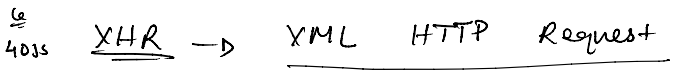
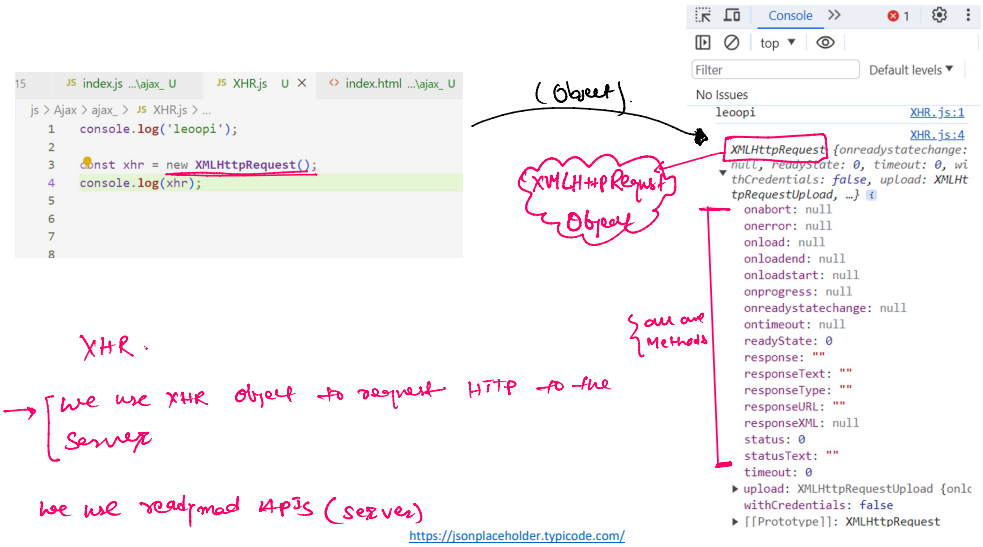
Ajax
06 November 2023 21:01

A Hand Book Of AJAX

By -Sauraj

```
// AJAX : - asynchronous javascript and XML

//IT IS SET OF WEB DEVELOPMENT TECHMIQUE"

// USING MANY WEB APPLICATION ON THE CLIENTN SIDE TO CREATE A
ASYNCHRONONUS WEB APPLICATION

// WITH AJAX WEB APPLICTION SEND AND RETRIVE DATA FROM A SERVER
ASYNCHRONOUS(in the background)

//without interfacing the display and behaviour of the existing page
// we don't use data in XML format anymore.
// we use json now

-------------------------------------------------------------------------
----
// we have three most common ways to create send request to the server
-------------------------------------------------------------------------
----

// 1.) xmlHTTPRequest (old way to ding
// 2.) fetch API (new way to doing)
//3.)axios( this is the third party library)
```





XHR.

→ [we use XHR object to request HTTP to the Server

we use readymad APIs (servers)

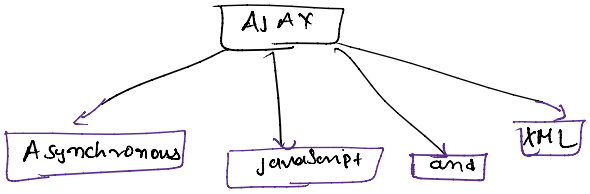
https://jsonplaceholder.typicode.com/

Routes
All HTTP methods are supported. You can use http or https for your requests.

GET /posts
GET /posts/1
GET /posts/1/comments
GET /comments?postId=1
POST /posts
PUT /posts/1
PATCH /posts/1
DELETE /posts/1
Note: see guide for usage examples.

Resources
JSONPlaceholder comes with a set of 6 common resources:

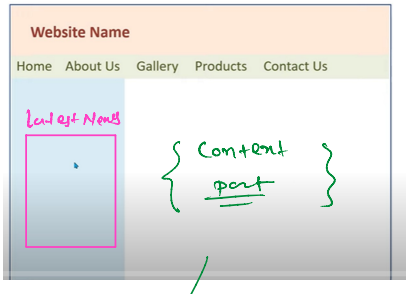
/posts 100 posts
/comments 500 comments
/albums 100 albums
/photos 5000 photos
/todos 200 todos
/users 10 users



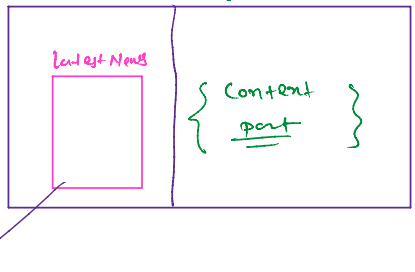
→ AJAY is a technique for creating fast and dynamic web pages.

How ?

Let's Suppose we have a website having Some headers and one side bar



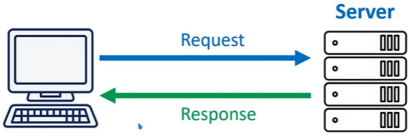
If is a news web website, whenever we click any news my heady should not be changed on content part will be changed. (load)



I want to change only latest new and not content part (which is already someone reading)

In case of normal

website visitor will be request to the Server for particular data (file/ news/ image --)

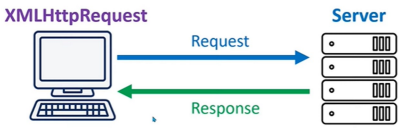


but in case of AJAX

→ [User request will not go directly to the server
We use JavaScript b/w user and server

and this class which is belong to JavaScript Name is

XMLHttpRequest → class



[It will show page without the refreshing the Page.

[we can update the page content without refreshing the page

↓

[we have 3 type of

we have 3 type of Content

1. Text File
2. XML Data
3. JSON Data

the whole process divides into 5 steps and we say that

readyState

5 Steps

readyState

0: request not initialized
1: server connection established
2: request received
3: processing request
4: request finished and response is ready

When server send request

1. status (kind of code which show about result)
2. responseText/ response XML

1. Status

200: "OK"
403: "Forbidden"
404: "Not Found"

How AJAX work in JavaScript

1. first we will create an object of XMLHttpRequest() class

```
var xhttp = new XMLHttpRequest();
```

using this object or reference we can send or receive (request and response respectively) to the server

to do this task we have 2 method
1. open, 2. send

```
1 xhttp.open("GET", "filename.txt", true);
2 xhttp.send();
```

send the request to server

Synchronous or asynchronous

1. open ("GET", any file, true/false)

Method

if is about asynchronize

If it is about asynchronous mode on/off

If I make it false then it could not we send my response immediately

Here one work done then another will be done then another

* we check ready state always → it is help us to know my response is coming from server or not

To check it we check one condition.

```
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
    }
};
```

this property will check ki kya server se response aaya ya nhi

OK

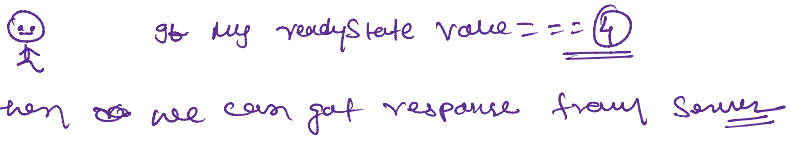
then we can perform our

→ work according to our requirement

Ex →

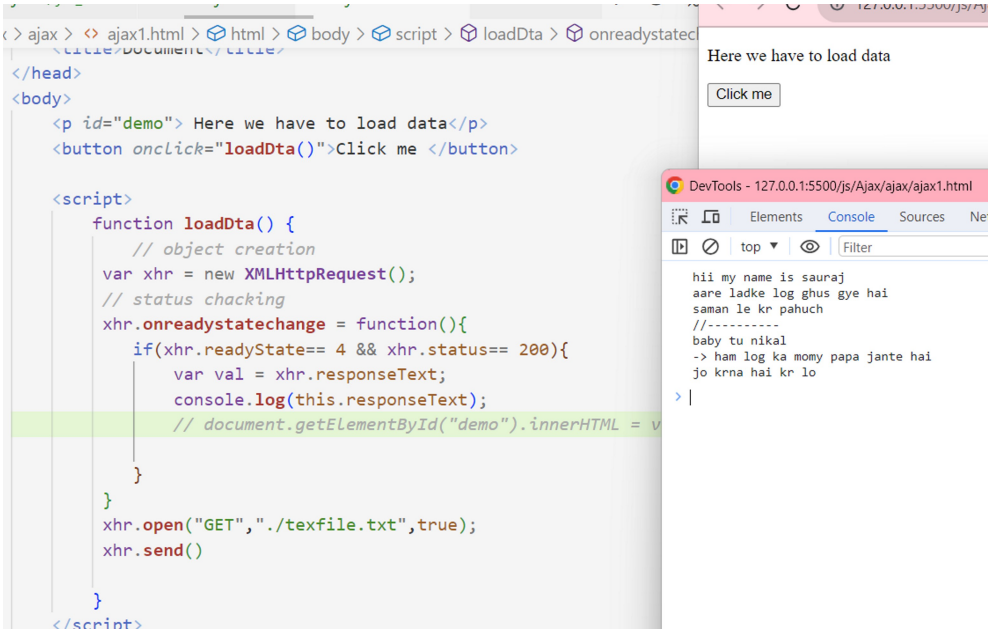
```
var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        document.getElementById("demo").innerHTML = this.responseText;
    }
};

xhttp.open("GET", "filename.txt", true);
xhttp.send();
```

checking my ready state every time

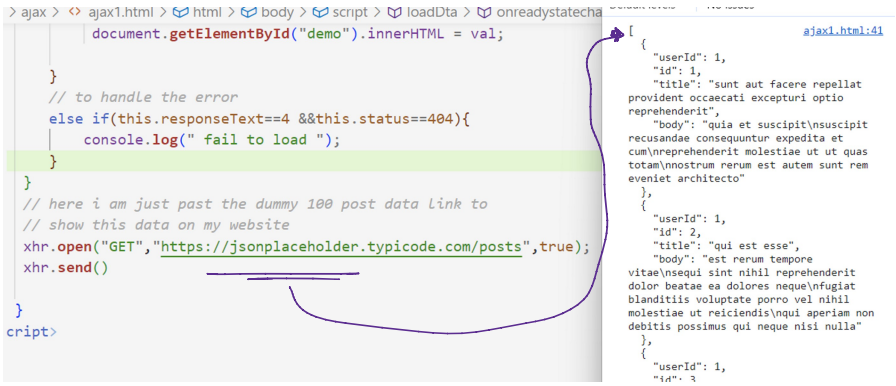


if my readyState value = = = 4

then we can get response from Server

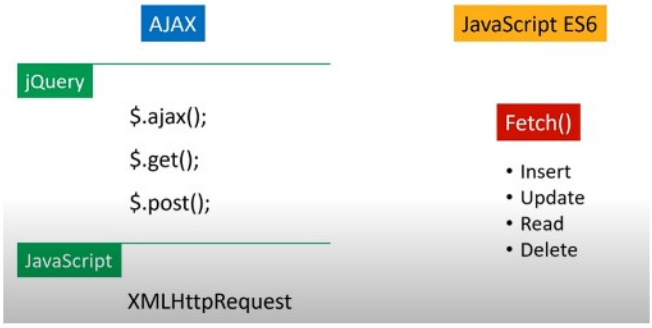


Now using a fack post website URL to show thair data on my website



☆ Fetch Method ( ) ES6V

What is Feth ( ) Method ?



fetch Method return a promise

fetch (file / URL)

resolved

reject

So we can use then → function

If fetch return Sucessful then we can use then

then (callback fn) → it will take any callback

Now see How fetch () work in My case

fetch (file / URL) → return → Promise

fetch (file/URL) . then () → return promise

fetch (file/URL) . then (function (response) {

return response . data ;

}) ;

from server

.text()

JSON()

any variable name

it will go inside 2nd then function

```
fetch(file / URL).then(function(response){
    return response.data;
}).then(function(result){
    console.log(result);
});
```

then Method take a function as a argument

(for more details see promise

```
    console.log(result);
});
```

a function as a argument

(for more details see promise note)

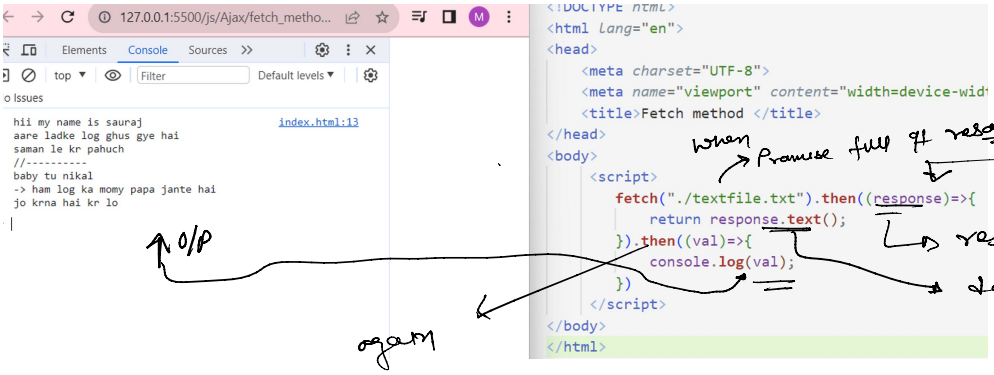
If my promise fail then we use catch to show error message

```
.catch(function(error){
  console.log(error);
```

fetch method work on Live Server

```
fetch(file / URL).then(function(response){
    return response.data;
}).then(function(result){
    console.log(result);
}).catch(function(error){
    console.log(error);
});
```

Example (working with simple text file)



Now working with fake json data

https://jsonplaceholder.typicode.com/users → 10 users fake data we are using here

```
<script>
    // 10 fake data we are using here
    // it will return json file
    fetch("https://jsonplaceholder.typicode.com/users").then((respon
se)=>{
        return response.text();
    }).then((val)=>{
        // console.log(val);
```

```
        // // document.write(val);
        for(var key in val){
           document.write(`${val[key].name}-${val[key].email}<br>`);
        }
    }).catch((err)=>{
        console.log(err +"  ->can't fetch data");
    })
</script>
```

```
<!-- ------------------------------------------------------------ -->
    <!-- how we will save data in server   -->
    <!-- i want to insert data , update data and delete the data  -->
    <!-- ------------------------------------------------------------
```

So ~~we~~ with the help of fetch ()

Fetch() – Insert, Update, Delete

fetch(file / URL);

↳ Server side language file
(php, python ... --)

Syntax

fetch (file/URL , { })

↳ parameter

Ex → some Properties

fetch (file/URL , {

① method : "post" , → for insert the data
put → for update the data
Delete → to delete the data

② body : data,

↳ this data is use to store in server

It will be either
form data or JSON Data or Text

③ header : {

↳ in side header we pass data formate which
I will pass through body (Content-type)

'Content-Type' : 'application/json',

},

↳ basically it is supporting

'Content-Type': 'application/json',

}

↳ basically gt is showing data is in json form which i passed through body

gt is not a json data then we have are many option (form data)

Syntax for form data

↳ we have a method in javaScript name → form data

Content-Type: 'application/x-www-form-urlencoded'

```
fetch(file / URL, {
    method : "POST",  ⟶ "PUT"  "DELETE"  "GET"
    body : data,  ⟶ Form Data / JSON Data / Text
    header : {
      'Content-Type': 'application/json',
    },
});

'Content-Type': 'application/x-www-form-urlencoded'
```

Let's code

I want to store my data into data base

we will use fake json data

go → json placeholder

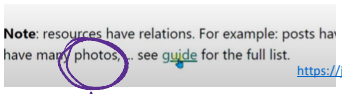
Resources

JSONPlaceholder comes with a set of 6 common reso

/posts	100 posts
/comments	500 comments
/albums	100 albums
/photos	5000 photos
/todos	200 todos
/users	10 users

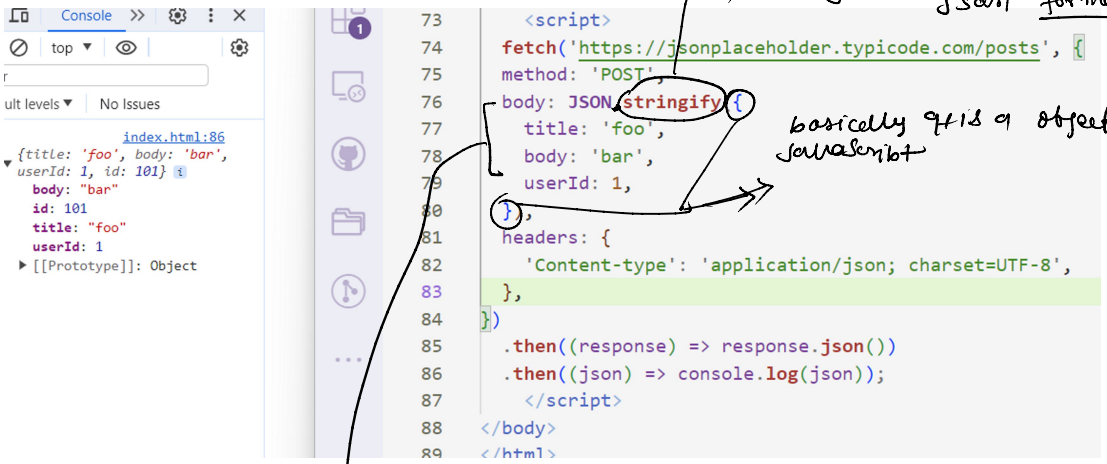
Note: resources have relations. For example: posts ha
have many photos, ... see guide for the full list.

https://jsonplaceholder.typicode.com/guide/



click

we can't send object directly using JSON.stringify we convert object to json formate



basically this is a object of javascript

we can write it in diff ways

```
<script>
  var obj = {
    title: 'foo',
    body: 'bar',
    userId: 1,
  }
  fetch('https://jsonplaceholder.typicode.com/posts', {
    method: 'POST',
    body: JSON.stringify(obj),
    headers: {
      'Content-type': 'application/json; charset=UTF-8',
    },
```

→ first we create object then pass here to convert it into json formate

to update

```
ript>
    //---------------------------------------------------------
    // Updating a resource
    //---------------------------------------------------------
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));
```

→ where we have to update path

→ put

→ Thrugh form I am storing the data in database

```
.then((json) => console.log(json));
```

→ Thraugh form I am storing the data in database