JSONPlaceholder API Documentation

> This technical documentation provides details for interacting with the JSONPlaceholder test API.

JSONPlaceholder is a fake online REST API used for testing and prototyping.
It supports common HTTP methods like `GET`, `POST`, `PUT`, and `DELETE`, and returns predictable JSON responses.

You can use it to practice:

- Fetching and filtering resources using query and path parameters
- Creating new posts
- Viewing, editing, and deleting example data

No authentication is required.
Data does not persist—it's for testing only.

Available Endpoints

Method	Endpoint	Description
GET	/users	Retrieve all users
GET	/comments?postId=1	Get comments for a specific post
GET	/posts/{id}	Get one post by ID
POST	/posts	Create a new post
PUT	/posts/{id}	Updates an existing post by replacing all its fields.
DELETE	/posts/{id}	Deletes a post by specific ID.
PATCH	/posts/{id}	Partially updates an existing post by ID

GET /users

Retrieves a list of user profiles from the JSONPlaceholder testing API.

Base URL: `https://jsonplaceholder.typicode.com`

Full Request Example:

```
GET https://jsonplaceholder.typicode.com/users
```

Response


```
[
 {
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz"
```

```
  }
]
```

Response Fields

Field	Type	Description
`id`	number	The unique ID of the user
`name`	string	The full name of the user
`username`	string	A unique name that identifies the user. Alphanumeric, no spaces.
`email`	string	The email address of the user

Success Response Code:

`200 OK`

Possible Errors:

GET /comments

Retrieves a list of comments associated with a specific post using a query parameter.

Base URL:

`https://jsonplaceholder.typicode.com`

Full Request Example:

`https://jsonplaceholder.typicode.com/comments?postId=1`

Query Parameters

Parameter	Type	Required	Description
`postId`	number	Yes	The ID of the post to fetch comments for


```
[
 {
        "postId": 1,
        "id": 1,
        "name": "id labore ex et quam laborum",
        "email": "Eliseo@gardner.biz",
        "body": "laudantium enim quasi est quidem magnam voluptate ipsam
eos\ntempora quo necessitatibus\ndolor quam autem quasi\nreiciendis et nam
sapiente accusantium"
    }
]
```

Response Fields

Field	Type	Description
`postId`	number	The ID of the post this comment belongs to
`id`	number	The unique ID of the comment
`name`	string	The name/title of the comment
`email`	string	The email address of the comment author
`body`	string	The full text of the comment

Success Status Code:

`200 OK`

Possible Errors:

This mock endpoint does not return real errors, but in production, errors might include:

- `400 Bad Request` (if a parameter is malformed)
- `404 Not Found` (if the postId doesn't match any resource)

GET /posts/{id}

Retrieves a single post by its unique ID.

Base URL

`https://jsonplaceholder.typicode.com`

Full Request Example:

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

Path Parameters:

Parameter	Type	Required	Description
`id`	number	Yes	The ID of the post to retrieve

Success Response

Status Code:

`200 OK`


```
{
  "userId": 1,
  "id": 1,
```

```
    "title": "sunt aut facere repellat provident occaecati excepturi optio
reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et"
 }
```


Response Fields

Field	Type	Description
`userId`	number	The ID of the user who created the post
`id`	number	The unique ID post
`title`	string	The title of the post
`body`	string	Main content of the post

POST /posts

Base URL

`https://jsonplaceholder.typicode.com`

Full Request Example

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

Success Response:

`201 Created`

Request Body

Field	Type	Required	Description
`title`	string	Yes	Title of the new post
`body`	string	Yes	Content/body of the new post
`userId`	number	Yes	ID of the user creating the post

Success Response

Status Code:

`201 Created`


```
{
 "id": 1,
 "title": "Updated SparkPost",
```

```
  "body": "This post has been overwritten by the admin.",
  "userId": 1
  }
```

DELETE /posts/{id}

Deletes a specific post by its ID.

Base URL

`https://jsonplaceholder.typicode.com`

Full Request Example:

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

Path Parameters

Parameter	Type	Required	Description
`id`	number	Yes	ID of the post to delete

Success Response

Status Code

- `200 OK`
- `204 No Content` (standard for successful deletion)

Possible Errors

- `404 Not Found` if the post does not exist
- `403 Forbidden` or `401 Unauthorized` (in real APIs with authentification)

Response Body:


```
  {}
```

PATCH /posts/{id}

Partially updates an existing post by ID. Only the fields provided in the request body will be updated. Other fields remain unchanged.

Base URL:

`https://jsonplaceholder.typicode.com`

Full Example:

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

Request Body

```
{
  "title": "Updated Title with PATCH"
}
```

Success Response

Status Code:

```
200 OK
```

```
{
  "userId": 1,
  "id": 1,
  "title": "Updated Title with PATCH",
  "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et
cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem
eveniet architecto"
}
```

Notes:

This mock API does not persist changes, but it simulates a valid PATCH behavior.

PUT /posts/{id}

Updates an existing post by replacing all of its fields. The full post object must be included in the request body.

Base URL

```
https://jsonplaceholder.typicode.com
```

Full Request Example

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

Path Parameters

Parameter	Type	Required	Description
`id`	number	Yes	The ID of the post to update

Request Body

Field	Type	Required	Description
`title`	string	Yes	Updated title for the post
`body`	string	Yes	Updated content of the post
`userId`	number	Yes	ID of the user who owns the post
`id`	number	Yes	ID of the post (must match in body)

Success Response

Status Code:

`200 OK`


```
{
 "id": 1,
 "title": "Updated SparkPost",
 "body": "This post has been overwritten by vaultcore style.",
 "userId": 1
}
```

Authentication

> Note: This API does not require authentication.

JSONPlaceholder does not require authentication.
All endpoints are publicly accessible and do not require tokens, API keys, or login credentials.

In real APIs, you might expect:

- `401 Unauthorized` – if no auth token is provided
- `403 Forbidden` – if the token is valid but access is denied

Error Handling

This mock API simulates successful responses for all standard requests.
It does not generate real errors for malformed or unauthorized requests.

In a real-world API, typical error responses include:

Status Code	Meaning	When it Happens
`400`	Bad Request	Missing required fields, invalid format
`401`	Unauthorized	Missing or invalid auth credentials
`403`	Forbidden	Valid auth, but not allowed

Status Code	Meaning	When it Happens
404	Not Found	Resource does not exist
500	Internal Server Error	Server failed to process the request

{ "error": "Invalid post ID", "status": 404 }
