

Hoja de ejercicios Tema 8

Servicios web, AJAX

Rich Internet Applications

Aplicaciones Web
Año 2017/2018
Grado en Ing. Software
Fdi - UCM

▷ 1. Supongamos una variable `currencyRates` que contiene la siguiente información:

```
var currencyRates = {  
  "USD": 1.00000,    // Dólar estadounidense  
  "EUR": 0.94466,    // Euro  
  "GBP": 0.82314,    // Libra  
  "INR": 68.1762,    // Rupia india  
  "AUD": 1.35989,    // Dólar australiano  
  "CAD": 1.32303,    // Dólar canadiense  
  "ZAR": 13.6627,    // Rand sudafricano  
  "NZD": 1.42890,    // Dólar neozelandés  
  "JPY": 115.933     // Yen japonés  
};
```

Las propiedades de este objeto son las distintas denominaciones de monedas (EUR - Euros, USD - dólares estadounidenses, etc.). Cada una de ellas tiene como valor el factor de conversión entre dólares estadounidenses (USD) y la moneda correspondiente. Por ejemplo, 1 dólar estadounidense equivale a 1,42890 dólares neozelandeses, a 0.94466 euros y a 0.82314 libras.

Utilizando esta información puede realizarse la conversión entre dos tipos de monedas cualesquiera contenidos dentro de este objeto. Por ejemplo, para convertir 25 libras (GBP) a euros (EUR), convertimos primero las libras a dolares (USD) utilizando el factor de conversión inverso,

$$25 * \frac{1}{0,82314} = 30,371504 \text{ USD}$$

para luego realizar la conversión entre dólares y euros,

$$30,371504 * 0,94466 = 28,690745 \text{ EUR.}$$

El objetivo de este ejercicio es desarrollar una pequeña aplicación de conversión de divisas. Para ello:

(a). Desarrollar un servidor web que ofrezca los siguientes servicios:

Servicio 1: Obtener los distintos tipos de moneda disponibles.

Método: GET

URL: /currencies

Parámetros de entrada: Ninguno.

Códigos de respuesta: 200 (OK) o 500 si hubo error.

Tipos resultado: JSON.

Resultado: Lista con las denominaciones de las distintas divisas contenidas en el objeto `currencyRates`. Debe ser una lista de cadenas como, por ejemplo, ["USD", "EUR", "GBP", ...].

Servicio 2: Convertir una cantidad entre dos divisas:

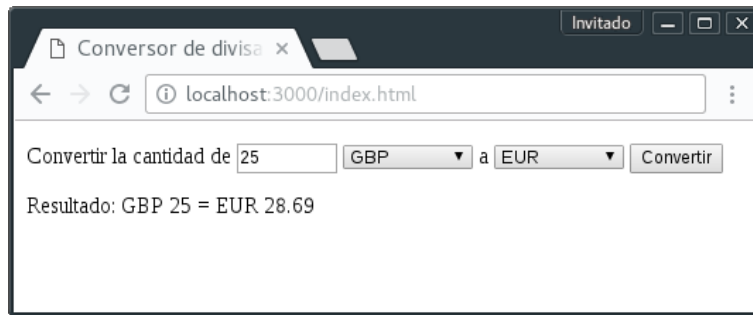


Figura 1: Ventana de la aplicación de conversión de divisas.

Método: GET

URL: /currency

Parámetros de entrada: Nombre de divisa origen (**from**), divisa destino (**to**) y cantidad a convertir (**quantity**). Estos tres parámetros van incluidos en la propia URL, como una *query string*.

Códigos de respuesta: 200 (OK), 400 (Bad request) si se introdujo algún parámetro incorrectamente, o 500 si hubo cualquier otro tipo de error.

Tipos resultado: JSON.

Resultado: Objeto con dos atributos: **currency**, con la denominación de la divisa destino y **result** con la cantidad resultante en la divisa destino.

Por ejemplo, la llamada /currency?from=GBP&to=EUR&quantity=25 debe devolver lo siguiente:

```
{
  "currency": "EUR",
  "result": 28.69074519522803
}
```

- (b). Diseña una página web que utilice estos servicios mediante AJAX. La página web tendrá dos *combobox* que deberán completarse con los tipos de divisas disponibles. Utiliza para ello el Servicio 1 del apartado anterior. Una vez introducida una cantidad y seleccionadas una divisa origen y una destino, deberá mostrarse el resultado de la conversión, tal y como se muestra en la Figura 1.

▷ 2. Implementa un servidor que proporcione dos servicios:

- Devolver un listado con los nombres de las comunidades autónomas de España.
- Dada una comunidad autónoma pasada como parámetro, devolver un listado con los nombres provincias de dicha comunidad.

Una vez hecho esto, implementa una página web que realice asincrónicamente llamadas a ambos servicios. La página mostrará una lista desplegable (**<select>**) de la que el usuario deberá seleccionar una comunidad autónoma. Si la comunidad seleccionada tiene más de una provincia, se mostrará una segunda lista desplegable con las provincias de dicha comunidad.

▷ 3. **Examen febrero 2017**

Supongamos un juego en línea que almacena un array con las puntuaciones obtenidas por cada jugador, ordenadas de mayor a menor puntuación:



Figura 2: Tabla de puntuaciones más altas

```
var records = [
  { nombre: "Fran", puntos: 955 },
  { nombre: "Rafael", puntos: 865 },
  { nombre: "Carmen", puntos: 563 },
  { nombre: "Rosario", puntos: 534 },
  { nombre: "Juan", puntos: 234 },
  { nombre: "Estela", puntos: 107 },
  ...
];
```

- (a). Implementa un servicio web que devuelva un JSON con los cinco primeros objetos (es decir, los de más puntuación) del array `records`. En caso de que el array contenga menos de cinco objetos, se devolverá todo el array.

Método: GET

URL: `/highestRecords`

Parámetros de entrada: Ninguno

Código de respuesta: 200 (OK).

Tipo resultado: JSON con un array de objetos. Cada uno de ellos tiene dos propiedades: `nombre` y `puntos`.

Resultado: Devuelve las cinco personas que han obtenido una puntuación más alta. Para cada una de ellas, se devuelve su nombre y su puntuación.

- (b). Supongamos una página como la de la Figura 2, en la que se muestra una tabla con las cinco mejores puntuaciones. Escribe una función `actualizarLista()` para que, mediante `jQuery` y una petición AJAX al servicio implementado en el apartado anterior, actualice dicha tabla con los datos devueltos por la petición AJAX.

Indicación: La sentencia `$(selector).empty();` elimina del DOM todos los hijos que se encuentren debajo del elemento indicado por el `selector` dado.



Figura 3: Carro de la compra.

- (c). Suponemos que añadimos a la página anterior un formulario invitando al usuario introducir un nombre, junto con un botón [Enviar]. Modifica la aplicación para que, cuando se pulse dicho botón, se añada el usuario a la tabla de records con una puntuación aleatoria entre 1 y 1000 y actualice la tabla de records. Para ello puedes suponer implementado el siguiente servicio:

Método: POST

URL: /newRecord

Parámetros de entrada: En el cuerpo de la petición POST, un objeto JSON con un único atributo (llamado "nombre") que contenga el nombre introducido en el formulario.

Código de respuesta: 201 (Created).

Tipo resultado: Ninguno.

Resultado: Ninguno.

► 4. Examen septiembre 2017

Partimos de una página como la de la Figura 3. Esta página contiene dos elementos `<div>` que contienen, respectivamente, una lista de productos disponibles (izquierda) y un carro de la compra inicialmente vacío (derecha).

```
<div id="stockD">
  <ul id="stock">
    <li>Aceite</li>
    <li>Huevos</li>
    ...
  </ul>
</div>
<div id="carroD">
  <ul id="carro">
  </ul>
</div>
```

- (a). Utilizando *jQuery* añade el manejador de eventos necesario para que al hacer clic en uno de los `` de la lista de productos disponibles, se elimine el elemento seleccionado de la lista de productos disponibles y se añada a la lista del carro de la compra.

- (b). Modifica la página anterior para que la lista inicial de productos disponibles se obtenga mediante una petición AJAX al servidor mediante la ruta `/stock`. Implementa dicha ruta en el servidor suponiendo que los productos están almacenados en el siguiente array:

```
var stock = ["Aceite", "Galletas", "Pan", "Cebolla", "Cereales", "Huevos",  
            "Tomate", "Mantequilla", "Leche", "Café"];
```

Tras realizar la petición AJAX, se debe rellenar el `` de la lista de productos disponibles a partir del resultado devuelto por el servidor.