

AJAX: Overview

- AJAX (Asynchronous JavaScript and XML) is an improved browser protocol model.
- It was made possible by Microsoft's introduction of - and other browsers' subsequent adoption of - the "XmlHttpRequest" object in JavaScript.
- JavaScript was already being used for a lot of front end work.
- The XmlHttpRequest made it possible for JavaScript to send "sub-requests" to the server, without needing to use navigation.
- Google was an early adopter of this technology and helped popularize it.
- The name "AJAX" was coined for the new technology by Jesse Garrett of Adaptive Path.

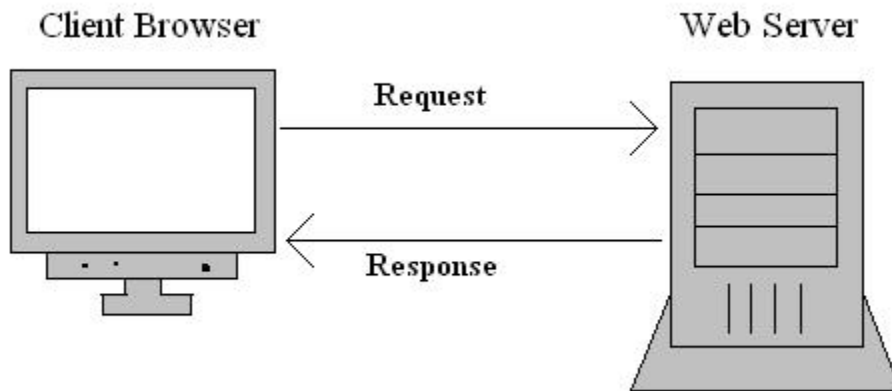
Traditional Browser Model

- Request-Response
- Browser to `www.server.com`

Please send file /MyFile.html

- Server Response:

```
<HTML>  
<BODY  
.....  
stuff goes here  
.....  
</BODY>  
</HTML>
```



Modifying the Page

- Issue: How to change one item on the page?

```
<HTML>
<BODY
.....
stuff goes here
<P>There are 791 members in our organization.</P>
.....
</BODY>
</HTML>
```

The user clicked "I quit", so we want to change the 791 to 790.

- Solution: No problems. Get the browser to send another request.
- When user clicks "I quit", the browser sends a request for a fresh page, and the server responds with:

```
<HTML>
<BODY
.....
stuff goes here
<P>There are 790 members in our organization.</P>
.....
</BODY>
</HTML>
```

- To change that 1 to a 0, the entire page has to be re-sent, and the browser has to redraw the entire page.

New Approach with Ajax

- Locate and change **just** the item that has changed.

```
<HTML>
<BODY
.....
stuff goes here
<P id="count">There are 791 members in our organization.</P>
.....
</BODY>
</HTML>
```

- For convenience, we have tagged the para with ID of "count".
- When user clicks "I quit", the browser sends a "mini-request" "What's the current text for the count para"?
- The server sends

There are 790 members in our organization.

- The browser updates just one para.
- There is less data transfer, and the page updates smoothly.
- The performance improvement expectations are a bit counter-intuitive -- even though a lot less bytes are being transferred, the perceived performance is not necessarily going to be particularly improved!
- Reason: Internet is "packet" oriented, not "byte" oriented. It doesn't matter how much you stuff in one packet, it will still take the same time for one packet transfer!

Some Actual Advantages of AJAX

- Programming can be easier.
- Better User Interfaces can be designed.
- Changes can be made continuously to a page without needing to simulate them by navigation techniques.
- Large pages will indeed benefit from small updates.

Jaxcent

- Java Only. Many frameworks involve JavaScript work.
- Server Side. Even if frameworks don't require JavaScript programming, they are client-side.
- Incremental. Can be added to existing applications without disrupting existing behavior.
- JavaScript compatible. Can work with JavaScript widgets, such as menu systems, mapping/scrolling systems, etc.

Jaxcent Examples

Hello, World

HTML File HelloWorld.html:

```
<HTML>
<HEAD>
<TITLE>Hello World Sample</TITLE>
<SCRIPT SRC="/jaxcent20.js"></SCRIPT>
</HEAD>
<BODY>
<P>
The following paragraph has been marked with an ID for
ease of
finding it from Java.
<P ID="text">Some old text</P>
</BODY>
</HTML>
```

Java File HelloWorld.java:

```
package jaxlet;
import jaxcent.*;
public class HelloWorld extends jaxcent.JaxcentPage {
}
```

The Jaxcent Configuration File maps these two:

```
<Page>
  <PagePath>/HelloWorld.html</PagePath>
  <PageClass>jaxlet.HelloWorld</PageClass>
</Page>
```

The Java code doesn't do anything at this time.

Hello, World in Jaxcent

This declares an object for the P(ara) with id “text”, and modifies text to say Hello, World.

```
package jaxlet;
import jaxcent.*;

public class HelloWorld extends jaxcent.JaxcentPage {
    HtmlPara para = new HtmlPara( this, "text" );

    public HelloWorld()
    {
        try {
            para.setInnerHTML( "Hello, World!" );
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

Further change to make this an interesting display:

```
public HelloWorld()
{
    try {
        para.setInnerHTML( "Hello, Charlotte!" );
        new Thread() {
            public void run() {
                update();
            }
        }.start();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

void update()
{
    try {
        int size = 10;
        while ( true ) {
            for ( int i = 0; i < 100; i++ ) {
                size++;
                para.setStyle( "font-size",
                    String.valueOf( size ) + "px" );
                Thread.sleep( 10 );
            }
            for ( int i = 0; i < 100; i++ ) {
                size--;
                para.setStyle( "font-size",
                    String.valueOf( size ) + "px" );
                Thread.sleep( 10 );
            }
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
```

Forms Processing in Jaxcent

What else is AJAX (and Jaxcent) good for besides fun and games?

Jaxcent can make data entry easy by auto-saving all form fields in the session.

Data can be restored into forms from session at page load.

Programmer can retrieve all data as a map.

Forms processing demo is available at <http://www.jaxcent.com/>

Other Examples in Jaxcent

Jaxcent can be used for enhanced user interfaces, such as Drag and Drop. Drag and Drop demo is available at <http://www.jaxcent.com/>

A simple Chat program is demonstrated at <http://www.jaxcent.com/>

Existing Javascript widgets can be easily integrated into Jaxcent. A menu demo is also available at <http://www.jaxcent.com/>