**《项目设计文档》**

**基于百度超级链的NFT交易平台**

**项 目 名 称： 基于百度超级链的NFT交易平台**

**成 员 名 单： 朱前生 朱彬 朱光可**

**导 师： 郭燕**

**研 究 方 向： 区块链**

**中国科学技术大学软件学院**

1引言

## 1.1编写目的

本文档为基于百度超级链的NFT交易平台的项目设计说明书。为了进一步细化需求分析说明书中对该系统的整体框架和模块结构的设计，设计所有对象的操作和系统数据存储结构，安排项目规划和进度，撰写本文档。

# 2任务概述

## 2.1 目标

该文档主要的内容为该软件系统的总体设计，模块的处理流程，各个模块与其他模块的接口设计，数据结构的逻辑结构设计以及数据库设计，程序运行时的出错处理等。

## 2.2 运行环境

### 2.2.1 硬件环境

服务器端硬件配置要求如下：

* CPU：双核2.2GHz
* RAM：建议2GB
* 可用磁盘空间：200GB，转速>5400RPM

### 2.2.2 软件环境

1. 开发语言：Java，solidity，JavaScript
2. 开发环境：

* 底层支持：xuperchain
* 数据库：MySQL，Redis

c) 测试软件：

* 测试工具：JUnit

d) 服务器端软件配置环境：

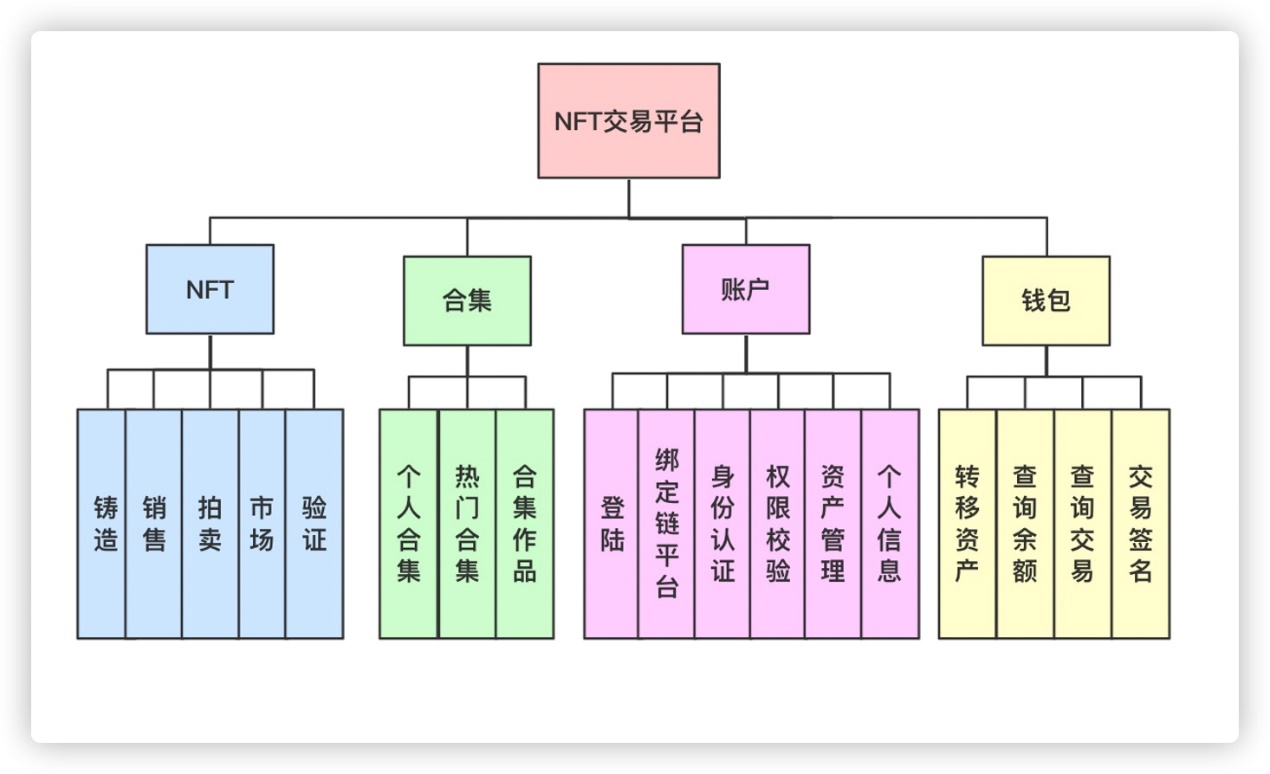
* 网络操作系统：Linux
* 数据库：MySQL

## 2.3 需求概述

该系统的关键模块由四大部分组成：1.NFT模块；2.合集模块；3.账户管理模块；4.钱包模块。

# 3总体设计

## 3.1系统功能模块



## 3.2各模块处理流程

### 3.2.1 钱包模块

用户注册百度超级链开放网络，得到自己的地址和私钥，上传私钥到钱包，就绑定了超级链平台。可以进行转移藏品，输入接收方address，藏品ID，数量，执行后自动进行交易签名，验证通过后，向后台发出转移藏品的请求，后台会创建订单，检查账户状态，锁住库存，新增NFT交易记录，由平台代理用户执行safeTransferForm合约函数将资产转移，随后库存解锁，即转移成功。

用户查询余额将调用balanceOf(address)合约函数，通过验证后，返回用户拥有的NFT资产余额。

每一个交易，都有一个唯一的交易哈希，通过交易哈希，可以查询到具体的交易。

### 3.2.2 账户模块

用户注册交易平台账号后，登录交易平台，后台会验证用户身份，查询用户的角色与权限，用于身份认证和权限校验。进行交易需要与钱包连接，在钱包中上传私钥，绑定百度超级链平台，会发送一个query型交易，签名验证通过后，平台与钱包就连接起来。

### 3.2.3 合集模块

利用合集模块进行聚合搜索，按条件筛选NFT，查看个人合集，热门合集。

### 3.2.3 NFT模块

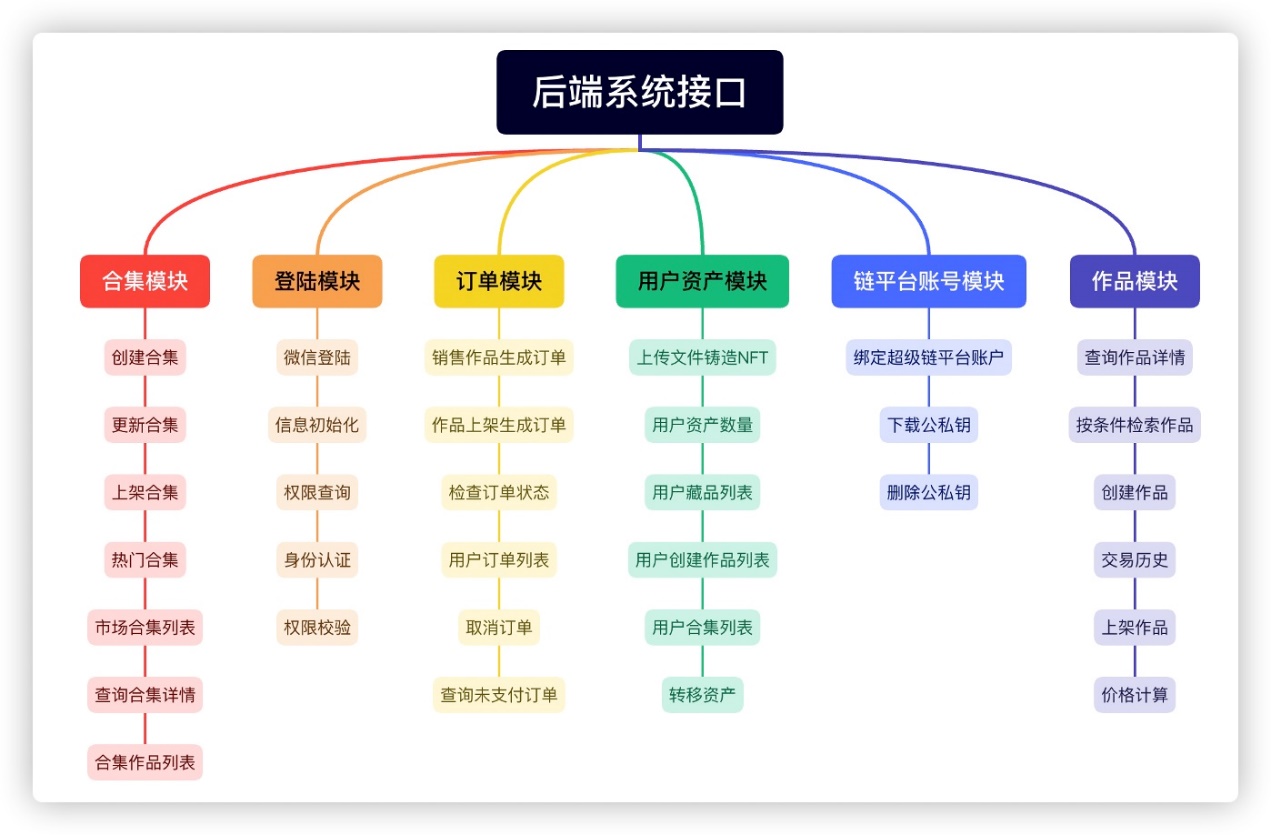
用户申请铸造NFT，填写好基本信息，上传NFT文件，向后端发送请求后，后端将NFT文件上传到IPFS，得到文件的聚合hash，CID，通过一定协议得到URI用于定位这个文件，它就作为NFT的externalLink，同时前端会向xuperchain发出合约函数mint的调用，增加用户的资产，记录铸造交易。

销售NFT需要先进行上架，安全性检查通过后，调用合约函数addNewToken预执行，预估gas费用，然后为作品生成上架订单，用户需要支付上架费用后才能真正上架，将NFT资产授权给平台，这样就可以通过平台售卖。

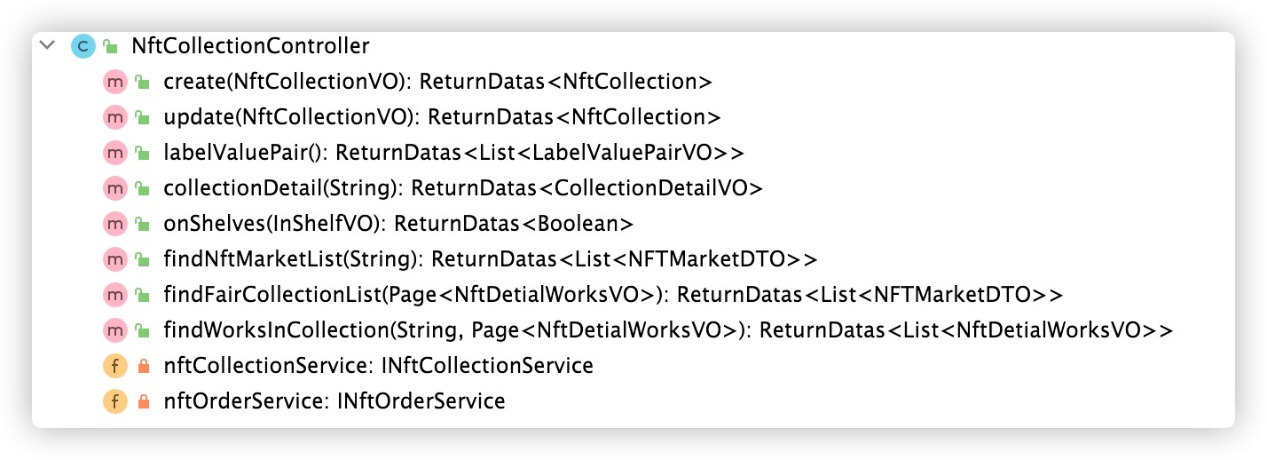
购买NFT需要进行签名认证，检查库存并上锁等操作，创建订单，交易成功后，要记录NFT的交易历史。

NFT验证是调用了百度超级链平台的存证验证平台，将NFT的地址上传，就可以进行NFT真实性验证，验证结果包含存证地址，存证时间，存证内容，区块高度等信息。

# 4接口设计

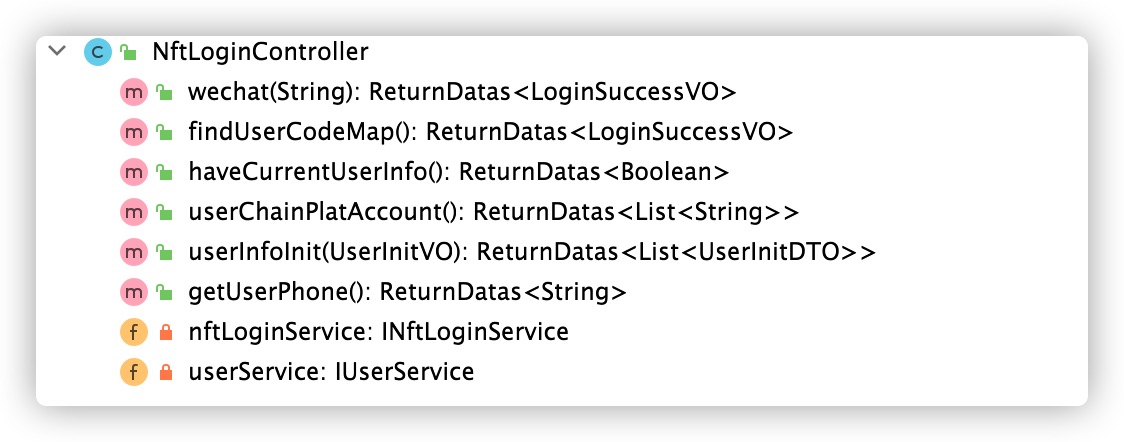


## 4.1合集模块接口



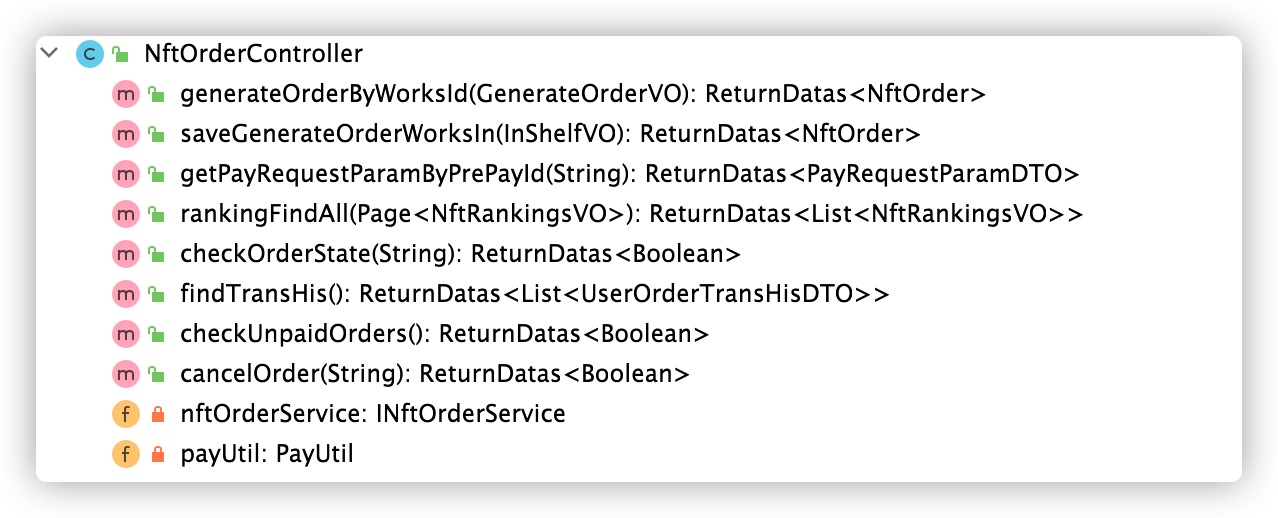
合集模块提供了创建合集，更新合集，查看合集详情，上架作品，在合集中查找NFT，查询合集交易记录等接口

## 4.2登录注册模块接口



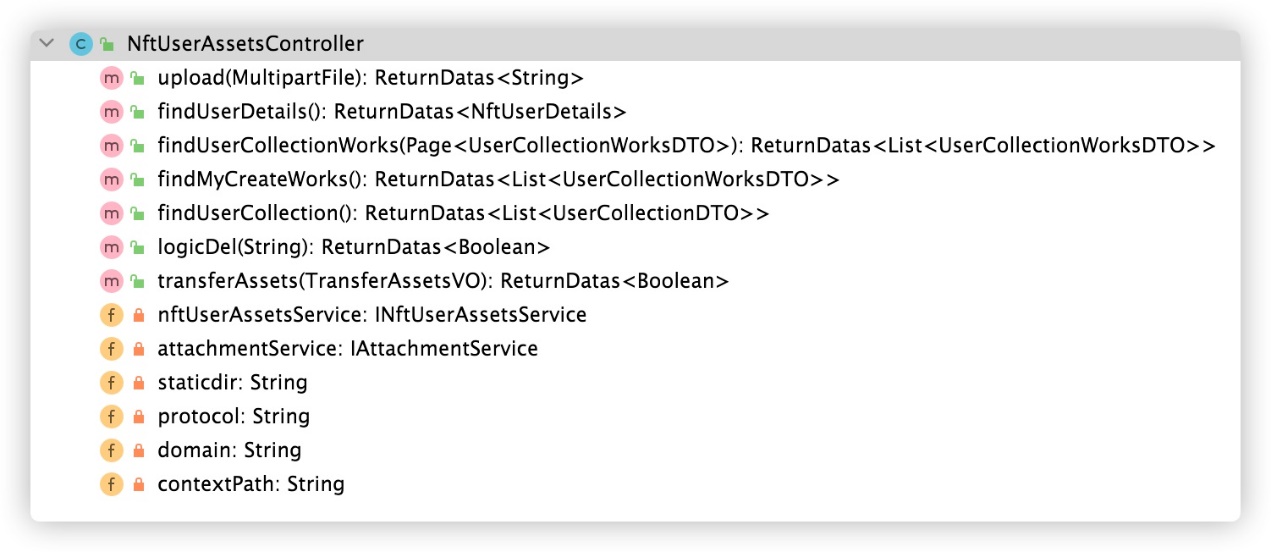
登陆模块提供了微信登陆接口，信息初始化，权限查询，身份认证，权限校验等接口。

## 4.3订单模块接口



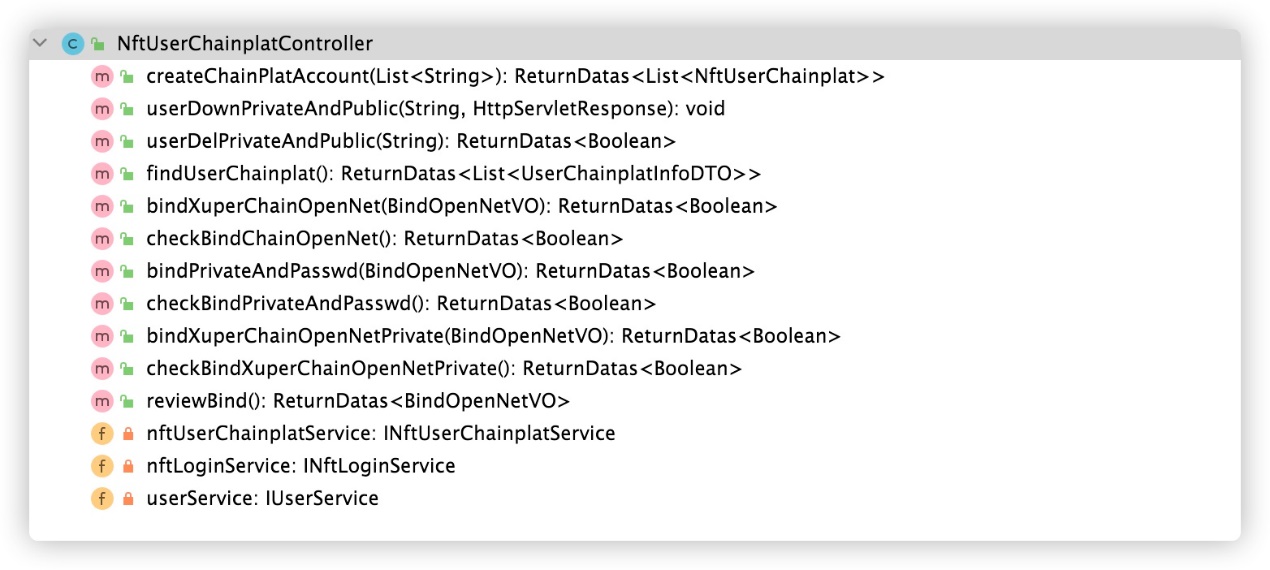
订单模块提供了生成订单，检查订单状态，查找交易历史，取消订单，发起支付请求等接口。

## 4.4用户资产模块接口



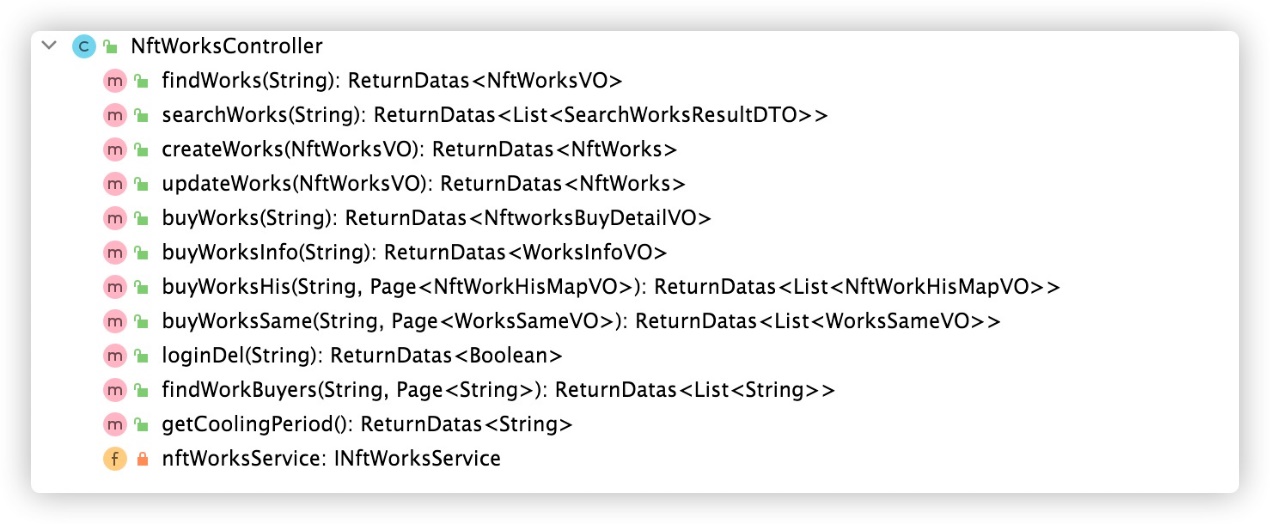
用户资产模块提供了查询拥有的所有资产，查询自己创建的NFT，转移资产，铸造NFT等接口。

## 4.5 链平台账号模块接口

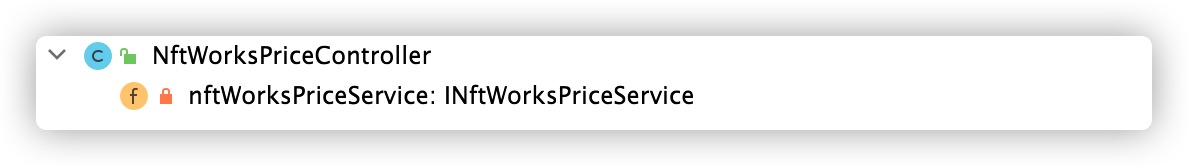


链平台模块提供了绑定链平台，创建链平台账户，查询绑定的用户链平台，下载私钥等接口

## 4.6作品模块接口







作品模块提供了查询NFT交易历史，计算作品价格，按条件筛选NFT，购买NFT等接口。

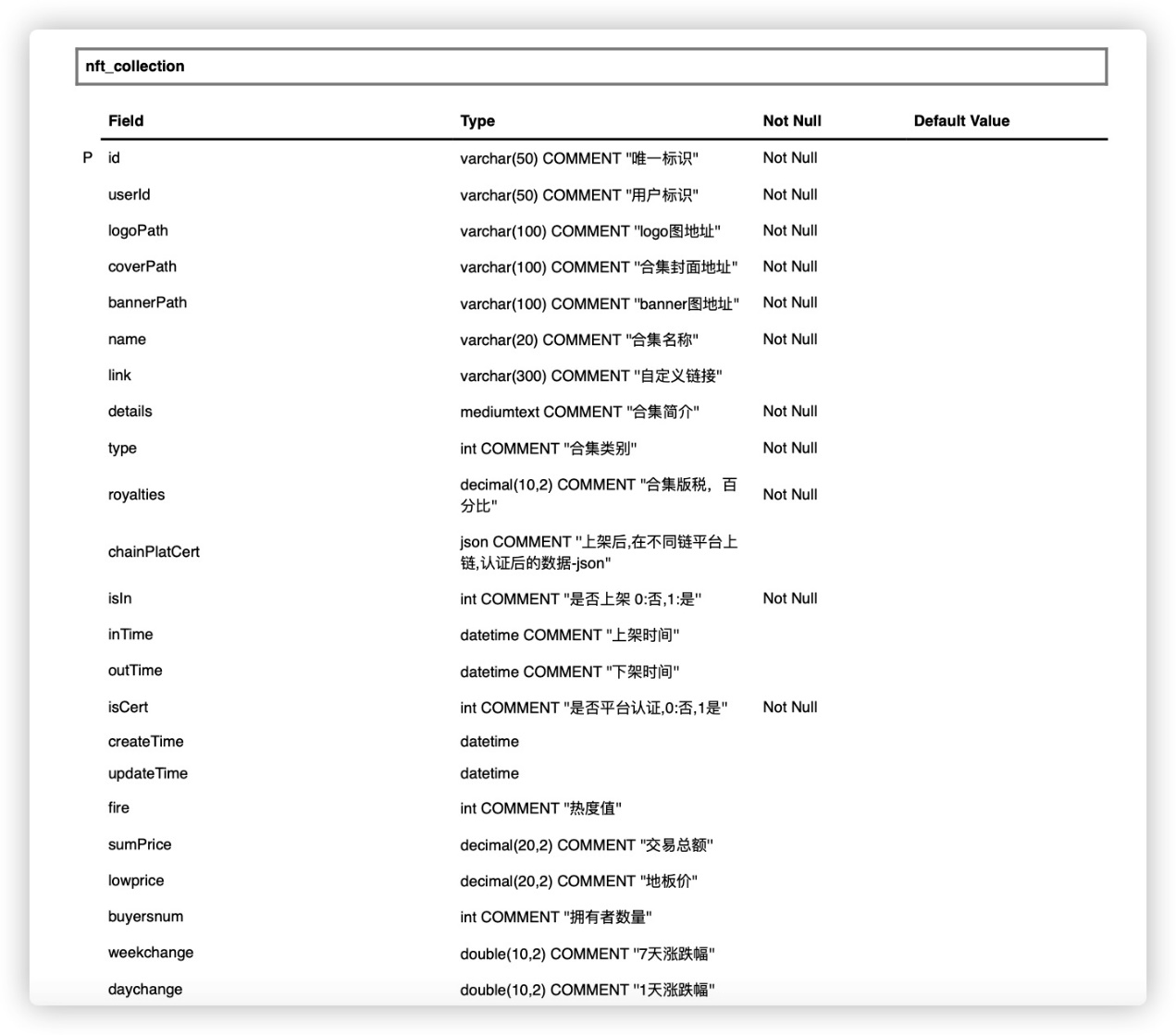
# 5.数据库设计

系统采用MySQL数据库，应用系统对数据库中的表拥有完全操作权限。

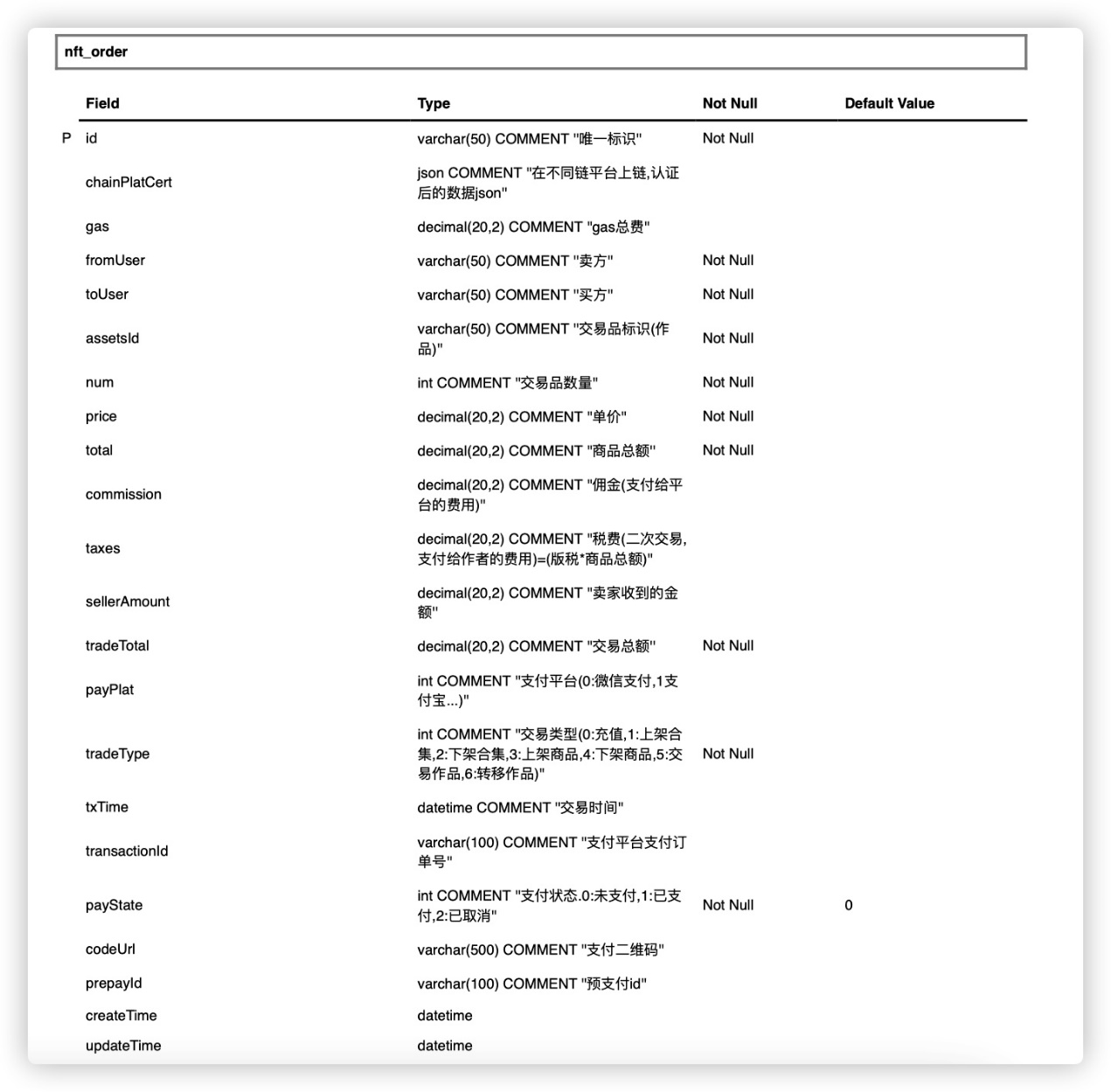
**1.用户表**



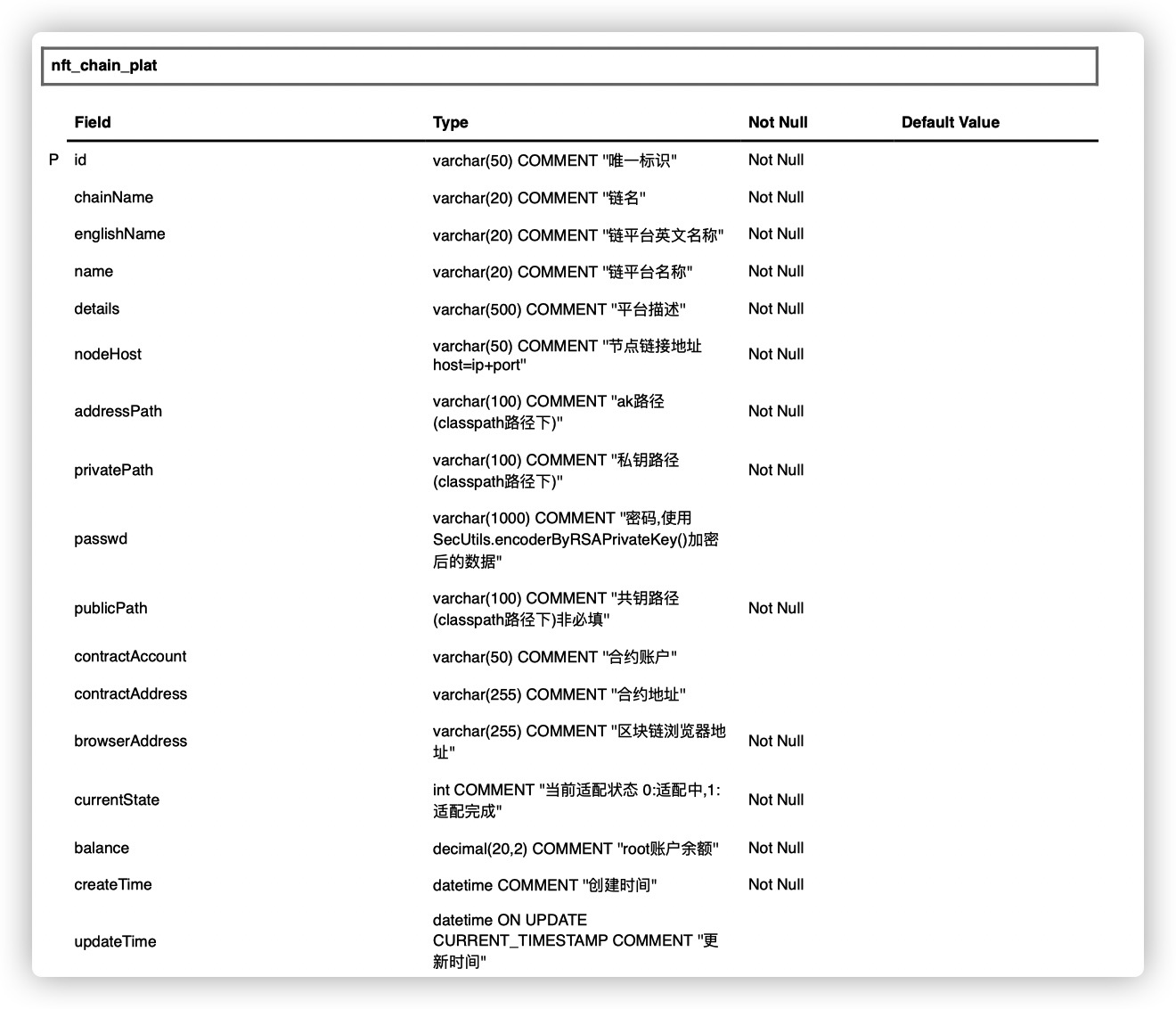
**2.合集表**



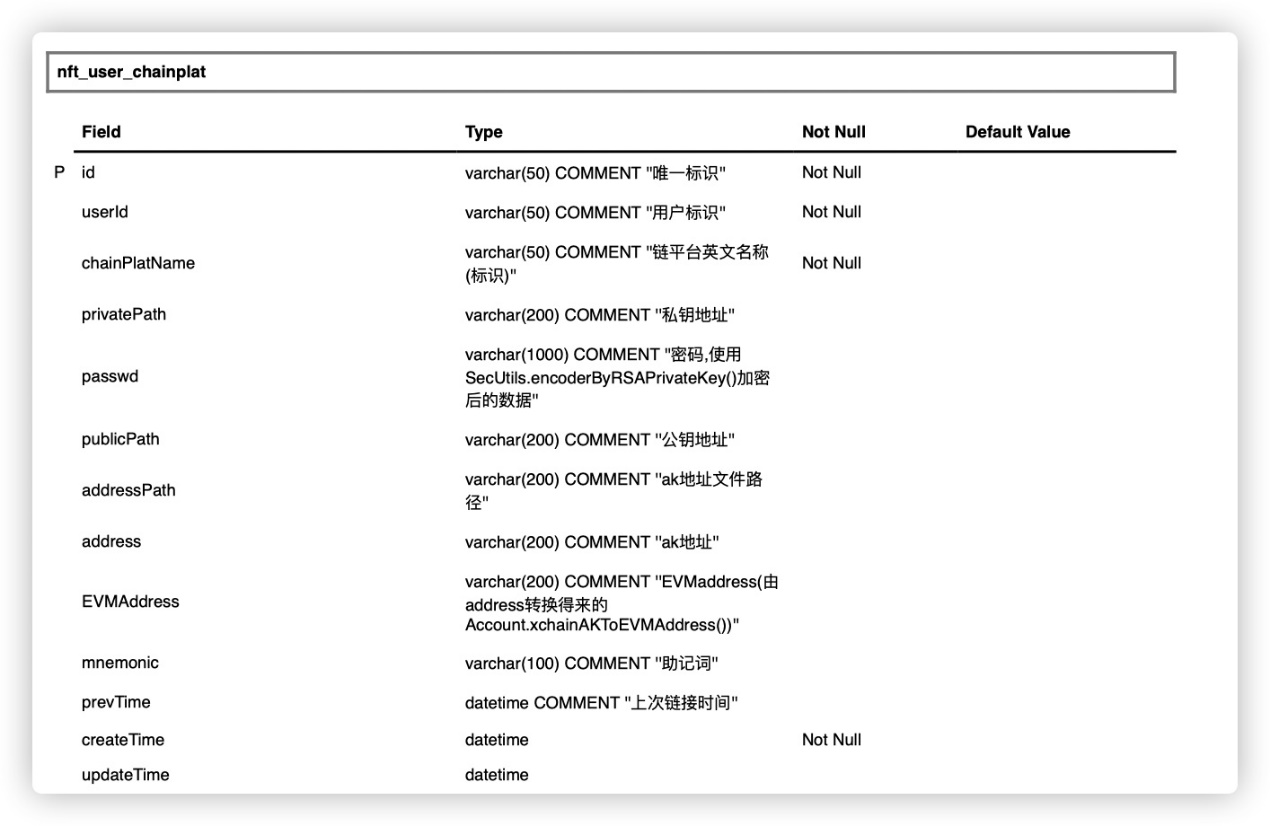
1. **订单表**



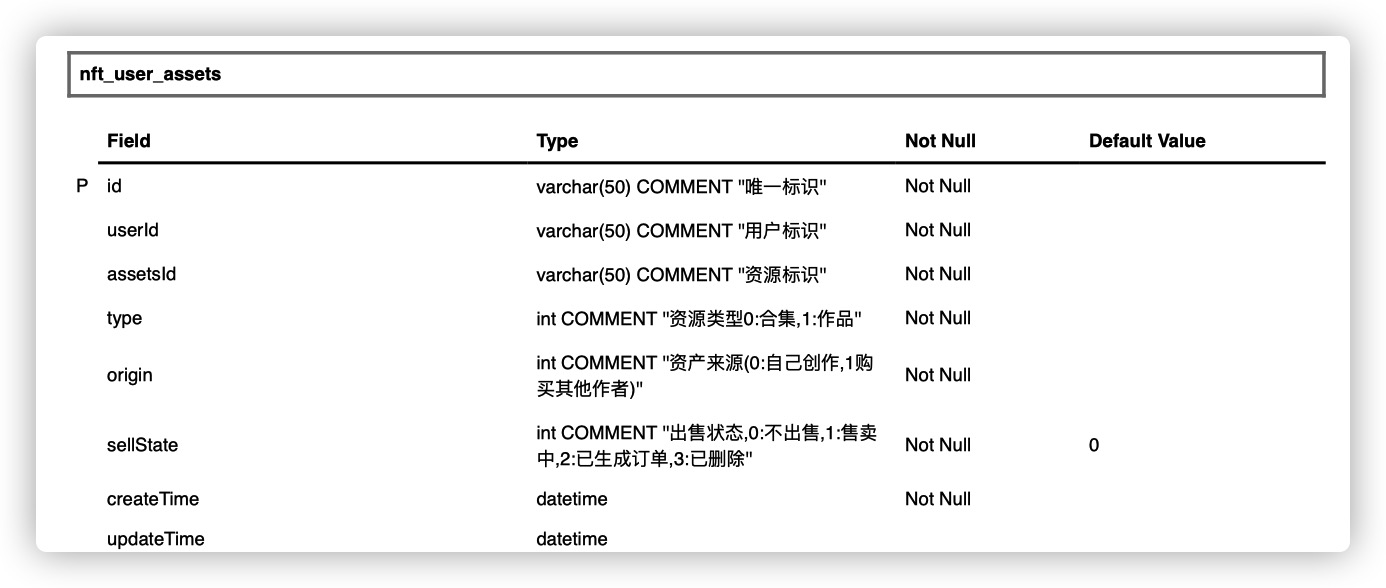
1. **链平表台**



1. **用户链平台绑定表**



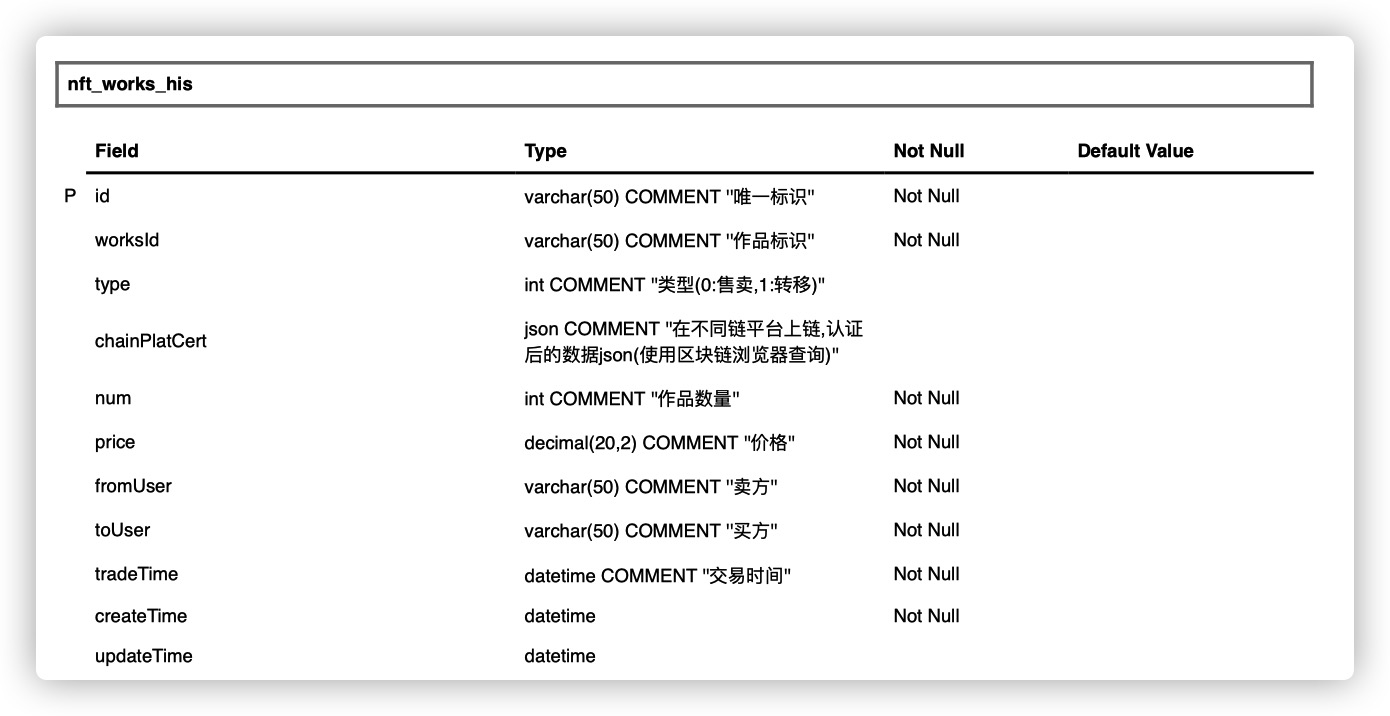
1. **用户资产表**



1. **Asset表**



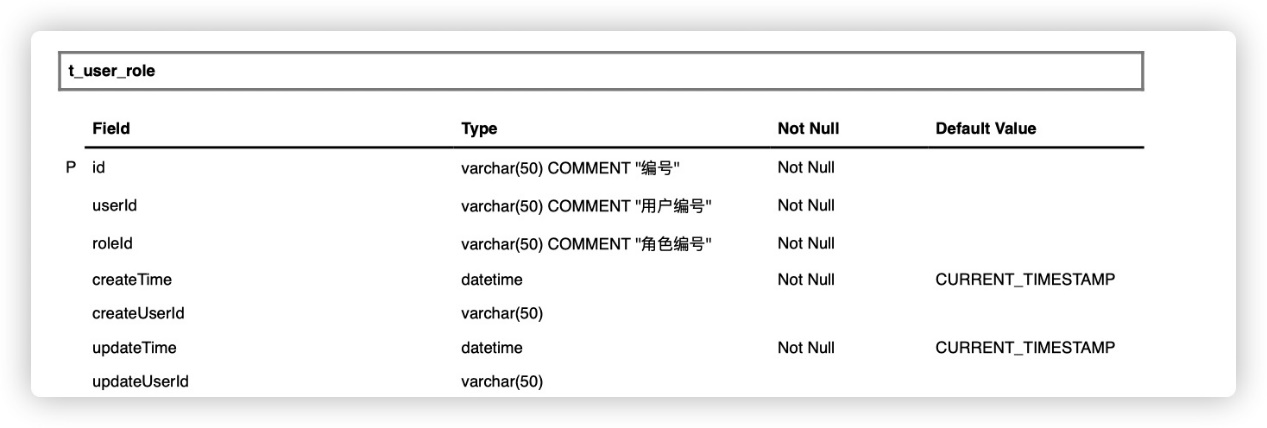
1. **Asset交易历史表**



1. **角色表**



1. **用户角色表**



# 6框架设计

## 6.1 封装响应数据

ReturnDatas类用于封装Controller层的返回数据，它是一个泛型数据结构，继承Serializable和HashMap，SpringBoot框架会将返回对象序列化为json对象格式进行传输。ReturnDatas对象可视作(status，message，result)的三元组，每个返回的对象都有一个状态码，默认为200，message是响应消息，result是结果数据，在Controller层可以调用setStatus，setMessage方法来填充信息，因为ReturnDatas类继承了HashMap，故使用put(key，value)填充数据。

## 6.2 全局异常类

NFTException类用于抛出运行时异常，它继承自RuntimeException，可以携带异常信息，在此基础上定义常用的异常常量NFTExceptionConst，如：参数为空，验证码错误，Asset不存在，操作失败等异常。

## 6.3 工具类

1. IPFSUtil

封装了上传文件和下载文件两个接口，上传文件接口会将NFT文件上传到IPFS服务器，获得文件CID，按照link协议进行拼接就可以得到IPFS URI，可以作为NFT文件的外部链接，下载文件接口需要传入CID，从IPFS服务器下载文件。

2. JWTUtil

JWT用于做身份认证，一方面可将JWT生成的token作为api-key，另一方面可从token中提取调用者id，要求在前端发送ajax请求时把token放入header。实现了三个方法。

 - createToken：根据密钥和加密算法，绑定userId值，生成随机字符串作为token  
   - getUserId：从token中提取userId  
   - verifierToken：验证token有效性和正确性，token有过期时间，存在redis中。

# 7安全设计

## ECC

构建区块链的去中心化交易，需要一种加密算法，使交易发起人使用持有的密钥对交易数据进行数字签名，而交易验证者只需要知道交易发起人的公开信息，即可对交易有效性进行验证，确定该交易确实来自交易发起者。在XuperChain 中，密码学技术广泛应用在账户体系、交易签名、数据隐私保护等方面，主要以ECC(椭圆曲线密码体系)以及多种Hash散列算法为基础，发展出的一个单独的模块。

在 XuperChain 中涉及到的算法有：

* ECIES (Elliptic Curve Integrated Encryption Scheme): 椭圆曲线集成加密算法，主要用于基于椭圆曲线的数据加解密。
* ECDH (Elliptic Curve Diffie–Hellman): 基于Diffie–Hellman算法的一种密钥协商算法，定义了双方如何安全的生成和交换密钥。
* ECDSA (Elliptic Curve Digital Signature Algorithm): 是使用椭圆曲线密码学实现的DSA(数字签名算法)，一般发起人对消息摘要使用私钥签名，验证者可以通过公钥对签名有效性进行验证。

## 用户公私钥账户

XuperChain 的用户账户体系基于非对称公私钥对，每个用户账户对应这一组公私钥对，并采用一定的哈希算法将公钥摘要成一个字符串作为用户账户地址(address)。考虑到密钥不具备可读性，为了帮助用户保存密钥， XuperChain 实现了BIP39提案的助记词技术。

* 助记词的生成过程：首先生成一个长度在128~256bit之间的随机熵，由此在助记词表中选出对应的单词列表，形成助记词。
* 助记词产生私钥：使用基于口令的密钥派生算法PBKDF2，将上述生成的助记词和用户指定的密钥作为密钥派生算法参数，生成长度为512bit的种子，以此种子作为生成密钥的随机参数，便产生了了从助记词生成的私钥。
* 通过助记词恢复密钥：由于用户持有生成密钥的助记词和口令，因此在用户私钥遗忘或丢失时，可以通过同样的助记词和口令，执行助记词产生私钥的过程，从而恢复出账户密钥。

## 数据签名

XuperChain 中，每个交易都需要交易发起人以及交易背书人的签名；在每个块生成时，也需要加上打包块的节点的签名。

* **交易签名：** 基于交易数据摘要，会包含交易输入输出、合约调用、合约读写集、发起人和背书人信息等，并将交易数据序列化后的字节数组使用双重SHA256得到摘要数据，最后对摘要数据用ECDSA或其他数字签名算法产生交易签名。
* **块签名：** 基于区块数据摘要，会包含区块元信息如前序块Hash值、交易Merkle树根、打包时间、出块节点等数据，并在序列化后使用双重SHA256得到摘要数据，最后对摘要数据用ECDSA或其他数字签名算法产生区块签名。

## 通信加密

XuperChain 底层采用P2P网络传播交易和区块数据，在一些许可区块链网络场景中，需要对节点间的数据传输进行加密提升安全和隐私性，因此 XuperChain 的P2P连接支持基于ECDH的密钥交换算法的TLS连接。

ECDH的原理是交换双方可以在不共享任何秘密的情况下协商出一个密钥，双方只要知道对方的公钥，就能和自己的私钥通过计算得出同一份数据，而这份数据就可以作为双方接下来对称加密的密钥。

* 第一阶段是Propose阶段，这一阶段，对等节点间互相交换双方永久公钥PK。
* 第二阶段是Exchange阶段，本质是基于ECDH的密钥交换。双方通过ECC算法随机生成一组临时密钥对(tempPK, tempSK)，然后用自己的永久私钥对临时公钥tempPK进行签名并交换。这时，双方可以通过第一步的公钥进行验签，同时拿到供本次会话使用的临时公钥。使用临时公钥的好处是一话一密，即使本次会话密钥泄露也不会导致以前的加密数据被破解。ECDH算法使得双方通过对方的公钥和自己的私钥，可以获得一致的共享密钥SharedKey。
* 第三阶段是Verify阶段。双方使用ShareKey产生两组密钥Key1，Key2分别作为读写密钥，并使用支持的对称加密算法(AES/blowfish)加密传输第一步中发送给对方的Nonce，而接收方则使用刚才协商的密钥对数据解密，并验证Nonce是不是等于第一步中自己发送给对方的值。

通过这三次握手，双方建立了加密通信通道，并且节点间通信加密信道满足一话一密的高安全等级。