StormGain public API

CoinGecko standards

Introduction

This document covers API endpoint details needed for integration of cryptocurrency exchange or lending/borrowing platforms on CoinGecko. Cryptocurrency exchange teams/operators are expected to support all required endpoints to ensure uptime and data availability.

Please refer to **Section A for Spot Exchanges API** requirements, **Section B for Derivatives Exchanges API** and **Section C for Earn Platforms.**

Generally, exchanges are divided into two main categories - (i) Spot Exchanges and (ii) Derivatives Exchanges.

For a list of platforms under CoinGecko Earn, please refer to our Earn page (DAI).

For applications, please ensure that you are submitting your applications to the CoinGecko Application Form.

Standards and conventions

You'll find some of the general requirements by CoinGecko to ensure smooth integration and data availability. The integration process may be affected if an exchange is unable to fulfill the requirements.

1. No API authentication required on public market endpoints.
2. Full market queries are available every 60 seconds. Updates are available once per minute, per market pair.
3. Data is available in JSON format
4. Versioning is guaranteed to avoid breaking changes (api/v1/asset, api/v2/asset, etc.)

Base URL

Use this public accessible url for query data: **https://public-api.stormgain.com**

A. Spot Exchanges

Spot Exchanges - Endpoints Overview

No.	Endpoint	Description
1.	/api/v1/cg/spot/pairs	Details on cryptoassets traded on an exchange.
2.	/api/v1/cg/spot/tickers	Market related statistics for all markets for the last 24 hours.
3.	/api/v1/cg/spot/orderbook/	Order book depth of any given trading pair, split into two different arrays for bid and ask orders.
4.	/api/v1/cg/spot/historical_trades	Historical trade data for any given trading pair.

Endpoint 1 - /pairs (Overview)

The /pairs endpoint provides a summary on cryptoasset trading pairs available on the exchange. For example, for Bitcoin (BTC):


```
{
  “ticker_id”: “BTC_ETH”,
  "base": "BTC",
  "target": "ETH",
}
```


/pairs endpoint response description:

Name	Data Type	Category	Description
ticker_id	string	Mandatory	Identifier of a ticker with delimiter to separate base/target, eg. BTC_ETH
base	string	Mandatory	Symbol/currency code of a the base cryptoasset, eg. BTC
target	string	Mandatory	Symbol/currency code of the target

			cryptoasset, eg. ETH

Endpoint 2 - /tickers (Market Info)

The /tickers endpoint provides 24-hour pricing and volume information on each market pair available on an exchange.

```
{
    "ticker_id": "BTC_ETH",
    "base_currency": "BTC",
    "target_currency": "ETH",
    "last_price":"50.0",
    "base_volume":"10",
    "target_volume":"500",
    "bid":"49.9",
    "ask":"50.1",
    "high":”51.3”,
    “low”:”49.2”,
}
```

/tickers endpoint response description:

Name	Data Type	Category	Description
ticker_id	string	Mandatory	Identifier of a ticker with delimiter to separate base/target, eg. BTC_ETH
base_currency	string	Mandatory	Symbol/currency code of base pair, eg. BTC
target_currency	string	Mandatory	Symbol/currency code of target pair, eg. ETH
last_price	decimal	Mandatory	Last transacted price of base currency based on given target currency
base_volume	decimal	Mandatory	24 hour trading volume in base pair volume
target_volume	decimal	Mandatory	24 hour trading volume in target pair volume

bid	decimal	Mandatory	Current highest bid price
ask	decimal	Mandatory	Current lowest ask price
high	decimal	Mandatory	Rolling 24-hours highest transaction price
low	decimal	Mandatory	Rolling 24-hours lowest transaction price

Endpoint 3 - /orderbook (Order book depth details)

The /orderbook/ticker_id endpoint is to provide order book information with at least depth = 100 (50 each side) returned for a given market pair/ticker.

Endpoint parameters:

Name	Type	Status	Description
ticker_id	string	Mandatory	A ticker such as "BTC_ETH", with delimiter between different cryptoassets
depth	integer	Recommended	Orders depth quantity: [0, 100, 200, 500...]. 0 returns full depth. Depth = 100 means 50 for each bid/ask side. Note that for more liquid or closely priced pairs, the lack of order depth may result in miscalculation of depth/spread.

Example query:
.../api/orderbook?ticker_id=BTC_ETH&depth=200


```
{
  "ticker_id": "BTC_ETH",
  "timestamp":"1700050000",
  "bids":[
    [
      "49.8",
      "0.50000000"
    ],
    [
      "49.9",
      "6.40000000"
    ]
  ],
```

```
  "asks":[
    [
      "50.1",
      "9.20000000"
    ],
    [
      "50.2",
      "7.9000000"
    ]
  ]
}
```

Order book response descriptions:

Name	Data Type	Category	Description
ticker_id	string	Mandatory	A pair such as "BTC_ETH", with delimiter between different cryptoassets
timestamp	timestamp	Recommended	Unix timestamp in milliseconds for when the last updated time occurred.
bids	decimal	Mandatory	An array containing 2 elements. The offer price and quantity for each bid order
asks	decimal	Mandatory	An array containing 2 elements. The ask price and quantity for each ask order

Endpoint 4 - /historical_trades (Historical Data)

The /historical_trades/ticker_id is used to return data on historical completed trades for a given market pair for the 24hours depth

Endpoint parameters:

Name	Data Type	Category	Description
ticker_id	string	Mandatory	A pair such as "BTC_ETH", with delimiter between different cryptoassets
type	string	Mandatory	To indicate nature of trade - buy/sell
limit	integer	Recommended	Number of historical trades to retrieve from time of query. [0, 200, 500...]. 0 returns full history.

start_time	date	Recommended	Start time from which to query historical trades from
end_time	date	Recommended	End time for historical trades query

Example query:
.../api/historical_trades?ticker_id=BTC_ETH&limit=10

```
“buy”: [
  {
    "trade_id":1234567,
    "price":"50.1",
    "base_volume":"0.1",
    "target_volume":"1",
    "trade_timestamp":"1700050000",
    "type":"buy"
  }
],
“Sell”: [
  {
    "trade_id":1234567,
    "price":"50.1",
    "base_volume":"0.1",
    "target_volume":"1",
    "trade_timestamp":"1700050000",
    "type":"sell"
  }
]
```

/historical_trades response descriptions:

Name	Data Type	Category	Description
trade_id	integer	Mandatory	A unique ID associated with the trade for the currency pair transaction Note: Unix timestamp does not qualify as trade_id.
price	decimal	Mandatory	Transaction price in base pair volume.
base_volume	decimal	Mandatory	Transaction amount in base pair volume.
target_volume	decimal	Mandatory	Transaction amount in target pair volume.

trade_timestamp	timestamp	Mandatory	Unix timestamp in milliseconds for when the transaction occurred.
type	string	Mandatory	Used to determine the type of the transaction that was completed. Buy – Identifies an ask that was removed from the order book. Sell – Identifies a bid that was removed from the order book.

B. Derivative Exchanges

Derivative Exchanges - Endpoints Overview

For CoinGecko Integration, 3 separate endpoints must be available:

No.	Endpoint	Description
1.	/api/v1/cg/derivatives/contracts	Summary of contracts traded on the exchange, helps to differentiate between different products available.
2.	/contract_specs	Combined with Endpoint 1 (/contracts)
3.	/api/v1/cg/derivatives/orderbook/	Order book depth of any given trading pair, split into two different arrays for bid and ask orders.

Endpoint 1 - /contracts (info & types of products available)

The /contracts endpoint provides a summary of all contracts traded on the exchange. There should be a clear distinction between the type of contracts such as perpetual, futures, options, etc. See below for specifications of data required for the /contracts endpoint:

Name	Data Type	Category	Description
ticker_id	string	Mandatory	Identifier of a ticker with delimiter to separate base/target, eg. BTC-PERP
base_currency	string	Mandatory	Symbol/currency code of base pair, eg. BTC
target_currency	string	Mandatory	Symbol/currency code of target pair, eg. ETH
last_price	decimal	Mandatory	Last transacted price of base currency based on given target currency
base_volume	decimal	Mandatory	24 hour trading volume in base pair volume

target_volume	decimal	Mandatory	24 hour trading volume in target pair volume
bid	decimal	Mandatory	Current highest bid price
ask	decimal	Mandatory	Current lowest ask price
high	decimal	Mandatory	Rolling 24-hours highest transaction price
low	decimal	Mandatory	Rolling 24-hours lowest transaction price
product_type	string	Mandatory	What product is this? Futures, Perpetual, Options?
open_interest	decimal	Mandatory	The open interest in the last 24 hours in contracts
index_price	decimal	Mandatory	Underlying index price
index_name	string	Mandatory	Name of the underlying index if any
index_currency	string	Mandatory	Underlying currency for index
start_timestamp	integer	Mandatory	Starting of this derivative product (relevant for expirable futures or options)
end_timestamp	integer	Mandatory	Ending of this derivative product (relevant for expirable futures or options)
funding_rate	decimal	Mandatory	Current funding rate
next_funding_rate	decimal	Mandatory	Upcoming predicted funding rate
next_funding_rate_timestamp	integer	Mandatory	Timestamp of the next funding rate change

Endpoint 2 - /contract_specs (Contract specifications)

Describes the specification of the contracts, mainly the pricing of the contract and its type (vanilla, inverse, or quanto). See below for specifications of data required for the /contract_specs endpoint:

Name	Data Type	Category	Description
contract_type	string	Mandatory	Describes the type of contract - Vanilla, Inverse or Quanto?
contract_price	decimal	Mandatory	Describes the price per contract.
contract_price_currency	string	Mandatory	Describes the currency which the contract is priced in.

Endpoint 3 - /orderbook (Order book depth details)

The /orderbook/ticker_id endpoint is to provide order book information with at least depth = 100 (50 each side) returned for a given market pair/ticker. See below for specifications of data required for the /orderbook endpoint:

Name	Data Type	Category	Description
ticker_id	string	Mandatory	A pair such as "BTC-PERP", with delimiter between different cryptoassets
timestamp	timestamp	Recommended	Unix timestamp in milliseconds for when the last updated time occurred.
bids	decimal	Mandatory	An array containing 2 elements. The offer price and quantity for each bid order
asks	decimal	Mandatory	An array containing 2 elements. The ask price and quantity for each ask order

Order book depth of any given trading pair, split into two different arrays for bid and ask orders.

(Similar to endpoint for spot markets, kindly refer to Section A Endpoint 3 for extra information).

C. Earn Platforms

CoinGecko Earn Platforms - Endpoints Overview

For CoinGecko Earn Integration, 1 endpoint is required:

No.	Endpoint	Description
1.	/assets	Details on cryptoassets supported by platform, along with the borrow/lending rates

Endpoint 1 - /assets (info of products available)

The /assets endpoint provides a summary of all assets supported by a platform. It also details the lending & borrow rates of a given asset on the platform. See below for specifications of data required for the /assets endpoint:

Name	Data Type	Category	Description
asset_symbol	string	Mandatory	Identifier of an asset, eg. BTC
asset_name	string	Mandatory	Name of the asset, eg. Bitcoin
lending_rate	decimal	Mandatory	This shows how much a user will earn for providing funds on a yearly basis (APR)
borrow_rate	decimal	Optional	This shows how much a user will pay from borrowing funds on a yearly basis (APR)
start_date	integer	Optional	Timestamp - start date if lending product is flexible deposit type
end_date	integer	Optional	Timestamp - end date if lending product is flexible deposit type