

#GitHub 주소 : <https://github.com/00GYAI/GYDHAL.git>

#그룹 인원: 김가영, 임동혁

```
import time
```

```
import requests
```

```
import pandas as pd
```

```
import datetime
```

```
def cal_mid_price(gr_bid_level, gr_ask_level):
```

```
    if len(gr_bid_level) > 0 and len(gr_ask_level) > 0:
```

```
        bid_top_price = gr_bid_level.iloc[0].price
```

```
        ask_top_price = gr_ask_level.iloc[0].price
```

```
        mid_price = (bid_top_price + ask_top_price) * 0.5
```

```
        return mid_price
```

```
    else:
```

```
        print('Error: serious cal_mid_price')
```

```
        return -1
```

```
def cal_book_imbalance(gr_bid_level, gr_ask_level):
```

```
    total_bid_quantity = gr_bid_level['quantity'].sum()
```

```
    total_ask_quantity = gr_ask_level['quantity'].sum()
```

```
    return total_bid_quantity - total_ask_quantity
```

```
def cal_book_delta(gr_bid_level, gr_ask_level):
```

```
    return len(gr_bid_level) - len(gr_ask_level)
```

```
#Title printing outside the loop
```

```
print("book-delta | book-imbalance | Mid-Price | Timestamp")
```

```
start_time = time.time()
```

```
end_time = start_time + (2 * 60 * 60) # 2 hours
```

```
while True:
```

```
    book = {}
```

```
    response = requests.get('https://api.bithumb.com/public/orderbook/BTC_KRW/?
```

```
count=5')
```

```
    book = response.json()
```

```
    data = book['data']
```

```
    bids = pd.DataFrame(data['bids']).apply(pd.to_numeric, errors='coerce')
```

```
    bids.sort_values('price', ascending=False, inplace=True)
```

```
    bids = bids.reset_index(drop=True)
```

```
    bids['type'] = 0
```

```
    asks = pd.DataFrame(data['asks']).apply(pd.to_numeric, errors='coerce')
```

```
    asks.sort_values('price', ascending=True, inplace=True)
```

```
    asks = asks.reset_index(drop=True)
```

```
    asks['type'] = 1
```

```

df = pd.concat([bids, asks])

timestamp = datetime.datetime.now()
req_timestamp = timestamp.strftime('%Y-%m-%d %H:%M:%S')

df['timestamp'] = req_timestamp

mid_price = cal_mid_price(bids, asks)

# Calculate mid-price
mid_price = cal_mid_price(bids, asks)
df['mid_price'] = [mid_price] * len(df) # Ensure the mid_price is repeated for each
row

# Calculate and include the book imbalance
book_imbalance = cal_book_imbalance(bids, asks)
df['book_imbalance'] = [book_imbalance] * len(df) # Ensure the book_imbalance
is repeated for each row

# Calculate and include the book delta
book_delta = cal_book_delta(bids, asks)
df['book_delta'] = [book_delta] * len(df) # Ensure the book_delta is repeated for
each row

df['timestamp'] = req_timestamp

print(df[['book_delta', 'book_imbalance', 'mid_price',
'timestamp']].iloc[0].astype(str).str.cat(sep=' | '))

df.to_csv("./2023-12-07-bithumb-BTC-feature.csv", index=False, header=False,
mode='a')

time.sleep(1)

```