SYSTEM BASED TRADE EXECUTION

ALGORITHMIC TRADING API

Understanding Zerodha Kite API

Contents [hide]

Advice for Beginners

If you do not have programming background, or you are new to REST APIs and streaming data, please consider yourself as a beginner and adhere to this advice- Learn with Patience. That's the three word advice we will like to repeat again- Learn with Patience.

These tutorials are written from scratch, specially keeping the beginners in mind. However, when you are reading for the first time, you may feel bombarded with many new concepts. If you do not understand something, do not worry at all. You can skip and revisit the particular concept/code again. If you make a commitment to spend daily or weekly time for learning algorithmic trading, you will soon achieve it!

If you are new to python, you can take the beginner-to-intermdiate level short course "Python for Algo Trading" created by Mr. Saurabh Lohiya. It is accessible in the python section of AlgoPro forum.

[author][author_info]Author Bio: Gunjan Narulkar has more than 10 years in problems solving with data using technology and mathematics. He is interested in exploring the applications of mathematics and intelligent agents in the field of trading and finance in general.[/author_info][/author]

Introduction to Algo Trading via API

Trading, as we know it, involves multiple steps and requires us as humans to make decisions at each of the steps individually, using different facets of knowledge that we possess by virtue of our education, training, experience and above all, "intelligence". You can refer What is Algo Trading to establish differences in various jargons used to describe the activities related to tradir
world. An Automated Trading platform may have below capabilities at the least:

● Algorithm should be able to collect historical and real time feeds of market data
● Using this data, it should automatically create trading signals
● It should also consider all the complexities and execute the trades effectively without human intervention

Let's try understanding the pointers in detail. Selecting Broker for Algo Trading provides the basic information about providers of data and trading platforms in Indian market. One of the most prominent providers among them is Zerodha, which provides wide range of services at competitive pricing. There are many ways in which they allow their users access to historic and real time data along with trade execution capabilities. But from an algorithmic trading perspective, the preferred way is using Kite Connect (KC) interface. It is an Application Programming Interface (API) provided by Zerodha to us using which, we can even build full-fledged trading platforms with capabilities from capturing the data to executing and monitoring trades – all in real time! However, before we get all excited and start building our own Algorithmic Trading Platform (ATP), let's understand what an API means and how it functions, from a conceptual level.

What is an API?

An application programming interface (API) is a protocol intended to be used as an interface by software components to communicate with each other. It's basically a set of programming instructions and standards for accessing a Web-based software application. A software company (like Amazon, Google etc) releases its API to the public so that other software developers can design products that are powered by its service.
Using an API is like driving a car. Each API comes with a "user manual" and once you understand it, you can use it without worrying about how engine works internally – just by understanding the various buttons, or "endpoints" as is the case for any API – a user or program can get things done. This type of integration is called seamless, since the user never notices when software functions are handed from one application to another.

REST API and Websocket Connections

A REST API is also called a Web Service or simply, a service, and is typically hosted by a Web Server. A web server is a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web. Simply put, if you run a program on web server, it can be accessed over internet and loosely speaking, that program becomes the API which is designed to serve a set of specific tasks. Now this API is called by the programs written by users to complete a set of tasks in a specific order automatically. The particular "address" which is called by the user programs to provide the exact instructions, about specifics of task that needs to be executed by the API, are called "endpoints" and API itself, is a part of a larger Software Development Interface (SDK) which constitutes complete functionality offered by the provider. Typically, each API constitutes multiple endpoints, each serving a specific purpose and when connected to, responds to the requester using a data format named JSON – a lightweight and well accepted format for data transfer. An example can be – to extract historical data for a set of securities, the KC API has a specific endpoint, which accepts specific inputs and provides predefined output as per input provided. So far, we are talking about use cases which are request driven – a request is made And responded to by API and connection is severed – but many times real time or "streaming" data access is required. It is here that Web Socket technology comes in. It allows for active connections between requester (the user program) and provider (the API) which both parties can use to send data in real time. Obviously, data security and authorizations are valid concerns but the technology is pretty matured to address all these areas.

So by now, you should be familiar with the various jargons around API world and generally, would have gotten a handle on how this world works. Let's now understand how to work with an API.

A side note on how to create an API?
API can be written in multiple programming languages. Python is one of the more popular language's and Flask is one of the many servers which can be used to create APIs. Refer this link for a basic demonstration on how to create an API using both Python and Flask. But since the aim of these tutorials is largely to understand how to consume an API, let's focus on that for now.

Creating App with Zerodha Kite API

Since we are focusing on Python and Kite API, let's see how using Python, we can connect to Kite API. The first step is to create an app after login to Kite dashboard. This will give us API Key and Secret, which we can use in python code for algorithmic trading. Next, we will see some basic python libraries required for writing a siimple code.

You can name the app to anything fancy you want, provide your zerodha client id and add description. The only field which requires explantion here is the Redirect URL and Postback URL.

Redirect URL in Zerodha Kite API App

If you are building a webhosted software for use by several people, you should provide your webserver app url. However, for most of us attempting algorithmic trading at a individuala level, the localhost address will be sufficient.

All computers use a specially reserved IP address 127.0.0.1 called the localhost. This IP address does not let them communicate with other devices like a real IP address does.

Your computer might have something like 103.240.79.175 assigned to it as private IP address, which can be used to communicate with a router and other networked devices. However, the localhost address 127.0.0.1 attached to it to mean "this computer," or the computer you are currently using. This is unlike a regular IP address that is used to transfer files to and from other networked devices.

For buildin my first app I have used http://127.0.0.1:19291/kite-redirect. Here 127.0.0.1 means the localhost, because I will be using the same computer for algorithmic trading using python via the zerodha kite api. The port address (1929) is just to ensure that only my computer's specific python code will connect to it.

Python Libraries for Zerodha Kite API

Other than the common python libraries, you will need to pip install the kiteconnect and json libraries.

You may face OS specific dependency issues which can be resolved by following clues in this link.

That's all! Now you should be able to use multiple endpoints related to placing order and getting data etc. We'll focus on each of these aspects in details in separate tutorials in time to come.

Telegram Channel for Algo Developers

You can join telegram channel of algoji to get updates: https://t.me/algoji

13 Comments

Yash Joshi Reply

Want to be added in kite api group

Vinay Reply

I want to build an app in z connect for handling multiple demate accoumts

Nagesh Kuravadi Reply

Please add me to the WhatsApp group

Souman Jyoti Reply

The content a good for begginer

Abhay Reply

Please add me to the whatsapp grp

pranit bagadia Reply

pls add me to whatsapp group

Vignesh Reply

Pls add me to the group

Phillis Reply