Introduction to Ajax

kainos

Ajax Theory

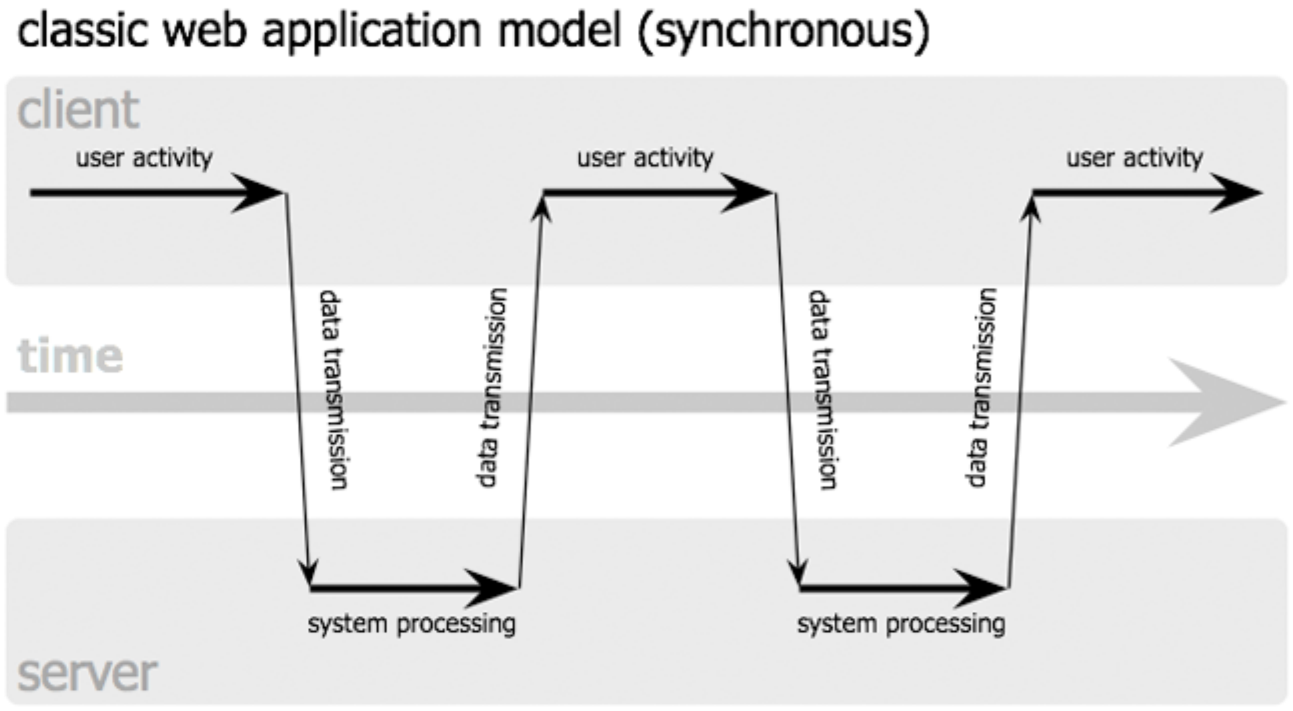
What is Ajax

"Ajax is a group of interrelated technologies used to create interactive web applications or rich Internet applications. With Ajax, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behaviour of the existing page."

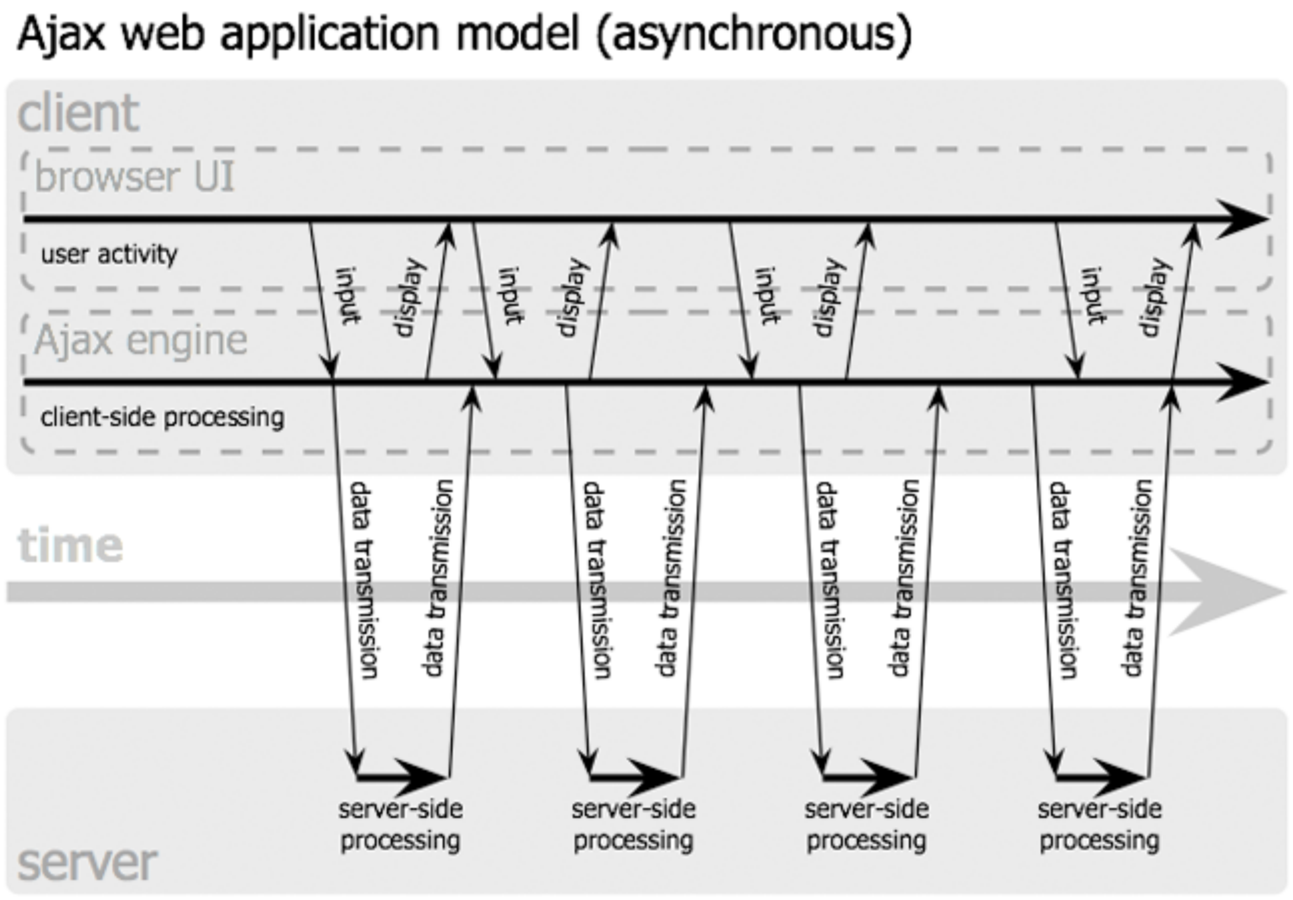
Ajax or AJAX?

"The acronym AJAX has changed to the term Ajax, which does not represent specific technologies"

Classic Communication

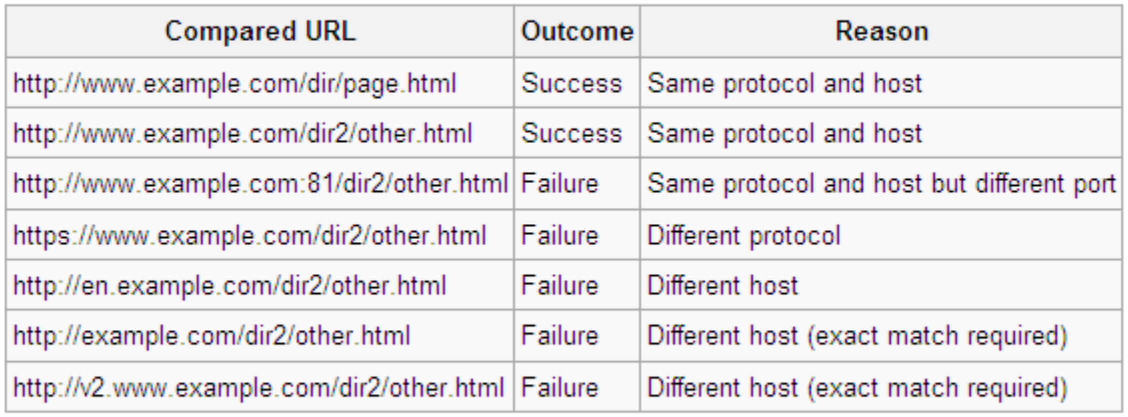


Ajax Communication



Same Origin Policy

http://www.example.com/dir/page.html



Compared URL	Outcome	Reason
http://www.example.com/dir/page.html	Success	Same protocol and host
http://www.example.com/dir2/other.html	Success	Same protocol and host
http://www.example.com:81/dir2/other.html	Failure	Same protocol and host but different port
https://www.example.com/dir2/other.html	Failure	Different protocol
http://en.example.com/dir2/other.html	Failure	Different host
http://example.com/dir2/other.html	Failure	Different host (exact match required)
http://v2.www.example.com/dir2/other.html	Failure	Different host (exact match required)

W http://en.wikipedia.org/wiki/Same_origin_policy

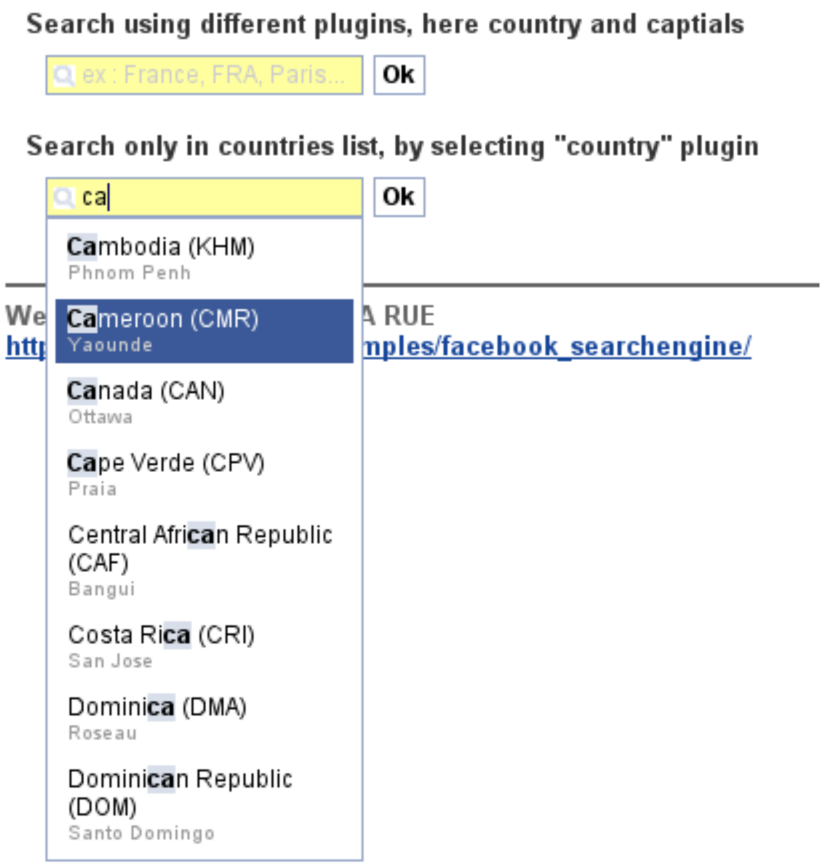
Advantages

- The interface is much more responsive. The user has the feeling that changes are instantaneous.
- Makes better use of connections (scripts and CSS loaded once)
- Waiting time is reduced. Reduced blocking
- The application fails graciously
- Traffic to and from the server is reduced considerably

Disadvantages

- Breaks the browser behaviour (back, forward, refresh)
- State is difficult to bookmark
- Opens up yet another attack vector for malicious code that needs to tested.
- **Greater Potential = Increased development time and cost due to increase in complexity and "richness"**

Uses: Facebook Auto Suggest



Search using different plugins, here country and captials

ex : France, FRA, Paris... Ok

Search only in countries list, by selecting "country" plugin

ca Ok

Cambodia (KHM)
Phnom Penh

Cameroon (CMR)
Yaounde

Canada (CAN)
Ottawa

Cape Verde (CPV)
Praia

Central African Republic (CAF)
Bangui

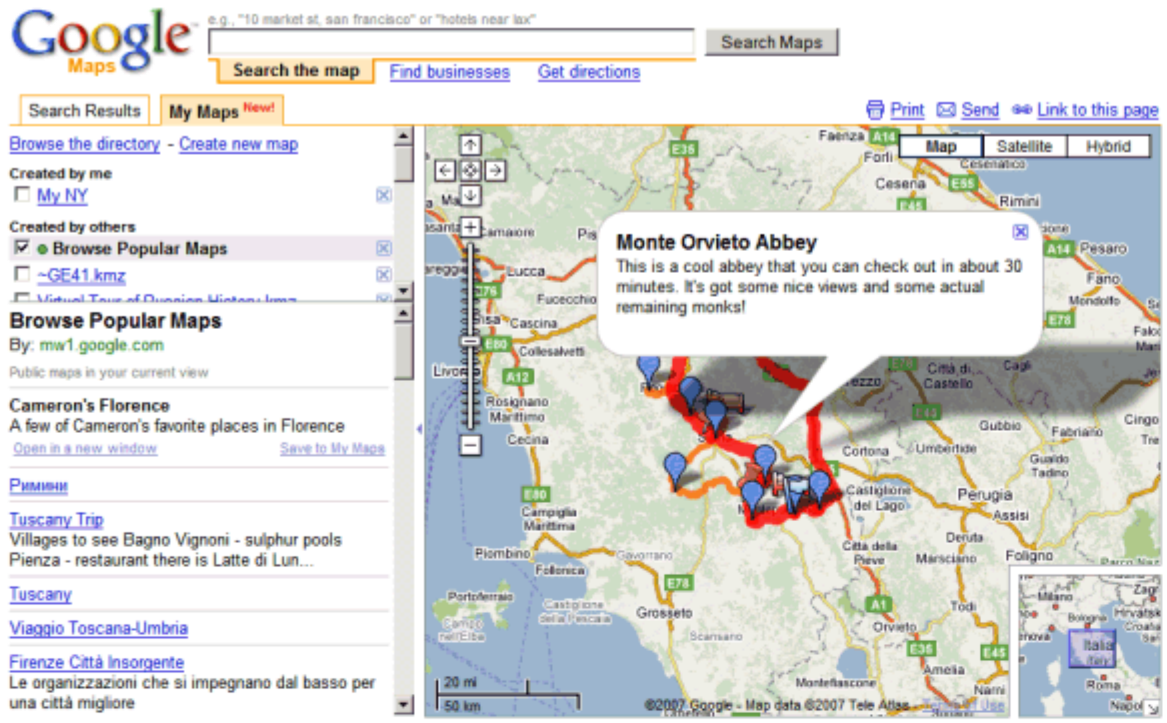
Costa Rica (CRI)
San Jose

Dominica (DMA)
Roseau

Dominican Republic (DOM)
Santo Domingo

We A RUE
http mples/facebook_searchengine/

Uses: Google Maps



Browse the directory - Create new map

Created by me

☐ My NY

Created by others

☑ ● Browse Popular Maps

☐ ~GE41.kmz

☐ Virtual Tour of Russian History.kmz

Browse Popular Maps

By: mw1.google.com

Public maps in your current view

Cameron's Florence

A few of Cameron's favorite places in Florence

Open in a new window — Save to My Maps

Римини

Tuscany Trip

Villages to see Bagno Vignoni - sulphur pools Pienza - restaurant there is Latte di Lun...

Tuscany

Viaggio Toscana-Umbria

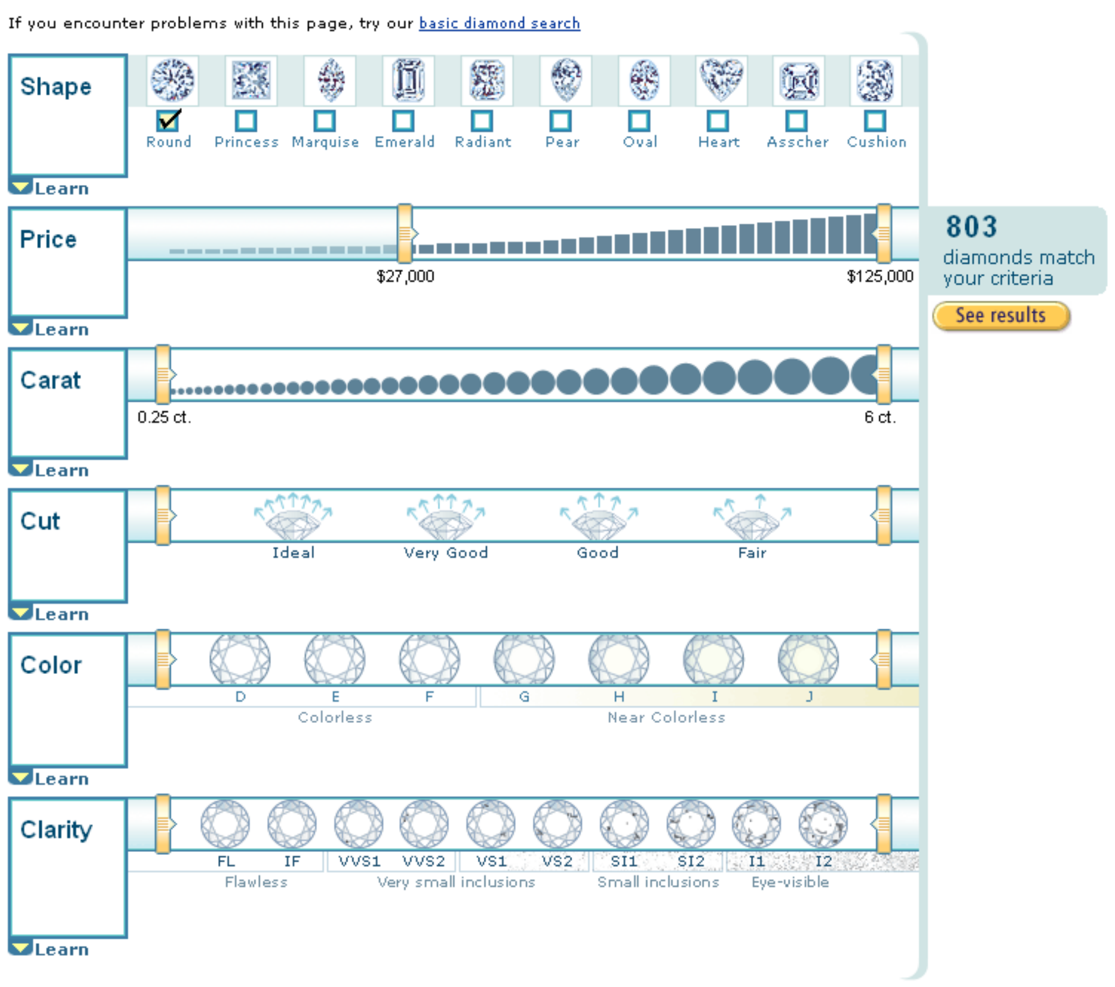
Firenze Città Insorgente

Le organizzazioni che si impegnano dal basso per una città migliore

Monte Orvieto Abbey

This is a cool abbey that you can check out in about 30 minutes. It's got some nice views and some actual remaining monks!

Uses: Amazon Diamond Search



If you encounter problems with this page, try our basic diamond search

Practical Ajax

XMLHttpRequest

- "Official" Ajax Strategy though other methods/hacks exist.
- Greater control over requests – timeouts, state monitoring
- Ability to detect and handle errors
- Browser Inconsistencies (IE's non-native ActiveX based implementation)

XMLHttpRequest Object Properties

Property	**Description**
readyState	Status of request **0 = Uninitialized** 1 = Loading 2 = Loaded 3 = Interactive (99.9% pointless) **4 = Complete**
responseText	Plain text version of response body
responseXML	XML value of response body (if XML)
status	HTTP Code returned by server (404 etc.)
statusText	HTTP Code Message

XMLHttpRequest Object Methods

Method	**Description**
abort	Aborts the current request
getAllResponseHeaders	Gets all response headers returned
getResponseHeader	Get specific response header returned
open	Set' s up an Ajax call
send	Transmits data
setRequestHeader	Sets a specific Request header

XMLHttpRequest Object Events

Method	Description
onreadystatechange	event handler that deals with state change (readyState property)

Step by Step

Step By Step: Getting an XHR Object

```
/* Get XMLHttpRequest Instance */
var _XMLHttpRequest;

try {
    /* FF, Safari, Opera, Chrome */
    _XMLHttpRequest = new XMLHttpRequest();
} catch (e) {
    try {
        /* IE (v3.0) */
        _XMLHttpRequest  = new ActiveXObject('Msxml2.XMLHTTP');
    } catch (e2) {
        try {
            /* IE (< v3.0) */
            _XMLHttpRequest = new ActiveXObject('Microsoft.XMLHTTP');
        } catch (e3) { /* Fallback? */ }
    }
}
```

Step By Step: Sending a GET Request

```
/* ------------------------------------------------------------------------ */
/*
 *  Sending an Ajax Request
 */

/* Set up the request - Asynchronous GET Method */
_XMLHttpRequest.open("GET", "http://localhost:8080/ajax/getData");

/* Send the GET request */
_XMLHttpRequest.send(null);
```

Step By Step: Sending a POST Request

```
/* ------------------------------------------------------------ */
/*
 *  Sending an Ajax Request
 */

/* Append POST Header */
_XMLHttpRequest.setRequestHeader('Content-Type',
      'application/x-www-form-urlencoded');

/* Set up the request - Asynchronous POST Method */
_XMLHttpRequest.open("POST", "http://localhost:8080/ajax/getData");

/* Send POST with Parameters as URL Encoded String */
_XMLHttpRequest.send("name=" + name);
```

Step By Step: Detecting Response

```
/* -------------------------------------------------------------------------- */
/*
 *  Detecting a Response
 */

/* Monitor readystatechange event */
_XMLHttpRequest.onreadystatechange = function() {
   /* is call complete? */
   if (_XMLHttpRequest.readyState != 4) {
      return;
   }else{
      alert(_XMLHttpRequest.responseText);
   }
}

/* Send the GET request */
_XMLHttpRequest.send(null);
```

Step By Step: Detecting Errors

```
/* ------------------------------------------------------------------------ */
/*
 *  Detecting a Response
 */


/* Monitor readystatechange event */
_XMLHttpRequest.onreadystatechange = function() {
   /* is call complete? */
   if (_XMLHttpRequest.readyState != 4) {
      return;
   }else{
      if(_XMLHttpRequest.status != 200){
         alert("Error: " + _XMLHttpRequest.statusText )
      }else{
         alert(_XMLHttpRequest.responseText);
      }
   }
}
```

Step By Step: Timing Out

```
/* ---------------------------------------------------------------------- */
/*
 *  Set a timeout to abort long running connections
 */
_XMLHttpRequest.onreadystatechange = function() {
   if (_XMLHttpRequest.readyState == 4) {
         clearTimeout(timeout);
   }
}

/* Send the GET request */
_XMLHttpRequest.send(null);

/* Set the timeout once sent */
var timeout = setTimeout(function(){
   _XMLHttpRequest.abort();
   alert("Request Timed Out");
}, 10000);
```

Alternative Techniques

Hidden IFRAME Technique

```
<html>
 <head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>
  <script>
    /* ---------------------------------------------------------------------------- */
    /*
    *  Perform the Ajax Request
    */
    function ajax(url, callback) {
      var iframe = $("#server-tunnel")[0];
      iframe.onload = function() {
        /* Find response from IFRAME */
        var r = iframe.contentWindow.document.getElementById("serverData");
        /* call user defined callback */
        callback(r.innerHTML);
      };
      iframe.src = url;
    }
    /* ---------------------------------------------------------------------------- */
    /*
    *  Handle Button Click
    */
    function doGET() {
      ajax("iframe-contents.html?name=" + $("#name").val(), function(contents) {
        $("#output").html(contents);
      });
    }
  </script>
 </head>
 <body>
  <!-- Hidden IFRAME as Server Tunnel -->
  <iframe id='server-tunnel' style='display : none'></iframe>
  <!-- User Input -->
  Name: <input type="text" id="name"/>
  <button id="btnGet" onclick="doGET()">Get</button>
  <!-- Server Response -->
  <h1>Output</h1><div id="output"></div>
 </body>
</html>
```

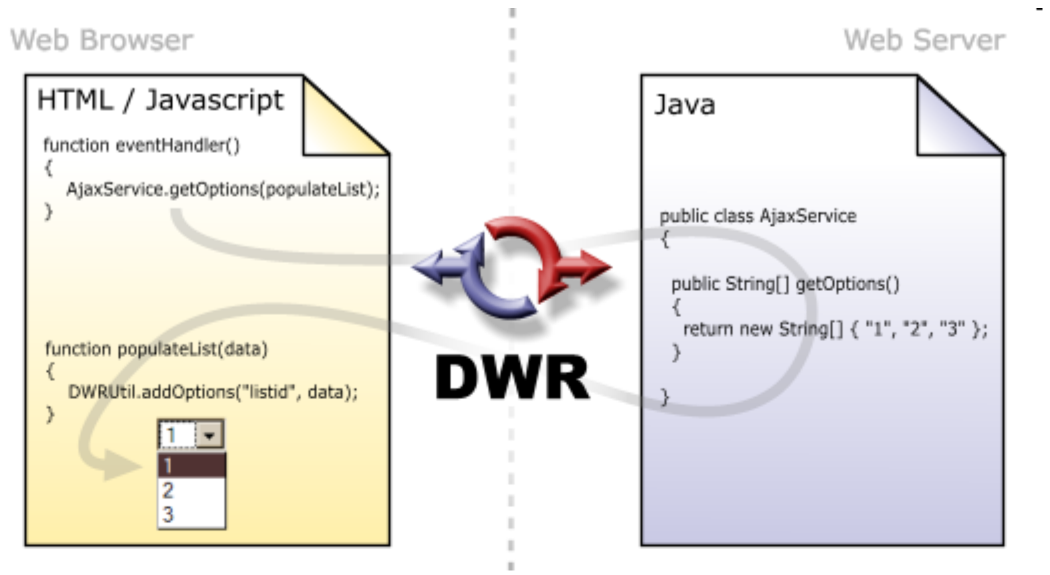
Script Tag Technique

```
<html>
 <head>
   <script src="http://code.jquery.com/jquery-latest.js"></script>
   <script>
     /* ---------------------------------------------------------------------------- */
     /*
     *  Handle Button Click
     */
     function doGET() {

       /* create new script node */
       var script = document.createElement("script");

       /* set it's source and onload event */
       script.src = "script-contents.js?name=" + $("#name").val();
       script.onload = function() {
         $("#output-script").html(serverResponse);
       }

       /* append to body - loads script */
       document.body.appendChild(script);
     }
   </script>
 </head>
 <body>
   <!-- Hidden IFRAME as Server Tunnel -->
   <iframe id='server-tunnel' style='display : none'></iframe>
   <!-- User Input -->
   Name: <input type="text" id="name"/>
   <button id="btnGet" onclick="doGET()">Get</button>
   <!-- Server Response -->
   <h1>Output</h1><div id="output"></div>
 </body>
</html>
```

Ajax Frameworks/Concepts

- Frameworks (providing Ajax abstractions):
  - jQuery
  - Prototype
  - Dojo
  - Yahoo! UI
  - … many more!
- Concepts (ideas powered by Ajax)
  - Taconite
  - Server Side Remoting: DWR (Java), JayRock (.Net)
  - Reverse Ajax: Comet, Bayeux Protocol

Reverse Ajax

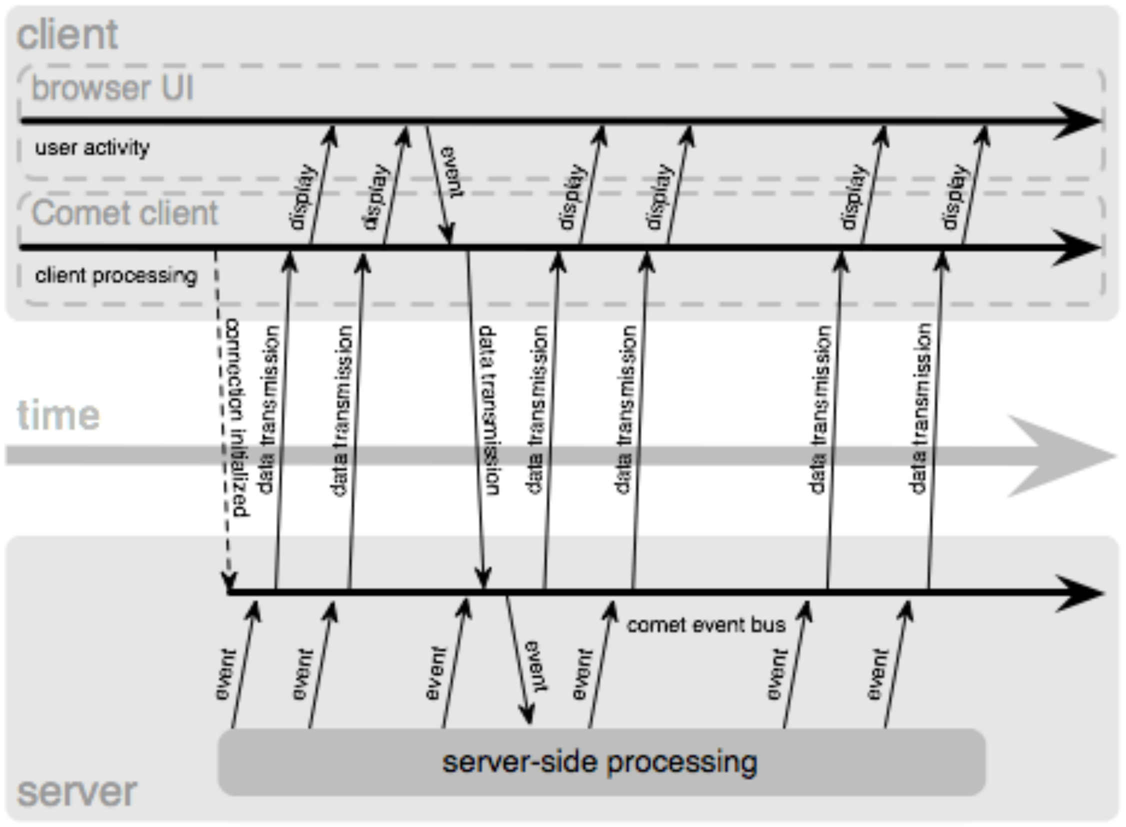


```
function eventHandler()
{
   AjaxService.getOptions(populateList);
}

function populateList(data)
{
   DWRUtil.addOptions("listid", data);
}
```

```
public class AjaxService
{
  public String[] getOptions()
  {
    return new String[] { "1", "2", "3" };
  }
}
```

Comet

Comet web application model



Considerations

- Maximum Concurrent Connections
- Same Origin Policy
- Asynchronous Nature
- Request Throttling
- Client Side Scripting reliant
- Cross Browser Issues
- Extra Capability = Extra Work

Conclusion

- Ajax isn't special, magic or difficult. To the server it's just a plain old request (though frameworks append special headers for identification)
- Ajax is an alternative not the next step
- Always ask yourself what you gain from using Ajax in the situation
- Be aware of the asynchronous nature.
- Almost all Ajax examples you seen today are abstracted through the various frameworks.

Questions?