

Nft/OpenSea交易平台开发系统（详情方案）源码

产品名称	Nft/OpenSea交易平台开发系统（详情方案）源码
公司名称	广州杰肯狸网络科技有限公司
价格	.00/件
规格参数	
公司地址	广州天河区中山大道
联系电话	18125913365 19927739756

产品详情

OpenSea市场上的交易是可以实现智能交易的控制，这意味着是在去中心化平台上进行的交易。而是通过保证公平交易来促进卖方获得这些报酬。OpenSea在双方之间进行交易。尽管如此平台交易仍没有在市场上以2.5%的价格【系统176搭建·0206-可电可微5616-】与每个市场的竞争者达成交易。OpenSea架构协议是由Wyvern的一组智能协议平台，提供了该链的跨链和以太坊的数字化交易平台支持。

调用ProxyRegistry合约的registerProxy()方法就创建了，ProxyRegistry是使用opensea用户的合约账户的管理合约更多了解

```
function registerProxy()
```

```
public
```

```
returns(OwnableDelegateProxy proxy)
```

```
{
```

```
require(proxies[msg.sender]==address(0));
```

```
//new一个钱包。这个钱包合约是个可升级的代理合约，
```

```
proxy=new OwnableDelegateProxy(msg.sender,delegateProxyImplementation,abi.encodeWithSignature("initialize(a  
ddress,address)",msg.sender,address(this)));
```

```
//保存到mapping以所有的的地址作为索引的key
```

```
proxies[msg.sender]=proxy;
```

```
return proxy;
```

```
}
```

合约账户的初始化

```
contract OwnableDelegateProxy is OwnedUpgradeabilityProxy{
```

```
constructor(address owner,address initialImplementation,bytes calldata)
```

```
public
```

```
{
```

```
setUpgradeabilityOwner(owner);
```

```

_upgradeTo(initialImplementation);

require(initialImplementation.delegatecall(calldata));

}

}

```

OwnableDelegateProxy 合约账户的代理合约（存储）它继承至 OwnedUpgradeabilityProxy & Proxy, OwnedUpgradeabilityStorage, 这个合约主要是代理合约的逻辑升级，和代理调用的逻辑，用户是可以【系统176搭建 · 0206-可电可微5616-】修改这个合约的实现的更多了解

AuthenticatedProxy 这个合约是合约账户的逻辑合约也就是 OwnableDelegateProxy 的逻辑功能。他继承了 TokenRecipient, OwnedUpgradeabilityStorage。这个合约主要有三个方法

1 初始化

// 这个方法是 new OwnableDelegateProxy 时调用的设置了所属用户和管理合约这两个值时可以修改的 1 是在 proxy 方法 delegatecall 修改，还有就是升级逻辑合约

```
function initialize(address addrUser, ProxyRegistry addrRegistry)
```

```
public
```

```
{
```

```
require(!initialized);
```

```
initialized=true;
```

```
user=addrUser;
```

```
registry=addrRegistry;
```

```
function setRevoke(bool revoke)
```

```
public
```

```
{
```

```
require(msg.sender==user);
```

```
revoked=revoke;
```

```
emit Revoked(revoke);
```

```
function proxy(address dest,HowToCall howToCall,bytes calldata)
```

```
public
```

```
returns(bool result)
```

```
{
```

```
require(msg.sender==user||(!revoked&&registry.contracts(msg.sender)));//setRevoke后就只有用户自己可以调用
```

```
if(howToCall==HowToCall.Call){
```

```
result=dest.call(calldata);
```

```
}else if(howToCall==HowToCall.DelegateCall){
```

```
result=dest.delegatecall(calldata);
```

```
}
```

```
return result;
```

```
function proxyAssert(address dest,HowToCall howToCall,bytes calldata)
```

```
public
```

```
{
```

```
require(proxy(dest,howToCall,calldata));
```

```
}
```

TokenTransferProxy->专门转账的合约

```
contract TokenTransferProxy{
```

```
/* Authentication registry.*/
```

```
ProxyRegistry public registry;
```

```
/**
```

```
* Call ERC20 `transferFrom`
```

```
*
```

```
* dev Authenticated contract only
```

```
* param token ERC20 token address
```

```
* param from From address
```

```
* param to To address
```

```
* param amount Transfer amount
```

```
*/
```

```
function transferFrom(address token,address from,address to,uint amount)
```

public