

INTERACCIÓN CON EL SERVIDOR

Envío de parámetros y respuestas JSON

Aunque el formato XML está soportado por casi todos los lenguajes de programación, por muchas aplicaciones y es una tecnología madura y probada, en algunas ocasiones es más útil intercambiar información con el servidor en formato JSON.

JSON es un formato mucho más compacto y ligero que XML. Además, es mucho más fácil de procesar en el navegador del usuario. Afortunadamente, cada vez existen más utilidades para procesar y generar el formato JSON en los diferentes lenguajes de programación del servidor (PHP, Java, C#, etc.)

El ejemplo mostrado anteriormente para procesar las respuestas XML del servidor se puede reescribir utilizando respuestas JSON. En este caso, la respuesta que genera el servidor es mucho más concisa:

```
{
  mensaje: "...",
  parametros: {telefono: "...", codigo_postal: "...", fecha_nacimiento: "..."}
}
```

Considerando el nuevo formato de la respuesta, es necesario modificar la función que se encarga de procesar la respuesta del servidor:

```
function procesaRespuesta() {
  if(http_request.readyState == READY_STATE_COMPLETE) {
    if(http_request.status == 200) {
      var respuesta_json = http_request.responseText;
      var objeto_json = eval("(" + respuesta_json + ")");
      var mensaje = objeto_json.mensaje;
      var telefono = objeto_json.parametros.telefono;
      var fecha_nacimiento = objeto_json.parametros.fecha_nacimiento;
      var codigo_postal = objeto_json.parametros.codigo_postal;
```

```

        document.getElementById("respuesta").innerHTML = mensaje + "<br>" + "Fecha nacimiento = " + fecha_nacimiento + "<br>" + "Codigo postal = " + codigo_postal + "<br>" + "Telefono = " + telefono;
    }
}
}

```

La respuesta JSON del servidor se obtiene mediante la propiedad `responseText`:

```
var respuesta_json = http_request.responseText;
```

Sin embargo, esta propiedad solamente devuelve la respuesta del servidor en forma de cadena de texto. Para trabajar con el código JSON devuelto, se debe transformar esa cadena de texto en un objeto JSON. La forma más sencilla de realizar esa conversión es mediante la función `eval()`, en la que deben añadirse paréntesis al principio y al final para realizar la evaluación de forma correcta:

```
var objeto_json = eval("(" + respuesta_json + ")");
```

Una vez realizada la transformación, el objeto JSON ya permite acceder a sus métodos y propiedades mediante la notación de puntos tradicional. Comparado con las respuestas XML, este procedimiento permite acceder a la información devuelta por el servidor de forma mucho más simple:

```
// Con JSON
```

```
var fecha_nacimiento = objeto_json.parametros.fecha_nacimiento;
```

```
// Con XML
```

```
var parametros = root.getElementsByTagName("parametros")[0];
```

```
var fecha_nacimiento = parametros.getElementsByTagName("fecha_nacimiento")[0].firstChild.nodeValue;
```

EJERCICIO

Rehacer el ejercicio 2 para procesar respuestas del servidor en formato JSON.
Los cambios producidos son:

- 1) El script del servidor se llama `compruebaDisponibilidadJSON.php` y el parámetro que contiene el nombre se llama `login`.
- 2) La respuesta del servidor es un objeto JSON con la siguiente estructura:

El nombre de usuario está libre:

```
{ disponible: "si" }
```

El nombre de usuario está ocupado:

```
{ disponible: "no", alternativas: ["...", "...", ..., "..."] }
```