📘 API Testing - JSON Schema Validation (Manual + Automation)

This document provides a detailed explanation of the 'JSON Schema Validation' topic as taught in your Udemy course. It includes both manual validation and automation using Postman.

# 1. What is JSON Schema?

JSON Schema is a format for describing the structure of JSON data.  
It allows you to define the expected fields, data types, required properties, and nested structures in a JSON response.  
It's like a contract that defines what a valid JSON should look like.

# 2. Manual JSON Validation

In manual validation, you make a GET API request and visually inspect the JSON response to ensure that:  
- All required fields are present.  
- Data types are correct (e.g., number, string, array).  
- No unexpected fields are present.  
This is done manually by observing the response and comparing it with expectations or documentation.

# 3. Automated JSON Schema Validation

In automation, you use a predefined schema to validate the structure of the response automatically. This is done in Postman using the Tests tab.  
Steps:  
1. Make a GET request and copy the response JSON.  
2. Paste it into https://jsonschema.net to generate a schema.  
3. Copy the schema and paste it in the Tests tab in Postman.  
4. Add the assertion code:

pm.test("Validate schema", function () {  
 var schema = { ...your schema here... };  
 pm.response.to.have.jsonSchema(schema);  
});

# 4. Why Schema Validation is Important?

- Ensures the API response structure hasn't changed.  
- Automatically catches missing or incorrect fields.  
- Saves time and increases accuracy in testing.  
- Useful in CI/CD pipelines.

# 5. Manual vs Automation - Key Differences

| Feature | Manual Validation | Automation (Schema Validation)   
|-----------------------|-------------------------------|------------------------------------|  
| Effort | High | Low (after setup)   
| Accuracy | Human-dependent | Very high   
| Scalability | Low | High   
| Speed | Slow | Fast   
| Tool Required | None | Postman + Schema generator website

# 6. Tools Used

- Postman (API Testing Tool)  
- JSONSchema.net (to generate JSON schema from sample)  
- Optional: VS Code or other JSON viewers

<https://jsonschema.dev>

# 7. Summary - What We Learned

- JSON schema defines the structure of valid JSON data.  
- Manual schema validation is done by inspecting JSON response manually.  
- Automation schema validation is done using Postman and JSONSchema tools.  
- Automated testing increases accuracy and efficiency.

# 8. Home Task

1. Make a GET request to any public API (e.g., https://reqres.in/api/users/2).  
2. Copy the response JSON and generate a schema using JSONSchema.net.  
3. Paste that schema in Postman → Tests tab.  
4. Run the GET request and observe the test results.