

Guide

You can use JSONPlaceholder with any type of project that needs to get JSON data (React, Vue, Node, Rails, Swift, Android, ...).

Below you'll find examples using [Fetch API](#). You can copy paste them in your browser Console to quickly test JSONPlaceholder.

Get a resource

```
fetch('https://jsonplaceholder.typicode.com/posts/1')
  .then(response => response.json())
  .then(json => console.log(json))

// Output
{
  id: 1,
  title: '[...]',
  body: '[...]',
  userId: 1
}
```

List all resources

```
fetch('https://jsonplaceholder.typicode.com/posts')
  .then((response) => response.json())
  .then((json) => console.log(json))[
  // Output
  ({ id: 1, title: '[...]' /* ... */ },
```

```
/* ... */  
{ id: 100, title: '[...]' /* ... */ }  
]
```

Create a resource

```
fetch('https://jsonplaceholder.typicode.com/posts', {  
  method: 'POST',  
  body: JSON.stringify({  
    title: 'foo',  
    body: 'bar',  
    userId: 1  
  }),  
  headers: {  
    "Content-type": "application/json; charset=UTF-8"  
  }  
})  
  .then(response => response.json())  
  .then(json => console.log(json))  
  
// Output  
{  
  id: 101,  
  title: 'foo',  
  body: 'bar',  
  userId: 1  
}
```

Important: the resource will not be really created on the server but it will be faked as if. In other words, if you try to access a post using 101 as an id, you'll get a 404 error.

Update a resource

With PUT

```
fetch('https://jsonplaceholder.typicode.com/posts/1', {  
  method: 'PUT',  
  body: JSON.stringify({  
    id: 1,  

```

```
    title: 'foo',
    body: 'bar',
    userId: 1
  )),
  headers: {
    "Content-type": "application/json; charset=UTF-8"
  }
})
.then(response => response.json())
.then(json => console.log(json))

// Output
{
  id: 1,
  title: 'foo',
  body: 'bar',
  userId: 1
}
```

With PATCH

```
fetch('https://jsonplaceholder.typicode.com/posts/1', {
  method: 'PATCH',
  body: JSON.stringify({
    title: 'foo'
  }),
  headers: {
    "Content-type": "application/json; charset=UTF-8"
  }
})
.then(response => response.json())
.then(json => console.log(json))

// Output
{
```

This service is maintained and provided for free, please consider supporting it on [GitHub Sponsors](#) ❤️

Important: the resource will not be really updated on the server but it will be faked as if.

Delete a resource

```
fetch('https://jsonplaceholder.typicode.com/posts/1', {  
  method: 'DELETE',  
})
```

Important: the resource will not be really deleted on the server but it will be faked as if.

Filter resources

Basic filtering is supported through query parameters.

```
// Will return all the posts that belong to the first user  
fetch('https://jsonplaceholder.typicode.com/posts?userId=1')  
  .then((response) => response.json())  
  .then((json) => console.log(json))
```

Nested resources

One level of nested route is available.

```
// Equivalent to /comments?postId=1  
fetch('https://jsonplaceholder.typicode.com/posts/1/comments')  
  .then((response) => response.json())  
  .then((json) => console.log(json))
```

Available nested routes:

<https://jsonplaceholder.typicode.com/posts/1/comments>
<https://jsonplaceholder.typicode.com/albums/1/photos>
<https://jsonplaceholder.typicode.com/users/1/albums>
<https://jsonplaceholder.typicode.com/users/1/todos>
<https://jsonplaceholder.typicode.com/users/1/posts>

 [Sponsor this project on GitHub](#)

Coded and built with ❤️ by [typicode](#)
Source code and CHANGELOG available on [GitHub](#)