hebys.

All things NFT

Hebys API Documentation v2.0

Hebys API: A Powerful Tool for Web3 and Blockchain Projects

Introduction

Hebys API is designed to simplify the complexity of the evolving digital asset ecosystem and provide powerful, user-friendly tools to developers, data analysts, and system integrators. This document will comprehensively address the basic use cases, authorization processes, and various API functions of Hebys API.

Purpose of Hebys API

Hebys API offers a flexible and reliable interface for blockchain-based projects and Web3 applications. The primary goal of the API is to enable users to conduct in-depth analyses on market trends, price changes, and asset management. This is particularly ideal for market analysis and portfolio management applications.

Target Audience

Hebys API targets the following user groups:

- **Software Developers:** Professionals developing blockchain-based applications and solutions.
- **Data Analysts:** Experts conducting in-depth analyses of crypto markets and NFT collections.
- **System Integrators:** Specialists providing data integration and workflow automation among various systems and applications.

Using Swagger UI and API Methods

Introduction to Swagger UI

Swagger UI is an interactive tool that significantly enhances your experience with the Hebys API. It provides a visual interface to interact with the API, allowing you to explore, test, and understand the API's functionalities in real-time. This powerful feature is essential for both new and experienced developers working with the Hebys API.

Accessing Swagger UI

To access the Swagger UI of the Hebys API, follow these steps:

1. **Navigate to the Swagger UI Page**:
   - Test Environment: https://testindexerapi.hebys.io/swagger/index.html
   - Prod Environment: https://searchengineapi.hebys.io/swagger/index.html
   - Upon visiting the page, you will be greeted with a list of all available API endpoints and methods.

Exploring API Methods

1. **Browse Available Methods**: Each API endpoint and method is listed with its HTTP method type (e.g., GET, POST) and a brief description.
2. **View Method Details**: Click on any method that interests you to expand its details. Here, you'll find in-depth information about the method, including its parameters, request format, and expected responses.

Making API Calls

1. **Interactive Testing**: For each method, you have the option to 'Try it out'. This allows you to make real API calls directly from the interface.
2. **Enter Required Parameters**: Provide the necessary parameters in the provided fields.
3. **Execute the Call**: Click the 'Execute' button to send the request to the API.
4. **Review the Response**: The response from the API, including the HTTP status code, message, and returned data, will be displayed on the page.

Practical Examples for Web3 Applications

The Hebys API offers a range of functionalities crucial for Web3 applications such as market analysis tools and portfolio management. These features provide real-time data on crypto assets and NFTs, assisting in informed decision-making.

Conclusion

Swagger UI is an invaluable part of the Hebys API documentation. It not only simplifies the understanding of the API's capabilities but also provides an interactive playground for testing and experimentation, ensuring that you have a hands-on experience with the API's features.

Authorization Process for Hebys API

Secure Authorization Mechanism Some functions of the Hebys API require a secure authorization mechanism due to security and data protection reasons. This section focuses on the secure authorization process and steps to obtain a Bearer Token.

Token Acquisition Process

1. **Creating and Logging into a User Account**
   - Before you start using the Hebys API, you need to create a user account and log in. To create a user account, visit https://testindexerapi.hebys.io/Auth/GetToken.
2. **Obtaining a Bearer Token**
   - Use the /Auth/GetToken endpoint to obtain a Bearer Token. This endpoint returns a Bearer Token along with the user email and password.
   - Example input: {"Email": "user_email", "Password": "user_password"}.
   - The Bearer Token is obtained for use in your API calls and is valid for a certain period.

Security Tips

- **Token Security**: Store your Bearer Token in a secure location and never share it with third parties. Transmit your token over a non-public, secure network.
- **Best Practices**: Regularly renew your token and cancel old ones. Keep your access rights at the minimum level needed and avoid granting unnecessary permissions.

General Response Model and API Calls in Hebys API

Understanding and Processing Responses from Hebys API The Hebys API employs a standard response model for all method calls. This model facilitates effective processing of responses by API users and simplifies handling of error situations.

Features and Types

- **success (boolean)**: Represents whether the operation was successful or not. It takes the value true for successful operations and false for unsuccessful ones.
- **message (string)**: A text message that describes the result of the operation. It generally contains information about whether the operation was successful or unsuccessful.
- **errors (Array of JSON Objects)**: Contains the errors that occurred during the operation. Each error is represented as a JSON object and may include an error code (code), a message (message), and a stack trace (stackTrace) of the exception situation.

- **data (JSON Object or Array)**: Represents the data obtained as a result of the operation. In successful operations, it contains data; in unsuccessful operations, it can be an empty JSON object ({}).

Response Examples

- **Successful Response Example**:

```
{
  "success": true,
  "message": "Successfully listed",
  "errors": [],
  "data": [
    { "fullName": "John Doe", "profession": "Engineer" },
    { "fullName": "Jane Doe", "profession": "Doctor" }
  ]
}
```

In this example, the success field is set to true, and the message field contains a message indicating the operation was completed successfully. The errors field is an empty array, and the data field contains the data resulting from the operation.

- **Unsuccessful Response Example**:

```
{
  "success": false,
  "message": "An error occurred",
  "errors": [
    {
      "code": "1020",
      "message": "fullName cannot be empty",
      "stackTrace": "<Stack trace of the exception situation>"
    }
  ],
  "data": {}
}
```

In this example, the success field is set to false, and the message field contains an error message indicating the operation was unsuccessful. The errors field contains details about the failure, including an error code, message, and stack trace. The data field is an empty JSON object as the operation was unsuccessful and no data was obtained.

Error Handling The Hebys API returns different error codes and messages in various error situations. Here are some common error codes and their meanings:

Error Codes and Meanings

- **401 Unauthorized:** Invalid or missing authentication credentials. Check token validity.
- **403 Forbidden:** Unauthorized access attempt. Verify your access rights.
- **404 Not Found:** The requested resource was not found. Check the URL and parameters.
- **500 Internal Server Error:** A server-side error occurred. Retry later or contact the support team.

Auto-Suggestion API: Enhancing User Search Experience with Smart Suggestions

General Description: The Auto-Suggestion API is used to retrieve a list of items such as collections, contracts, and NFTs associated with a specific search keyword. This API allows users to access search results based on a particular keyword, enriching the user's search experience and helping them find what they are looking for more effectively and quickly.

API Endpoint and Usage:

- **Endpoint URL**: https://testindexerapi.hebys.io/Token/GetSearchSuggestions
- **Method**: GET

Input Parameters:

- **text (string)**: Represents the keyword for which the search will be conducted. For example, a JSON object like {"text": "azuki"}.

Example Usage:

- **Sample Request**:

```
GET https://testindexerapi.hebys.io/Token/GetSearchSuggestions?text=azuki
```

- **Response**: The response returned by the API will be in the following format:

```
[
 {"collectionName": "Sample Collection 1", "contractName": "Sample Contract 1", "nftName": "Sample NFT 1"},
 {"collectionName": "Sample Collection 2", "contractName": "Sample Contract 2", "nftName": "Sample NFT 2"}
]
```

In this example, the text parameter "azuki" is sent, and the response returned by the API contains a list of collections, contracts, and NFTs associated with the keyword "azuki".

Collection Search API: Effective Searching and Listing of Collections

General Description: The Collection Search API is used to retrieve a list of verified collections based on specific criteria. This API enables users to search for and list collections according to specific criteria, facilitating quicker and more targeted access to desired collections.

API Endpoint and Usage:

- **Endpoint URL**: https://testindexerapi.hebys.io/Collection/Search
- **Method**: POST

Input Parameters:

- **criteria (JSON Object)**: Contains the search criteria. These include:
    - **isVerified (boolean)**: Used to search for verified collections. If set to true, only verified collections are returned.
    - **searchKeyword (string)**: Represents the search keyword.
    - **chainIds (array)**: Contains the identifiers of relevant chains, e.g., [1, 56] for Ethereum and Binance Smart Chain.
    - **sortBy (integer)**: Specifies the sorting criterion of the results, e.g., 0 represents sorting by Relevance.
- **pagination (JSON Object)**: Contains pagination settings. These include:
    - **currentPage (integer)**: Specifies the number of the current page.
    - **maxRowsPerPage (integer)**: Specifies the maximum number of collections to be displayed per page.

Example Usage:

- **Sample Request**:

```
        POST https://testindexerapi.hebys.io/Collection/Search
        Content-Type: application/json

        {
         "criteria": {
          "isVerified": true,
          "searchKeyword": "azuki",
          "chainIds": [1, 56],
          "sortBy": 0
         },
         "pagination": {
          "currentPage": 1,
          "maxRowsPerPage": 10
         }
}
```

- **Response**: The response returned by the API will include a list of collections based on the search results. The exact structure of the response is determined by the specific API and needs to be processed by users.

Additional Example API Call:
For making a sample API call, you can use the following URL:
[https://hebys.io/collections?status=["VerifiedOnly"]&sortout=["Relevancy"]&q=azuki&chain=["Ethereum","BinanceSmartChain"]](https://hebys.io/collections?status=["VerifiedOnly"]&sortout=["Relevancy"]&q=azuki&chain=["Ethereum","BinanceSmartChain"])

Purpose of Collection Search API: The Collection Search API provides users the ability to search and list NFT collections based on specific criteria. This API allows querying collections by their names, creators, or other characteristics, thereby enabling users to find their desired collections more efficiently.

Collection Get API: Accessing Detailed Information of a Specific Collection

General Description: The Collection Get API enables users to obtain details about a specific collection. This API allows users to access detailed information about a particular collection, including its name, creator, total number of items, characteristics, and statistics.

API Endpoint and Usage:

- **Endpoint URL**: https://testindexerapi.hebys.io/Collection/Get
- **Method**: POST

Input Parameters:

- **id (string)**: Represents the unique identifier of the collection. For example, a JSON object like {"id": "1/0xed5af388653567af2f388e6224dc7c4b3241c544/1"}**.**

Example Usage:

- **Sample Request**:

```
POST https://testindexerapi.hebys.io/Collection/Get
Content-Type: application/json

{
  "id": "1/0xed5af388653567af2f388e6224dc7c4b3241c544/1"
}
```

- **Response**: The response from the API will include the details of the specified collection. The exact structure of the response is determined by the API.

Additional Endpoint for Detailed Collection Information:

- **Endpoint URL**: https://testindexerapi.hebys.io/Token/GetCollectionDetails
- **Method**: POST
- **Input Parameters**:
  - **collectionId (string)**: Represents the identity number of the desired collection.

Example Usage for Detailed Information:

- **Sample Request**:

```
http
POST https://testindexerapi.hebys.io/Token/GetCollectionDetails
{
  "collectionId": "123456"
}
```

- **Response**: The response will include detailed information about the collection corresponding to the specified collection ID, such as the collection name, creator, total items, a detailed description of the collection, features, and statistics.

Purpose of Collection Get API:

The Collection Get API provides users with the ability to access detailed information about a specific NFT collection. This API offers both basic information such as the collection's name, creator, and total item count, as well as more detailed information including the collection's characteristics and statistics.

This section provides information on how to use the Collection Get API and how users can access detailed information about a specific collection.

NFT Search API: Detailed Criteria-Based Searching and Listing of NFTs

General Description: The NFT Search API is used to obtain a list of NFTs based on specific criteria. This API allows users to search for and list NFTs according to various criteria, enabling them to find the NFTs they are looking for more quickly and effectively.

API Endpoint and Usage:

- **Endpoint URL**: https://testindexerapi.hebys.io/Token/Search
- **Method**: POST

Input Parameters:

- **criteria (JSON Object)**: Contains the search criteria, including:
    - **searchKeyword (string)**: Represents the keyword for the search.
    - **chainIds (array)**: Contains identifiers of relevant chains, e.g., [1, 43114] for Ethereum and Avalanche.
    - **verifiedOnly (boolean)**: Used to search only for verified NFTs.
    - **newOnly (boolean)**: Used to search only for new NFTs.
    - **buyNowOnly (boolean)**: Used to search only for NFTs available for immediate purchase.
    - **onAuctionOnly (boolean)**: Used to search only for NFTs available on auction.
    - **hasOffersOnly (boolean)**: Used to search for NFTs with offers.
    - **isOnlyHebysMarket (boolean)**: Used to search for NFTs listed only on Hebys Market.
    - **sortBy (integer)**: Specifies the sorting criterion, e.g., 0 for relevance.
- **pagination (JSON Object)**: Contains pagination settings, including:
    - **currentPage (integer)**: Specifies the current page number.
    - **maxRowsPerPage (integer)**: Specifies the maximum number of NFTs per page.

Example Usage:

- **Sample Request:**

```
POST https://testindexerapi.hebys.io/Token/Search
Content-Type: application/json

{
  "criteria": {
    "searchKeyword": "azuki",
    "chainIds": [1, 43114],
    "verifiedOnly": true,
    "newOnly": false,
    "buyNowOnly": true,
    "onAuctionOnly": false,
    "hasOffersOnly": false,
    "isOnlyHebysMarket": false,
    "sortBy": 0
  },
  "pagination": {
    "currentPage": 1,
    "maxRowsPerPage": 10
  }
}
```

- **Response**: The response from the API will include a list of NFTs filtered according to the specified criteria.

Additional Endpoint for NFT Search:

- **Endpoint URL**: https://testindexerapi.hebys.io/Token/GetNFTSearch
- **Method**: POST
- **Input Parameters**:
  - **query (string)**: Search query for NFT name, creator, or other characteristics.
  - **limit (int)**: Limit for the number of results to return.
  - **offset (int)**: Determines the starting record for the search.

Example Usage for Detailed Search:

- **Sample Request**:

```
POST https://testindexerapi.hebys.io/Token/GetNFTSearch
{
  "query": "CryptoPunk",
  "limit": 10,
  "offset": 0
}
```

- **Response**: The response will include a list of NFTs that meet the specified search criteria.

Purpose of NFT Search API: The NFT Search API offers users the ability to search and list NFTs based on various criteria, including names, creators, ownership status, and other features. This functionality enables users to conduct more targeted and effective searches for NFTs.

NFT Get API: Accessing Detailed Information of a Specific NFT

General Description: The NFT Get API enables users to obtain detailed information about a specific NFT. This API allows users to access in-depth details of a particular NFT, enhancing their understanding and interaction with the asset.

API Endpoint and Usage:

- **Endpoint URL**: https://testindexerapi.hebys.io/Token/GetDetail
- **Method**: POST

Input Parameters:

- **id (string)**: Represents the unique identifier of the NFT. For example, a JSON object like {"id": "1/0xff9c1b15b16263c61d017ee9f65c50e4ae0113d7/lootproject/4125"}.

Example Usage:

- **Sample Request**:

```
POST https://testindexerapi.hebys.io/Token/GetDetail
Content-Type: application/json

{
  "id": "1/0xff9c1b15b16263c61d017ee9f65c50e4ae0113d7/lootproject/4125"
}
```

- **Response**: The response from the API will include the details of the specified NFT. The exact structure of the response is determined by the API and needs to be processed by users.

Example API Call: To make a sample API call, you can use the following URL: https://hebys.io/asset/1/0x6fc355d4e0ee44b292e50878f49798ff755a5bbc/deadheads/3514