

INDEX

- 1. GENERAL INSTRUCTIONS**
- 2. ASSIGNMENT EXPECTATIONS**
- 3. PROBLEM STATEMENT**

1. GENERAL INSTRUCTIONS

- Please carefully read all the pages of the assignment.
- You can seek help from the Internet or anyone for learning and reference purpose.
- Do not copy the content as it is, use your genuineness and creativity.
- Any kind of plagiarism is highly prohibited.
- At any time if you need any clarifications, please do not hesitate to contact (using same email you received this assignment).
- Completed assignment must be committed to GitHub repo at regular intervals to show the incremental working history.
- Above assignment repo URL must be shared with public access through the email in which you received this assignment.
- Daily work progress needs to be send everyday by email in which you received this assignment. **THIS IS IMPORTANT**

It must follow the following format.

For example:

subject: [Assignment update-day-1][YOUR_NAME]

Hello Shan,

I have completed research on the following topics

- Please note that Daily work progress report sending is more important than the actual work progress.

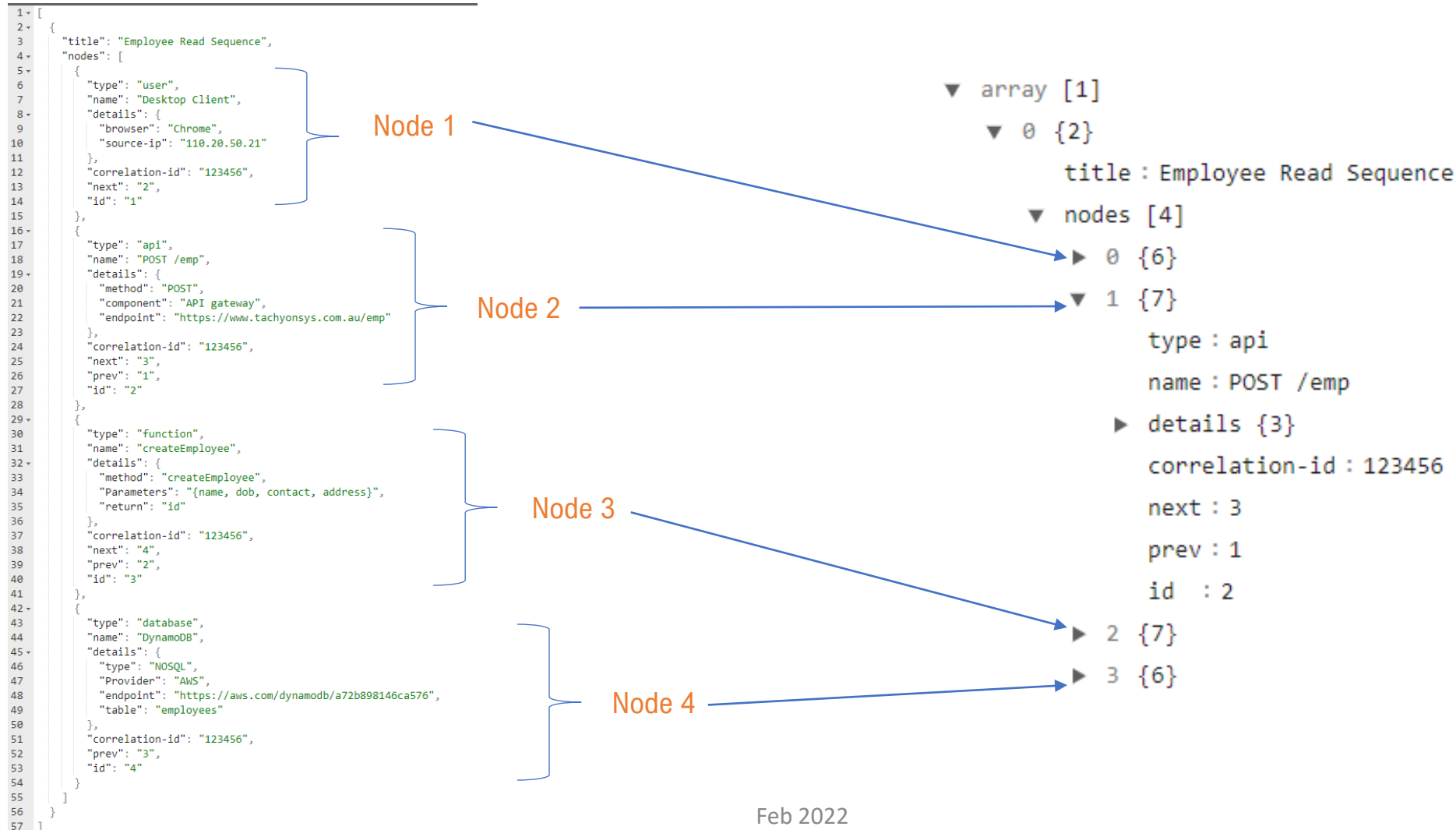
2. ASSIGNMENT EXPECTATIONS

- Should solved all four requirements.
- Should able to show a demo of the working solution locally by sharing a screen (discussion will be scheduled after assignment submission).
- Should be able to articulate their code logic and design clearly.
- Should know every bit of their source code.
- Should be able to debug the code by keeping breakpoints.
- Should be able to demonstrate source code versioning history in Github.





3. PROBLEM STATEMENT

- Develop web based front-end for the given graph data
 - Test-data-1.json
 - Test-data-2.json
- There are four requirement needs to be completed.
- The fifth requirement is optional, start work on this only if you have completed the all four requirements.
- Please refer subsequent pages for the requirements.
- **Technology preference**
 - Use any one of the following technology
 - React.js + HTML + CSS
 - Plain Vanilla JS + HTML + CSS
 - Angular + HTML + CSS
 - Vue.js + HTML + CSS
- For API Mocking, JSON-server or any other library that are comfortable with.

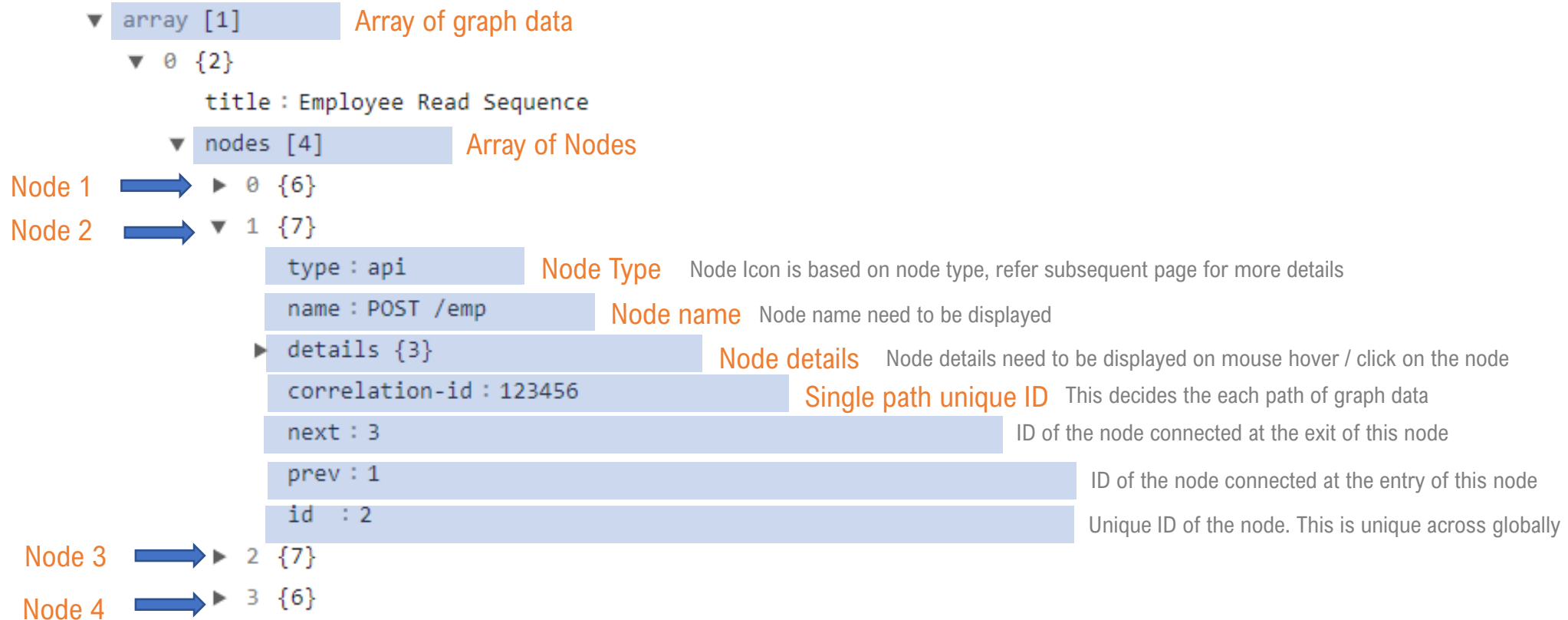
3. PROBLEM STATEMENT – JSON format description



3. PROBLEM STATEMENT – Node Type with Icons

Node UX	Node Type name	Node Icon Name from MaterialUI. Here given for the reference purpose only. You are free to use any UI library for icons. And also even Icons itself you are free to use anything but ensure icons are unique for each nodes.
	User	PersonOutline
	API	Api
	Database	Storage
	Function	Functions

3. PROBLEM STATEMENT – JSON format description

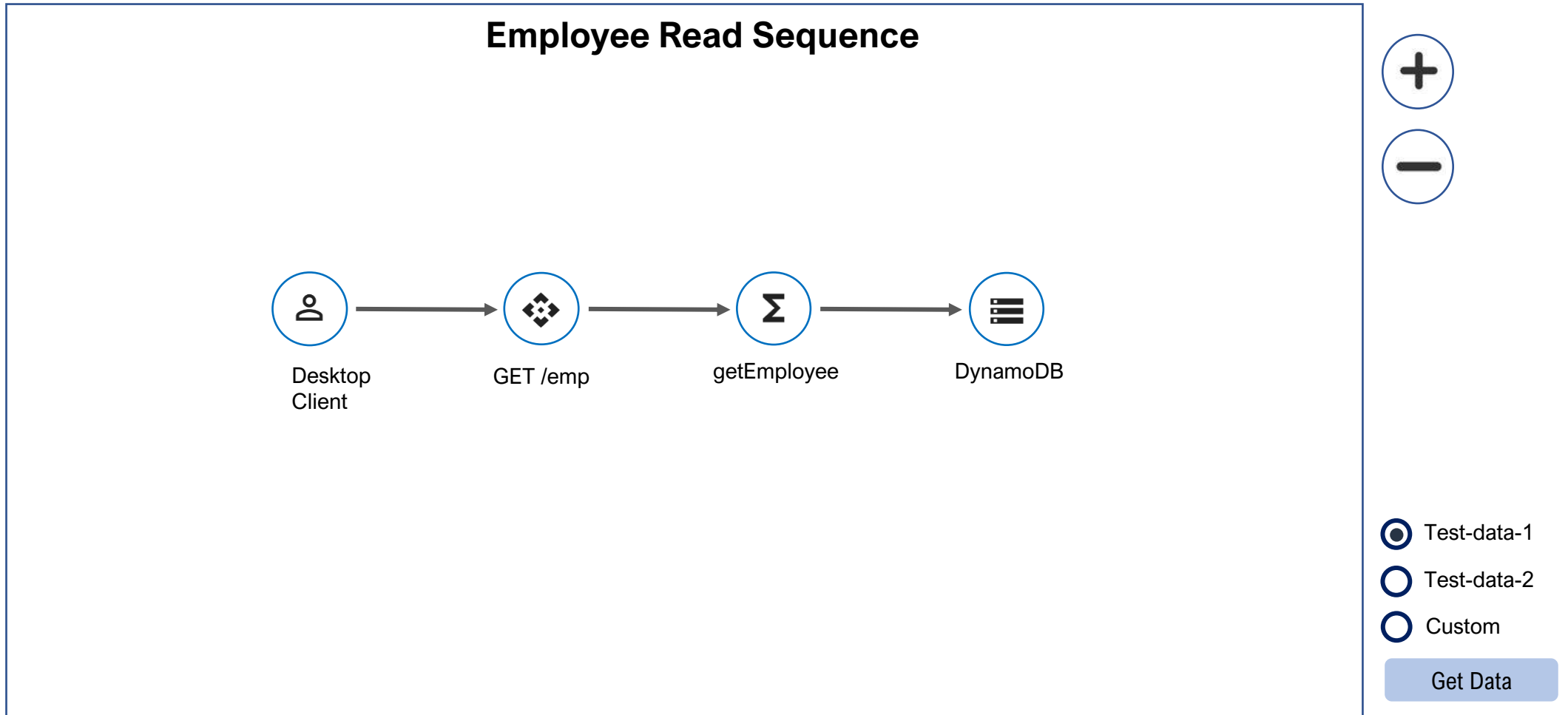


Output for the data - [test-data-1.json](#)

Load by selecting Test-data-1 option and get data using Mocked API server (eg. JSON server)

Requirement -1

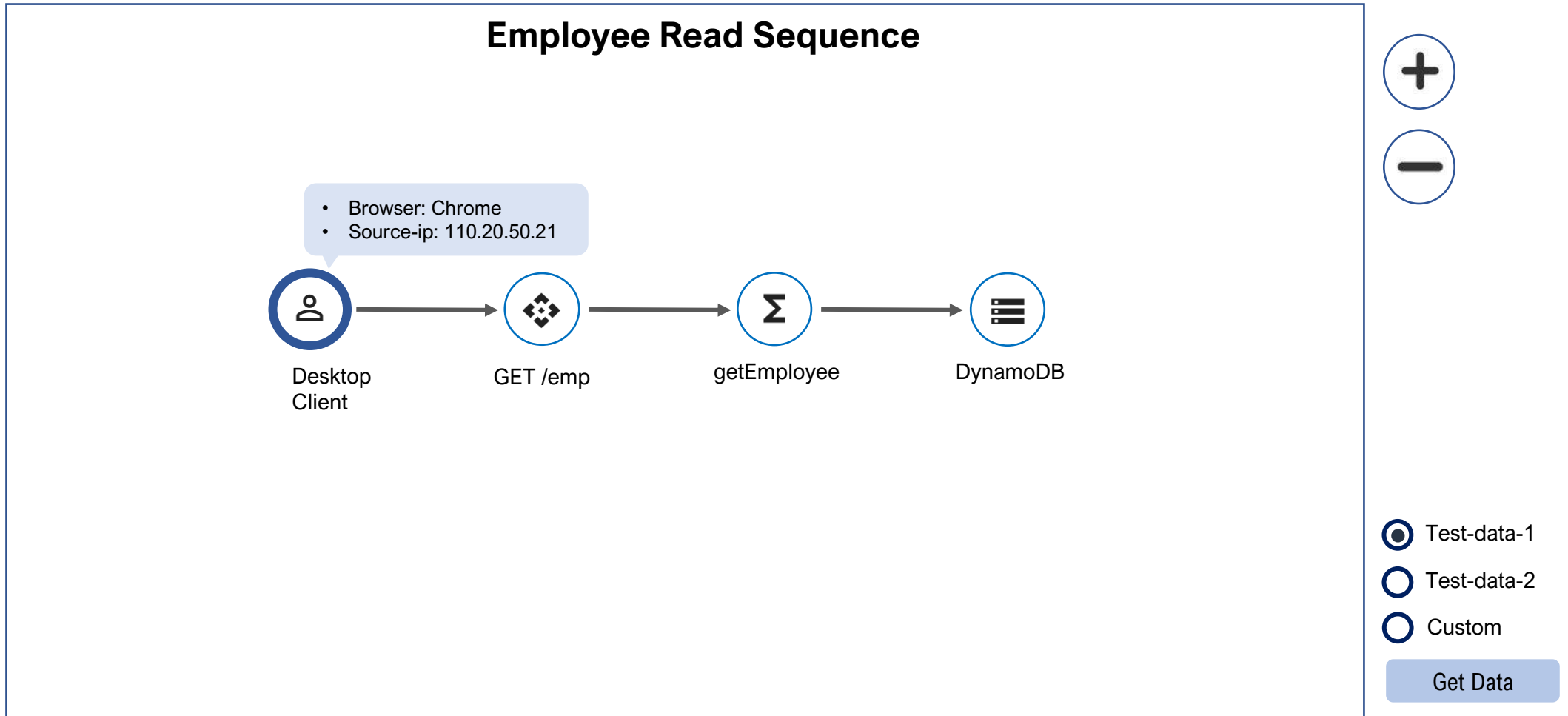
Must have Title, nodes with names, Icons & connected nodes arrow, Zoom In & Zoom out button as shown below



Output for the data - [test-data-1.json](#)

Requirement -2

On Mouse over on each node should display the details as shown below



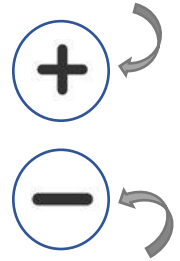
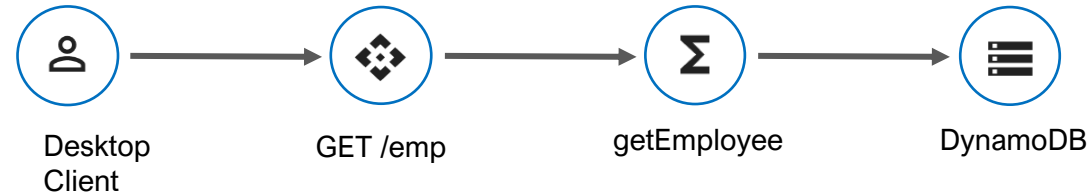
Output for the data - [test-data-1.json](#)

Requirement -3

On Click Zoom In /Zoom out must work for 5X levels

must support 5
level of Zoom In

Employee Read Sequence



must support 5
level of Zoom Out

- ☒ Test-data-1
- ☐ Test-data-2
- ☐ Custom

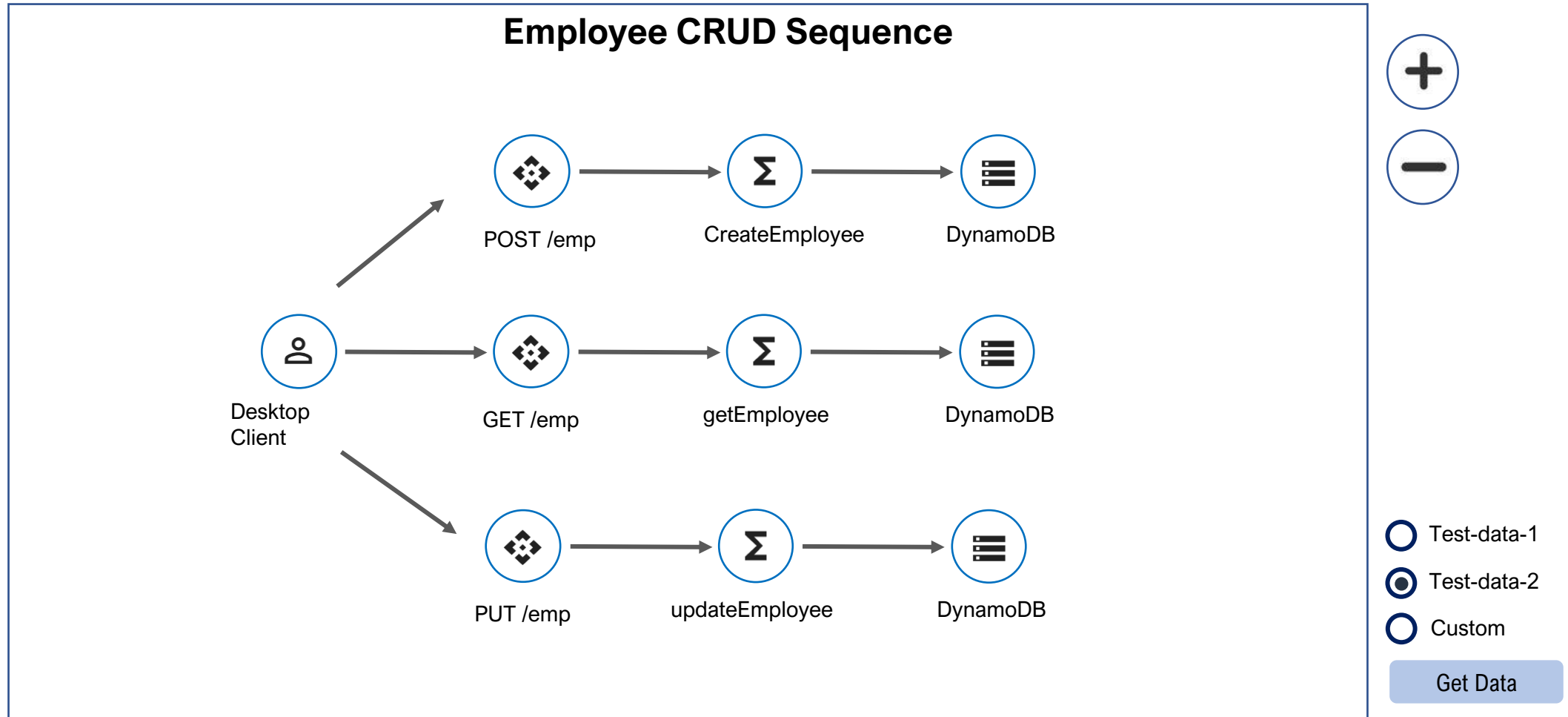
Get Data

Output for the data - [test-data-2.json](#)

Load by selecting Test-data-2 option and get data using Mocked API server (eg. JSON server)

Requirement -4

Must display multiple paths as shown below and all three requirements mentioned previously



Apply your logical reasoning and creativity in creating new large data with multiple paths, more node types. Load by selecting custom option.

Requirement -5

Create your own generic data with more nodes (say at least 25) with multiple path and ensure all requirements are working

This is optional requirement, work on this one only if you have successfully completed all the four requirements.



- ☐ Test-data-1
- ☐ Test-data-2
- ☒ Custom

Get Data

