

Ajax

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad Rey Juan Carlos

gsync-profes (arroba) gsync.urjc.es

Abril de 2018



©2018 GSyC
Algunos derechos reservados.
Este trabajo se distribuye bajo la licencia
Creative Commons Attribution Share-Alike 4.0

Es un conjunto de técnicas para enviar información asíncrona entre servidor web y cliente

- Aunque hay antecedentes desde 1996, el Ajax actual lo crea google en 2004, para gmail y google maps
- Evita que el usuario necesite pulsar *enviar*. La aplicación web recibe información continuamente, resultando una experiencia de usuario similar a la de una aplicación de escritorio
- Originalmente significaba Asynchronous JavaScript and XML, pero cuando empieza a usarse sin JavaScript y sin XML, el acrónimo AJAX se abandona para usar la palabra Ajax

Funcionamiento de Ajax

- 1 El usuario solicita una URL desde su navegador
- 2 El servidor web devuelve la página, que contiene un script
- 3 El navegador presenta la página y ejecuta el script
- 4 El script hace peticiones HTTP asíncronas a una URI del servidor, sin que el usuario intervenga. El servidor suele ser RESTful
- 5 El script interpreta las respuestas HTTP y modifica el HTML, sin que el usuario intervenga

Ajax con jQuery

jQuery facilita mucho el uso de Ajax. El método principal es `ajax()`

Recibe:

- Un *plain object* que puede tener varios atributos. El principal es `url`, la dirección del servicio

Devuelve:

- Un objeto de tipo `jqXHR`, que tiene varios atributos. Los principales son:
 - Un método llamado `done` al que le pasamos la función que se invocará si la petición tiene éxito. Tendrá un parámetro, que en nuestro caso será un objeto JSON
 - Un método llamado `fail` al que le pasamos la función a invocar en caso de error

El siguiente ejemplo llama a `http://fixer.io`, un servicio que ofrece tipos de cambio de divisas

- Por motivos de seguridad, para cumplir con la *same-origin policy*, solo podemos cargar la página desde el disco duro local, no desde el web
- Si este servicio ofreciera respuestas en JSONP, sí podríamos usarlo normalmente. jQuery se ocupa del manejo de JSONP, resulta transparente para el programador
- Otra solución para los sitios sin soporte de JSONP (que son muchos), sería un proxy en el mismo sitio web que sirve el HTML

```
$(document).ready(function() {  
  let urlServicio = 'http://data.fixer.io/latest';  
  petition = $.ajax({  
    url: urlServicio,  
    data: {  
      access_key: "xxxxxx",  
      symbols: "USD, GBP"  
    }  
  })  
  petition.done(manajeRespuesta);  
  petition.fail(manajeError);  
  
  function manejaRespuesta(json) {  
    $("#div01").text(JSON.stringify(json));  
  };  
  
  function manejaError(jqXHR) {  
    $("#div01").text("Error: " + jqXHR.status);  
  };  
});
```

Cambio de divisas

- Cada divisa tiene un código ISO 4217, que es un identificador de 3 letras mayúsculas. Por ejemplo USD (United States Dollar) para el dólar, EUR para el euro, GBP para la libra esterlina, etc
- El precio de una divisa frente a otra se indica mediante la concatenación de sus dos códigos ISO 4217. La primera se denomina *divisa base* y la segunda, *divisa cotizada*
- Ejemplo EURUSD = 1.13
Aquí la divisa base es el euro, y la cotizada, el dólar. Este valor (1.13) indica cuántas unidades de la divisa cotizada hacen falta para comprar una unidad de la divisa base
- Dicho de otro modo, la primera moneda es la que queremos y la segunda, la que tenemos, la que usamos para pagar.

setInterval

Para que la consulta Ajax se repita periódicamente, podemos usar la función `setInterval()`

Recibe

- Como primer argumento, una función
- Como segundo argumento, un intervalo de tiempo en milisegundos

Como resultado, evalúa la función periódicamente, con el intervalo especificado. El siguiente ejemplo se ejecutaría cada 60 segundos

```
setInterval(function() {  
    miTexto=actualizaTexto();  
    $("#p01").text(miTexto);  
},  
60000  
);
```