

Cardano Software Engineering assignment

Disclaimer: When executing this assignment you will make several engineering choices as you would normally do during any SDLC during your employment. These choices should be taken as you would normally do if this was a production system ready to deliver – including among other stability, resilience, robustness, failure management – but adapted to the time slot of this assignment. You will have the opportunity to explain and debate such choices. However, its deployment and execution can remain as local on your machine. We are constraining you on one choice: use Python or C#.

Send us all your code and artifacts in a working condition. Please include instructions on how we could run it on our machines as well.

Context

Cardano has advanced business teams who can create simple Python scripts or Excel VBA scripts to perform their own self-service calculations and simulations. In this assignment you will create production-ready software that supplies the data to be pulled/imported into such scripts.

It is your choice how that software is designed and set up – including how the output data is accessible.

Requirements

Your software should take a dataset as input, enrich it and output the enriched dataset to be available to your colleagues, as explained before.

The input dataset is attached in CSV format, but also copy pasted here at the end of the document. All data should be available in the output.

The first enrichment to perform on such dataset is by adding data which is fetched from an external api. If you want a deep dive in the api, documentation and demo are present in:

<https://documenter.getpostman.com/view/7679680/SVYrrxuU?version=latest>

<https://api.gleif.org/demo>

But our use case is simple. The endpoint to use is:

[https://api.gleif.org/api/v1/lei-records?filter\[lei\]=INPUT_GOES_HERE](https://api.gleif.org/api/v1/lei-records?filter[lei]=INPUT_GOES_HERE)

Please feel free to explore other possibilities in the api, if you are certain the same data is returned.

(This external api has high availability but if it is offline or if you are offline, we are adding a sample of the endpoint output at the end of the document, so you can mock it)

The “INPUT_GOES_HERE” argument should be taken from column **lei** on each row on the input dataset.

What we expect is that fields “legalName” and “bic” on the result of the external api are part of each enriched input row which will become the output.

The second enrichment to perform is a calculation to be made on top of the existing data. This calculation is Cardano specific logic and will produce a new field (transaction_costs) to be added to the

output dataset. The logic for the calculation can be seen below, using the same names as the input dataset fields:

If the field legalAddress.country in the external api (gleif above) is “GB”, transaction_costs = notional*rate-notional

If the field legalAddress.country in the external api (gleif above) is “NL”, transactions_costs= Abs function(notional*(1/rate) - notional)

Annex

CSV for input dataset

```
transaction_utl,isin,notional,notional_currency,transaction_type,transaction_datetime,rate,lei
1030291281MARKITWIRE000000000000112874138,EZ9724VTXK48,763000.0,GBP,Sell,2020-11-25T15:06:22Z,0.0070956000,XKZZ2JZF41MRHTR1V493
0000452AMARKITWIRE97461020,EZN7BQZMQBR8,5000000.0,GBP,Buy,2020-12-17T12:15:39Z,0.0062469000,213800MBWEIJDMS5CU638
1030244641MARKITWIRE0000000000000072110232,EZNB2LV26CY9,1.957E7,GBP,Sell,2020-12-16T14:38:43Z,0.0131500000,K6Q0W1PS1L1O4IQL9C32
1030291281MARKITWIRE0000000000000080152595,EZD7JRS42975,6700000.0,GBP,Sell,2020-12-21T14:54:14Z,0.0137000000,XKZZ2JZF41MRHTR1V493
1030244641MARKITWIRE0000000000000072697124,EZQW6HTFKNZ9,1.951E7,GBP,Sell,2020-12-16T14:37:01Z,0.0136050000,K6Q0W1PS1L1O4IQL9C32
0000452AMARKITWIRE26225363,EZKK6069DP48,1.41E7,GBP,Sell,2020-12-17T11:52:14Z,0.0060063000,MP6I5ZYZBEU3UXPYFY54
0000452AMARKITWIRE106469723,EZ65LX7J3NL1,1.2E7,GBP,Buy,2020-12-15T14:17:29Z,0.0094600000,MP6I5ZYZBEU3UXPYFY54
1030244641MARKITWIRE0000000000000112880849,EZ6W26XXZTS6,1853000.0,GBP,Sell,2020-11-25T15:10:25Z,0.0061606000,K6Q0W1PS1L1O4IQL9C32
0000452AMARKITWIRE88889673,EZTH907TJ6W0,2.2E7,GBP,Buy,2020-12-17T12:05:48Z,0.0058325000,MP6I5ZYZBEU3UXPYFY54
0000452AMARKITWIRE73224269,EZSJCJ5GMYW6,8000000.0,GBP,Buy,2020-12-11T11:42:03Z,0.0054013000,BFXS5XCH7N0Y05NIXW11
0000452AMARKITWIRE188040944,EZ29KLNM4G1,2.4E7,EUR,Sell,2022-06-22T14:09:23Z,0.0028900000,BFXS5XCH7N0Y05NIXW11
0000452AMARKITWIRE189056073,EZ53S44K7JL2,1.47E7,EUR,Buy,2022-07-07T12:12:38Z,0.0023800000,BFXS5XCH7N0Y05NIXW11
1030238917C2B4C890B9D211E9984D070361192EBB,EZ6Y30GC29Z2,2.14E8,EUR,Buy,2019-08-08T11:50:00Z,0.0004800000,K6Q0W1PS1L1O4IQL9C32
0000452AMARKITWIRE189502871,EZWXXNZCG4HW5,5.0E7,EUR,Buy,2022-07-14T10:28:57Z,0.0044800000,BFXS5XCH7N0Y05NIXW11
0000452AMARKITWIRE190106524,EZ848JR3VBM1,6900000.0,EUR,Buy,2022-07-25T12:34:28Z,0.0063300000,BFXS5XCH7N0Y05NIXW11
0000452AMARKITWIRE150674490,EZH7KHTJD052,4.5E7,EUR,Buy,2019-09-27T12:32:11Z,0.0042700000,BFXS5XCH7N0Y05NIXW11
0000452AMARKITWIRE188631428,EZDYJN1BK9K2,4600000.0,EUR,Buy,2022-06-30T09:50:48Z,0.0006000000,4PQUHN3JPFGFNF3BB653
0000452AMARKITWIRE190106543,EZF6Q1GFJ316,7400000.0,EUR,Buy,2022-07-25T12:35:59Z,0.0063300000,BFXS5XCH7N0Y05NIXW11
0000452AMARKITWIRE187215187,EZ2KJCJ083B8,6.0E7,EUR,Buy,2022-06-10T14:08:07Z,0.0006300000,BFXS5XCH7N0Y05NIXW11
10302389174A38DCE0DBA811E9B389B9699A7DC0CC,EZ69M3115B39,1.1E7,GBP,Buy,2015-03-04T12:00:00Z,0.0011000000,4PQUHN3JPFGFNF3BB653
```

Example of gleif api output

```
{
  "meta": {
    "goldenCopy": {
      "publishDate": "2022-07-29T00:00:00Z"
    },
    "pagination": {
      "currentPage": 1,
      "perPage": 15,

```

```
    "from": 1,

    "to": 1,

    "total": 1,

    "lastPage": 1

  }

},

"links": {

  "first": "https://api.gleif.org/api/v1/lei-records?filter%5Blei%5D=724500ZKCJ34UP47MW77&page%5Bnumber%5D=1&page%5Bsize%5D=15",

  "last": "https://api.gleif.org/api/v1/lei-records?filter%5Blei%5D=724500ZKCJ34UP47MW77&page%5Bnumber%5D=1&page%5Bsize%5D=15"

},

"data": [

  {

    "type": "lei-records",

    "id": "724500ZKCJ34UP47MW77",

    "attributes": {

      "lei": "724500ZKCJ34UP47MW77",

      "entity": {

        "legalName": {

          "name": "Stichting Pensioenfonds Alliance",

          "language": "nl"

        },

        "otherNames": [],

        "transliteratedOtherNames": [],

        "legalAddress": {

          "language": "nl",

          "addressLines": [

            "Stroombaan 14"

          ],

          "addressNumber": null,

          "addressNumberWithinBuilding": null,

          "mailRouting": null,

          "city": "Amstelveen",

          "region": null,

          "country": "NL",

          "postalCode": "1181VX"

        },

        "headquartersAddress": {

          "language": "nl",

          "addressLines": [

            "Stroombaan 14"

          ],

          "addressNumber": null,

          "addressNumberWithinBuilding": null,

          "mailRouting": null,

          "city": "Amstelveen",

          "region": null,
```

```
        "country": "NL",
        "postalCode": "1181VX"
    },
    "registeredAt": {
        "id": "RA000463",
        "other": null
    },
    "registeredAs": "41199776",
    "jurisdiction": "NL",
    "category": "GENERAL",
    "legalForm": {
        "id": "V44D",
        "other": null
    },
    "associatedEntity": {
        "lei": null,
        "name": null
    },
    "status": "ACTIVE",
    "expiration": {
        "date": null,
        "reason": null
    },
    "successorEntity": {
        "lei": null,
        "name": null
    },
    "successorEntities": [],
    "creationDate": "1934-01-01T00:00:00Z",
    "subCategory": null,
    "otherAddresses": [],
    "eventGroups": []
},
"registration": {
    "initialRegistrationDate": "2013-12-17T23:00:00Z",
    "lastUpdateDate": "2021-12-18T06:11:44Z",
    "status": "ISSUED",
    "nextRenewalDate": "2022-12-18T17:25:27Z",
    "managingLou": "724500A9328V1MJ5349",
    "corroborationLevel": "FULLY_CORROBORATED",
    "validatedAt": {
        "id": "RA000463",
        "other": null
    },
    "validatedAs": "41199776",
    "otherValidationAuthorities": []
},
"bic": {
```

```
        "STDANL21XXX"
    ]
},
"relationships": {
    "managing-lou": {
        "links": {
            "related": "https://api.gleif.org/api/v1/lei-records/724500ZKCJ34UP47MW77/managing-lou"
        }
    },
    "lei-issuer": {
        "links": {
            "related": "https://api.gleif.org/api/v1/lei-records/724500ZKCJ34UP47MW77/lei-issuer"
        }
    },
    "field-modifications": {
        "links": {
            "related": "https://api.gleif.org/api/v1/lei-records/724500ZKCJ34UP47MW77/field-modifications"
        }
    },
    "direct-parent": {
        "links": {
            "reporting-exception": "https://api.gleif.org/api/v1/lei-records/724500ZKCJ34UP47MW77/direct-parent-reporting-exception"
        }
    },
    "ultimate-parent": {
        "links": {
            "reporting-exception": "https://api.gleif.org/api/v1/lei-records/724500ZKCJ34UP47MW77/ultimate-parent-reporting-exception"
        }
    }
},
"links": {
    "self": "https://api.gleif.org/api/v1/lei-records/724500ZKCJ34UP47MW77"
}
}
}
```