**1: Use fetch() to GET data**

fetch('https://jsonplaceholder.typicode.com/posts')

.then(response => response.json())

.then(posts => {

posts.slice(0, 5).forEach(post => {

console.log(post.title);

});

})

.catch(error => console.error('Error:', error));

**2: Send a POST request using fetch()**

const newPost = {

title: 'My New Post',

body: 'This is the body of my new post.'

};

fetch('https://jsonplaceholder.typicode.com/posts', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(newPost)

})

.then(response => {

console.log('Status Code:', response.status);

return response.json();

})

.then(data => console.log('Response:', data))

.catch(error => console.error('Error:', error));

**3: Send a PUT request to update data**

const updatedPost = {

title: 'Updated Title',

body: 'Updated body content.'

};

fetch('https://jsonplaceholder.typicode.com/posts/1', {

method: 'PUT',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(updatedPost)

})

.then(response => {

console.log('Status Code:', response.status);

return response.json();

})

.then(data => console.log('Updated Response:', data))

.catch(error => console.error('Error:', error));

**4: Send a DELETE request**

fetch('https://jsonplaceholder.typicode.com/posts/1', {

method: 'DELETE'

})

.then(response => {

console.log('Status Code:', response.status);

if (response.status === 200 || response.status === 204) {

console.log('Post deleted successfully');

} else {

console.log('Delete failed');

}

})

.catch(error => console.error('Error:', error));

**5: Convert between JavaScript Object and JSON**

const person = {

name: 'Alice',

age: 25,

city: 'New York'

};

const jsonString = JSON.stringify(person);

const parsedObject = JSON.parse(jsonString);

console.log('Original Object:', person);

console.log('JSON String:', jsonString);

console.log('Parsed Object:', parsedObject);

**HTTP Status Codes Explanation**

| **Code** | **Meaning** | **Category** |
| --- | --- | --- |
| **200** | OK – The request succeeded. | 2xx (Success) |
| **201** | Created – Resource was successfully created. | 2xx (Success) |
| **204** | No Content – The request succeeded but returns no content. | 2xx (Success) |
| **400** | Bad Request – The server couldn’t understand the request due to invalid syntax. | 4xx (Client Error) |
| **404** | Not Found – The requested resource could not be found. | 4xx (Client Error) |
| **401** | Unauthorized – Authentication is required and has failed or not been provided. | 4xx (Client Error) |
| **403** | Forbidden – The server understood the request but refuses to authorize it. | 4xx (Client Error) |
| **500** | Internal Server Error – The server encountered an unexpected condition. | 5xx (Server Error) |
| **502** | Bad Gateway – Server received an invalid response from the upstream server. | 5xx (Server Error) |
| **503** | Service Unavailable – The server is not ready to handle the request. | 5xx (Server Error) |