

```
<script>
  function ajax(){
    const http = new XMLHttpRequest()
    const url = 'http://localhost:3000/api/magazine'
    http.onreadystatechange = function(){
      if (http.readyState == 4) {
        console.log(this.responseText)
        document.getElementById('result').innerHTML += this.responseText + '\n'
      }
    }
  }

```

AJAX



Js

Fernando Saez
saezfernando@Gmail.com

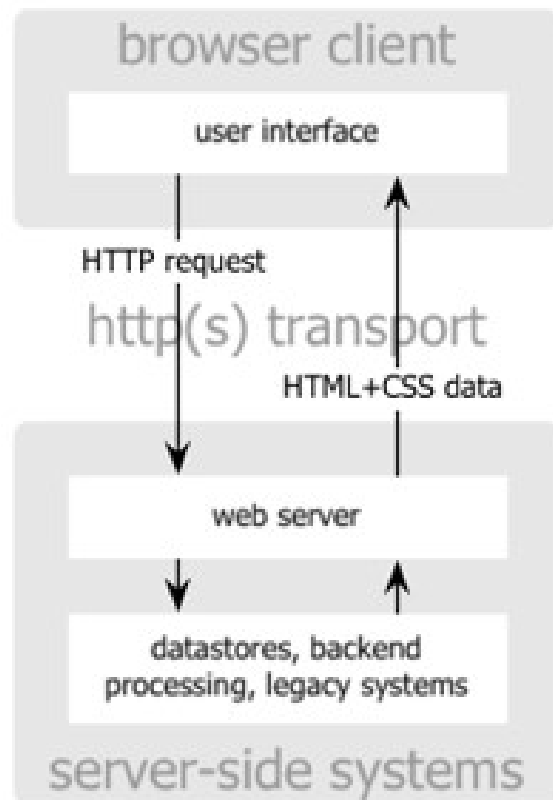
AJAX: Asynchronous JavaScript & XML

- Técnica de desarrollo Web para crear aplicaciones Web Interactivas
- Las aplicaciones se ejecutan en el navegador del usuario y se comunican asíncronamente con el servidor en background.

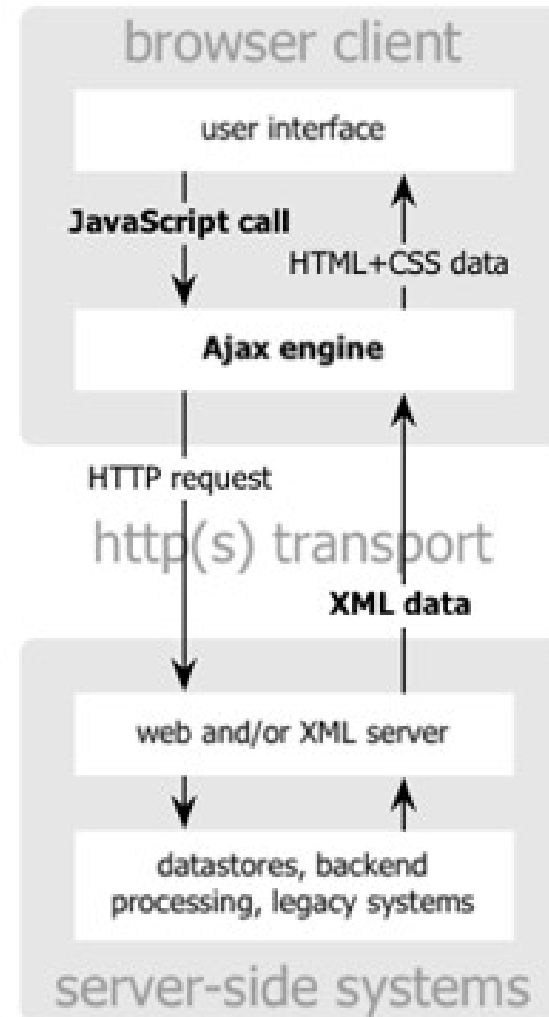
Ajax engloba varias tecnologías

- HTML y CSS para la presentación
- DOM para mostrar e interactuar con la información dinámica
- Intercambio de datos usando XML y XSLT (también JSON)
- XMLHttpRequest para el intercambio de datos asíncronamente.
- Javascript para unir todo

AJAX vs Apps Web Tradicionales

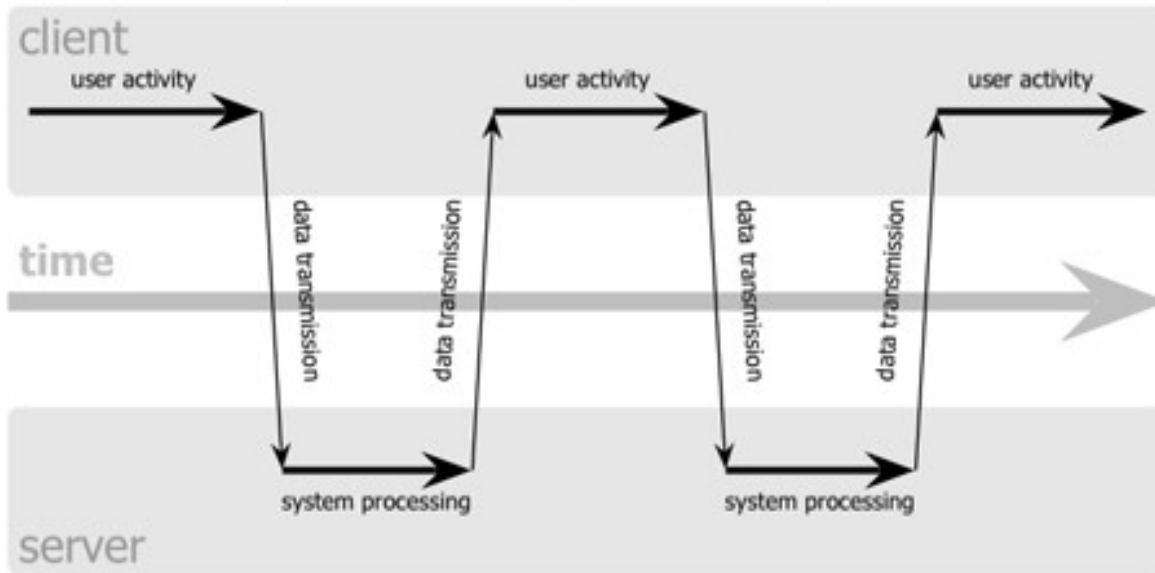


classic
web application model



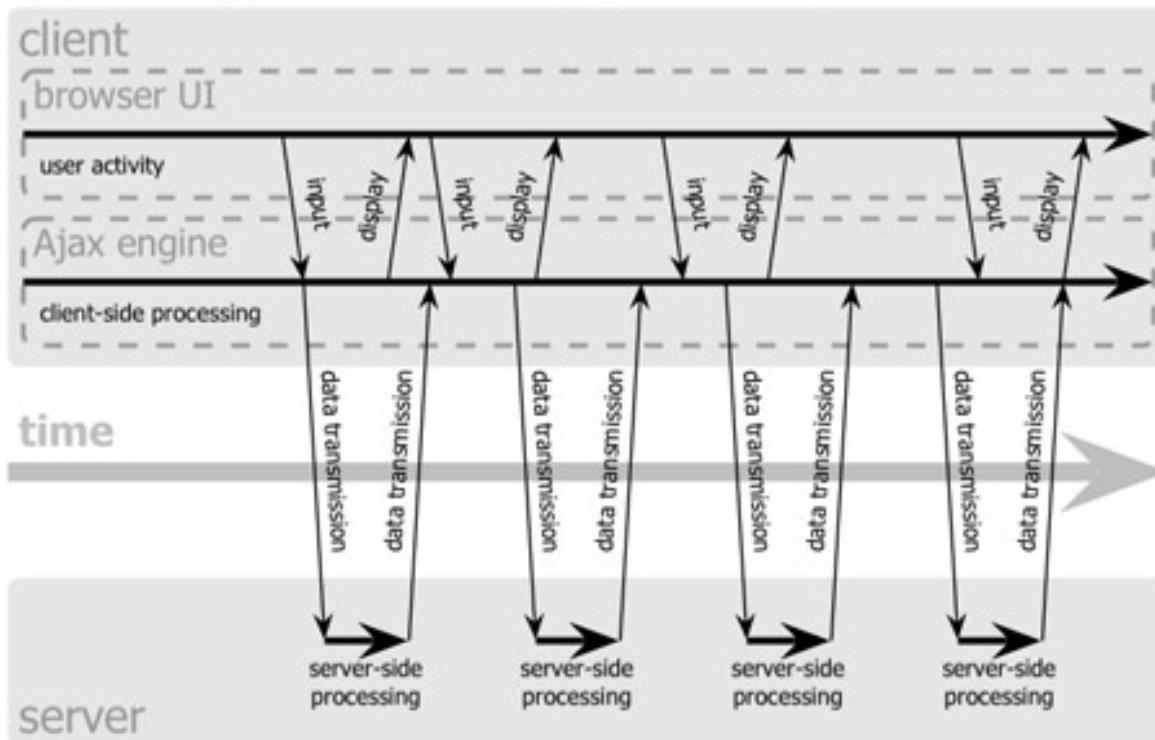
Ajax
web application model

classic web application model (synchronous)



Síncrono vs. Asíncrono

Ajax web application model (asynchronous)



Hacer un request de una URL

- Objeto `xmlHttpRequest`

1. Crear la URL

```
url = "http://server/personas"
```

2. Crear el objeto XMLHttpRequest

```
objXMLHttp=new XMLHttpRequest()
```

3. Indicar al browser el nombre de la función manejadora de la respuesta.

```
objXMLHttp.onreadystatechange = function() { ... }
```

4. Realizar la petición

```
objXMLHttp.open("GET", url, true);  
objXMLHttp.send();
```

XMLHttpRequest

En el Cliente

```
onclick="makeRequest()">
...
function makeRequest(){
xmlHttp = new XMLHttpRequest();
var url="servertime.php"
xmlHttp.onreadystatechange=actualizarD
ocumento;
    xmlHttp.open("GET",url,true)
    xmlHttp.send(null)
}
function actualizarDocumento(){
let data = xmlHttp.responseXML;
var nodo =
data.getElementsByTagName('time').item
(0);
alert(nodo.firstChild.data);
}
```

// En el servidor servertime.php

```
<?
// a time document
header("Content-type: text/xml");
print("<time>");
print( date('Y-m-d'));
print("</time>");
?>
```

Retorna

```
<?xml version="1.0" ?>
<time>
    2021-03-18.
</time>
```

JSON

//Si el servidor retorna un JSON

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021 },
    "phoneNumbers": [
      {
        "type": "home",
        "number": "212 555-1234"
      },
      {
        "type": "fax",
        "number": "646 555-4567"
      }
    ]
}
```

//Leer JSON en el cliente

```
var my_JSON_object = {};
var http_request = new XMLHttpRequest();
http_request.open("GET", url, true);
http_request.onreadystatechange = function ()
{
    var done = 4, ok = 200;
    if (http_request.readyState == done &&
        http_request.status == ok) {
        my_JSON_object =
        JSON.parse(http_request.responseText);
    }
};
http_request.send(null);
```

//Actualiza DOM con valores

```
My_JSON_object.firstname="John";
My_JSON_object.lastname="Smith";
My_JSON_object.age=25;
...
```

jQuery/AJAX

```
$.ajax({  
  type: "POST",  
  url: "example.php",  
  data: "name=John&location=Boston"  
}).done( function(msg) {  
  alert( "Data Saved: " + msg );  
}).fail( function( xmlHttpRequest, statusText, errorThrown ) {  
  alert(  
    "Your form submission failed.\n\n"  
    + "XML Http Request: " + JSON.stringify( xmlHttpRequest )  
    + ",\nStatus Text: " + statusText  
    + ",\nError Thrown: " + errorThrown );  
});
```


Métodos jQuery AJAX

```
<script  
  src="https://code.jquery.com/jquery-3.6.0.js"  
  integrity="sha256-H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk="  
  crossorigin="anonymous"  
></script>
```

// Cargar el contenido de una página HTML en un elemento

```
$("div#noticias").load("noticias.html");
```

// Descargar un script cliente desde el servidor y ejecutarlo

```
$.getScript("/ruta/hasta/miScript.js");
```

// Petición GET al servidor con parámetros y función de respuesta

// La petición POST es idéntica, salvo que se cambia \$.get por \$.post

```
$.get("/ruta/hasta/el/scriptDelServidor.php",{
```

```
  idProducto: "AX00342", cantidad: "3" },
```

```
function(data){  alert("Se ha añadido al carrito: " + data);  });
```

Métodos jQuery AJAX

- JQuery: Librería poderosa de Javascript
 - Simplificar tareas comunes de javascript
 - Acceder a “partes” de una página
 - Usando CSS o XPath
 - Modificar la apariencia de una página
 - Generar fácilmente peq. animaciones
 - Soporte simplificado para Ajax
 - Validación de formularios mas sencilla

Fetch

- Provee un método global `fetch()` para obtener recursos de forma asíncrona por http.
- Fetch proporciona una alternativa mejor a XMLHttpRequest

```
fetch('http://example.com/movies.json')  
  .then(response => response.json())  
  .then(data => console.log(data));
```

Fetch (requerir un recurso estático)

```
fetch('flores.jpg').then(function(response) {  
  if(response.ok) {  
    response.blob().then(function(miBlob) {  
      var objectURL = URL.createObjectURL(miBlob);  
      milmagen.src = objectURL;  
    });  
  } else {  
    console.log('Respuesta de red OK pero respuesta HTTP 404');  
  }  
})  
.catch(function(error) {  
  console.log('Hubo un problema con la petición Fetch:' + error.message);  
});
```

Fetch (datos JSON)

```
const URL =  
"https://api.github.com/users/MahekUnnisa/repos"
```

```
function listrepos() {  
  fetch(URL)  
    .then(result => result.json())  
    .then(data => console.log(data))  
}  
listrepos()
```

Fetch (enviar datos por POST)

```
const URL = ....
```

```
function PostData(url = "", object = {}) {
```

```
  fetch(url, {
```

```
    method: 'POST',
```

```
    headers: { 'Content-Type': 'application/json' },
```

```
    body: JSON.stringify(object)
```

```
  })
```

```
  .then(result => result.json())
```

```
  .then(data => console.log(data))
```

```
}
```

```
PostData(URL, { name: "fsaez" } )
```

Axios

- Axios es un cliente HTTP basado en promesas para node.js y el navegador.
- En servidor `$ npm install axios`
- En cliente `<script
src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>`

```
axios('http://example.com/movies.json')  
  .then(response => console.log(response.data));
```

API Axios

```
axios.post('/user/12345',  
  {  
    firstName: 'Alejandro',  
    lastName: 'Magno'  
  })  
  .then(function (response) {  
    console.log(response);  
  });
```


API Axios

```
axios({  
  method: 'post',  
  url: '/user'  
  data : {nombre:'jose', edad: 34},  
  headers: {'X-Requested-With': 'XMLHttpRequest'}  
  params: {ID:12345}  
})  
}).then(function (response) {  
  console.log(response);  
});
```

¿¿ALGUNA PREGUNTA??



GRACIAS