

# Quickstart guide

## How to use the 1inch Token API with Python and Requests

In this tutorial, we will walk you through the process of using the 1inch Token API with Python and the `requests` library. We'll cover how to search for tokens, get detailed information about individual tokens, and obtain a list of supported tokens from the 1inch API.

Now, let's get started!

### Step 1: Prerequisites

#### Python

Make sure you have Python installed on your system. If you don't, follow the [official instructions](#).

#### Requests library

This script uses the `requests` library to make HTTP requests to the 1inch Token API. Install it by running:

```
```bash
pip install requests
```

```
```
```

### Set up environment variables

To securely manage your API key, store it as an environment variable. For macOS:

1. Open the Terminal.
2. Add `API_KEY` variable to your shell configuration file. Replace `your_api_key` with your DevPortal API key:

```
export API_KEY=your_api_key
```

3. Save and reload your configuration file:

```
source ~/.zshrc # For zsh (default for macOS)
```

## Prepare for rate limits

The script includes mechanisms to handle API rate limits. It introduces small delays between API calls using the time library. You don't need to install `time`, as it's part of Python's standard library.

The `time.sleep(1)` function pauses the script for one second between calls, ensuring compliance with API restrictions.

## Organize imports

Now, the opening part of your script should look something like this:

```
import requests # For making API requests
import time     # For introducing delays between requests (to respect API rate limits)
import os       # For accessing environment variables
```

## Step 2: Define the API base URL and headers

Next, we'll define the base URL and headers for the 1inch Token API. We'll use this URL to construct our API requests. Additionally, we make sure `API_KEY` is retrieved from the environment variable.

```
API_KEY = os.getenv("API_KEY")
BASE_URL = "https://api.1inch.dev/token"
HEADERS = {
    "Authorization": f"Bearer {API_KEY}",
    "accept": "application/json"
}
```

## Step 3: Search for tokens

To search for tokens, we'll use the `/v1.2/{chain_id}/search` endpoint. We'll provide a query in the name, symbol, or description of the token to search for matches.

```
def search_tokens(query, chain_id, limit=10, ignore_listed="false"):
    endpoint = f"{BASE_URL}/v1.2/{chain_id}/search"
    params = {
        "query": query,
        "limit": limit,
        "ignore_listed": ignore_listed
    }
    response = requests.get(endpoint, headers=HEADERS, params=params)
    if response.status_code == 200:
        return response.json()
```

```

else:
    print(f"Failed to search tokens. Status code: {response.status_code}")
    return None

```

## Step 4: Get detailed information about specific tokens

To get detailed information about specific tokens, we'll use the

`/v1.2/{chain_id}/custom/{addresses}` endpoint. We'll provide a list of token addresses for which we want to obtain information.

```

def get_tokens_info(chain_id, addresses):
    endpoint = f"{BASE_URL}/v1.2/{chain_id}/custom/{','.join(addresses)}"
    response = requests.get(endpoint, headers=HEADERS )
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Failed to get tokens info. Status code: {response.status_code}")
        return None

```

## Step 5: Get all tokens info

To get information about all tokens supported by 1inch on a specific network, we'll use the

`/v1.2/{chain_id}` endpoint.

```

def get_all_tokens_info(chain_id, provider="1inch"):
    endpoint = f"{BASE_URL}/v1.2/{chain_id}"
    params = {
        "provider": provider,
    }
    response = requests.get(endpoint, headers=HEADERS, params=params)
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Failed to get all tokens info. Status code: {response.status_code}")
        return None

```

## Step 6: Get 1inch token list

To get the list of 1inch tokens, we'll use the `/v1.2/{chain_id}/token-list` endpoint.

```

def get_1inch_token_list(chain_id, provider="1inch"):
    endpoint = f"{BASE_URL}/v1.2/{chain_id}/token-list"
    params = {
        "provider": provider,
    }
    response = requests.get(endpoint, headers=HEADERS, params=params)
    if response.status_code == 200:

```

```

return response.json()
else:
    print(f"Failed to get 1inch token list. Status code: {response.status_code}")
    return None

```

## Step 7: Putting it all together

Now, let's put it all together and use the functions to interact with the 1inch Token API.

```

if __name__ == "__main__":
    # Step 3: Search for tokens
    search_query = "1inch"
    chain_id = 1 # Replace with the chain ID you want to search on
    search_results = search_tokens(search_query, chain_id)
    print("Search Results:")
    print(search_results)
    # sleep one second because of RPS limit
    time.sleep(1)

    # Step 4: Get detailed information about specific tokens
    token_addresses = ["0x11111111117dc0aa78b770fa6a738034120c302"] # Replace with token addresses yo
u want to query
    tokens_info = get_tokens_info(chain_id, token_addresses)
    print("Tokens Info:")
    print(tokens_info)
    # sleep one second because of RPS limit
    time.sleep(1)

    # Step 5: Get information about all tokens on a token list
    all_tokens_info = get_all_tokens_info(chain_id)
    print("All Tokens Info:")
    print(all_tokens_info)
    # sleep one second because of RPS limit
    time.sleep(1)

    # Step 6: Get 1inch token list
    token_list = get_1inch_token_list(chain_id)
    print("1inch Token List:")
    print(token_list)

```

That's it! You've successfully used the 1inch Token API with Python and the `requests` library to search for tokens, get detailed token information, and obtain a list of supported tokens from 1inch. Happy coding!

## Full script

Below you can find the full script with consideration of the default RPS limit:

```

import requests
import time
import os

```

```

API_KEY = os.getenv("API_KEY")
BASE_URL = "https://api.1inch.dev/token"
HEADERS = {
    "Authorization": f"Bearer {API_KEY}",
    "accept": "application/json"
}

```

# for a list of providers check <https://tokenlists.org/>

```

def search_tokens(query, chain_id, limit=10, ignore_listed="false"):
    endpoint = f"{BASE_URL}/v1.2/{chain_id}/search"
    params = {
        "query": query,
        "limit": limit,
        "ignore_listed": ignore_listed
    }
    response = requests.get(endpoint, headers=HEADERS, params=params)
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Failed to search tokens. Status code: {response.status_code}")
        return None

def get_tokens_info(chain_id, addresses):
    endpoint = f"{BASE_URL}/v1.2/{chain_id}/custom/{','.join(addresses)}"
    response = requests.get(endpoint, headers=HEADERS )
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Failed to get tokens info. Status code: {response.status_code}")
        return None

def get_all_tokens_info(chain_id, provider="1inch"):
    endpoint = f"{BASE_URL}/v1.2/{chain_id}"
    params = {
        "provider": provider,
    }
    response = requests.get(endpoint, headers=HEADERS, params=params)
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Failed to get all tokens info. Status code: {response.status_code}")
        return None

def get_1inch_token_list(chain_id, provider="1inch"):
    endpoint = f"{BASE_URL}/v1.2/{chain_id}/token-list"
    params = {
        "provider": provider,
    }
    response = requests.get(endpoint, headers=HEADERS, params=params)
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Failed to get 1inch token list. Status code: {response.status_code}")
        return None

```

```
if __name__ == "__main__":
    # Step 3: Search for tokens
    search_query = "1inch"
    chain_id = 1 # Replace with the chain ID you want to search on
    search_results = search_tokens(search_query, chain_id)
    print("Search Results:")
    print(search_results)
    # sleep one second because of RPS limit
    time.sleep(1)

    # Step 4: Get detailed information about specific tokens
    token_addresses = ["0x11111111117dc0aa78b770fa6a738034120c302"] # Replace with token addresses yo
u want to query
    tokens_info = get_tokens_info(chain_id, token_addresses)
    print("Tokens Info:")
    print(tokens_info)
    # sleep one second because of RPS limit
    time.sleep(1)

    # Step 5: Get information about all tokens on a token list
    all_tokens_info = get_all_tokens_info(chain_id)
    print("All Tokens Info:")
    print(all_tokens_info)
    # sleep one second because of RPS limit
    time.sleep(1)

    # Step 6: Get 1inch token list
    token_list = get_1inch_token_list(chain_id)
    print("1inch Token List:")
    print(token_list)
```

Previous

[< Introduction](#)

Next

[1inch whitelisted multi-chain tokens >](#)