付费P2P文件分享软件

开发项目解决方案说明书(SO)

|  |  |  |
| --- | --- | --- |
| 文件状态  [V]草稿  [ ]正式发布  [ ]正在修改 | 文件标识 | 米唻哆-iSeek 1.0.0.1-SO |
| 当前版本 | 1.0 |
| 作 者 |  |
| 完成日期 | 2016-09- |

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 提交日期 | 版本号 | 作 者 | 修 改 内 容 | 备注 |
| 2016-09- | 1.0 |  | 初始版本 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

术语缩写与解释

|  |  |  |  |
| --- | --- | --- | --- |
| 缩 写 | 解 释 | 缩 写 | 解 释 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目 录

1. 解决方案概述 5

2. 交互设计与使用说明 5

3. 技术选型与研发环境 5

4. 总体设计 5

4.1 网络结构 5

4.2 子系统 5

4.3 接口 5

5. 售后服务 5

## 1. 解决方案概述

## 2. 交互设计与使用说明

## 3. 技术选型与研发环境

## 4. 总体设计

### 4.1 网络结构

### 4.2 子系统

### 4.3 身份认证安全设计

身份认证体系借鉴和简化自Kerberos V4。

1）总体约定：

界面输入的任何密码，在系统内一律使用：

**HEX(MD5("5X#6423D79C74b6b+实际输入的密码+AB8471%VAED8A76E"))**

以下简称“变换后的密码”。

（其中“+”表示拼接字串，在实际拼接出的字串中不包括这个加号，下同。）

2）注册及修改密码：凡需传输密码，前端须对“变换后的密码”做AES加密再传输，即：

**Base64(AES("变换后的密码", KEY("CEEF0B3D241F4f3798FF89384BE7ACA1"), "AES/CFB/NoPadding")**

服务端存入数据库时，先解密出“变换后的密码”，再换KEY做AES加密保存，即：

**Base64(AES("变换后的密码", KEY("6A38719F22424b2d94227923E966F9AC"), "AES/CFB/NoPadding")**

3）登录（包括PC端及Web端登录）：前端登录请求中包含登录名，以及：

**Base64(AES("****32Byte随机密钥+登录名+前端系统UTC时间yyyyMMddhhmmss", KEY("变换后的密码"), "AES/CFB/NoPadding")**

其中：

随机密钥由前端在每次登录时随机生成，服务端用它来加密登录响应。

服务端收到请求后，从数据库中读取并解密出“变换后的密码”，作为密钥解密上述密文，核对登录名及前端系统时间，如果登录成功，服务端响应中包含：

**Base64(AES("32Byte传输密钥+登录名+服务器系统UTC时间yyyyMMddhhmmss", KEY("32Byte随机密钥"), "AES/CFB/NoPadding")**

其中：

传输密钥由服务端在每次登录时重新分配，前端登出或断开时销毁，可用来加密后续的关键私密消息。

4）单点登录：通过PC客户端点击链接免输密码进入充值等Web页。在每次点击链接时，先从服务端获取单点登录字串，请求消息中包含登录名，以及：

**Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding")**

如果获取成功，服务端响应包含如下单点登录字串：

**URLCode（Base64(AES("8Byte令牌+登录名+服务器系统UTC时间yyyyMMddhhmmss", KEY("32Byte服务端密钥"), "AES/CFB/NoPadding"))**

其中：

令牌按一定跳进算法（或随机算法）生成，每个令牌只能用来登录一次，且有效期只有3分钟。

服务端密钥只在服务端保存，不下发到客户端。

获取单点登录字串成功后，调用系统浏览器打开URL，在URL中携带单点登录字串进行登录。

如果登录成功，服务端响应中包含：

**Base64(AES("登录名+服务器系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding")**

### 4.4 iSeek.exe与业务服务器的接口

iSeek.exe与业务服务端的接口基于HTTP协议，字符编码为UTF-8，数据封装格式为JSON。

所有数字值，0都表示无效值。

返回字串为json字串（包含flag ，msg ,totalSize,rc对象）

其中flag msg为必备返回字段；totalSize在需要分页的地方需要返回

flag标示请求成功与否的状态（flag=0代表请求成功 flag为非0代表请求失败）

flag=0状态下 如果需要则返回rc数据对象

1：rc对象为数组 返回示例 rc:[{},{}]

2：rc对象为对象 返回示例rc:{}

## 检测软件版本

#### 接口地址

/ iseek/api/pcclient/update

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| String | client\_ver | 客户端版本号 |
| 示例 | / iseek/api/pcclient/update?client\_ver =1 | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc | int | update\_flag: //软件更新标志，u16，1：无需更新，2：非强制更新，3：强制更新 |
|  | List<Package> | update\_file：可选，如果软件有更新，这里是要更新的版本文件列表  Packege结构  url:"xx" //URL  type:0 //文件类型，u16，1：安装包 |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: {  " update\_flag ":1,  "update\_file":  [{  url:"xx" //URL  type:0 //文件类型，u16，1：安装包  }]  }  } | |
|  |  | |

## 获取资讯页内容

#### 接口地址

/ iseek/api/pcclient/getinfopage

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| 无 |  |  |
| 示例 | / iseek/api/pcclient/getinfopage | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc | String | big\_picture\_url //大图URL |
|  | String | small\_picture\_url //小图URL |
|  | List | links: //资讯链接列表 |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: {  " big\_picture\_url:"xx", //大图URL  "small\_picture\_url:"xx" ,//小图URL  links: //资讯链接列表  [{  update\_time:0L //更新时间，u64，UTC时间戳  text:"xx" //显示文字  url:"xx" //URL  }]  }  } | |
|  |  | |

## 用户登录接口

#### 接口地址

/ iseek/api/pcclient/login

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| String | login\_name | 用户登录名 |
| String | login\_str | //登录请求的密文串，生成规则为Base64(AES("32Byte随机密钥+登录名+前端系统UTC时间yyyyMMddhhmmss", KEY("变换后的密码"), "AES/CFB/NoPadding") |
| 示例 | / iseek/api/pcclient/ login? login\_name=root&login\_str=xxx.... | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc | int | user\_id：/用户ID u64 |
|  | String | login\_ret\_str //登录成功响应的密文串，生成规则为Base64(AES("32Byte传输密钥+登录名+服务器系统UTC时间yyyyMMddhhmmss", KEY("32Byte随机密钥"), "AES/CFB/NoPadding") |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: {  user\_id:112L,  login\_ret\_str:"xx"  }  } | |
|  |  | |

## 获取单点登录字串接口

#### 接口地址

/ iseek/api/pcclient/getssostr

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| Long | user\_id | 用户Id u64 |
| String | auth\_str | //身份认证密文串，生成规则为Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding") |
| 示例 | / iseek/api/pcclient/ getssostr? user\_id =11233&auth\_str =xxx.... | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc | String | sso\_str：///获取成功时返回的单点登录字串，客户端无需解析 |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: {  sso\_str:"xx"  }  } | |
|  |  | |

## 获取系统消息(该消息需要客户端每10秒调用一次)

#### 接口地址

/ iseek/api/pcclient/getnotice

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| long | user\_id | 用户Id u64 |
| String | auth\_str | //身份认证密文串，生成规则为Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding") |
| 示例 | / iseek/api/pcclient/ getnotice? user\_id =11233&auth\_str =xxx.... | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc | List<Notice> | Notice的结构:  type：//消息类型，u8，1：会员续费或充值成功提醒，2：会员到期提醒，3：公告信息  notice:"xx" //消息内容  url:"xx" //可选，链接URL |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: [  {  type:1 ,  notice:"xx1",  url:"xx1" //可选，链接URL  },  {  type:2,  notice:"xx2",  url:"xx2" //可选，链接URL  }  ]  } | |
|  |  | |
|  |  | |

## 获取用户信息接口

#### 接口地址

/ iseek/api/pcclient/getuserinfo

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| int | user\_id | 用户Id u32 |
| String | auth\_str | //身份认证密文串，生成规则为Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding") |
| 示例 | / iseek/api/pcclient/getuserinfo? user\_id =11233&auth\_str =xxx.... | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc |  |  |
|  | String | login\_name:"xx" //登录名 |
|  | String | nickname:"xx" //昵称 |
|  | Int | sex:0 //性别，u32，1：男，2：女 |
|  | String | mobile:"xx" //手机号码 |
|  | String | email:"xx" //邮箱 |
|  | long | member\_expiry\_day:0L //会员到期日，u64，UTC时间戳，精确到天 |
|  | Int | share\_compress :0L //是否能共享压缩包 1可以 2不可以 |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: {  login\_name:"xx" //登录名  nickname:"xx" //昵称  sex:0 //性别，u8，1：男，2：女  mobile:"xx" //手机号码  email:"xx" //邮箱  member\_expiry\_day:0L //会员到期日，u64，UTC时间戳，精确到天  share\_compress：1 //是否能共享压缩包 1可以 2不可以  }  } | |
|  |  | |

## 获取关注的用户信息

#### 接口地址

/ iseek/api/pcclient/ getfriends

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| int | user\_id | 用户Id u32 |
| String | auth\_str | //身份认证密文串，生成规则为Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding") |
| 示例 | / iseek/api/pcclient/ getfriends? user\_id =11233&auth\_str =xxx.... | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc | List<Friend> | Friend的结构  friend\_id:2 //用户ID，u64  login\_name:"xx" //登录名  nickname:"xx" //昵称 |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: [{  friend\_id:1 //用户ID，u64  login\_name:"xx" //登录名  nickname:"xx" //昵称  },  {  friend\_id:2 //用户ID，u64  login\_name:"xx" //登录名  nickname:"xx" //昵称  },  ]  } | |
|  |  | |

## 关注或取消关注用户

#### 接口地址

/ iseek/api/pcclient/ setfriend(关注是单向的，即A关注B，B并不一定关注A)

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| Long | user\_id | 用户Id u64 |
| String | auth\_str | //身份认证密文串，生成规则为Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding") |
| Long | friend\_id | 关注或取消对象的id u64 |
| String | is\_friend | 1：关注，2：取消关注 |
| 示例 | / iseek/api/pcclient/ setfriend? user\_id =11233&auth\_str =xxx....& is\_friend=1 | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc |  | 空 |
| 示例 | {  “flag”: 0,  “msg”:"成功",  } | |
|  |  | |

## 搜索文件

#### 接口地址

/ iseek/api/pcclient/ seekfile

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| long | user\_id | 用户Id u64 |
| String | auth\_str | //身份认证密文串，生成规则为Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding") |
| String | key\_word | 搜索关键字 |
| String | file\_type | //要搜索的文件类型，u8，1：音频文件，2：压缩文件，3：文件专辑 0: 全部 |
| String | style\_type | //要搜索的文件风格类型 如果搜索全部风格类型请传空 |
| 示例 | / iseek/api/pcclient/ seekfile? user\_id =11233&auth\_str =xxx.... | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc | List<Files> | Files的结构  file\_id:0L //文件ID，u64  hash:"xx" //文件MD5  type:0 //文件类型，1：MP3，2：Wav，3：压缩文件  name:"xx" //文件名  size:0L //文件大小，u64，字节数  time\_span:0 //可选，音乐时间长度，u32，秒数  kbps:0 //可选，音乐比特率，u32  src\_count:0 //源数量，u32 |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: [{  file\_id:0L //文件ID，u64  hash:"xx" //文件MD5  type:0 //文件类型，1：MP3，2：Wav，3：压缩文件  name:"xx" //文件名  size:0L //文件大小，u64，字节数  time\_span:0 //可选，音乐时间长度，u32，秒数  kbps:0 //可选，音乐比特率，u32  src\_count:0 //源数量，u32  },  {  file\_id:0L //文件ID，u64  hash:"xx" //文件MD5  type:0 //文件类型，1：MP3，2：Wav，3：压缩文件  name:"xx" //文件名  size:0L //文件大小，u64，字节数  time\_span:0 //可选，音乐时间长度，u32，秒数  kbps:0 //可选，音乐比特率，u32  src\_count:0 //源数量，u32  },  ]  } | |
|  |  | |

## 获取文件所有者

#### 接口地址

/ iseek/api/pcclient/getfileowners

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| long | user\_id | 用户Id u64 |
| String | auth\_str | //身份认证密文串，生成规则为Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding") |
| String | file\_id | 文件id |
| 示例 | / iseek/api/pcclient/ getfileowners? user\_id =11233&auth\_str =xxx.... | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc | List<User> | User的结构  user\_id:0L //用户ID，u64  login\_name:"xx" //登录名  nickname:"xx" //昵称  share\_file\_count:0 //共享的文件数量，u32  is\_friend:0 //是否已关注，u8，1：已关注，2：未关注 |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: [{  user\_id:0L //用户ID，u64  login\_name:"xx" //登录名  nickname:"xx" //昵称  share\_file\_count:0 //共享的文件数量，u32  is\_friend:0 //是否已关注，u8，1：已关注，2：未关注  },  {  user\_id:0L //用户ID，u64  login\_name:"xx" //登录名  nickname:"xx" //昵称  share\_file\_count:0 //共享的文件数量，u32  is\_friend:0 //是否已关注，u8，1：已关注，2：未关注  },  ]  } | |
|  |  | |

## 获取单个用户共享的文件

#### 接口地址

/ iseek/api/pcclient/ seekfilebyuser

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| long | user\_id | 用户Id u64 |
| String | auth\_str | //身份认证密文串，生成规则为Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding") |
| String | files\_owner\_user\_id | //文件拥有者用户ID，u64 |
| 示例 | / iseek/api/pcclient/ seekfilebyuser? user\_id =11233&auth\_str =xxx.... | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc | List<Files> | Files的结构  file\_id:0L //文件ID，u64  hash:"xx" //文件MD5  type:0 //文件类型，1：MP3，2：Wav，3：压缩文件  name:"xx" //文件名  size:0L //文件大小，u64，字节数  time\_span:0 //可选，音乐时间长度，u32，秒数  kbps:0 //可选，音乐比特率，u32  src\_count:0 //源数量，u32  sub\_dir:"xx" //文件所在目录，软件设置的共享目录之下的相对路径，两级目录间用“\”隔开，路径最后要带上“\” |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: [{  file\_id:0L //文件ID，u64  hash:"xx" //文件MD5  type:0 //文件类型，1：MP3，2：Wav，3：压缩文件  name:"xx" //文件名  size:0L //文件大小，u64，字节数  time\_span:0 //可选，音乐时间长度，u32，秒数  kbps:0 //可选，音乐比特率，u32  src\_count:0 //源数量，u32  sub\_dir:"xx" //文件所在目录，软件设置的共享目录之下的相对路径，两级目录间用“\”隔开，路径最后要带上“\”  },  {  file\_id:0L //文件ID，u64  hash:"xx" //文件MD5  type:0 //文件类型，1：MP3，2：Wav，3：压缩文件  name:"xx" //文件名  size:0L //文件大小，u64，字节数  time\_span:0 //可选，音乐时间长度，u32，秒数  Kbps:0 //可选，音乐比特率，u32  src\_count:0 //源数量，u32  sub\_dir:"xx" //文件所在目录，软件设置的共享目录之下的相对路径，两级目录间用“\”隔开，路径最后要带上“\”  },  ]  } | |
|  |  | |

## 上传共享的文件列表

#### 接口地址

/ iseek/api/pcclient/ putfilesinfo

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| long | user\_id | 用户Id u64 |
| String | auth\_str | //身份认证密文串，生成规则为Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding") |
| String(Json格式) | files | 如:[{  hash:"xx" //文件MD5  type:0 //文件类型，1：MP3，2：Wav，3：压缩文件  name:"xx" //文件名  size:0L //文件大小，u64，字节数  time\_span:0 //可选，音乐时间长度，u32，秒数  kbps:0 //可选，音乐比特率，u32  share\_dir:"xx" //文件所在的，软件设置的共享目录，两级目录间用“\”隔开，路径最后要带上“\”  sub\_dir:"xx" //软件设置的共享目录之下的相对路径，两级目录间用“\”隔开，路径最后要带上“\”  }] |
| 示例 | / iseek/api/pcclient/ putfilesinfo? user\_id =11233&auth\_str =xxx....&files=[{}] | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc |  |  |
| 示例 | {  “flag”: 0,  “msg”:"成功",  } | |
|  |  | |

## 获取系统公告

#### 接口地址

/ iseek/api/pcclient/getsysnotice

#### 请求参数

|  |  |  |
| --- | --- | --- |
| **类型** | **说明** | |
| int | user\_id | 用户Id u32 |
| String | auth\_str | //身份认证密文串，生成规则为Base64(AES("登录名+客户端系统UTC时间yyyyMMddhhmmss", KEY("32Byte传输密钥"), "AES/CFB/NoPadding") |
| 示例 | / iseek/api/pcclient/ getsysnotice? user\_id =11233&auth\_str =xxx.... | |

#### 返回结果

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| flag | int | 状态标志  0:成功  非0:失败 |
| msg | string | 提示信息 |
| rc |  |  |
|  | String | notice\_content:"xx" //公告内容 |
|  | String | url:"xx" //公告url 可为空 |
|  | Long | created\_time //公告创建时间戳 u64 |
| 示例 | {  “flag”: 0,  “msg”:"成功",  “rc”: {  notice\_content:"xx" //公告内容  url:"xx" //公告url 可为空  created\_time //公告创建时间戳 u64  }  } | |
|  |  | |

## 一些web界面的地址

1.下载首页 ：/index.html 不需要带sso\_str

2.注册：/reg1 不需要带sso\_str

3.修改资料:/ updateuser 需要带sso\_str

4.支付: /pay 需要带sso\_str

5.修改密码:/ resetpass 不需要带 sso\_str

需要带sso\_str得地址 示例：

访问修改资料界面: http://14.152.95.42 /upateuser?sso\_str=xxxxxxxxxxxxxxxx

#### 4.4.13 返回码定义

0：正常

10001：内部错误

10002：参数错误

10003：网络错误

20001：验证错误

...

### 4.5. UDPClient.dll提供的调用接口

UDPClient.dll提供的调用接口，也是整个P2P体系对外提供的接口，为C格式的API。

详细定义见IUDPClient.h。

### 4.6. 数据库

## 5. 质量保证及维护说明