

Ambientes no propietarios

Ajax

Edwin Salvador

30 de junio de 2015

Sesión 13

Contenido I

1 ¿Qué es AJAX?

2 ¿Cómo funciona?

3 El objeto XMLHttpRequest

4 Ejemplo

5 jQuery para Ajax

¿Qué es AJAX?

- AJAX = Asynchronous JavaScript and XML.

¿Qué es AJAX?

- AJAX = Asynchronous JavaScript and XML.
- No solo XML si no también JSON (principalmente).

¿Qué es AJAX?

- AJAX = Asynchronous JavaScript and XML.
- No solo XML si no también JSON (principalmente).
- Procesamiento asíncrono = se pueden hacer otras cosas mientras el explorador está en espera de datos.

¿Qué es AJAX?

- AJAX = Asynchronous JavaScript and XML.
- No solo XML si no también JSON (principalmente).
- Procesamiento asíncrono = se pueden hacer otras cosas mientras el explorador está en espera de datos.
- Mejora la experiencia del usuario.

¿Qué es AJAX?

- AJAX = Asynchronous JavaScript and XML.
- No solo XML si no también JSON (principalmente).
- Procesamiento asíncrono = se pueden hacer otras cosas mientras el explorador está en espera de datos.
- Mejora la experiencia del usuario.
- Normalmente JS es síncrono (sin bloqueo), es decir mientras se ejecuta el código no se puede realizar nada más. Ejemplo: la carga de scripts.

¿Qué es AJAX?

- AJAX = Asynchronous JavaScript and XML.
- No solo XML si no también JSON (principalmente).
- Procesamiento asíncrono = se pueden hacer otras cosas mientras el explorador está en espera de datos.
- Mejora la experiencia del usuario.
- Normalmente JS es síncrono (sin bloqueo), es decir mientras se ejecuta el código no se puede realizar nada más. Ejemplo: la carga de scripts.
- El explorador no espera los datos del servidor si no que puede seguir ejecutándose.

¿Qué es AJAX?

- AJAX = Asynchronous JavaScript and XML.
- No solo XML si no también JSON (principalmente).
- Procesamiento asíncrono = se pueden hacer otras cosas mientras el explorador está en espera de datos.
- Mejora la experiencia del usuario.
- Normalmente JS es síncrono (sin bloqueo), es decir mientras se ejecuta el código no se puede realizar nada más. Ejemplo: la carga de scripts.
- El explorador no espera los datos del servidor si no que puede seguir ejecutándose.
- Antes para realizar un cambio en la página debíamos volver a cargar toda la página, con AJAX podemos cargar solo la sección que queremos cambiar.

¿Qué es AJAX?

- AJAX = Asynchronous JavaScript and XML.
- No solo XML si no también JSON (principalmente).
- Procesamiento asíncrono = se pueden hacer otras cosas mientras el explorador está en espera de datos.
- Mejora la experiencia del usuario.
- Normalmente JS es síncrono (sin bloqueo), es decir mientras se ejecuta el código no se puede realizar nada más. Ejemplo: la carga de scripts.
- El explorador no espera los datos del servidor si no que puede seguir ejecutándose.
- Antes para realizar un cambio en la página debíamos volver a cargar toda la página, con AJAX podemos cargar solo la sección que queremos cambiar.
- AJAX se ejecuta generalmente después de algún evento del usuario (clic, envío de formulario, etc)

Contenido I

1 ¿Qué es AJAX?

2 ¿Cómo funciona?

3 El objeto XMLHttpRequest

4 Ejemplo

5 jQuery para Ajax

¿Cómo funciona?

Podemos dividirlo en tres pasos:

- **La petición (Request)** El navegador hace una petición al servidor incluyendo la información necesaria (datos del formulario, etc). Los navegadores implementan un objeto **XMLHttpRequest** para manejar peticiones AJAX. Esta petición se envía y el navegador sigue trabajando.

¿Cómo funciona?

Podemos dividirlo en tres pasos:

- **La petición (Request)** El navegador hace una petición al servidor incluyendo la información necesaria (datos del formulario, etc). Los navegadores implementan un objeto **XMLHttpRequest** para manejar peticiones AJAX. Esta petición se envía y el navegador sigue trabajando.
- **Trabajo en el servidor** El servidor recibe la petición y ejecuta el código correspondiente. Esto no es parte de lo que es AJAX (es simplemente trabajo del servidor). El servidor nos responde con datos (HTML, XML o **JSON**).

¿Cómo funciona?

Podemos dividirlo en tres pasos:

- **La petición (Request)** El navegador hace una petición al servidor incluyendo la información necesaria (datos del formulario, etc). Los navegadores implementan un objeto **XMLHttpRequest** para manejar peticiones AJAX. Esta petición se envía y el navegador sigue trabajando.
- **Trabajo en el servidor** El servidor recibe la petición y ejecuta el código correspondiente. Esto no es parte de lo que es AJAX (es simplemente trabajo del servidor). El servidor nos responde con datos (HTML, XML o **JSON**).
- **La respuesta (Response)** Cuando el servidor ha terminado de responder se lanza un evento que es procesado en una función JS para recibir el contenido, procesarlo e incluirlo en la página o realizar su trabajo.

Ejemplo

```
<script>
function ajaxRequest() {
    try { // No IE?
        var request = new XMLHttpRequest()
    } catch(e1) {
        try { // IE 6+?
            request = new ActiveXObject("Msxml2.XMLHTTP")
        } catch(e2) {
            try{ // IE 5?
                request = new ActiveXObject("Microsoft.XMLHTTP")
            } catch(e3) { // There is no AJAX Support
                request = false
            }
        }
    }
    return request
}
</script>
```

Antiguas versiones de IE no implementaban el objeto XMLHttpRequest.

Contenido I

- 1 ¿Qué es AJAX?
- 2 ¿Cómo funciona?
- 3 El objeto XMLHttpRequest
- 4 Ejemplo
- 5 jQuery para Ajax

Propiedades

Property	Description
<code>onreadystatechange</code>	Specifies an event-handling function to be called whenever the <code>readyState</code> property of an object changes.
<code>readyState</code>	An integer property that reports on the status of a request. It can have any of these values: 0 = Uninitialized, 1 = Loading, 2 = Loaded, 3 = Interactive, and 4 = Completed.
<code>responseText</code>	The data returned by the server in text format.
<code>responseXML</code>	The data returned by the server in XML format.
<code>status</code>	The HTTP status code returned by the server.
<code>statusText</code>	The HTTP status text returned by the server.

Method	Description
<code>abort()</code>	Aborts the current request.
<code>getAllResponseHeaders()</code>	Returns all headers as a string.
<code>getResponseHeader(<i>param</i>)</code>	Returns the value of <i>param</i> as a string.
<code>open('method', 'url', 'asynch')</code>	Specifies the HTTP method to use (Get or Post), the target URL, and whether the request should be handled asynchronously (<code>true</code> or <code>false</code>).
<code>send(<i>data</i>)</code>	Sends <i>data</i> to the target server using the specified HTTP method.
<code>setRequestHeader('param', 'value')</code>	Sets a header with a parameter/value pair.

Contenido I

- 1 ¿Qué es AJAX?
- 2 ¿Cómo funciona?
- 3 El objeto XMLHttpRequest
- 4 Ejemplo**
- 5 jQuery para Ajax

Ejemplo

`http://localhost/anp/ejemplos/s13/urlpost.html`

`...`

`...`

`...`

`// En caso de querer hacerlo con el metodo GET`

`nocache = "&nocache=" + Math.random() * 1000000`

`request = new ajaxRequest()`

`request.open("GET", "urlget.php?url=amazon.com/gp/aw" +
 nocache, true)`

`...`

`...`

`...`

`request.send(null)`

`...`

`...`

`...`

Contenido I

1 ¿Qué es AJAX?

2 ¿Cómo funciona?

3 El objeto XMLHttpRequest

4 Ejemplo

5 jQuery para Ajax

- El ejemplo anterior no utiliza ningún frameworks para trabajar con AJAX, esto podría generar errores y falta de manejo de estándares.

jQuery para Ajax

- El ejemplo anterior no utiliza ningún frameworks para trabajar con AJAX, esto podría generar errores y falta de manejo de estándares.
- Utilizaremos **jQuery** para manejar las llamadas AJAX.

- El ejemplo anterior no utiliza ningún frameworks para trabajar con AJAX, esto podría generar errores y falta de manejo de estándares.
- Utilizaremos **jQuery** para manejar las llamadas AJAX.
- **Sugerencia:** Si usan SublimeText pueden instalar el plugin de jQuery para autocompletar código. Necesitan instalar el **Package Control** (<https://packagecontrol.io/installation>)

- El ejemplo anterior no utiliza ningún frameworks para trabajar con AJAX, esto podría generar errores y falta de manejo de estándares.
- Utilizaremos **jQuery** para manejar las llamadas AJAX.
- **Sugerencia:** Si usan SublimeText pueden instalar el plugin de jQuery para autocompletar código. Necesitan instalar el **Package Control** (<https://packagecontrol.io/installation>)
- Veamos un ejemplo.

Ejemplo

- Realizar una página que contenga un link de contacto que abra una ventana modal de bootstrap que contenga un formulario con: nombre (texto), email (texto), mensaje (textarea), noticias (checkbox).

Ejemplo

- Realizar una página que contenga un link de contacto que abra una ventana modal de bootstrap que contenga un formulario con: nombre (texto), email (texto), mensaje (textarea), noticias (checkbox).
- El formulario debe enviarse a través de AJAX para ser guardado en un tabla mensajes.

Ejemplo

- Realizar una página que contenga un link de contacto que abra una ventana modal de bootstrap que contenga un formulario con: nombre (texto), email (texto), mensaje (textarea), noticias (checkbox).
- El formulario debe enviarse a través de AJAX para ser guardado en una tabla mensajes.
- Utilizar el plugin jQuery.validation (<http://jqueryvalidation.org/>) para validar el formulario antes de ser enviado al servidor.

Ejemplo

- Realizar una página que contenga un link de contacto que abra una ventana modal de bootstrap que contenga un formulario con: nombre (texto), email (texto), mensaje (textarea), noticias (checkbox).
- El formulario debe enviarse a través de AJAX para ser guardado en un tabla mensajes.
- Utilizar el plugin jQuery.validation (<http://jqueryvalidation.org/>) para validar el formulario antes de ser enviado al servidor.
- Utilizar los estilos de bootstrap para realizar la página.

Ejemplo

- Realizar una página que contenga un link de contacto que abra una ventana modal de bootstrap que contenga un formulario con: nombre (texto), email (texto), mensaje (textarea), noticias (checkbox).
- El formulario debe enviarse a través de AJAX para ser guardado en un tabla mensajes.
- Utilizar el plugin jQuery.validation (<http://jqueryvalidation.org/>) para validar el formulario antes de ser enviado al servidor.
- Utilizar los estilos de bootstrap para realizar la página.
- El formulario debe vaciarse cada vez que se envían los datos o se cierra el form.

Ejemplo

- Realizar una página que contenga un link de contacto que abra una ventana modal de bootstrap que contenga un formulario con: nombre (texto), email (texto), mensaje (textarea), noticias (checkbox).
- El formulario debe enviarse a través de AJAX para ser guardado en una tabla mensajes.
- Utilizar el plugin jQuery.validation (<http://jqueryvalidation.org/>) para validar el formulario antes de ser enviado al servidor.
- Utilizar los estilos de bootstrap para realizar la página.
- El formulario debe vaciarse cada vez que se envían los datos o se cierra el form.
- Entrega: Viernes 3 de julio 23:59 pm

- Realizar un página de registro de usuarios que pida: nombre, email, teléfono, dirección, usuario, contraseña, verificar contraseña.

- Realizar un página de registro de usuarios que pida: nombre, email, teléfono, dirección, usuario, contraseña, verificar contraseña.
- Antes de permitir el registro del usuario se debe validar el formulario con jQuery.validation.

- Realizar un página de registro de usuarios que pida: nombre, email, teléfono, dirección, usuario, contraseña, verificar contraseña.
- Antes de permitir el registro del usuario se debe validar el formulario con jQuery.validation.
 - Todos los campos deben ser obligatorios menos el campo dirección.

- Realizar un página de registro de usuarios que pida: nombre, email, teléfono, dirección, usuario, contraseña, verificar contraseña.
- Antes de permitir el registro del usuario se debe validar el formulario con jQuery.validation.
 - Todos los campos deben ser obligatorios menos el campo dirección.
 - El email debe ser válido.

- Realizar un página de registro de usuarios que pida: nombre, email, teléfono, dirección, usuario, contraseña, verificar contraseña.
- Antes de permitir el registro del usuario se debe validar el formulario con jQuery.validation.
 - Todos los campos deben ser obligatorios menos el campo dirección.
 - El email debe ser válido.
 - La contraseña debe coincidir en los dos campos. (Revisar el método equalTo <http://jqueryvalidation.org/equalTo-method>)

- Realizar un página de registro de usuarios que pida: nombre, email, teléfono, dirección, usuario, contraseña, verificar contraseña.
- Antes de permitir el registro del usuario se debe validar el formulario con jQuery.validation.
 - Todos los campos deben ser obligatorios menos el campo dirección.
 - El email debe ser válido.
 - La contraseña debe coincidir en los dos campos. (Revisar el método equalTo <http://jqueryvalidation.org/equalTo-method>)
 - Verificar que el usuario ingresado no exista en la BDD. Si ya existe se debe mostrar un mensaje indicando que el usuario ya existe y debe cambiarlo para poder registrarse. Revisar el método remote (<http://jqueryvalidation.org/remote-method>), este método utiliza Ajax para realizar la validación remota contra la BDD. **Opcional** Esta validación normalmente se la hace apenas el usuario ingresa el texto en el campo antes de enviar el formulario.

- **Debido al feriado del 7 de julio:** Seguir el siguiente tutorial “Aprendiendo APIs: Login con Facebook SDK V4 en PHP” (<http://www.codejobs.biz/es/blog/2014/08/14/aprendiendo-apis-login-con-facebook-sdk-v4-en-php#sthash.MSKChRAT.oKBBYswQ.dpbs>) realizar un ejercicio propio que utilice el SDK de Facebook para realizar alguna acción.

- **Debido al feriado del 7 de julio:** Seguir el siguiente tutorial “Aprendiendo APIs: Login con Facebook SDK V4 en PHP” (<http://www.codejobs.biz/es/blog/2014/08/14/aprendiendo-apis-login-con-facebook-sdk-v4-en-php#sthash.MSKChRAT.oKBBYswQ.dpbs>) realizar un ejercicio propio que utilice el SDK de Facebook para realizar alguna acción.
- Ejemplo: Permitir el registro a si página a través de FB. O simplemente mostrar la lista de sus amigos en alguna página personalizada.

- **Debido al feriado del 7 de julio:** Seguir el siguiente tutorial “Aprendiendo APIs: Login con Facebook SDK V4 en PHP” (<http://www.codejobs.biz/es/blog/2014/08/14/aprendiendo-apis-login-con-facebook-sdk-v4-en-php#sthash.MSKChRAT.oKBBYSwQ.dpbs>) realizar un ejercicio propio que utilice el SDK de Facebook para realizar alguna acción.
- Ejemplo: Permitir el registro a si página a través de FB. O simplemente mostrar la lista de sus amigos en alguna página personalizada.
- **Entrega Martes 14 de julio 23:59**

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.

Deber

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.
- El proyecto debe tener:

Deber

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.
- El proyecto debe tener:
 - Utilización de PHP, HTML, CSS, JS, MySQL.

Deber

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.
- El proyecto debe tener:
 - Utilización de PHP, HTML, CSS, JS, MySQL.
 - Manejar directivas con el .htaccess

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.
- El proyecto debe tener:
 - Utilización de PHP, HTML, CSS, JS, MySQL.
 - Manejar directivas con el .htaccess
 - Manejar todos los archivos por separado, PHP, JS, CSS, etc.

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.
- El proyecto debe tener:
 - Utilización de PHP, HTML, CSS, JS, MySQL.
 - Manejar directivas con el .htaccess
 - Manejar todos los archivos por separado, PHP, JS, CSS, etc.
 - Incluir archivos de configuración general del sitio.

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.
- El proyecto debe tener:
 - Utilización de PHP, HTML, CSS, JS, MySQL.
 - Manejar directivas con el .htaccess
 - Manejar todos los archivos por separado, PHP, JS, CSS, etc.
 - Incluir archivos de configuración general del sitio.
 - Manejar estilos con Bootstrap y utilización de plugins (slideshow).

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.
- El proyecto debe tener:
 - Utilización de PHP, HTML, CSS, JS, MySQL.
 - Manejar directivas con el .htaccess
 - Manejar todos los archivos por separado, PHP, JS, CSS, etc.
 - Incluir archivos de configuración general del sitio.
 - Manejar estilos con Bootstrap y utilización de plugins (slideshow).
 - Manejar validaciones y manipulación del DOM con jQuery.

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.
- El proyecto debe tener:
 - Utilización de PHP, HTML, CSS, JS, MySQL.
 - Manejar directivas con el .htaccess
 - Manejar todos los archivos por separado, PHP, JS, CSS, etc.
 - Incluir archivos de configuración general del sitio.
 - Manejar estilos con Bootstrap y utilización de plugins (slideshow).
 - Manejar validaciones y manipulación del DOM con jQuery.
 - Interactuar con el servidor a través de AJAX para al menos una acción.

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.
- El proyecto debe tener:
 - Utilización de PHP, HTML, CSS, JS, MySQL.
 - Manejar directivas con el .htaccess
 - Manejar todos los archivos por separado, PHP, JS, CSS, etc.
 - Incluir archivos de configuración general del sitio.
 - Manejar estilos con Bootstrap y utilización de plugins (slideshow).
 - Manejar validaciones y manipulación del DOM con jQuery.
 - Interactuar con el servidor a través de AJAX para al menos una acción.
- Entrega del proyecto 28 de julio

- Trabajar en el proyecto aplicando todas las técnicas y tecnologías que hemos revisado hasta ahora.
- El proyecto debe tener:
 - Utilización de PHP, HTML, CSS, JS, MySQL.
 - Manejar directivas con el .htaccess
 - Manejar todos los archivos por separado, PHP, JS, CSS, etc.
 - Incluir archivos de configuración general del sitio.
 - Manejar estilos con Bootstrap y utilización de plugins (slideshow).
 - Manejar validaciones y manipulación del DOM con jQuery.
 - Interactuar con el servidor a través de AJAX para al menos una acción.
- Entrega del proyecto 28 de julio
- Examen 21 de julio