

## 1 Ajax and PHP

Tim Margush

Based on Programming the WWW by Robert Sebesta, 4<sup>th</sup> Edition

## 2 What is Ajax?

- Asynchronous JavaScript and XML
  - Coined by Jesse James Garrett of Adaptive Path in an article published in Feb 2005
- The name gave this technology momentum that sparked an interest in innovative Web development
  - Java Applets and Remote Scripting were predecessors – also iframes
- Key component is the XMLHttpRequest object, introduced in 2002

## 3 Ajax Communication

- After a page loads, a JavaScript function is invoked by some event
- This function issues a request to the WWW server through an XMLHttpRequest object
- Returned information is often used to update the page
- Key features
  - Asynchronous capability
  - Page interaction without full page reloads

## 4 Instantiating XMLHttpRequest

- `req = new XMLHttpRequest();`
  - A different syntax is required for this to work in Internet Explorer versions below 7
- This object has properties and methods that allow communication back through the WWW server that served the page
  - Note this limitation – only requests back to the same server will be honored

## 5 Methods: open and send


- The connection to the server is established through the open method
  - `req.open(method, URL, async, user, password)`
    - `method`: GET, POST, HEAD, PUT, ...
    - `url`: relative or complete
    - `async`: true=blocking, false=blocking
- The actual request is transmitted via the send method
  - `req.send(data)`


## 6 Send


- GET requests have all data specified in the URL
  - `req.open("GET", "s.php?user=fred&action=3", true);`
  - `req.send(null); //no data`
- POST requests allow a query string as an argument to the send method
  - `req.open("POST", "s.php", true);`
  - `req.setRequestHeader("Content-type", "application/x-www-form-urlencoded");`
  - `req.send("user=fred&action=3"); //data`


## 7 S(ynchronous)jax

- `req.open("GET", "http://abc.htm");`
- `req.send(null);`
- `//Javascript WAITS for response from WWW server, then continues to next statement`
- `alert(req.responseText)`
  
- Possible browser lockups

- 8  **A(synchronous)jax**
- req.open("GET","http://abc.htm", true);
  - req.onreadystatechange = callback\_fn
  - req.send(null);
  - //Javascript continues with no delay
- Server responses are registered by the XMLHttpRequest object and call the onreadystatechange handler each time the state changes


- 9  **req.onreadystatechange**
- Registers a callback function that runs each time the state of the request changes.
    - req.onreadystatechange = function\_name
  - There are 5 states
  - The onreadystatechange function is called when
    - a connection is opened
    - send is called
    - headers are received
    - data is being received
    - receipt of data is complete


- 10  **Request Phases: req.readyState**
- 0 = no connection to a server yet established
  - 1 = a connection was made, but the request has not yet been sent
  - 2 = the request was sent and headers are available
  - 3 = the reply is being received
  - 4 = reply received
    - req.responseText holds the information when the state reaches 4

- 11  **Callback Function**
- ```

if (req.readyState==4){
    //Check status (should be 200)
    //Process information
}
//Common design pattern
req.onreadystatechange=function(){
    if (req.readyState==4 && req.status==200)
        process(req.responseText);
};

```

- 12  **Multiple Requests**
- What if one ajax request is in progress when another begins?
    - Different XHR objects
    - Same XHR Objects
    - Different callback functions
    - Same callback functions
  - Opening a new connection cancels previous request
    - req.abort()

- 13  **Object nonReuse**
- ```

if (req==null) req=new XMLHttpRequest();
req.open(...)
req.onreadystatechange=new function(){

```

```

    if (req.readyState==4){
    if (req.status==200)
        process(req.responseText);
    else
        process("Error");
        req = null;
    }
}

```

#### 14 Ajax and DHTML

- Ajax requests combined with DHTML provide highly interactive and responsive web pages
  - Server-side data validation accomplished on a per-item basis
  - User interaction can affect page content dynamically, reducing the amount of data that would otherwise have to be loaded

#### 15 Example

- Auto-complete form fields
  - Keystrokes are transmitted to a server-side script and suggestions are returned

#### 16 Asynchronous Request

```

function update(city){
    req.open("GET","find.php?city="+city,true);
    req.onreadystatechange=showcities;
    req.send(null);
}

```

Assumes XMLHttpRequest object already was instantiated and automatically aborts pending request

#### 17 Asynchronous Reply

```

function showcities(){
    if (req.readyState==4){
        cd = document.getElementById('citylist')
        if (req.status==200)
            cd.innerHTML = req.responseText;
        else
            cd.innerHTML = "Ajax Error";
    }
}

```

#### 18 Matching Host Limitation

#### 19 Proxy

myproxy.php?http://find.php?city=akr

```

myproxy.php:
<?php
    header("Content-type: text/plain");
    $url = getenv("QUERY_STRING");
    readfile($url);
?>

```

## 20 Auto-complete

- Fill in likely matches while user enters data in a text field
  - Each keypress must replace the current suggested completion
  - This is facilitated by selecting the tail of the text

```
tf=document.getElementById('city');  
tf.focus();  
tf.setSelectionRange(startpos, startpos+len);
```

## 21 Select Range – Browser Dependent

```
function select(txtf, startpos, len){  
  if(txtf.selectionRange){ //not IE  
    txtf.focus();  
    txtf.setSelectionRange(startpos, startpos+len);  
  } else if (txtf.createTextRange) { //IE  
    var r = txtf.createTextRange();  
    r.collapse(true); //start at beginning  
    r.moveStart("character", startpos);  
    r.moveEnd("character", len);  
    r.select();  
  }  
}
```

## 22 AutoComplete

```
onkeyup="complete(this.value)"  
  
function complete(city){  
  req=new XMLHttpRequest()  
  req.open("GET", "find.php?...  
  req.onreadystatechange =  
  new function(){  
    if (req.readyState==4 && req.status=200)  
      autocomplete(req.responseText)  
  }  
  req.send();  
}
```

## 23 Auto-complete

```
function autocomplete(suggestion){  
  tf = document.getElementById('city');  
  prefix = tf.value;  
  selectstart = tf.value.length;  
  selectlen = suggestion.length-selectstart;  
  tf.value = suggestion;  
  select(tf, selectstart, selectlength);  
}
```