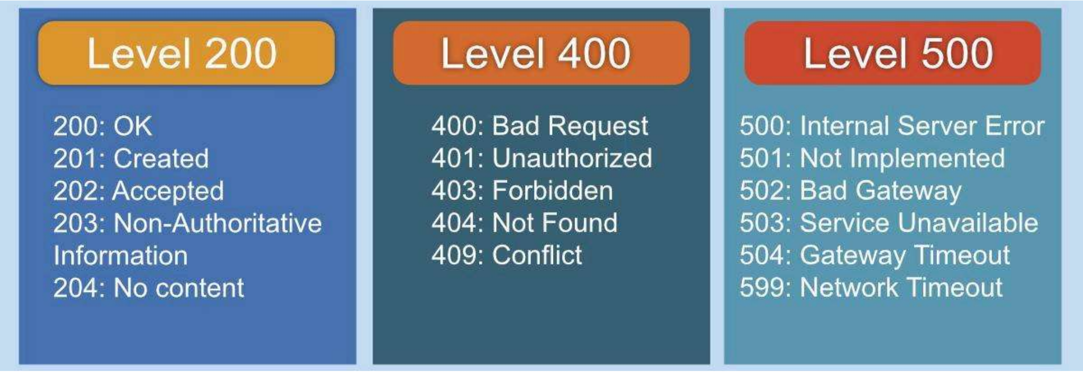
3. Response Validations(Assertions)

When testing APIs, we must know the type and structure of the response. To do this, we should run the API once to observe the response. Without this, we cannot set proper validation points.

Types of Validations or Assertions on Response

1. **Status Code:** Indicates the result of the API request. It helps identify the outcome of the request.



Level 200	Level 400	Level 500
200: OK	400: Bad Request	500: Internal Server Error
201: Created	401: Unauthorized	501: Not Implemented
202: Accepted	403: Forbidden	502: Bad Gateway
203: Non-Authoritative Information	404: Not Found	503: Service Unavailable
204: No content	409: Conflict	504: Gateway Timeout
		599: Network Timeout

2. **Headers:** Provide additional metadata in the form key-value pairs about the response, such as the type of data returned (e.g., JSON), server details, caching rules, or security policies.

3. **Cookies:** Small pieces of data sent by the server to the client, often used to maintain sessions, track user activity, or store user preferences. Cookies are not found when running the API locally; they are only visible when accessed through the internet.

4. **Response Time:** Measures how long the server takes to process a request and send back a response. It is critical for performance and user experience.

   a. Response time depends on factors like server performance, network latency, payload size, API logic, database, traffic, caching, third-party services, distance, and error handling.

   b. Therefore, we cannot expect the same response time every time; we can expect a range.

5. **Response Body:** Contains the actual data returned by the API, such as user details, product information, or error messages, usually in formats like JSON or XML.

6. **JSON Schema:** A blueprint that defines the structure and data types of the JSON response. It ensures the API's response matches the expected format, making validation easier.

Workout in Postman Tool

1. Create duplicate of **Day2_StudentsAPI collection** and rename as **Day3_StudentAPIs_ResponseValidations** collection

2. Run Student API locally from command prompt **json-server –watch students.json** Access URL in browser.

3. Send **Get Single Student** Request and observe Response like Body, Cookies, Headers, Status Code, Response Time, Response data.

4. Do response validations on all requests by writing JavaScript statements in **Post-response Scripts Tab** in Postman Tool.

5. All validations are added using **Chai Assertion Library** or **Framework** in the Scripts tab under Post-response in Postman.

6. Refer **Postman Response Validations Document** for understanding Chai assertion library and different validations for Response.

7. Send multiple reqests specified in **Student API excel sheet**