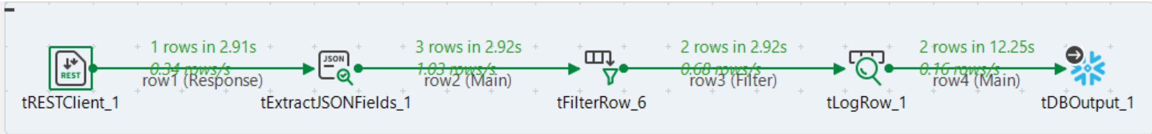
Exercise: Real-Time Company Data Integration from Public API to Snowflake using Talend

Objective

Create a Talend job that pulls company data (stock symbol, price, employee count) from a public REST API, filters the data, and loads it into a Snowflake table using JDBC.

Workflow



Tools & Technologies

- Talend Open Studio (Trial Edition)
- Public REST API: Financial Modeling Prep (https://financialmodelingprep.com)
- Snowflake (free trial or educational account)
- JDBC integration/tSnowflakeoutput

Input Data Source

REST API URL:
https://financialmodelingprep.com/api/v3/profile/AAPL,GOOGL,MSFT?apikey=demo

Note : In Place of demo paste API key

Expected JSON Response (array of company profiles):

```
[
 {
  "symbol": "AAPL",
  "price": 190.5,
  "fullTimeEmployees": 164000
 },
 {
  "symbol": "GOOGL",
  "price": 175.33,
  "fullTimeEmployees": 156719
 },
 ...
]
```

Part 1: Prepare Snowflake Environment

1. Log in to Snowflake UI.
2. Create a new database and schema:

```
  CREATE OR REPLACE DATABASE STOCKDB;
  CREATE OR REPLACE SCHEMA STOCKDB.PUBLIC;
```

3. Create the table:

```
  CREATE OR REPLACE TABLE stock_data (
    symbol STRING,
    price FLOAT,
    fullTimeEmployees NUMBER
  );
```

Part 2: Build Talend Job

1. Create a new job in Talend: LoadCompanyDataToSnowflake
2. Add components: tRESTClient, tExtractJSONFields, tFilterRow, tLogRow, tSnowflakeOutput (or tJDBCOutput)

3. Configure tRESTClient:

- Http Method: GET

Accept Type: JSON

- URL:

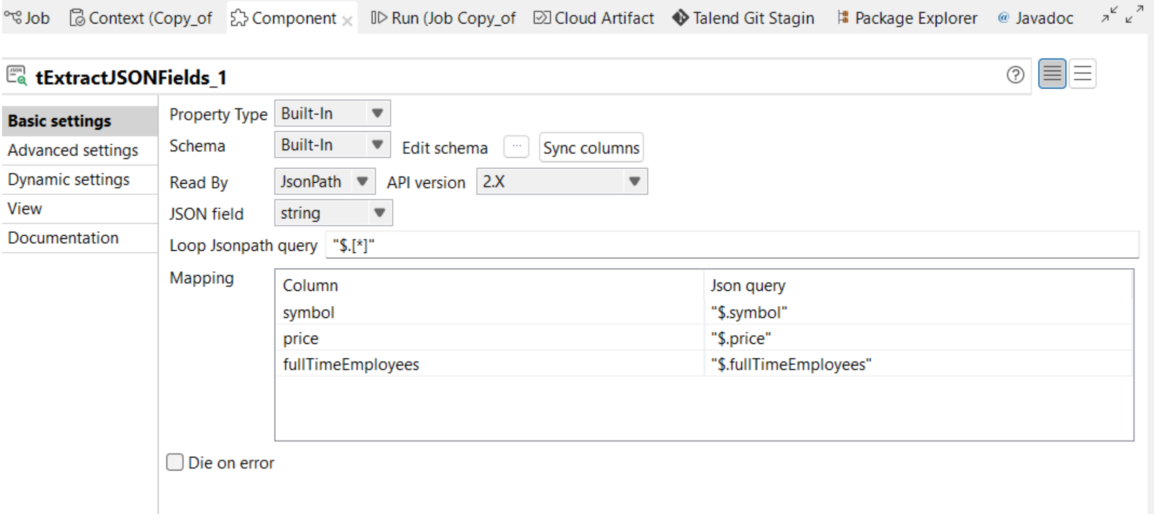
"https://financialmodelingprep.com/api/v3/profile/AAPL,GOOGL,MSFT?apikey=x3EtydwLqr3szst96y7XiBuh0OTnVbzc"

In advanced Settings: Disable :: Convert Response to DOM Document

4. Configure tExtractJSONFields:

Add 3 output columns in tExtractJSONFields,

Symbol : Type->String, price: Type->Float, fullTimeEmployees: Type-> Integer
- JSON Field: string
- Read By: JSONPath
- Loop: "$.[*]"
- Mappings:
  symbol → $.symbol
  price → $.price
  fullTimeEmployees → $.fullTimeEmployees



tExtractJSONFields_1

Property Type: Built-In

Schema: Built-In

Read By: JsonPath API version: 2.X

JSON field: string

Loop Jsonpath query: "$.[*]"

Mapping

Column	Json query
symbol	"$.symbol"
price	"$.price"
fullTimeEmployees	"$.fullTimeEmployees"

☐ Die on error

5. Configure tFilterRow:

Click on SYNC Columns

Input column: fullTimeEmployees, Function:Empty, Operator: Greater Or equal to,value:180000

6. Configure tLogRow to show the results.

Steps:

1. Drag tLogRow onto the Job canvas.
2. Connect it to the previous component (usually tFilterRow or tMap):
   - Right-click the previous component → Row → Main → Connect to tLogRow.
3. Double-click tLogRow to open its properties.
4. Set:
   - Mode: Table (or Vertical for better readability)
   - Print header: true (default)

- Print component name in console: optional

5. Click OK.

When you run the job, this will print filtered and mapped rows like:

symbol	**price**	**fullTimeEmployees**
GOOGL	175.3	185719
MSFT	478.7	228000

7. Configure tSnowflakeOutput.

Steps:

1. Drag tSnowflakeOutput into your job.
2. Connect: tFilterRow (or tMap) → Row → Main → tSnowflakeOutput
3. Double-click tSnowflakeOutput to configure.

Setting	**Value**
Property Type	Built-In
Account	Your account identifier
Username	your Snowflake login name
Password	your password
Warehouse	COMPUTE_WH
Database	STOCKDB
Schema	PUBLIC
Table	stock_data
Action on table	None or Create if not exists

Setting	**Value**
Action on data	Insert
Use Batch	true (recommended)

Part 3: Run and Verify

1. Run the job in Talend.
2. Verify in Snowflake:

```
SELECT * FROM STOCKDB.PUBLIC.stock_data;
```

Learning Outcomes

- Working with REST APIs in Talend
- Parsing JSON with tExtractJSONFields
- Applying filters using tFilterRow
- Writing data to Snowflake using JDBC/ tSnowflakeoutput