

# Part 2: API Testing via Postman

## BASE URL

<https://jsonplaceholder.typicode.com>

---

### 1. GET /users – Retrieve All Users

#### Test Case ID: API-TC-01

**Title:** Retrieve all users successfully

##### Request Details

- **Method:** GET
- **URL:** <https://jsonplaceholder.typicode.com/users>
- **Headers:**
  - Content-Type: application/json
- **Body:** None

##### Response Details

- **Status Code:** 200 OK
- **Headers:**
  - Content-Type: application/json; charset=utf-8
- **Response Body:**

```
[  
  {  
    "id": 1,  
    "name": "Leanne Graham",  
    "username": "Bret",  
    "email": "Sincere@april.biz",  
    "address": {  
      "street": "Kulas Light",  
      "suite": "Apt. 556",  
      "city": "Gwenborough",  
      "zipcode": "92998-3874",  
      "geo": {  
        "lat": "-37.3159",  
        "lng": "81.1496"  
      }  
    },  
    "phone": "1-770-736-8031 x56442",  
    "website": "hildegard.org",  
    "company": {  
      "name": "Romaguera-Crona",  
      "catchPhrase": "Multi-layered client-server neural-net",  
      "bs": "harness real-time e-markets"  
    }  
  },  
  {  
    "id": 2,  
    "name": "Ervin Howell",  
    "username": "Antonette",  
    "email": "Shanna@melissa.tv",  
    "address": {  
      "street": "Victor Plains",  
      "suite": "Suite 879",  
      "city": "Wisokyburgh",  
      "zipcode": "90566-7771",  
      "geo": {  
        "lat": "-43.9509",  
        "lng": "-34.4618"  
      }  
    },  
    "phone": "010-692-6593 x09125",  
    "website": "anastasia.net",  
  }]
```

```
        "company": {
            "name": "Deckow-Crist",
            "catchPhrase": "Proactive didactic contingency",
            "bs": "synergize scalable supply-chains"
        }
    },
    {
        "id": 3,
        "name": "Clementine Bauch",
        "username": "Samantha",
        "email": "Nathan@yesenia.net",
        "address": {
            "street": "Douglas Extension",
            "suite": "Suite 847",
            "city": "McKenziehaven",
            "zipcode": "59590-4157",
            "geo": {
                "lat": "-68.6102",
                "lng": "-47.0653"
            }
        },
        "phone": "1-463-123-4447",
        "website": "ramiro.info",
        "company": {
            "name": "Romaguera-Jacobson",
            "catchPhrase": "Face to face bifurcated interface",
            "bs": "e-enable strategic applications"
        }
    },
    {
        "id": 4,
        "name": "Patricia Lebsack",
        "username": "Karianne",
        "email": "Julianne.OConner@kory.org",
        "address": {
            "street": "Hoeger Mall",
            "suite": "Apt. 692",
            "city": "South Elvis",
            "zipcode": "53919-4257",
            "geo": {
                "lat": "29.4572",
                "lon": "-95.7179"
            }
        }
    }
]
```

```
        "lng": "-164.2990"
    }
},
"phone": "493-170-9623 x156",
"website": "kale.biz",
"company": {
    "name": "Robel-Corkery",
    "catchPhrase": "Multi-tiered zero tolerance productivity",
    "bs": "transition cutting-edge web services"
}
,
{
    "id": 5,
    "name": "Chelsey Dietrich",
    "username": "Kamren",
    "email": "Lucio_Hettinger@annie.ca",
    "address": {
        "street": "Skiles Walks",
        "suite": "Suite 351",
        "city": "Roscoeview",
        "zipcode": "33263",
        "geo": {
            "lat": "-31.8129",
            "lng": "62.5342"
        }
    },
    "phone": "(254) 954-1289",
    "website": "demarco.info",
    "company": {
        "name": "Keebler LLC",
        "catchPhrase": "User-centric fault-tolerant solution",
        "bs": "revolutionize end-to-end systems"
    }
},
{
    "id": 6,
    "name": "Mrs. Dennis Schulist",
    "username": "Leopoldo_Corkery",
    "email": "Karley_Dach@jasper.info",
    "address": {
        "street": "Norberto Crossing",
        "suite": "Apt. 550",
        "city": "South Jenniferport",
        "zipcode": "90263",
        "geo": {
            "lat": "-31.8129",
            "lng": "62.5342"
        }
    },
    "phone": "493-170-9623 x156",
    "website": "kale.biz",
    "company": {
        "name": "Robel-Corkery",
        "catchPhrase": "Multi-tiered zero tolerance productivity",
        "bs": "transition cutting-edge web services"
    }
}
```

```
        "suite": "Apt. 950",
        "city": "South Christy",
        "zipcode": "23505-1337",
        "geo": {
            "lat": "-71.4197",
            "lng": "71.7478"
        },
    },
    "phone": "1-477-935-8478 x6430",
    "website": "ola.org",
    "company": {
        "name": "Considine-Lockman",
        "catchPhrase": "Synchronised bottom-line interface",
        "bs": "e-enable innovative applications"
    }
},
{
    "id": 7,
    "name": "Kurtis Weissnat",
    "username": "Elwyn.Skiles",
    "email": "Telly.Hoeger@billy.biz",
    "address": {
        "street": "Rex Trail",
        "suite": "Suite 280",
        "city": "Howemouth",
        "zipcode": "58804-1099",
        "geo": {
            "lat": "24.8918",
            "lng": "21.8984"
        }
    },
    "phone": "210.067.6132",
    "website": "elvis.io",
    "company": {
        "name": "Johns Group",
        "catchPhrase": "Configurable multimedia task-force",
        "bs": "generate enterprise e-tailers"
    }
},
{
    "id": 8,
```

```
"name": "Nicholas Runolfsdottir V",
"username": "Maxime_Nienow",
"email": "Sherwood@rosamond.me",
"address": {
    "street": "Ellsworth Summit",
    "suite": "Suite 729",
    "city": "Aliyaview",
    "zipcode": "45169",
    "geo": {
        "lat": "-14.3990",
        "lng": "-120.7677"
    }
},
"phone": "586.493.6943 x140",
"website": "jacynthe.com",
"company": {
    "name": "Abernathy Group",
    "catchPhrase": "Implemented secondary concept",
    "bs": "e-enable extensible e-tailers"
},
},
{
    "id": 9,
    "name": "Glenna Reichert",
    "username": "Delphine",
    "email": "Chaim_McDermott@dana.io",
    "address": {
        "street": "Dayna Park",
        "suite": "Suite 449",
        "city": "Bartholomebury",
        "zipcode": "76495-3109",
        "geo": {
            "lat": "24.6463",
            "lng": "-168.8889"
        }
    },
    "phone": "(775) 976-6794 x41206",
    "website": "conrad.com",
    "company": {
        "name": "Yost and Sons",
        "catchPhrase": "Switchable contextually-based project",
```

```
        "bs": "aggregate real-time technologies"
    }
},
{
    "id": 10,
    "name": "Clementina DuBuque",
    "username": "Moriah.Stanton",
    "email": "Rey.Padberg@karina.biz",
    "address": {
        "street": "Kattie Turnpike",
        "suite": "Suite 198",
        "city": "Lebsackbury",
        "zipcode": "31428-2261",
        "geo": {
            "lat": "-38.2386",
            "lng": "57.2232"
        }
    },
    "phone": "024-648-3804",
    "website": "ambrose.net",
    "company": {
        "name": "Hoeger LLC",
        "catchPhrase": "Centralized empowering task-force",
        "bs": "target end-to-end models"
    }
}
]
```

## Short Analysis

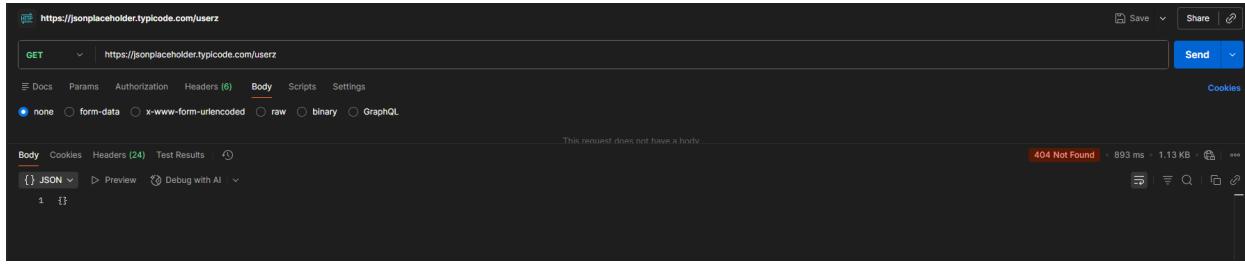
The API successfully returned a list of users. Data is correctly structured in JSON format and contains valid user attributes.

---

## Edge Case – Invalid Endpoint

URL: <https://jsonplaceholder.typicode.com/userz>

- **Status Code:** 404 Not Found
- **Screenshot**



The API correctly handles invalid resource paths.

---

## 2. GET /users/1 – Retrieve Specific User

### Test Case ID: API-TC-02

**Title:** Retrieve a valid user by ID

#### Request Details

- **Method:** GET
- **URL:** <https://jsonplaceholder.typicode.com/users/1>
- **Headers:**
  - Content-Type: application/json
- **Body:** None

#### Response Details

- **Status Code:** 200 OK
- **Headers:**
  - Content-Type: application/json; charset=utf-8
- **Response Body:**

The screenshot shows a Postman interface with a successful API call. The URL is <https://jsonplaceholder.typicode.com/users/1>. The response code is 200 OK, with a duration of 116 ms and a size of 1.43 KB. The response body is a JSON object representing a user:

```
1 {  
2   "id": 1,  
3   "name": "Leanne Graham",  
4   "username": "Bret",  
5   "email": "Sincere@april.biz",  
6   "address": {  
7     "street": "Kulas Light",  
8     "suite": "Apt. 556",  
9     "city": "Gwenborough",  
10    "zipcode": "92998-3874",  
11    "geo": {  
12      "lat": "-37.3190",  
13      "lng": "81.1496"  
14    }  
15  },  
16  "phone": "1-770-736-8031 x56442",  
17  "website": "hildegard.org",  
18  "company": {  
19    "name": "Romaguera-Crona",  
20    "catchPhrase": "Multi-layered client-server neural-net",  
21    "bs": "harness real-time e-markets"  
22  }  
23 }
```

## Short Analysis

The correct user is returned based on the provided ID. The response structure is accurate and complete.

---

## Edge Case – Non-existing User

URL: <https://jsonplaceholder.typicode.com/users/999>

- **Status Code:** 404 Not Found and sends an {} empty object
- **Screenshot**

The screenshot shows a Postman interface with a failed API call. The URL is <https://jsonplaceholder.typicode.com/users/999>. The response code is 404 Not Found, with a duration of 721 ms and a size of 1.12 KB. The response body is an empty JSON object:

```
1 {}
```

---

The API properly handles a request for a non-existent resource.

### 3. POST /posts – Create New Post

#### Test Case ID: API-TC-03

**Title:** Create a new post successfully

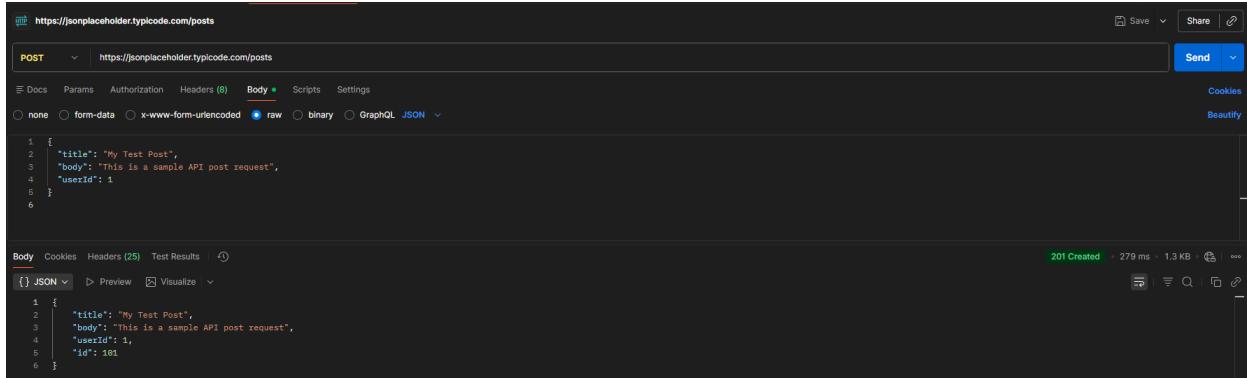
#### Request Details

- **Method:** POST
- **URL:** <https://jsonplaceholder.typicode.com/posts>
- **Headers:**
  - Content-Type: application/json
- **Request Body:**

```
{  
  "title": "My Test Post",  
  "body": "This is a sample API post request",  
  "userId": 1  
}
```

#### Response Details

- **Status Code:** 201 Created
- **Headers:**
  - Content-Type: application/json; charset=utf-8
- **Response Body:**



The screenshot shows a Postman interface with a POST request to `https://jsonplaceholder.typicode.com/posts`. The request body is a JSON object with three fields: `title`, `body`, and `userId`. The response status is `201 Created`.

```
1 {
2   "title": "My Test Post",
3   "body": "This is a sample API post request",
4   "userId": 1
5 }
```

```
201 Created
```

## Short Analysis

The API simulates successful resource creation and returns a new ID. Since this is a mock API, no actual data persistence is expected.

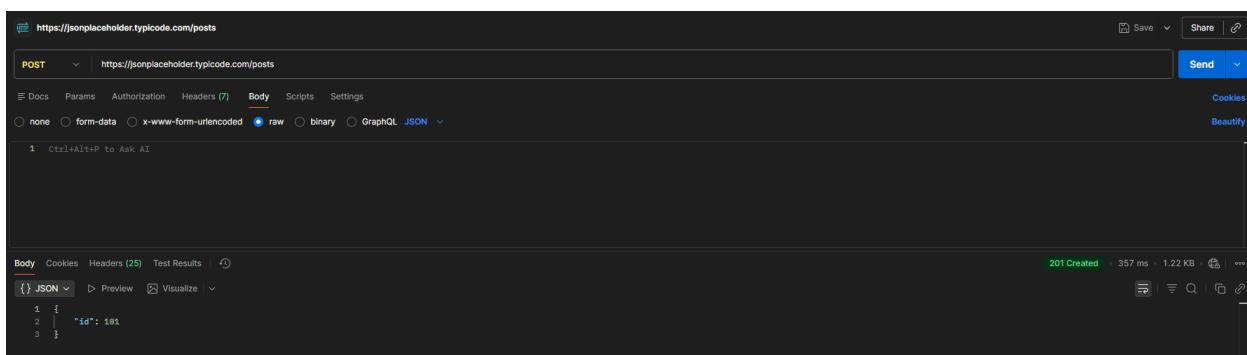
---

## Edge Case – Missing Required Fields

### Request Body:

```
{}
```

- **Status Code:** 201 Created
- **Actual Behavior:** Still allows post creation



The screenshot shows a Postman interface with a POST request to `https://jsonplaceholder.typicode.com/posts`. The request body is an empty JSON object. The response status is `201 Created`.

```
1 Ctr+Alt+P to Ask All
```

```
201 Created
```

This highlights that validation is not enforced in the mock API.

---

## 4. PUT /posts/1 – Update Existing Post

### Test Case ID: API-TC-04

**Title:** Update an existing post successfully

#### Request Details

- **Method:** PUT
- **URL:** <https://jsonplaceholder.typicode.com/posts/1>
- **Headers:**
  - Content-Type: application/json
- **Request Body:**

```
{  
  "title": "Updated Title",  
  "body": "Updated Body Content",  
  "userId": 1  
}
```

#### Response Details

- **Status Code:** 200 OK
- **Headers:**
  - Content-Type: application/json; charset=utf-8
- **Response Body:**

https://jsonplaceholder.typicode.com/posts/1

PUT https://jsonplaceholder.typicode.com/posts/1

Body (JSON)

```
{  
  "title": "Updated Title",  
  "body": "Updated Body Content",  
  "userId": 1  
}
```

200 OK

## Short Analysis

The API successfully simulates updating an existing resource. Data returned matches the updated request body.

---

## Edge Case – Invalid ID

URL: <https://jsonplaceholder.typicode.com/posts/101>

- Status Code: 500 Internal Server Error

https://jsonplaceholder.typicode.com/posts/101

PUT https://jsonplaceholder.typicode.com/posts/101

Body (JSON)

```
{  
  "title": "Updated Title",  
  "body": "Updated Body Content",  
  "userId": 1  
}
```

500 Internal Server Error

```
TypeError: Cannot read properties of undefined (reading 'id')  
at update (/app/node_modules/json-server/lib/server/router/plural.js:262:24)  
at layer.handle [as handle_request] (/app/node_modules/express/lib/router/layer.js:95:5)  
at next (/app/node_modules/express/lib/router/route.js:137:13)  
at next (/app/node_modules/express/lib/router/route.js:131:14)  
at Route.dispatch (/app/node_modules/express/lib/router/route.js:112:3)  
at Layer.handle [as handle_request] (/app/node_modules/express/lib/router/layer.js:95:5)  
at /app/node_modules/express/lib/router/index.js:28:5  
at /app/node_modules/express/lib/router/index.js:28:5  
at parse (/app/node_modules/express/lib/router/index.js:394:14)  
at /app/node_modules/express/lib/router/index.js:369:14  
at parse (/app/node_modules/express/lib/router/index.js:369:14)  
at Function.process_params (/app/node_modules/express/lib/router/index.js:410:3)
```

The API returns a server-side error when attempting to update a non-existent resource using this specific ID. However, I believe that a 404 Not Found would be more appropriate for an invalid resource identifier.

---