

Yahoo Finance API for Java

段維瀚老師 2023-12

pom.xml

```
<dependencies>
  <!-- Apache Commons CSV -->
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-csv</artifactId>
    <version>1.8</version>
  </dependency>

  <!-- Apache HttpClient for making HTTP requests -->
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.13</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
  <dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.8</version> <!-- Use the latest version available -->
  </dependency>
</dependencies>
```

單筆(最近紀錄)

```
package test;
import org.apache.commons.csv.CSVFormat;
import org.apache.commons.csv.CSVParser;
import org.apache.commons.csv.CSVRecord;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;
import com.google.gson.Gson;
import java.io.IOException;
import java.time.LocalDate;
import java.time.ZoneId;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
public class StockDataFetcher {

    public static void main(String[] args) throws IOException {
        String stockNo = "0050.TW";

        // 使用當前日期作為結束日期
        LocalDate endDate = LocalDate.now();
        // 設置開始日期為前一天
        LocalDate startDate = endDate.minusDays(1);
        // 將開始和結束日期轉換為 Unix 時間戳(秒為單位)
        long period1 = startDate.atStartOfDay(ZoneId.of("UTC")).toEpochSecond();
        long period2 = endDate.atStartOfDay(ZoneId.of("UTC")).toEpochSecond();

        String url = "https://query1.finance.yahoo.com/v7/finance/download/"
            + stockNo
```

Yahoo Finance API for Java

段維瀚老師 2023-12

```
+ "?period1=" + period1
+ "&period2=" + period2
+ "&interval=1d&events=history";

System.out.println(url);

try (CloseableHttpClient httpClient = HttpClients.createDefault()) {
    HttpGet request = new HttpGet(url);
    String csvData =
EntityUtils.toString(httpClient.execute(request).getEntity());
    // 將 CSV 轉換為 JSON
    String json = convertCsvToJson(csvData);
    System.out.println(json);
}

// CSV 轉 Json
private static String convertCsvToJson(String csvData) throws IOException {
    try (CSVParser parser = CSVParser.parse(csvData,
CSVFormat.DEFAULT.withFirstRecordAsHeader())) {
        List<Map<String, String>> records = new ArrayList<>();
        for (CSVRecord record : parser) {
            Map<String, String> map = new HashMap<>();
            record.toMap().forEach(map::put);
            records.add(map);
        }
        if(records.size() == 1) {
            return new Gson().toJson(records.get(0));
        } else {
            return new Gson().toJson(records);
        }
    }
}
}
```

多筆(歷史紀錄)

```
package test;
import org.apache.commons.csv.CSVFormat;
import org.apache.commons.csv.CSVParser;
import org.apache.commons.csv.CSVRecord;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;
import com.google.gson.Gson;
import java.io.IOException;
import java.time.LocalDate;
import java.time.ZoneId;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
public class StockDataFetcher2 {
    public static void main(String[] args) throws IOException {
```

Yahoo Finance API for Java

段維瀚老師 2023-12

```
String stockNo = "0050.TW";
// 使用當前日期作為結束日期
LocalDate endDate = LocalDate.now();
// 設置開始日期為一年前
LocalDate startDate = endDate.minusYears(1);
// 將開始和結束日期轉換為 Unix 時間戳(秒為單位)
long period1 = startDate.atStartOfDay(ZoneId.of("UTC")).toEpochSecond();
long period2 = endDate.atStartOfDay(ZoneId.of("UTC")).toEpochSecond();
String url = "https://query1.finance.yahoo.com/v7/finance/download/"
    + stockNo
    + "?period1=" + period1
    + "&period2=" + period2
    + "&interval=1d&events=history";
try (CloseableHttpClient httpClient = HttpClient.createDefault()) {
    HttpGet request = new HttpGet(url);
    String csvData =
EntityUtils.toString(httpClient.execute(request).getEntity());
    // 將 CSV 轉換為 JSON
    String json = convertCsvToJson(csvData);
    System.out.println(json);
}

// CSV 轉 Json
private static String convertCsvToJson(String csvData) throws IOException {
    try (CSVParser parser = CSVParser.parse(csvData,
CSVFormat.DEFAULT.withFirstRecordAsHeader())) {
        List<Map<String, String>> records = new ArrayList<>();
        for (CSVRecord record : parser) {
            Map<String, String> map = new HashMap<>();
            record.toMap().forEach(map::put);
            records.add(map);
        }
        if(records.size() == 1) {
            return new Gson().toJson(records.get(0));
        } else {
            return new Gson().toJson(records);
        }
    }
}
```