def **fetch\_lme\_data**():

    headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"}

    response = requests.get(lme\_url, *headers*=headers)

    response.raise\_for\_status()

    tables = pd.read\_html(io.StringIO(response.text))

    df = tables[0]

    df['抓取時間'] = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

    df['資料來源'] = 'LME'

*return* df

def **fetch\_westmetall\_lme\_data**():

    headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"}

    response = requests.get(westmetall\_url, *headers*=headers)

    response.raise\_for\_status()

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

    table = soup.find("table")

    rows = table.find\_all("tr")

    data = []

*for* row *in* rows[1:]:

        cols = row.find\_all("td")

*if* len(cols) >= 3:

            metal = cols[0].get\_text(*strip*=True)

            settlement\_kasse = cols[1].get\_text(*strip*=True)

            three\_months = cols[2].get\_text(*strip*=True)

            data.append({

                "金屬": metal,

                "Settlement Kasse": settlement\_kasse,

                "3 months": three\_months,

                "抓取時間": datetime.now().strftime('%Y-%m-%d %H:%M:%S'),

                "資料來源": "Westmetall"

            })

    df = pd.DataFrame(data)

*return* df

def **fetch\_bot\_fx\_data**():

    headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"}

    response = requests.get(bot\_url, *headers*=headers)

    response.raise\_for\_status()

    tables = pd.read\_html(io.StringIO(response.text), *header*=[0,1])

    df = tables[0]

    currency\_col = [col *for* col *in* df.columns *if* '幣別' in col[0]][0]

    buy\_cols = [col *for* col *in* df.columns *if* col[1] == '本行買入']

    sell\_cols = [col *for* col *in* df.columns *if* col[1] == '本行賣出']

    def **pick\_spot\_col**(*cols\_to\_check*, *df\_to\_check*):

*for* col *in* *cols\_to\_check*:

            vals = pd.to\_numeric(*df\_to\_check*[col], *errors*='coerce')

*if* vals.notna().sum() > 0 and vals.max() < 100 and vals.min() > 0.1:

*return* col

*return* None

    spot\_buy\_col = pick\_spot\_col(buy\_cols, df)

    spot\_sell\_col = pick\_spot\_col(sell\_cols, df)

*if* currency\_col and spot\_buy\_col and spot\_sell\_col:

        df\_fx = df[[currency\_col, spot\_sell\_col, spot\_buy\_col]].copy()

        df\_fx.columns = ['幣別', '即期買入', '即期賣出']

        df\_fx['即期中間價'] = (

            pd.to\_numeric(df\_fx['即期買入'], *errors*='coerce') +

            pd.to\_numeric(df\_fx['即期賣出'], *errors*='coerce')

        ) / 2

        df\_fx['抓取時間'] = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

        df\_fx['資料來源'] = 'BOT'

*# 保證欄位順序*

        df\_fx = df\_fx[['幣別', '即期買入', '即期賣出', '即期中間價', '抓取時間', '資料來源']]

*return* df\_fx

*else*:

        log("找不到正確的即期買入/賣出欄位", 'warning')

*return* pd.DataFrame()

def **fetch\_all\_bot\_rates**():

    url = "https://rate.bot.com.tw/xrt/all/day"

    log(f"抓取網址: {url}")

*try*:

        response = requests.get(url)

        response.encoding = 'utf-8'

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

        time\_element = soup.find(*text*=lambda *text*: *text* *and* "掛牌時間" *in* *text*)

*if* time\_element:

            time\_str = time\_element.split('：')[-1].strip()

            listing\_time = datetime.strptime(time\_str, '%Y/%m/%d %H:%M')

            date\_str = listing\_time.strftime('%Y-%m-%d')

*else*:

            date\_str = datetime.now().strftime('%Y-%m-%d')

            log("警告: 無法從網頁取得掛牌時間，使用系統當前日期", 'warning')

        current\_time = datetime.now().strftime('%Y-%m-%d %H:%M:00')

        tables = pd.read\_html(response.text, *header*=1)

        df = tables[0]

*if* df.iloc[0, 0] == "幣別":

            df = df.iloc[1:].reset\_index(*drop*=True)

        df.columns = ['幣別', '現金買入', '現金賣出', '即期買入', '即期賣出'] + list(df.columns[5:])

        df = df[['幣別', '現金買入', '現金賣出', '即期買入', '即期賣出']]

        df['日期'] = date\_str

        df['抓取時間'] = current\_time

        df['資料來源'] = 'BOT\_Web'

        df = df[['日期', '幣別', '現金買入', '現金賣出', '即期買入', '即期賣出', '抓取時間', '資料來源']]

        log(df.to\_string(*index*=False))

*return* df

*except* Exception *as* e:

        log(f"取得匯率資料時發生錯誤: {e}", 'error')

*import* traceback

        traceback.print\_exc()

*return* None