

# 1. Log in

**Log in** 模块下的 API 接口用于管理用户的信息，用户需要使用 BTC Taproot 地址进行登录。该模块下使用统一的地址：<http://119.147.213.61:38080>。

## 1.1 获取挑战信息

challenge 接口会获取需要用户签名的挑战信息。

**注意：**调用 challenge 接口时，需要在 headers 的 Origin 字段中指定域名，例如域名 <http://memo.io>。

请求地址

```
/v1/btc/challenge
```

请求方式

GET

Header 参数

属性名	必需?	类型	描述
Origin	yes	string	网站域名

Path 参数

无

Query 参数

属性名	必需?	类型	描述
address	yes	string	BTC 地址

Body 参数

无

返回结果

属性名	类型	描述
message	string	挑战信息

请求示例

返回示例

## 1.2 登录

login 接口可以用于登录服务器，从而获取访问文档中所有 POST 接口的权限。

**注意：**在调用 challenge 接口获取挑战信息后，需要在 30s 内完成登录，否则登录失败。

请求地址

```
/v1/btc/login
```

请求方式

POST

**Header 参数**

无

**Path 参数**

无

**Query 参数**

无

**Body 参数**

属性名	必需?	类型	描述
message	yes	string	挑战信息
signature	yes	string	用户私钥对挑战信息的签名

签名的生成方式可以参考[文档](#)，也可以使用 BTC 的 signmessage 接口签名。

返回结果

属性名	类型	描述
accessToken	string	访问令牌

refreshToken	string	刷新令牌
--------------	--------	------

请求示例

返回示例

## 1.2 刷新令牌

refresh 接口可以用于刷新访问令牌，获取全新的访问令牌。

请求地址

```
/v1/btc/refresh
```

请求方式

POST

Header 参数

属性名	值
Authorization	Bearer YOUR_REFRESH_TOKEN

Path 参数

无

Query 参数

无

Body 参数

无

返回结果

属性名	类型	描述
accessToken	string	访问令牌

请求示例

返回示例

## 2. Inscriptions

**Inscriptions** 模块下 API 接口用于查询链上 inscription 信息。该模块下使用统一的地址：  
http://119.147.213.61:38080。

### 2.1 获取铭文信息

该接口将通过铭文 ID 索引到具体的铭文，并返回铭文的信息。

请求地址

```
/v1/inscription/{inscriptionId}/info
```

请求方式

GET

**Header 参数**

无

**Path 参数**

属性名	类型	描述
inscripitionId	string	铭文唯一 ID

**Query 参数**

无

**Body 参数**

无

返回结果

属性名	类型	描述
utxoInfo	Object	铭文所属 UTXO 信息
inscriptionInfo	Object	铭文的基本信息

utxoInfo 的结构

属性名	类型	描述
txid	string	UTXO 的交易 ID
vout	integer	UTXO 的交易偏移量
satoshi	integer	UTXO 的余额
scriptPk	string	UTXO 的锁定脚本
address	string	UTXO 的持有者地址

inscriptionInfo 的结构

属性名	类型	描述
inscriptionId	string	铭文的唯一 ID
commitTransactionHash	string	commit 交易哈希
revealTransactionHash	string	reveal 交易哈希
blockHeight	integer	包含 reveal 交易的区块高度
owner	string	铭文的拥有者

请求示例

返回示例

2.2 获取铭文内容

该接口将通过铭文 ID 索引到具体的铭文，并返回铭文的具体内容。

请求地址

```
/v1/inscription/{inscriptionId}/content
```

请求方式

GET

**Header 参数**

无

**Path 参数**

属性名	类型	描述
inscripitionId	string	铭文唯一 ID

**Query 参数**

无

**Body 参数**

无

返回结果 (string)

属性名	类型	描述
content	string	铭文内容

请求示例

返回示例

## 2.3 获取地址持有的铭文

该接口将获取地址持有的所有铭文信息。

请求地址

/v1/address/{address}/inscription/info

## 请求方式

GET

## Header 参数

无

## Path 参数

属性名	类型	描述
address	string	BTC 地址

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

## Body 参数

无

## 返回结果

属性名	类型	描述
infos	Array<Object>	地址持有的铭文信息
total	integer	地址持有的铭文总数

## infos 的结构

属性名	类型	描述
utxoInfo	Object	铭文所属 UTXO 信息
inscriptionInfo	Object	铭文的基本信息

## utxoInfo 的结构

属性名	类型	描述
-----	----	----

inscriptionId	string	铭文的唯一 ID
commitTransactionHash	string	commit 交易哈希
revealTransactionHash	string	reveal 交易哈希
blockHeight	integer	包含 reveal 交易的区块高度
owner	string	铭文的拥有者

请求示例

返回示例

## 2.4 获取 UTXO 持有的铭文

该接口将根据 UTXO 获取铭文信息。

请求地址



/v1/output/{output}/inscription/info

请求方式

GET

Header 参数

无

Path 参数

属性名	类型	描述
output	string	表示为{txid:vout}，例如 {cd8e555a9d988224f5de1a52ad17f988c65b15086b11ff5ebdf12bf5ccaa3246:0}

Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

Body 参数

无

返回结果

属性名	类型	描述
utxoInfo	Object	铭文所属 UTXO 信息
inscriptionInfo	Object	铭文的基本信息

utxoInfo 的结构

属性名	类型	描述
txid	string	UTXO 的交易 ID
vout	integer	UTXO 的交易偏移量
satoshi	integer	UTXO 的余额

## 3. Fungible Token

**Fungible Token** 模块下的 API 接口用于管理用户持有的代币。该模块下使用统一的地址：  
<http://119.147.213.61:38080>。

### 3.1 获取已部署 FT 列表

list 接口会显示 BTC 链上已有的全部代币信息。

请求地址

```
/v1/ft/list
```

请求方式

GET

Header 参数

无

Path 参数

无

Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

Body 参数

无

返回结果

属性名	类型	描述
ftInfos	Array<Object>	在 BTC 上部署的全部 FT 信息
total	integer	FT 集合总数

ftInfos 的结构

属性名	类型	描述
tick	string	FT 标识符
max	integer	FT 最大供应量
lim	integer	FT 最小铸造量
minted	integer	FT 总铸币量

请求示例

返回示例

## 3.2 获取地址的余额

balance 接口会显示账户的余额，包括可转移余额以及可用余额。需要您提供 BTC 地址以及 token 标识符。

请求地址

```
/v1/address/{address}/ft/{tick}/balance
```

请求方式

GET

Header 参数

无

Path 参数

属性名	类型	描述
address	string	BTC 地址
tick	string	FT 标识符

Query 参数

无

Body 参数

无

返回参数

属性名	类型	描述
tick	string	FT 标识符
address	string	地址
overallBalance	integer	总余额
transferableBalance	integer	可转移余额
availableBalance	integer	可用余额

请求示例

返回示例

### 3.3 创建 Deploy 铭文

deploy 接口会由服务器代为创建 Deploy 铭文，从而发行全新的代币。

/v1/ft/deploy

## 请求方式

POST

## Header 参数

属性名	变量
Content-Type	application/json
Authorization	Bearer YOUR_ACCESS_TOKEN

## Path 参数

无

## Query 参数

无

## Body 参数

属性名	类型	描述
tick	string	token 名
max	integer	最大供应量
lim	integer	最小铸造量

## 返回结果

属性名	类型	描述
inscriptionId	string	Deploy 铭文的唯一 ID
commitTransactionHash	string	commit 交易哈希
revealTransactionHash	string	reveal 交易哈希
blockHeight	integer	包含 reveal 交易的区块高度
owner	string	Deploy 铭文的拥有者

## 请求示例

## 返回示例

/v1/ft/mint

请求方式

POST

Header 参数

属性名	变量
Content-Type	application/json
Authorization	Bearer YOUR_ACCESS_TOKEN

Path 参数

无

Query 参数

无

Body 参数

属性名	类型	描述
tick	string	token 名
amount	integer	铸造的代币数量

返回结果

属性名	类型	描述
inscriptionId	string	Mint 铭文的唯一 ID
commitTransactionHash	string	commit 交易哈希
revealTransactionHash	string	reveal 交易哈希
blockHeight	integer	包含 reveal 交易的区块高度
owner	string	Mint 铭文的拥有者

请求示例

返回示例

/v1/ft/transfer

请求方式

POST

Header 参数

属性名	变量
Content-Type	application/json
Authorization	Bearer YOUR_ACCESS_TOKEN

Path 参数

无

Query 参数

无

Body 参数

属性名	类型	描述
tick	string	token 标识符
amount	integer	余额

返回参数

属性名	类型	描述
inscriptionId	string	Mint 铭文的唯一 ID
commitTransactionHash	string	commit 交易哈希
revealTransactionHash	string	reveal 交易哈希
blockHeight	integer	包含 reveal 交易的区块高度
owner	string	Transfer 铭文的拥有者

请求示例

返回示例

## 4.1 获取 DA 列表

list 接口会显示部署在 BTC 链上全部 DA 系统信息。

请求地址

```
/v1/da/list
```

请求方式

GET

Header 参数

无

Path 参数

无

Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

Body 参数

无

返回参数

属性名	类型	描述
daInfos	Array<Object>	DA 系统的集合信息
total	integer	DA 系统总数

daInfos 结构

属性名	类型	描述
tick	string	DA 标识符
storage	string	存储地址（定期提交可用性证明）
foundation	string	基金会地址（收取证明失败时收益）



interval	integer	证明提交周期时间（单位为秒）
token	string	DA 接收的代币
price	integer	上传数据至 DA 需要支付的费用

请求示例

返回示例

## 4.2 获取地址持有的数据列表

hold 接口会显示给定地址持有的所有数据信息。

请求地址

```
/v1/address/{address}/da/{tick}/hold
```

## 请求方式

GET

## Header 参数

无

## Path 参数

属性名	类型	描述
address	integer	BTC 地址
tick	integer	DA 标识符

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

## Body 参数

无

## 返回结果

属性名	类型	描述
infos	Array<Object>	持有的数据信息集合
total	integer	持有的数据总数

## infos 的结构

属性名	类型	描述
tick	string	数据所在 DA 的标识符
id	string	数据承诺值
inscriptionId	string	Upload 铭文的唯一 ID

## 请求示例

/v1/da/upload/create

请求方式

POST

Header 参数

属性名	变量
Content-Type	application/json
Authorization	Bearer YOUR_ACCESS_TOKEN

Path 参数

无

Query 参数

无

Body 参数

属性名	类型	描述
data	string	数据的 base64 编码值

返回参数

属性名	类型	描述
id	string	数据承诺值

请求示例

返回示例

## 4.4 确认上传数据请求

该接口会将服务器上的数据上传至 DA。

请求地址

/v1/da/upload/confirm

## 请求方式

POST

## Header 参数

属性名	变量
Content-Type	application/json
Authorization	Bearer YOUR_ACCESS_TOKEN

## Path 参数

无

## Query 参数

无

## Body 参数

属性名	类型	描述
id	string	数据的承诺值
signature	string	用户私钥对 id 的签名，签名将会上链

签名的构建方式如下：

- 构建签名原始信息 **message**：按照字典序对 id 排序，并按照 **key=value** 的方式拼接。例如：**id=xxx**。
- 对原始信息 **message** 进行两次 SHA256 哈希，得到哈希值 **h**；
- 使用私钥对哈希值 **h** 使用 ECDSA secp256k1 签名，得到签名 **s**，将签名 **s** 进行 base64 编码得到 **signature**。

## 返回参数

属性名	类型	描述
inscriptionId	string	Upload 铭文的唯一 ID
commitTransactionHash	string	commit 交易哈希
revealTransactionHash	string	reveal 交易哈希

/v1/da/download

**请求方式**

GET

**Header 参数**

无

**Path 参数**

无

**Query 参数**

属性名	必需?	类型	描述
id	yes	string	数据的承诺值

**Body 参数**

无

**返回结果**

数据的二进制结果

**请求示例**

**返回示例**

## 5. NFT

### 5.1 获取已部署 NFT 集合列表

list 接口会返回所有部署在 BTC 链上所有 NFT 集合的信息。

**请求地址**

/v1/nft/list

## 请求方式

GET

## Path 参数

无

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

## Body 参数

无

## 返回参数

属性名	类型	描述
nftInfos	Array<Object>	NFT 部署信息集合
total	integer	已部署 NFT 集合总数

## nftInfos 的结构

属性名	类型	描述
tick	string	NFT 标识符
da	string	存储 NFT 数据内容的 DA 标识符
description	string	NFT 集合的描述
id	string	NFT 集合的描述图片数据承诺
deployer	string	deployer 地址

## 请求示例

## 返回示例

/v1/nft/offchain/list

## 请求方式

GET

## Path 参数

无

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

## Body 参数

无

## 返回参数

属性名	类型	描述
nftInfos	Array<Object>	未部署 NFT 集合信息
total	integer	未部署 NFT 集合总数

## nftInfos 的结构

属性名	类型	描述
tick	string	NFT 标识符
da	string	存储 NFT 数据内容的 DA 标识符
description	string	NFT 集合的描述
id	string	NFT 集合的描述图片数据承诺
deployer	string	deployer 地址
max	integer	允许发行的 NFT 总量
uploaded	integer	已上传的 NFT 数据的数量
createTime	integer	deploy 请求创建时间

## 请求示例

```
/v1/address/{address}/nft/{tick}/hold
```

## 请求方式

GET

## Path 参数

属性名	类型	描述
address	integer	BTC 地址
tick	integer	DA 标识符

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

## Body 参数

无

## 返回结果

属性名	类型	描述
nfts	Array<Object>	用户持有的 NFT 集合
total	integer	用户持有的 NFT 总数

## infos 的结构

属性名	类型	描述
tick	string	NFT 标识符
id	string	数据内容的承诺值
inscriptionId	string	Mint 铭文的唯一 ID

## 请求示例

## 返回示例



## 5.4 获取链下 NFT 内容信息列表

该接口会显示所有由 deployer 上传但还没被其他人 Mint 的 NFT 数据 ID 列表。

请求地址

```
/v1/nft/offchain/{tick}/content/list
```

请求方式

GET

Path 参数

属性名	类型	描述
tick	integer	DA 标识符

Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

Body 参数

无

返回结果

属性名	类型	描述
dataIds	Array<string>	NFT 数据内容 ID 集合
total	integer	未上链 NFT 数据内容总数

请求示例

返回示例

## 5.5 创建 deploy 铭文

创建 deploy 铭文分为 3 步：

1) 创建 deploy 请求，上传有关 deploy 铭文的相关信息，如使用的 da 信息，发行的 NFT 总数

量;

2) 批量上传数据, deployer 批量上传已审核的数据 (用于后续 mint nft);

3) 批量上传完成后, deployer 确认创建 deploy 铭文。

### 5.5.1 创建 deploy 请求

该接口会在服务器上创建新的 deploy 请求, 需要 deployer 上传全部的 NFT 数据, 才能完成 deploy 铭文的创建。

请求地址

```
/v1/nft/deploy/create
```

请求方式

POST

Header 参数

属性名	变量
Content-Type	application/json
Authorization	Bearer YOUR_ACCESS_TOKEN

Path 参数

无

Query 参数

无

Body 参数

属性名	类型	描述
tick	string	NFT 标识符
da	string	NFT 数据内容所在的 DA
description	string	NFT 集合的描述
data	string	NFT 集合的图片数据, base64 编码
max	integer	允许发行的 NFT 总量

返回结果

无

请求示例

返回示例

## 5.5.2 批量上传 NFT 数据

该接口可以批量上传 NFT 数据至服务器，后续铸造 NFT 时，使用 deployer 上传的数据铸造 NFT。

请求地址

```
/v1/nft/deploy/batch_upload
```

请求方式

POST

Header 参数

属性名	变量
Content-Type	multipart/form-data
Authorization	Bearer YOUR_ACCESS_TOKEN

Path 参数

无

Query 参数

无

Body 参数

属性名	类型	描述
tick	string	NFT 标识符
files	Array<file>	上传数据集合

返回结果

属性名	类型	描述
dataIds	Array<string>	数据承诺的集合

请求示例

返回示例

### 5.5.3 确认 deploy 请求

当 deployer 上传全部的 NFT 数据后，需要调用该接口已确认创建 deploy 铭文。

请求地址

```
/v1/nft/deploy/confirm
```

请求方式

POST

Header 参数

属性名	变量
Content-Type	application/json
Authorization	Bearer YOUR_ACCESS_TOKEN

Path 参数

无

Query 参数

无

Body 参数(form-data)

属性名	类型	描述
tick	string	NFT 标识符
id	string	NFT 集合的图片数据的承诺值
signature	string	用户私钥对 id 的签名，签名将会上链

签名的构建方式如下：

- 构建签名原始信息 **message**：按照字典序对 id 排序，并按照 **key=value** 的方式拼接。例如：**id=xxx**。
- 对原始信息 **message** 进行两次 SHA256 哈希，得到哈希值 **h**；
- 使用私钥对哈希值 **h** 使用 ECDSA secp256k1 签名，得到签名 **s**，将签名 **s** 进行 base64 编码得到 **signature**。

返回结果

属性名	类型	描述
inscriptionId	string	Mint 铭文的唯一 ID
commitTransactionHash	string	commit 交易哈希
revealTransactionHash	string	reveal 交易哈希
blockHeight	integer	包含 reveal 交易的区块高度
owner	string	Deploy 铭文的拥有者

请求示例

返回示例

## 5.6 创建 mint 铭文

mint 接口会铸造一枚 NFT 并发送到您的账户。注意，您在铸造 NFT 时隐含上传数据至 DA 中，因此，需要您支付一定的 token。您需要确保您的账户的余额足够，否则铸造操作会失败。铸造 NFT 后，您同样可以通过 DA 的 upload 接口下载数据。

请求地址

/v1/nft/mint

## 请求方式

POST

## Header 参数

属性名	变量
Content-Type	application/json
Authorization	Bearer YOUR_ACCESS_TOKEN

## Path 参数

无

## Query 参数

无

## Body 参数

属性名	类型	描述
tick	string	NFT 标识符
id	string	NFT 的数据内容的承诺值
signature	string	用户私钥对 id 的签名，签名将会上链

签名的构建方式如下：

- 构建签名原始信息 **message**：按照字典序对 id 排序，并按照 **key=value** 的方式拼接。例如：**id=xxx**。
- 对原始信息 **message** 进行两次 SHA256 哈希，得到哈希值 **h**；
- 使用私钥对哈希值 **h** 使用 ECDSA secp256k1 签名，得到签名 **s**，将签名 **s** 进行 base64 编码得到 **signature**。

## 返回结果

属性名	类型	描述
inscriptionId	string	Mint 铭文的唯一 ID
commitTransactionHash	string	commit 交易哈希
revealTransactionHash	string	reveal 交易哈希

/v1/market/ft/list

## 请求方式

GET

## Path 参数

无

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）
order	no	string	排序规则（默认是 volum_sec，按照成交总金额由高到低排序）

### 可选的 order 参数：

1. volum\_sec：按照成交金额由高到低排序；
2. volum\_dsec：按照成交金额由低到高排序；
3. listed\_sec：按照卖单数量由高到低排序；
4. listed\_desc：按照卖单数量由低到高排序；
5. price\_sec：按照地板价由高到低排序；
6. price\_desc：按照地板价由低到高排序；

## Body 参数

无

## 返回参数

属性名	类型	描述
ftInfos	Array<Object>	FT 部署信息集合
total	integer	FT 集合总数

### ftInfos 的结构

属性名	类型	描述
tick	string	FT 标识符

lim	integer	FT 最小铸造量
minted	integer	FT 总铸币量
listed	integer	卖单数量
transactions	integer	市场上成功交易数量
holders	integer	持有该 FT 的地址数量
volume	integer	成交金额
capitalization	integer	FT 总市值
floorPrice	integer	地板价 (sats/token)

请求示例

返回示例

### 6.1.2 获取所有卖单列表

该接口会返回所有已上架在 BRC-985 Market 中所有 FT 卖单信息。

请求地址



```
/v1/market/ft/{tick}/order/list
```

## 请求方式

GET

## Path 参数

属性名	类型	描述
tick	string	FT 标识符

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）
order	no	string	排序规则（默认是 price_dsec，按照每个 token 的平均价格由低到高排序）

## 可选的 order 参数：

1. price\_sec：按照 token 平均价格由高到低排序；
2. price\_dsec：按照 token 平均价格由低到高排序；
3. amount\_sec：按照出售 token 数量由高到低排序；
4. amount\_dsec：按照出售 token 数量由低到高排序；
5. recent：按照上架顺序排序；

## Body 参数

无

## 返回参数

属性名	类型	描述
orderInfos	Array<Object>	FT 卖单信息集合
total	integer	卖单总量

## orderInfos 的结构

属性名	类型	描述
tick	string	FT 标识符

```
/v1/market/address/{address}/ft/{tick}/order/list
```

## 请求方式

GET

## Path 参数

属性名	类型	描述
tick	string	FT 标识符

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）
order	no	string	排序规则（默认是 recent，按照上架顺序排序）

## 可选的 order 参数：

1. price\_sec：按照 token 平均价格由高到低排序；
2. price\_dsec：按照 token 平均价格由低到高排序；
3. amount\_sec：按照出售 token 数量由高到低排序；
4. amount\_dsec：按照出售 token 数量由低到高排序；
5. recent：按照上架顺序排序；

## Body 参数

无

## 返回参数

属性名	类型	描述
orderInfos	Array<Object>	FT 卖单信息集合
total	integer	地址创建的所有卖单总量

## orderInfos 的结构

属性名	类型	描述
tick	string	FT 标识符

orderId	string	FT 卖单 ID
inscriptionId	string	FT 转移铭文唯一 ID
seller	string	卖方地址
amount	integer	售卖数量
price	integer	售卖价格

请求示例

返回示例

### 6.1.4 获取所有 FT 交易历史

该接口会返回市场上已完成的所有交易信息。

请求地址

```
/v1/market/ft/{tick}/transaction/list
```

## 请求方式

GET

## Path 参数

属性名	类型	描述
tick	string	FT 标识符

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

## Body 参数

无

## 返回参数

属性名	类型	描述
transactionInfos	Array<Object>	FT 市场上已完成交易历史信息集合
total	integer	已完成历史交易总数

## transactionInfos 的结构

属性名	类型	描述
tick	string	FT 标识符
from	integer	交易卖方地址
to	integer	交易买方地址
amount	integer	交易 FT 数量
price	integer	交易价格
time	integer	交易完成时的时间戳
transactionId	string	交易 ID

```
/v1/market/address/{address}/ft/{tick}/transaction/list
```

### 请求方式

GET

### Path 参数

属性名	类型	描述
address	string	BTC 地址
tick	string	FT 标识符

### Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

### Body 参数

无

### 返回参数

属性名	类型	描述
transactionInfos	Array<Object>	FT 市场上已完成且与给定地址相关的交易历史信息集合
total	integer	已完成且与给定地址相关的交易总数

### transactionInfos 的结构

属性名	类型	描述
tick	string	FT 标识符
from	integer	交易卖方地址
to	integer	交易买方地址
amount	integer	交易 FT 数量
price	integer	交易价格

请求地址

/v1/market/ft/order/sell/create

## 请求方式

POST

## Path 参数

无

## Query 参数

无

## Body 参数

属性名	request?	type	description
inscriptionId	yes	string	FT 转移铭文的唯一 ID
price	yes	integer	FT 价格
receivedAddress	yes	string	接收 BTC 的地址

### 返回参数

属性名	类型	描述
orderId	string	FT 卖单 ID
psbt	string	未签名的 PSBT,base64 格式
signIndex	Array<integer>	指定 PSBT 中需要签名的输入位置

请求示例

## 返回示例

### 6.1.7 确认出售订单

由卖方对 PSBT 交易签名后, 再由卖方调用 `psbt::Finalize`

请求地址

/v1/market/ft/order/sell/confirm

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
orderId	yes	string	FT 卖单 id
psbt	yes	string	签名后 PSBT

返回参数

无

请求示例

返回示例

## 6.1.8 创建购买订单

该接口由买方调用，在服务器上创建买单，并匹配对应的卖单。服务器会返回给您 PSBT 交易并指示卖方需要再哪些交易输入中签名，请使用 SIGHASH\_ALL。

请求地址

```
/v1/market/ft/order/purchase/create
```

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
orderId	yes	string	FT 卖单 ID
paidAddress	yes	string	支付 BTC 的地址
receivedAddress	yes	string	接收 FT 的地址

返回参数

属性名	类型	描述
orderId	string	FT 卖单 ID
psbt	string	未签名的 PSBT,base64 格式
signIndex	Array<integer>	指定 PSBT 中需要签名的输入位置

请求示例

返回示例

### 6.1.9 确认购买订单

由买方对 PSBT 交易签名后，再由买方调用

请求地址



```
/v1/market/ft/order/purchase/confirm
```

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	requery?	type	description
orderId	yes	string	FT 卖单 id
psbt	yes	string	签名后 PSBT

返回参数

无

请求示例

返回示例

### 6.1.10 创建修改价格订单

该接口由卖方调用，可以修改在服务器上已创建的卖单的价格等□

请求地址

/v1/market/ft/order/modify/create

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
orderId	yes	string	FT 转移铭文 ID
newPrice	yes	integer	FT 的新价格
receivedAddress	yes	string	接收 BTC 的地址

返回参数

属性名	类型	描述
orderId	string	FT 卖单 ID
psbt	string	未签名的 PSBT,base64 格式
signIndex	Array<integer>	指定 PSBT 中需要签名的输入位置

请求示例

返回示例

### 6.1.11 确认修改价格订单

由卖方对 PSBT 交易签名后，再由卖方调用

请求地址

```
/v1/market/ft/order/modify/confirm
```

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
orderId	yes	string	FT 卖单 id
psbt	yes	string	签名后 PSBT

返回参数

无

请求示例

返回示例

## 6.1.12 创建取消出售订单

该接口由卖方调用，会取消已创建的出售订单。后续需要卖方签名 rawTransaction 从而将 Transfer 铭文转移给自己。

请求地址

/v1/market/ft/order/cancel/create

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
orderId	yes	string	FT 转移铭文 ID

返回参数

属性名	类型	描述
orderId	string	FT 卖单 ID
rawtransaction	string	未签名的转账交易,base64 格式

请求示例

返回示例

### 6.1.13 确认取消出售订单

该接口由卖方调用，将 rawTransaction 签名后发送给服务器。

请求地址

/v1/market/nft/list

## 请求方式

GET

## Path 参数

无

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量 (默认是 0)
end	no	integer	结束偏移量 (默认是 10)
order	no	string	排序规则 (默认是 volum_sec, 按照成交总金额由高到低排序)

### 可选的 order 参数:

1. volum\_sec: 按照成交金额由高到低排序;
2. volum\_dsec: 按照成交金额由低到高排序;
3. listed\_sec: 按照卖单数量由高到低排序;
4. listed\_desc: 按照卖单数量由低到高排序;
5. price\_sec: 按照地板价由高到低排序;
6. price\_desc: 按照地板价由低到高排序;

## Body 参数

无

## 返回参数

属性名	类型	描述
nftInfos	Array<Object>	NFT 部署信息集合
total	integer	NFT 集合总数

### nftInfos 的结构

属性名	类型	描述
tick	string	NFT 标识符

max	integer	NFT 最大供应量
minted	integer	NFT 总铸币量
listed	integer	卖单数量
transactions	integer	市场上成功交易数量
holders	integer	持有这类 NFT 的地址数量
volume	integer	成交总金额
floorPrice	integer	地板价 (sats/token)

请求示例

返回示例

## 6.2.2 获取卖单列表

list 接口会返回所有已上架在 BRC-985 Market 中所有 NFT 卖单信息。

请求地址

```
/v1/market/nft/{tick}/order/list
```

## 请求方式

GET

## Path 参数

属性名	类型	描述
tick	string	FT 标识符

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）
order	no	string	排序规则（默认是 price_desc, 按照 NFT 价格由低到高排序）

## 可选的 order 参数：

1. price\_desc：按照 token 平均价格由低到高排序；
2. price\_sec：按照 token 平均价格由高到低排序；
3. recent：按照上架顺序排序；

## Body 参数

无

## 返回参数

属性名	类型	描述
orderInfos	Array<Object>	NFT 卖单合集
total	integer	NFT 卖单总数

## orderInfos 的结构

属性名	类型	描述
tick	string	NFT 标识符
orderId	string	NFT 买单 ID

```
/v1/market/address/{address}/nft/{tick}/order/list
```

## 请求方式

GET

## Path 参数

属性名	类型	描述
address	string	BTC 地址

## Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）
order	no	string	排序规则（默认是 recent，按照上架顺序排序）

## 可选的 order 参数：

1. price\_desc：按照 token 平均价格由低到高排序；
2. price\_sec：按照 token 平均价格由高到低排序；
3. recent：按照上架顺序排序；

## Body 参数

无

## 返回参数

属性名	类型	描述
orderInfos	Array<Object>	NFT 卖单信息集合
total	integer	地址创建的 NFT 卖单总数

## orderInfos 的结构

属性名	类型	描述
tick	string	NFT 标识符
orderId	string	NFT 卖单 ID



返回示例

## 6.2.4 获取所有 NFT 交易历史

该接口会返回市场上已完成的所有 NFT 交易信息。

请求地址

```
/v1/market/nft/{tick}/transaction/list
```

请求方式

GET

Path 参数

属性名	类型	描述
tick	string	NFT 标识符

Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

Body 参数

无

返回参数

属性名	类型	描述
transactionInfos	Array<Object>	NFT 市场上已完成交易历史信息集合
total	integer	已完成交易总数

transactionInfos 的结构

属性名	类型	描述
tick	string	NFT 标识符

from	integer	交易卖方地址
to	integer	交易买方地址
inscriptionId	string	NFT 铸造铭文唯一 ID
price	integer	交易价格
time	integer	交易完成时的时间戳
transactionId	string	交易 ID

请求示例

返回示例

### 6.2.5 获取地址的 NFT 交易历史

该接口会返回市场上已完成的所有 NFT 交易信息。

请求地址

```
/v1/market/address/{address}/nft/{tick}/transaction/list
```

### 请求方式

GET

### Path 参数

属性名	类型	描述
address	string	BTC 地址
tick	string	NFT 标识符

### Query 参数

属性名	必需?	类型	描述
start	no	integer	起始偏移量（默认是 0）
end	no	integer	结束偏移量（默认是 10）

### Body 参数

无

### 返回参数

属性名	类型	描述
transactionInfos	Array<Object>	NFT 市场上已完成且与给定地址相关的交易历史信息集合
total	integer	已完成且与给定地址相关的交易总数

### transactionInfos 的结构

属性名	类型	描述
tick	string	NFT 标识符
from	integer	交易卖方地址
to	integer	交易买方地址
inscriptionId	string	NFT 铸造铭文唯一 ID
price	integer	交易价格

```
/v1/market/nft/order/sell/create
```

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
inscriptionId	yes	string	NFT 铭文的唯一 ID
price	yes	integer	NFT 价格
receivedAddress	yes	string	接收 BTC 的地址

返回参数

属性名	类型	描述
orderId	string	FT 卖单 ID
psbt	string	未签名的 PSBT,base64 格式
signIndex	Array<integer>	指定 PSBT 中需要签名的输入位置

请求示例

返回示例

### 6.2.7 确认出售订单

由卖方对 PSBT 交易签名后，再由卖方调用

请求地址

/v1/market/nft/order/sell/confirm

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
orderId	yes	string	NFT 卖单 id
psbt	yes	string	签名后 PSBT

返回参数

无

请求示例

返回示例

## 6.2.8 创建购买订单

该接口由买方调用，在服务器上创建买单，并匹配对应的卖单。服务器会返回给您 PSBT 交易并指示卖方需要再哪些交易输入中签名，请使用 SIGHASH\_ALL。

请求地址

/v1/market/nft/order/purchase/create

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
orderId	yes	string	NFT 卖单 ID
paidAddress	yes	string	支付 BTC 的地址
receivedAddress	yes	string	接收 NFT 的地址

返回参数

属性名	类型	描述
orderId	string	NFT 卖单 ID
psbt	string	未签名的 PSBT,base64 格式
signIndex	Array<integer>	指定 PSBT 中需要签名的输入位置

请求示例

返回示例

### 6.2.9 确认购买订单

由买方对 PSBT 交易签名后，再由买方调用

请求地址

/v1/market/nft/order/purchase/confirm

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
orderId	yes	string	NFT 卖单 id
psbt	yes	string	签名后 PSBT

返回参数

无

请求示例

返回示例

## 6.2.10 创建修改价格订单

该接口由卖方调用，可以修改在服务器上已创建的卖单的价格等□

请求地址

```
/v1/market/nft/order/modify/create
```

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
orderId	yes	string	NFT 转移铭文 ID
newPrice	yes	integer	NFT 的新价格
receivedAddress	yes	string	接收 BTC 的地址

返回参数

属性名	类型	描述
orderId	string	NFT 卖单 ID
psbt	string	未签名的 PSBT,base64 格式
signIndex	Array<integer>	指定 PSBT 中需要签名的输入位置

请求示例

返回示例

### 6.2.11 确认修改价格订单

由卖方对 PSBT 交易签名后，再由卖方调用

请求地址



## 6.1.12 创建取消出售订单

该接口由卖方调用，会取消已创建的出售订单。后续需要卖方签名 rawTransaction 从而将 NFT 的铭文转移给自己。

请求地址

```
/v1/market/nft/order/cancel/create
```

请求方式

POST

Path 参数

无

Query 参数

无

Body 参数

属性名	required?	type	description
orderId	yes	string	FT 转移铭文 ID

返回参数

无

请求示例

返回示例