



# Responsible Asynchronous Scripting

Presented by **Shaun Inman**

# Remoting 101

## Traditional request model

- 1 Client requests resource
- 2 Server serves resource
- 3 Repeat

## Of note

Subsequent requests replace the previously viewed resource

## Asynchronous model

- 1 Client requests resource
- 2 Server serves resource
- 3 Client issues sub-requests
- 4 Client updates the existing resource

### Of note

The original resource is not replaced by sub-requests

# A brief **history**

## Remote scripting techniques

Using a dedicated Java applet,  
ActiveX control, or Flash movie as  
proxy

## Remote scripting techniques

Piggybacking data on *image* requests, receiving a response via *cookie*

## Remote scripting techniques

# Submitting data to an invisible *iframe*

## Remote scripting techniques

Attaching a new *script element* to the document, receiving a JavaScript response

## Remote scripting techniques

Using the `XMLHttpRequest` object, receiving an XML, JSON, JavaScript or XHTML response

# Using XMLHttpRequest

## Browsers that support XMLHttpRequest

Internet Explorer PC 5+

Firefox

Safari 1.2+

Netscape 7.1+

Opera 8+

```
// A minimal XHR example  
var request = new XMLHttpRequest();  
request.open('GET', url);  
request.send();
```

```
// Still a minimal XHR example  
var request = new XMLHttpRequest();  
request.open('GET', url);  
request.send(null);
```

```
// now *this* looks like JavaScript!
```

```
var request = false;
```

```
if (window.XMLHttpRequest) { request = new XMLHttpRequest();};
```

```
/*@cc_on @*/
```

```
/*@if (@_jscript_version >= 5)
```

```
if (!request)
```

```
{
```

```
    try {request = new ActiveXObject("Microsoft.XMLHTTP");}
```

```
    catch (e) {request = false;};
```

```
};
```

```
@end @*/
```

```
request.open('GET', url);
```

```
request.send(null);
```

```
<a href="ping.php" onclick="return get(this.href);">Ping</a>
```

```
function get(url)
{
    var request = false;
    if (window.XMLHttpRequest) { request = new XMLHttpRequest(); };
    /*@cc_on @*/
    /*@if (@_jscript_version >= 5)
    if (!request)
    {
        try { request = new ActiveXObject("Microsoft.XMLHTTP"); }
        catch (e) { request = false; };
    };
    @end @*/
    if (!request) { return; };
    request.open('GET', url);
    request.send(null);
    return false;
};
```

```
function get(url, target)
{
    // XHR detection code removed to fit slide

    request.open('GET', url);
    request.send(null);

    request.onreadystatechange = function()
    {
        if (request.readyState == 4 && request.status == 200)
        {
            target.innerHTML = request.responseText;
        }
    };
};

return false;
};
```

```
<div id="ping"><?php include('ping.php'); ?></div>  
<a href="index.php" onclick="  
    return get('ping.php', document.getElementById('ping'));  
">Ping</a>
```

# Content or action?

## The impact on findability

Remoting is appropriate for  
certain actions and transient data  
–not content navigation

**Good**

Live search

**Bad**

Any sort of navigation

**Good**

Auto-complete and as-you-type  
data validation

**Bad**

The one-page e-commerce site

**Good**

Content rating and tagging

**Bad**

Inline login forms

## A litmus test

Refreshing a page immediately after that page has been updated by remote scripting results *essentially* in an identical page.

# Remoting doesn't break the back button

```
function post(url, data, target)
{
    // XHR detection code removed to fit slide
    request.open('POST', url);
    request.setRequestHeader('Content-Type',
                            'application/x-www-form-urlencoded');
    request.send(data);

    request.onreadystatechange = function()
    {
        if (request.readyState == 4 && request.status == 200)
        {
            target.innerHTML = request.responseText;
        }
    };
};

return false;
};
```

```
<div id="rating"><?php include('rate.php'); ?></div>
```

```
<a href="index.php" onclick="
```

```
    return post('rate.php',
```

```
                'rating=1',
```

```
                document.getElementById('rating'));
```

```
">1</a>
```

```
<a...>2</a>
```

```
<a...>3</a>
```

```
<a...>4</a>
```

```
<a...>5</a>
```

```
<form method="post" action="index.php">  
  <input type="hidden" name="rating" value="1" />  
  <input type="submit" value="1" />  
</form>
```

```
<form... value="2" /></form>
```

```
<form... value="3" /></form>
```

```
<form... value="4" /></form>
```

```
<form... value="5" /></form>
```

```
<form method="post" action="index.php" onsubmit="
    return post( 'rate.php',
                'rating=1',
                document.getElementById( 'rating' ) );
">
    <input type="hidden" name="rating" value="1" />
    <input type="submit" value="1" />
</form>

<form...value="2" /></form>
<form...value="3" /></form>
<form...value="4" /></form>
<form...value="5" /></form>
```

```
<form method="post" action="index.php" onsubmit="
    return post(this,
                'rate.php',
                document.getElementById('rating'));
">
    <input type="hidden" name="rating" value="1" />
    <input type="submit" value="1" />
</form>

<form... value="2" /></form>
<form... value="3" /></form>
<form... value="4" /></form>
<form... value="5" /></form>
```

```
function post(form, action, target)
{
    // XHR detection code removed to fit slide
    var data = new Array();
    for (var i = 0; i < form.elements.length; i++)
    {
        var e = form.elements[i];
        // Pseudo-code to add each form element name
        // and value to the data array
        data.push(escape(e.name) + '=' + escape(e.value));
    };
    data = data.join('&');
    // XHR send and callback code removed to fit slide
};
```

# Communicating **change**

# Spinners, hourglasses and progress bars

# Animation and motion

Deleting remaining slides 

Slides deleted

Questions? 

# Etceteras

```
url += ((url.indexOf('?') != -1) ? '&' : '?') + (new Date()).getTime();
```