IONCHAIN离子链版权所有

**钱包接口规范文档**

版本号：v1.0.0



**版本控制信息**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修改时间** | **主要内容** | **作者** |
| v1.0.0 | 209年06月1日 | 初始文档 | Yves |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目 录**

一、 文档说明 1

1． 功能描述 1

2． 阅读对象 1

3． 技术服务 1

二、 接口调用流程 2

三、 API协议说明 2

1． 协议规定 2

2． 请求参数 3

3． 请求参数说明 3

四、 API接口定义 3

A附录 4

A.1 签名机制 4

A.2 错误码 4

# 文档说明

## 功能描述

本文档的目的是提供 api接口规范文档开发中涉及的第三方（APP、H5以及HttpClient等方式）与之间的接口规范，以便于开发人员在统一规范下的开发及相互间的对接。

## 阅读对象

系统设计人员、产品开发人员及测试人员。该人员需要具备一定的网站开发能力，了解JAVA、PHP、JAVASCRIPT等开发语言，并对网络通信协议、信息安全等有一定的认识。

## 技术服务

如果您在开发过程中遇到任何技术问题，可以添加QQ群已得到技术支持。

作者：IONCHAIN

网址：<http://www.ionchain.org>

## 钱包服务端源码github地址

<https://github.com/ionchain/ionchain-wallet-server>

readme文件详细技术了接口的请求细则。

# 接口调用流程



# API协议说明

## 协议规定

|  |  |
| --- | --- |
| 传输方式 | 采用HTTP传输 |
| 提交方式 | 数据以GET/POST方式提交 |
| 数据格式 | 应答数据格式采用JSON格式 |
| 字符编码 | 统一采用UTF-8字符编码 |
| 请求URL | ${hostname}/api |
| 签名算法 | 签名算法采用MD5，具体见附录 |
| 签名要求 | 请求数据需要校验签名 |
| 证书要求 | 不使用证书 |
| 秘钥 | 开发前由系统管理员提供 |

## 请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 变量名 | 类型 | 示例值 | 说明 | **必填** |
| 版本号 | version | String | 1.0.0 | 接口协议的版本号 | 是 |
| 接口码 | apiNo | String | 123123123123 | 用于匹配输入输出 | 是 |
| 方法名 | method | String | pageArticle | 调用方法名称 | 是 |
| 用户名 | apiUser | String | api1 | 该请求的处理方法 | 是 |
| 时间戳 | time | String | 20170314150605 | 格式为：年月日时分秒 | 是 |
| 校验和 | checkSum | String | aaaaaaaaaaaaaaaa | 校验和用于验证参数是否被修改过 | 是 |
| 页数 | pageNo | Integer | 1 | 如果需要分页，必传 | 否 |
| 页大小 | pageSize | Integer | 20 | 如果需要分页，必传 | 否 |
| 参数 | p | String | {folderId:2,name:“文章”} | 协议相关参数，如果没有传空 | 是 |

## 请求参数说明

1. 版本号：会随着接口及文档的升级有所调整。
2. 接口码：每次请求需要生成唯一的接口码，用于匹配输入输出。
3. 方法名：每个接口都会存一个自己唯一的方法名。
4. 用户名：每个接入系统都会有自己唯一的用户名，具体获取方式联系系统管理员。
5. 时间戳：格式为年月日时分秒，每次接口调用请重新生成。
6. 校验和：具体生成规则见附录。
7. 参数：具体参数见相关接口。

参数加密，需要先进行base64加密，再进行URL编码，编码格式使用utf-8

加密示例：URLEncoder.encode(Base64.encode(p),"UTF-8");

解密示例：URLDecoder.decode(Base64.decode(p),"UTF-8");

特别说明：接口请求：先计算checkSum再进行p参数加密。

接口应答：先进行p参数解密再进行checkSum校验。

# API接口定义

# A附录

## A.1 签名机制

checkSum校验和：先通过登录接口获取key，然后进行报文的签名生成；

报文的签名生成步骤如下：

1. 对报文中出现签名域（sign）之外的所有数据元采用key=value的形式按照参数名ASCII码从小到大排序（字典序），然后以&作为连接符拼接成待签名串（key1=value1&key2=value2…）stringA。
2. 在stringA最后拼接上key得到stringB字符串，并对stringB进行MD5运算，再将得到的字符串所有字符转换为大写，得到sign值。

签名验证方法：  
 根据前面描述的签名参数sign生成的方法规则，计算得到参数的签名值，和参数中通知过来的sign对应的参数值进行对比，如果是一致的，那么就校验通过，如果不一致，说明参数被修改过。

示例：

假设传送的参数如下：

*version：1.0.0*

*method：getDataLebal*

*p：{a:1,b:"你好"}*

第一步：对参数按照key=value的格式，并按照参数名ASCII字典序排序如下：

*stringA=" method= getDataLebal&p={a:1,b:"你好"}&version=1.0.0";*

第二步：拼接API密钥：

*string signTemp="stringA&key=31650ba948d0103fe71353920fcb53b1"*

第三步：MD5加密：

*sign=MD5(signTemp).toUpperCase()="DD8E263079FCC6503A9751AD936095BE"。*

注意：密钥请妥善保管。

## A.2 错误码

通用错误码