TBP02: Lecture Summary

Overview:

This document summarises the TBP-02 REST API lecture that gives an understanding of the building blocks of REST API and presents a demonstration of building and deploying trading strategies using it. After this lecture, you should be able to create trading algorithms and deploy them in live markets using REST API.

The lecture covers the following topics:

- Why REST API?
- Limitations of REST API
- What is an API?
- What is REST?
- Fundamental building blocks of REST API
  - URL Components
  - HTTP Methods
  - Headers, Body, Statuses
- Perform various trading operations using the REST API
- References

Why REST API?

REST API is one of the most commonly used interfaces to deploy live trading strategies in the market. It is supported by most of the brokers who provide API, such as Interactive Brokers, Zerodha, IIFL Securities, TD Ameritrade, ICICI Direct, Mastertrust, CoinDCX and many more. This enables a user to deploy strategies on multiple platforms using the same API.

LImitations of REST API

- REST API is not meant for HFT trading.
- It's most suitable for LFT or MFT trading.
- Most brokers would have API limits in place. For example - you can place only 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?

It's an interface that acts as a medium of communication between different software applications. API defines the rules for communication between two systems. To give an analogy, you cannot just go to a printing facility and print a cheque book of a bank for yourself. There's a procedure that you have to follow.

You go to the bank website, fill-up the form, attach your documents and submit. The system then processes in the background, and when your account is ready, account details and the cheque book is handed over to you.

This is precisely how an API works. The bank is the system that we want to communicate with, and we don't have access to the system's internals; we can only talk through the API layer, i.e. the bank website where we fill-up the form and submit the documents. This is called an endpoint.

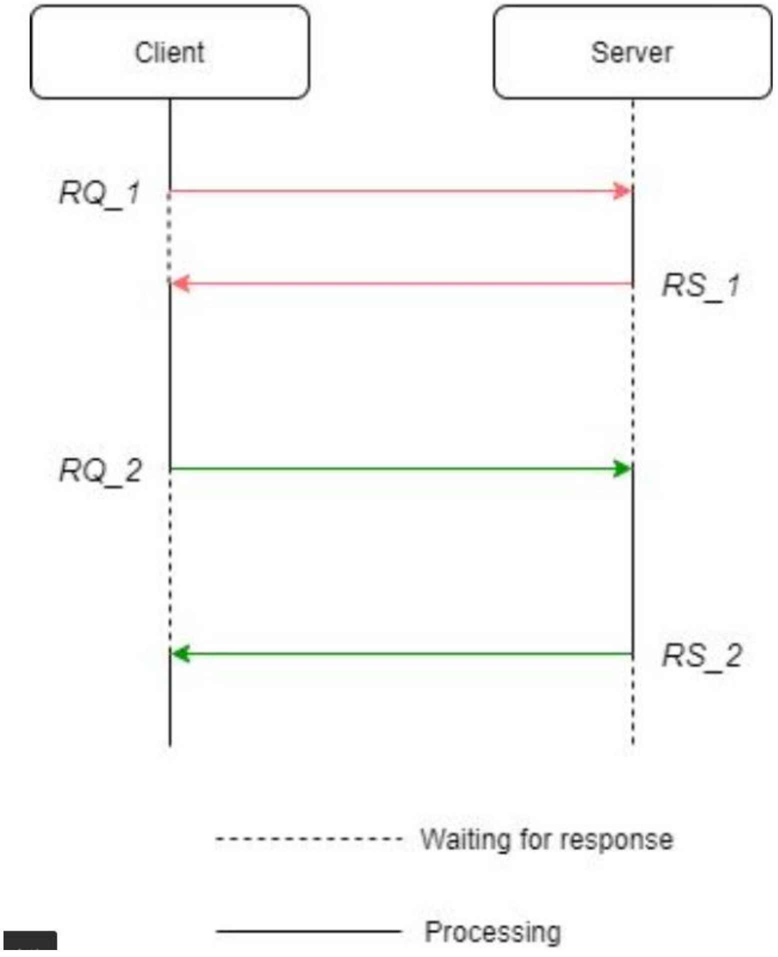
While fetching data, you also need an API key for the system to keep a record of your identity, just as you need your bank account no. for any transaction.

An API call provides the inputs to the system to access the data we need. Each system has a different API, just as each bank's website has a different web address.

What is REST?

It stands for **RE**presentational **S**tate **T**ransfer. This is the philosophy on which API can be built. It's majorly implemented on client-server architecture which proposes the use of requests and responses. REST is primarily used for synchronous communication.

The following image is an example of synchronous communication



<u>Fundamental building blocks of REST API</u>

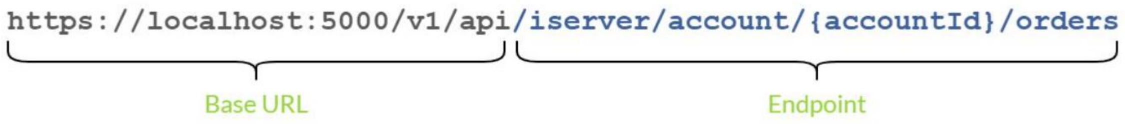
Base URL

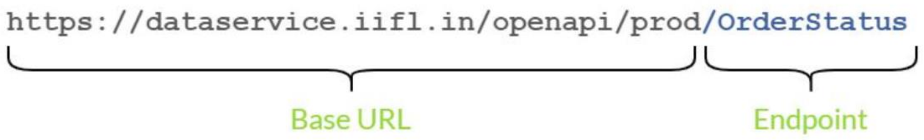
- It remains static almost every time and is provided by a broker.

Endpoint

- It will change depending on the request we want to make and is provided by a broker.

The following is an example of a base URL and the endpoint


https://localhost:5000/v1/api/iserver/account/{accountId}/orders
Base URL
Endpoint



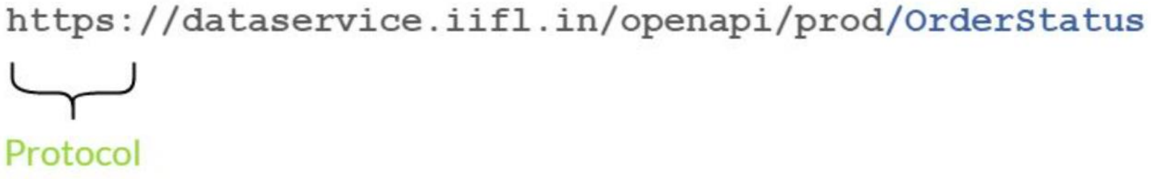
HTTP Method

- Different operations require different HTTP methods.
- Which one to use depends on the kind of request we make.
- It is specified by a broker for each type of request.
- Usually, GET or POST is used.
- A single request cannot have multiple 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.

The following is an example of the HTTPS method



Headers

- They are the additional information that we need to send to the broker's server.
- It often remains the same for all requests.
- User information such as authentication, type of client or type of information needs to be specified when we request data from the server.
- Headers are required to specify such information.

Body (optional)

- It is used to pass information in form of parameters.
- Body changes depending on the requests we make.
- The information that we need to pass is specified by the broker.

- Some requests require additional information to be passed to the server to be able to execute the request successfully.
- For example, order requests, historical data requests, etc.

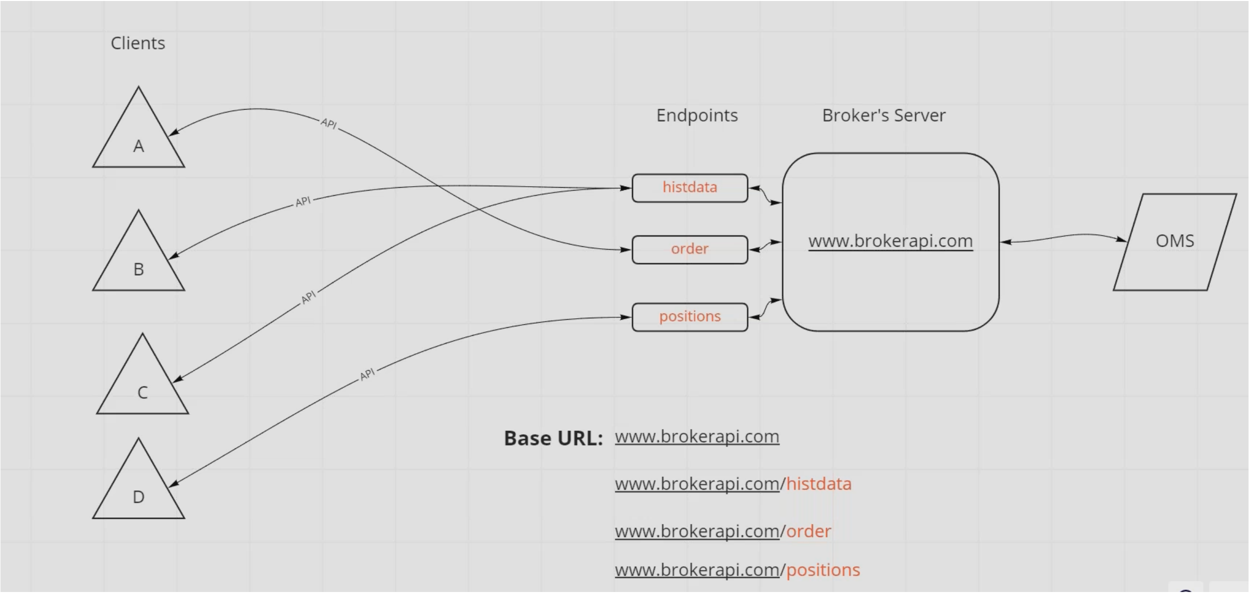
Response status codes

- 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

JSON (Java Script Object Notation)

- It's a format to store and transport data between applications which is often used to transfer data from a server to a web page and vice versa.
- Its format is similar to a key-value format.
- JSON package is used in Python work with json data.
  - json.dumps() converts dictionary into json format.
  - json.loads() converts data in json format into dictionary.

The following snapshot presents an example of interaction between client and server.



<u>Perform various trading operations using the REST API</u>

The following are the steps required to implement a trading strategy using REST API -

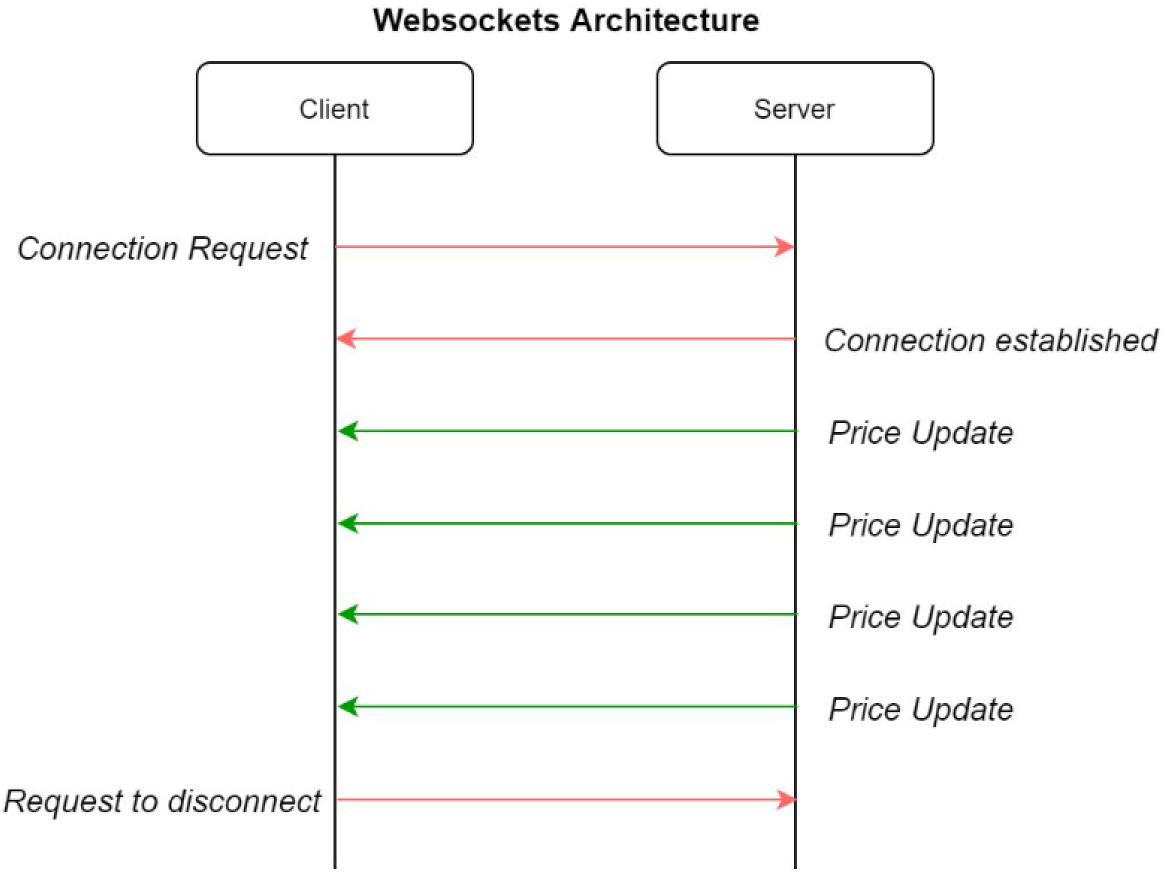
- Load contract details

- Fetch historical data
- Place orders
- Fetch account information
- Work with streaming data

Note

- Synchronous communication is done using REST API whereas WebSocket connection is used for asynchronous communication.
- Almost all the brokers provide live data using WebSockets. The examples for doing this in Python can be found in your LMS portal in the TBP module as discussed in the lecture.

The following image is an example of WebSocket architecture implementation for fetching live data regularly.



References

The following are the resources you can refer to in case you want detailed knowledge and want to use REST API for a particular broker -

- IB Client Portal Web API
- IIFL Securities Trading API

- REST API Specifications
- HTTP Methods
- HTTP Status Codes

User flow from kite.trade/docs/connect/v3/user

Copy the URL, enter the login id and password after running the bash file

Authentication status

interactivebrokers.github.io/

Show the process through screenshots

Goto the directory
Run the command
Copy paste the url
Login details
Run the tickle file
Reload functions from the spyder notebook.
Show the results
Save the response in json file
Print the parsed code

Datastructure to convert Java/.net/c++ to Python.

JSON is a formatting structure allows to store data in a key value pair.

Response codes:
200 - everything correct
401 - error in endpoint

Websocket architecture - Non synchronous
REST API - synchronous