CS3220 Web and Internet Programming

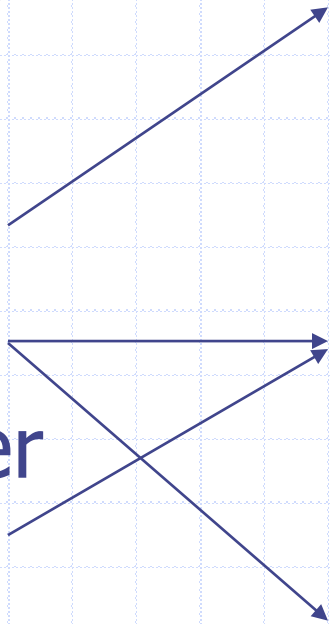
Asynchronous JavaScript and XML (AJAX)

Chengyu Sun

California State University, Los Angeles

Improve Responsiveness of Web Applications

- Improve user experience
  - Interactive
  - Responsive
- Reduce server workload



Handle input events

Implement as much functionality on the client-side as possible

Hide the inevitable communication overhead from the user

Communicate with Server

- The *synchronous* request-response model is still a limiting factor in responsiveness
- Solution: XMLHttpRequest
  - A JavaScript object
    - Send request and receive response in JavaScript
  - Response can be handled *asynchronously*
    - Do not need to wait for the response

Understand Asynchronous …

- Synchronous

```
send( request );

// wait for response

process( response );

// do other things
…
```

- Asynchronous

```
send( request );

// don't wait for response

process( response );

// do other things
…
```

What's the problem??
What's the solution??

… Understand Asynchronous

- Asynchronous

```
// callback function
function foo( response )
{
  process( response );
}
```

Same as handling events like click event in GUI programming.

```
// set a callback function
// which will be called
// when the response comes
// back
… …

send( request );

// do other things
…
```

An XMLHttpRequest Example

- Servlet `Number`
- `number1.html`
  - A client scripts sends an XMLHttpRequest
  - A server program responds with a random number
  - When the message arrives on the client, a *callback function* is invoked to update the document

About the Example

- clickHandler()
- newXMLHttpRequest()
- updateDocument()
- getReadyStateHandler()

XMLHttpRequest - Properties

- onreadystatechange
- readyState
  - 0 – uninitialized
  - 1 – loading
  - 2 – loaded
  - 3 – interactive
  - 4 – complete
- status
- statusText
- responseBody
- responseStream
- responseText
- responseXML

XMLHttpRequest - Methods

- abort()
- getAllResponseHeaders()
- getResponseHeader(header)
- open(method, url, asyncFlag, username, password)
  - asyncFlag, username, password are optional
- send(messageBody)
- setRequestHeader(name, value)

So What is AJAX?

- Asynchronous JavaScript and XML
  - JavaScript + XMLHttpRequest
- Characteristics of AJAX
  - *Non-blocking* – the server response is handled asynchronously with a callback function
  - *Partial page update* using JavaScript

More About AJAX

- XMLHttpRequest used to be an IE specific feature that received little attention
- Then Mozilla added it to Firefox
- Then it was used to create Google Maps → the beginning of "Web 2.0"

Key Elements of an AJAX Operation

Client

- Event
- Event handler
  - Create a XMLHttpRequest
  - Attach a callback function
  - Send the request
- Callback function
  - Process the response
  - Update the HTML Page

Server

- Process the request
- Send back a response

XMLHttpRequest is the name of the JavaScript object; the actual request is still an HTTP Request, i.e. there is no difference on the server side.

Problems of Plain JavaScript + XMLHttpRequest

- Each browser has their own JavaScript implementation
  - Code that works on some browsers may not work on others
- Implementing AJAX operations is quite tedious

jQuery To The Rescue

- `Number2.html`
  - http://api.jquery.com/category/ajax/
  - $.ajax()
    - URL can be specified as an argument or as a field of the `settings` object
    - `success` callback function will be called if the request is successful (i.e. response status code 200)

More About `settings`

Data Fields

- url
- method
- data

Callback Functions

- success
- error

Example: AJAX GuestBook

- Implement Add and Delete Entry using AJAX

About The Example

- User-defined attributes in HTML 5: `data-*`
- DOM traversal with `closest()`
- Return an empty response