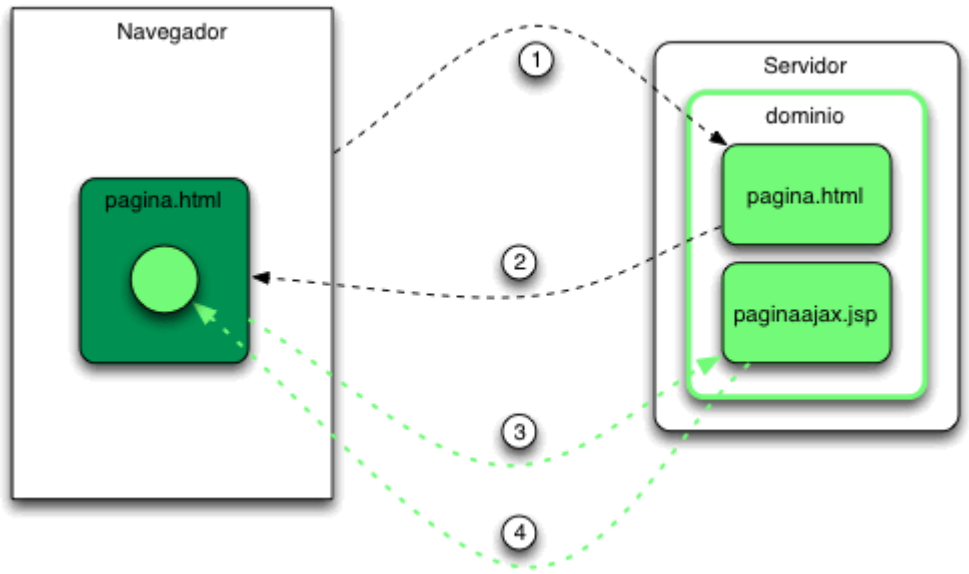
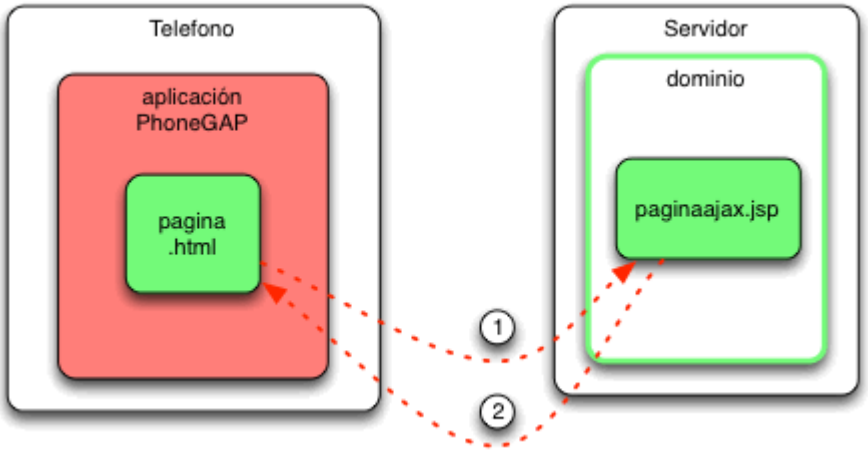
CORS (Cross Origin Resource Sharing) es una tecnología que cada día necesitamos utilizar más a la hora de desarrollar aplicaciones móviles y web. ¿Cómo funciona exactamente? . En una aplicación web clásica nosotros podemos cargar una página y solicitar que esta cargue dinámicamente datos vía AJAX.



Eso sí esos datos que vienen vía AJAX tienen que pertenecer al mismo dominio. En el caso de que esto no sea así la aplicación no podrá cargar los datos por limitaciones de seguridad. Los problemas comienzan cuando nosotros tenemos aplicaciones que necesitan acceder a esos datos pero no se encuentran bajo el mismo dominio. Un ejemplo muy habitual es una aplicación Movil empaquetada con PhoneGAP.



En este caso la petición no nos funcionará ya que hemos cargado una página HTML sin descargarla del servidor por lo tanto la página con datos AJAX que solicitamos no está evidentemente en el mismo dominio.

Utilizando CORS Java

Para solventar este problema tenemos que configurar el servidor para que algunas URL soporten invocaciones CORS (desde otros dominios). Imaginemonos que tenemos las siguientes páginas.

```
&amp;amp;lt;!DOCTYPE html&amp;amp;gt;
&amp;amp;lt;html&amp;amp;gt;
&amp;amp;lt;head&amp;amp;gt;
&amp;amp;lt;meta charset="UTF-8"&amp;amp;gt;
&amp;amp;lt;title&amp;amp;gt;Insert title
here&amp;amp;lt;/title&amp;amp;gt;
&amp;amp;lt;script type="text/javascript"
src="jquery-1.11.2.js"&amp;amp;gt;&amp;amp;lt;/script&amp;amp;gt;
&amp;amp;lt;script type="text/javascript"&amp;amp;gt;
```

```
$(document).ready(function() {

 $.get("http://localhost:8080/CORS/holaAjax.jsp",function(datos) {

 alert(datos);
 });

});

&amp;amp;lt;/script&amp;amp;gt;

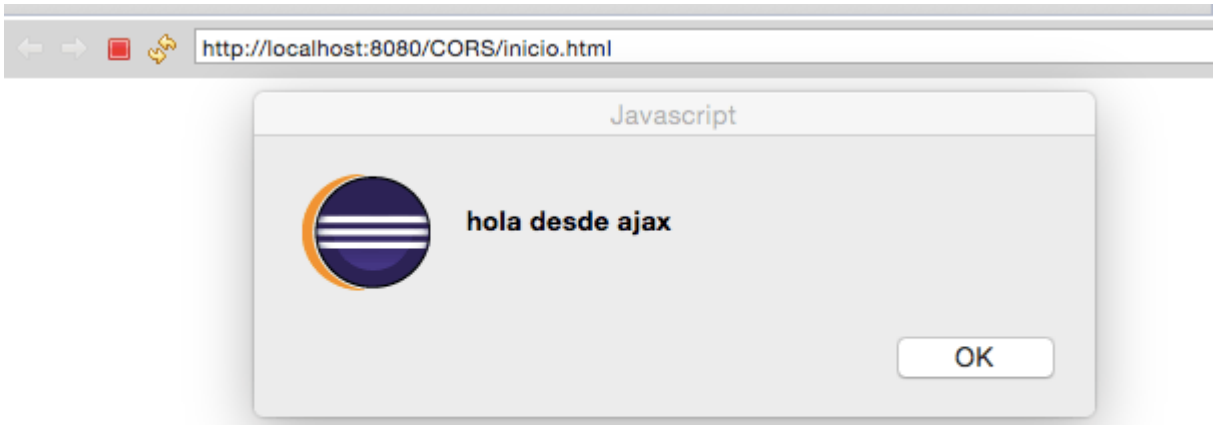
&amp;amp;lt;/head&amp;amp;gt;
&amp;amp;lt;body&amp;amp;gt;

&amp;amp;lt;/body&amp;amp;gt;
&amp;amp;lt;/html&amp;amp;gt;
```

Como vemos se trata de una petición sencilla al servidor que carga los datos que tenga la página holaAjax.jsp. Vamos a ver su código:

```
&amp;amp;lt;%@ page language="java" contentType="text/html;
charset=UTF-8"
 pageEncoding="UTF-8"%&amp;amp;gt;
hola desde ajax
```

Si pedimos la primera página al servidor , ella misma cargará la segunda página y nos mostrará un alert por pantalla.



Sin embargo si hubiera solicitado esta página de forma directa (abriendola como fichero) la aplicación no hubiera funcionado y tendríamos el siguiente error a nivel de consola.

XMLHttpRequest cannot load http://localhost:8080/CORS/holaAjax.jsp. No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'null' is therefore not allowed access.

Así pues no podremos acceder acceder de forma directa a la página y nos encontramos con un problema.Para solventar esto tenemos que actualizar la configuración de nuestro Servidor de tal forma que podamos acceder desde URL ubicadas en otros dominios. En el caso de Tomcat 8 tenemos que configurar un filtro.

```
&amp;amp;lt;filter&amp;amp;gt;

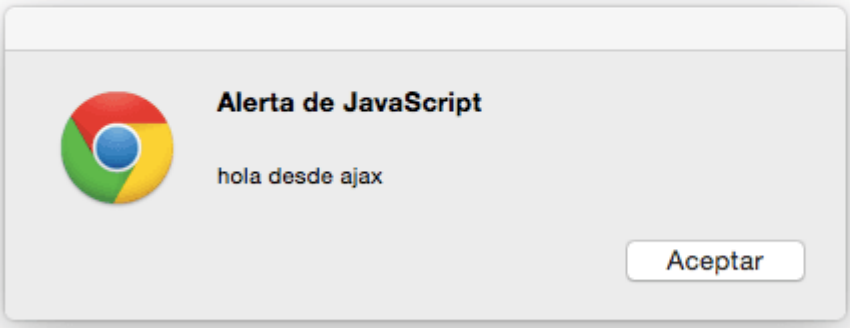
&amp;amp;lt;filter-name&amp;amp;gt;CorsFilter&amp;amp;lt;/filter-name&amp;amp;gt;

 &amp;amp;lt;filter-class&amp;amp;gt;org.apache.catalina.filters.CorsFilter&amp;amp;lt;/fi
```

```
lter-class&amp;amp;gt;
 &amp;amp;lt;/filter&amp;amp;gt;
 &amp;amp;lt;filter-mapping&amp;amp;gt;
 &amp;amp;lt;filter-name&amp;amp;gt;CorsFilter&amp;amp;lt;/filter-
name&amp;amp;gt;
 &amp;amp;lt;url-pattern&amp;amp;gt;/*&amp;amp;lt;/url-
pattern&amp;amp;gt;
 &amp;amp;lt;/filter-mapping&amp;amp;gt;
```

Este filtro simplemente nos indica que aceptemos peticiones CORS de todo tipo de dominio. Realizado este cambio en la aplicación reiniciamos el Tomcat y podremos acceder a la página(ojo Tomcat 7 da algunos problemas).





Otros artículos relacionados :

Ajax eventos globales

jQuery $.get, $.post