

# 1. Validating Response Codes

The screenshot displays the Postman application interface. The top bar shows the workspace name 'My Workspace' and a list of recent requests. The left sidebar contains a tree view of collections and environments. The main area is divided into two sections: the top section for request configuration and the bottom section for response viewing.

**Request Configuration:**

- Method:** GET (highlighted with a red box)
- URL:** `https://automationexercise.com/api/productsList` (highlighted with a red box)
- Params:** Params, Authorization, Headers (8), Body, Scripts, Settings
- Query Params:** A table with columns Key, Value, and Description. The first row is empty, and the second row has 'Key' in the Key column and 'Value' in the Value column.

**Response Viewing:**

- Status:** 200 OK (highlighted with a red box)
- Body:** JSON (highlighted with a red box)
- Response Content:** A JSON object with the following structure:

```
1 {
2   "responseCode": 200,
3   "products": [
4     {
5       "id": 1,
6       "name": "Blue Top",
7       "price": "Rs. 500",
8       "brand": "Polo",
9       "category": {
10        "usertype": {
11          "usertype": "Women"
12        }
13      }
14    }
15  ]
16 }
```

## 2. Verifying API Response Data

The screenshot displays the Postman application interface. On the left sidebar, the 'API's Graduated Project' is expanded, showing a list of requests. The 'GET Verifying API Response Data' request is selected. The main panel shows the details of this request, which is a GET request to the URL `https://run.mocky.io/v3/68c366d6-c46b-46d3-afd9-80027f999703`. The 'Pre-request' tab is active, showing a JavaScript snippet for verifying the response body. The snippet is as follows:

```
1 const responseBody = pm.response.json();
2 const expectedBody = {
3   "id": 1,
4   "name": "John Doe",
5   "email": "john.doe@example.com"
6 };
```

The 'Post-response' tab is also active, showing the same JavaScript snippet. The 'Body' tab is selected, displaying the response body in JSON format. The response is a 200 OK status, with a response time of 315 ms and a body size of 294 B. The response body is a JSON object with the following structure:

```
{
  "id": 1,
  "name": "John Doe",
  "email": "john.doe@example.com"
}
```

The status '200 OK' and the response body are highlighted with red circles. The JavaScript snippets in both the 'Pre-request' and 'Post-response' tabs are also highlighted with red circles.

# 3. Authentication & Authorization Testing

The screenshot displays the Postman application interface for testing an API endpoint. The left sidebar shows the 'My Workspace' with a collection named 'API's Graduated Project' containing several test cases. The main panel shows a 'POST' request to 'https://reqres.in/api/login'. The request body is a JSON object with 'email' and 'password' fields. The response is a 200 OK status with a JSON body containing a 'token' field. Red hand-drawn boxes highlight the request method, URL, body, and the response status and body.

**Request Details:**

- Method: POST
- URL: https://reqres.in/api/login
- Body (JSON):

```
{  "email": "eve.holt@reqres.in",  "password": "cityslicka"}
```

**Response Details:**

- Status: 200 OK
- Time: 391 ms
- Size: 1.11 KB
- Body (JSON):

```
{  "token": "QpwL5tke4Pnpja7X4"}
```