

Announcements

- ❖ No posting of code in the forum
- ❖ Check class announcements daily

Web Services

- ❖ Web Service
 - ❖ Web API (Application Programming Interface) that can be accessed over a network and executed at a remote system
 - ❖ Allows client applications to build interfaces to the service
 - ❖ Services can range from simple requests to complicated business processes
 - ❖ Payment processing
 - ❖ Content Syndication
 - ❖ Currency conversion
 - ❖ Language translation

Web Services

- ❖ Any internet protocol can be used to build web services but HTTP and XML are often used
- ❖ By using web services, your application can publish its function to the world
- ❖ Web services can be created in any programming language
- ❖ Web services enable us to solve interoperability problems between systems by allowing data exchange between different applications and different platforms
 - ❖ With web services a company billing system can connect with a supplier server
- ❖ Web Service example
 - ❖ http://www.w3schools.com/webservices/ws_example.asp
 - ❖ Using web service example
 - ❖ http://www.w3schools.com/webservices/ws_use.asp

Web Services (Examples)

- ❖ Many popular organizations provide web services
 - ❖ Amazon → <http://aws.amazon.com/>
 - ❖ Netflix → <http://developer.netflix.com/docs>
 - ❖ Netflix Web API allows you to
 - ❖ Search movies, TV series, etc.
 - ❖ Retrieve catalog titles
 - ❖ Manage and displaying queues for users
 - ❖ eBay → <http://developer.ebay.com/products/shopping/>
 - ❖ Last.fm → <http://www.last.fm/api/intro>
- ❖ Several protocols and techniques have been developed to create and utilize web services. Two main ones:
 - ❖ REST → Representational State Transfer
 - ❖ SOAP → Simple Object Access Protocol

Web Services (REST)

- ❖ REST (Representational State Transfer)
 - ❖ Resources are represented by URLs
 - ❖ Resource → document, person, location
 - ❖ Each resource has a unique URL
 - ❖ Each resource does not need to have an actual page/document. It can be generated dynamically
 - ❖ A resource is considered a “noun”
 - ❖ Operations are performed via HTTP methods (GET, POST, PUT, DELETE)
 - ❖ Methods are considered “verbs”

Web Services (REST)

- ❖ REST → designed to operate with resource-oriented services (locate/manipulate resource)
- ❖ Example:
 - ❖ Web service that allows individuals to manage file backups
 - ❖ Each backup has an URL
<http://backupFake.doesnotexist.org/backups/1938>
 - ❖ Server responses will use XML
 - ❖ Using HTTP GET we can get the backup
 - ❖ Using HTTP PUT we can update a backup
 - ❖ Using HTTP POST we can upload a backup
 - ❖ We can receive a URL that corresponds to the new backup
 - ❖ Using HTTP DELETE we can delete a backup
- ❖ Notice that REST relies on a familiar approach (HTTP methods) to ask for services (we don't need to create a new interface/approach)

Web Services(SOAP)

- ❖ SOAP (Simple Object Access Protocol)
 - ❖ Re-termed ***Services-Oriented Access Protocol***
- ❖ SOAP → Designed to for action-oriented services (actions a web server can carry out)
- ❖ Designed as a way to package remote procedure calls into XML wrappers
- ❖ SOAP is an XML-based messaging protocol
- ❖ SOAP request
 - ❖ XML document
 - ❖ Has three components
 - ❖ Envelop → defines document as SOAP request
 - ❖ Body → provides information about the call and responses
 - ❖ Optional header and fault elements
- ❖ SOAP response is an XML document

Web Services (Platform Elements)

- ❖ WSDL (Web Services Description Language)
 - ❖ XML-based language for describing and locating web services
 - ❖ W3C standard
- ❖ UDDI (Universal Description, Discovery and Integration)
 - ❖ Directory service where companies can search and register for web services described by WSDL

Mashups

- ❖ Mashup

- ❖ Web page or application that uses and combines data and/or functionality from several sources. Sources are often based on web services

- ❖ Relies on open APIs

- ❖ Allow us to create new views of data

- ❖ Example:

- ❖ <http://chicago.everyblock.com/crime/>

- ❖ Combines crime and map information

Mashups

- ❖ Mashup Genres
 - ❖ Video and Photo Mashups
 - ❖ Mapping Mashups
 - ❖ Big player → Google Maps API
 - ❖ Search and Shopping Mashups
 - ❖ News Mashups
 - ❖ Example: Diggdot.us (combines news from Digg.com, Del.icio.us and Slashdot.org)

Mashups

- ❖ When a site does not provide an API, a mashup developer could rely on “Screen scrapping”
 - ❖ Using software tools to parse contents originally written for human consumption
- ❖ Mashup Examples:
 - ❖ Map Your Buddies →
<http://people.emich.edu/mchiang4/MapYourBuddies/>
 - ❖ Google vs. Yahoo →
 - ❖ <http://www.langreiter.com/exec/yahoo-vs-google.html>
 - ❖ Popular MashUp Listing →
<http://www.programmableweb.com/popular>

JSON vs XML

- ❖ JSON → JavaScript Object Notation
 - ❖ Data interchange format use to represent data structures
 - ❖ Text-based and human readable
- ❖ Alternative to XML
- ❖ Language Independent
- ❖ JSON example:
 - ❖ <http://www.json.org/example.html>

Traditional Server/Client Interaction

- ❖ Nothing happens until we submit data
- ❖ We must wait until the server request is processed (can do anything with the page)
- ❖ A page must be completely loaded even if most of the content identical to previous page
- ❖ Compare with a desktop application
- ❖ Can we do better? Can the page be updated without requiring a page load?
- ❖ AJAX is the answer

AJAX

- ❖ AJAX → Asynchronous JavaScript and XML
- ❖ Combination of technologies
- ❖ Adds a layer between the browser and the web server, handling server requests and processing the results
 - ❖ Layer Name → Ajax Framework/Ajax Engine
- ❖ The requests are not synchronized with user actions (e.g., clicking on links, buttons, etc.) User can continue interacting with the browser while request is being processed

AJAX

- ❖ In the traditional client/server model we submit server requests by clicking on a link or via submit (this generates the HTTP request for us)
 - ❖ Notice we get as a result a new web page
- ❖ **XMLHttpRequest**
 - ❖ JavaScript object that will issue the HTTP request
 - ❖ No page load is generated as a result of the request
 - ❖ Can only issue request to URLs within the same domain
 - ❖ Cannot directly access a remote server
- ❖ There is nothing the server needs to do just because the request is associated with AJAX. The server is just receiving an HTTP request
- ❖ AJAX application just care about receiving an HTTP response
- ❖ **EXAMPLE:** directoryLookup.html, directory.php, processMemo.php
- ❖ <http://www.turtle.cs.umd.edu/classes/122/AjaxExample/directoryLookup.html>

AJAX Reference

- ❖ “Ajax in 10 Minutes” by Phil Ballard, ISBN 0-672-32868-2

References

- ❖ <http://www.w3schools.com/webservices/>
- ❖ http://en.wikipedia.org/wiki/Mashup_%28web_application_hybrid%29
- ❖ <http://www.ibm.com/developerworks/xml/library/x-mashups.html>