JSON

Importante

Recordemos el formato de una variable apta para ser convertida a JSON:

var objetoEnFormatoJSON = '{ "atributo": "valor", "atributo1": 1, "atributo2": [], "atributo3": null, "atributo4": false }';

Luego contamos con las funciones JSON.parse() y JSON.stringify()

1. Generar un objeto JSON que contenga únicamente una propiedad que sea un string.

Luego reconvertirlo a string.

2. Generar un objeto JSON amigos que contenga un número con la cantidad de amigos y un array con los strings de los nombres de mis amigos. Luego reconvertirlo a string.

3. Generar un objeto JSON que contenga un array con 3 objetos. Cada objeto debe tener un número y un string. Luego reconvertirlo a string.

a. Una vez que contamos con el objeto JSON probar acceder a cada una de sus variables.

AJAX

Importante: Para los ejercicios sobre AJAX utilizaremos URLs descriptas sobre el final de esta guía.

A continuación se propone un ejempo para el método getPaises

```
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
console.log(xmlhttp.responseText); console.log(JSON.parse(xmlhttp.responseText)); //Mi código }
}; xmlhttp.open("GET", "http://pilote.techo.org/admin/?do=api.getPaises", true);
xmlhttp.send();
```

Importante 2: Al hacer un request via POST agregar la línea

```
xmlhttp.setRequestHeader("Contenttype", "application/xwwwformurlencoded");
```

luego de llamar a open()

Recomendación: Al probar un método nuevo podemos hacer un html vacío que solamente pruebe llamar a dicho método para sentirnos cómodos a la hora de incluirlo en el producto final.

A continuación de una serie de ejercicios realizaremos un complejo formulario de registración para

TECHO. Se recomienda seguir estos ejercicios secuencialmente.

4. Generar un archivo HTML con un título de registración y un formulario que únicamente cuente con un <select> de paises. Este <select>, en principio, no tendrá ninguna opción.

5. Relacionamos el HTML con un archivo JS que al cargar el sitio haga un pedido mediante el método de getPaises. Imprimir en la consola el resultado de este pedido en texto y en un JSON.

6. Agregar un botón “siguiente” que al clickearlo desaparezca el <select> de países y aparezca el resto del formulario. Los <select> en principio estarán vacíos. Los campos restantes son:

a. Nombre

b. Apellido

c. Mail

d. Contraseña

e. Confirmar Contraseña

f. DNI

g. Sexo

h. Fecha de nacimiento

i. Celular

j. Provincia (<select>)

k. Sede en la que participará (<select>)

l. Ciudad (<select>)

7. Cargaremos los <select> de provincia y sede utilizando los métodos getUnidadesOrganizacionales y getRegiones.

8. Modificaremos nuestro código para que al modificar la provincia se actualice el <select> de ciudades utilizando getCiudades.

9. Agregaremos un botón de “finalizar” que utilice el método registrarPersona pero no es necesario enviarle datos. Probar que devuelve el pedido al enviarlo.

10. Mostrar los errores que pueda devolver registrarPersona mediante un alert.

a. opcional: En vez de mostrarlo con un alert, mostrarlo dentro de un en color rojo (que indique que es un error). Este cartel debe aparecer solo cuando hay errores y desaparecer al enviar el formulario

11. Modificar el pedido de registrarPersona para que envie los datos de nuestro formulario

12. En caso de que registrarPersona sea exitoso, ocultar el formulario y mostrarle al usuario un mensaje de éxito.

13. Opcional: Realizar la validación del formulario desde Javascript.

Métodos para utilizar vía AJAX

Método getPaises

- URL: http://pilote.techo.org/admin/?do=api.getPaises
- Método: GET
- Descripción: Devuelve los países disponibles en la base de datos.
- Parametros: Ninguno
- Valores de retorno:
  - "error": booleano indicando si hubo un error. false indica que no hubo errores
  - "contenido": objeto. Cada atributo es el nombre de un país mientras que el valor es el id que lo representa.

Método getUnidadesOrganizacionales

- URL: http://pilote.techo.org/admin/?do=api.getUnidadesOrganizacionales?idPais=[idPais]
- Método: GET
- Descripción: Devuelve las sedes disponibles en la base de datos para un país determinado
- Parametros: 1
  - idPais: numero. id del país.
- Valores de retorno:
  - "error": booleano indicando si hubo un error. false indica que no hubo errores
  - "contenido": objeto. Cada atributo es el nombre de una sede mientras que el valor es el id que lo representa.

Método getRegiones

- URL: http://pilote.techo.org/admin/?do=api.getRegiones?idPais=[idPais]
- Método: GET
- Descripción: Devuelve las provincias/regiones disponibles en la base de datos para un país determinado
- Parametros: 1
  - idPais: numero. id del país.
- Valores de retorno:
  - "error": booleano indicando si hubo un error. false indica que no hubo errores
  - "contenido": objeto. Cada atributo es el nombre de una provincia/región mientras que el valor es el id que lo representa.

Método getCiudades

- URL: http://pilote.techo.org/admin/?do=api.getCiudades?idRegionLT=[idRegionLT]

● Método: GET

● Descripción: Devuelve las ciudades disponibles en la base de datos para una provincia/región determinada

● Parametros: 1

○ idRegionLT: numero. id de la provincia/región.

● Valores de retorno:

○ “error”: booleano indicando si hubo un error. false indica que no hubo errores

○ “contenido”: objeto. Cada atributo es el nombre de una ciudad mientras que el valor es el id que lo representa.

Método registrarPersona

● URL: http://pilote.techo.org/admin/?do=api.registrarPersona

● Método: POST

● Descripción: Registra a una persona

● Parametros: 1

○ oficina: numero. id de la sede.

○ nombres: string. nombres de la persona.

○ apellido: string. apellido de la persona.

○ email: string. email de la persona.

○ password: string. contraseña de la persona.

○ cpassword: string. confirmación de la contraseña de la persona.

○ dni: string/número. DNI de la persona.

○ fechaNacimiento: string. Fecha de nacimiento de la persona en formato YYYYMMDD. El usuario debe ser mayor a 18 años.

○ sexo: string. Sexo de la persona. Se espera “M” o “F”.

○ celular: string/número. Celular de la persona.

○ nacionalidad: número. Nacionalidad de la persona. Se espera el id del país.

○ provincia: número. Provincia/región de la persona. Se espera el id de la provincia/región.

○ ciudad: número. Ciudad de la persona. Se espera el id de la ciudad.

● Valores de retorno:

○ “error”: booleano indicando si hubo un error. false indica que no hubo errores

○ “contenido”: array. Si error devuelve true, contenido tiene un array de strings con los errores que se encontraron.