REST API for Algorithmic Trading

Jay Parmar

Objective

What will you be able to do after this session?

- Understand how the REST API works
- Know to work with web sockets in Python
- Trade algorithmically

Agenda

What will you learn in this session?

- Fundamental building blocks of REST API
  - URL Components
  - HTTP Methods
  - Headers, Body, Statuses
- Perform various trading operations using the REST API
  - Load contract details
  - Fetch historical data
  - Place orders
  - Fetch account information
  - Work with streaming data

Why?

Why REST API?

It is supported by...

- Interactive Brokers
- E*TRADE
- TD Ameritrade
- Alpaca
- Saxo Bank
- IG Markets
- Questrade
- Binance
- Kraken
- Bitfinex
- KuCoin
- Coinbase
- CoinDCX
- Bitstamp
- Zerodha
- 5paisa
- IIFL Securities
- FYERS
- Angel One
- COMPOSITEDGE
- AliceBlue
- ICICI Direct
- Trade Smart
- Finvasia
- Kotak Securities
- mastertrust
- FXCM
- TradeStation

What it is NOT.

What to expect from REST API

Pointers to keep in mind

- REST API's are usually **not** meant for HFT trading.
- Most suitable for LFT or MFT trading.
- Most brokers would have API limits in place. Ex. : 60 API calls in a minute.
- Depending on how a broker implements the REST API, all functionalities (provided by the broker) might not be available.
- REST API is not the replacement of existing trading avenues, rather complements them.

What is an API?

What is an API?

Application Programming Interface

- Interface: Acts as a medium of communication between different software applications. Usually, it involves a client (who requests information) and server (who responds).

What is an API?

Application Programming Interface

- Interface: Acts as a medium of communication between different software applications. Usually, it involves a client (who requests information) and server (who responds).
- Programming: It has a structured way of communicating. Requests and responses can be programmed.

What is an API?

Application Programming Interface

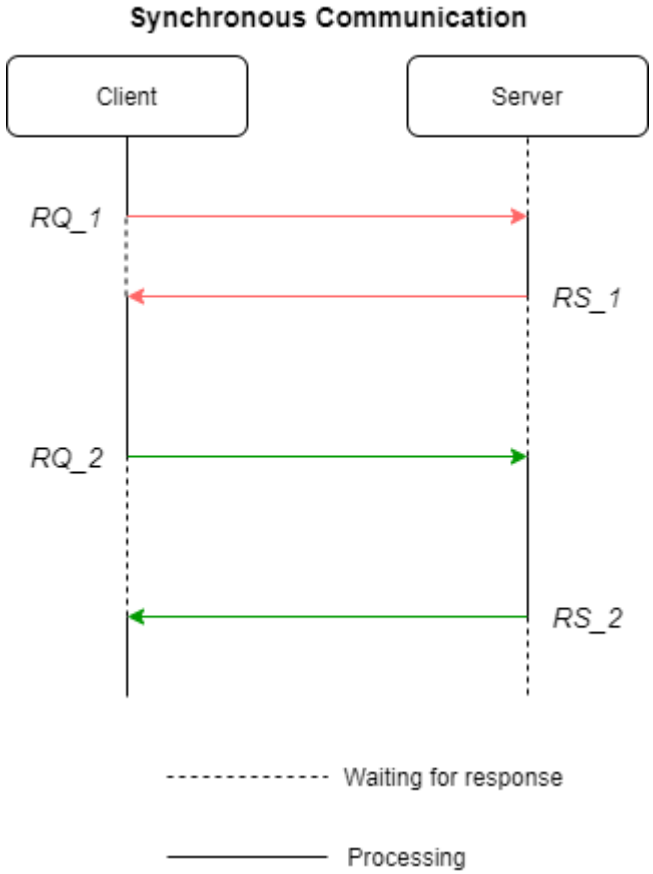
- Interface: Acts as a medium of communication between different software applications. Usually, it involves a client (who requests information) and server (who responds).
- Programming: It has a structured way of communicating. Requests and responses can be programmed.
- Application: A piece of software.

What is REST?

What is REST?

REpresentational State Transfer

- A philosophy on which API can be built
- Majorly implemented on client server architecture
- Proposes the use requests and response
- Primarily used for synchronous communication



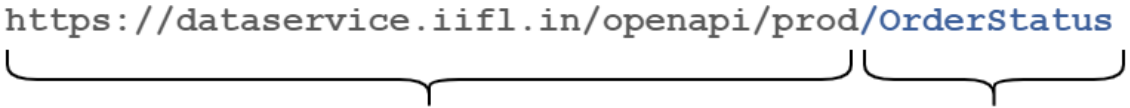
Understanding
URLs

URLs

Uniform Resource Locator

- Base URL
  - Base URL provided by a broker
- Endpoint
  - Usually represents the actions to be performed

Base URL Endpoint



Base URL Endpoint

URLs

Examples

- https://localhost:5000/v1/api/iserver/auth/status
- https://localhost:5000/v1/api/iserver/account/trades
- https://localhost:5000/v1/api/iserver/account/orders
- https://localhost:5000/v1/api/iserver/marketdata/snapshot
- https://localhost:5000/v1/api/portfolio/{accountId}/summary
- https://dataservice.iifl.in/openapi/prod/MarketFeed
- https://dataservice.iifl.in/openapi/prod/BackoffLedger
- https://dataservice.iifl.in/openapi/prod/OrderBookV2
- https://dataservice.iifl.in/openapi/prod/TradeBook
- https://dataservice.iifl.in/openapi/prod/NetPosition

URL Metadata

URL Metadata

HTTP Methods

`https://dataservice.iifl.in/openapi/prod/OrderStatus`

Protocol

URL Metadata

HTTP Methods

`https://dataservice.iifl.in/openapi/prod/OrderStatus`

Protocol

- **`GET`**
  - The **`GET`** method requests a representation of the specified resource. Almost every time, using this method should only retrieve data.

URL Metadata

HTTP Methods

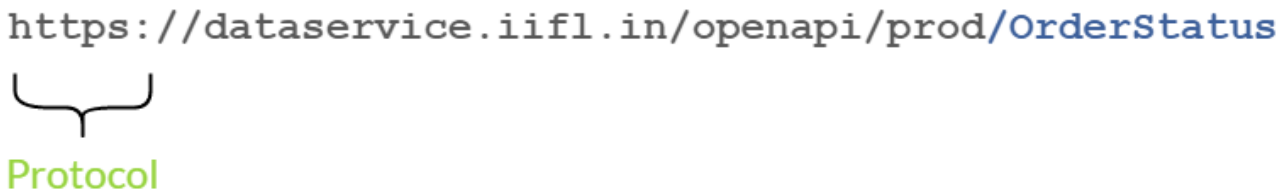
`https://dataservice.iifl.in/openapi/prod/OrderStatus`

Protocol

- **`GET`**
  - The **`GET`** method requests a representation of the specified resource. Almost every time, using this method should only retrieve data.
- **`POST`**
  - The **`POST`** method is used to submit information to the specified resource, often causing a state on the server.

URL Metadata

HTTP Methods



- **GET**
  - The **GET** method requests a representation of the specified resource. Almost every time, using this method should only retrieve data.
- **POST**
  - The **POST** method is used to submit information to the specified resource, often causing a state on the server.
- Other methods
  - **PUT**, **DELETE**, etc.

URL Metadata

Headers

- What it is
  - It lets the client (and server) pass additional information while making requests (or getting responses).
- Why it is required
  - Depending on how a broker implements the REST API, user information such as authentication, type of client or type of information needs to be specified when we requests data from the server. Headers are required to specify such information.
- How to specify headers
  - They are specified using key value pairs separated by colon. In Python, we use a dictionary to specify headers.

URL Metadata

Headers - Example I

```
- headers = {
      'x-auth-token':
          'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfbmFtZSI6Il
          BBUk1BUkpMIiwicm9sZSI6IjYwIiwibmJmIjoxNjI0MDg0MTk3LCJleHAiOj
          E2MjQwODc3OTcsImlhdCI6MTYyNDA4NDE5N30.uYPCG5qJCJXHM68z9ceJGR
          rPzZy41BjcTea7S8ZT008',
      'Ocp-Apim-Subscription-Key': 'fc928d4e9b83935a13a15baa448ff34b',
      'x-clientcode': 'EASYAPP'
  }
```

URL Metadata

Headers - Example II

```
- headers = {
        'Connection': 'keep-alive',
        'Accept': 'application/json, text/plain, */*',
        'X-Device-Type': 'web',
        'User-Agent':
            'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
            like Gecko) Chrome/79.0.3945.88 Safari/537.36',
        'Content-Type': 'application/json;charset=UTF-8'
    }
```

URL Metadata

Body

- What it is
  - A component of a request that contains all the information that clients are required to send to a server.
- Why it is required
  - Some requests requires additional information to be passed to the server to be able to execute the request successfully. For example, order requests, historical data requests, etc.
- How to specify body
  - They are specified using key value pairs separated by colon. In Python, we use a dictionary to specify body. Some some requests, it is *optional*.

URL Metadata

Body - Example I

```
payload = {
        'acctId': 'DU2966568',
        'conid': 56985419,
        'secType': 'STK',
        'orderType': 'MKT',
        'listingExchange': 'NSE',
        'outsideRTH': False,
        'side': 'BUY',
        'ticker': 'COLPAL',
        'tif': 'DAY',
        'quantity': 3,
        'useAdaptive': True
    }
```

URL Metadata

Body - Example II

```
payload = {
        'exchange': 'NSE',
        'order_type': 'LIMIT',
        'instrument_token': 4717,
        'quantity': 1,
        'disclosed_quantity': 0,
        'price': 89.7,
        'order_side': 'BUY',
        'trigger_price': 0,
        'validity': 'DAY',
        'product': 'MIS',
        'user_order_id': 10002
    }
```

Building a Request

Putting it all Together

- Base URL
  - Almost every time it will be static. Provided by a broker.
- Endpoint
  - It will change depending on the request we want to make. Provided by a broker.
- HTTP Method
  - Which one to use depends on the kind of request we make. Specified by a broker for each type of request. Usually, GET or POST is used. A single request cannot have multiple methods.
- Headers
  - Additional information that we need to send to the broker's server. Often remains the same for all requests.
- Body (optional)
  - Used to pass information in form of parameters. Changes depending on the requests we make. What information we need to pass is specified by the broker.

Build 'Request' in Python

Server Responses

What is JSON?

JavaScript Object Notation

- A format to store and transport data between applications.
- It is often used to transfer data from a server to a web page and vice versa.
- Mostly, it is self-describing and easy to understand.
- Format is similar to a key-value format.
- We use **`JSON`** package in Python work with json data.
  - **`json.dumps()`** converts dictionary into json format.
  - **`json.loads()`** converts data in json format into dictionary.

Response Status Codes

What are they?

- Response status codes indicate whether a specific request has been successfully completed.
- Commonly used response codes are:
  - 200 OK: The request has succeeded.
  - 4xx: Client error responses
  - 5xx: Server error responses

Python Demo

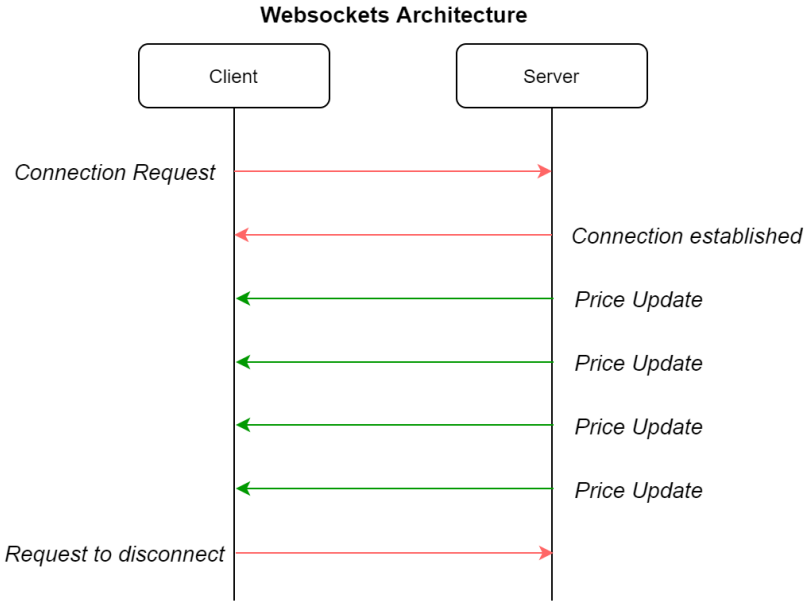
Implementation

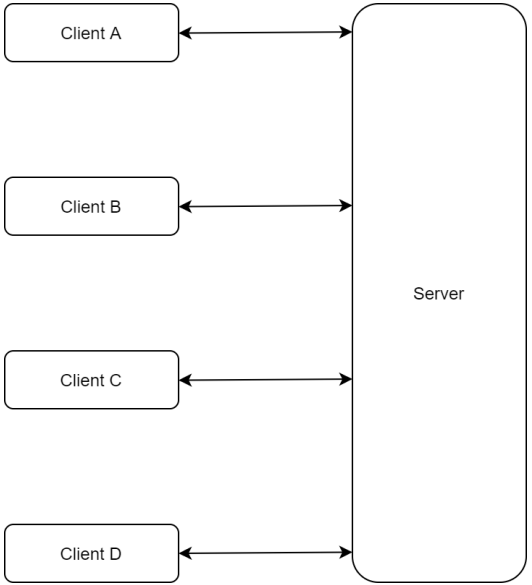
- Authentication
- Account data
- Contract details
- Historical data
- Market snapshot data
- Order placement
- Account positions

Python Demo - Web Sockets

Implementation

- Working with streaming market data





References

To Dive Deeper

- IB Client Portal Web API
  - https://www.interactivebrokers.com/api/doc.html
- IIFL Securities Trading API
  - https://api.iiflsecurities.com/index.html
- REST API Specifications
  - https://en.wikipedia.org/wiki/Representational_state_transfer
- HTTP Methods
  - https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods
- HTTP Status Codes
  - https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

The End!