

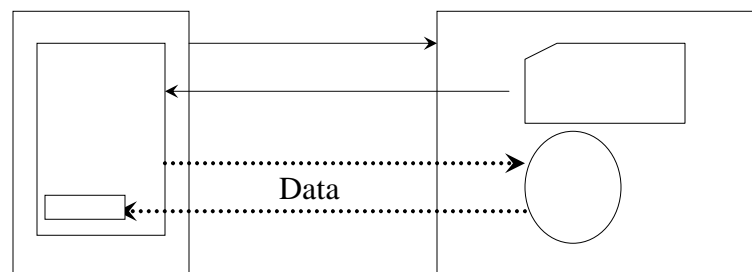
# Ajax

## What's Ajax?

- AJAX is a combination of a few technologies that has come together in the past few years
- AJAX used to be an acronym for Asynchronous JavaScript And XML (AJAX)
- It has become a name (Ajax) for a technique

## What does it do?

- It all started with Microsoft's XMLHttpRequest ActiveX object!
- Make a quite remote call to server to fetch and update new data on browser
- To make the browser experience a rich UI



Venkat Subramaniam – svenkat@cs.uh.edu

3

## Why Ajax?

- But, why do we want to Ajax an app?
- Is it because it will improve performance of server?
- Is it because it will make the application run faster?
- One of the main reasons is user experience – to make the user feel that the application is responsive

Venkat Subramaniam – svenkat@cs.uh.edu

4

## How to make the request?

- Rather than making a normal HTTP request on form submit, ask browser to
  - make a request in the background
  - keep the window responsive, and
  - display result when response arrives
- How to make that request?
- Involves browser support
  - But which browser?
    - No standard
      - This is where frameworks come in to help

## Let's Create a Contrived Example

- Would be nice to see Ajax at work without worrying about server side code
- So, here is a contrived example
- We will have static pages on server
- Based on user selection, we will pull appropriate data from these pages and display

# Starting with HTML

```
<HTML>
<head>
</head>
<body>
State: <select id="statelist">
  <option>MA</option>
  <option>CO</option>
  <option>CA</option>
  <option>TX</option>
</select>
<BR/>
Cities: <DIV id="cities"><select></select></DIV>
</body>
</HTML>
```

# Providing an Event Handler

```
State: <select id="statelist" onchange="getCities(this.value);">
  <option>MA</option>
```

## Using the XMLHttpRequest

```
</body>
<script type="text/javascript">
function getCities(state)
{
    xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4)
        {
            str = "<SELECT>";
            var cities = xhr.responseText.split(',');
            for(var i = 0; i < cities.length; i++)
            {
                str += "<OPTION>" + cities[i] + "</OPTION>";
            }
            str += "</SELECT>";
            document.getElementById('cities').innerHTML = str;
        }
    };

    xhr.open("GET", "/" + state + ".htm", true);
    xhr.send(null);
}
</script>
</HTML>
```

Venkat Subramaniam – svenkat@cs.uh.edu

9

## Using it in FireFox

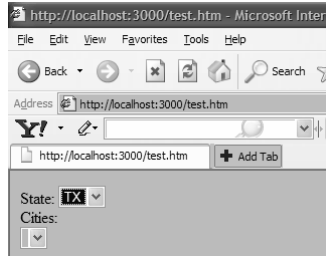
The image displays four overlapping screenshots of the Mozilla Firefox browser. The top-left screenshot shows the browser window with the URL `http://localhost:3000/test.htm` and a form with a 'State' dropdown set to 'MA' and an empty 'Cities' dropdown. The top-right screenshot shows the 'State' dropdown set to 'TX' and the 'Cities' dropdown populated with 'MA', 'CO', 'CA', and 'TX'. The bottom-left screenshot shows the 'State' dropdown set to 'TX' and the 'Cities' dropdown populated with 'Austin'. The bottom-right screenshot shows the 'State' dropdown set to 'TX' and the 'Cities' dropdown populated with 'Austin', 'Houston', and 'Dallas'. Below the browser windows, the Firebug console is open, showing a GET request to `http://localhost:3000/TX.htm` with a response containing the text 'Austin, Houston, Dallas'.

Firebug showing the async request and response

Venkat Subramaniam – svenkat@cs.uh.edu

10

## How about using IE?



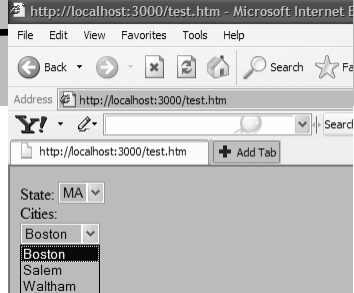
Does not work!

## Microsoft started it...

- Microsoft started it with the ActiveX object
- Other browsers have followed, but...
- No standard
- Making it work with IE...

## For IE

```
State: <select id="statelist" onchange="getCities(this.value);">
  <option value="MA">MA</option>
  <option value="CO">CO</option>
  <option value="CA">CA</option>
  <option value="TX">TX</option>
</select>
<BR/>
Cities: <DIV id="cities"><select></select></DIV>
</body>
<script type="text/javascript">
  function getCities(state)
  {
    xhr = new ActiveXObject("Microsoft.XMLHTTP");
    // Earlier versions used new ActiveXObject("Msxml2.XMLHTTP");
    xhr.onreadystatechange = function() {
      if (xhr.readyState == 4)
      {
        str = "<SELECT>";
```



Venkat Subramaniam – svenkat@cs.uh.edu

3

## But, then...

- Firefox does not work any more!
- What about other browser variations?
- Need to deal with these differences...

Venkat Subramaniam – svenkat@cs.uh.edu

14

## Dealing with differences...

```
function createXHR()
{
    var xhr;

    try
    {
        xhr = new ActiveXObject("Msxml2.XMLHTTP");
    }
    catch(e)
    {
        try
        {
            xhr = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch(e)
        {
            xhr = false;
        }
    }

    if (!xhr && typeof XMLHttpRequest != 'undefined')
    {
        xhr = new XMLHttpRequest();
    }

    return xhr;
}

function getCities(state)
{
    xhr = createXHR();
```

Venkat Subramanian

15

## Giving a Visual Effect of Change

- Users are not used to change happening
- Need to keep them aware of what's going on
- However, be careful how you do this...
- Do not turn your application into an annoyance for the users

Venkat Subramanian – svenkat@cs.uh.edu

16



## A Very Simple Effect

```
for(var i = 0; i < cities.length; i++)
{
    str += "<OPTION>" + cities[i] + "</OPTION>";
}
str += "</SELECT>";
cities = document.getElementById('cities');
cities.innerHTML = str;

cities.style.background = 'red';
setTimeout('cities.style.background = null;', 1000);
}
```

Does not work in IE



State: MA  
Cities: Boston

State: MA  
Cities: Boston

## Frameworks

- To ease the pain of dealing with
  - JavaScript
  - Browser differences
  - API
- Several frameworks have emerged
  - Too many frameworks out there
  - For various platforms, languages
    - Java, .NET, ...
- To mention a few
  - Dojo, Django, Mochikit, Rico, Prototype, JQuery, ...