

JavaScript strings clarification



- ❑ Non-standard syntax used in an example
 - ❑ Using `s[i]` to access the i^{th} character of string `s`
 - ❑ Works in Firefox, does not work in IE
- ❑ Standard functions for a working with strings
 - ❑ Use `s.charAt(i)` to access the i^{th} character
 - ❑ Use `s.length` to find its length
 - ❑ Use `s.substring(i, j)` to get substring from $i+1^{\text{th}}$ to j^{th} character
 - ❑ Use `s.indexOf('a')` to find index of first occurrence of character `'a'`

CS 638 Web Programming – Eistan & Kivolowitz

Web services

Lecture 21

CS 638 Web Programming



Lecture outline



- ❑ Web Services
 - ❑ Remote procedure calls
- ❑ AJAX (asynchronous requests)

CS 638 Web Programming – Eistan & Kivolowitz

What are web services?



- ❑ A form of remote procedure call: your program (the client) asks another computer (the server) to run a procedure for you
 - ❑ Parameters sent over the network from client to server
 - ❑ Results sent over network from server to client
- ❑ Why would you ever want to do a remote procedure call?
 - ❑ Data needed for answer not (easily) accessible to your computer
 - ❑ You want to re-use existing procedures that run in a different environment than your program
 - ❑ Your computer lacks the resources (i.e. processor capacity, memory, network connection speed) to compute the result
- ❑ There are many other forms of RPC older than web services
 - ❑ CORBA, DCOM, SunRPC, RMI

CS 638 Web Programming – Estan & Kivolowitz

Internals of an RPC framework



- ❑ Code for marshalling/unmarshalling – encoding and decoding parameters/results
 - ❑ A.k.a. serializing objects
- ❑ Description of the available procedures (methods)
 - ❑ Using an interface description language (IDL)
- ❑ Framework that turns these descriptions into “stubs”
 - ❑ On the client the stub makes it look to your program like the stub is executing the procedure locally
 - ❑ On the server the stub invokes the procedure
 - ❑ The client and server stub interact over the network

CS 638 Web Programming – Estan & Kivolowitz

Specific to web services



- ❑ They run over http
 - ❑ Procedure call is in an http request
 - ❑ Result is in an http response
- ❑ They use XML to
 - ❑ Encode responses
 - ❑ Encode requests (sometimes)
 - ❑ Describe the procedures (incl. arguments and results)
- ❑ Client and server often use different languages
 - ❑ Client may be JavaScript code in browser – AJAX
- ❑ Client and server are often in different organizations

CS 638 Web Programming – Estan & Kivolowitz

More on web services



- Many companies allow access to their data / services through web services (some free)
 - Examples: Amazon, ebay, Google, Yahoo, USPS
- Two popular flavors
 - SOAP (**S**imple **O**bject **A**ccess **P**rotocol) – requests in XML
 - REST (**R**epresentational **S**tate **T**ransfer) – requests are like query strings of HTML forms (name-value pairs)
 - SOAP more powerful, REST has lower overhead
- WSDL (web service description language) used for XML file describing method names, parameters and results for a given web service

CS 638 Web Programming – Estan & Kivolowitz

ASP.NET and web services



- Writing web services
 - Create new web site of type “web service”
 - Write C# methods exposed to clients
 - The system does the rest (incl. generating WSDL)
- Using web services
 - Add “web reference” to your site/project
 - Give URL of WSDL file
 - Use web service in your code
 - System adds “glue” – class definitions, stubs for building requests and parsing responses, etc.

CS 638 Web Programming – Estan & Kivolowitz

The screenshot shows the Visual Studio IDE with the following elements:

- Code Editor:** Shows the code for `AmazonWebService.aspx.cs`. Annotations include:
 - A yellow circle around the `using System.Web.Services;` line.
 - A yellow circle around the `MyAmazonAccessKeyID` property definition.
 - A yellow circle around the `AmazonWebService` class definition.
 - A yellow circle around the `AmazonWebService` class name in the `protected void Page_Load` method.
 - A yellow circle around the `AmazonWebService` class name in the `try` block.
- Solution Explorer:** Shows the project structure with annotations:
 - A yellow circle around the `App_WebReferences` folder.
 - A yellow circle around the `amazon` folder.
 - A yellow circle around the `AmazonWebService.aspx` file.

Lecture outline



- ❑ Web Services
 - ❑ Remote procedure calls
- ❑ AJAX (asynchronous requests)

CS 638 Web Programming – Estan & Kivolowitz

What is AJAX?



- ❑ Dynamic, interactive web pages that feel more like a desktop application than a static page
- ❑ Key technologies – Asynchronous requests, JavaScript, DOM manipulation, CSS (especially positioning of elements), web services using XML
- ❑ Core tenet – in response to user actions, do not refresh full page from server, send small requests as needed and update individual elements of web page
- ❑ Examples: dragging/zooming maps at Google and Yahoo, “tooltip details” at Netflix, etc.

CS 638 Web Programming – Estan & Kivolowitz

Asynchronous requests



- ❑ Requests from browser other than those triggered by the user following a link or submitting a form
 - ❑ Not postbacks
- ❑ Explicitly triggered by JavaScript code
 - ❑ Building a request object
 - ❑ Registering a handler with the browser
 - ❑ Browser calls handler when various events occur (e.g. the reply arrives)
- ❑ May not be a direct result of an user action
 - ❑ E.g. pre-fetching neighboring tiles of a map

CS 638 Web Programming – Estan & Kivolowitz

AJAX vs. postbacks



- ❑ Replies to asynchronous requests may be smaller
 - ❑ Typically contain less information than full page
- ❑ When coupled with animations, asynchronous requests can achieve effects that are not possible with postbacks
- ❑ Use postbacks triggered by a submit button when you want to be sure that the user understands that she is submitting information to the server

CS 638 Web Programming – Estan & Kivolowitz

The XMLHttpRequest object



```
LiveSearch.js
var xmlhttp;
function createXMLHttpRequest() {
  if (window.ActiveXObject) {
    return new ActiveXObject("Microsoft.XMLHTTP");
  } else if (window.XMLHttpRequest) {
    return new XMLHttpRequest();
  }
}

function do_liveSearch(pattern) {
  xmlhttp = createXMLHttpRequest();
  xmlhttp.onreadystatechange = handleStateChange;
  xmlhttp.open("GET", document.URL.substring(0, document.URL.lastIndexOf("/") +
    "/SearchService.aspx?pattern="+pattern, true);
  xmlhttp.send(null);
}

function handleStateChange() {
  if (xmlhttp.readyState == 4) {
    if (xmlhttp.status == 200) {
      updatePage();
    } else { alert("Status "+xmlhttp.status); }
  }
}

//var xmlDoc = xmlhttp.responseXML;
```
