

Programación AJAX en JS

Módulo: **Desarrollo Web en entorno cliente**

CFGs Desarrollo de Aplicaciones Web

Métodos Ajax de jQuery

- Método: `$.ajax`
- Documentación: <http://api.jquery.com/jQuery.ajax/>

Algunos métodos (ver la documentación para el resto)

- **Async:** Establece si la petición es asíncrona o no. Por defecto es true.
- **Cache:** Establece si la petición es cacheada en el navegador. Por defecto es true salvo cuando el dataType está establecido a “script” o “jsonp”. Cuando posee el valor false, se agrega una cadena de caracteres al final de la URL de la petición para evitar la caché.
- **Complete:** Establece una función de devolución de llamada que se ejecuta cuando la petición está completa, aunque haya errores. La función recibe como argumentos el objeto jqXHR y un texto especificando el estatus de la misma petición (success, notmodified, error, timeout, abort o parsererror).
- **Data:** Establece la información que se enviará al servidor (objeto o cadena).
- **dataType:** Establece el tipo de información que se espera recibir como respuesta del servidor. Si no se especifica ningún valor jQuery revisa el tipo de MIME que posee la respuesta.

Métodos Ajax de jQuery

- **Error:** Establece una función de devolución de llamada a ejecutar si se produce algún error en la petición. Dicha función recibe como argumentos el objeto jqXHR, un texto especificando el estatus de la misma petición (timeout, error, abort, o parsererror) y un texto con la descripción del error que haya enviado el servidor.
- **Success:** Establece una función a ejecutar si la petición ha sido satisfactoria. Dicha función recibe como argumentos el objeto, un texto especificando el estatus de la misma petición y la información de la petición (convertida a objeto JS en el caso que dataType sea JSON), el estatus de la misma.
- **Timeout:** Establece un tiempo en milisegundos para considerar a una petición como fallada.
- **type:** De forma predeterminada su valor es “GET”. Puede ser POST o GET.
- **url:** Establece la URL en donde se realiza la petición.

Ejemplo 1

```
$.ajax({
  url : 'pagina1.php', //URL de la petición
  data : { id : 123 }, //información a enviar, puede ser una cadena
  type : 'GET',        //tipo de la petición: POST o GET
  dataType : 'json',   //tipo de dato que se espera
  success : function(json) { //función a ejecutar si es satisfactoria
    $('<h1/>').text(json.title).appendTo('body');
    $('<div class="content"/>')
      .html(json.html).appendTo('body');
  },
  error : function(jqXHR, status, error) { //función error
    alert('Disculpe, existió un problema');
  },
  // función a ejecutar sin importar si la petición falló o no
  complete : function(jqXHR, status) {
    alert('Petición realizada');
  }
});
```

Nota: Si el servidor devuelve información que es diferente al formato especificado en `dataType` se producirá un error. Es importante especificar correctamente el tipo de datos que se espera recibir del servidor (Content-type).

Eventos Ajax

jQuery provee la posibilidad de vincular eventos Ajax a elementos seleccionados, por ejemplo, mostrar u ocultar un texto o imagen, cuando comienza o termina la petición.

Existen eventos globales asociados al document y eventos locales asociados a un objeto concreto de Ajax.

```
//Eventos globales
```

```
$('#loading_indicator')  
  .ajaxStart(function(){$(this).show(); })  
  .ajaxStop(function(){$(this).hide(); });
```

```
// Eventos locales
```

```
$.ajax({  
  beforeSend: function(){  
    // Código antes de enviar  
  },  
  complete: function(){  
    // Código cuando está completa  
  }  
  // .....  
});
```

Consultar documentación oficial: http://docs.jquery.com/Ajax_Events.

Otros métodos

Cuando no necesitamos controlar los errores existen otros métodos más adecuados para realizar las peticiones Ajax. Realmente son métodos basados en \$.ajax pero con valores preestablecidos para algunos parámetros.

\$.get: realiza una petición GET de la URL especificada

\$.post: realiza una petición POST de la URL especificada

\$.getScript: añade un script a la página

\$.getJSON: realiza una petición GET a la URL especificada y espera un dato JSON

Argumentos de los métodos:

- **url** (obligatorio): url de la petición
- **data** (opcional): información que se remite al servidor (objeto o cadena)
- **success callback** (opcional): una función que se ejecutará en caso de que la petición haya sido satisfactoria. Tiene como argumentos la información de la petición y el objeto en bruto de dicha petición.
- **Data type** (opcional): tipo de datos que se espera recibir del servidor.

Ejemplo 2

```
// Obtiene texto plano
$.get('/pagina1.php', { clave : 5213 }, function(respuesta) {
    console.log(respuesta);
});

// Añade un script a la página y luego ejecuta una función
$.getScript('js/app.js', function() {
    iniciar();
});

// Obtiene información en formato JSON desde el servidor
$.getJSON('pagina2.php', function(respuesta) {
    $.each(respuesta, function(k, v) {
        console.log(k + ' : ' + v);
    });
});
```