FORMACIÓN PROFESIONAL
MONTECASTELO

AJAX

Desarrollo Web en Entorno Cliente

Ciclo Superior de Desarrollo de Aplicaciones Web

2020/2021

Serialización de información

- El intercambio de información a través de la Web requiere que los datos sean debidamente serializados.
- La serialización es el proceso de convertir las estructuras de datos en un formato que pueda ser almacenado o transmitido de un modo independiente de lenguaje y entorno.

JSON

- JSON (*JavaScript Object Notation*) es un formato de serialización muy popular.
- Se trata de un formato ampliamente empleado como formato de almacenamiento y transmisión de la información en la Web, incluso en lenguajes de programación distintos de JavaScript.

JSON

- El formato JSON es similar al formato de los objetos de JavaScript, con las restricciones adicionales presentadas a continuación.

1. Todos los nombres de propiedad tienen que estar entre comillas dobles.
2. Solamente están permitidas expresiones simples de datos, sin funciones ni código que involucre cálculos.
3. No se permiten comentarios.

JSON

- Ejemplo de serialización JSON:

```
datos:{
“nombre": “Juan”,
“edad”: 25,
“titulado”: true,
“eventos”: [“trabajo", “comer",
   “dormir", “estudiar"]
}
```

JSON

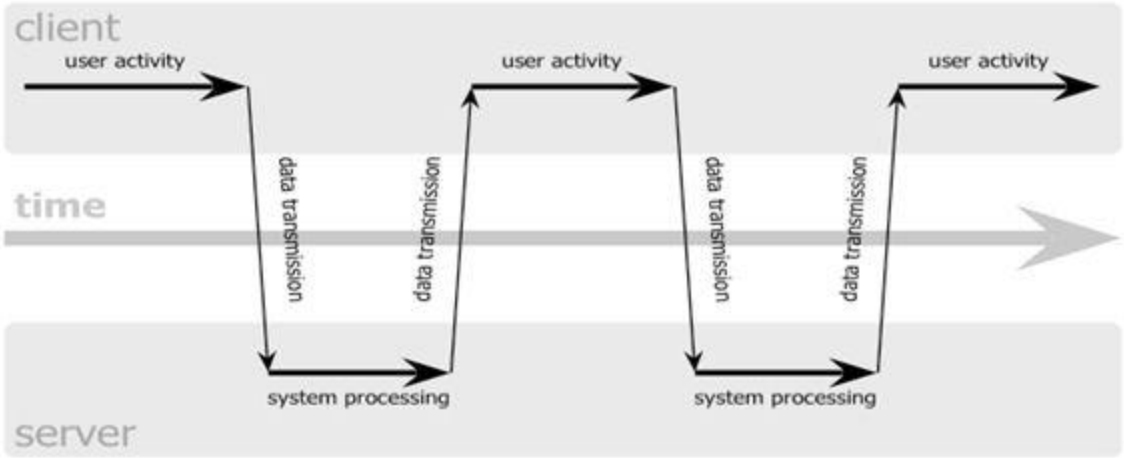
- JavaScript ofrecen los siguientes dos métodos para la gesitón de JSON.
- *JSON.stringify(obj)* → convierte un objeto Java Script en una cadena codificada en JSON.
- *JSON.parse(cadena)* → convierte una cadena codificada en JSON en un objeto JavaScript.

Modelo web tradicional

- Tradicionalmente, el modo de solicitar información desde una página web a un servidor web consiste en cambiar a otra página, o bien recargar la página actual.
- Esto supone que se realiza una nueva petición HTTP al servidor y la información devuelta se carga de nuevo en el navegador.

Modelo web tradicional

- En el instante de tiempo entre que se realiza la petición y se muestra la respuesta, el usuario no puede interactuar con el navegador.



AJAX

- AJAX (*Asynchronous JavaScript + XML*) es un conjunto de tecnologías que colaboran en el navegador para generar contenido web dinámicamente.
- El propósito de AJAX es que las aplicaciones web puedan enviar y recibir información de modo asíncrono (de fondo) sin interferir con la presentación y comportamiento de la página actual.

AJAX

- Aunque el formato XML se incluye en el nombre del término, en la práctica se emplea JSON para la serialización de la información.
- Así mismo, se emplean habitualmente los formatos HTML, JavaScript o texto plano como formato de la información intercambiada.

Modelo AJAX

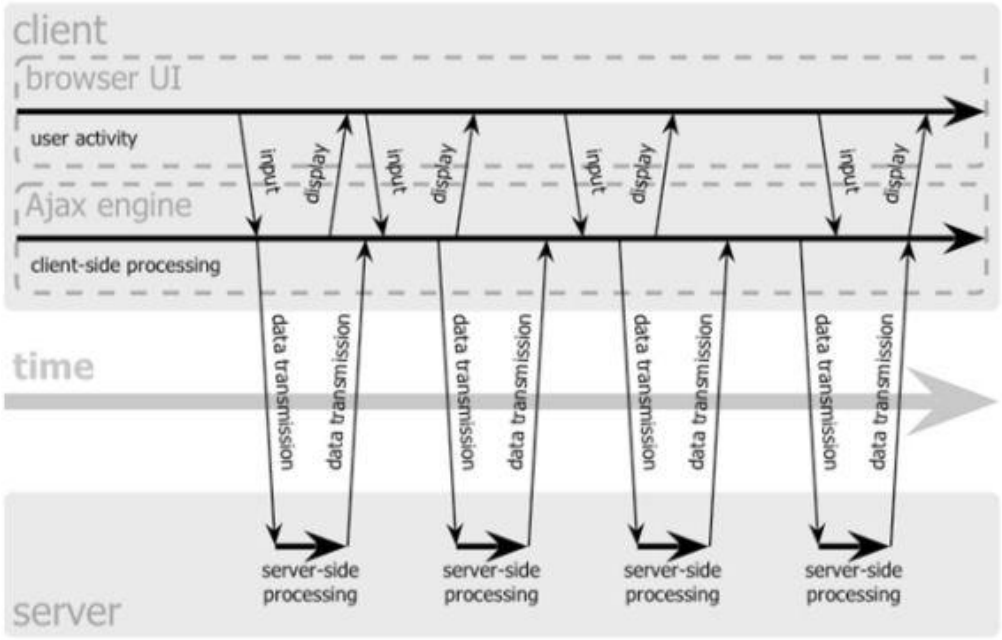
- AJAX se fundamenta en la realización de peticiones HTTP a través de JavaScript.
- El código realiza la petición y procesa la respuesta de manera transparente al navegador.
- De este modo se pueden actualizar fragmentos concretos de un documento HTML (a través del DOM) sin necesidad de recargar la página completa.

Modelo AJAX

- Por tanto AJAX permite a una web actualizarse continuamente sin tener que volver a cargar una página completa.
- Esto crea aplicaciones más rápidas y con mejor respuesta a las acciones del usuario.

Modelo AJAX

- Mediante AJAX las webs se actualizan dinámicamente sin afectar a la interacción del usuario.



AJAX con jQuery

- En la práctica, AJAX se implementa mediante código puro JavaScript, o mediante la ayuda de alguna biblioteca, como jQuery.
- jQuery incluye una serie de métodos que permiten implementar interacciones mediante AJAX en las aplicaciones web.

AJAX con jQuery

- El método $.ajax() contiene toda la funcionalidad AJAX de jQuery.
- Este método permite establecer, entre otros parámetros:
  - La URL a la que se envía la solicitud.
  - El método de HTTP empleado.
  - Los datos enviados al servidor.
  - El modo de procesar la respuesta del servidor.

AJAX con jQuery

- Además de $.ajax(), existen otras alternativas más especializadas que particularizan las interacciones AJAX para casos concretos.
- Estas alternativas realizan la interacción AJAX empleando un determinado método de HTTP y/o asumen que la información intercambiada posee un determinado formato: HTML, JSON, JavaScript…

AJAX con jQuery

- $.get(): carga de información mediante el método GET de HTTP.
- $.post(): carga de información mediante el método POST de HTTP.
- $.load(): carga un fragmento HTML mediante el método GET de HTTP y lo inserta en un contenedor.

AJAX con jQuery

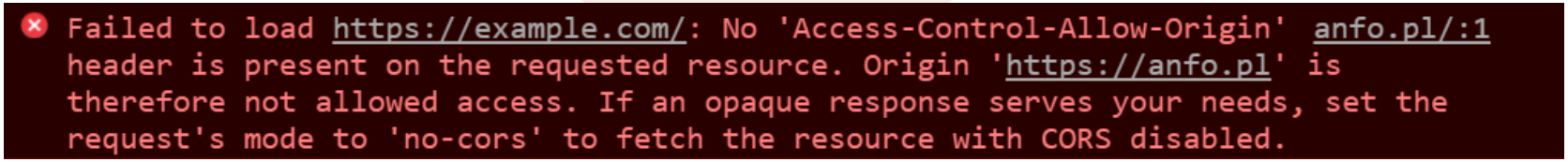
- $.getJSON(): carga un fragmento de información serializada en formato JSON mediante el método GET de HTTP.
- $.getScript(): carga un script de JavaScript mediante el método GET de HTTP y lo ejecuta.

CORS

- El concepto de CORS (*Cross-origin Resource Sharing*) hace referencia a un mecanismo de seguridad implementado en las interacciones web.
- CORS restringe peticiones de determinados recursos a un dominio distinto al dominio que realiza la petición.

CORS

- Una web puede solicitar recursos JavaScript, CSS, imágenes o vídeos a cualquier dominio.
- Sin embargo, las peticiones AJAX a un dominio distinto al que realiza la solicitud están restringidas por defecto y generan un error.



CORS

- Existen varios mecanismos para evitar el error CORS.
- Podemos evitarlo realizando las peticiones AJAX desde el dominio *localhost* y hacia el mismo dominio *localhost*.

CORS

- Debe tenerse en cuenta que las peticiones a URLs locales (formato *file:///*) generan el error CORS.
- Por tanto deberán realizarse las peticioens a un servidor web mediante URLs (formato *http://*).
- Por ejemplo:

http://localhost/dwec/ejemplo.html

Referencias

- JavaScript en w3schools
- JavaScript en Mozilla Developer Network

FORMACIÓN PROFESIONAL

MONTECASTELO