**Final Project: DSC-650 : Week 11-12: Big Data Project**

**Introduction**: In the final project of DCS-650, I am using an open API data and trying to ingest data from NIFI and publish the results from the NIFI API into a SOLR and finally loading the data into HIVE tables.

**API Details**:

From the below website, <https://app.exchangerate-api.com/dashboard> , I registered and generated the key to use the API.

Name: Exchange Rate API

URL: [https://v6.exchangerate-api.com/v6/<API KEY>/latest/USD](https://v6.exchangerate-api.com/v6/%3cAPI%20KEY%3e/latest/USD)

Key: 0aad1244185b05131f15e6db

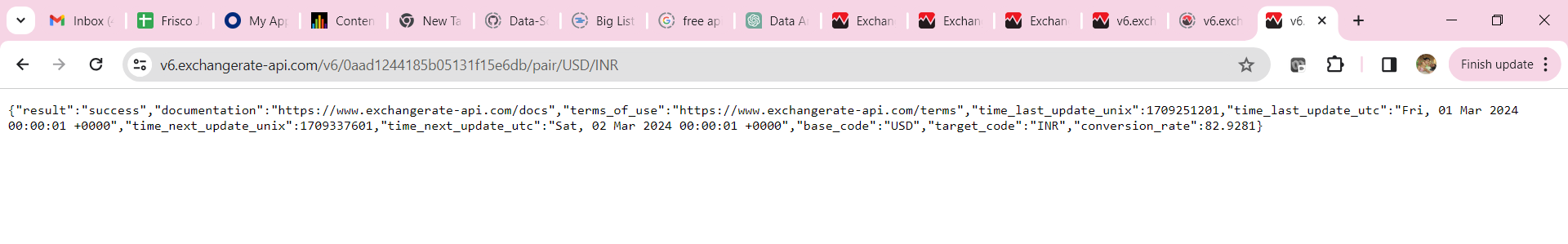
**How this API works**: The exchange rate api, will take input in 2 different ways.

1. Input as a currency pair and on request, it would provide the response with the exchange rate for the input currency pair.

E.g.: <https://v6.exchangerate-api.com/v6/0aad1244185b05131f15e6db/pair/USD/INR>

Result : {"result":"success","documentation":"https://www.exchangerate-api.com/docs","terms\_of\_use":"https://www.exchangerate-api.com/terms","time\_last\_update\_unix":1709251201,"time\_last\_update\_utc":"Fri, 01 Mar 2024 00:00:01 +0000","time\_next\_update\_unix":1709337601,"time\_next\_update\_utc":"Sat, 02 Mar 2024 00:00:01 +0000","base\_code":"USD","target\_code":"INR","conversion\_rate":82.9281}

Screen Print:



1. Input with one base currency and return the exchange rate for all the supported currencies:

e.g.: <https://v6.exchangerate-api.com/v6/0aad1244185b05131f15e6db/latest/USD>

Result:

{

"result":"success",

"documentation":"https://www.exchangerate-api.com/docs",

"terms\_of\_use":"https://www.exchangerate-api.com/terms",

"time\_last\_update\_unix":1709251201,

"time\_last\_update\_utc":"Fri, 01 Mar 2024 00:00:01 +0000",

"time\_next\_update\_unix":1709337601,

"time\_next\_update\_utc":"Sat, 02 Mar 2024 00:00:01 +0000",

"base\_code":"USD",

"conversion\_rates":{

"USD":1,

"AED":3.6725,

"AFN":73.0706,

"ALL":95.9471,

"AMD":405.0915,

"ANG":1.7900,

"AOA":843.8053,

"ARS":842.2500,

"AUD":1.5392,

"AWG":1.7900,

"AZN":1.6991,

"BAM":1.8086,

"BBD":2.0000,

"BDT":109.7439,

"BGN":1.8084,

"BHD":0.3760,

"BIF":2845.9607,

"BMD":1.0000,

"BND":1.3456,

"BOB":6.9223,

"BRL":4.9727,

"BSD":1.0000,

"BTN":82.9280,

"BWP":13.7988,

"BYN":3.2579,

"BZD":2.0000,

"CAD":1.3575,

"CDF":2728.9205,

"CHF":0.8825,

"CLP":977.6216,

"CNY":7.1998,

"COP":3933.4650,

"CRC":512.9072,

"CUP":24.0000,

"CVE":101.9627,

"CZK":23.4212,

"DJF":177.7210,

"DKK":6.8894,

"DOP":58.6715,

"DZD":134.4706,

"EGP":30.9200,

"ERN":15.0000,

"ETB":56.6042,

"EUR":0.9247,

"FJD":2.2649,

"FKP":0.7916,

"FOK":6.8891,

"GBP":0.7915,

"GEL":2.6575,

"GGP":0.7916,

"GHS":12.6869,

"GIP":0.7916,

"GMD":66.7600,

"GNF":8572.3899,

"GTQ":7.8019,

"GYD":208.9477,

"HKD":7.8288,

"HNL":24.6917,

"HRK":6.9672,

"HTG":132.7184,

"HUF":363.2600,

"IDR":15732.3862,

"ILS":3.5735,

"IMP":0.7916,

"INR":82.9281,

"IQD":1307.5714,

"IRR":42012.7710,

"ISK":137.8362,

"JEP":0.7916,

"JMD":155.9939,

"JOD":0.7090,

"JPY":149.9640,

"KES":146.1194,

"KGS":89.4355,

"KHR":4082.6173,

"KID":1.5392,

"KMF":454.9253,

"KRW":1334.1613,

"KWD":0.3077,

"KYD":0.8333,

"KZT":450.3843,

"LAK":20742.0106,

"LBP":89500.0000,

"LKR":309.5922,

"LRD":192.4393,

"LSL":19.2044,

"LYD":4.8281,

"MAD":10.0875,

"MDL":17.7717,

"MGA":4526.1298,

"MKD":56.9871,

"MMK":2102.1922,

"MNT":3419.0073,

"MOP":8.0637,

"MRU":39.8124,

"MUR":45.7434,

"MVR":15.4476,

"MWK":1688.2312,

"MXN":17.0669,

"MYR":4.7584,

"MZN":63.8085,

"NAD":19.2044,

"NGN":1585.5446,

"NIO":36.7990,

"NOK":10.6120,

"NPR":132.6847,

"NZD":1.6433,

"OMR":0.3845,

"PAB":1.0000,

"PEN":3.7879,

"PGK":3.7767,

"PHP":56.1842,

"PKR":279.1544,

"PLN":3.9894,

"PYG":7300.0886,

"QAR":3.6400,

"RON":4.5921,

"RSD":108.1561,

"RUB":91.3326,

"RWF":1292.2618,

"SAR":3.7500,

"SBD":8.3336,

"SCR":14.1670,

"SDG":511.0266,

"SEK":10.3616,

"SGD":1.3456,

"SHP":0.7916,

"SLE":22.6955,

"SLL":22695.5182,

"SOS":570.8442,

"SRD":35.3422,

"SSP":1411.9169,

"STN":22.6553,

"SYP":12929.1498,

"SZL":19.2044,

"THB":35.8690,

"TJS":10.9479,

"TMT":3.4959,

"TND":3.1212,

"TOP":2.3434,

"TRY":31.2904,

"TTD":6.7539,

"TVD":1.5392,

"TWD":31.6389,

"TZS":2540.4864,

"UAH":38.0357,

"UGX":3935.6257,

"UYU":39.0980,

"UZS":12539.6897,

"VES":36.1518,

"VND":24637.8010,

"VUV":120.7257,

"WST":2.7431,

"XAF":606.5671,

"XCD":2.7000,

"XDR":0.7521,

"XOF":606.5671,

"XPF":110.3470,

"YER":250.2807,

"ZAR":19.2049,

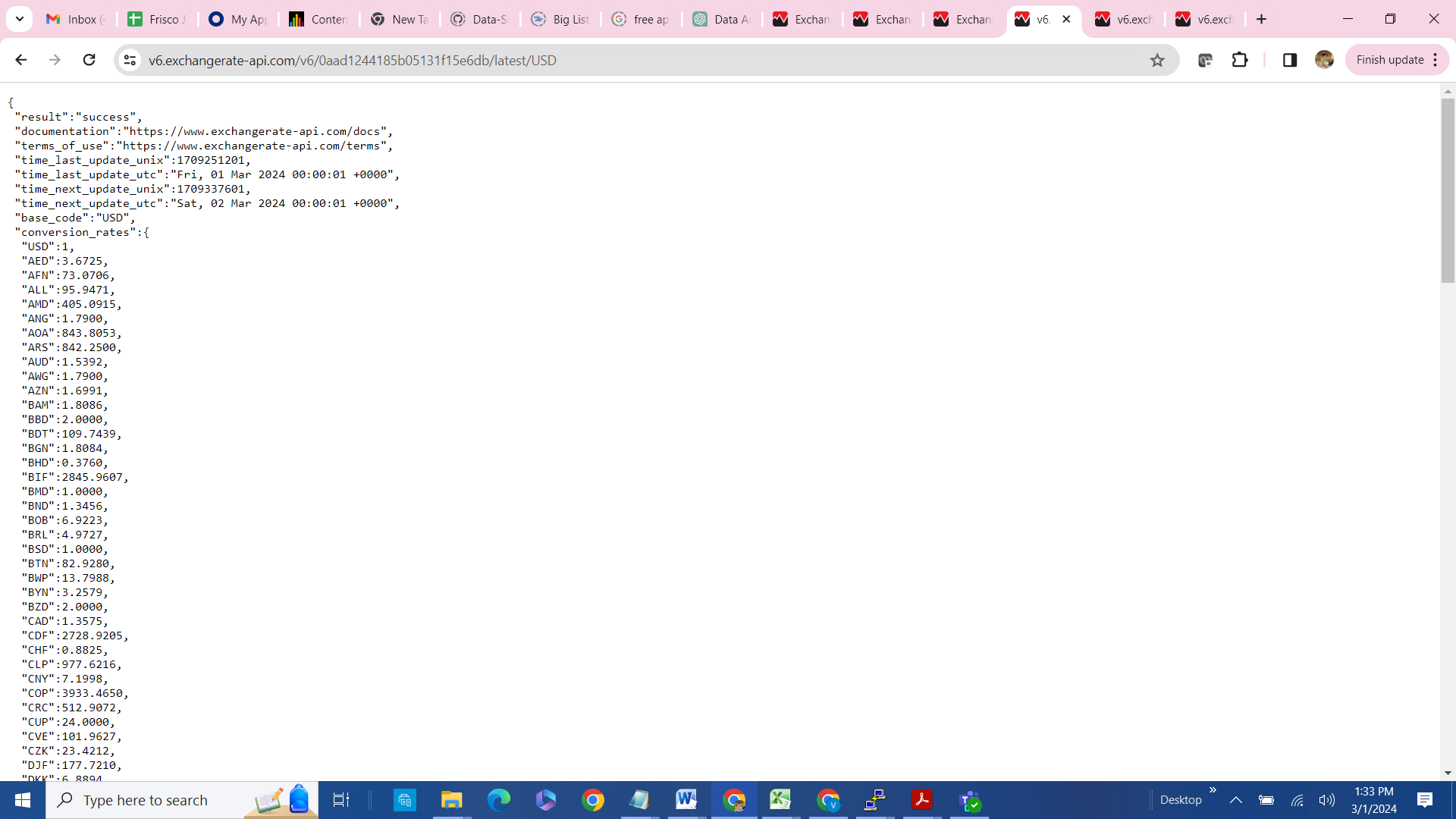
"ZMW":23.3546,

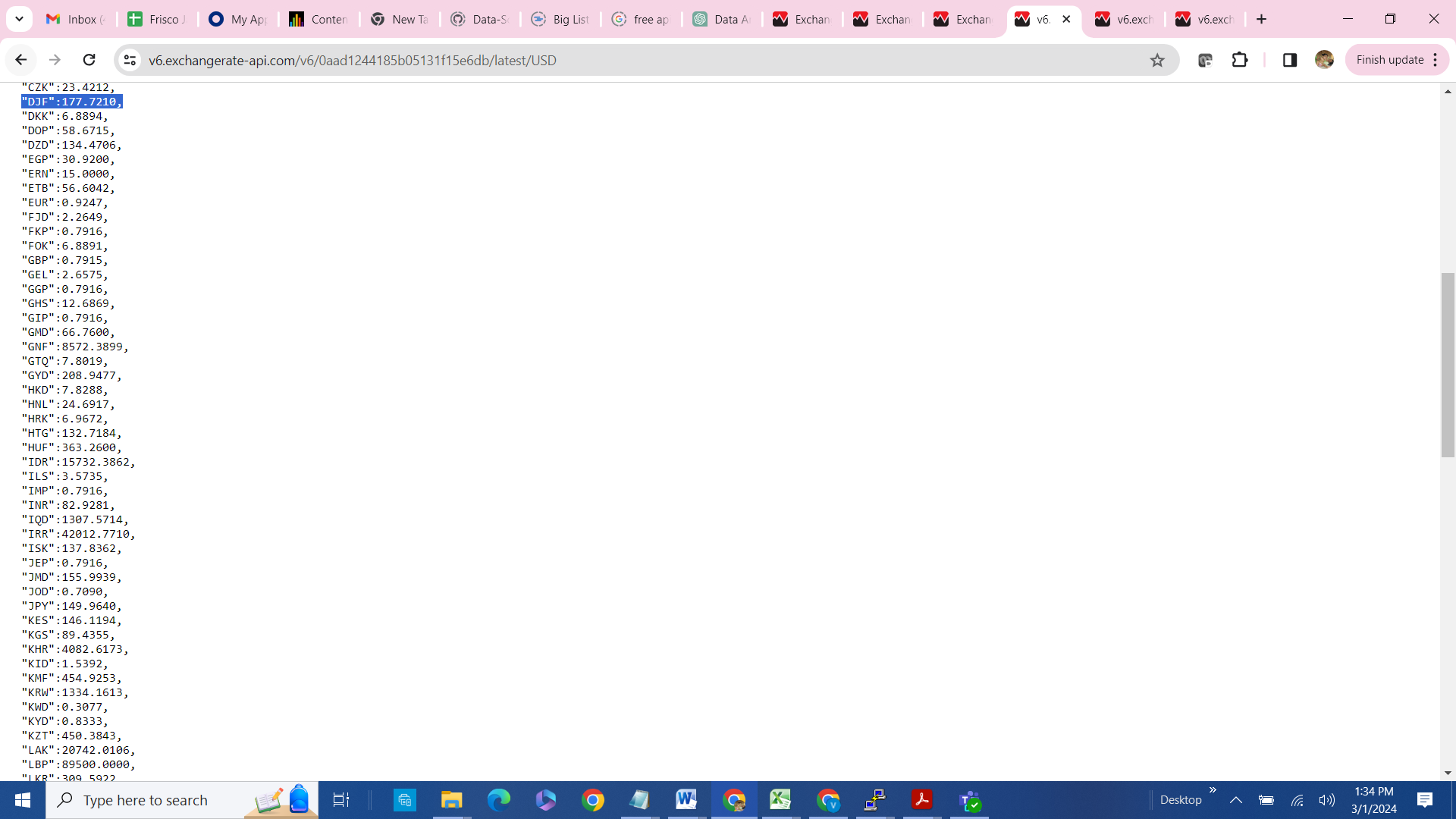
"ZWL":14764.2103

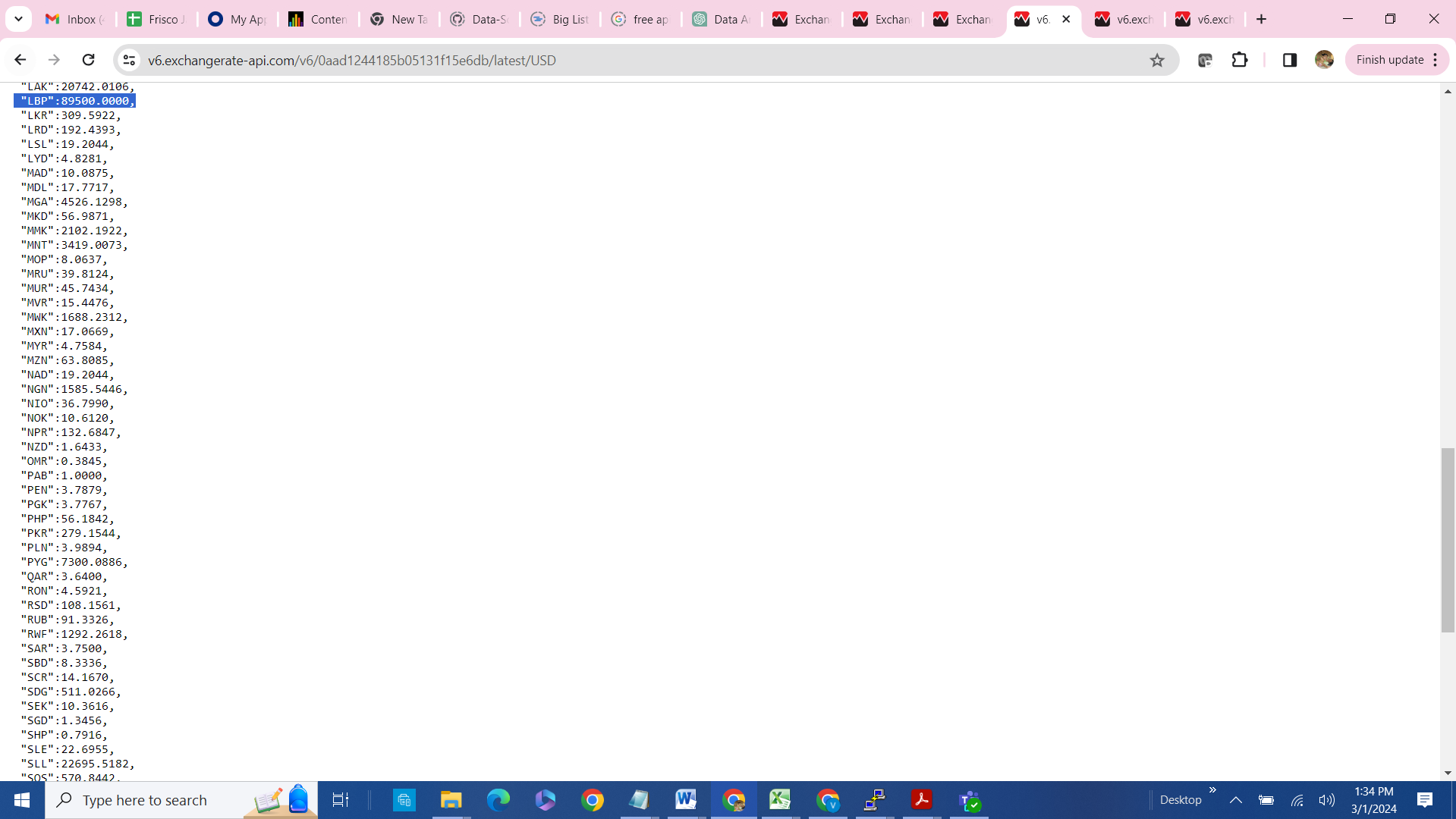
}

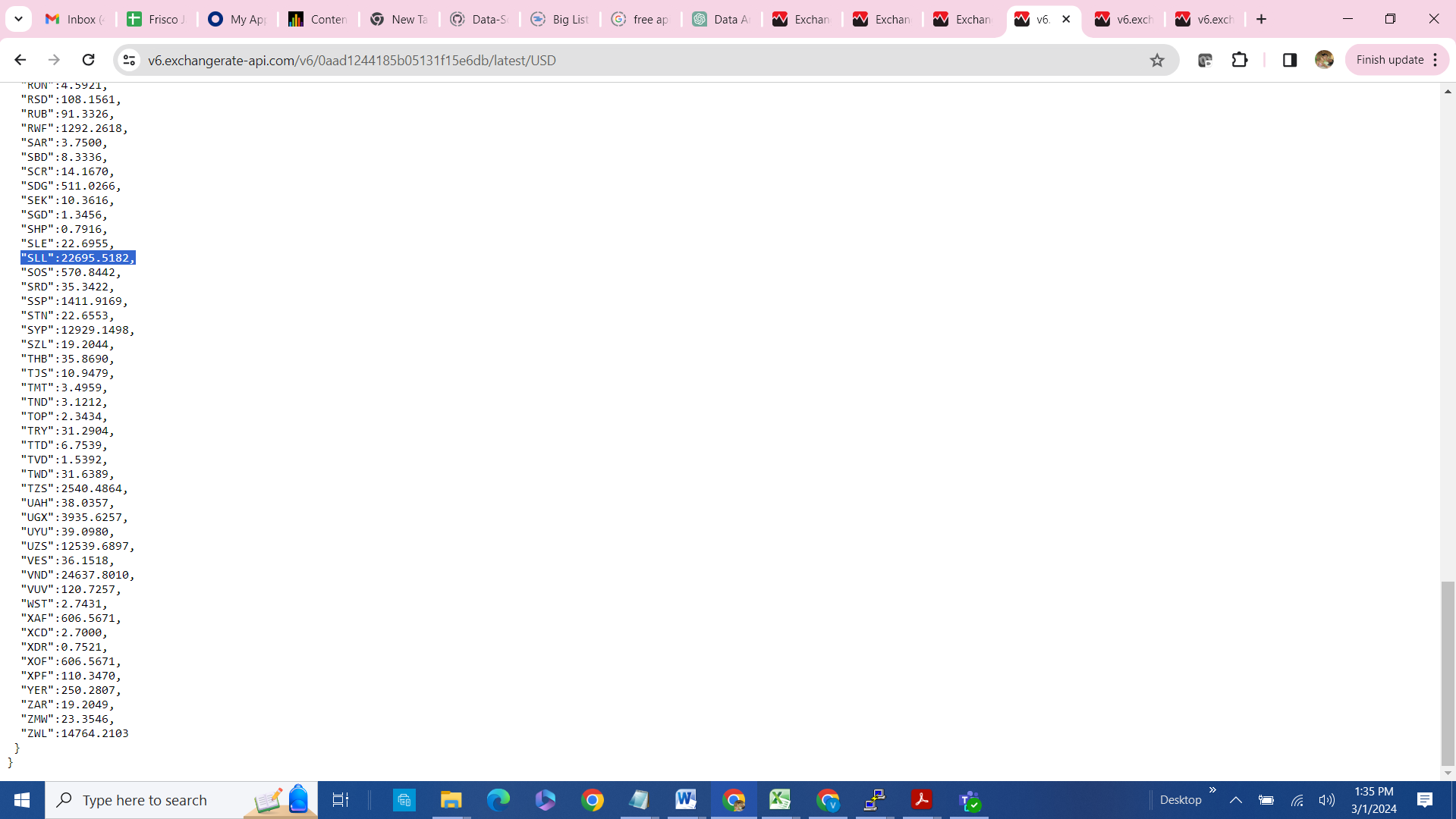
}

Screen Print:





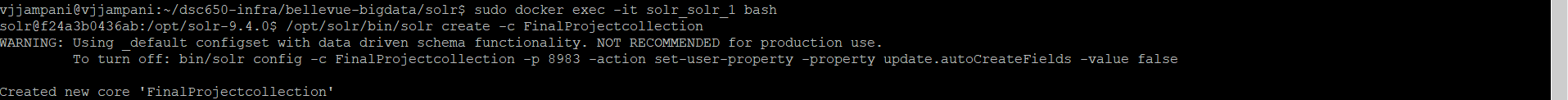




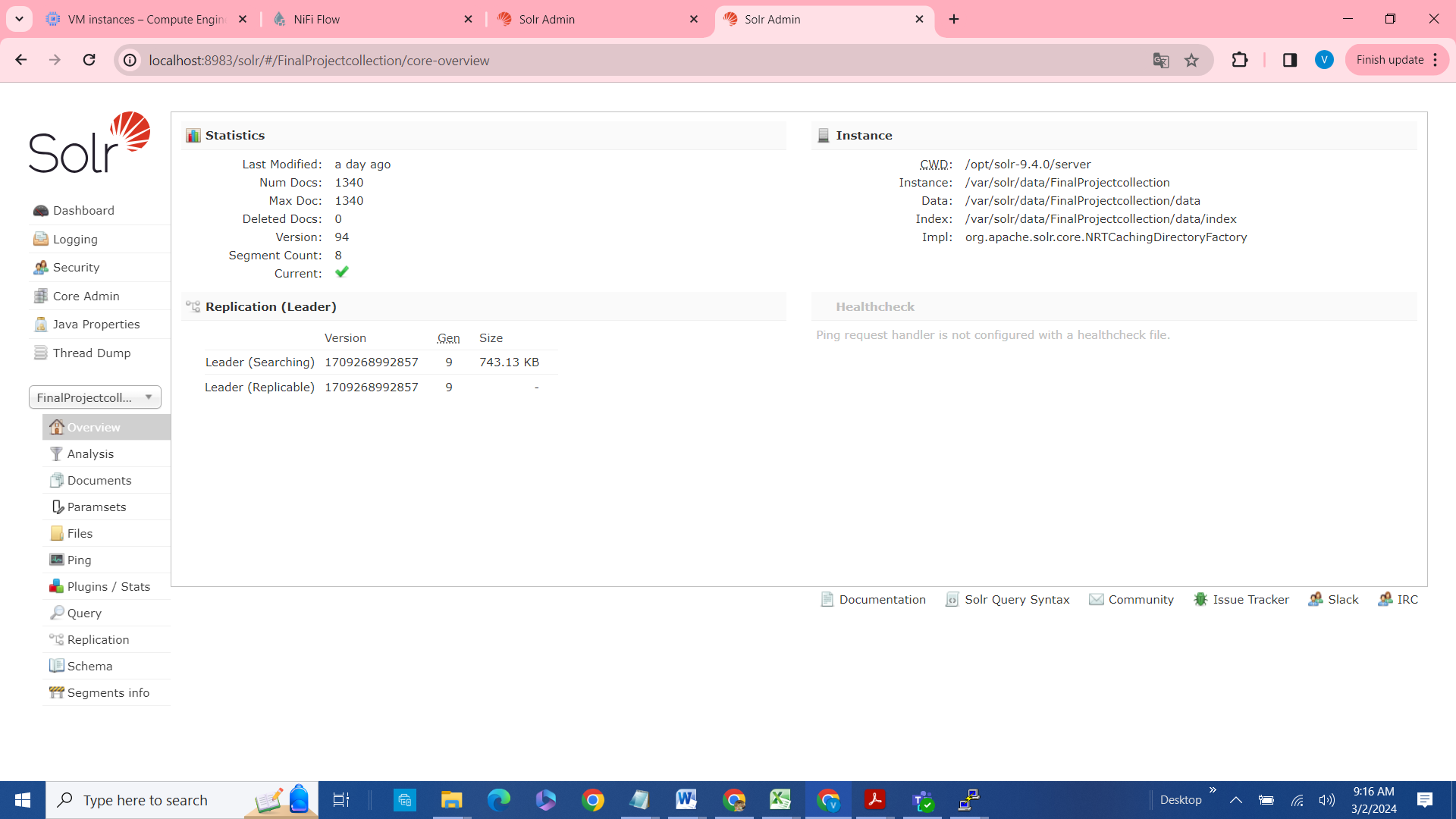
For this project, I am using the 2nd part of the API, i.e. take input as one currency and generate the exchange rate for all the supported currencies as mentioned above.

SOLR:

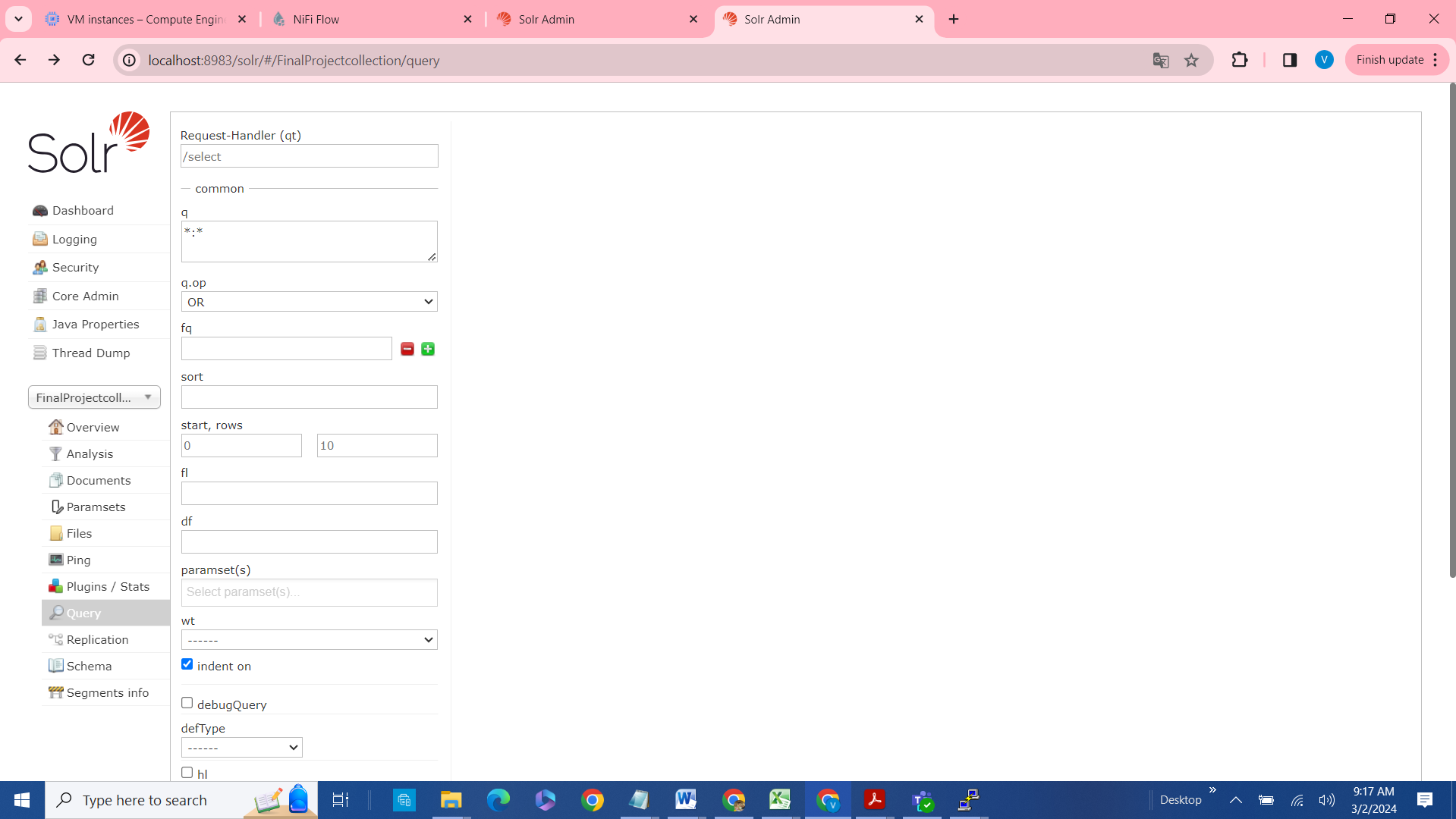
1. Collection creation: Created a SOLR collection name: FinalProjectcollection



1. Screen Print: Screen print from SOLR UI with new collection: **FinalProjectCollection**.



1. Query: Screen Print for FinalProjectCollection

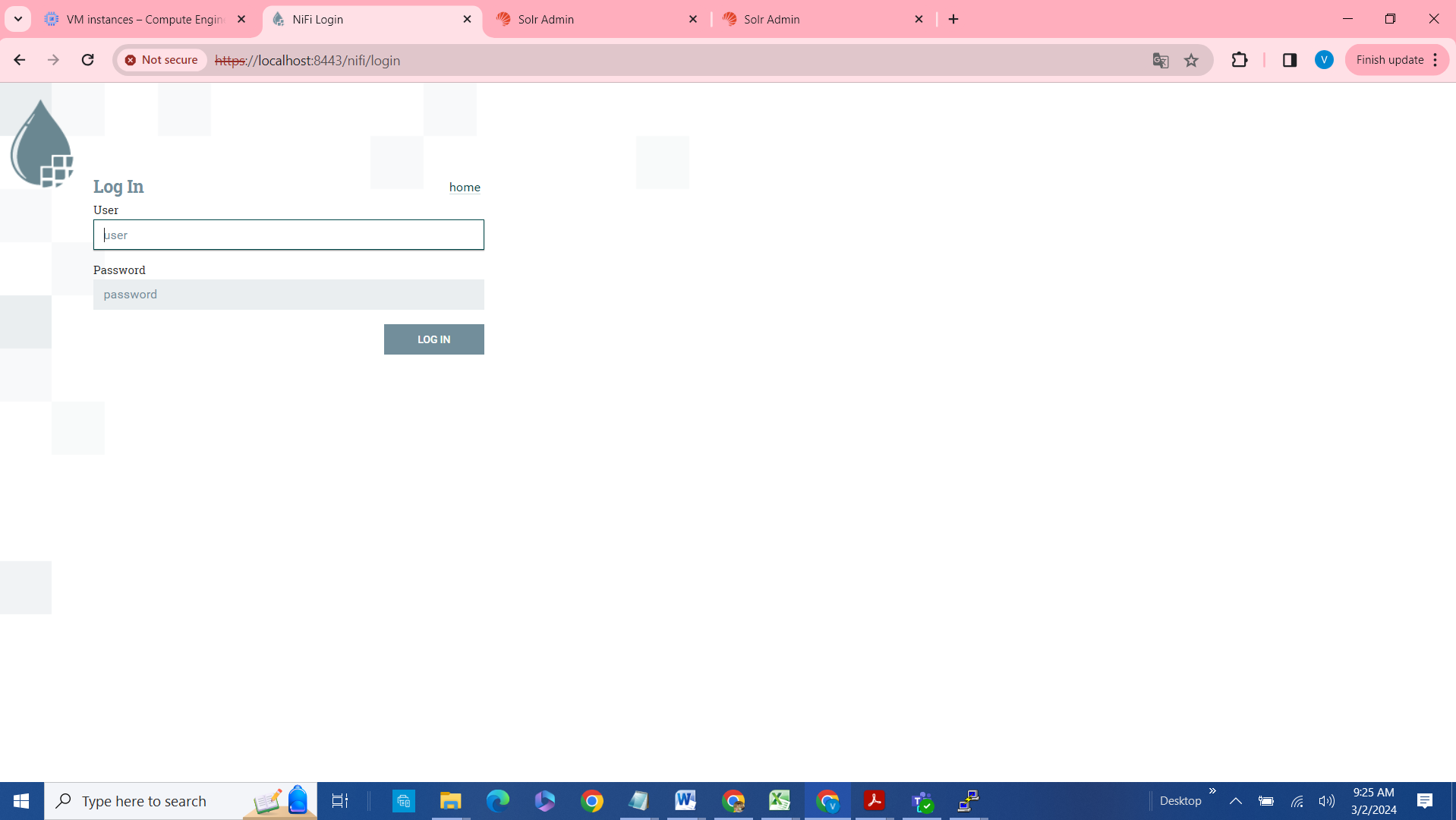


**NIFI:**

NIFI is used to ingest the data from the API source. The API used in here is **exchangerate-api.com**

**Creating process flow in NIFI with Screen Prints:**

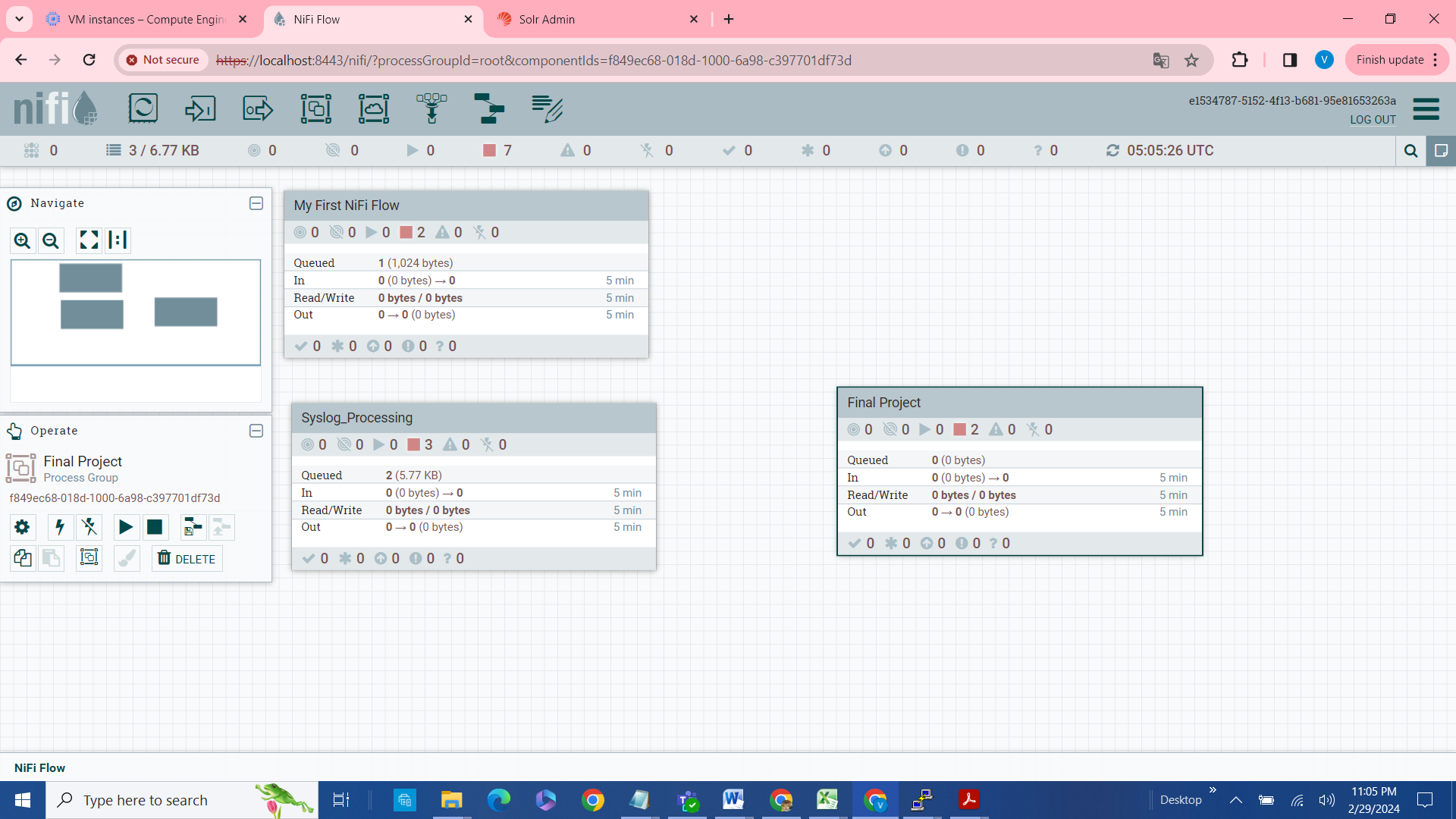
1. **Nifi web ui:**

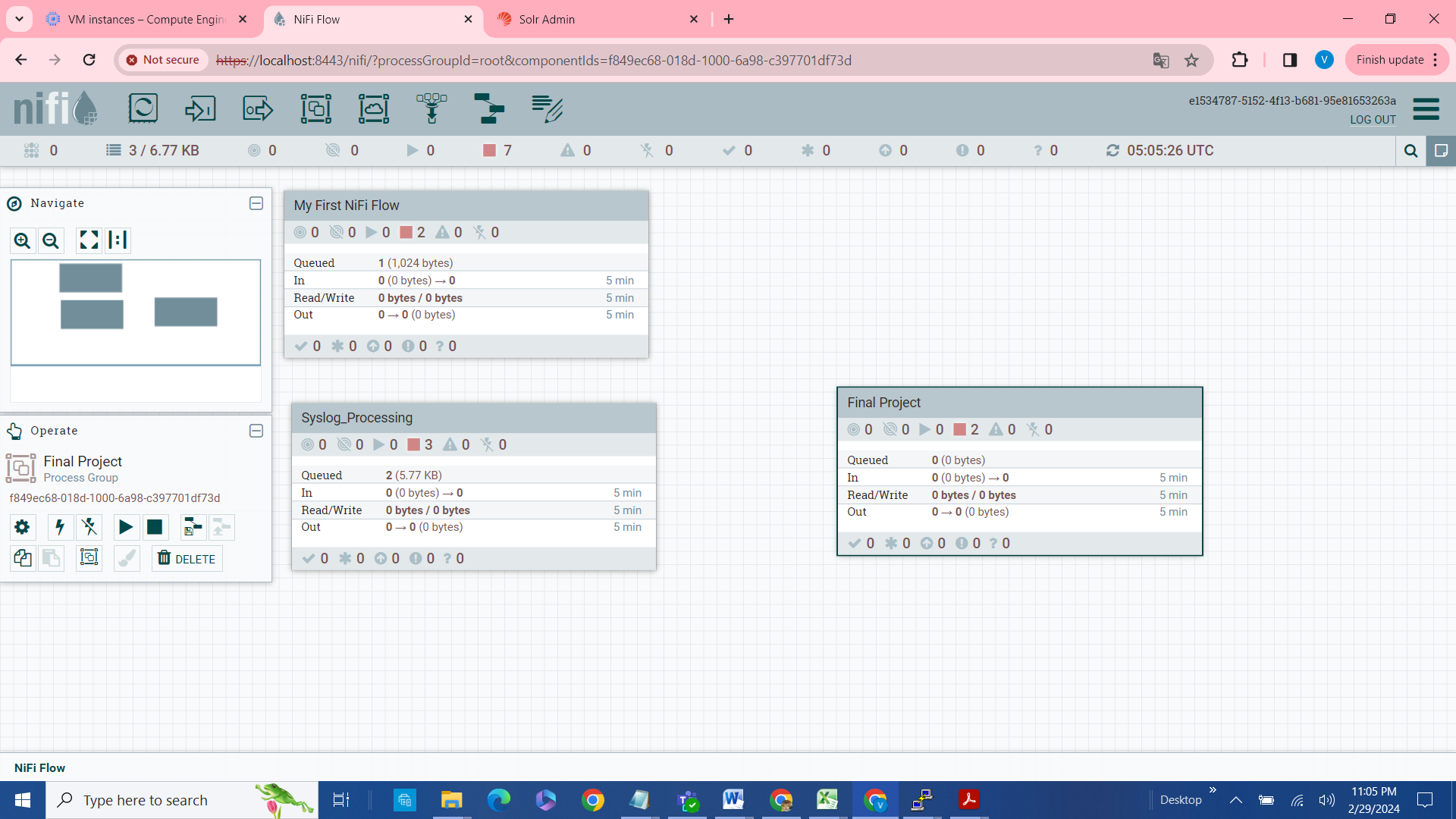


1. Login with the user name and password by executing the below from nifi directory from the terminal.
   1. grep Generated nifi-\*/logs/\*
   2. nifi-1.24.0/logs/nifi-app\_2023-12-03\_18.0.log:Generated Username [e1534787-5152-4f13-b681-95e81653263a]
   3. nifi-1.24.0/logs/nifi-app\_2023-12-03\_18.0.log:Generated Password [jqRS2MJZwKQF+8yhAyE1t/fjmnddV6vv]

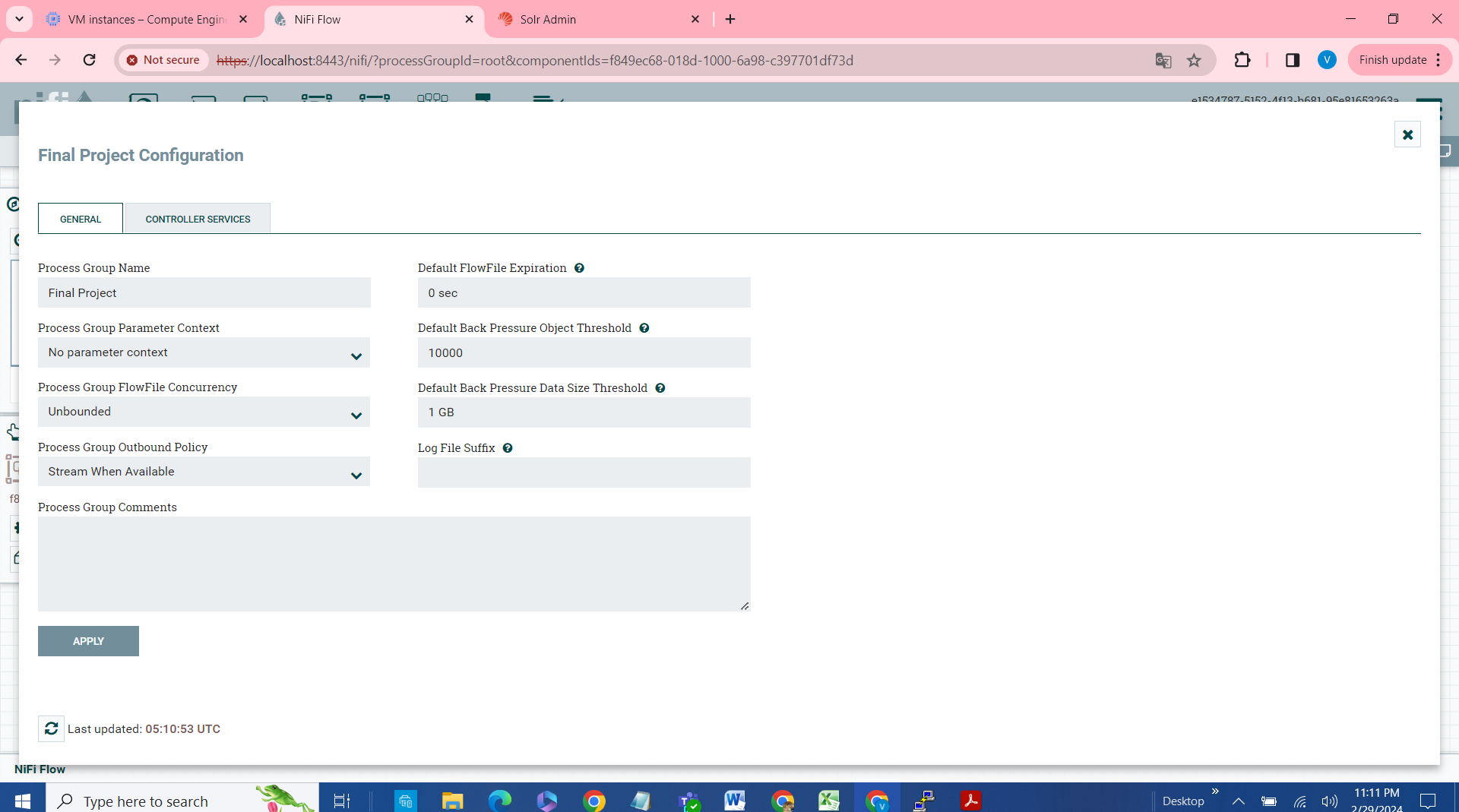


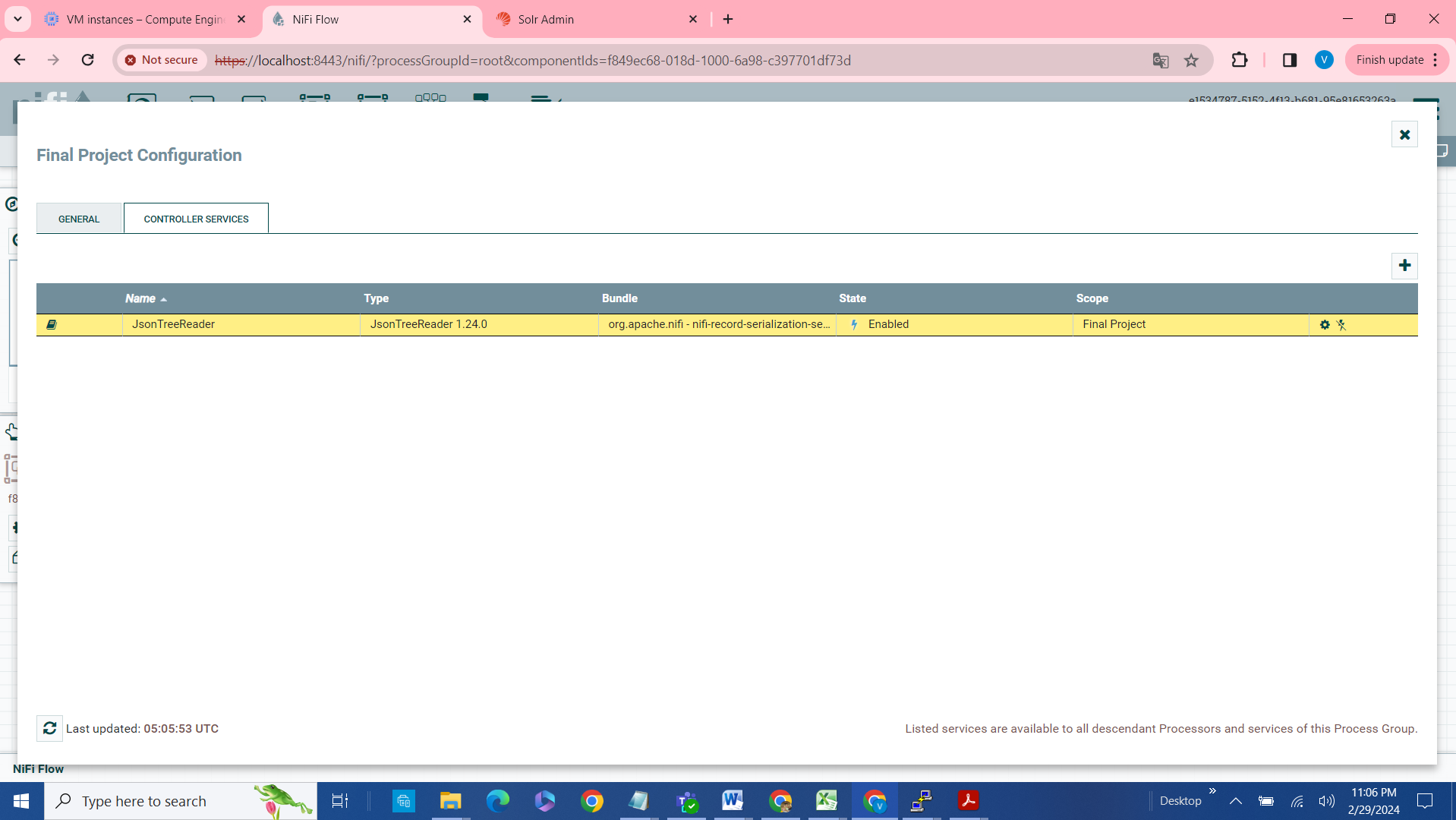
1. Create a new process group and name it as: **Final Project.**





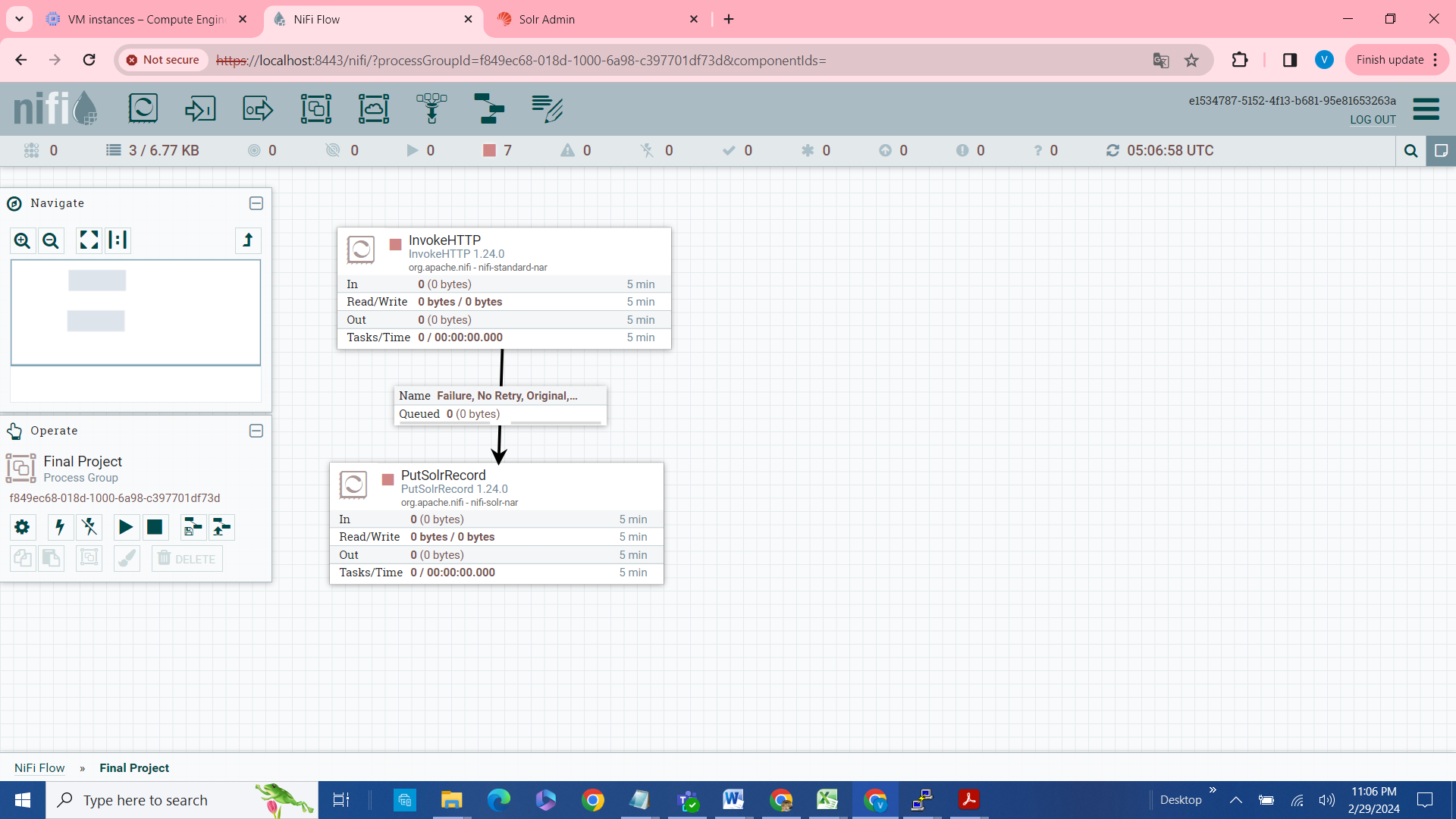
Do the configuration changes as mentioned in the below screen prints.





Add 2 processors,

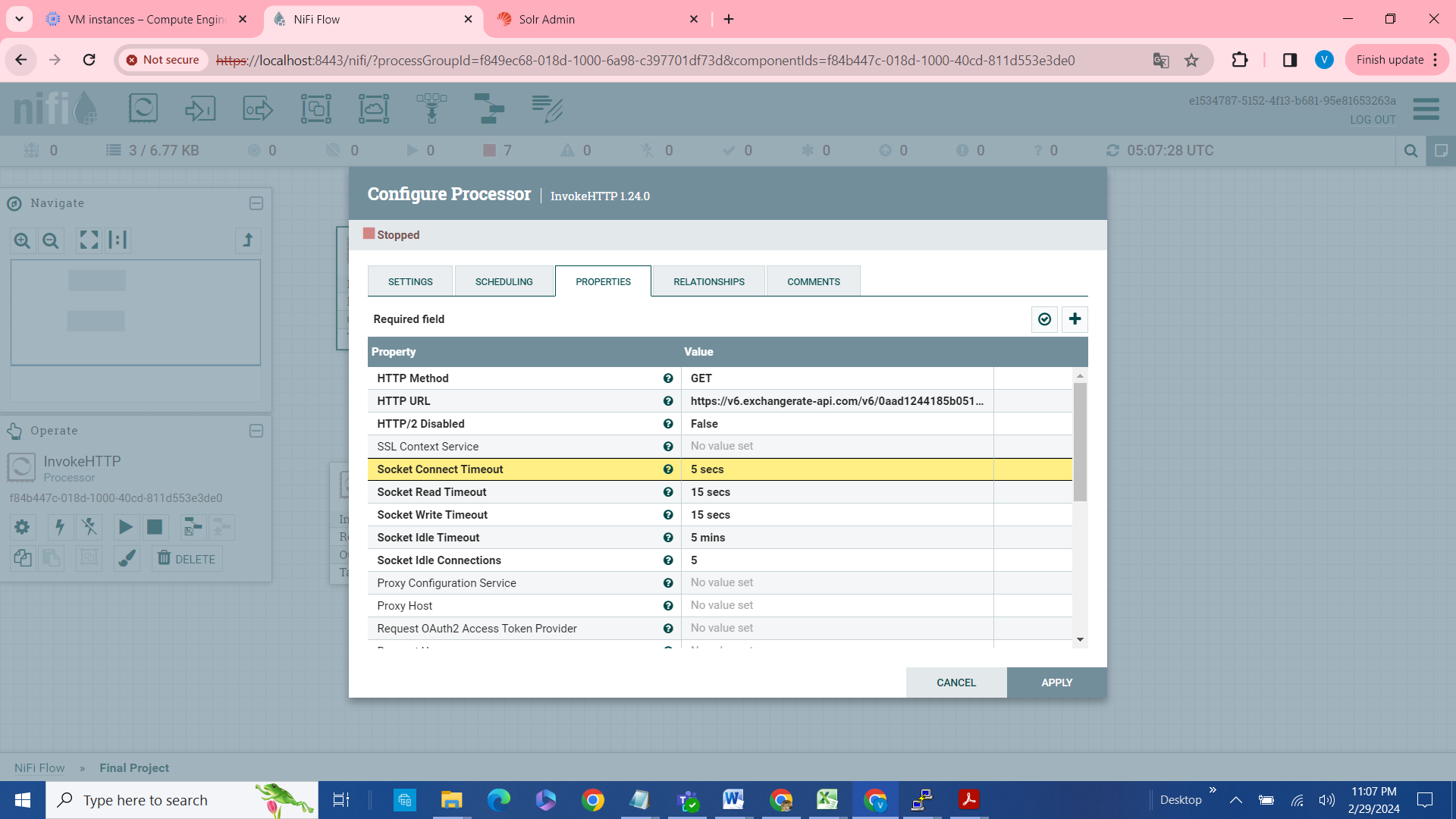
1. InvokeHTTP
2. PutSolrRecord



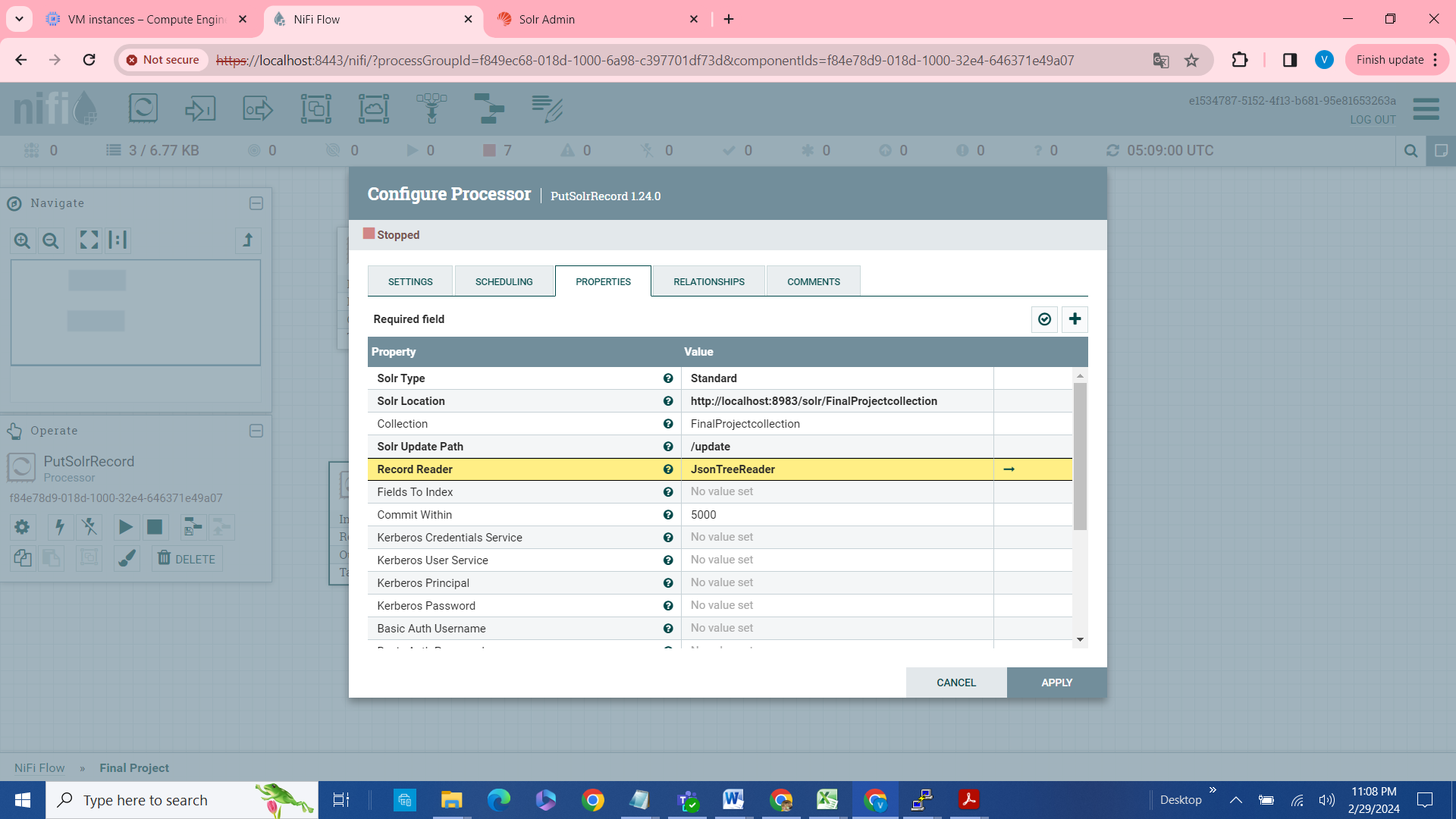
1. **InvokeHTTP**

This processor is used to read the data from the API(exchange rate).

In the http url key in the actual api url : <https://v6.exchangerate-api.com/v6/0aad1244185b05131f15e6db/pair/EUR/GBP>



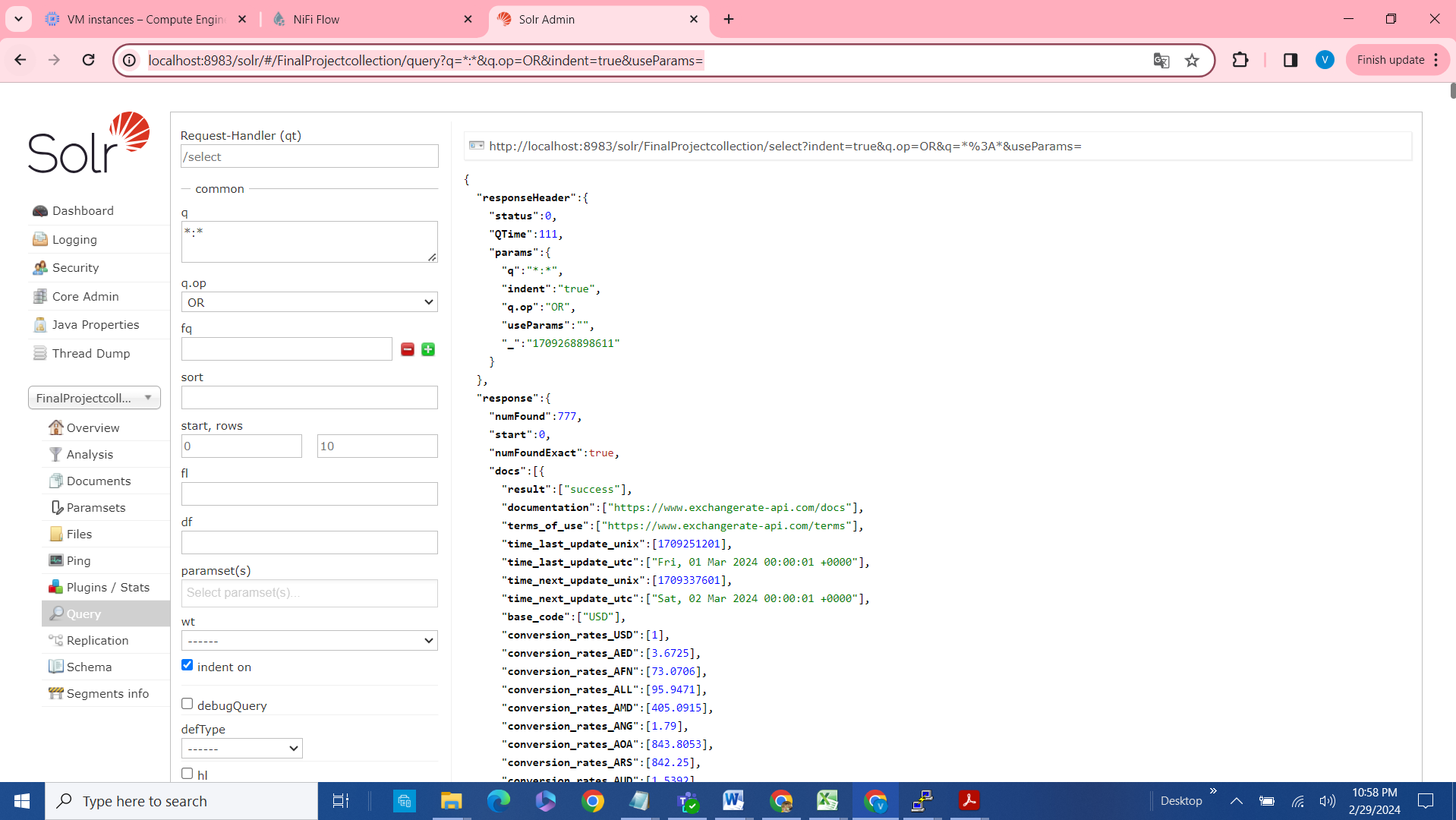
1. **PutSolrRecord processor**:
   1. This processor is used to consume the data from InvokeHttp processor and put the data into the SOLR collection.
   2. Key in the solr collection created above in the solr Location Property
   3. Start the Record Reader.

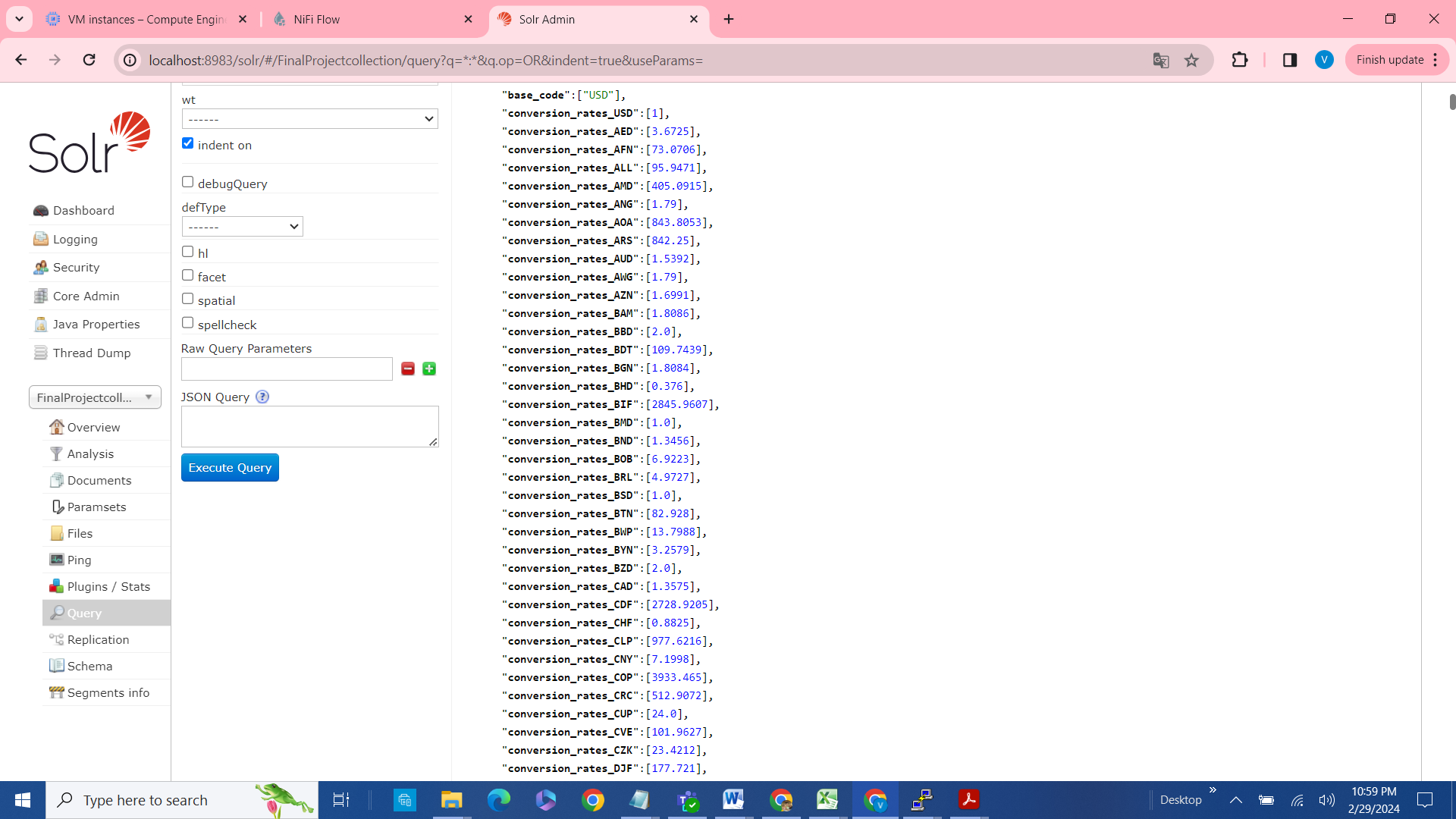


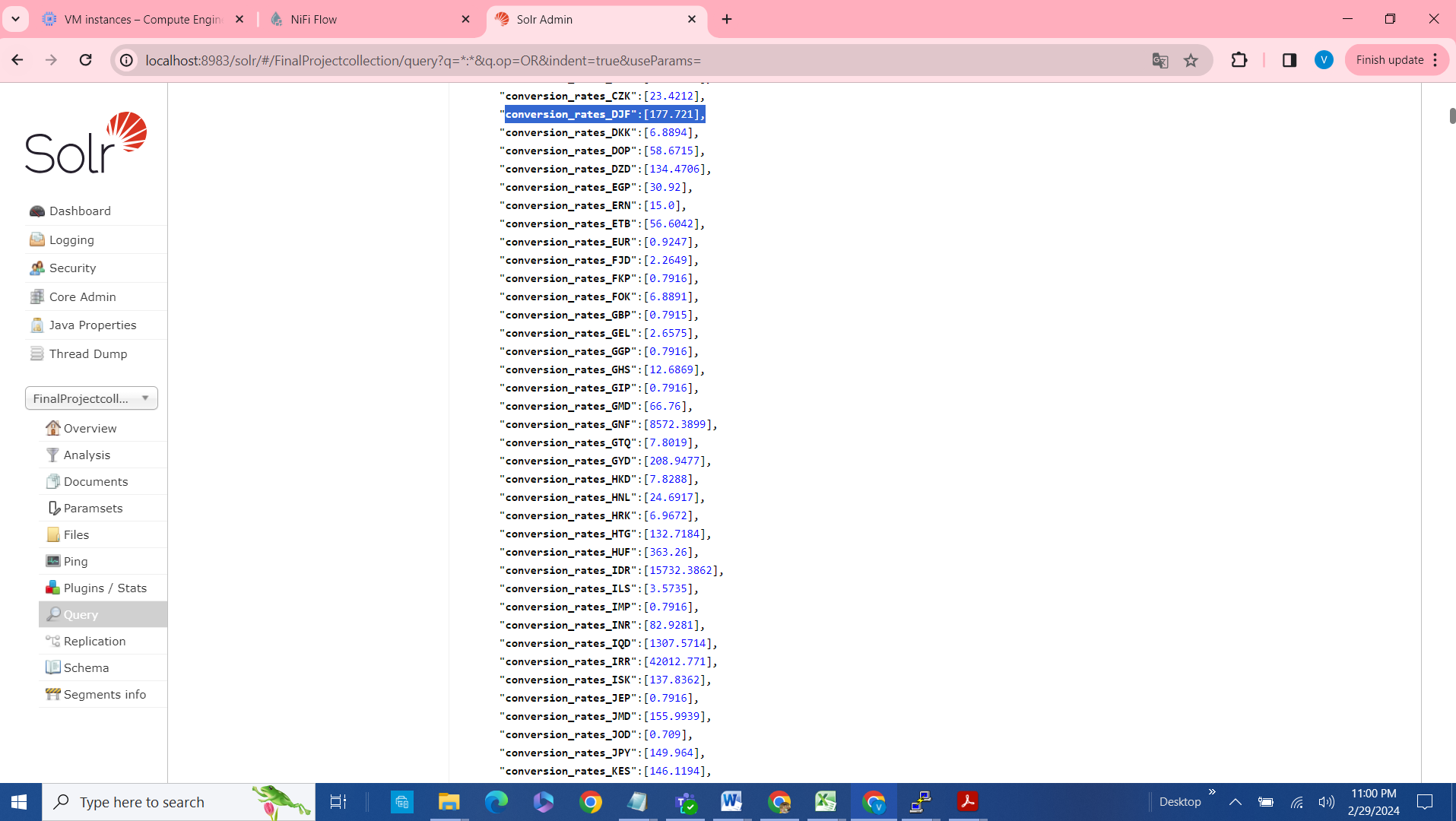
1. Create a join between the 2 processors with a connector and make sure that all the errors are clear and ready for data transformation from API to the NIFI process group.
2. Right click on each of the processor and start the process.
3. We can see the data from the API is consumed and queued in and finally put in to the solr collection.

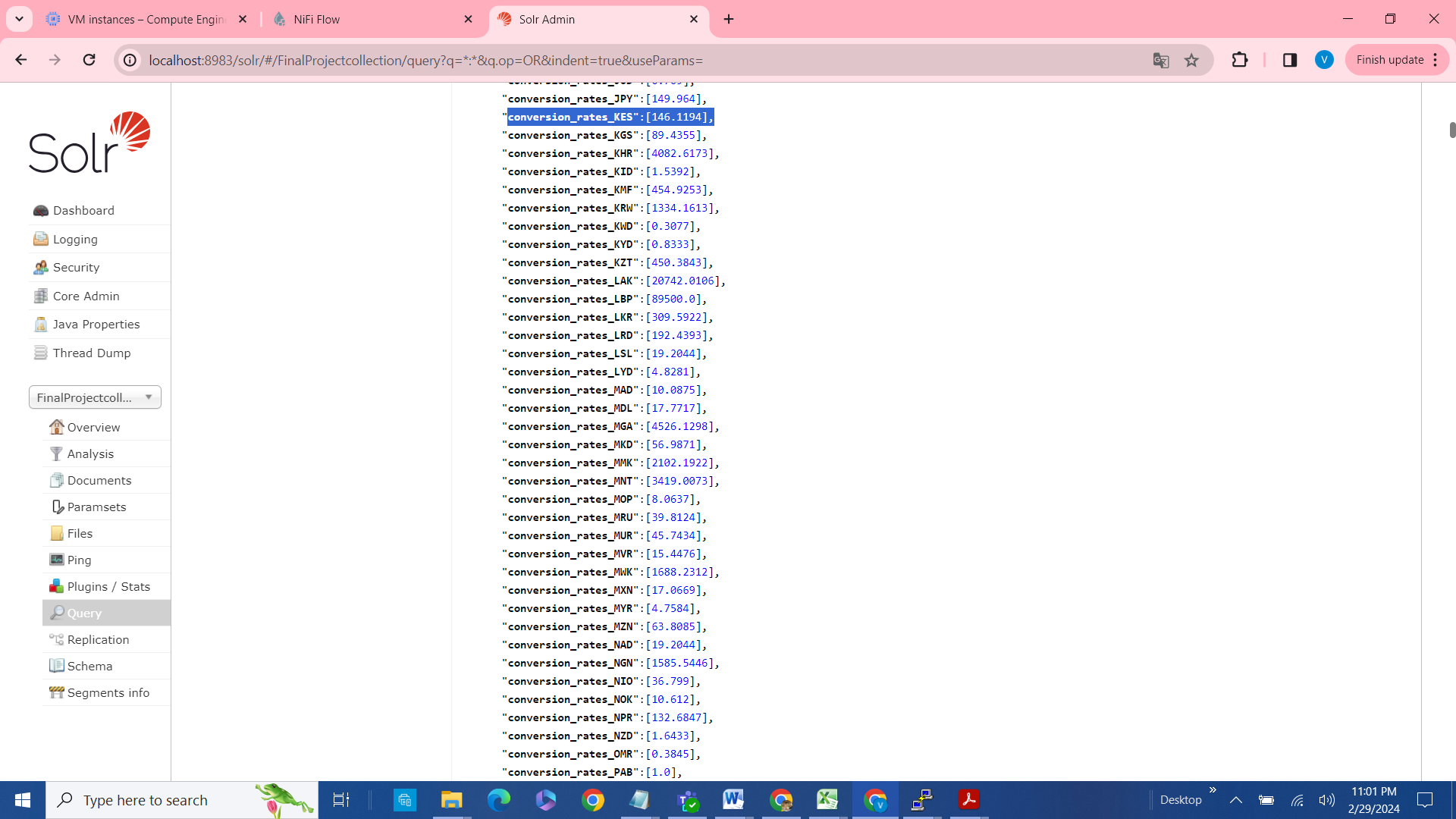
SOLR : Viewing the result set of the API, from NIFI to SOLR collection : FinalProjectcollection

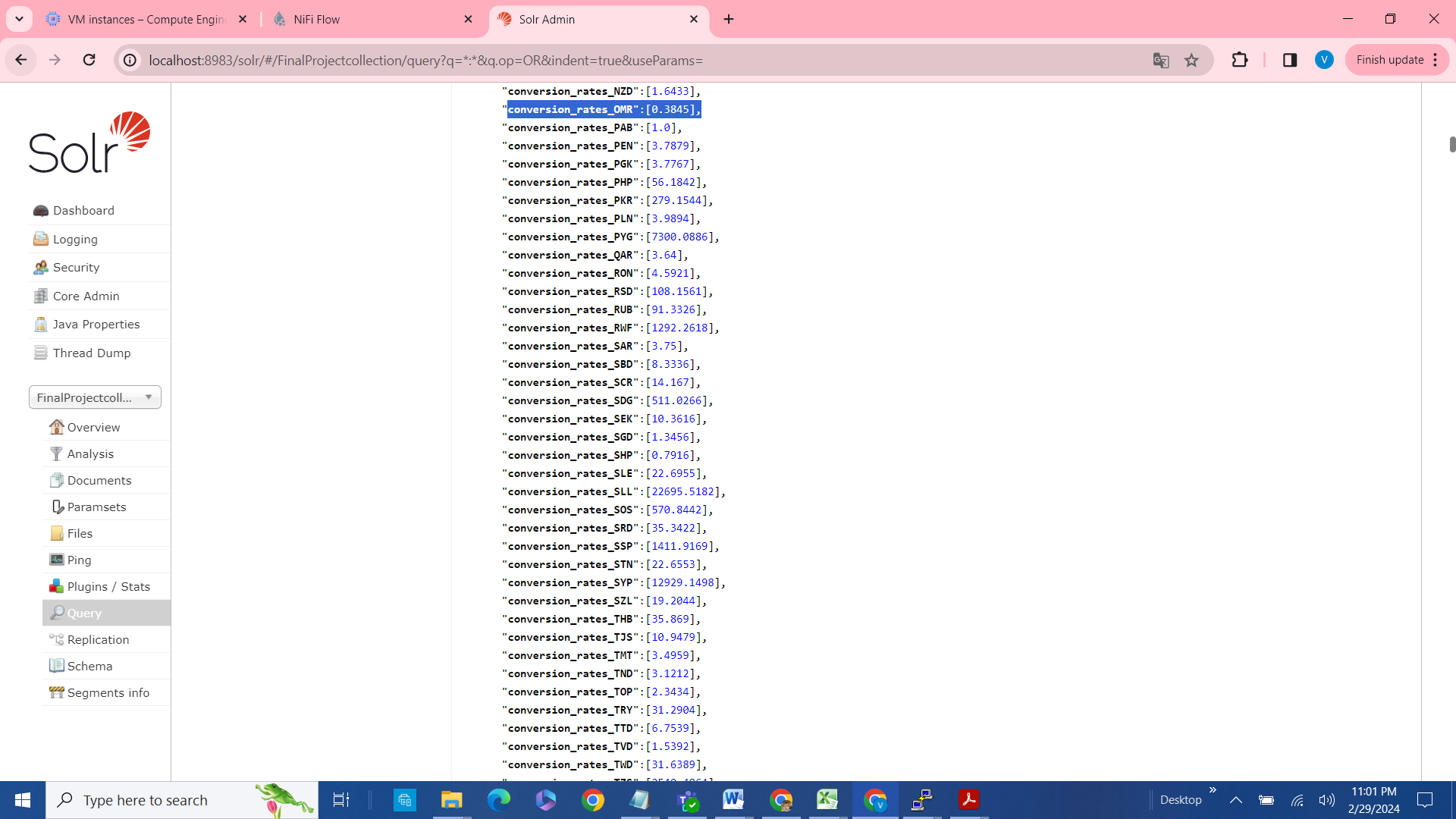
[http://localhost:8983/solr/#/FinalProjectcollection/query?q=\*:\*&q.op=OR&indent=true&useParams](http://localhost:8983/solr/#/FinalProjectcollection/query?q=*:*&q.op=OR&indent=true&useParams)=

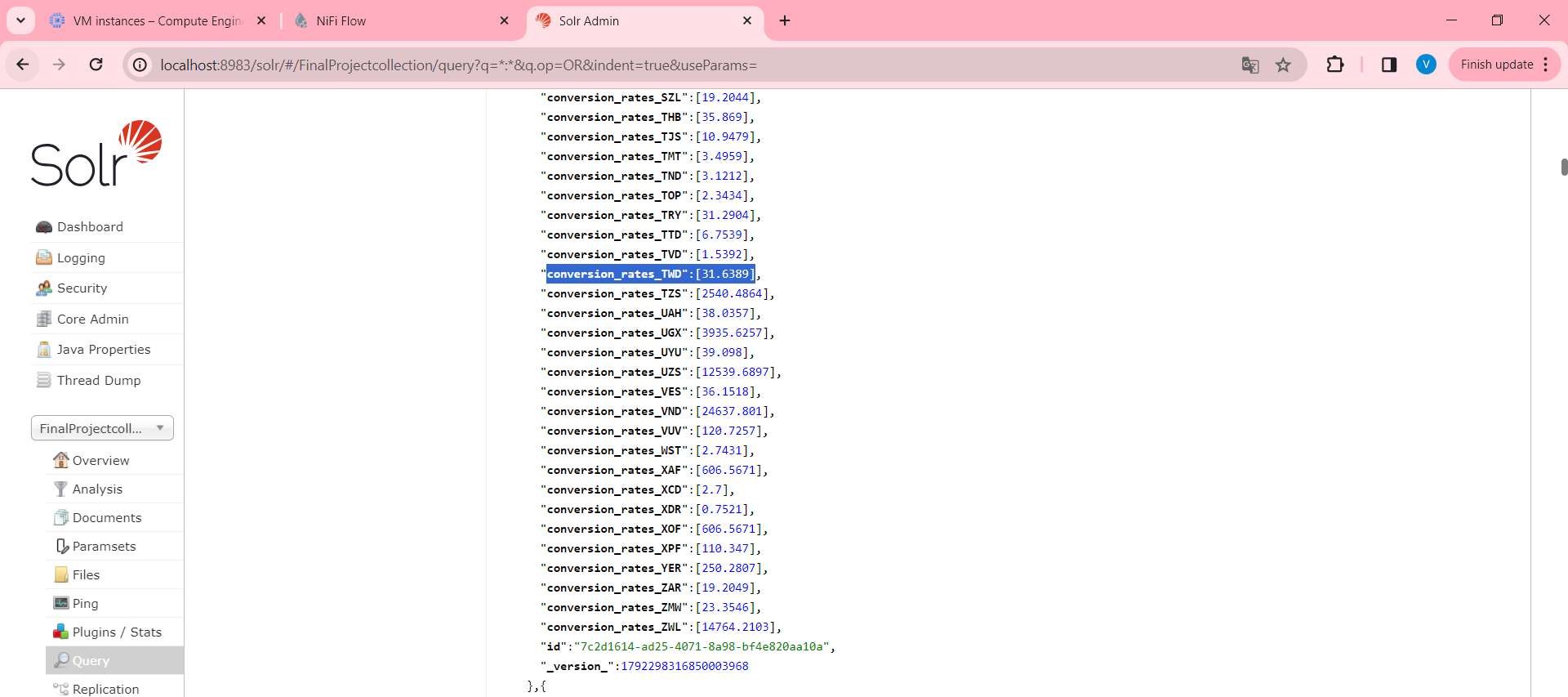












Conclusion: As a part of this project , I used an open source API, and NIFI for data ingestion and further querying the data into Solr collection.