

接口文档

目录

1.	开放平台接口介绍	3
1.1.	获取唯一标识码接口	3
1.1.1.	应用场景	3
1.1.2.	请求格式	3
1.1.3.	返回格式	3
1.1.4.	请求样例	3
1.2.	批量获取唯一标识码接口	4
1.2.1.	应用场景	4
1.2.2.	请求格式	4
1.2.3.	返回格式	4
1.2.4.	请求样例	4
1.3.	单条数据上链接口	5
1.3.1.	应用场景	5
1.3.2.	请求格式	5
1.3.3.	返回格式	5
1.3.4.	请求样例	5
1.4.	批量数据上链接口	6
1.4.1.	应用场景	6
1.4.2.	请求格式	6
1.4.3.	返回格式	7
1.4.4.	请求样例	7
1.5.	哈希值信息单条查询接口	8
1.5.1.	应用场景	8
1.5.2.	请求格式	8
1.5.3.	返回格式	9
1.5.4.	请求样例	9
1.6.	唯一标识码信息单条查询接口	10
1.6.1.	应用场景	10
1.6.2.	请求格式	10
1.6.3.	返回格式	10
1.6.4.	请求样例	11
1.7.	唯一标识码信息批量查询接口	12
1.7.1.	应用场景	12
1.7.2.	请求格式	12
1.7.3.	返回格式	12
1.7.4.	请求样例	14
2.	签名校验说明	14
2.1.	需要参与签名的参数	14
2.2.	生成待签名串	14
2.3.	令牌和密钥说明	15
2.4.	加签说明	15

2.5. 验签说明.....	15
----------------	----

1. 开放平台接口介绍

1.1. 获取唯一标识码接口

1.1.1. 应用场景

获取一条唯一标识码,注册到平台,并将该唯一标识 Id 返回给调用者。

1.1.2. 请求格式

请求参数:

名称	类型	含义	是否必要	备注
credential	String	令牌	是	请求参数

1.1.3. 返回格式

返回参数:

字段名	含义	类型	是否必要	备注
code	错误代码	String	是	成功: 0000 失败 9999
message	错误信息	String	是	成功: 成功 失败: 失败+ 原因
msgBody	唯一标识码	String	是	唯一标识码

1.1.4. 请求样例

请求类型: POST

请求地址: <http://ip:port/upload/getProductId>

请求参数: credential=042b487466amad4934f0231b7dc4ece2

返回参数:

```
{“code”:”0000”,”message”:”成功”,”msgBody”: ”10851593760873657”}
```

1.2. 批量获取唯一标识码接口

1.2.1. 应用场景

批量获取多条唯一标识码,注册到后台,并将获取的多条唯一标识码返回给调用者。

1.2.2. 请求格式

请求参数:

名称	类型	含义	是否必要	备注
credential	String	令牌	是	
num	int	数量	是	获取数量 (≤ 1000)

1.2.3. 返回格式

返回参数:

字段名	含义	类型	是否必要	备注
code	错误代码	String	是	成功: 0000 失败 9999
message	错误信息	String	是	成功: 成功
msgBody	返回结果集	List<String>	是	

1.2.4. 请求样例

请求类型: POST

请求地址: `http://ip:port/upload/getProductIdList`

请求参数: `num=3`

返回参数:

```
{“code”:”0000”,”message”:”成功”,”msgBody”:[“78841593769153312”,”73261593769153312”,”49701593769153312”]}
```

12”]}}

1.3. 单条数据上链接口

1.3.1. 应用场景

将数据保存到区块链上,并将保存返回的交易哈希保存到后台,并将交易哈希返回给调用者。

1.3.2. 请求格式

请求参数:

字段名	含义	类型	是否必要	备注
credential	令牌	String	是	平台唯一
productId	唯一标识码	String	是	
depositMessage	上链数据	String	是	

1.3.3. 返回格式

返回参数:

字段名	含义	类型	是否必要	备注
code	错误代码	String	是	成功: 0000 失败 9999
message	错误信息	String	是	成功: ”成功” 失败: ”失败”+ 错误原因
msgBody	返回结果集	String	是	单条数据 hash 值

1.3.4. 请求样例

请求类型: POST

请求地址: <http://ip:port/upload/uploadInfo>

请求参数:

credential=042b487466amad4934f0231b7dc4ece2&productId=74001593744686777
&depositMessage=这是一条测试数据”

返回参数:

成功:

{“code”:”0000”,”message”:”成功”,”msgBody”: ”8124167d0c5fec756c6df89879acbcd6c3fa9e49370b75d5c0a8caad4890c630”}

失败:

{“code”:”9999”,”message”:”产品唯一标识未注册! ”}

1.4. 批量数据上链接口

1.4.1. 应用场景

将多条不同数据保存到区块链上,并将保存返回的交易哈希保存到后台,并将交易哈希返回给调用者。

1.4.2. 请求格式

请求参数:

字段名	含义	类型	是否必要	备注
credential	令牌	String	是	平台唯一
uploadBodyList	上链数据	List<uploadBody>	是	

uploadBody 结构说明

字段名	含义	类型	是否必要	备注
productId	唯一标识码	String	是	
depositMessage	消息体	String	是	上传数据
distinctId	数据 id	String	是	不允许重复

1.4.3. 返回格式

返回参数：

字段名	含义	类型	是否必要	备注
code	错误代码	String	是	成功：0000 失败 9999 部分成功： 1111
message	错误信息	String	是	成功：成功 失败：失败+ 错误信息 部分成功：部 分成功+失败 productId:+错 误信息
msgBody	返回结果集	List<uploadResult>	是	

uploadResult 结构说明

字段名	含义	类型	是否必要	备注
productId	唯一标识吗	String	是	
txhash	哈希值	String	是	
code	错误代码	Int	是	成功：0 失败： 1
message	成功或失败信 息	String	是	成功或失败原 因
distinctId	数据 id	String	是	不允许重复

1.4.4. 请求样例

请求类型：POST

请求地址：http://ip:port/upload/asynUploadInfoList

请求参数:

```
“[{“depositMessage”: {“createId”:100,”createName”:“管理员”,...},”productId”:“47171593653291948”},{“depositMessage”: {“createId”:100,”createName”:“管理员”,...},”productId”:“47171593653291948”}]”
```

返回参数:

成功:

```
{“code”:“0000”,“message”:“成功”,“msgBody”: [{“code”:“0”,“message”:“成功”,“txhash”:“f8f78a2588473c5195a9043c6666b914bb806f0d9f9869b180cf4e53f9b980ec”,“productId”:“74001593744686777”,“distinctId”:“123”},{“code”:“0”,“message”:“成功”,“txhash”:“5107f83d718fbc5f618a8d804c6d1004b4bf4d305350fa08a7813285a01af2a3”,“productId”:“18901593744686777”,“distinctId”:“234”}]}
```

失败:

```
{“code”:“9999”,“message”:“第 2 条数据异常:productId:47171593653291948 未注册;”}  
{“code”:“9999”,“message”:“第 1 条数据异常:数据 id 为空,第 2 条数据异常:productId:47171593653291948 未注册;”}
```

1.5. 哈希值信息单条查询接口

1.5.1. 应用场景

根据之前上链成功返回的交易哈希来查询该交易哈希在区块链上保存的数据信息。并将信息返回。

1.5.2. 请求格式

请求参数:

字段名	含义	类型	是否必要	备注
credential	令牌	String	是	平台唯一
txHash	交易哈希值	String	是	交易哈希值

1.5.3. 返回格式

返回参数：

字段名	含义	类型	是否必要	备注
code	错误代码	String	是	成功：0000 失败 9999
message	错误信息	String	是	成功：“成功” 失败：“失败”+ 错误原因
msgBody	链上信息	depositInfo	是	链上信息

depositInfo 结构说明

字段名	含义	类型	是否必要	备注
code	错误代码	int	是	成功：0 失败：1
message	成功或失败信息	String	是	成功或失败原因
msgBody	链上数据	String	是	区块链上保存数据
blockNum	区块高度	String	是	区块高度
blockCreateTime	区块创建时间	String	是	区块创建时间
blockHash	区块哈希	String	是	区块哈希值
uploadTime	上链时间	String	是	上链时间
txhash	哈希值	Strign	是	交易哈希值

1.5.4. 请求样例

请求类型：POST

请求地址：http://ip:port/search/getMsgByHash

请求参数：

credential=042b487466amad4934f0231b7dc4ece2&txHash=dd66ee97f93c5d178656c

4e41943cdefeff25546a717251c4c18cd4e6f1af9d1

返回参数:

成功:

```
{“code”: “0000”, “message”: “成功”, “msgBody”: {“code”: “0”, “message”: “成功”, “msgBody”: {“createId”: 100, “createName”: “管理员”, ...}, “blockNum”: “2323187”, “blockCreateTime”: “2020-07-03 10:53:48”, “blockHash”: “ee712c8365885410c374ea44a4ff6cb6257a388fb566030774390e64cea4ec0c”, “txhash”: “dd66ee97f93c5d178656c4e41943cdefeff25546a717251c4c18cd4e6f1af9d1”}}
```

失败:

```
{“code”: “9999”, “message”: “哈希查询链上信息失败!”, “msgBody”: {“code”: “1”, “message”: “下链失败: 代码=404 原因=null”}}
```

1.6. 唯一标识码信息单条查询接口

1.6.1. 应用场景

根据唯一标识码(ProductId)查询在区块链上与该标识码相关的所有数据信息。并将查询到的信息返回。

1.6.2. 请求格式

请求参数:

字段名	含义	类型	是否必要	备注
credential	令牌	String	是	平台唯一
productId	唯一标识码	String	是	唯一标识码

1.6.3. 返回格式

返回参数:

字段名	含义	类型	是否必要	备注
code	错误代码	String	是	成功: 0000

				失败 9999
message	错误信息	String	是	成功: "成功" 失败: "失败"+ 错误原因
msgBody	存证数据集合	list<depositInfo>	是	链上数据集合

depositInfo 结构说明

字段名	含义	类型	是否必要	备注
code	错误代码	int	是	成功:0 失败:1
message	成功或失败信息	String	是	成功或失败原因
msgBody	链上数据	String	是	区块链上保存数据
blockNum	区块高度	String	是	区块高度
blockCreateTime	区块创建时间	String	是	区块创建时间
blockHash	区块哈希	String	是	区块哈希值
uploadTime	上链时间	String	是	上链时间
txhash	哈希值	String	是	交易哈希值

1.6.4. 请求样例

请求类型: POST

请求地址: <http://ip:port/search/getMsgById>

请求参数:

productId=74001593744686777&credential=042b487466amad4934f0231b7dc4ece2

返回参数:

成功:

{“code”: “0000”, “message”: “成功”, “msgBody”: [{“message”: “成

功”,”msgBody”: “{“createId”:100,”createName”:”管理员”,...}”,”blockNum”:
 “2323187”,”blockCreateTime”: “2020-07-03
 10:53:48”,”blockHash”:”ee712c8365885410c374ea44a4ff6cb6257a388fb566030774
 390e64cea4ec0c”,”txhash”:”dd66ee97f93c5d178656c4e41943cdefeff25546a717251c
 4c18cd4e6f1af9d1”}}, {...}, {...}]]

失败:

{“code”:”9999”,”message”:”产品唯一标识未注册! ”}

1.7. 唯一标识码信息批量查询接口

1.7.1. 应用场景

根据唯一标识码批量查询在区块链上与这些标识码相关的所有数据信息。并将查询到的信息返回。

1.7.2. 请求格式

请求参数:

字段名	含义	类型	是否必要	备注
credential	令牌	String	是	平台唯一
productIdList	唯一标识码集合	List<String>	是	唯一标识码集合

1.7.3. 返回格式

返回参数:

字段名	含义	类型	是否必要	备注
code	错误代码	String	是	成功: 0000 失败 9999 部分成功: 1111
message	错误信息	String	是	成功: ”成功”

				失败：”失败”+错误原因 部分成功：”部分成功”+失败原因 map
msgBody	返回结果集	List<ProductInfo>	是	产品信息集合

productInfo 结构说明

字段名	含义	类型	是否必要	备注
productId	唯一标识码	productid	是	
depositList	存证数据集合	List<depositInfo>	是	链上数据集合

depositInfo 结构说明

字段名	含义	类型	是否必要	备注
code	错误代码	int	是	成功:0 失败:1
message	成功或失败信息	String	是	成功或失败原因
msgBody	链上数据	String	是	区块链上保存数据
blockNum	区块高度	String	是	区块高度
blockCreateTime	区块创建时间	String	是	区块创建时间
blockHash	区块哈希	String	是	区块哈希值
uploadTime	上链时间	String	是	上链时间
txhash	哈希值	String	是	交易哈希值

1.7.4. 请求样例

请求类型: POST

请求地址: `http://ip:port/search/getMsgByIdList`

请求参数:

`credential=042b487466amad4934f0231b7dc4ece2&ploadBody=[74001593744686777,18901593744686777]`

返回参数:

`{“code”: “0000”, “message”: “成功”, msgBody: “[{“productId”: “74001593744686777”, “depostList”: “[{...}, {...}]”}, {“productId”: “18901593744686777”, “depostList”: “[{...}, {...}, {...}]”}]”}`

2. 签名校验说明

2.1. 需要参与签名的参数

在请求参数列表中,除去 `signature` 参数外,其他需要使用到的参数皆是要签名的参数。

2.2. 生成待签名串

对于如下的业务参数集合:

```
Map<String,String> parameters={  
    “orgId=1002”,  
    “userId=100103”,  
    “credential=042b487498fbad4934f0231b7dc4ece3”  
};
```

对集合里的每一个键值对从 `a` 到 `z` 的顺序排序,若遇到相同首字母,则看第二个字母,以此类推。`signature` 不参与签名

排序完成之后,再把所有数组值以“&”字符连接起来,如:

`credential=042b487498fbad4934f0231b7dc4ece3& orgId =1002& userId =100103` 这

串字符串便是待签名串。

注意:

- 没有值的参数无需传递, 传递需包含到待签名数据中;
- 根据 HTTP 协议要求, 传递参数的值中如果存在特殊字符 (如: &、@等), 那么该值需要做 URL Encoding, 这样请求接收方才能接收到正确的参数值。这种情况下, 待签名数据应该是原生值而不是 encoding 之后的值。例如: 调用某接口需要对请求参数 media_id 进行数字签名, 那么待签名数据应该是 media_id=test@msn.com, 而不是 media_id=test%40msn.com。

2.3. 令牌和密钥说明

用户注册以后,点击申请令牌,平台会分配给用户一个令牌和令牌对应的密钥也就是 credential 和 secretkey 这两个参数。此令牌和密钥是与平台进行通信的通信密钥。将会参与签名摘要计算。

2.4. 加签说明

加签是对整个待签名串、令牌以及令牌对应的密钥进行 MD5,把这个 MD5 生成的结果当作参数 signature 的值。在加签过程中把所有的键对应值都 encode 并且把 signature 拼接到请求参数的最后面。

示例如下:

```
credential=MDQyYjQ4NzQ5OGZiYWQ0OTM0ZjAyMzFiN2RjNGVjZTM=&
orgId=MTAwMg==&userId=MTAwMTAz&signature=924fef8780f95b5c5411b5981f
7dd586
```

加签方法: public static String createSign

输入参数:

- 1、待签名字符串
- 2、平台分配给用户令牌以及令牌对应的密钥

输出参数:摘要串.

2.5. 验签说明

当接收到请求参数以后,先把 signature 进行分离出来保存用作对比,然后会根

据参数上的令牌去数据库里查找和令牌对应的密钥,将截取后的参数串进行 decode 然后和令牌以及密钥进行拼接做 MD5。将得到的结果与 signature 的值进行比较,实现签名校验。

输入参数:

- 1.待签名字符串
- 2.平台分配给用户的令牌和令牌对应的密钥
- 3.接收到的摘要串

输出参数: 无输出参数,验证成功放行,失败拦截。