

The screenshot shows a Jupyter Notebook interface with a dark theme. The top menu bar includes File, Edit, View, Insert, Runtime, Tools, and Help. Below the menu is a toolbar with search, command, code, text, and run all buttons. A sidebar on the left contains icons for file operations like new, open, save, and refresh. The main area displays a code cell with Python code for downloading AAPL stock data from Yahoo Finance. The output shows a warning message and the resulting DataFrame. The DataFrame has columns: Date, Price, Close, High, Low, Open, and Volume. It lists data from January 3, 2011, to December 30, 2021. The final output summary indicates 2768 rows by 5 columns.

```
[2]  ✓ Os
import pandas as pd
import yfinance as yf

df = yf.download("AAPL",
                 start="2011-01-01",
                 end="2021-12-31",
                 progress=False)
print(f"Downloaded {len(df)} rows of data.")
df
```

```
... /tmp/ipython-input-2149336923.py:4: FutureWarning: YF.download() has changed argument auto_adjust default to True
... df = yf.download("AAPL",
Downloaded 2768 rows of data.
```

Date	Price	Close	High	Low	Open	Volume
Ticker	AAPL	AAPL	AAPL	AAPL	AAPL	
2011-01-03	9.884150	9.904843	9.742292	9.766285	445138400	
2011-01-04	9.935735	9.972024	9.841562	9.970224	309080800	
2011-01-05	10.017009	10.027206	9.882049	9.883549	255519600	
2011-01-06	10.008911	10.054497	9.984018	10.038602	300428800	
2011-01-07	10.080589	10.087487	9.954027	10.016708	311931200	
...
2021-12-23	172.658051	173.216346	171.668808	172.236893	68356600	
2021-12-27	176.624878	176.713025	173.431864	173.451443	74919600	
2021-12-28	175.606216	177.604310	174.861837	176.458351	79144300	
2021-12-29	175.694366	176.918683	174.479839	175.645391	62348900	
2021-12-30	174.538620	176.859935	174.430880	175.782530	59773000	

2768 rows × 5 columns

Getting AAPL data from yahoo finance.

```
import pandas as pd
import nasdaqdatalink

nasdaqdatalink.ApiConfig.api_key = "hNuK1_zyP-wGBD91wbey"

df = nasdaqdatalink.get(dataset="EOD/AAPL",
                        start_date="2011-01-01",
                        end_date="2021-12-31")

print(f"Downloaded {len(df)} rows of data.")
df.head()

...
-----
JSONDecodeError                                 Traceback (most recent call last)
/usr/local/lib/python3.12/dist-packages/requests/models.py in json(self, **kwargs)
    975         try:
--> 976             return complexjson.loads(self.text, **kwargs)
    977         except JSONDecodeError as e:

          ^ 12 frames
JSONDecodeError: Expecting value: line 1 column 1 (char 0)

During handling of the above exception, another exception occurred:

JSONDecodeError                                 Traceback (most recent call last)
JSONDecodeError: Expecting value: line 1 column 1 (char 0)

During handling of the above exception, another exception occurred:

DataLinkError                                    Traceback (most recent call last)
/usr/local/lib/python3.12/dist-packages/nasdaqdatalink/connection.py
    90         return response.json()
    91     except ValueError:
--> 92         raise DataLinkError(http_status=response.status_
    93
    94     @classmethod

DataLinkError: (Status 403) Something went wrong. Please try again. [connect@data.nasdaq.com.
```

J Try downloading data using the 'EOD/AAPL' dataset

♦ I will modify the code to download data using the 'EOD/AAPL' dataset.

It seems that changing the dataset to `EOD/AAPL` did not resolve the issue, and you are still encountering a `DataLinkError: (Status 403)`. This error typically indicates that your API key is either invalid or lacks the necessary

[Verify the nasdaqdatalink API key](#) [Explain how to get a valid API key for EOD/AAPL](#) [Find](#)

What can I help you build?

+ Gemini 2.5 Flash ▶

Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)

Attempting to download AAPL data from Nasdaq. WIKI/APPL from textbook is defunct and throws an error. API code given from Nasdaq does not have adequate permissions and throws an error preventing me from obtaining the necessary data for the chart. The code from the textbook does not work.

The screenshot shows a dark-themed IDE window titled "Abel-Lab1". The menu bar includes File, Edit, View, Insert, Runtime, Tools, Help, Commands, + Code, + Text, and Run all. A toolbar at the top has icons for up, down, edit, delete, and more. The code editor contains Python code for interacting with the Intrinio API to get stock prices for AAPL from 2011-01-01 to 2021-12-31. The code uses the intrinio_sdk library and pandas. An error occurs at line 7, resulting in an ApiException (401). The stack trace points to the rest.py file in the intrinio_sdk package.

```
[25] ⓘ Os
import intrinio_sdk as intrinio
import pandas as pd

intrinio.ApiClient().set_api_key("OjUwMTM2N2VlZDU4NDEyNGZlNWZmYjU0ZjNmODVkJGU2")
security_api = intrinio.SecurityApi()

r = security_api.get_security_stock_prices(
    identifier="AAPL",
    start_date="2011-01-01",
    end_date="2021-12-31",
    frequency="daily",
    page_size=10000
)
df = (
    pd.DataFrame(r.stock_prices_dict)
    .sort_values("date")
    .set_index("date")
)
print(f"Downloaded {df.shape[0]} rows of data.")
df.head()

...
-----
ApiException                                         Traceback (most recent call last)
/tmp/ipython-input-649438207.py in <cell line: 0>()
      5     security_api = intrinio.SecurityApi()
      6
----> 7     r = security_api.get_security_stock_prices(
      8         identifier="AAPL",
      9         start_date="2011-01-01",

      ▾ 7 frames
/usr/local/lib/python3.12/dist-packages/intrinio_sdk/rest.py in request(self, method, url, query_params, headers, body, post_params, _preload_content, _request_timeout)
    232
    233         if not 200 <= r.status <= 299:
--> 234             raise ApiException(http_resp=r)
    235
    236     return r

ApiException: (401)
```

Obtaining AAPL data from Intrinio. Once again, the API key does not have the necessary permissions to gather the data. The code in the textbook does not work and throws errors.

```
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
[34] ① 0s ALPHA_VANTAGE_API_KEY = "QMWCE2725VC80LEH"
crypto_api = CryptoCurrencies(key=ALPHA_VANTAGE_API_KEY,
                               output_format= "pandas")
data, meta_data = crypto_api.get_digital_currency_daily(
    symbol="BTC",
    market="EUR"
)
{'1. Information': 'Daily Prices and Volumes for Digital Currency',
 '2. Digital Currency Code': 'BTC',
 '3. Digital Currency Name': 'Bitcoin',
 '4. Market Code': 'EUR',
 '5. Market Name': 'Euro',
 '6. Last Refreshed': '2022-08-25 00:00:00',
 '7. Time Zone': 'UTC'}

crypto_api.get_digital_currency_exchange_rate(
    from_currency="BTC",
    to_currency="USD"
)[0].transpose()

...
ValueError: Traceback (most recent call last)
/tmp/ipython-input-842629326.py in <cell line: 0>()
    16     '7. Time Zone': 'UTC'}
    17
--> 18 crypto_api.get_digital_currency_exchange_rate(
    19     from_currency="BTC",
    20     to_currency="USD"

      2 frames
/usr/local/lib/python3.12/dist-packages/alpha_vantage/alphavantage.py in _handle_api_call(self, url)
    359         raise ValueError(json_response["Error Message"])
    360     elif "Information" in json_response and self.treat_info_as_error:
--> 361         raise ValueError(json_response["Information"])
    362     elif "Note" in json_response and self.treat_info_as_error:
    363         raise ValueError(json_response["Note"])

ValueError: Thank you for using Alpha Vantage! Please consider spreading out your free API requests more sparingly (1 request per second). You may subscribe to any of the premium plans at https://www.alphavantage.co/premium/ to lift the free key rate limit (25 requests per day), raise the per-second burst limit, and instantly unlock all premium endpoints
```

Attempting to get BitCoin data using Alpha Vantage. The API key from Alpha Vantage throws an error claiming the free code should be used sparingly. This was the first time running the code and it already threw that error. It made the same error on subsequent run attempts.

The screenshot shows the Abel-Lab1 IDE interface. The code editor contains Python code for interacting with the CoinGecko API to get historical price data for Bitcoin and the top 7 trending coins. The resulting DataFrame is displayed below the code.

```
[37] ✓ 0s
from pycoingecko import CoinGeckoAPI
from datetime import datetime
import pandas as pd

cg = CoinGeckoAPI()

ohlc = cg.get_coin_ohlcs_by_id(
    id="bitcoin", vs_currency="usd", days="14"
)
ohlc_df = pd.DataFrame(ohlc)
ohlc_df.columns = ["date", "open", "high", "low", "close"]
ohlc_df["date"] = pd.to_datetime(ohlc_df["date"], unit="ms")
ohlc_df

trending_coins = cg.get_search_trending()
(
    pd.DataFrame([coin["item"] for coin in trending_coins["coins"]])
    .drop(columns=["thumb", "small", "large"])
)

...
```

	id	coin_id	name	symbol	market_cap_rank	slug	price_btc	score	data
0	nietzschean-penguin	71738	Nietzschean Penguin	PENGUIN	474	nietzschean-penguin	8.995377e-07	0	{'price': 0.08012246803547766, 'price_btc': '0.000...'}
1	linea	68507	Linea	LINEA	399	linea	7.237381e-08	1	{'price': 0.006446387162417462, 'price_btc': '0.000...'}
2	seeker	70974	Seeker	SKR	291	seeker	3.619223e-07	2	{'price': 0.03401809615066124, 'price_btc': '0.000...'}
3	rain	69134	Rain	RAIN	45	rain	1.116160e-07	3	{'price': 0.009941721761778148, 'price_btc': '0.000...'}
4	river	69318	River	RIVER	93	river	6.287465e-04	4	{'price': 56.00290498386819, 'price_btc': '0.000...'}
5	pudgy-penguins	52622	Pudgy Penguins	PENGU	137	pudgy-penguins	1.103307e-07	5	{'price': 0.00982723237297798, 'price_btc': '0.000...'}
6	pax-gold	9519	PAX Gold	PAXG	57	pax-gold	5.677186e-02	6	{'price': 5056.710139724864, 'price_btc': '0.000...'}
7	bittensor	28452	Bittensor	TAO	53	bittensor	2.651243e-03	7	{'price': 236.14809411692012, 'price_btc': '0.000...'}
8	tether-gold	10481	Tether Gold	XAUT	50	tether-gold	5.641937e-02	8	{'price': 5025.313594463914, 'price_btc': '0.000...'}
9	monad	38927	Monad	MON	283	monad	2.095808e-07	9	{'price': 0.018667508145397135, 'price_btc': '0.000...'}
10	ondo-finance	26580	Ondo	ONDO	64	ondo	3.862958e-06	10	{'price': 0.3440764637935509, 'price_btc': '0.000...'}
11	axie-infinity	13029	Axie Infinity	AXS	161	axie-infinity	2.962900e-05	11	{'price': 2.639076669538257, 'price_btc': '0.000...'}
12	solana	4128	Solana	SOL	7	solana	1.421461e-03	12	{'price': 126.61050813288374, 'price_btc': '0.000...'}
13	zcash	486	Zcash	ZEC	28	zcash	4.081096e-03	13	{'price': 363.50613821423514, 'price_btc': '0.000...'}
14	hyperliquid	50882	Hyperliquid	HYPE	31	hyperliquid	2.579826e-04	14	{'price': 22.978691438934, 'price_btc': '0.000...'}

Obtaining data from CoinGecko. This chart shows the top 7 trending coins. This code and the yahoo finance codes are the only two codes from the textbook that worked. The similarity is that neither of them required me to make an account on a website to get an API key. The textbook was published in 2022. The code and website permissions may have changed in four years so they throw errors now.