

1.264 Lecture 19

Web database: Forms and controls

We continue using Web site Lecture18 in this lecture

Next class: ASP.NET book, chapters 11-12. Exercises due after class

Forms

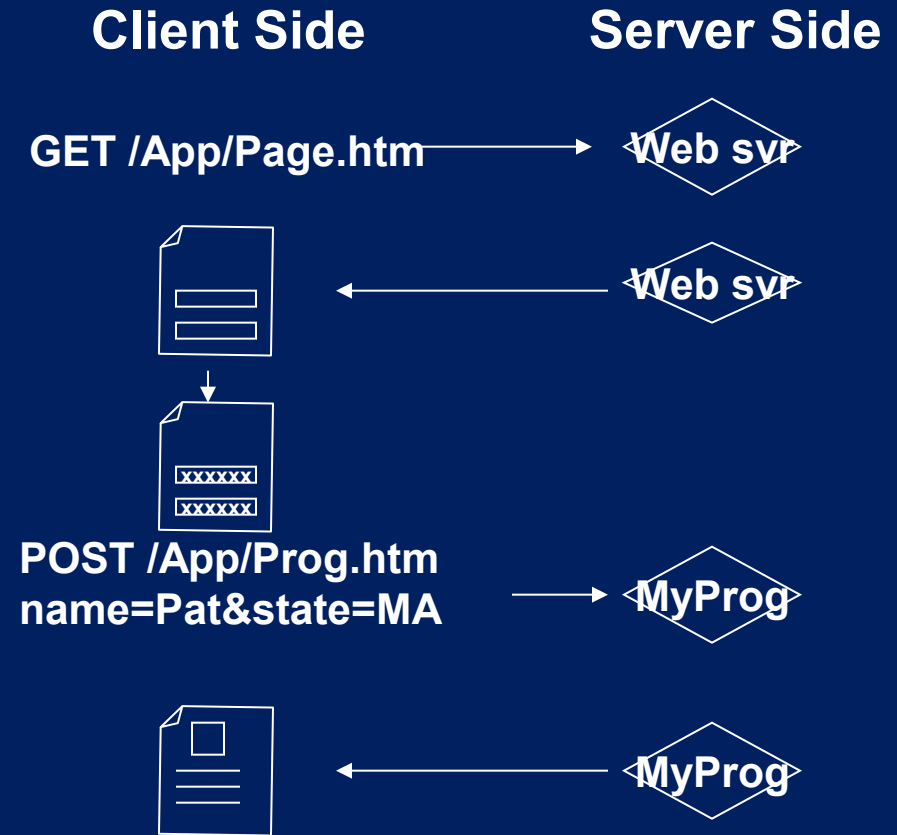
- **Web server and its pages are a communications channel between databases**
- **We first create communications between a user and a database by using forms**
 - **More visible and intuitive, use the same principles**
 - **Based on XHTML pages and HTTP**
- **Database-database communications will be covered next**
 - **Based on XML and HTTP, packaged in Web services**
 - **Machine-readable, can be validated by machines**
 - **Some human review is still used because there isn't complete trust among trading partners, engineering collaborators, etc.**
 - **Machine-machine communications create SQL database queries to fill in 'XML forms' and send them to the database**

How XHTML forms transmit data

- Forms allow a series of controls to be placed on the page
 - Controls are text boxes, dropdowns, radio buttons, check boxes...
 - Each control has a name and a value
- Form data is sent when user presses 'Submit' button. Options:
 - Data is sent to URL with HTTP POST command as string of form:
 - Name1=Value1&Name2=Value2&...NameN=ValueN
 - POST data is string sent after blank line after HTTP headers
 - Data is sent to URL with HTTP GET command, appended to end of GET string after a ?:
 - GET/Index.html?Name1=Value1&Name2=Value2& ...
 - Data is sent to Web services
 - POST followed by an XML document
 - We cover XML later: it allows validation and business rules
- Form's 'Submit' button is associated with the URL of an application server method that will process the input data
 - Server programs have methods (function points) to extract the data from the POST or GET command and use it in the program
 - We counted Web pages and estimated function points earlier

Browser-server forms interaction

- Step 1: Browser requests form without any parameters
- Step 2: Server fetches and returns input form
- Step 3: User fills out or edits input form and presses 'submit' button
- Step 4: Browser packages form into query string and sends to server
- Step 5: Program synthesizes a response document and returns it

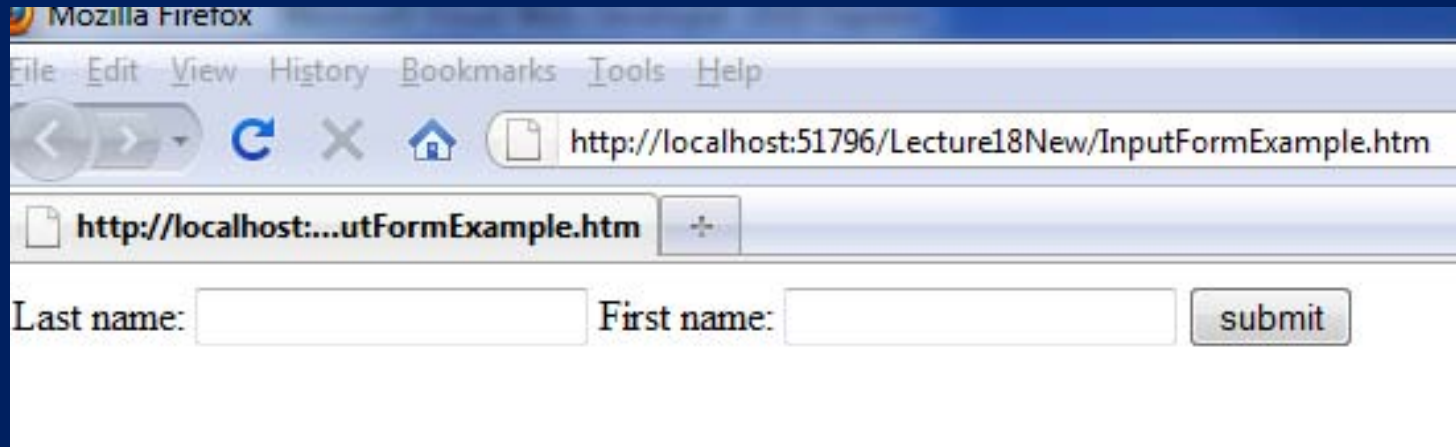


Forms also support dynamic controls on an xhtml (aspx) page, such as .NET's SqlDataSource, GridView, etc.

In .NET, all pages that return dynamic results are .aspx pages and are forms

Example XHTML input form

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
</head>
<body>
<form action="inputForm.aspx" method="post">
  Last name: <input type="text" name="lName" />
  First name: <input type="text" name="fName" />
  <input type="submit" value="submit" />
</form>
</body>
</html>
```



Review: Set SQL Server database access

- **Create login for SQL Server in SSMS if you haven't:**
 - In Object Explorer, open Security-> Logins
 - Right click, select 'New login'
 - Select SQL Server authorization; enter name, password
 - Select default database= MIT1264
 - Check all server roles, check user mapping for MIT1264

Review: Create Web site, database connection

- In VSW, use Lecture 18 site from last class
 - Delete Default.aspx, Site.master. We'll replace them.
- Download Lecture18 zip file if you didn't already
 - Unzip it in a temporary directory and then move its files to the Lecture18 Web site at C:\inetpub\wwwroot\Lecture18
 - MIT.master has common graphics for each page and a placeholder for active content
 - Images folder contains graphics used in MIT.master
 - Web.sitemap contains navigation structure of your site
- Refresh the solution explorer after moving files into site
- Tools-> Connect to Database
 - Server: .\SQLEXPRESS or SQLEXPRESS
 - Login name and password you set up earlier
 - Database name: MIT1264
 - Test the connection
- Look at database in Database Explorer in VWD

Review: Create XHTML form (.aspx page)

- **View->Toolbox**
 - To make toolbox appear along left margin of page
- **File-> New File-> Web Form: Orders.aspx**
 - Check master page and select MIT.master
 - Uncheck 'code in separate file'
- **In Design view, from the Toolbox**
 - Drag a SqlDataSource control into the ContentPlaceHolder
 - Click the arrow at its top right, then on configure source
 - Select connection you already created
 - Select Orders table
 - Check * to get all columns, ORDER BY Cust
 - Test query and finish
 - Drag a GridView control on top of the SqlDataSource
 - Choose data source
 - Check 'enable sorting' and 'enable paging'
 - Add header 2 <h2> with "Orders". Save the page
 - Use the browser symbol on the toolbar to test the page

Exercise 1: XHTML Customers page

- **Create a Web page to display the Customers table: Customers.aspx. Same steps as before.**
 - Use MIT.master for its appearance
 - Enable sorting and paging
- **Create Default.aspx with new XHTML that uses MIT.master**
 - Keep the name Default.aspx
 - Place short text in the ContentPlaceHolder on the page
- **Change links in MIT.master to point to your Orders, Customers pages, not Default**
- **When done, test with the browser**
 - Start at Default.aspx
 - Navigate to Orders and Customers and then back to Default
 - In browser, “View Source” to see what XHTML has been generated

Solution 1

MIT 1.264/ESD.264

Data Internet & Sys Integ Tech

[Home](#) > [Customers](#)

Customers

| CustNbr | Company | CustRep | CreditLimit |
|---------|------------------------|---------|--------------|
| 211 | Connor Co | 89 | 50000.0000 |
| 522 | Amaratunga Enterprises | 89 | 75000.0000 |
| 890 | Feni Fabricators | 53 | 1000000.0000 |

[More info](#)

Copyright © 2013 MIT 1.264/ESD.264

Exercise 2: Display, edit and delete data

- **File -> New File-> Web Form: Customers2**
- **Drag SqlDataSource into ContentPlaceHolder**
 - **SELECT * FROM Customers ORDER BY Company**
 - **Click 'Advanced' and generate edit, update, delete SQL**
 - **Test and save**
- **Drag GridView onto SqlDataSource**
 - **Set data source**
 - **Enable paging, sorting, edit, delete**
- **Put a header <h2> at the top of the ContentPlaceHolder: "Customers"**
- **Save and test: edit, delete**
 - **Can't delete a customer if they have an order**
 - **Insert not present in GridView**

Solution 2

MIT 1.264/ESD.264

Data Internet & Sys Integ Tech

[Home](#) > [Customers](#) > Customers

Customers

| | CustNbr | Company | CustRep | CreditLimit |
|---|---------|------------------------|---------|--------------|
| Edit Delete | 522 | Amaratunga Enterprises | 89 | 75000.0000 |
| Edit Delete | 211 | Connor Co | 89 | 50000.0000 |
| Edit Delete | 890 | Feni Fabricators | 53 | 1000000.0000 |

Copyright © 2013 MIT 1.264/ESD.264

Exercise 3: Display, edit, insert, delete data

- **File -> New File-> Web Form: SalesReps**
- **Drag SqlDataSource into ContentPlaceHolder**
 - **SELECT * FROM SalesReps ORDER BY Name**
- **Drag GridView onto SqlDataSource**
 - **Enable paging, sorting, selection**
- **Drag a second SqlDataSource into ContentPlaceHolder, below the first one**
 - **SELECT * FROM SalesReps WHERE ...**
 - **Click WHERE button: RepNbr, =, Control, GridView1, Add**
 - **Click Advanced button: Generate INSERT,UPDATE,DELETE**
- **Drag FormView control onto 2nd SqlDataSource**
 - **Choose second SqlDataSource as data source**
- **Add <h2>, save, and test in browser**

Exercise 3, cont: Display, edit, insert, delete

- **GridView doesn't update when you change data. To make it do so:**
 - In Design view, click on FormView
 - Click Events button (lightning bolt) in Properties window
 - Double-click ItemInserted event
 - Add following code
 - `GridView1.DataBind();`
 - Then double-click ItemDeleted event and add same code
 - And double-click ItemUpdated event and add same code
- **Save and test your page**
 - Add Title at top of all the pages you created

Solution 3

MIT 1.264/ESD.264

Data Internet & Sys Integ Tech

[Home](#) > [SalesReps](#)

Sales Reps

| | RepNbr | Name | RepOffice | Quota | Sales |
|------------------------|--------|------------|-----------|-------------|-------------|
| Select | 53 | Bill Smith | 1 | 100000.0000 | 0.0000 |
| Select | 89 | Jen Jones | 2 | 50000.0000 | 130000.0000 |

RepNbr: 53

Name: Bill Smith

RepOffice: 1

Quota: 100000.0000

Sales: 0.0000

[Edit](#) [Delete](#) [New](#)

[More info](#)

Copyright © 2013 MIT 1.264/ESD.264

Data transmittal across Web

- **These pages request a response from the server**
 - Your Web browser sends an HTTP POST request to the Web server
 - URL is, e.g., localhost/Customers.aspx
 - .aspx page is a program that dynamically generates the response xhtml (or XML) page
 - In your browser:
 - Right click, select “View Source” on result page
 - It is an xhtml page; the logic on the .aspx page has been executed and the response returned to your browser
- **Note that we are dragging and dropping controls, without doing any programming**
 - This will probably become the standard way to build business systems: it can be done by analysts, not IT
 - Specialized controls for supply chain, etc.

HTTP and XHTML

- **HTTP**
 - Is only direct form of interaction between browser and server
 - Was an extremely perceptive extension of email, ftp protocols by Tim Berners-Lee to enable Web browsers
 - Request-response paradigm
 - Connection made for each request/response pair
 - Core Web protocol, very stable
- **XHTML**
 - Page definition language, based on tags
 - High level page description, not specific formatting
 - Forms used to enter and submit data to Web server, and to invoke dynamic Web pages on Web server

MIT OpenCourseWare
<http://ocw.mit.edu>

1.264J / ESD.264J Database, Internet, and Systems Integration Technologies
Fall 2013

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.