网站用户  2:手机用户  3:手机游客 |
| UserID | 用户ID | 是 | String | - |
| RoomID | 所在房间ID | 是 | String | - |
| StarID | 主播的ID | 是 | String | - |

**请求示例：**

* POST JSON数据放在content里

{

“UsageType” :1,

“UserType”:0,

“UserID”:”123456”,

“RoomID”:”111111”,

“StarID”:”222222”

}

* **返回参数：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名 | | 参数说明 | 是否必选 | 数据类型 | 多选 |
| Result | | 返回成功与否 | 是 | int | 0:成功  1:获取失败 |
| AddressList | | 可选可用流服务器列表，默认情况下第一个是最优的。 | 否 | Array | - |
|  | Addr | 域名或IP | 是 | String |  |
|  | Port | 端口 | 是 | String |  |

**返回示例1：**

{

“Result”:”0”,

“AddressList”:[

{“Addr”:”src.live.fanxing.com”,”Port”:”1935”},

{“Addr”:”src.live.fanxing.com”,”Port”:”80”}

]

}

**返回示例2：**

{

“Result”:”0”,

“AddressList”:[

{“Addr”:”192.168.1.1”,”Port”:”1935”},

]

}

### JS获取测速服务器列表

* **URL：**http://[ip:port]/api/access/getTestNodeList.json
* **请求方式：**POST/GET
* **支持格式：**JSON
* **编码方式：**UTF8
* **是否需要用户认证：**否（暂时不做认证加密）
* **请求参数：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 参数说明 | 是否必选 | 数据类型 | 多选 |
| UsageType | 期望回复的测速结点列表是用于测试发布(public)或者是播放(play)媒体流。 | 是 | int | 1:发布(publish) 2:播放(play) |
| UserType | 用户类型 | 是 | int | 0:游客  1:网站用户  2:手机用户 |
| UserID | 用户ID | 是 | String | - |
| RoomID | 房间ID | 是 | String | - |
| StarID | 主播ID | 是 | String | - |

**请求示例：**

* POST JSON数据放在content里

{

“UsageType” : 1,

“UserType”:0,

“UserID”:”str123456”,

“RoomID”:”123456”,

“StartID”:”1023234”

}

* **返回参数：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名 | | 参数说明 | 是否必选 | 数据类型 | 多选 |
| Result | | 返回成功与否 | 是 | int | 0:成功  1:获取失败  2:UsageType错误  3:User错误 |
| UserAddr | | 用户IP | 是 | String | - |
| UserCarrier | | 用户网络所属运营商，为空表示找不到 | 是 | String | - |
| UserArea | | 用户网络所属省份，为空表示找不到 | 是 | String | - |
| NodeList | | 可选可用测试结束列表 | 否 | Array | - |
|  | ID | 服务器组ID | 是 | String | - |
|  | Addr | 域名或IP | 是 | String | - |
|  | Port | 端口 | 是 | String | - |
|  | Carrier | 运营商 | 是 | String | - |
|  | Area | 城市 | 是 | String | - |

**返回示例：**

{

“Result”:0,

“UserCarrier”:”电信”,

“UserArea”:”广东”,

“NodeList”:[

{“ID”:”001”,“Addr”:”192.168.1.1”,”Port”:”1935”,”Carrier”:”CMCC”,” Area”:”Guangzhou”},

{“ID”:”001”,“Addr”:”192.168.1.2”,”Port”:”80”,”Carrier”:”CMCC”,” Area”:”Guangzhou”}

]

}

### JS报告测速结果

* **URL：**http://[ip:port]/api/access/reportTestResult.json
* **请求方式：**POST/GET
* **支持格式：**JSON
* **编码方式：**UTF8
* **是否需要用户认证：**否（暂时不做认证加密）
* **请求参数：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 参数说明 | 是否必选 | 数据类型 | 多选 |
| UsageType | 本次测速结果是用于发布(public)或者是播放(play)媒体流。 | 是 | int | 1:发布(publish) 2:播放(play) |
| UserType | 用户类型 | 是 | int | 0:游客  1:网站用户  2:手机用户 |
| UserID | 用户ID | 是 | String | 用户ID |
| RoomID | 房间ID | 是 | String | 房间ID |
| ID | 服务器组ID | 是 | String | - |
| Speed | 测速值(单位kb/s) | 是 | int | - |

**请求示例：**

* POST JSON数据放在content里

{

“UsageType” : 1,

“UserType”:0,

“UserID”:”str123456”,

“RoomID”:”123456”,

“ID”:”001”,

“Speed”:50

}

* **返回参数：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 参数名 | | 参数说明 | | 是否必选 | | 数据类型 | 多选 |
| Result | | 返回成功与否 | | 是 | | int | 0:成功  1:结点错误  2:UsageType错误  3:User错误 |
| AddressList | | | 如果成功，返回这个被测试服务器组的流服务器列表，默认情况下只返回一个。 | 否 | Array | | - |
|  | Addr | | 域名或IP | 是 | String | |  |
|  | Port | | 端口 | 是 | String | |  |

**返回示例：**

{

“Result”:0,

“AddressList”:[

{“Addr”:”192.168.1.2”,”Port”:”80”}

]

}

### JS只报告测速结果(不返回流服务器)

* **URL：**http://[ip:port]/api/access/justReportTestResult.json
* **请求方式：**POST/GET
* **支持格式：**JSON
* **编码方式：**UTF8
* **是否需要用户认证：**否（暂时不做认证加密）
* **请求参数：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 参数说明 | 是否必选 | 数据类型 | 多选 |
| UsageType | 本次测速结果是用于发布(public)或者是播放(play)媒体流。 | 是 | int | 1:发布(publish) 2:播放(play) |
| UserType | 用户类型 | 是 | int | 0:游客  1:网站用户  2:手机用户 |
| UserID | 用户ID | 是 | String | 用户ID |
| RoomID | 房间ID | 是 | String | 房间ID |
| ID | 服务器组ID | 是 | String | - |
| Speed | 测速值(单位kb/s) | 是 | int | - |

**请求示例：**

* POST JSON数据放在content里

{

“UsageType” : 1,

“UserType”:0,

“UserID”:”str123456”,

“RoomID”:”123456”,

“ID”:”001”,

“Speed”:50

}

* **返回参数：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 参数说明 | 是否必选 | 数据类型 | 多选 |
| Result | 返回成功与否 | 是 | int | 0:成功  1:结点错误  2:UsageType错误  3:User错误 |

**返回示例：**

{

“Result”:0

}

### 同步数据库

* **URL：**http://[ip:port]/api/access/updateDatabase.json
* **请求方式：**GET
* **支持格式：**JSON
* **编码方式：**UTF8
* **是否需要用户认证：**否（服务端配置白名，判断是白名单IP请求才执行）
* **请求参数：(没有参数)**
* **返回参数：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 参数说明 | 是否必选 | 数据类型 | 多选 |
| Result | 返回成功与否 | 是 | int | 0:成功  1:失败 |

**返回示例：**

{

“Result”:0

}