CODE:

import requests

from bs4 import BeautifulSoup

import gspread

from oauth2client.service\_account import ServiceAccountCredentials

import time

# Define the URLs of the exchanges

exchange\_urls = {

"Wazirx": "https://wazirx.com/",

"Coinswitch": "https://coinswitch.co/",

"Mudrex": "https://mudrex.com/",

"CoinDCX": "https://coindcx.com/"

}

# Function to scrape cryptocurrency prices from each exchange

def scrape\_prices():

prices = {}

for exchange, url in exchange\_urls.items():

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

prices[exchange] = {}

# Example of scraping BTC price

btc\_price\_element = soup.find('div', class\_='btc-price')

btc\_price = btc\_price\_element.text.strip() if btc\_price\_element else 'N/A'

prices[exchange]["BTC"] = btc\_price

# Example of scraping ETH price

eth\_price\_element = soup.find('div', class\_='eth-price')

eth\_price = eth\_price\_element.text.strip() if eth\_price\_element else 'N/A'

prices[exchange]["ETH"] = eth\_price

if len(prices) == 20: # for top 20 prices

break

# Add more cryptocurrencies as needed

return prices

# Function to update Google Sheet with scraped prices

def update\_google\_sheet(prices):

scope = ['https://spreadsheets.google.com/feeds', 'https://www.googleapis.com/auth/drive']

credentials = ServiceAccountCredentials.from\_json\_keyfile\_name('credentials.json', scope)

client = gspread.authorize(credentials)

sheet = client.open("Cryptocurrency Prices").sheet1 # Replace with your Google Sheet name

for i, (exchange, price\_data) in enumerate(prices.items(), start=2):

sheet.update\_cell(i, 1, exchange)

for j, (crypto, price) in enumerate(price\_data.items(), start=2):

sheet.update\_cell(i, j, price)

# Main function to continuously scrape and update prices

def main():

while True:

prices = scrape\_prices()

update\_google\_sheet(prices)

print("Prices updated successfully at", time.strftime("%H:%M:%S"))

time.sleep(60) # Update prices every minute

if \_\_name\_\_ == "\_\_main\_\_":

main()

Alteration:

1. Cryptocurrency Prices is nothing but your Google sheet name, make sure you name it correctly.
2. Example for prices

prices[exchange] = {

"BTC": 10000.00,

"ETH": 500.00,

# Add more cryptocurrencies as needed

}

1. For the Google Sheets API authentication, you'll need to create a service account and obtain the credentials JSON file, and ensure that it's named **credentials.json** and placed in the same directory as the script.
2. You need to make the changes in scrapprices() function based on website input (check the html codes in console).

#Alternate code for top 20 individual website

# Function to scrape the top 20 cryptocurrencies from CoinMarketCap

def scrape\_top\_currencies():

url = "https://coinmarketcap.com/"

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

currencies = []

for row in soup.find\_all('tr', class\_='cmc-table-row'):

currency = row.find('p', class\_='cmc-table\_\_cell--sort-by\_\_name').text.strip()

currencies.append(currency)

if len(currencies) == 20:

break

return currencies

Steps for authentication for the Google Sheets API:

1. **Create a Google Cloud Platform (GCP) Project:**
   * Go to the [Google Cloud Console](https://console.cloud.google.com/).
   * Create a new project or select an existing one where you want to enable the Google Sheets API.
2. **Enable the Google Sheets API:**
   * In the Cloud Console, navigate to the "APIs & Services" > "Library" page.
   * Search for "Google Sheets API" and click on it.
   * Click the "Enable" button to enable the API for your project.
3. **Create a Service Account:**
   * Go to the "APIs & Services" > "Credentials" page.
   * Click on the "Create credentials" dropdown and select "Service account".
   * Enter a name for your service account, choose a role (e.g., Project > Editor), and click on "Continue".
   * Skip the optional step to grant users access to this service account and click on "Done".
   * After the service account is created, click on it from the list of service accounts.
   * Click on the "Add Key" dropdown and select "Create new key".
   * Choose the JSON key type and click on "Create". This will download a JSON key file to your computer.
4. **Grant Access to Google Sheets:**
   * Open your Google Sheet where you want to update the data.
   * Share the Google Sheet with the email address associated with your service account (found in the JSON key file).
   * Give the service account the necessary permissions (edit or view) based on your requirements.
5. **Use the JSON Key File in Your Python Script:**
   * Place the downloaded JSON key file in the same directory as your Python script.
   * Use the JSON key file in your Python script to authenticate with the Google Sheets API.