**接口文档**

**2024-06-25**

**用户类**

1. **用户登录**

**接口说明：**本接口用于用户登录

**URL：**/postLogin

**接口类型：**POST

**Params：**{

username : String

password : String

userhash : String

}

**响应格式：**JSON

**返回响应：**

状态码200 : {

successful : Boolean

error : Integer

userhash : String

}

**参数说明：**

username : 用户名

password : 密码

userhash : 自动登录校验

**接口功能：**

本接口功能分为主动登录和自动登录两种。

主动登录请求参数为username和password，自动登录请求参数为userhash。

主动登录：

1. 遍历数据库USERTABLE表的USERNAME字段，与username参数匹配，若无匹配，则返回以下响应：

{

    successful : false,

    error : 2

}

1. 在情况1中，若发现匹配，将username参数和password参数进行字符串相加，相加后的字符串使用SHA-1算法加密，将该记录的HASH字段与加密后的内容匹配，若不一致，则返回以下响应：

{

    successful : false,

    error : 1

}

1. 在情况2中，若匹配一致，则将该记录HASH字段按照以下格式返回响应：

{

    successful : true,

    userhash : userhash

}

自动登录：

1. 遍历数据库USERTABLE表的HASH字段，与userhash参数匹配，若无匹配，则与情况1返回相同响应。
2. 在情况4中，若匹配一致，则与情况3返回相同响应。
3. **用户注册**

**接口说明：**本接口用于用户注册

**URL：**/postLogon

**接口类型：**POST

**Params：**{

username : String

password : String

}

**响应格式：**JSON

**返回响应：**

状态码200 : {

successful : Boolean

error : Integer

}

**参数说明：**

username : 用户名

password : 密码

**接口功能：**

1. 遍历数据库USERTABLE表的USERNAME字段，若出现与username参数相同的记录，则返回以下内容:

{

    successful : false,

    error : 2

}

1. 在情况1中，若未发现相同记录，则增加新记录，AVATOR字段为默认头像，USERNAME字段为username参数，HASH字段为username参数和password参数进行字符串相加，相加后的字符串使用SHA-1算法加密得出的结果，TIME字段为当前日期，并返回以下内容：

{

    successful : true

}

1. **获取所有用户信息**

**接口说明：**获取所有用户信息

**URL：**/getUserAll

**接口类型：**GET

**Params：**无

**响应格式：**JSON

**返回响应：**

状态码200 : {

users : [

{

id : String

name : String

state : String

deal : String

count : String

phone : String

adress : String

time: [“”, “”, “”]

}

]

}

**参数说明：**

id : 用户ID

name : 用户昵称

state : 用户状态，分为“正常”和“黑名单”

deal : 用户订单数

count : 用户积分

phone : 联系方式

adress : 默认地址

time : 注册时间

**接口功能：**

接收到请求后，遍历USERTABLE表，读取所有记录的ID、NICKNAME、STATE、DEAL、VIPCOUNT、DEFAULTADDRESS、TIME字段。

将TIME字段数据分成年月日三段存为数组。

根据每个记录中的DEFAULTADDRESS字段，遍历USERADDRESSTABLE表，查找ID字段与DEFAULTADDRESS相同的记录，读取该记录的ADDRESS和PHONE字段。

最后将所有数据存为一个数组，按如下格式返回：

{

users: [

{

id: ID,

name: NICKNAME,

state: STATE,

deal: DEAL,

count: VIPCOUNT,

phone: PHONE,

address: ADDRESS,

time:[YEAR,MONTH,DAY]

}

]

}

1. **获取当前登录用户信息**

**接口说明：**获取当前登录用户信息

**URL：**/getUserMain

**接口类型：**GET

**Params：**{

userid : String

}

**响应格式：**JSON

**返回响应：**

状态码200 : {

user : {

userid : String

avator : String

user\_name : String

vip\_level : String

vip\_count : String

isroot : String

aboutme : String

sex : Integer

date : [“”, “”, “”]

time : [“”, “”, “”]

address : [

{

isdefault : Boolean

id : String

name : String

address : String

phone : String

}

]

}

}

**参数说明：**

userid : 用户ID

avator : 用户头像

user\_name : 用户名

vip\_level : 会员等级

vip\_count : 用户积分

isroot : 是否为管理员账户

aboutme : 个人介绍

sex : 性别

date : 生日

time : 注册时间

address : 收货地址

address. isdefault : 是否为默认地址

address. id : 地址ID

address. name : 收货人

address. address : 详细地址

address. phone : 联系电话

**接口功能：**

收到请求后，遍历USERTABLE表，将HASH字段与userid匹配，若无匹配，返回404，若匹配成功，读取该记录的ID、AVATOR、NICKNAME、VIPLEVEL、VIPCOUNT、ISROOT、ABOUTME、SEX、DATE、TIME、DEFAULTADDRESS字段。

将DATE和TIME字段数据分成年月日三段存为数组。

遍历USERADDRESSTABLE表，查找USERID字段与ID相同的记录，同时查找ID字段与DEFAULTADDRESS匹配的记录，读取这些记录的ID、NAME、ADDRESS、PHONE字段，将以上数据存为一个数组，并对ID字段与DEFAULTADDRESS匹配的记录数据标记为isdefault。

最后将所有数据存为一个JSON对象，按如下格式返回：

{

user: {

        userid: ID,

        avator: AVATOR,

        user\_name: NICKNAME,

        vip\_level: VIPLEVEL,

        vip\_count: VIPCOUNT,

        isroot:ISROOT

        aboutme: ABOUTME,

        sex: SEX,

        date: [YEAR,MONTH,DAY],

        time: [YEAR,MONTH,DAY],

        address: [

            {

                isdefault: isdefault(Boolean),

                id: ID(USERADDRESSTABLE),

                name: NAME(USERADDRESSTABLE),

                address: ADDRESS(USERADDRESSTABLE),

                phone: PHONE(USERADDRESSTABLE)

            }

        ]

    }

}

1. **获取用户注册数量**

**接口说明：**获取今年每月每日用户注册数量

**URL：**/getUserNumber

**接口类型：**GET

**Params：**无

**响应格式：**JSON

**返回响应：**

状态码200 : {

usernumber : {

time : String

total : String

month : [

{

time : String

total : String

dayly : [

{

time : String

total : String

}

]

}

]

}

total : String

}

**参数说明：**

time : 时间，年：“2024”，月：“2024-1”，日：“2024-1-1”

total : 当前年/月/日用户注册总量

month : 月度数据

dayly : 日度数据

**接口功能：**

收到请求后遍历USERTABLE表，将所有数据数量存为total，同时将TIME字段数据与当前年份匹配，若成功匹配，将成功匹配的记录数量存为yeartotal，并将以上数据按月份分组存为一个数组month，记录每个月份的数据数量存为month[].total，对每个月份的数据按日分组存为一个数组dayly，记录每日数据数量存为month[].dayly[].total。

最后将数据按以下格式返回：

{

usernumber: [

        {

            time: 当前年份,

            total: yeartotal,

            month: [

                {

                    time: 当前年份-月份,

                    total: month[].total,

                    dayly: [

                        {

                            time: 当前年份-月份-日,

                            total: month[].dayly[].total

                        }

                    ]

                }

            ]

        }

    total: total

}

1. **提交用户设置**

**接口说明：**用于设置页面提交

**URL：**/postUserInfo

**接口类型：**POST

**Params：**{

userid : String

avator : String

user\_name : String

aboutme : String

sex : Integer

date : [String, String, String]

address : [

{

}

]

}

**响应格式：**JSON

**返回响应：**

状态码200 : {

usernumber : {

successful : Boolean

}

**参数说明：**

time : 时间，年：“2024”，月：“2024-1”，日：“2024-1-1”

total : 当前年/月/日用户注册总量

month : 月度数据

dayly : 日度数据

**接口功能：**