Task: You’ve been asked to assist with some development to add a chart to a trader’s dashboard allowing them to better identify under/over-valued stocks.

Provided Solution:

```
import json
import random
import urllib.request
N = 5  # Number of iterations for querying the price
QUERY = "https://api.iextrading.com/1.0/tops/last?symbols={}"
def getDataPoint(quote):
    stock = quote['symbol']
    bid_price = float(quote['bidPrice'])
    ask_price = float(quote['askPrice'])
    price = (bid_price + ask_price) / 2
    return stock, bid_price, ask_price, price
def getRatio(price_a, price_b):
    """ Get ratio of price_a and price_b """
    """ Also create some unit tests for this function in client_test.py """
    if price_b == 0:
        return None
    return price_a / price_b
# Main
if __name__ == "__main__":
    # Query the price once every N seconds.
    for _ in range(N):
        symbols = ['ABC', 'DEF']  # Replace with the desired stock symbols
        query = QUERY.format(','.join(symbols))
        quotes = json.loads(urllib.request.urlopen(query).read())
  """ Update to get the ratio """
     prices = {}
        for quote in quotes:
```

```
        stock, bid_price, ask_price, price = getDataPoint(quote)

        prices[stock] = price

        print("Quoted {} at (bid:{}, ask:{}, price:{})".format(stock, bid_price, ask_price, price))


    ratio = getRatio(prices.get('ABC'), prices.get('DEF'))

    if ratio is not None:

        print("Ratio:", ratio)

    else:

        print("Unable to calculate ratio due to division by zero.")
```