

接口说明 v0.0

以下接口均为HTTP（Ajax）接口，如无特殊说明，接口URL均已"http://localhost/musi/"为前缀，以下接口与功能设计 v0.0中所罗列的功能一一对应。

*所有接口均为POST请求；接口及参数名称区分大小写

- 上传音乐

- URL: upload
- 参数:
 - msc_name : String
音乐文件名称
 - msc_data : String
音乐文件数据的Base64编码
- 返回值:
 - {success:true}
上传成功
 - {success:false, reason: "foo"}
上传失败，并说明原因（参数不正确或者服务器内部原因）
- 补充说明:
 - 上传成功后服务器将音乐文件存储至审核区，并在审核音乐的数据库表中加入相应的记录（music_id, music_name, music_path）

- 审核音乐

- URL: audit/submit

- 参数:

- msc_id : String

音乐文件的ID, 在上传成功之后由服务器分配

- adt_pass : Bool

审核结果, True为通过, False为拒绝

- msc_path : String

音乐路径

- msc_ctgy : Int

音乐的类别, 每一位代表一个类别

- msc_name : String

音乐的名称, 因为管理员可能会修改其名称

- msc_sngr : String

音乐的歌手, 值为歌手的ID (如果不知名, 则统一为“匿名歌手”的ID)

- msc_albm : String

音乐的专辑ID (如果不知名, 则统一为“网络歌曲”的ID)

- 返回值:

- {success:true}

审核成功

- {success:false, reason: "foo"}

审核失败, 并说明原因 (参数不正确或者服务器内部原因)

- 补充说明:

- 应该再提供一个接口供管理员后台查询所有待审核的歌曲, 接口说明如下

- URL: audit/query
- 无参数, 返回所有待审核歌曲
- 返回值:
 - {success:true, data:[{msc_id: 123, msc_name: "foo", msc_path: "bar"} ...]}
查询成功, 返回结果数组
 - {success:false, reason: "foo"}
查询失败, 并说明原因 (参数不正确或者服务器内部原因)

- 下载/播放音乐

- URL: music/download
- 参数:
 - usr_id : String
请求者的用户ID
 - msc_id : String
请求音乐的音乐ID
- 返回值: 返回音乐文件 (文件流)
- 补充说明:
 - 该音乐热度+1

- 喜欢音乐

- URL: like
- 参数:
 - usr_id : String

喜欢该音乐的用户ID

- msc_id : String

该用户喜欢的音乐ID

- 返回值:

- {success:true}

- 操作成功成功

- {success:false, reason: "foo"}

- 操作失败失败, 并说明原因 (参数不正确或者服务器内部原因)

- 补充说明:

- 该音乐热度+5

- 如重复出现2次喜欢则为撤销操作

- 添加音乐至歌单

- URL: playlist/add

- 参数:

- msc_id : String

- 要添加的音乐ID

- plylst_id : String

- 被添加的歌单ID

- 返回值:

- {success:true}

- 操作成功成功

- {success:false, reason: "foo"}

- 操作失败, 并说明原因 (参数不正确或者服务器内部原因)

- 补充说明：
 - 应该再提供一个从歌单删除音乐的接口，说明如下
 - URL: `playlist/remove`
 - 参数和返回值同上

- 关键字查询音乐

- URL: `music/query/keyword`
- 参数：
 - `kywd : String`
查询关键字
 - `page : Int`
查询的页数
- 返回值：
 - `{success:true, data: [{msc_id:123, msc_name:"foo", msc_lnth: 123456, msc_sngr: "bar", msc_albm: "foobar", msc_hot:456, msc_ctgy:789} ...]}`
查询成功，返回结果数组
 - `{success:false, reason: "foo"}`
查询失败，并说明原因（参数不正确或者服务器内部原因）
- 补充说明：
 - 查询规则是“歌曲名完全包含关键字”
 - 结果按如下规则排序：如果歌曲名与关键字完全相同，则该记录顶置；否则按照热度降序排列
 - 每页返回20条记录，页号从0开始

- 歌手查询音乐

- URL: music/query/singer

- 参数:

- sng_name : String

查询的歌手关键字

- sng_id : String

查询的歌手ID (UUID)

- page : Int

查询的页数

- 返回值:

- {success:true, data: [{msc_id:123, msc_name:"foo", msc_lnth: 123456, msc_sng: "bar", msc_albm: "foobar", msc_hot:456, msc_ctgy:789} ...]}

查询成功, 返回结果数组

- {success:false, reason: "foo"}

查询失败, 并说明原因 (参数不正确或者服务器内部原因)

- 补充说明:

- 当歌手ID为空时按照歌手关键字查询, 否则按照歌手ID查询

- 查询规则是“歌手名字与关键字完全匹配”

- 结果按按照热度降序排列

- 分页规则与 按关键字查询 一致

- 专辑/歌单查询音乐

- URL: music/query/album

- 参数:

- albm_name : String

查询的专辑关键字

- albm_id : String

查询的专辑ID

- page : Int

查询的页数

- 返回值:

- {success:true, data: [{msc_id:123, msc_name:"foo", msc_lnth: 123456, msc_sngr: "bar", msc_albm: "foobar", msc_hot:456, msc_ctgy:789} ...]}

查询成功, 返回结果数组

- {success:false, reason: "foo"}

查询失败, 并说明原因 (参数不正确或者服务器内部原因)

- 补充说明:

- 查询, 排列, 分页规则与 歌手查询音乐 一致

- 类别查询音乐

- URL: music/query/category
- 参数:
 - ctgy_id : int
查询的类别ID
 - page : Int
查询的页数
- 返回值:
 - {success:true, data: [{msc_id:"123", msc_name:"foo", msc_lnth: 123456, msc_sngr: "bar", msc_albm: "foobar", msc_hot:456, msc_ctgy:789} ...]}
 - 查询成功, 返回结果数组
 - {success:false, reason: "foo"}
 - 查询失败, 并说明原因 (参数不正确或者服务器内部原因)
- 补充说明:
 - 查询, 排列, 分页规则与 歌手查询音乐 一致

- 新上架歌曲榜单

- URL: music/new
- 参数: 无

- 返回值:

- {success:true, data: [{msc_id:"123", msc_name:"foo", msc_lnth: 123456, msc_sngr: "bar", msc_albm: "foobar", msc_hot:456, msc_ctgy:789} ...]}

查询成功, 返回结果数组

- {success:false, reason: "foo"}

查询失败, 并说明原因 (参数不正确或者服务器内部原因)

- 补充说明:

- 服务器内部维护最新添加的100首歌曲列表 (以内存循环队列的形式), 每次添加新的歌曲时更新该列表
- 一次返回所有音乐, 不分页, 无排序规则

- 创建歌单

- URL: playlist/create

- 参数:

- usr_id : String

创建者用户ID

- plylst_name : String

新创建的歌单名称

- 返回值:

- {success:true, plylst_id: "123"}

创建成功, 返回歌单ID

- {success:false, reason: "foo"}

创建失败, 并说明原因 (参数不正确或者服务器内部原因)

- 补充说明：
 - 歌单ID是实施创建的UUID

- 编辑歌单

- URL: `playlist/edit`
- 参数：
 - `plylst_id` : String
被编辑的歌单ID
 - `plylst_name` : String
歌单的新名字（选填）
 - `plylst_cover` : String
歌单的新封面的Base64数据（选填）
- 返回值：
 - `{success:true}`
修改成功
 - `{success:false, reason: "foo"}`
修改失败，并说明原因（参数不正确或者服务器内部原因）
- 补充说明：
 - 歌单名字或者歌单封面两个参数对应的值如果是空（null）则不进行修改
 - 歌单封面的存储规则和音乐一样，存到文件系统，然后在数据库记录其路径名

- 分享歌单

- URL: `playlist/share`

- 参数：
 - `playlist_id : String`
要分享的歌单ID
- 返回值：
 - `{success:true}`
分享成功
 - `{success:false, reason: "foo"}`
分享失败，并说明原因（参数不正确或者服务器内部原因）
- 补充说明：
 - 分享实际上就是将该歌单ID放入“分享歌单池”，从而对所有用户可见（通过浏览歌单的方式）
 - 如果连续两次操作则为取消分享

- 删除歌单

- URL: `playlist/delete`
- 参数：
 - `playlist_id : String`
要删除的歌单ID
- 返回值：
 - `{success:true}`
删除成功
 - `{success:false, reason: "foo"}`
删除失败，并说明原因（参数不正确或者服务器内部原因）
- 补充说明：

- 需注意 `playlist/remove` 为移除歌单中的歌曲， `playlist/delete` 为删除歌单（同时移除所有歌单中的歌曲）

- 浏览歌单

- URL: `playlist/query`
- 参数:
 - `page : Int`
浏览的页数
- 返回值:
 - `{success:true, data:[{plylst_id: 123, plylst_name: "foo", plylst_crtr: "bar", plylst_cvr: "/foo/bar", plylst_hot: 456}, ...]}`
操作成功，并返回结果数组，其中“`plylst_hot`”为歌单的热度值
 - `{success:false, reason: "foo"}`
操作失败，并说明原因（参数不正确或者服务器内部原因）
- 补充说明:
 - 排序规则按添加日期降序排列
 - 每页有10条记录

- 收藏歌单

- URL: `playlist/favorite`
- 参数:
 - `plylst_id : String`

要收藏的歌单ID

- `usr_id : String`

收藏者的用户ID

- 返回值:

- `{success:true}`

收藏成功

- `{success:false, reason: "foo"}`

收藏失败，并说明原因（参数不正确或者服务器内部原因）

- 补充说明:

- 连续操作两次为撤销收藏

- 注册

- URL: `register`

- 参数:

- `usr_phn_nmb : String`

注册手机号

- `usr_pwd : String`

注册用户密码 (32位MD5值)

- 返回值:

- {success:true, usr_id:123}

注册成功, 返回用户ID

- {success:false, reason: "foo"}

注册失败, 并说明原因 (参数不正确或者手机号被占用或者服务器内部原因)

- 补充说明:

- 用户手机号为11位字符串 (安全性检查)

- 注册后自动为用户填充用户名 ("u_" + UUID), 和头像的路径名 (默认图片的路径)

- 注册后应该为用户自动创建一个“我喜欢的”歌单

- 用户ID是UUID (实时创建)

- 登陆

- URL: login

- 参数:

- usr_name : String

登陆用户名 (昵称或手机号)

- `usr_pwd : String`

登陆用户密码（32位MD5值）

- 返回值：

- `{success:true, data: {usr_id: 123, usr_name: "foo",
usr_avatar: "foo/bar", fvrt_plylst: [{plylst_id: 456,
plylst_name: "bar", plylst_cvr: "bar/foo",
plylst_crtr: "vince"}], ...}}`

登陆成功，返回用户信息（用户ID，用户名，用户头像路径，用户歌单列表）

- `{success:false, reason: "foo"}`

登陆失败，并说明原因（密码错误，参数不正确或者服务器内部原因）

- 补充说明：

- 规定用户名不能以数字开头，且用户名唯一（unique），所以如果

`usr_name`以数字开头则通过手机号查询，否则通过用户名查询

- #待定功能：登录时将登陆IP和登录时间记录到日志中，并更新“上次登陆IP和时间”，如果IP不一致则应该通知用户

- 用户信息修改

- URL: `user/edit/info`

- 参数：

- `usr_id : String`

用户ID

- `usr_name : String`

新的用户名

- 返回值:
 - {success:true}
修改成功
 - {success:false, reason: "foo"}
修改失败, 并说明原因 (参数不正确或者服务器内部原因)
- 补充说明:
 - 检查用户名的合法性 (0<长度<20, 不能以数字开头, 要唯一)

- 用户头像更改

- URL: user/edit/avatar
- 参数:
 - usr_id : String
用户ID
 - usr_avtr : String
新的用户头像Base64编码
- 返回值:
 - {success:true}
修改成功
 - {success:false, reason: "foo"}
修改失败, 并说明原因 (参数不正确或者服务器内部原因)
- 补充说明:
 - 得到图片后将图片名改为UUID
 - #待定功能: 为每个用户保存上一次的头像

- 评论

- URL: comment/add
- 参数:
 - usr_id : String
评论者用户ID
 - msc_id : Int
被评论的音乐ID
 - cmnt : String
评论内容
- 返回值:
 - {success:true, cmnt_id: 123}
评论成功, 返回评论UUID
 - {success:false, reason: "foo"}
评论失败, 并说明原因 (参数不正确或者服务器内部原因)
- 补充说明:
 - 评论内容长度 < 100, 每次评论都实时分配一个UUID

- 删除评论

- URL: comment/delete
- 参数:
 - cmnt_id : String
要删除的评论ID
- 返回值:
 - {success:true}

删除成功

- {success:false, reason: "foo"}

删除失败，并说明原因（参数不正确或者服务器内部原因）

- 补充说明：无

- 浏览评论

- URL: comment/query

- 参数：

- msc_id : String

浏览的音乐ID

- 返回值：

- {success:true, data:[{cmnt_id: "123", usr_id: "456",
usr_avtr: "foo/bar" cmnt: "foo", cmnt_date:
"2018-1-28", rply_id: "123456", cmnt_like: 123}, ...]}

操作成功，返回评论列表

- {success:false, reason: "foo"}

操作失败，并说明原因（参数不正确或者服务器内部原因）

- 补充说明：评论列表中的“rply_id”的值为“被回复者的用户ID”，如果为空说明这条评论是正常评论，否则为“回复评论”

- 评论点赞

- URL: comment/like

- 参数：

- usr_id : String

点赞者的用户ID

- cmnt_id : String

喜欢的评论ID

- 返回值:

- {success:true}

点赞成功

- {success:false, reason: "foo"}

点赞失败，并说明原因（参数不正确或者服务器内部原因）

- 补充说明:

- 连续两次操作表示撤销点赞

- 回复评论

- URL: comment/reply

- 参数:

- usr_id : String

评论者用户ID

- cmnt_id : String

被回复的评论ID

- msc_id : Int

被评论的音乐ID

- cmnt : String

评论内容

- 返回值:

- {success:true, cmnt_id: 123}

- 评论成功, 返回评论UUID

- {success:false, reason: "foo"}

- 评论失败, 并说明原因 (参数不正确或者服务器内部原因)

- 补充说明:

- 除了多了一个参数以外, 其他都一样

- #暂定功能: 应当通知被回复者

- 查询推荐歌曲

- URL: recommend/music

- 参数:

- usr_id : String

- 查询者的用户ID

- page : Int

- 推荐歌曲序列的页数

- 返回值:

- {success:true, data: [{msc_id:"123", msc_name:"foo", msc_lnth: 123456, msc_sngr: "bar", msc_albm: "foobar", msc_hot:456} ...]}

- 查询成功, 返回音乐列表

- {success:false, reason: "foo"}

查询失败，并说明原因（参数不正确或者服务器内部原因）

- 补充说明：

- 每次查询都返回一个10首歌曲组成的序列

- 推荐“喜欢相同音乐”的其他用户的喜欢歌曲（目前可以用随机推荐代替）

- 查询推荐歌单

- URL: recommend/playlist

- 参数：

- page : Int

浏览的页数

- 返回值：

- {success:true, data:[{plylst_id: 123, plylst_name: "foo", plylst_crtr: "bar", plylst_cvr: "/foo/bar", plylst_hot: 456}, ...]}

查询成功，并返回歌单列表

- {success:false, reason: "foo"}

查询失败，并说明原因（参数不正确或者服务器内部原因）

- 补充说明：

- 每页20个歌单
- 推荐规则：推荐所有包括用户喜欢的任意一首歌的歌单