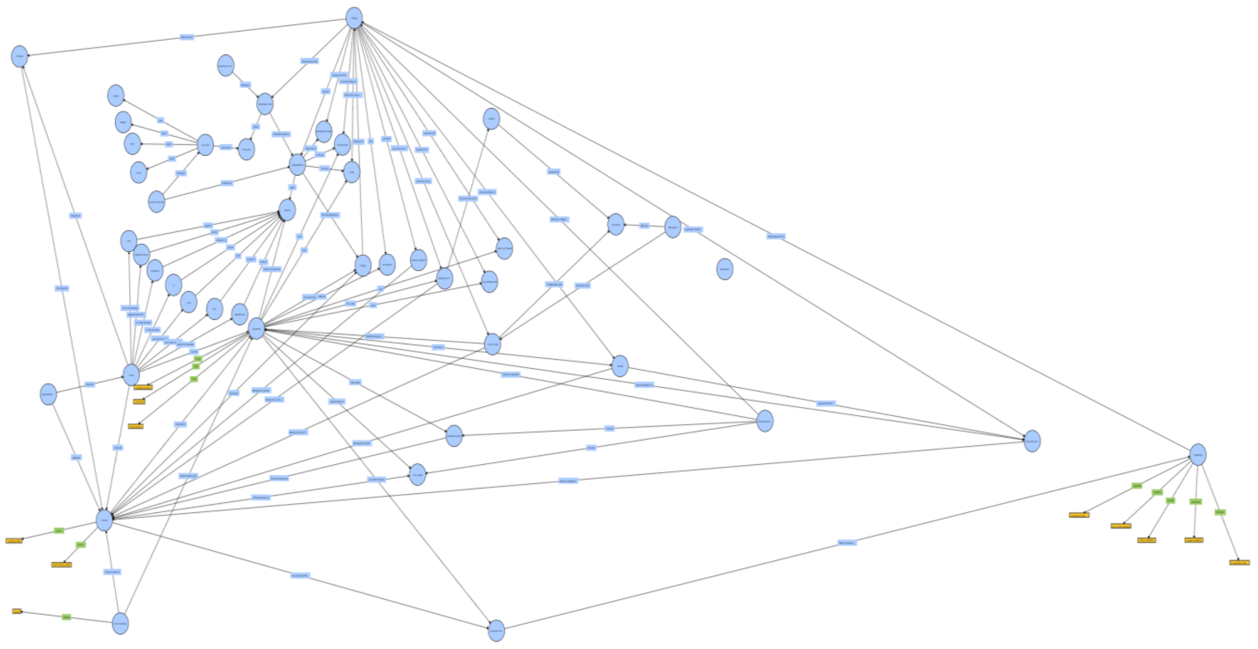
<u>(TR-102) MASTERING THE SEMANTIC WEB</u> –

<u>Training Day 9 Report</u> :

25 June 2024

<u>Task:</u>

Creating Architectural-Level RDFs using VOWL



[View SVG File Here]() By:

Inderjit kaur URN:2303086 CRN:2221159

Introduction to APIs and Postman :

- APIs (Application Programming Interfaces) are the building blocks that allow different software applications to communicate and exchange data with each other. APIs define the rules and protocols for this interaction, enabling the flow of information between systems.

- Postman is a popular API development and testing tool that simplifies the process of creating, executing, and automating API tests.

Key Features of Postman:

- Support for various HTTP request methods (GET, POST, PUT, PATCH, DELETE, etc.) and flexible request body formats.
- Simplified authentication management, handling methods like API keys, OAuth, and Basic Auth.
- Organized API testing through collections, which allow developers to categorize and manage API requests.
- Efficient API documentation generation directly from requests and collections.

Task:

To fetch data from a JSON file and display it in an HTML file using an API (like a mock server) tested with Postman.

Steps:

- Create a Mock Server in Postman
- Create a JSON File
- Test the API with Postman
- Fetch and Display Data in an HTML File

1. Create a Mock Server in Postman

- Open Postman.
- Click on "Mock Server" from the left panel.
- Click on "Create a mock server".
- Configure the mock server.
- Add a name for the mock server.
- Set the request type to GET.
- Set the endpoint to any name in my case it is '/data'.
- Add the response body, which should be the JSON data we want to serve.
- Here is an example of the JSON response body:
- {
    - "name": "John Doe",
    - "age": 30,
    - "city": "New York"
- }
- Click "Create Mock Server".
- Copy the mock server URL provided by Postman.

2. Create the HTML File

Create an Html file with the following content to extract some content :

```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <title>Display JSON Data</title>

</head>
<body>
    <h1>Data from API</h1>
    <div id="name"></div>
    <div id="age"></div>

    <script>
        // Replace with your Postman mock server URL
        const apiUrl = 'https://de2107eb-ba0a-4921-b263-
270c8347ff88.mock.pstmn.io/data';

        // Fetch data from the API endpoint
        fetch(apiUrl)
            .then(response => {
                if (!response.ok) {
                    throw new Error('Network response was not
ok ' + response.statusText);
                }
                return response.json();
            })
```

```
          .then(data => {
            // Display only the name and age in the HTML
            document.getElementById('name').innerText =
data.name;
            document.getElementById('age').innerText =
data.age;

          })
          .catch(error => console.error('Error fetching
data:', error));
    </script>
</body>
</html>
```

3. Test the API with Postman

- Open Postman.
- Create a new GET request to your mock server URL (e.g., https://your-mock-server-url/data).
- Send the request to ensure that the mock server is returning the correct JSON data.

4. Run the HTML File

- Make sure the html file is saved and located in an accessible directory.
- Open the html file in a web browser.
- You should see the JSON data displayed on the page.

To View Output Click Here