Day 4

* Format of data for a single issue is :


```
{
    "expand": "renderedFields,names,schema,operations,editmeta,changelog,versionedRepresentations",
    "id": "8399232",
    "self": "https://securityjira.wellsfargo.net/rest/api/latest/issue/8399232",
    "key": "EADS-41714",
    "fields": {
        "customfield_21521": null,
        "customfield_23700": "https://wim-jira.wellsfargo.com/servicedesk/customer/portal/441",
        "customfield_21520": null,
        "customfield_23703": {
            "self": "https://securityjira.wellsfargo.net/rest/api/2/customFieldOption/72232",
            "value": "No",
            "id": "72232",
            "disabled": false
        }
}
```


* Format of data for entire collection of issues is


```
{
        "expand":"names,schema",
        "startAt":0,
        "maxResults":5000,
        "total":1,
        "issues":
```

```
        "issues":
        [
                {
                "expand":"operations,versionedRepresentations,editmeta,changelog,renderedFields",
                "id":"8399232","self":"https://securityjira.wellsfargo.net/rest/api/latest/issue/8399232",
                "key":"EADS-41714",
                "fields":{
                        "customfield_21521":null,
                        "customfield_23700":"https://wim-jira.wellsfargo.com/servicedesk/customer/portal/441",
                        "customfield_21520":null
                },
                {
                "expand":"operations,versionedRepresentations,editmeta,changelog,renderedFields",
                "id":"8399233","self":"https://securityjira.wellsfargo.net/rest/api/latest/issue/8399232",
                "key":"EADS-41713",
                "fields":{
                        "customfield_21521":null,
                        "customfield_23700":"https://wim-jira.wellsfargo.com/servicedesk/customer/portal/441",
                        "customfield_21520":null
                }
        ]
}
```


* When we parse JSon data using alteryx, we get flattened structure with the following fields : expand, startAt, maxResults, total, issues.0.expand, issues.0.id, issues.0.key, issues.0.fields.customfield_21521, issues.1.expand, issues.1.id, issues.1.key etc . The number indicates the position of the issue in the array called issues

* To normalize column names, post flattening the json, we need to remove the numbers in between eg. we need to transform issues.0.key, issues.1.key, issues.2.key to key , and hence we parse the number within each field using regex so as to remove it. The regex to parse the whole field is issues\.([0-9]{1-5})\..* with the [0-9]{1-5} corresponding to the digits within the two decimal points.

* To use Regex, we use Regex_Replace function in the Formula tool

* Filter tool : To filter rows based on one or more columns. For filtering based on 1 column, we use Basic Filter, for conditions involving more than one column, we use Custom filter of Filter tool. Has 2 output anchors, True and False

Doubts

1. Why parse issueIndex?
2. What exactly is an anchor in Alteryx?
3. What does a crosstab tool do?

Day 5

* API request consists of :
    - Header
    - Payload/Body
    - Parameters

* Payload is the part of an API request or response that carries the actual data being transmitted. This data can take various forms, such as JSON, XML, etc.

* For GET request, most servers and clients have a limit of 8192 bytes (8 KB), If the limit is exceeded in either the browser or the server, most will just truncate the characters outside the limit without any warning

* If you need to send large data, then better use POST instead of GET. Its limit is much higher usually up to around 4 GB is allowed by the average web server.

* We use POST request when we need to send some additional info along with the request like in this case the JQL-

Doubts
1. Is request body and request payload the same thing?
2. Is there any limit on response of GET a request, or is it same as response of POST request?

References
1. https://apidog.com/articles/api-payload
2. https://stackoverflow.com/questions/5661596/do-i-need-a-content-type-header-for-http-get-requests
3. https://www.baeldung.com/cs/messages-payload-header-overhead
4. https://stackoverflow.com/questions/22069844/what-is-the-difference-between-a-request-payload-and-request-body
5. https://stackoverflow.com/questions/51429617/http-requests-body-vs-param-vs-headers-vs-data
6. https://apidog.com/blog/http-get-request-with-body/

Day 6
Learnings
* Tool container : To group a set of tools into a single container, so that we can logically group tools that together provide a functionality (easier for visual understanding and debugging)

* Macro vs container : purpose of macros is to repeat work and not have to re-create the code in Alteryx when you can reference a macro over and over again for doing the same thing. Container is for logically group parts of a huge flow into sub groups/sub sections.

* Calgary is a data retrieval engine designed to perform analyses on large scale databases containing millions of records. Calgary utilizes indexing methodology (database indexing) to quickly retrieve records.

* Dynamic select : To select columns dynamically based on conditions. For example suppose we want to select only numeric columns or only all columns beginning with a certain word or in wide data with dates as columns select only columns after today's date, then we can use this tool (present in Developer tab)

Doubts
1. Container or macro - when do we use which?
2. What is the difference bw tool container and control container?

References
1. https://community.alteryx.com/t5/Alteryx-Designer-Desktop-Discussions/macros-versus-containers/td-p/372416

Day 7

Learnings

* Worked on Alteryx flow for connecting to Gemini API

* Download tool can not only have static key-value pair as header, but it can also take dynamic headers i.e. output of the previous step can be passed as header

* Error : Invalid json payload. This was happening because in the api was not knowing the data was json. Hence in the header of the post request, added "Content-type" as "application/json"

* JSON Parse tool : To parse json content and convert it into table. Generally used as next step of Download tool. Columns names are formed using logic explained in day 5 i.e. by flattening the json and appending the name of the inner json field to the outer field name

* JSON Build tool : To create json from tabular data (this is in Laboratory tab). To create an array, we use dot-digit suffix.

The following tabular data gets converted

INPUT:
NAME, VALUE
contents.0.parts.text, What is capital of India
contents.0.role, user
generation_config.top_k, 1
generation_config.top_p, 0.5

OUTPUT
{"contents":[{"parts":{"text":"What is capital of India"},"role":"user"}],"generation_config":{"top_k":"1","top_p":"0.5"}}

So if there is a digit after ., it means an array, if there is a name, it means another dictionary

* To filter rows we use the Filter tool (refer day 4)

* To pass a JSON payload to POST request, I created the json and assigned it to a field as a string (shown below), and then passed that field in the Payload of the Download tool
'{"contents": [{"parts": {"text": "What is capital of India"},"role": "user"}], "generation_config": {"top_k": 1, "temperature": 0.5,"top_p": 0.5,"seed":42}}'

* Container : To group multiple tools. Serve 2 purposes:
    1. When we want to annotate a set of tools which together provide some functionality
    2. When we want to freeze/diable a part of the flow

* Even if one input of a tool is Frozen, the tool and all subsequent tools will not run

References
1. https://community.alteryx.com/t5/Alteryx-Designer-Desktop-Discussions/Download-Tool-issue-with-POST-json-payload-and-quotes-and-amp/td-p/877407/page/2
2. https://www.theinformationlab.nl/en/2023/04/28/building-jsons-in-alteryx-designer/

Day 8

Learnings
* Cross tab tool : Pivots data (i.e. from long to wide format). Change Column Headers is the Column containing the field name and Change Column values is the column containing the actual values.

* Transpose tool : Unpivot data (i.e. from wide format to long format)

* Both Cross tab tool and Transpose tool are in Transform tab (along with Summarize tool)

* Union tool : To do a union of 2 datasets. Has 3 different options:
    1. Auto Config By Name : Aligns the output columns by name, when columns in the input dataset are not in same order
    2. Auto Config By Position : Stacks the data using column order. The column names from first dataset are used in output
    3. Manually Configure Fields : Manually match the fields from both datasets (useful if one or both of the datasets does not have proper or consistent column names and column order)

* In union tool we can set warning or even stop flow if the columns from the 2 datasets do not match (when we use Auto Config)

* Join vs Union : Jon puts rows from 2 datasets next to each other, whereas Union puts rows from 2 datasets on top of each other

* Output tool : For Excel as output, we can
    1. Overwrite file
    2. Overwrite sheet or range
    3. Create new sheet
    4. Append to existing sheet (will give error if sheet does not exist)

* If team B wants to use output of the workflow of team A, team A can write the result of its workflow to a file/database, and team B can use that as input

Doubts

1. Can one workflow interact with another workflow?