

GET request

The screenshot shows the REST Client interface with a GET request configured. The URL is `https://jsonplaceholder.typicode.com/users/1`. The **Scripts** tab is active, displaying a series of tests and assertions. The interface includes tabs for Params, Authorization, Headers (6), Body, Scripts, and Settings. A **Send** button is visible on the right.

```
1 // Test for 200 status code
2 pm.test("Status code is 200", function () {
3   pm.response.to.have.status(200);
4 });
5
6 // Test for valid JSON response and presence of expected fields
7 pm.test("Response has expected fields", function () {
8   var jsonData;
9   try {
10    jsonData = pm.response.json();
11  } catch (e) {
12    pm.expect.fail("Response is not valid JSON");
13  }
14  // Replace the following array with the actual expected fields
15  var expectedFields = ["id", "name", "email"]; // Example fields
16  expectedFields.forEach(function(field) {
17    pm.expect(jsonData).to.have.property(field);
18  });
19 });
20
```

The screenshot shows the REST Client interface after the request has been executed. The **Test Results (2/2)** tab is active, displaying two passed tests. The status bar at the bottom indicates a **200 OK** response with a 466 ms duration and 1.44 KB of data. The interface includes tabs for Body, Cookies, Headers (25), Test Results (2/2), and Settings. A **Send** button is visible on the right.

```
1 // Test for 200 status code
2 pm.test("Status code is 200", function () {
3   pm.response.to.have.status(200);
4 });
5
6 // Test for valid JSON response and presence of expected fields
7 pm.test("Response has expected fields", function () {
8   var jsonData;
9   try {
10    jsonData = pm.response.json();
11  } catch (e) {
12    pm.expect.fail("Response is not valid JSON");
13  }
14  // Replace the following array with the actual expected fields
15  var expectedFields = ["id", "name", "email"]; // Example fields
16  expectedFields.forEach(function(field) {
17    pm.expect(jsonData).to.have.property(field);
18  });
19 });
20
```

Test Results (2/2)

- PASSED** Status code is 200
- PASSED** Response has expected fields

200 OK • 466 ms • 1.44 KB • Save Response

POST request

The screenshot shows the Postman interface for a POST request to `https://jsonplaceholder.typicode.com/posts`. The **Scripts** tab is active, displaying a JavaScript script for testing the response. The script includes tests for the status code (201), the Content-Type header (application/json), and the response body (JSON). It also defines a request body and expects the response to match it.

```
1 // Test: Status code is 201
2 pm.test("Status code is 201", function () {
3     pm.expect(pm.response.code).to.eql(201);
4 });
5
6 // Test: Content-Type header includes 'application/json'
7 pm.test("Content-Type is application/json", function () {
8     pm.response.to.have.header("Content-Type");
9     pm.expect(pm.response.headers.get("Content-Type")).to.include("application/json");
10 });
11
12 // Test: Response body contains submitted fields with correct values
13 pm.test("Response body contains submitted fields with correct values", function () {
14     var requestBody = {
15         title: "hello",
16         body: "this is a drill",
17         userId: 1
18     };
19     var jsonData = {};
20     try {
21         jsonData = pm.response.json();
22     } catch (e) {}
23     pm.expect(jsonData).to.have.property("title", requestBody.title);
24     pm.expect(jsonData).to.have.property("body", requestBody.body);
25     pm.expect(jsonData).to.have.property("userId", requestBody.userId);
26     pm.expect(jsonData).to.have.property("id"); // JSONPlaceholder adds an 'id' field
27 });
28
```

The screenshot shows the same Postman interface, but with the **Body** tab selected. The request body is displayed as raw JSON:

```
1 {
2   "title": "hello",
3   "body": "this is a drill",
4   "userId": 1
5 }
```

1 5

Body Cookies Headers (25) Test Results (3/3) ↺ 201 Created

Filter Results ▾

- PASSED** Status code is 201
- PASSED** Content-Type is application/json
- PASSED** Response body contains submitted fields with correct values

1 5

Body Cookies Headers (25) Test Results (3/3) ↺

{ } JSON ▾ ▶ Preview 🖼 Visualize ▾

```
1 {
2   "title": "hello",
3   "body": "this is a drill",
4   "userId": 1,
5   "id": 101
6 }
```