

Enhancing Public Transport

By Chen Xiaohong



Today's Topic:

1. PROBLEM

2. PREPARATION

3. API CALL

4. DATA DISPLAY

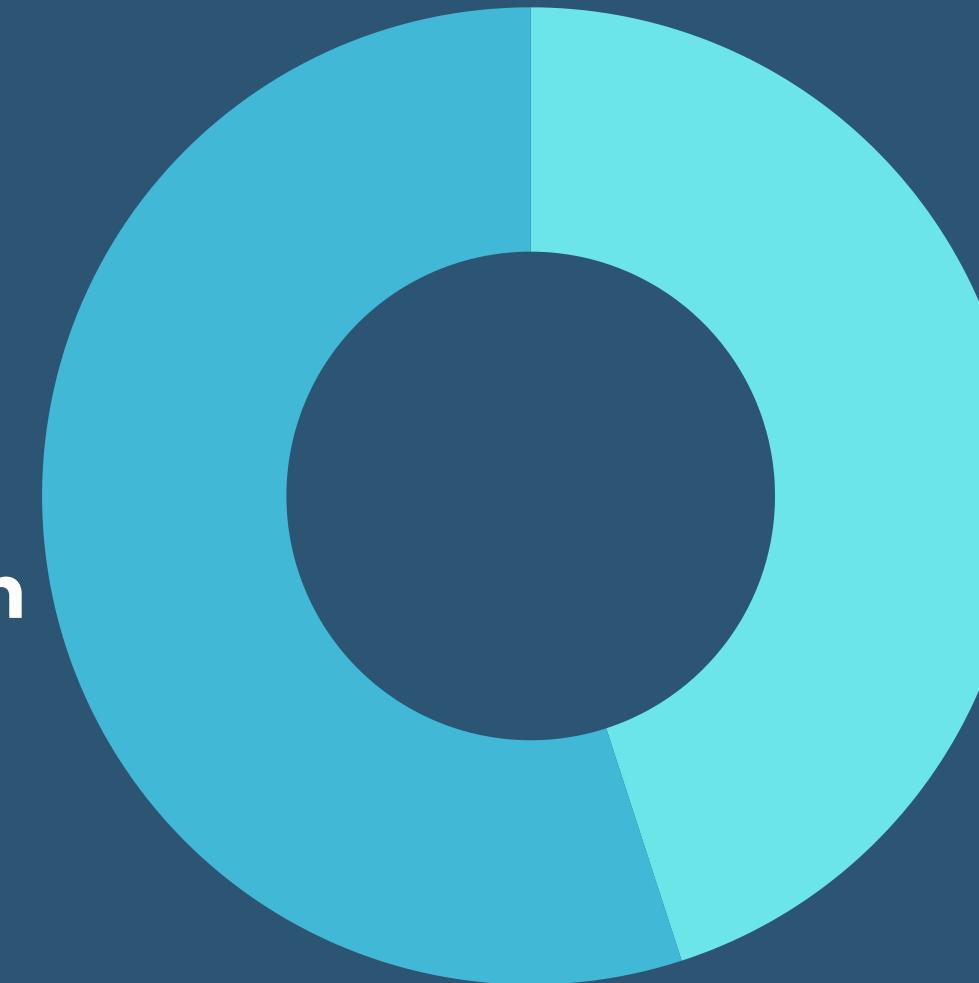
5. ENHANCEMENT

6. SUMMARY

7. REFLECTION

1) Problem

As of 2023, **7.2 million** passengers relied on public transportation **daily**



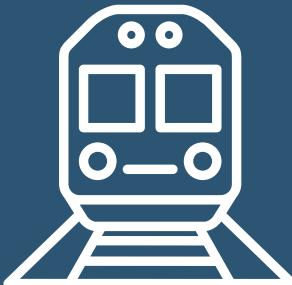
About **3.24 Million** passengers

**MRT
45%**

**Other Transportation
55%**



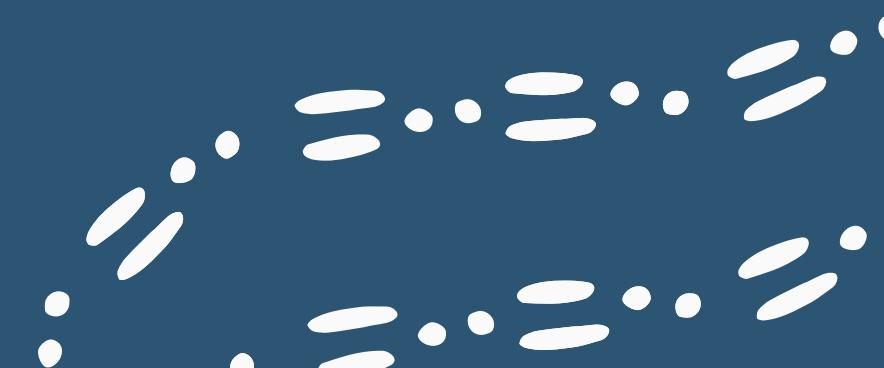
Buses



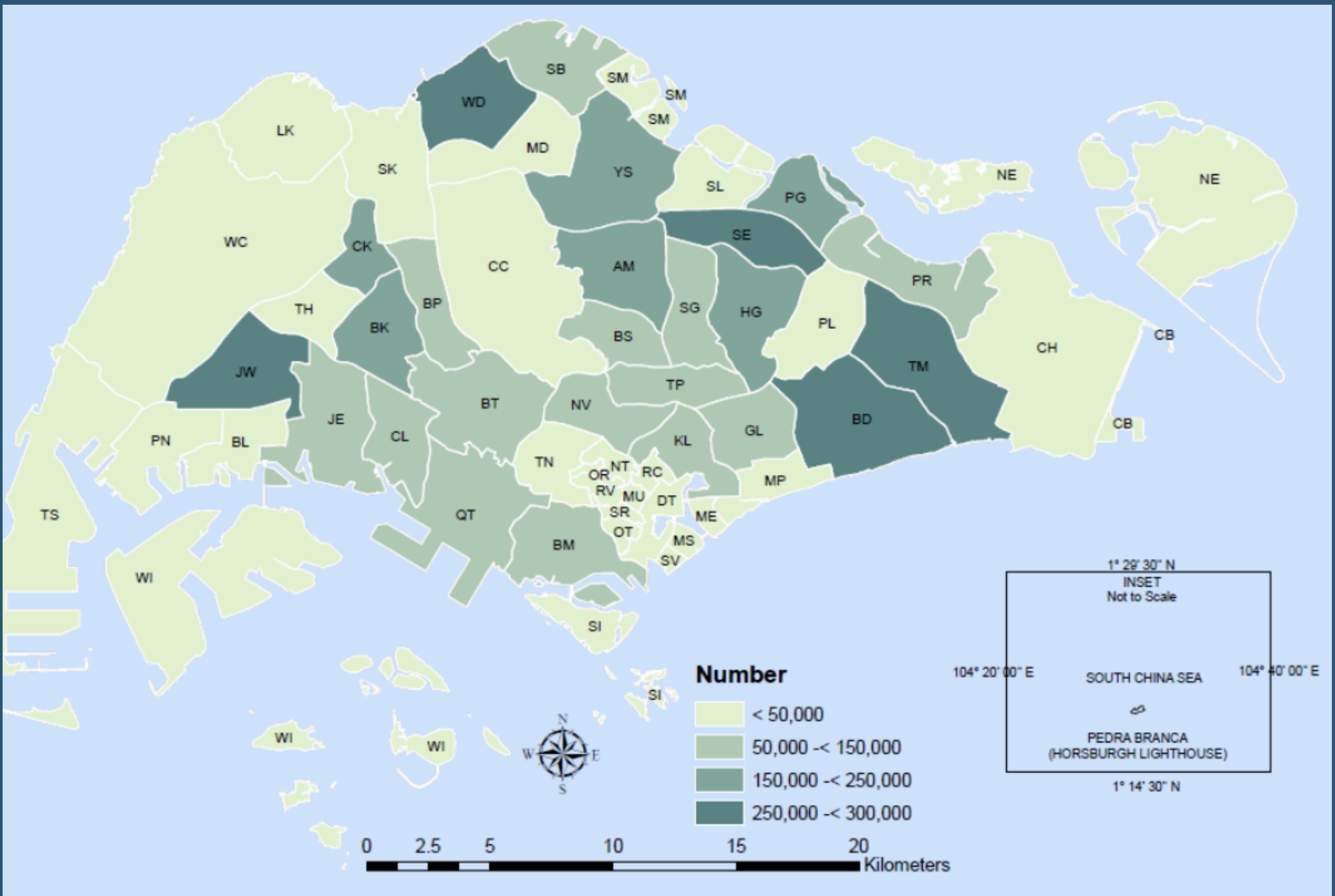
LRT



Taxi



Resident Population by Planning Area, June 2024



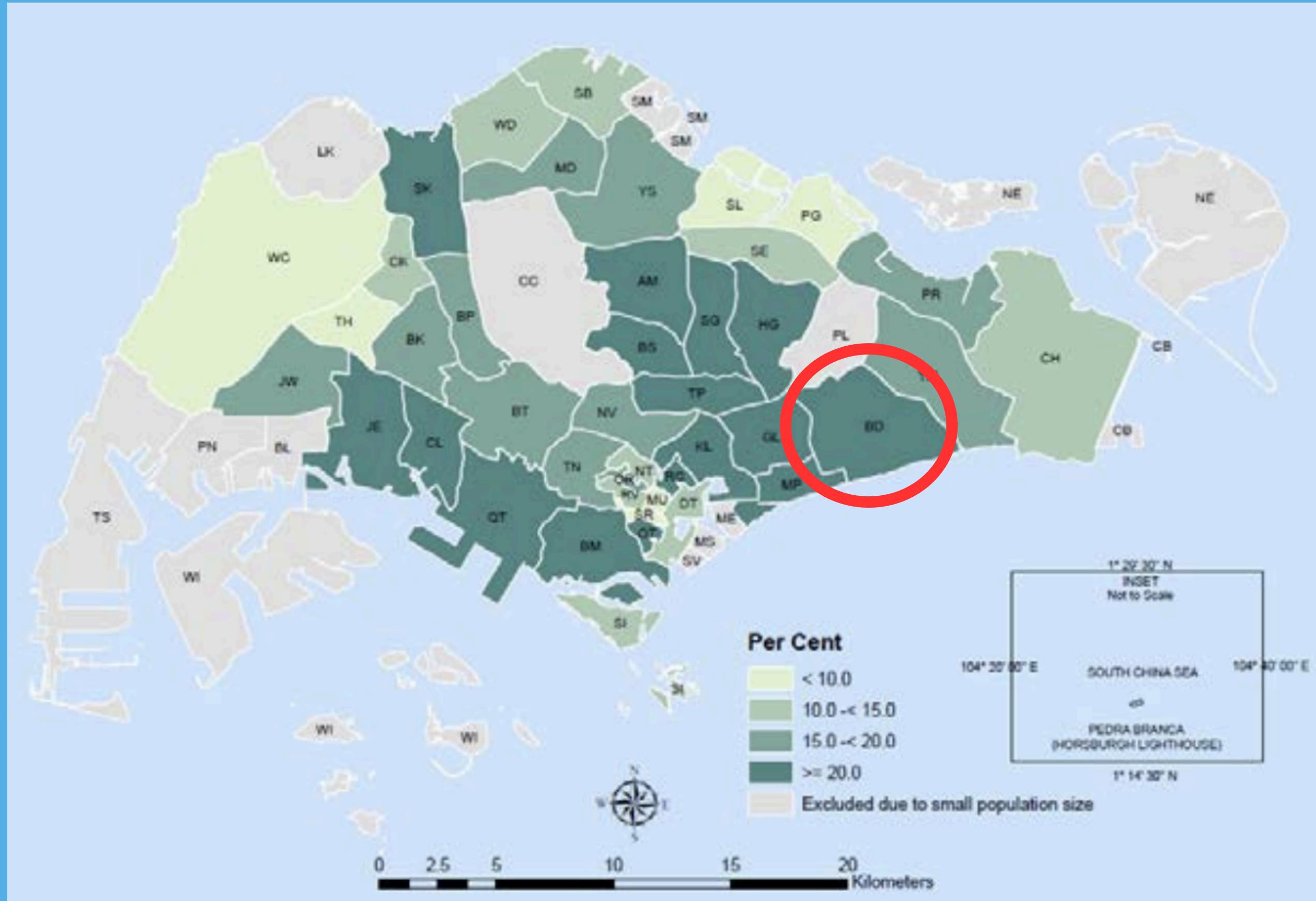
Densely populated areas:

**250,000
to
300,000**

- Jurong West
- Woodlands
- Seng Kang
- Tampines
- Bedok



Proportion of Resident Population Aged 65 Years and Over:



>20%
of population in
Bedok is Aging

1) Current Solution



MyTransport.SG



Singabus

User feedback:

Inaccurate timings



29 Nov
carissalyt

I don't use the app often at all because of how inaccurate these type of apps have been in my experience. This app is actually no different because literally the one time I decide to use it, the bus arrival times spontaneously change drastically. Today, 10 minutes was suddenly added to the bus arrival time, after I walke [more](#)

not accurate



24 Dec

linlins09

last bus timings is always wrong and i waited 30 minute for nonexistent busses as the app always states false timings. Give me back my time wasted

Unreliable app



15 Sep

SG Public User

Poor app design and not user friendly. Bus timing not always availability or missing. LTA cannot even build a proper app.

Overall problems:

*From the general public on these apps

- Inaccurate bus timings
- Long waiting times
- Unfriendly user interface

Problem Statement

How can **real-time transportation data** be optimized in **Bedok** to enhance the **accuracy, accessibility, and reliability** of **public transport**, particularly for the growing **elderly population**?

1) Preparation

APIs used:



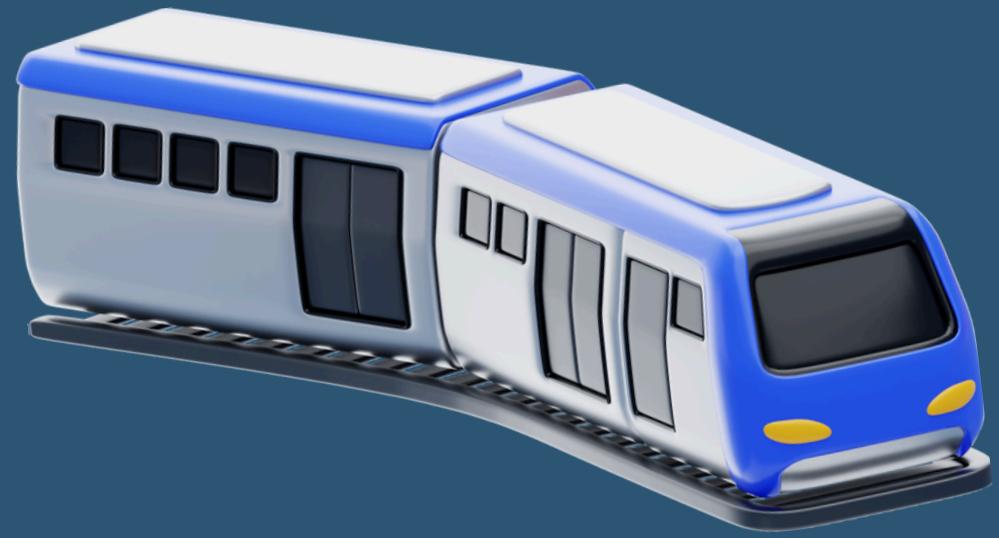
Bus Stops



Bus Arrival



Traffic Incidents



**Station Crowd Density
(real time)**

Land Transport Authority
We Keep Your World *Moving*

DATA MALL



Preparation for retrieval of Bus Arrival data



1) Retrieving the bus stop information



	BusStopCode	Description	Latitude	Longitude	RoadName
1	95159	Near SATS Flight Kitchen	1.341915	103.982251	Airport Blvd
2	82051	Aft Tg Katong Stn Exit 2	1.3008	103.897401	Tg Katong Rd Sth
3	84679	Opp Blk 156	1.318043	103.944766	Bedok Sth Ave 3
4	95041	CIAS Cargo Ters	1.372655	103.9957	Airport Cargo Rd
5	84031	Bedok Stn Exit B	1.324283	103.930036	New Upp Changi Rd
6	93111	Bef Elite Pk Ave	1.314513	103.922782	Siglap Rd
7	84701	Blk 151	1.336013	103.912142	Kaki Bt Ave 1
8	81031	Blk 134	1.315803	103.886049	Sims Ave
9	84431	Blk 3014	1.335136	103.950223	Bedok Nth Ave 4
10	81221	Opp Blk 1015	1.319763	103.890151	Geylang East Ctrl
11	96089	Opp Grace Independent Ch	1.336465	103.955947	Upp Changi Rd
12	84121	Aft Bedok Sth Ave 1	1.318437	103.931033	Bedok Sth Rd
13	81231	Opp Kh Plaza	1.313893	103.881896	Aljunied Rd
14	80239	Bef Blk 53	1.318464	103.880511	Sims Dr
15	81169	Opp Blk 56	1.310784	103.883132	Guillemard Rd
16	81141	Versailles Condo	1.312008	103.890358	Guillemard Rd

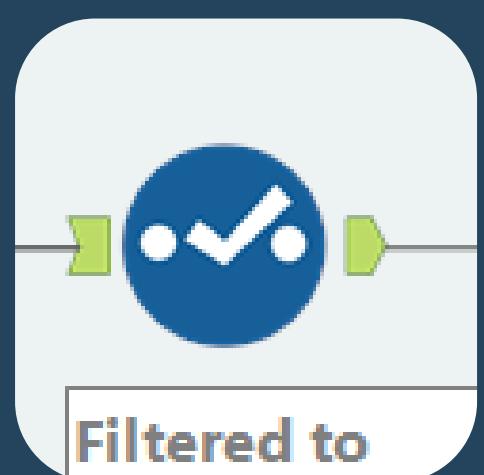
Bus Stop Locations



Preparation for retrieval of Bus Arrival data

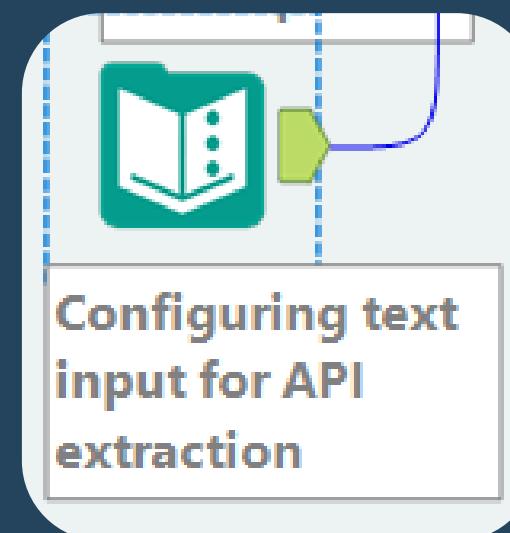


2) Filtering to only the **Bus Stop Code** column



	BusStopCode
1	95159
2	82051
3	84679
4	95041
5	84031
6	93111
7	84701
8	81031
9	84431
0	81221
1	96089
2	84121
3	81231
4	80239
5	81169

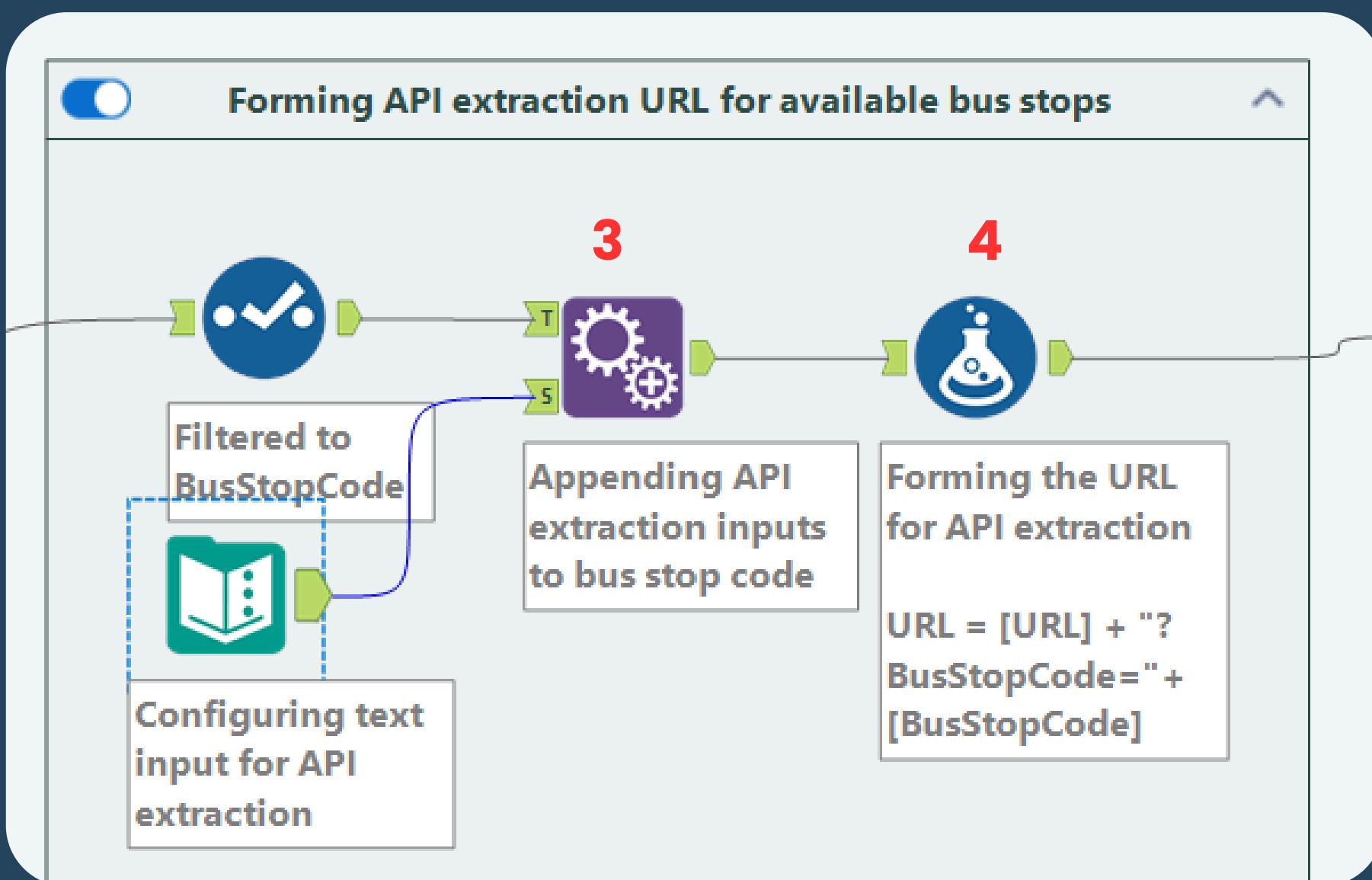
3) Configure text input with **partial url link** for bus arrival API endpoint



URL	AccountKey
1 https://datamall2.mytransport.sg/ltaodataservice...	P731Ap66S

Preparation for retrieval of Bus Arrival data

3) Combine Bus Stop Code and partial URL with Append Tool



4) Form the complete URL for extraction

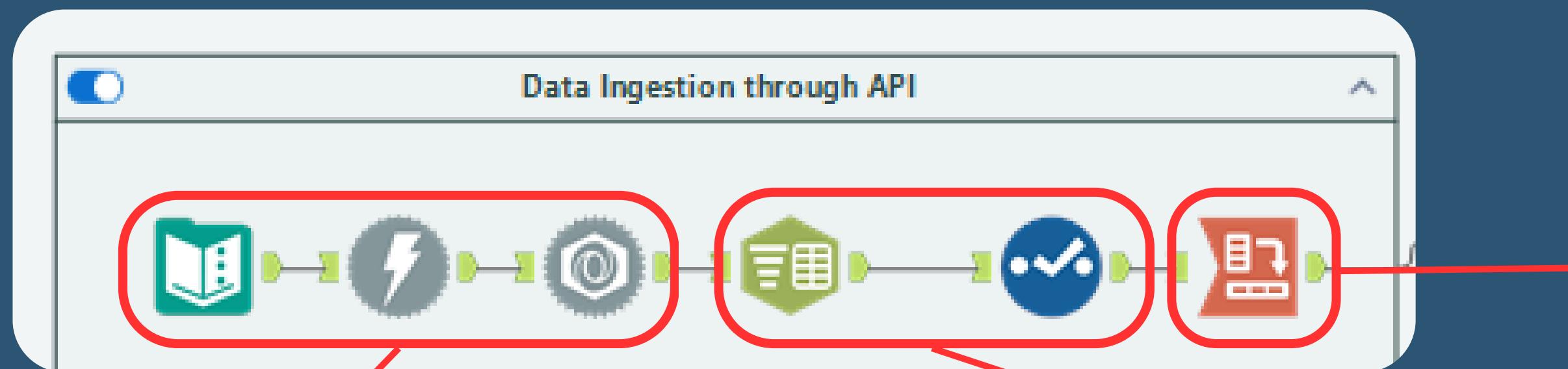
Formula used:

URL
[URL] + "?BusStopCode="+[BusStopCode]

URL
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=95159
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=82051
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=84679
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=95041
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=84031
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=93111
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=84701
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=81031
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=84431
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=81221
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=96089
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=84121
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=81231
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=80239
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=81169
https://datamall2.mytransport.sg/ltaodataservice/v3/BusArrivals?BusStopCode=81141

2) Making API Call for:

Alteryx workflow:



Downloads and Parses
data into key value pair

JSON_Name	JSON_ValueString
odata.metadata	http://datamall2.mytransport.sg/
value.0.BusStopCode	80091
value.0.RoadName	Sims Ave
value.0.Description	Bef Lor 23 Geylang
value.0.Latitude	1.31489775184137
value.0.Longitude	103.880951416408
value.1.BusStopCode	80101
value.1.RoadName	Lor 1 Geylang
value.1.Description	Kallang Str/Opp Blk 2C
value.1.Latitude	1.31251933500742

Splits JSON_Name to
smaller columns

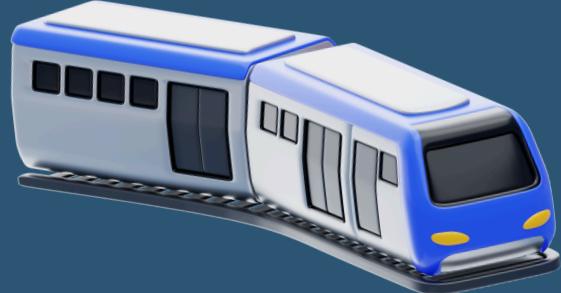
	JSON_Name1	JSON_Name2	JSON_Name3
value.0.	odata	metadata	[Null]
	value	0	Type
	value	0	Latitude
	value	0	Longitude
value.1.	value	0	Message
	value	1	Type
	value	1	Latitude
	value	1	Longitude



Bus Stops



Traffic
Incidents



Station
Crowd Density
(real time)

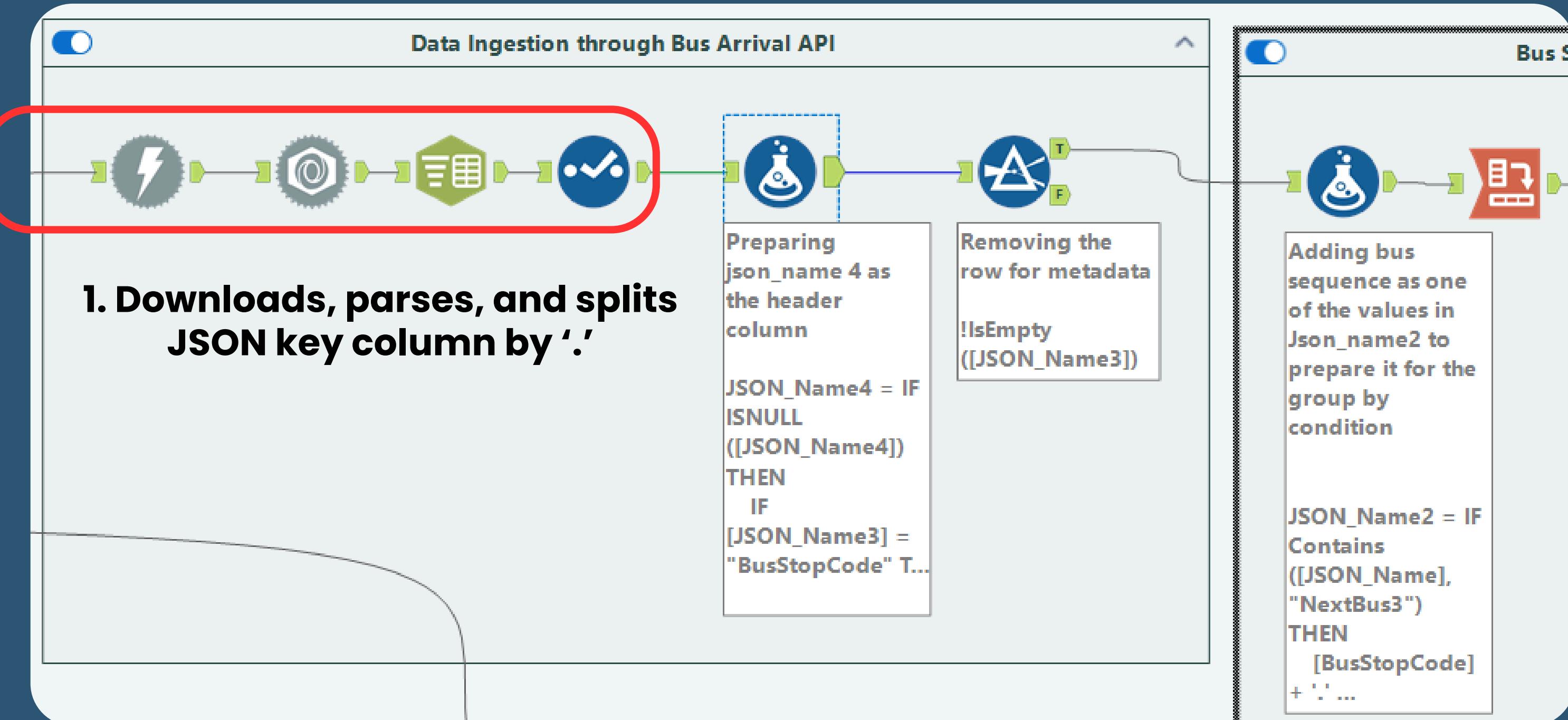
Pivots data into structured
tabular format

BusStopCode	Description	Latitude
95159	Near SATS Flight Kitchen	1.34191465289349
82051	Aft Tg Katong Stn Exit 2	1.30079999936599
84679	Opp Blk 156	1.31804305599346
95041	CIAS Cargo Ters	1.37265452174649
84031	Bedok Stn Exit B	1.32428292329305
93111	Bef Elite Pk Ave	1.31451280258696
84701	Blk 151	1.33601304352046
81031	Blk 134	1.31580283030533
84431	Blk 304	1.33513644378005
81221	Opp Blk 1015	1.31976331280568
96089	Opp Grace Independent Ch	1.33646502628787
84121	Aft Bedok Sth Ave 1	1.318437
81231	Opp Kh Plaza	1.31389305598354
80239	Bef Blk 53	1.31846350927886
81169	Opp Blk 56	1.31078371924435
81141	Versailles Condo	1.31200805596649
94089	Calvary Ably Of God Ch	1.31536333302999
80251	Aft Lor 14 Geylang	1.31009487384598
84049	Blk 27	1.32432719820227
80181	Kallang Squash Ctr	1.30508439498564

2) Making API Call for:



Bus Arrival API



Making API call for Bus Arrival API



2. Preparing values for Json_Name4 to be column headers



```
fx IF ISNULL([JSON_Name4]) THEN
    IF [JSON_Name] = "BusStopCode" THEN "BusStopCode"
    ELSEIF [JSON_Name3] = "ServiceNo" THEN "ServiceNo"
    ELSEIF [JSON_Name3] = "Operator" THEN "Operator"
    ELSE [JSON_Name4]
ENDIF
ELSE
    [JSON_Name4]
ENDIF
```

	JSON_Name2	JSON_Name3	JSON_Name4
metadata	[Null]	[Null]	[Null]
[Null]	[Null]	[Null]	[Null]
0	ServiceNo	[Null]	[Null]
0	Operator	[Null]	[Null]
0	NextBus	OriginCode	[Null]
0	NextBus	DestinationCode	[Null]
0	NextBus	EstimatedArrival	[Null]
0	NextBus	Monitored	[Null]
0	NextBus	Latitude	[Null]
0	NextBus	Longitude	[Null]



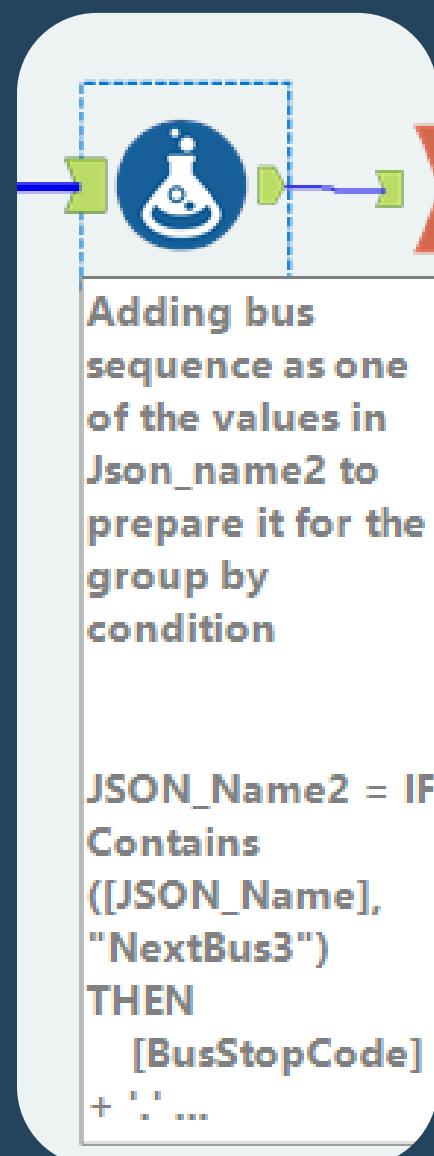
	JSON_Name2	JSON_Name3	JSON_Name4
	[Null]	[Null]	BusStopCode
	0	ServiceNo	ServiceNo
	0	Operator	Operator
	0	NextBus	OriginCode
	0	NextBus	DestinationCode
	0	NextBus	EstimatedArrival
	0	NextBus	Monitored
	0	NextBus	Latitude
	0	NextBus	Longitude

Making API call for Bus Arrival API



3. Configuring the Column to group data values by

1. Bus Stop Code
2. Bus Service
3. Bus Arrival Sequence



JSON_Name2

82051.

```
IF Contains([JSON_Name], "NextBus3") THEN  
    [BusStopCode] + '.' + [JSON_Name2] + '.3'  
ELSEIF Contains([JSON_Name], "NextBus2") THEN  
    [BusStopCode] + '.' + [JSON_Name2] + '.2'  
ELSEIF Contains([JSON_Name], "NextBus") THEN  
    [BusStopCode] + '.' + [JSON_Name2] + '.1'  
ELSE  
    [BusStopCode] + '.' + [JSON_Name2]  
ENDIF
```

JSON_Name2
82051.
82051.0
82051.0
82051.0.1
82051.0.1
82051.0.1
82051.0.1
82051.0.1
82051.0.1
82051.0.1
82051.0.1
82051.0.1
82051.0.1
82051.0.1
82051.0.1

Bus stop information

Bus service information

Information about the first bus that is arriving for the bus service

Making API call for Bus Arrival API

4. Pivot the data



Groupby : JSON_Name2
Header : JSON_Name4
Values : JSON_ValueString



Record	JSON_Name2	BusStopCode	DestinationCode	EstimatedArrival	Feature	Latitude	Load	Longitude	Monitored
1	92041.8								
2	82061.11.2		82009	2025-02-01T09:43:01+08:00	WAB	0.0	SEA	0.0	0
3	85109.3								
4	92091.2.1		84009	2025-02-01T09:31:12+08:00	WAB	1.3062215	SEA	103.8820486666666	1
5	84539.4								
6	92101.1.3		77009	2025-02-01T10:07:07+08:00	WAB	1.2753875	SEA	103.8338785	1
7	83109.13.2		46009	2025-02-01T09:39:22+08:00	WAB	1.3298585	SEA	103.847754	1
8	94049.5.3		11379	2025-02-01T09:54:36+08:00	WAB	0.0	SEA	0.0	0
9	94031.2								
10	84201.1.3		74009	2025-02-01T10:00:47+08:00	WAB	0.0	SEA	0.0	0
11	83319.0.3		94009	2025-02-01T09:55:57+08:00	WAB	1.3297915	SEA	103.8689245	1
12	84601.2								
13	94029.0								
14	80211.5.1		55509	2025-02-01T09:33:58+08:00	WAB	1.2778723333333333	SEA	103.848459	1
15	80251.	80251							
16	81199.0.2		16009	2025-02-01T09:47:08+08:00	WAB	1.314338833333332	SEA	103.935779	1
17	95151.0.3		63009	2025-02-01T10:03:15+08:00	WAB	0.0	SEA	0.0	0
18	83301.1.3		53009	2025-02-01T09:47:52+08:00	WAB	0.0	SEA	0.0	0
19	84041.2.3		99009	2025-02-01T10:00:13+08:00	WAB	1.3021701666666667	SEA	103.8571456666667	1
20	91061.2								
21	83131.1.3		84009	2025-02-01T09:57:17+08:00	WAB	1.2982918333333333	SEA	103.8538063333334	1

3) Data Display for

Bus Arrival API



Data after Cross-Tab:

	JSON_Name2	BusStopCode	DestinationCode	EstimatedArrival	Feature	Latitude
1	92041.8					
2	82061.11.2		82009	2025-02-01T09:43:01+08:00	WAB	0.0
3	85109.3					
4	92091.2.1		84009	2025-02-01T09:31:12+08:00	WAB	1.3062215
5	84539.4					
6	92101.1.3		77009	2025-02-01T10:07:07+08:00	WAB	1.2753875
7	83109.13.2		46009	2025-02-01T09:39:22+08:00	WAB	1.3298585
8	94049.5.3		11379	2025-02-01T09:54:36+08:00	WAB	0.0
9	94031.2					
10	84201.1.3		74009	2025-02-01T10:00:47+08:00	WAB	0.0
11	83319.0.3		94009	2025-02-01T09:55:57+08:00	WAB	1.3297915
12	84601.2					
13	94029.0					
14	80211.5.1		55509	2025-02-01T09:33:58+08:00	WAB	1.27787233
15	80251.	80251				
16	81199.0.2		16009	2025-02-01T09:47:08+08:00	WAB	1.31433883
17	95151.0.3		63009	2025-02-01T10:03:15+08:00	WAB	0.0
18	83301.1.3		53009	2025-02-01T09:47:52+08:00	WAB	0.0
19	84041.2.3		99009	2025-02-01T10:00:13+08:00	WAB	1.30217016
20	91061.2					
21	83131.1.3		84009	2025-02-01T09:57:17+08:00	WAB	1.29829183
22	84031.9.1		84009	2025-02-01T09:32:02+08:00	WAB	0.0
23	82069.17.3					
24	92061.11.2		97009	2025-02-01T09:46:34+08:00	WAB	1.30951483
25	84469.2					
26	95051.0.1		75009	2025-02-01T09:31:11+08:00	WAB	1.39033316
27	84271.0.1		77009	2025-02-01T09:22:01+08:00	WAB	1.22227566

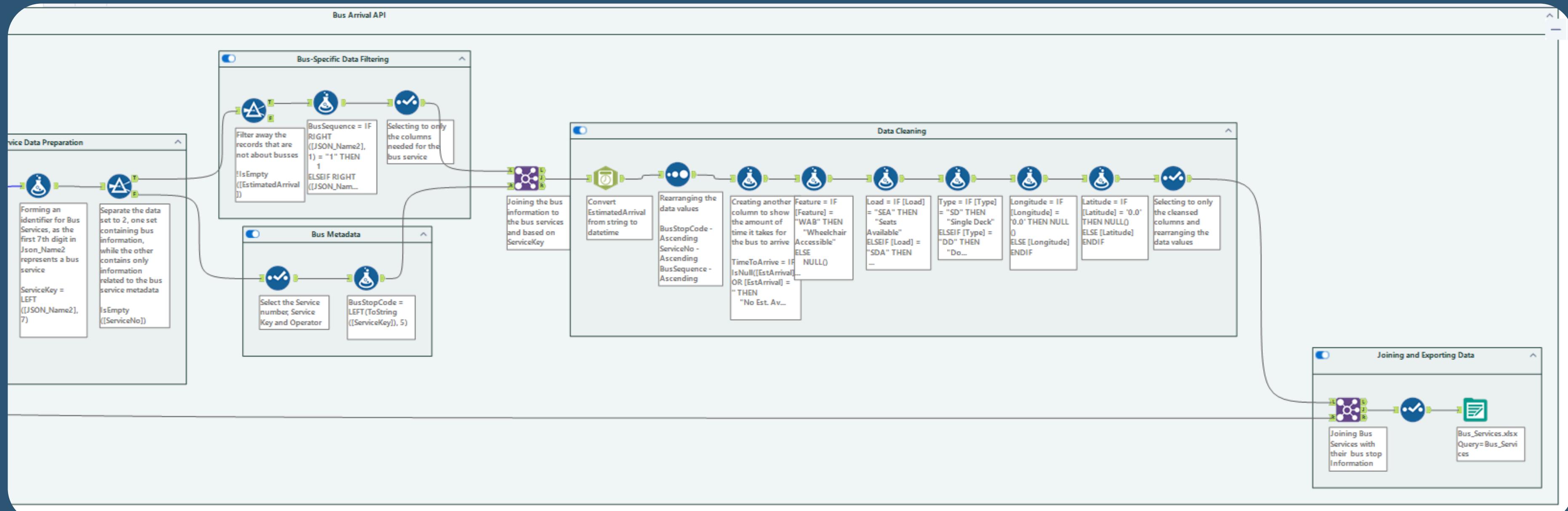
Observations:

- Some rows represent bus stops or bus service details
 - Eg. Service number
- Some rows represent bus specific data
 - Eg. Bus longitude and latitude
- Many values represented in short forms
- Date time values are represented in strings
 - In ISO 8601 format with timezone (+08:00)

3) Data Display for



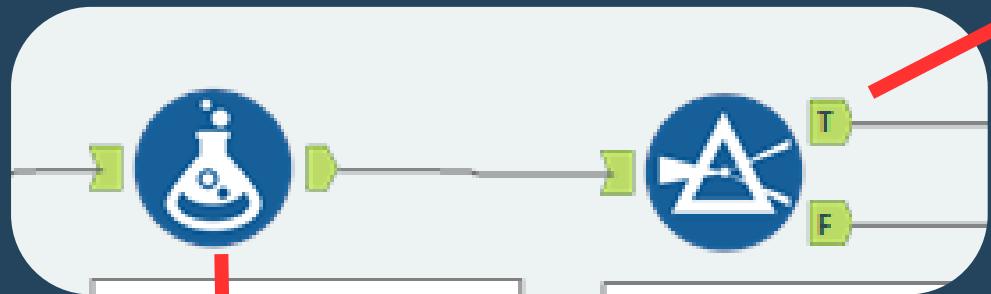
Bus Arrival API



Data Display for Bus Arrival API



1) Creating identifier for bus services



Extracting the first 7 digit in JSON_Name2 column



ServiceKey
92041.8
82061.1
85109.3
92091.2
84539.4
92101.1
83109.1
94049.5
94031.2
84201.1
83319.0
84601.2
94029.0
80211.5

As Service Key

2) Splitting data into 2 parts by :

Basic filter

ServiceNo Is empty

i. Bus Metadata (service details)

ServiceKey	Operator	OriginCode	ServiceNo	Type	VisitNumber
92041.8	SBST		48		
85109.3	SBST		38		
84539.4	SBST		87		
94031.2	SBST		137		

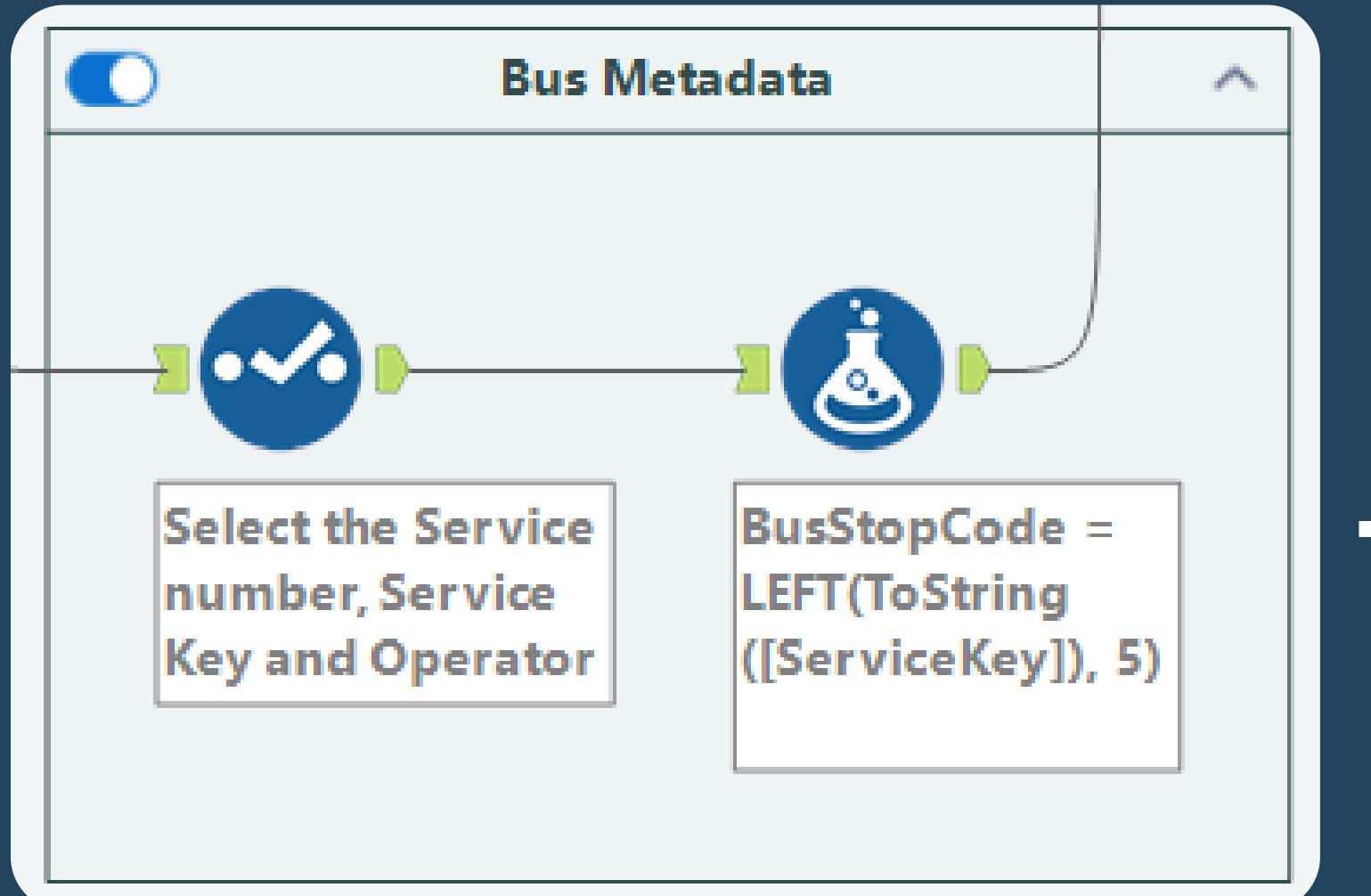
ii. Bus-Specific Data (live tracking details)

DestinationCode	EstimatedArrival	Feature	Latitude	Load	Longitude
82009	2025-02-01T09:43:01+08:00	WAB	0.0	SEA	0.0
84009	2025-02-01T09:31:12+08:00	WAB	1.3062215	SEA	103.88204866666666
77009	2025-02-01T10:07:07+08:00	WAB	1.2753875	SEA	103.8338785
46009	2025-02-01T09:39:22+08:00	WAB	1.3298585	SEA	103.847754
11379	2025-02-01T09:54:36+08:00	WAB	0.0	SEA	0.0
74009	2025-02-01T10:00:47+08:00	WAB	0.0	SEA	0.0

Data Display for Bus Arrival API



3) Bus Metadata Cleaning

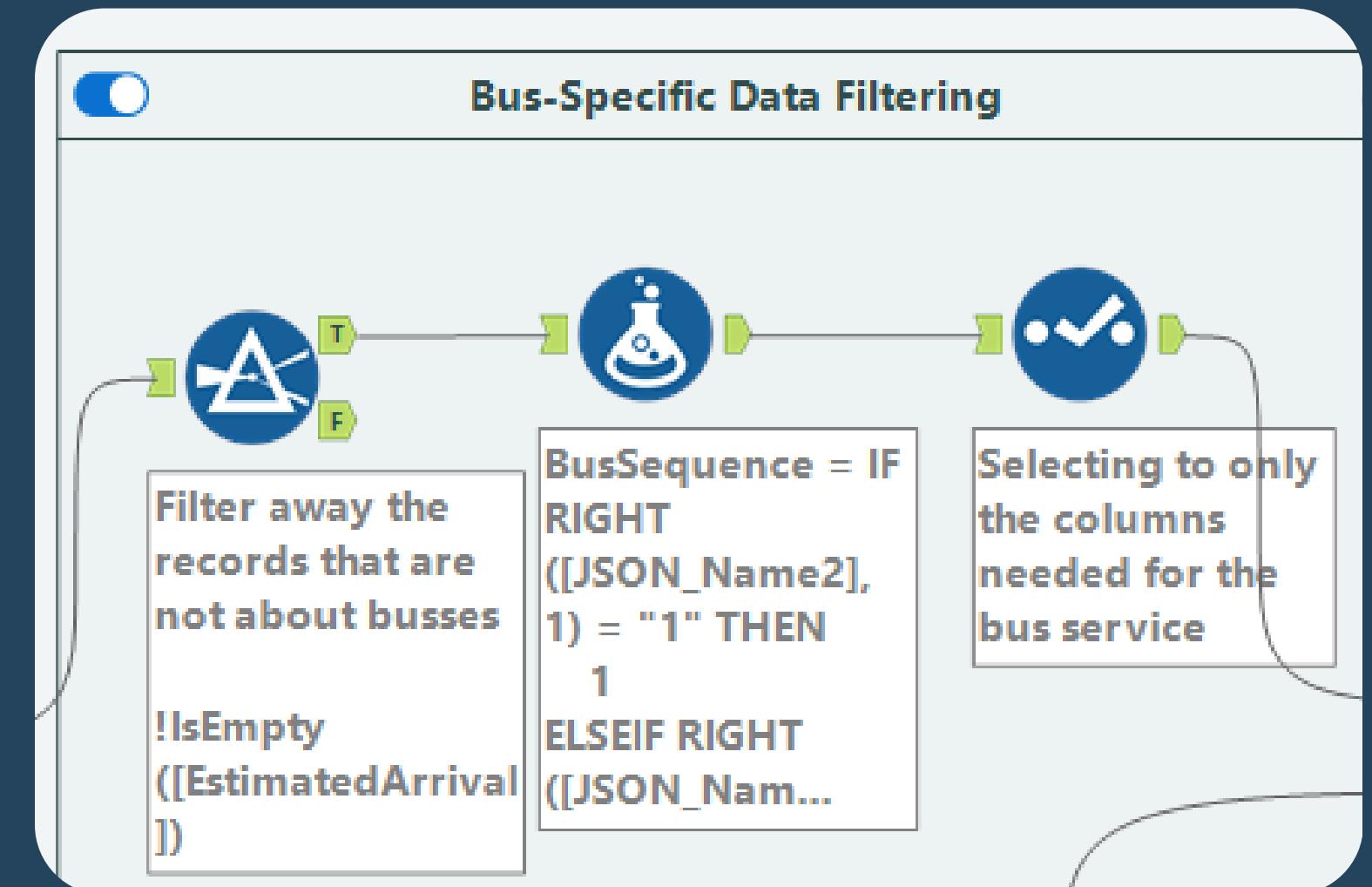


	ServiceNo	Operator	ServiceKey	BusStopCode
1	48	SBST	92041.8	92041
2	38	SBST	85109.3	85109
3	87	SBST	84539.4	84539
4	137	SBST	94031.2	94031
5	5	SBST	84601.2	84601
6	10	SBST	94029.0	94029
7	196	SBST	91061.2	91061
8	5	SBST	84469.2	84469
9	33	SBST	81171.6	81171
10	12	GAS	91061.0	91061
11	43	GAS	94029.4	94029
12	87	SBST	84309.3	84309
13	25	SBST	84069.3	84069
14	21	SBST	81031.3	81031
15	854	TTS	83101.1	83101
16	158	SBST	80091.3	80091
17	137	SBST	94089.2	94089
18	7	SBST	83062.8	83062
19	43	GAS	92241.6	92241
20	55	SBST	92071.1	92071

- Selecting the Necessary columns
- Extracting the bus stop number back from ServiceKey

Data Display for Bus Arrival API

4) Bus-Specific Data Cleaning



Output now contains only bus specific details:

Latitude	Load	Longitude	Monitored	OriginCode
0.0	SEA	0.0	0	82009
13062215	SEA	103.88204866666666	1	17009
12753875	SEA	103.8338785	1	10499
13298585	SEA	103.847754	1	46009
0.0	SEA	0.0	0	84431
0.0	SEA	0.0	0	84299
13297915	SEA	103.8689245	1	55509
1277872333333333	SEA	103.848459	1	03239
1314338833333332	SEA	103.935779	1	75009
0.0	SEA	0.0	0	63009
0.0	SEA	0.0	0	94009
13021701666666667	SEA	103.85714566666667	1	10499
1298291833333333	SEA	103.85380633333334	1	16009
0.0	SEA	0.0	0	84009
1309514833333333	SEA	103.92901183333333	1	97009
13903331666666667	SEA	103.99257533333333	1	75009
13303756666666666	SEA	103.92988816666667	1	84009
13073556666666666	SEA	103.87993633333333	1	80009
13202285	SEA	103.89165783333334	1	65009
13073465	SEA	103.87994533333334	1	66009
13172115	SDA	103.90329183333333	1	75009
1335435833333334	SEA	103.90995483333333	1	10009
13014551666666667	SEA	103.90109366666667	1	97009
13196676666666667	SDA	103.90510983333333	1	77009
13285425	SEA	103.94942516666667	1	84009
0.0	SEA	0.0	0	84431
0.0	SEA	0.0	0	84009
1356960833333334	SEA	103.94494566666667	1	75009
13271625	SDA	103.8802265	1	84009

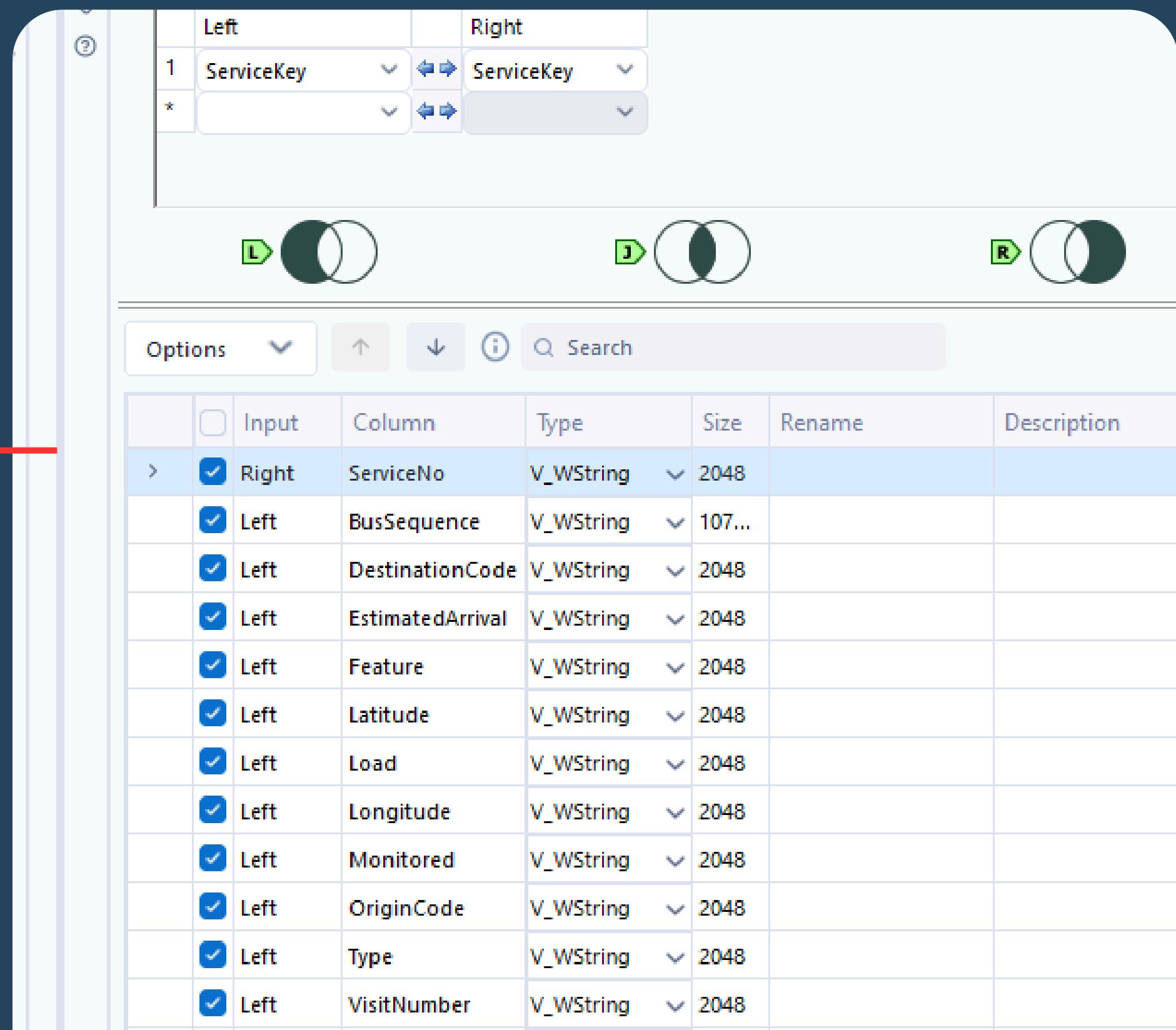
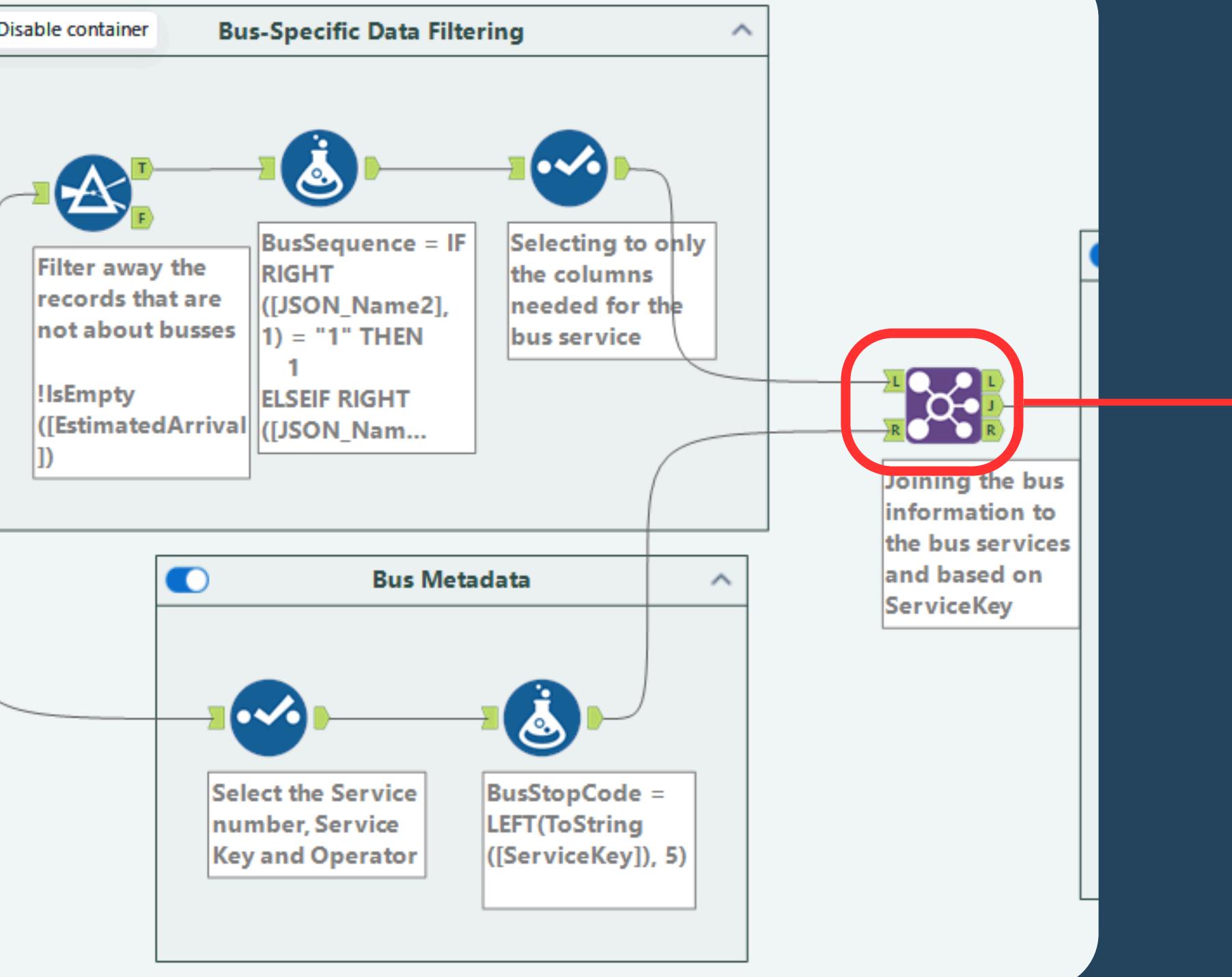
- **Filtered columns and records (**
Rows that contains only bus stop code)
- **Derived Bus arrival sequence from JSON_Name2**



Data Display for Bus Arrival API



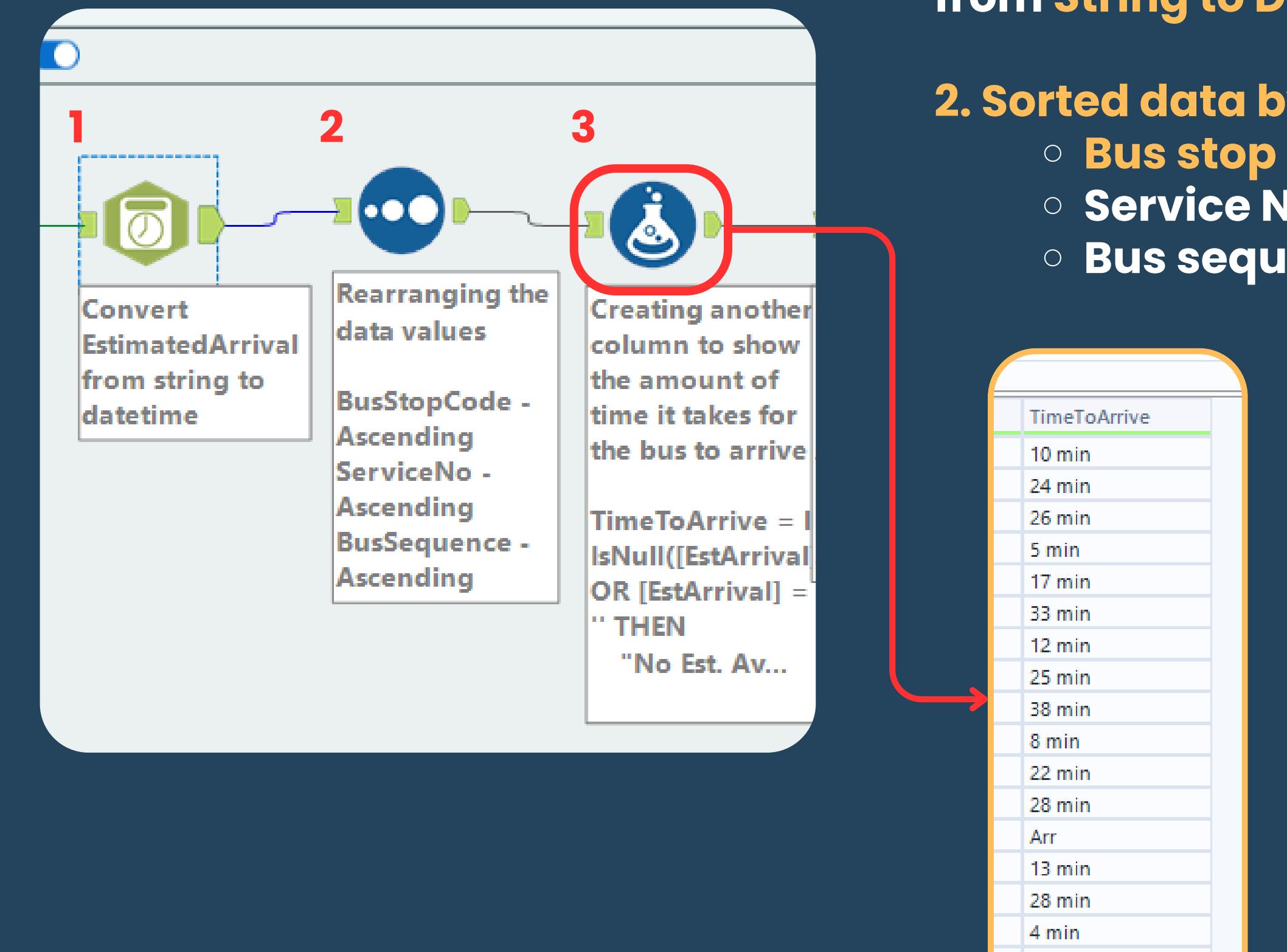
5) Joining the bus specific data with their corresponding bus metadata by ServiceKey



Data Display for Bus Arrival API



6) Data Cleaning for combined data



1. Converted Estimated Arrival from String to DateTime datatype

2. Sorted data by

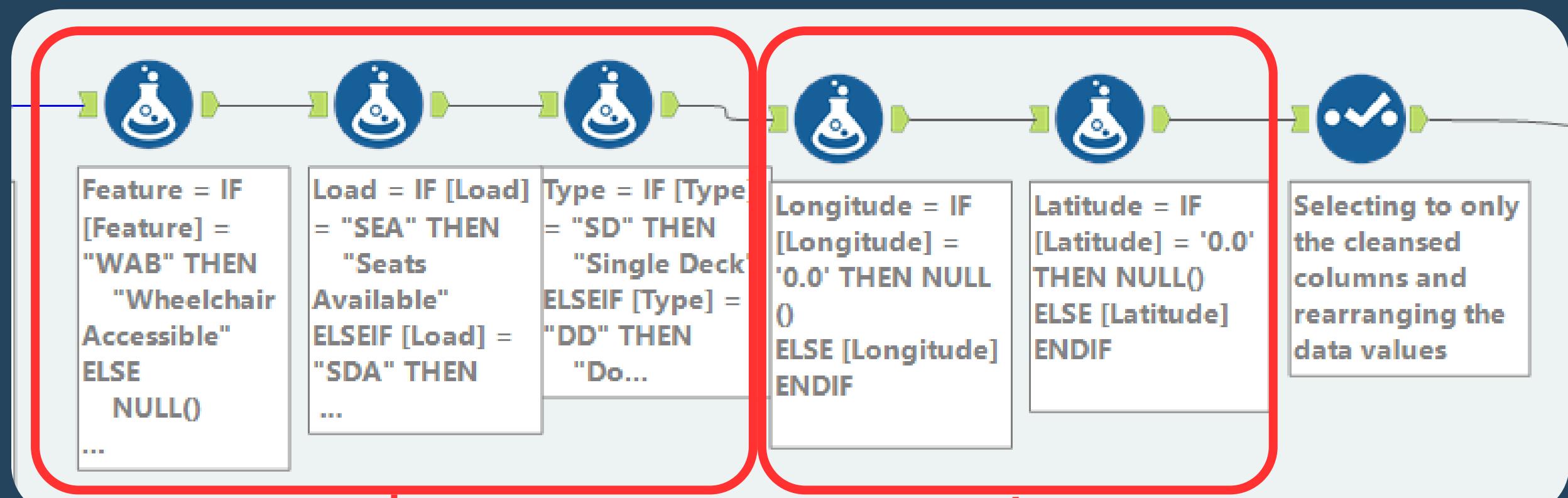
- Bus stop code**
- Service Number**
- Bus sequence**

3. Created a new column to show time taken for bus to arrive based on Estimated Arrival

Data Display for Bus Arrival API



6) Data Cleaning for combined data

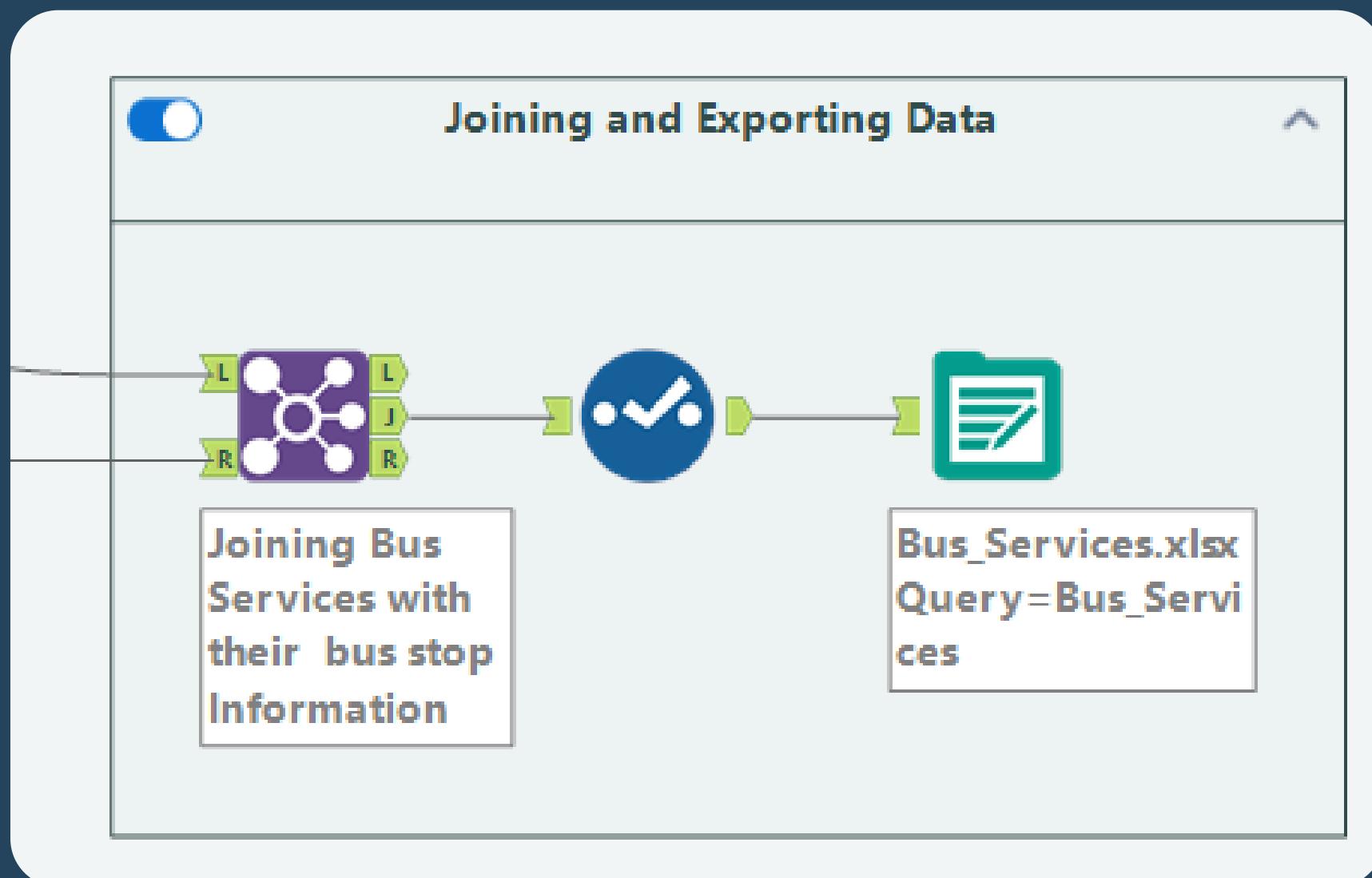


Converted Data
from short form to
full

Corrected Longitude
and Latitude from 0.0
to NULL

Data Display for Bus Arrival API

7) Combining Bus Arrival data with Bus Stops data



- Data is joined by **Bus stop code**
- Columns are rearraged with select tool

Final Attributes:

- Bus Stop Code
- ServiceNo
- BusStop_Description
- RoadName
- BusStop_Latitude
- BusStop_Longitude
- BusSequence
- DestinationCode
- TimeToArrive
- EstArrival time
- Feature
- Latitude
- Longitude
- Load
- Monitored
- OriginCode
- Type
- VisitNumber
- Operator

3) Data Display for

Traffic Incidents API



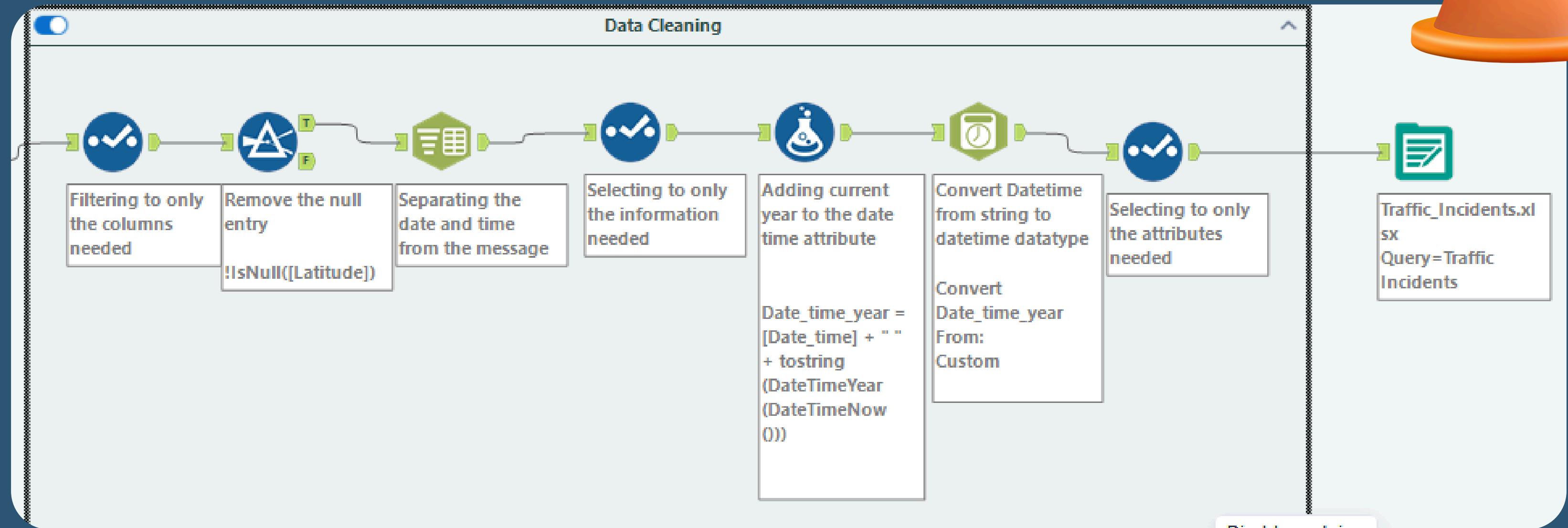
Data after Cross-Tab:

JSON_Name2	_Null_	Latitude	Longitude	Message	Type
1 2		1.28951063524124	103.851651966803	(1/2)16:03 Road Closure on St. Andrew's Road (to...	Road Block
2 4		1.41559435889076	103.771461880419	(1/2)14:45 Roadworks on BKE (towards PIE) after...	Roadwork
3 0		1.40095592354954	103.865811853345	(1/2)19:38 Vehicle breakdown on TPE (towards SL...	Vehicle breakdown
4 6		1.3917678183208	103.774666207164	(1/2)14:30 Roadworks on KJE (towards BKE) at BK...	Roadwork
5 1		1.32192688443142	103.87362036846	(1/2)19:14 Accident on PIE (towards Tuas) at Kall...	Accident
6 metadata	http://datamall2.mytransport.sg/taodataservice/...				
7 7		1.38908292484177	103.758711949473	(1/2)14:30 Roadworks on KJE (towards PIE) after...	Roadwork
8 5		1.34285725188122	103.746508511098	(1/2)14:31 Roadworks on PIE (towards Tuas) after...	Roadwork
9 3		1.34384178576306	103.860535723184	(1/2)15:42 Heavy Traffic on CTE (towards AYE) at...	Heavy Traffic

Observations:

- Row about API Metadata is not useful
- Date time of when the incident happened is combined under Message

Data Display for Traffic Incidents API



Transformations Done:

- Removed irrelevant rows and columns
- Derived Datetime variable from message column



Data Display for Traffic Incidents API



1) Separating the date and time from Message



Column to split

Message

Split to columns

Number of columns 2

Extra characters Leave extra in last column

Output root name Message

A configuration panel for a data splitting operation. It shows the 'Column to split' as 'Message', the 'Split to columns' option selected, and the 'Number of columns' set to 2. The 'Extra characters' setting is 'Leave extra in last column', and the 'Output root name' is 'Message'. A small icon bar on the left includes settings, history, and other options.

	Message1	Message2
	(1/2)16:03	Road Closure on St. Andrew's Road (towards Col...
	(1/2)14:45	Roadworks on BKE (towards PIE) after SLE Exit.
	(1/2)19:38	Vehicle breakdown on TPE (towards SLE) before...
	(1/2)14:30	Roadworks on KJE (towards BKE) at BKE(PIE) Exit.
	(1/2)19:14	Accident on PIE (towards Tuas) at Kallang Bahru...
	(1/2)14:30	Roadworks on KJE (towards PIE) after Woodlands...
	(1/2)14:31	Roadworks on PIE (towards Tuas) after Toh Guan...
	(1/2)15:42	Heavy Traffic on CTE (towards AYE) at PIE(Changi)...



Data Display for Traffic Incidents API



2) Deriving the date time attribute

The screenshot shows a data processing workflow. A 'Text' node (represented by a flask icon) is connected to a 'Format' node (represented by a clock icon). The 'Format' node has a configuration window open, titled 'HH:mm:ss Custom'. It asks 'Specify the format of the incoming string field' and contains the pattern '(d/M)hh:mm yyyy'. An example input '(2/1)03:04 2000' is shown next to its output '2000-01-02 03:04:00'.

Date_time_year (1/2)16:03 2025
fx [Date_time] + " " + toString(DateTimeYear(DateTimeNow()))

Adding current year to the date and time

	Date_time_year	DateTime
Col...	(1/2)16:03 2025	2025-02-01 16:03:00
l...	(1/2)14:45 2025	2025-02-01 14:45:00
re...	(1/2)19:38 2025	2025-02-01 19:38:00
xit.	(1/2)14:30 2025	2025-02-01 14:30:00
ru...	(1/2)19:14 2025	2025-02-01 19:14:00
nds...	(1/2)14:30 2025	2025-02-01 14:30:00
jan...	(1/2)14:31 2025	2025-02-01 14:31:00
ngi...	(1/2)15:42 2025	2025-02-01 15:42:00

Data Display for Traffic Incidents API



Final Traffic Incidents Data:

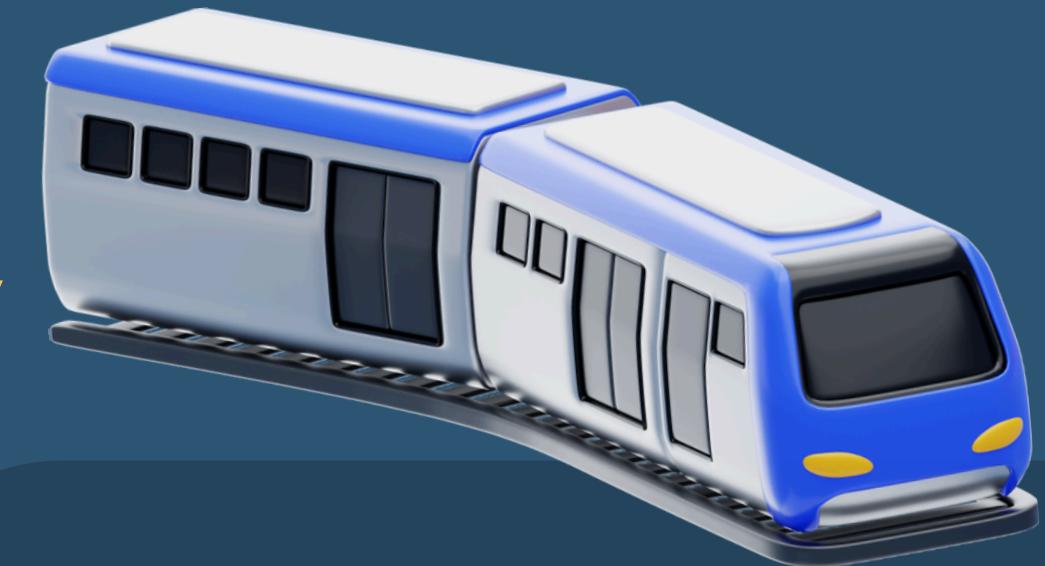
Field Information					
Record	Latitude	Longitude	Type	DateTime	Message
1	1.289511	103.851652	Road Block	2025-02-01 16:03:00	Road Closure on St. Andrew's Road (towards Col...
2	1.415594	103.771462	Roadwork	2025-02-01 14:45:00	Roadworks on BKE (towards PIE) after SLE Exit.
3	1.400956	103.865812	Vehicle breakdown	2025-02-01 19:38:00	Vehicle breakdown on TPE (towards SLE) before...
4	1.391768	103.774666	Roadwork	2025-02-01 14:30:00	Roadworks on KJE (towards BKE) at BKE(PIE) Exit.
5	1.321927	103.87362	Accident	2025-02-01 19:14:00	Accident on PIE (towards Tuas) at Kallang Bahru...
6	1.389083	103.758712	Roadwork	2025-02-01 14:30:00	Roadworks on KJE (towards PIE) after Woodlands...
7	1.342857	103.746509	Roadwork	2025-02-01 14:31:00	Roadworks on PIE (towards Tuas) after Toh Guan...
8	1.343842	103.860536	Heavy Traffic	2025-02-01 15:42:00	Heavy Traffic on CTE (towards AYE) at PIE(Changi)...

3) Data Display for

**Station Crowd Density
(real time)**

Data after Cross-Tab:

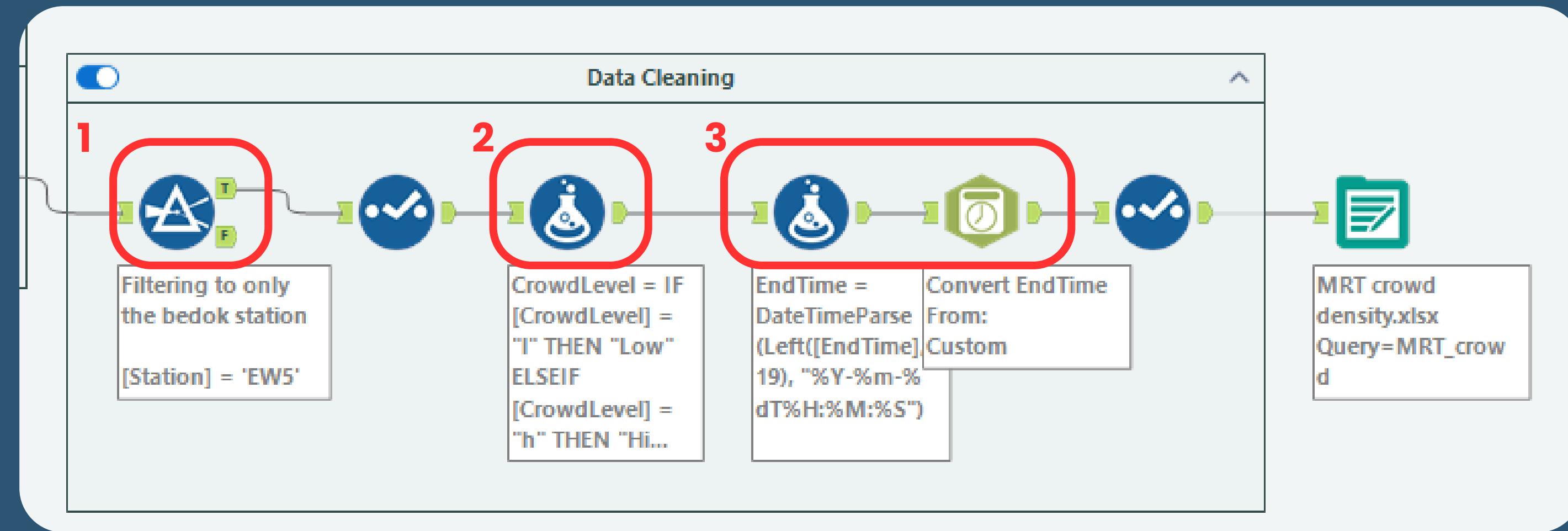
JSON_Name2	_Null_	CrowdLevel	EndTime	StartTime	Station
32			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW9
16		m	2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW24
4			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW13
14			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW22
31			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW8
17			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW25
0			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW1
6			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW15
1			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW10
metadata	http://datam...				
15			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW23
10			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW19
11			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW2
12			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW20
7			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW16
24			2025-02-01T19:20:00+08:00	2025-02-01T19:10:00+08:00	EW31
-					



Observations:

- Row about API Metadata is not useful
- Crowd level is in short form
- StartTime and EndTime attribute is in ISO 8601 format with timezone (+08:00)

Data Display for Station Crowd Density (real time)



Transformations Done:

1. Filtered row to only EW5 (Bedok Station)
2. Converted crowd level to full form
3. Converted date attributes to datetime datatype

Data Display for Station Crowd Density (real time)



Final Station Crowd Density Data:

Field Information

Record	Crowd Level	Station	Last Updated
1	Low	EWS	2025-02-01 19:20:00

4) Enhancement:

Features:

- **Language options**
- **Bus stop and Bus service input**
- **Bus arrival information display**
- **Interactive map, showing**
 - **Traffic incident locations**
 - **Bus stop location**
 - **Bus locations**

English 中文 (Chinese) தமிழ் (Tamil) Bahasa Melayu (Malay)

Bus Arrival Checker

Select a Bus Stop:

94069 - Opp Kew Green Condo

Available Bus Services:

10

Bus Arrival

4 min

Single Deck, Coming at 07:49 PM
Based on Live GPS

19 min

Single Deck, Coming at 08:04 PM
Based on Schedule

36 min

Double Deck, Coming at 08:21 PM
Based on Schedule

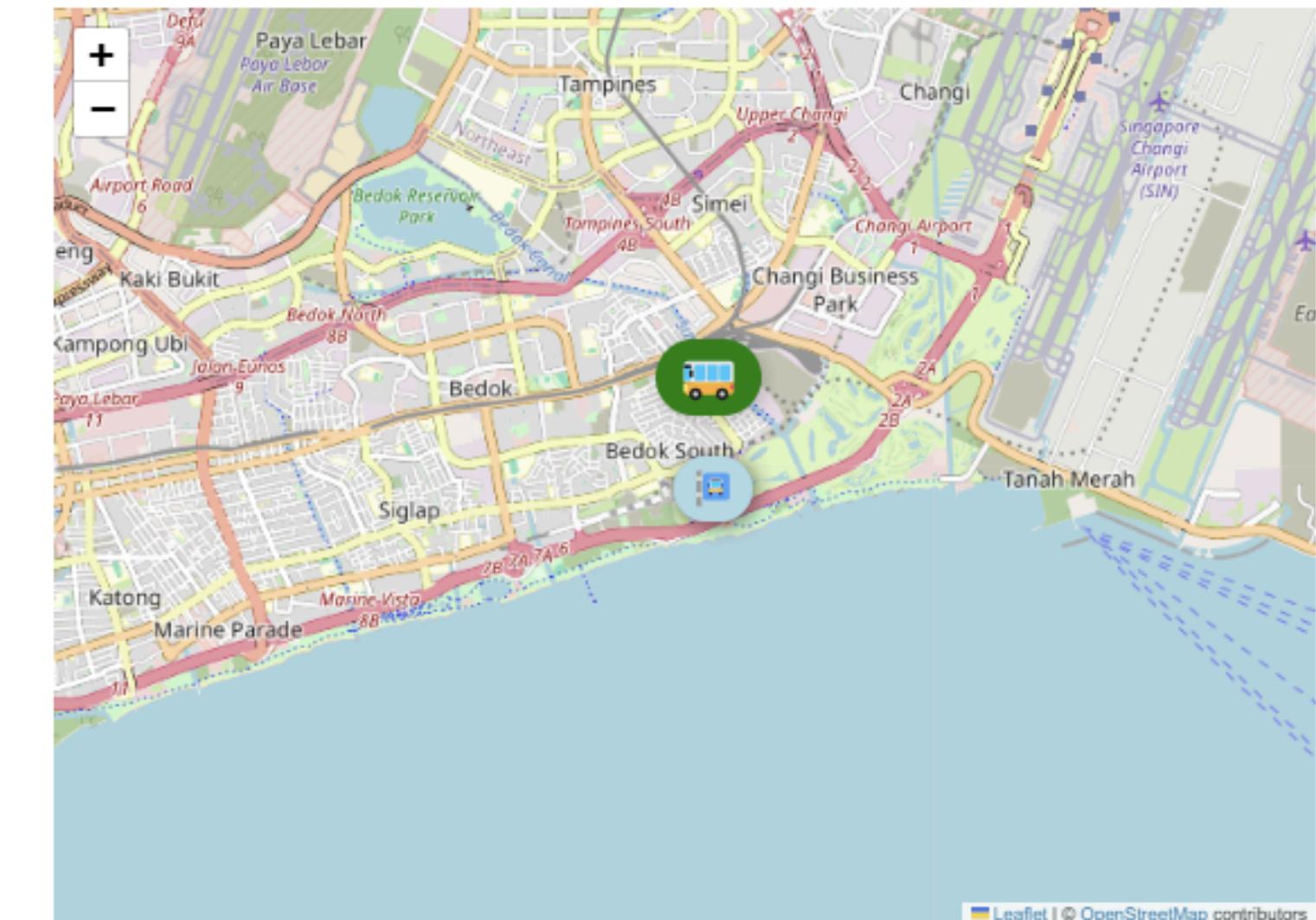
Bedok MRT Crowd Level:

Low

Station: EW5

Updated: 01 February 2025,
07:20 PM

Bus Locations & Traffic Condition



Leaflet | © OpenStreetMap contributors

Show full bus schedule data

Enhancement demo

[English](#) [中文 \(Chinese\)](#) [தமிழ் \(Tamil\)](#) [Bahasa Melayu \(Malay\)](#)

Bus Arrival Checker

Select a Bus Stop:

84489 - Opp Bedok Ind Pk C

Available Bus Services:

60

Bedok MRT Crowd Level:

Bedok MRT Crowd Level: **Low**

Station: EW5

Updated: 01 February 2025, 07:20 PM

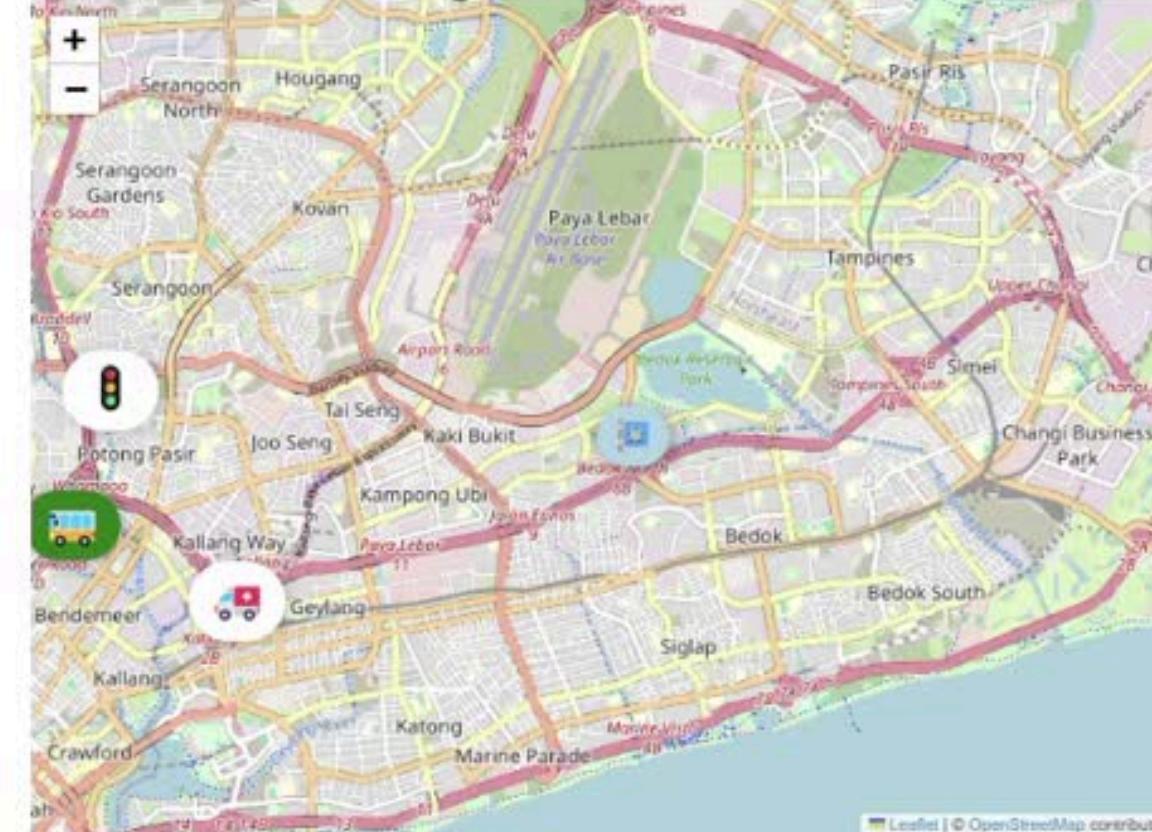
Bus Arrival

5 min Double Deck, Coming at 07:49 PM
Based on Live GPS

12 min Single Deck, Coming at 07:57 PM
Based on Live GPS

18 min Single Deck, Coming at 08:03 PM
Based on Schedule

Bus Locations & Traffic Condition



A map of Singapore showing bus locations and traffic conditions. A green circle marks the bus stop location, and a red circle marks the MRT station. The map includes labels for various neighborhoods like Serangoon, Hougang, Kovan, Paya Lebar, Tampines, and Bedok.

Show full bus schedule data

5) Summary

Objective Achieved:

- Improved accuracy, accessibility, and usability of bus arrival data for Bedok.
- Enhanced commuter experience, especially for elderly users.

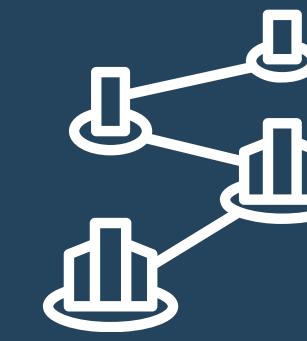
Key Features Implemented:



- Real-time bus arrival and incident tracking using LTA DataMall API.
- Streamlit web app with an interactive map and accessibility features.
- Multi-language support & wheelchair indicators for inclusivity.



Impact & Future Enhancements:



- More user-friendly journey planning for commuters.
- Potential to expand beyond Bedok and integrate predictive analytics.

6) Reflection

Key takeaways from Project

Hands-on Experience with Real-World Data Pipelines



- From retrieving real-time API data to cleaning, transforming, and displaying it.

Analytical and problem solving skills



- Understanding the key needs of community and develop data-driven solutions



Thank
You!