**Flow Learning platform**

**Requirements and Design Specifications**

**October 2022**

Document Version History

|  |  |  |  |
| --- | --- | --- | --- |
| date | version | editor | content |
| 2022.10.08 | V1.0 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Project overview

Build a development and learning platform for Flow developers. The project includes 3 modules, a code download module, a search module, and a display module. The flow study system is mainly used as an effective tool and development platform for learning flow. On the basis of the software platform, the project also provides functions such as code search. The specific features of the project are as follows:

* Code search
* Similar code detection
* Code migration
* Code generation
* Code optimization
* Code automatic prompt

# 1 Requirements overview

The overall requirements of the project are as follows:

## 1) Enrich the number of contracts and provide a more complete application ecosystem.

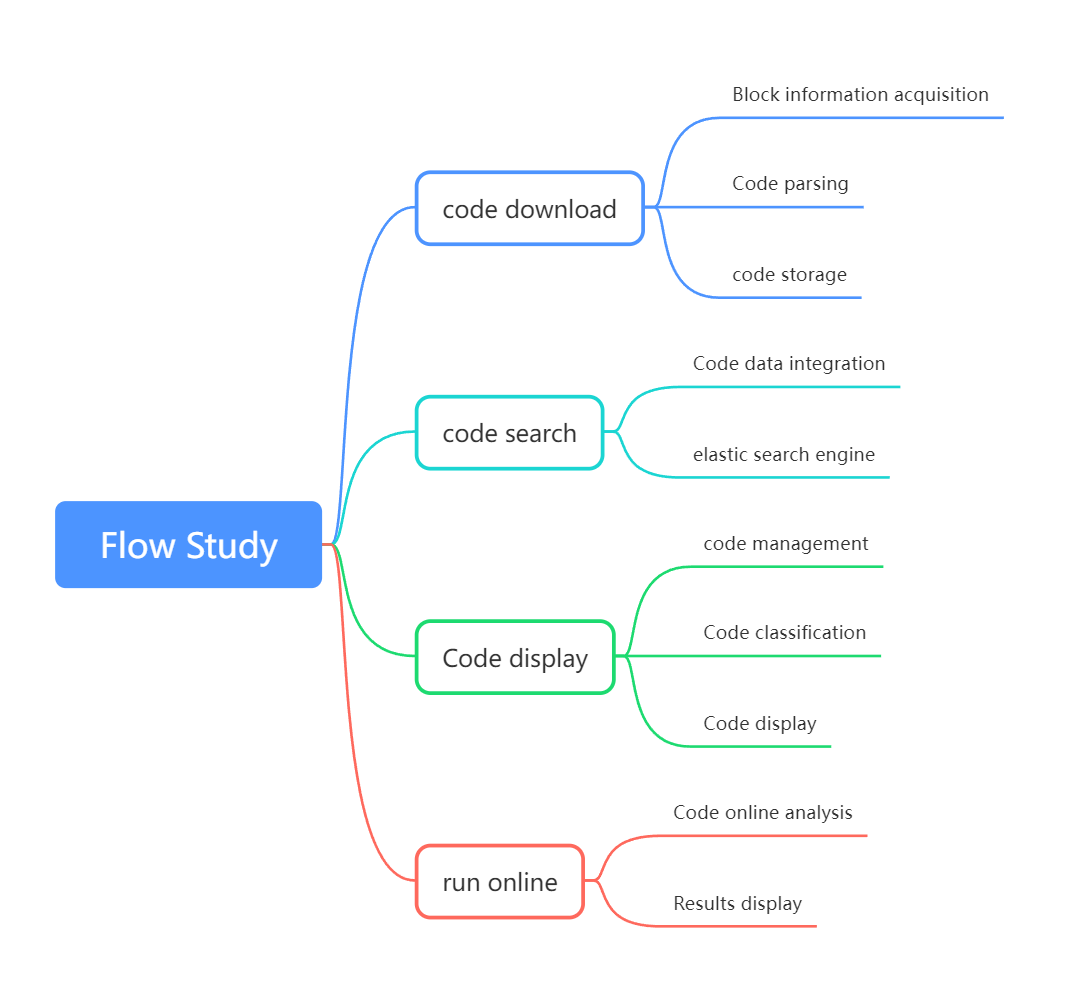
## 2) Big data support system: Code is the cornerstone of the ecology, and Flow currently needs more contract codes.

## 3) Basic Flow programming is difficult, and the logic of the code is not clear. It is difficult to understand why it is so difficult. There is an urgent need for a tool platform that can clearly view the code.

## 4) The code paradigm feels a bit awkward. Besides NFT contracts, I don’t know what else to write. The platform should provide a variety of code examples.

# 2 overall design

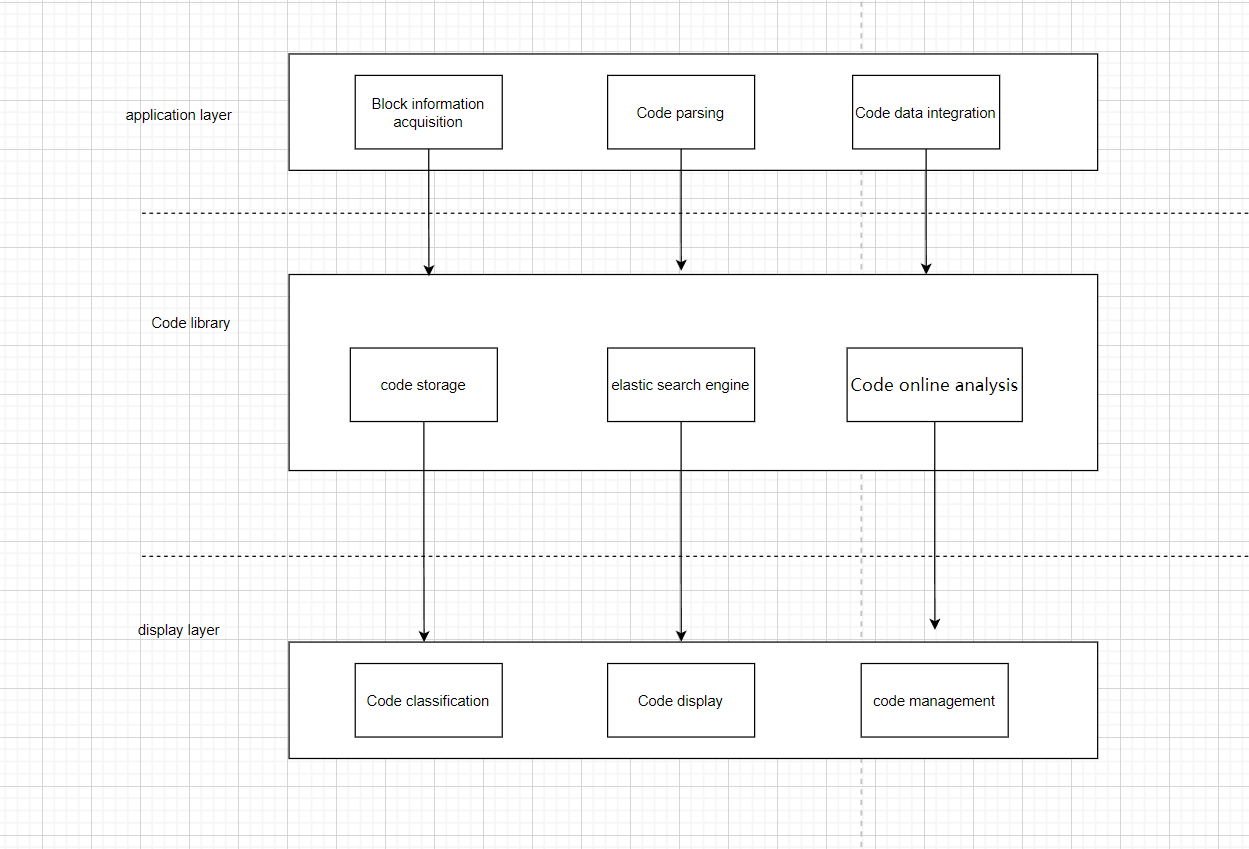
## 2.1 Product function diagram



2-1 Product function diagram

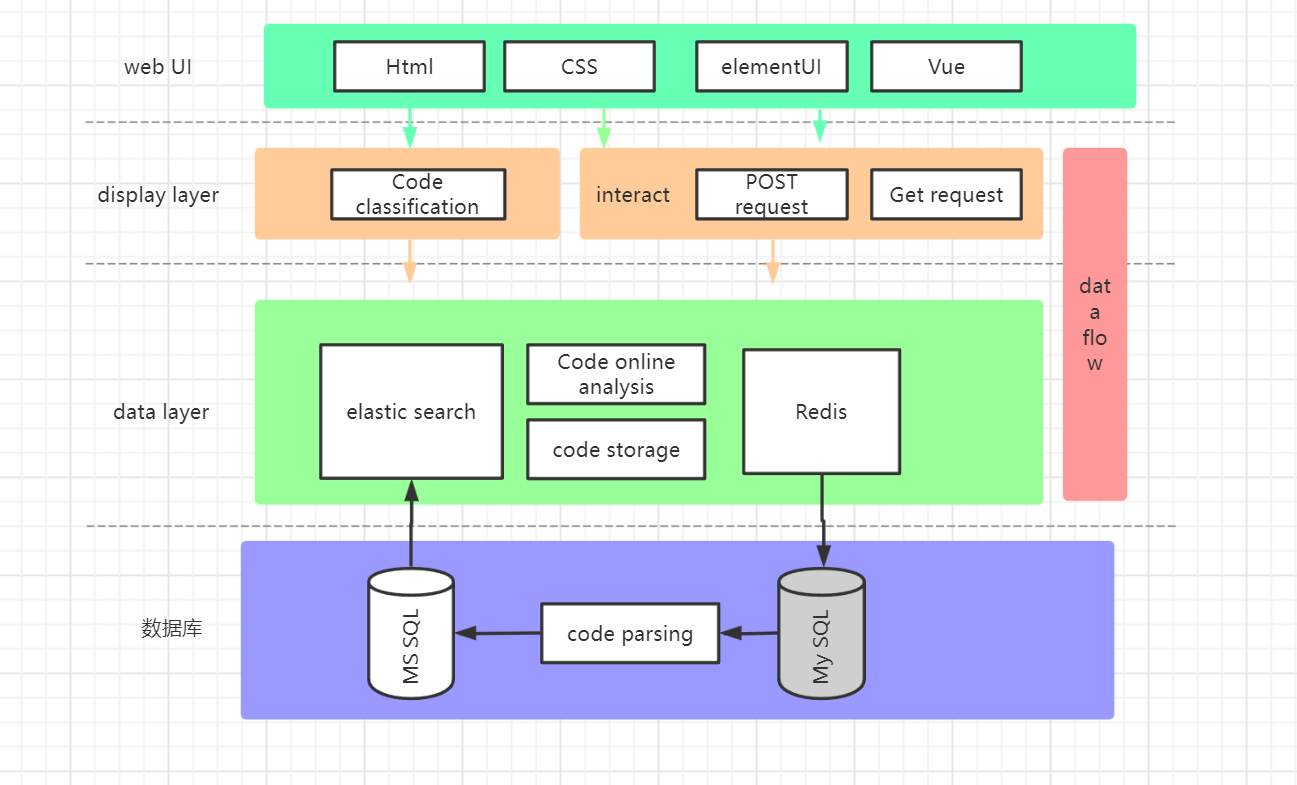
The overall functional brain map of the product is shown above. The first three modules are the work that needs to be completed at present.

The overall functional structure is as follows:



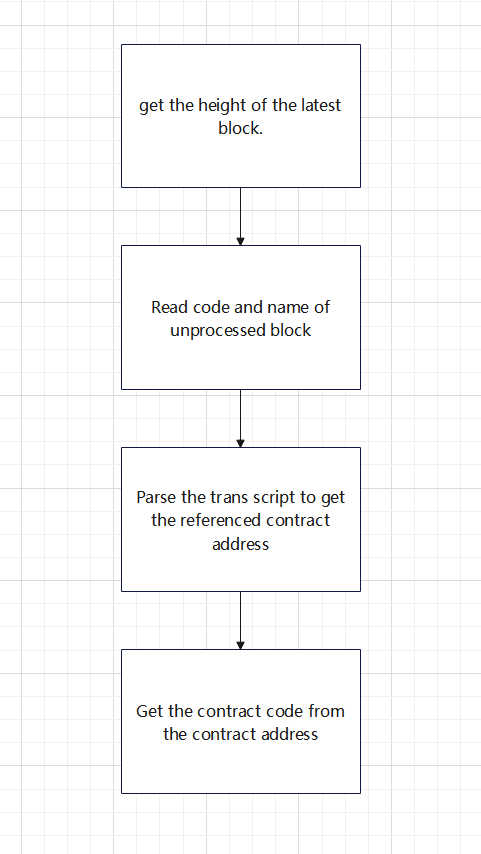
2-2 Functional structure diagram

## 2.2 System structure design

The overall design of the project is shown in the following figure:

2-3 System Architecture

## 2.3Flow Study internal information relationship



2-4 Internal Information Diagram

First get the height of the latest block and insert it into the flow\_block table

Then read the code and name of the unprocessed block from flow\_block, call the get\_trans method, and insert the information into the flow\_trans\_data table.

By parsing the transaction script, the referenced contract address is obtained, inserted into the data table flow\_trans\_data, from the flow\_trans\_data table, the contract address is obtained every hour and inserted into the flow\_contract\_address table.

Finally, the unprocessed contract\_address is obtained from the flow\_contract\_address table through the get\_contract method and inserted into the flow\_code table.

# 3 Module design and implementation

## 3.1 Goals

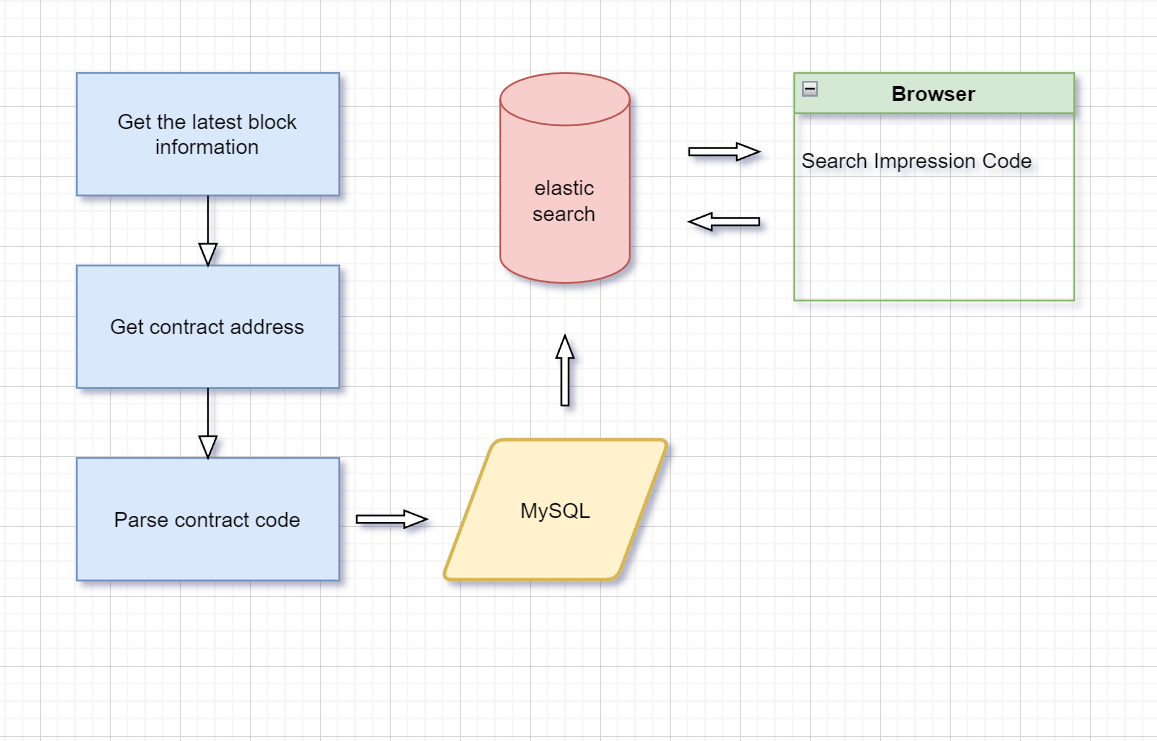
Mainly based on the typical learning platform design and the corresponding front-end interface design, the following goals are achieved:

1 The basic functions can be realized, and the collection code can be obtained automatically.

2 The project can be displayed and the parsed code can be searched.

## 3.2 Overall process

Specifically as shown in the figure below：



3-1 flow chart

## 3.3 data structure design

table flow\_block

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | int | 10 | 0 | N | Y |  |  |
| 2 | block\_seals | varchar | 255 | 0 | Y | N |  |  |
| 3 | timestamp | datetime | 19 | 0 | Y | N |  |  |
| 4 | collection\_guarantees | varchar | 255 | 0 | Y | N |  |  |
| 5 | block\_id | varchar | 255 | 0 | Y | N |  |  |
| 6 | height | bigint | 19 | 0 | Y | N |  |  |
| 7 | signatures | varchar | 255 | 0 | Y | N |  |  |
| 8 | parent\_id | varchar | 255 | 0 | Y | N |  |  |
| 9 | is\_updated | int | 10 | 0 | N | N | 0 |  |
| 10 | fetch\_time | datetime | 19 | 0 | Y | N |  |  |

table flow\_code

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | int unsigned | 10 | 0 | N | Y |  |  |
| 2 | contract\_name | varchar | 200 | 0 | Y | N |  |  |
| 3 | contract\_address | varchar | 200 | 0 | Y | N |  |  |
| 4 | contract\_code | longtext | 2147483647 | 0 | Y | N |  |  |
| 5 | contract\_type | varchar | 200 | 0 | Y | N |  |  |
| 6 | is\_process | int | 10 | 0 | N | N | 0 | The default is 0, which is not processed. 1 means processed. |

table flow\_contract\_address

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | int unsigned | 10 | 0 | N | Y |  |  |
| 2 | contract\_name | varchar | 200 | 0 | Y | N |  |  |
| 3 | contract\_address | varchar | 200 | 0 | Y | N |  |  |
| 4 | contract\_code | text | 65535 | 0 | Y | N |  |  |
| 5 | is\_process | int | 10 | 0 | N | N | 0 | The default is 0, which is not processed. 1 means processed. |

table flow\_token\_from (Flow token sending address)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | int unsigned | 10 | 0 | N | Y |  |  |
| 2 | from\_address | varchar | 200 | 0 | Y | N |  |  |
| 3 | amount | float | 12 | 0 | Y | N |  |  |
| 4 | trans\_id | varchar | 200 | 0 | Y | N |  |  |

table flow\_token\_from\_to (Flow token sending address)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | int unsigned | 10 | 0 | N | Y |  |  |
| 2 | from\_address | varchar | 200 | 0 | Y | N |  |  |
| 3 | amount | float | 12 | 0 | Y | N |  |  |
| 4 | trans\_id | varchar | 200 | 0 | Y | N |  |  |
| 5 | to\_address | varchar | 200 | 0 | Y | N |  |  |

table flow\_token\_to (Flow token sending address)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | int unsigned | 10 | 0 | N | Y |  |  |
| 2 | to\_address | varchar | 200 | 0 | Y | N |  |  |
| 3 | amount | float | 12 | 0 | Y | N |  |  |
| 4 | trans\_id | varchar | 200 | 0 | Y | N |  |  |

表flow\_trans\_data

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | int unsigned | 10 | 0 | N | Y |  |  |
| 2 | trans\_id | varchar | 200 | 0 | Y | N |  | 交易id |
| 3 | user\_address | varchar | 200 | 0 | Y | N |  | 用户地址 |
| 4 | contract\_name | varchar | 200 | 0 | Y | N |  | 合约名称 |
| 5 | contract\_address | varchar | 200 | 0 | Y | N |  | 合约地址 |
| 6 | fetch\_time | datetime | 19 | 0 | Y | N |  |  |
| 7 | height | bigint | 19 | 0 | Y | N |  |  |

表flow\_trans\_data\_copy1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | int unsigned | 10 | 0 | N | Y |  |  |
| 2 | trans\_id | varchar | 200 | 0 | Y | N |  | transaction id |
| 3 | user\_address | varchar | 200 | 0 | Y | N |  | User address |
| 4 | contract\_name | varchar | 200 | 0 | Y | N |  | contract name |
| 5 | contract\_address | varchar | 200 | 0 | Y | N |  | contract address |
| 6 | fetch\_time | datetime | 19 | 0 | Y | N |  |  |

table member (user table)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | int | 10 | 0 | N | Y |  | user id |
| 2 | email | varchar | 200 | 0 | N | N |  | user mail |
| 3 | password | varchar | 256 | 0 | N | N |  | user password，md5 |
| 4 | username | varchar | 200 | 0 | N | N |  | user name |
| 5 | regdate | datetime | 19 | 0 | N | N |  | registration date |
| 6 | user\_platform | int | 10 | 0 | N | N |  | User source (0 for aso, 1 for user microservice) |
| 7 | nickname | varchar | 200 | 0 | N | N |  | User nickname, Weibo is nickname, generally registered as a string before email @ |
| 8 | level | int | 10 | 0 | N | N | 20 | user level |
| 9 | phone\_num | varchar | 100 | 0 | Y | N |  | User mobile number |
| 10 | company | varchar | 200 | 0 | Y | N |  | Company Name |

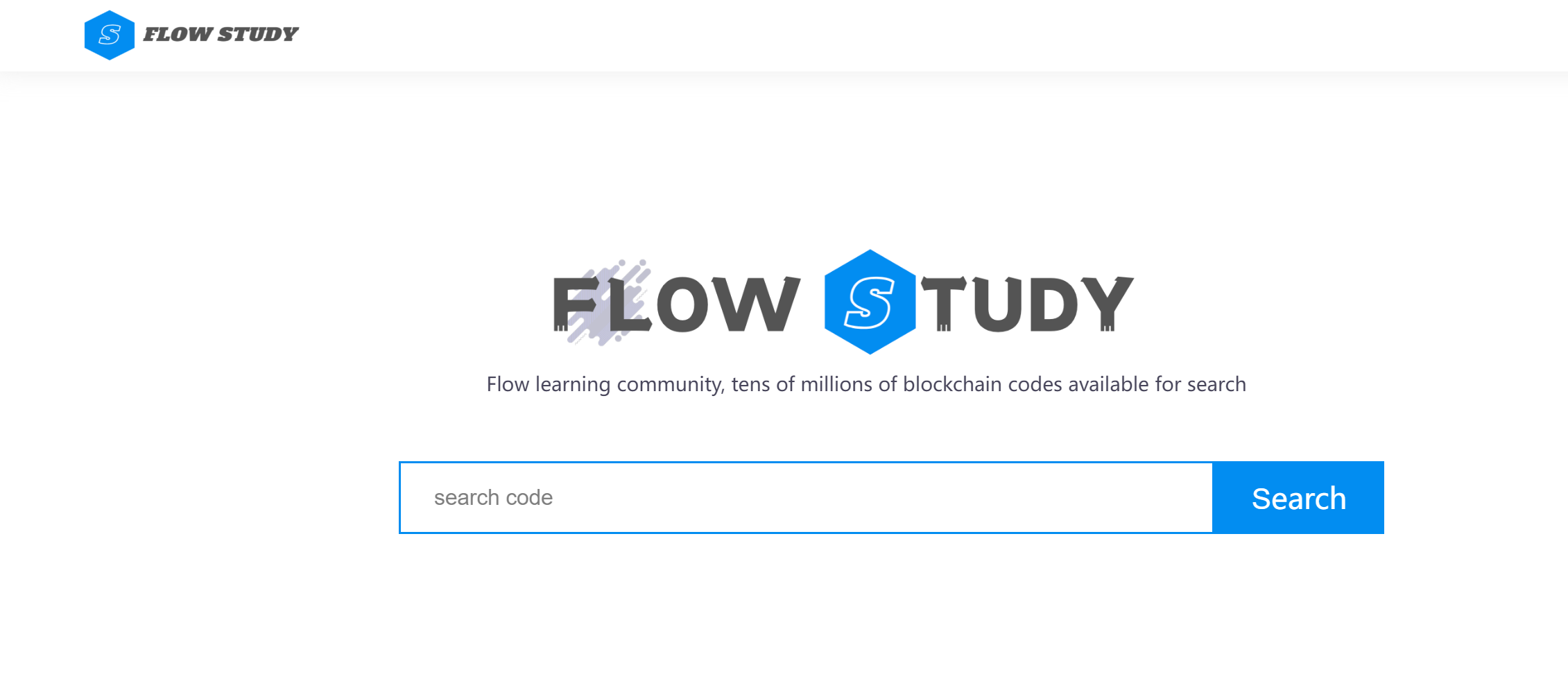
table member\_token

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | bigint | 19 | 0 | N | Y |  |  |
| 2 | email | varchar | 256 | 0 | N | N |  |  |
| 3 | token | varchar | 256 | 0 | N | N |  |  |
| 4 | login\_time | datetime | 19 | 0 | N | N |  | User login time |
| 5 | ip | varchar | 256 | 0 | N | N |  | user ip |

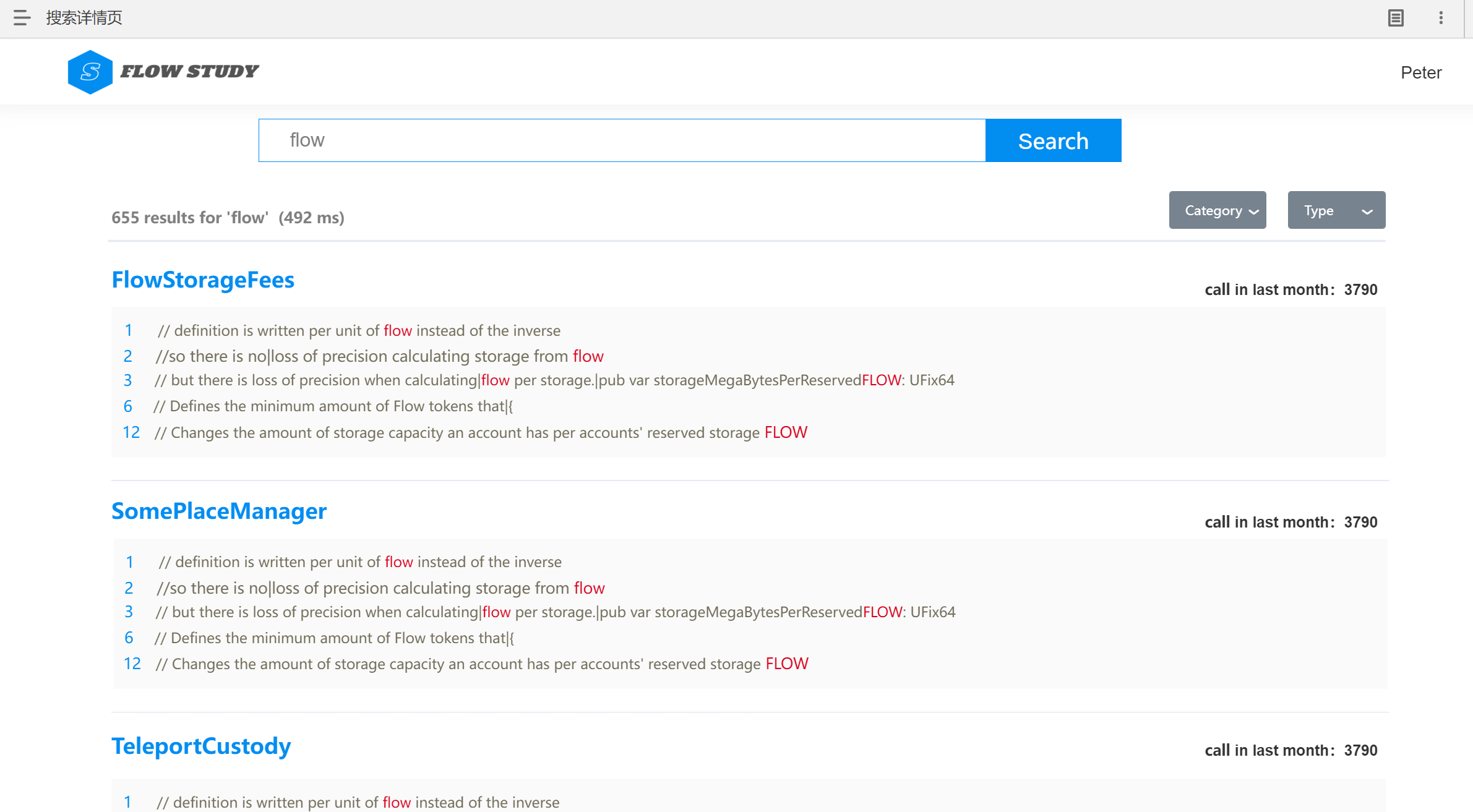
table opensea\_asset

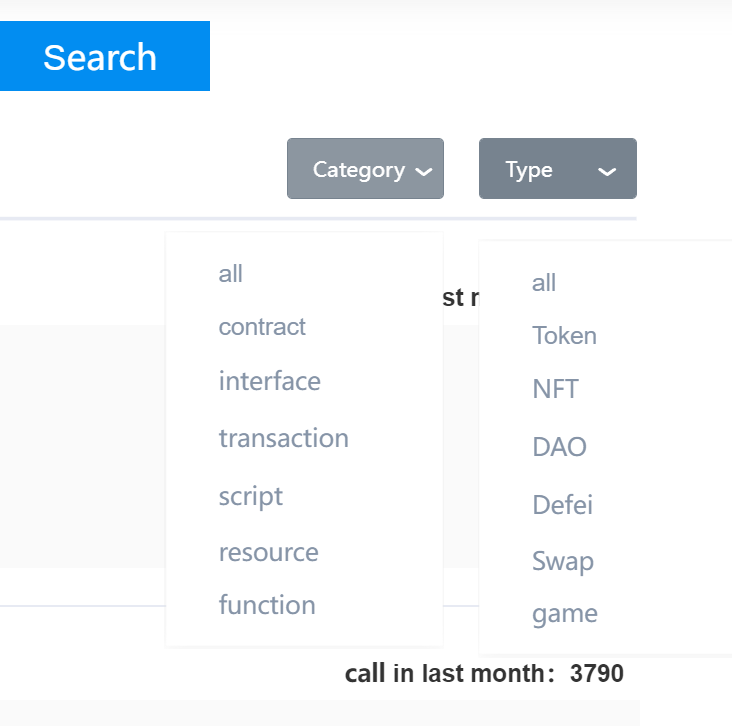
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Name | Data Type | Length | Decimals | Allow Nulls | Primary Key | Default | Description |
| 1 | id | int unsigned | 10 | 0 | N | Y |  |  |
| 2 | token\_id | varchar | 200 | 0 | Y | N |  |  |
| 3 | image\_url | varchar | 200 | 0 | Y | N |  |  |
| 4 | image\_original\_url | varchar | 200 | 0 | Y | N |  |  |
| 5 | background\_color | varchar | 200 | 0 | Y | N |  |  |
| 6 | name | varchar | 200 | 0 | Y | N |  |  |
| 7 | description | text | 65535 | 0 | Y | N |  |  |
| 8 | external\_link | varchar | 200 | 0 | Y | N |  |  |
| 9 | asset\_contract | text | 65535 | 0 | Y | N |  |  |
| 10 | owner | text | 65535 | 0 | Y | N |  |  |
| 11 | traits | text | 65535 | 0 | Y | N |  |  |
| 12 | last\_sale | varchar | 2000 | 0 | Y | N |  |  |
| 13 | fetch\_time | datetime | 19 | 0 | Y | N |  |  |
| 14 | next\_cursor | varchar | 200 | 0 | Y | N |  |  |
| 15 | page | int | 10 | 0 | Y | N |  |  |
| 16 | token\_metadata | text | 65535 | 0 | Y | N |  |  |
| 17 | permalink | varchar | 200 | 0 | Y | N |  |  |
| 18 | image\_cos\_url | varchar | 200 | 0 | Y | N |  | cos Link |
| 19 | contract\_address | varchar | 200 | 0 | Y | N |  |  |
| 20 | cid | varchar | 200 | 0 | Y | N |  | ipfs cid,image id |
| 21 | nft\_id | varchar | 100 | 0 | Y | N |  | new nft\_id |
| 22 | tx\_id | varchar | 200 | 0 | Y | N |  | transaction id of mint |
| 23 | nft\_metadata\_url | varchar | 200 | 0 | Y | N |  | nft metadata link |

## 3.4 Basic UI Design

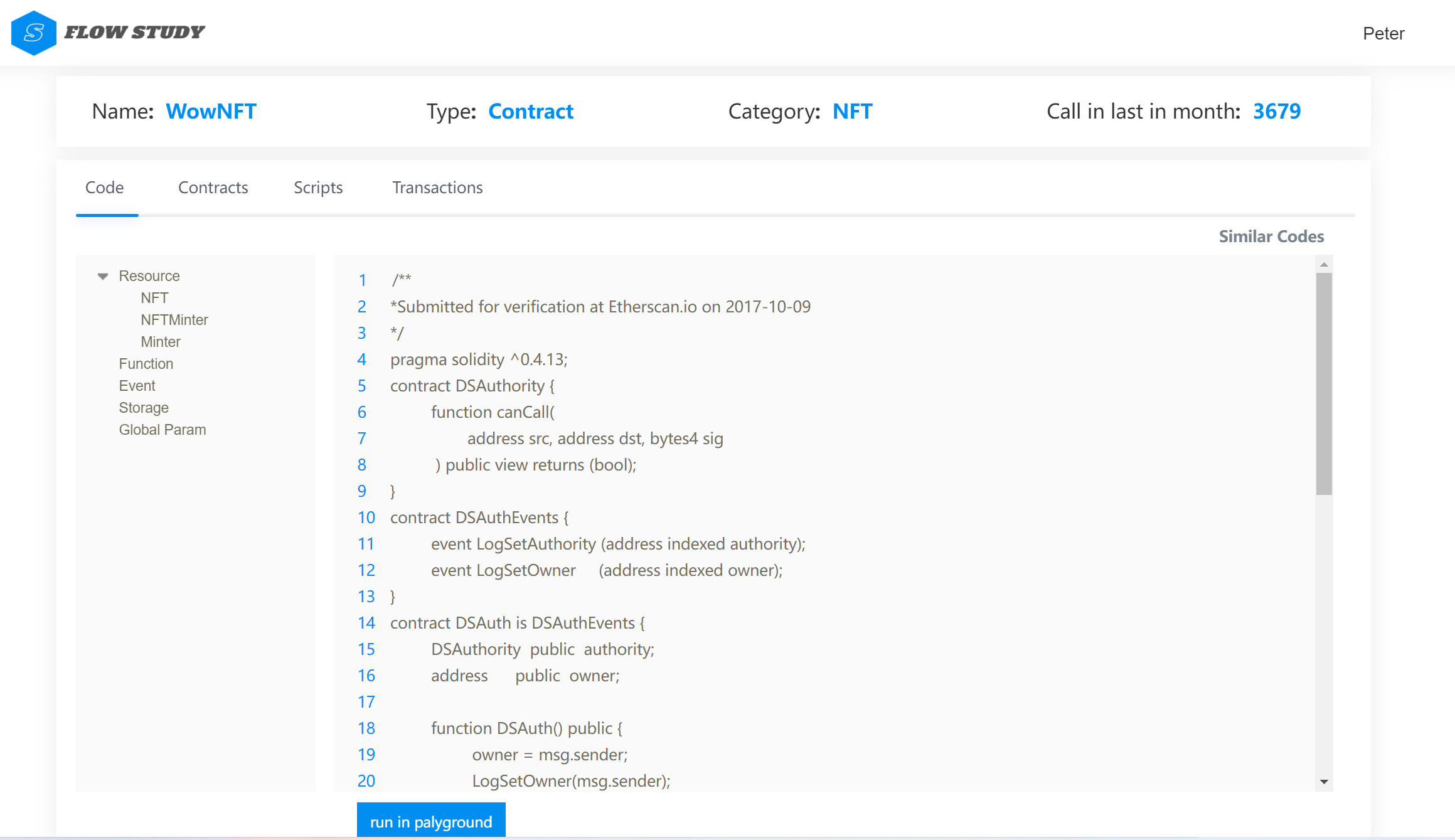
The UI pages of the main functions of the system are as follows:

3-1 search interface

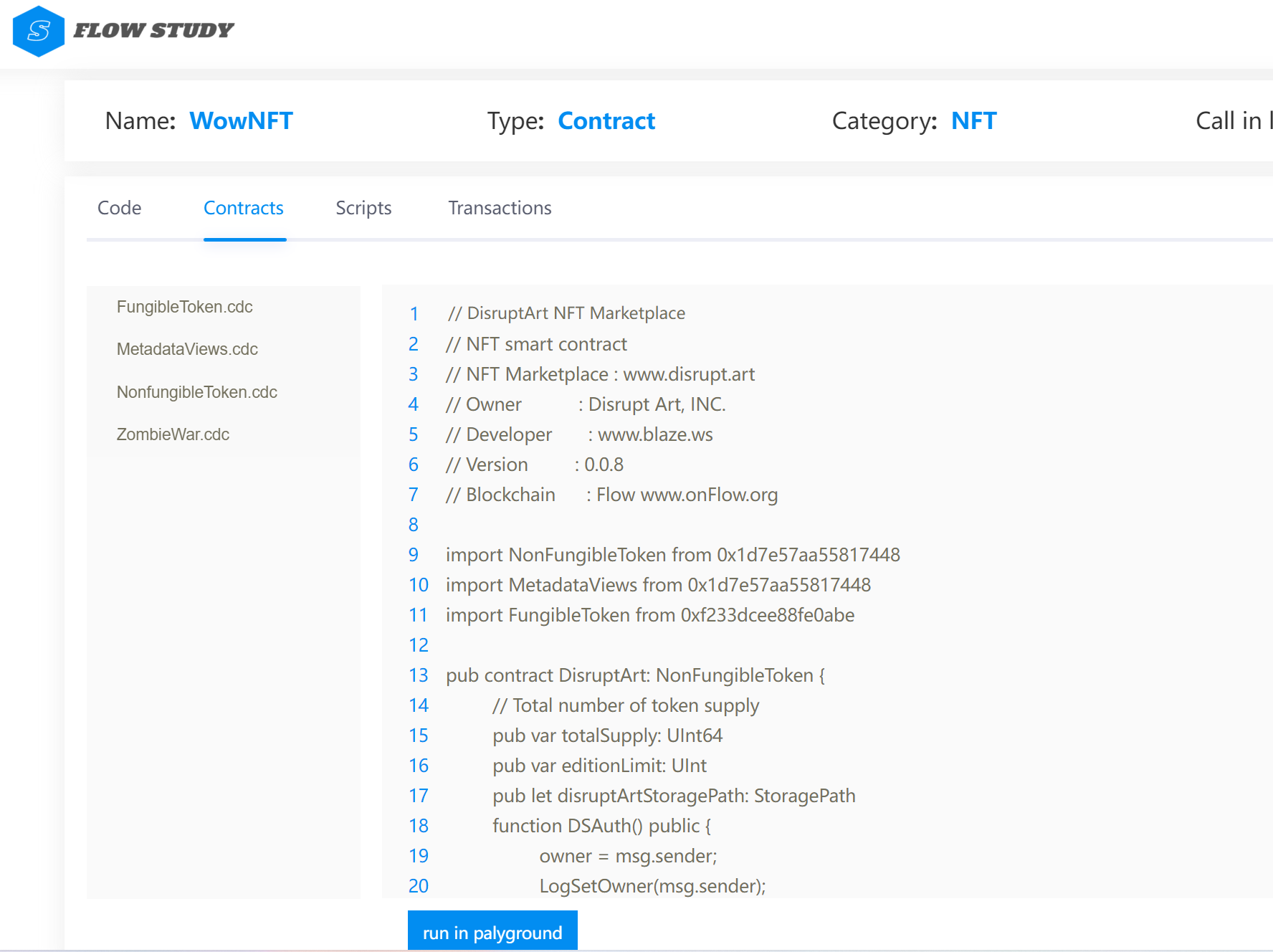


3-2 Search details

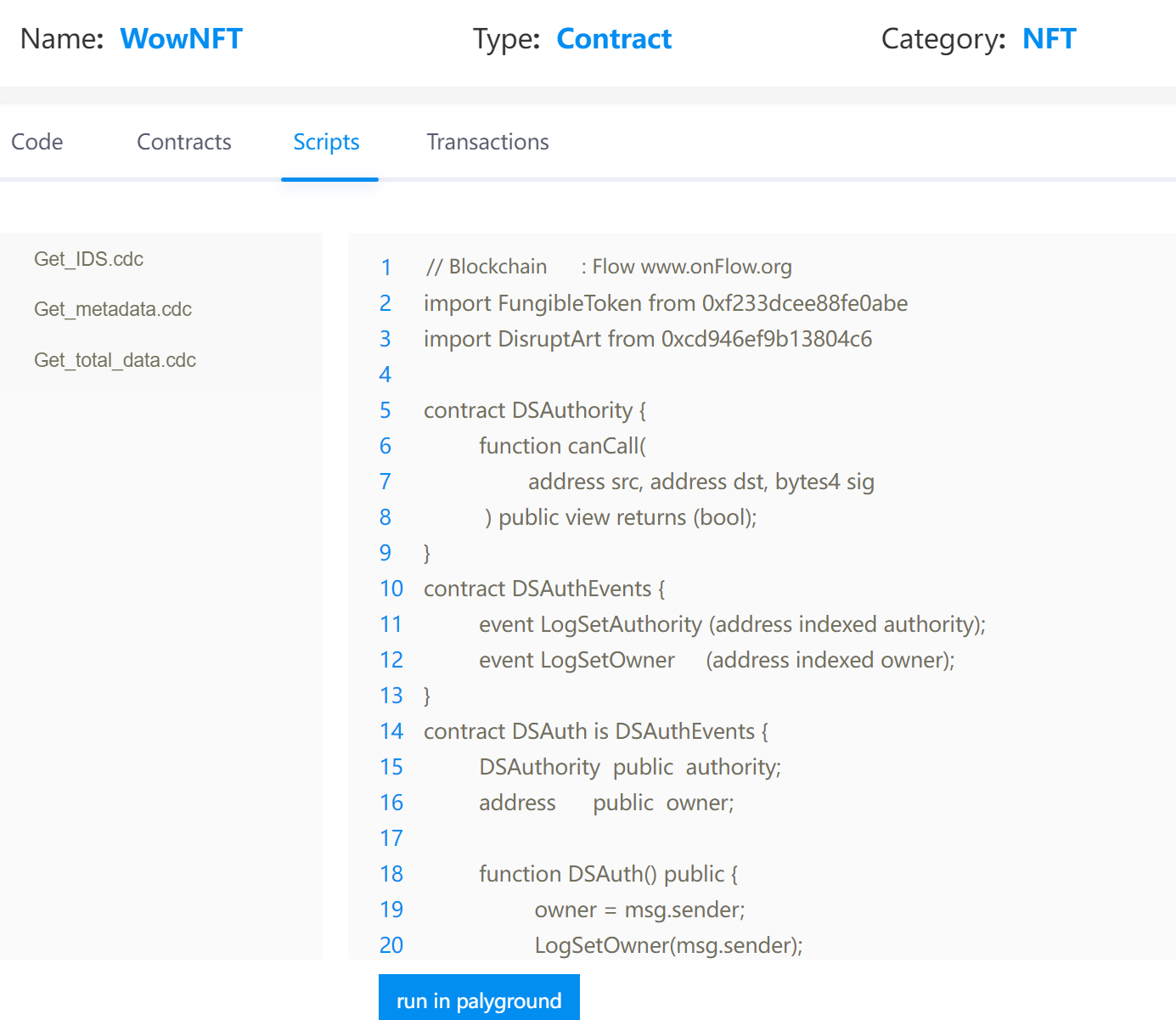
3-3 Category List



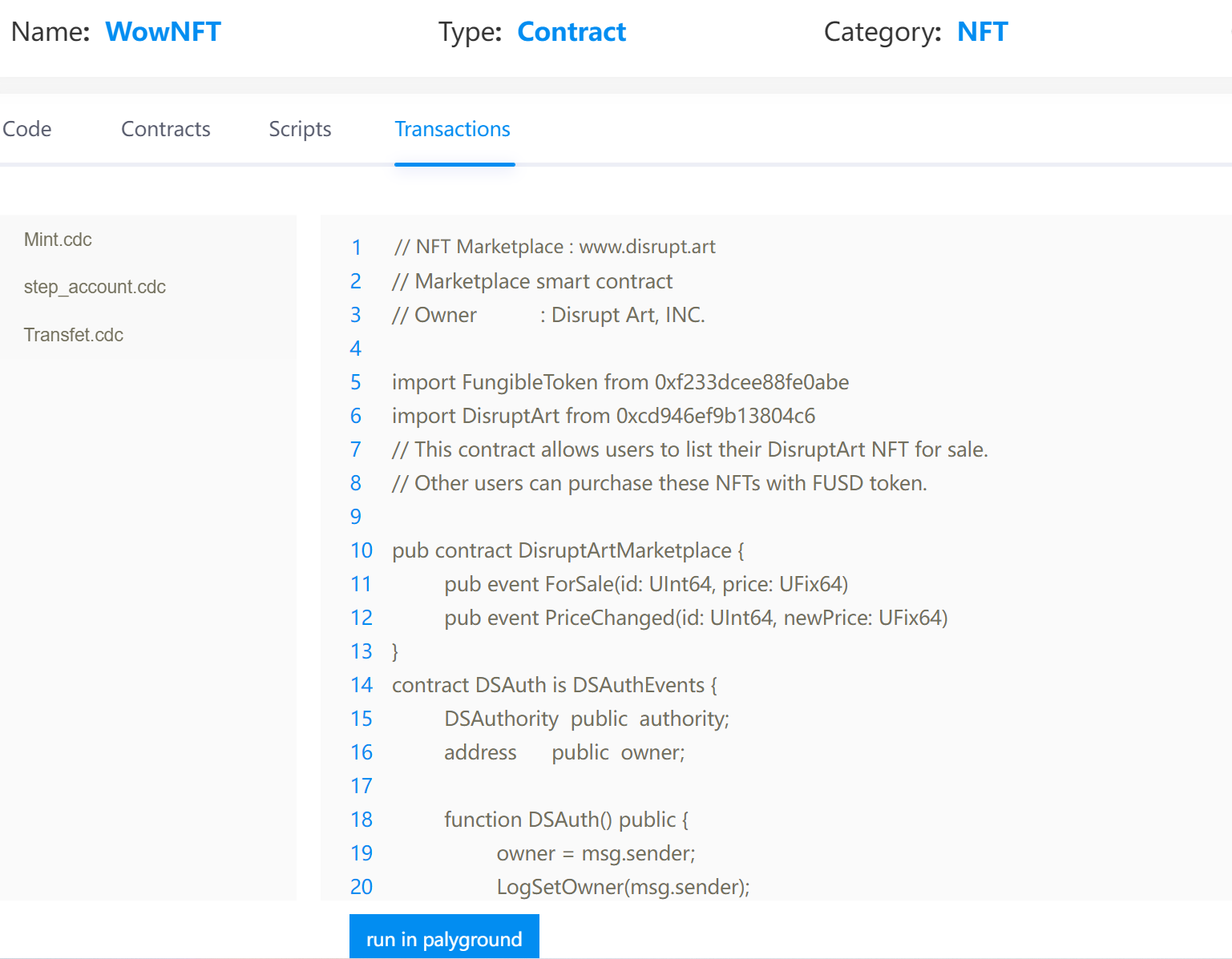
3-4 Project details page-code



3-5 Project details page-contract



3-6 Project details page-script

****

3-7 Project details page-transaction

# 4 function points

table：function points

|  |  |  |
| --- | --- | --- |
| function | function points | Function description |
| code download | Get the latest block. | Obtain the height information of the latest block through the official API. |
| Supplementary missing information. | Read the code and name of the unprocessed block. |
| Get the contract address. | Parse the transaction script to get the referenced contract address. |
| Store contract code. | Obtain the contract code through the contract address and store it in the MySQL database. |
| code search | search code with elasticsearch | MySQL database code information is synchronized to elastic search, and a search interface is provided. |
| Code classification. | Offers different types of code searches. |
| Code display. | Provides a rich code reading interface. |