

Basic Usage

The Fetch API provides a global `fetch()` method that only has one mandatory argument, which is the URL of the resource you wish to fetch. A very basic example would look something like the following piece of code:

Copy

```
fetch('https://example.com/data').then( // code that handles the response ).catch( // code that runs if the server returns an error )
```

This means that we can use an `if` block to check if the request was successful, and throw an error otherwise:

Copy

```
const url = 'https://example.com/data';

fetch(url).then((response) => {    if(response.ok) {        return response;    }    throw Error(response.statusText);}).then( response => // do something with response ).catch( error => console.log('There was an error!') )
```

Text Responses

The `text()` method takes a stream of text from the response, reads it to completion and then returns a promise that resolves to a USVString object that can be treated as a string in JavaScript.

Here is an example of how a text response promise would be resolved:

Copy

```
fetch(url).then( response => response.text() ); // transforms the text stream into a JavaScript string.then( text => console.log(text) ).catch( error => console.log('There was an error: ', error))
```

JSON Responses

JSON is probably the most common format for AJAX responses. The `json()` method is used to deal with these by transforming a stream of JSON data into a promise that resolves to a JavaScript object.

Here is an example of how a JSON response promise would be resolved:

Copy

```
fetch(url).then( response => response.json() ); // transforms the JSON
data into a JavaScript object.then( data =>
console.log(Object.entries(data)) ).catch( error => console.log('There
was an error: ', error))
```

Creating Response Objects

Although most of the time you will be dealing with a response object that is returned from a request you make, you can also create your own response objects using a constructor function:

Copy

```
const response = new Response( 'Hello!', {    ok: true,    status:
200,    statusText: 'OK',    type: 'cors',    url: '/api'});
```