CSI2441: Applications Development

Lecture 9

Layered Architecture, Multi-Page Webflows and Session Usage

Learning Objectives

- Introduce an overview of sessions from an ASP.Net perspective
- To understand the model, view and controller layers of a web application
- To be able to manage the routing of a webflow
- To understand the role of the HTTP request and response in a web form
- To be able to maintain user sessions
- To be able to apply the data transfer object and façade
- To discuss some more advanced uses of sessions in web applications

State Management in Web Applications

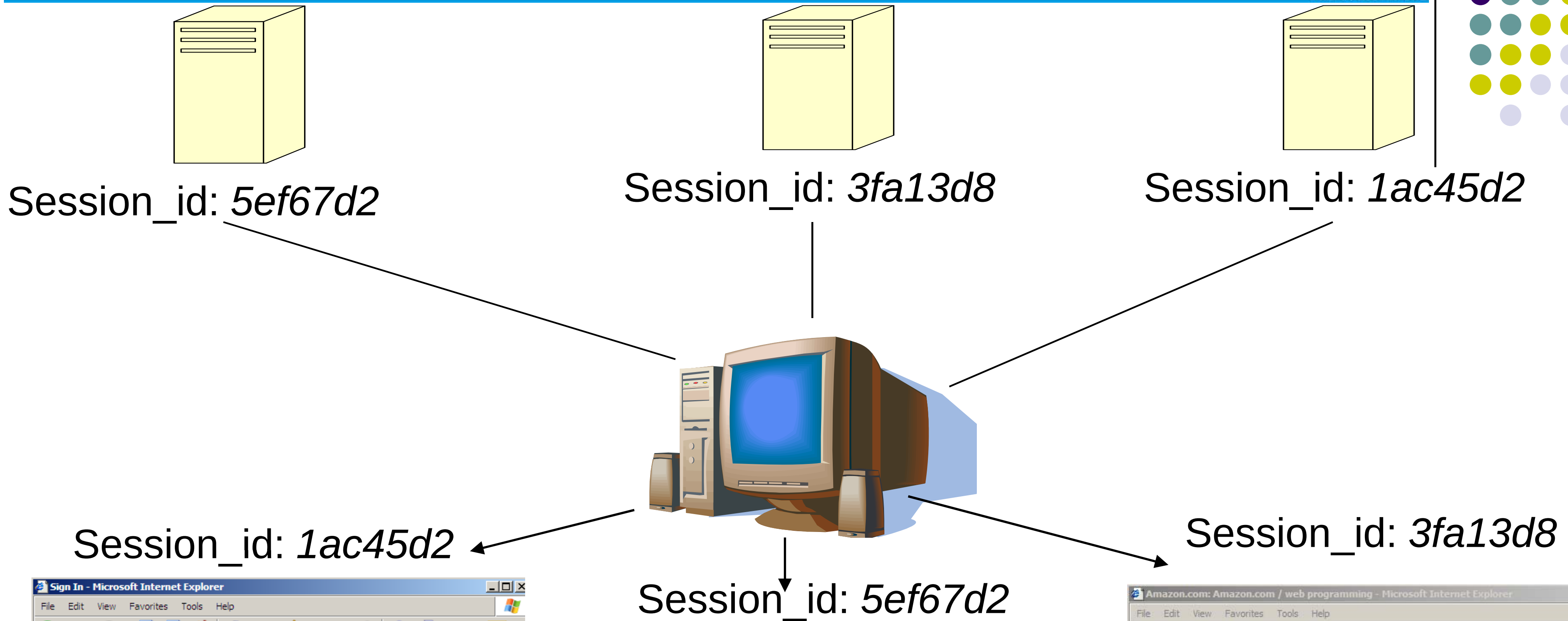
- State management in web applications is quite different to desktop-based development environments
- When developing an application that runs entirely on a single machine (like a Windows or Linux box), variables are defined and values stored in the local memory of that machine
- Typically, it is one user on one machine using one instance of the software
- The state of the software stays constant, in local memory, with the data on screen being that item that changes most often
- Until explicitly removed, variables stay in memory, items shown on screen remain until specifically removed

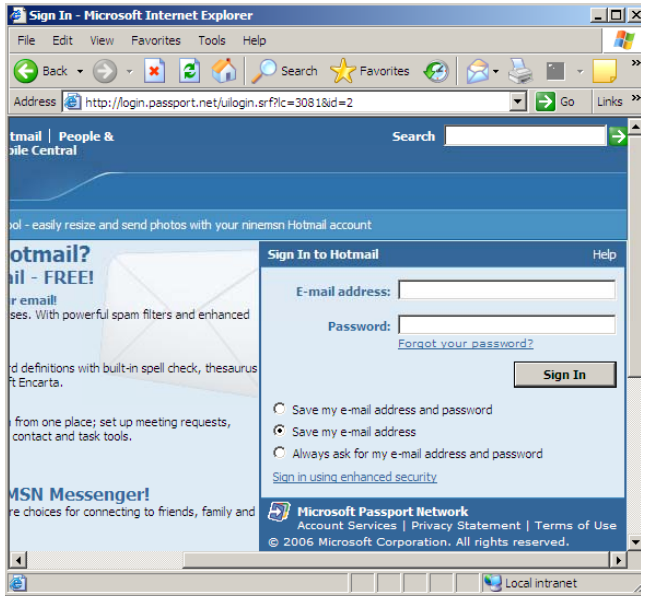
State Management in Web Applications

- In web applications, the code that creates the interface and interacts with data sources resides on a server, while the client user sits on a client-side machine
- Any interaction with the application by the user requires the web application to re-load and create the on-screen pages again according to the instructions (or information events) specified by the user
- Any on-screen settings of the previous screen will be lost in this process unless the values can be specifically stored and then recalled
- Because a web application might have tens, hundreds or thousands of users online at once, the server must be able to identify which user is currently in what particular state of interaction with the web application

Sessions

- Sessions are typically random 32 character alphanumeric values generated by the server-side scripting language in conjunction with the web-browser client
- Each open browser window, if connecting to the same or different servers, can have its own unique session value
- Obviously, you might be reading email in one browser window, news in another doing some online shopping in the other
- Though the incoming HTTP data stream comes in on a single ‘pipe’, the session data attached to the incoming packets can direct the correct HTML data to the correct browser window for rendering to screen




Sign In - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://login.passport.net/uilogin.srf?lc=3081&id=2

tmail | People & bile Central — Search

ol - easily resize and send photos with your ninemsn Hotmail account

otmail? il - FREE! r email! ses. With powerful spam filters and enhanced

d definitions with built-in spell check, thesaurus t Encarta.

from one place; set up meeting requests, contact and task tools.

MSN Messenger! re choices for connecting to friends, family and

Sign In to Hotmail — Help

E-mail address:

Password:

Forgot your password?

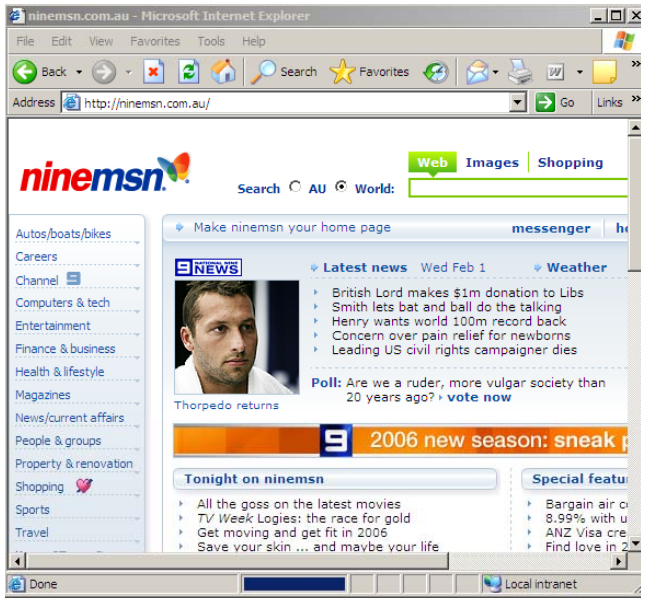
Sign In

Save my e-mail address and password
[✓ Save my e-mail address]
Always ask for my e-mail address and password

Sign in using enhanced security

Microsoft Passport Network
Account Services | Privacy Statement | Terms of Use
© 2006 Microsoft Corporation. All rights reserved.

Local intranet


ninemsn.com.au - Microsoft Internet Explorer

Address http://ninemsn.com.au/

ninemsn — Web Images Shopping — Search AU [✓ World:]

Autos/boats/bikes
Careers
Channel 9
Computers & tech
Entertainment
Finance & business
Health & lifestyle
Magazines
News/current affairs
People & groups
Property & renovation
Shopping
Sports
Travel

Make ninemsn your home page — messenger

9 NEWS — Latest news Wed Feb 1 — Weather

- British Lord makes $1m donation to Libs
- Smith lets bat and ball do the talking
- Henry wants world 100m record back
- Concern over pain relief for newborns
- Leading US civil rights campaigner dies

Thorpedo returns

Poll: Are we a ruder, more vulgar society than 20 years ago? › vote now

2006 new season: sneak

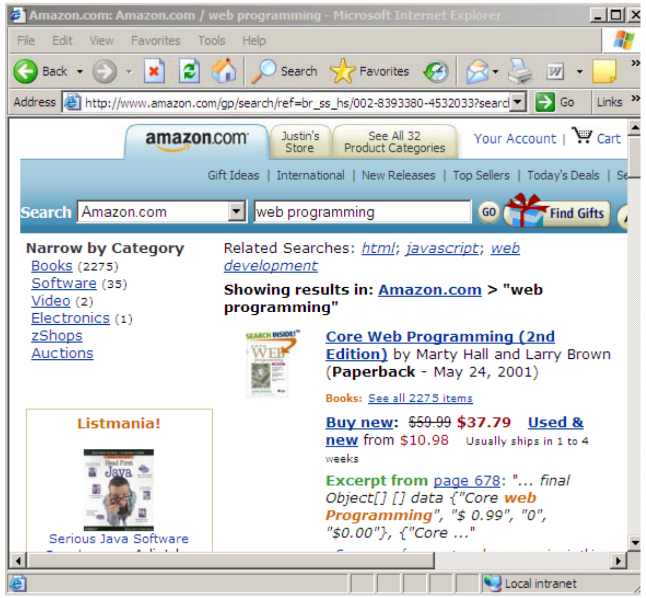
Tonight on ninemsn

- All the goss on the latest movies
- *TV Week* Logies: the race for gold
- Get moving and get fit in 2006
- Save your skin ... and maybe your life

Special featu

- Bargain air c
- 8.99% with u
- ANZ Visa cre
- Find love in 2

Done — Local intranet


Amazon.com: Amazon.com / web programming - Microsoft Internet Explorer

Address http://www.amazon.com/gp/search/ref=br_ss_hs/002-8393380-4532033?searc

amazon.com — Justin's Store — See All 32 Product Categories — Your Account | Cart

Gift Ideas | International | New Releases | Top Sellers | Today's Deals | Se

Search Amazon.com — web programming — GO — Find Gifts

Narrow by Category
Books (2275)
Software (35)
Video (2)
Electronics (1)
zShops
Auctions

Listmania!

Serious Java Software

Related Searches: *html*; *javascript*; *web development*

Showing results in: Amazon.com > "web programming"

Core Web Programming (2nd Edition) by Marty Hall and Larry Brown (Paperback - May 24, 2001)

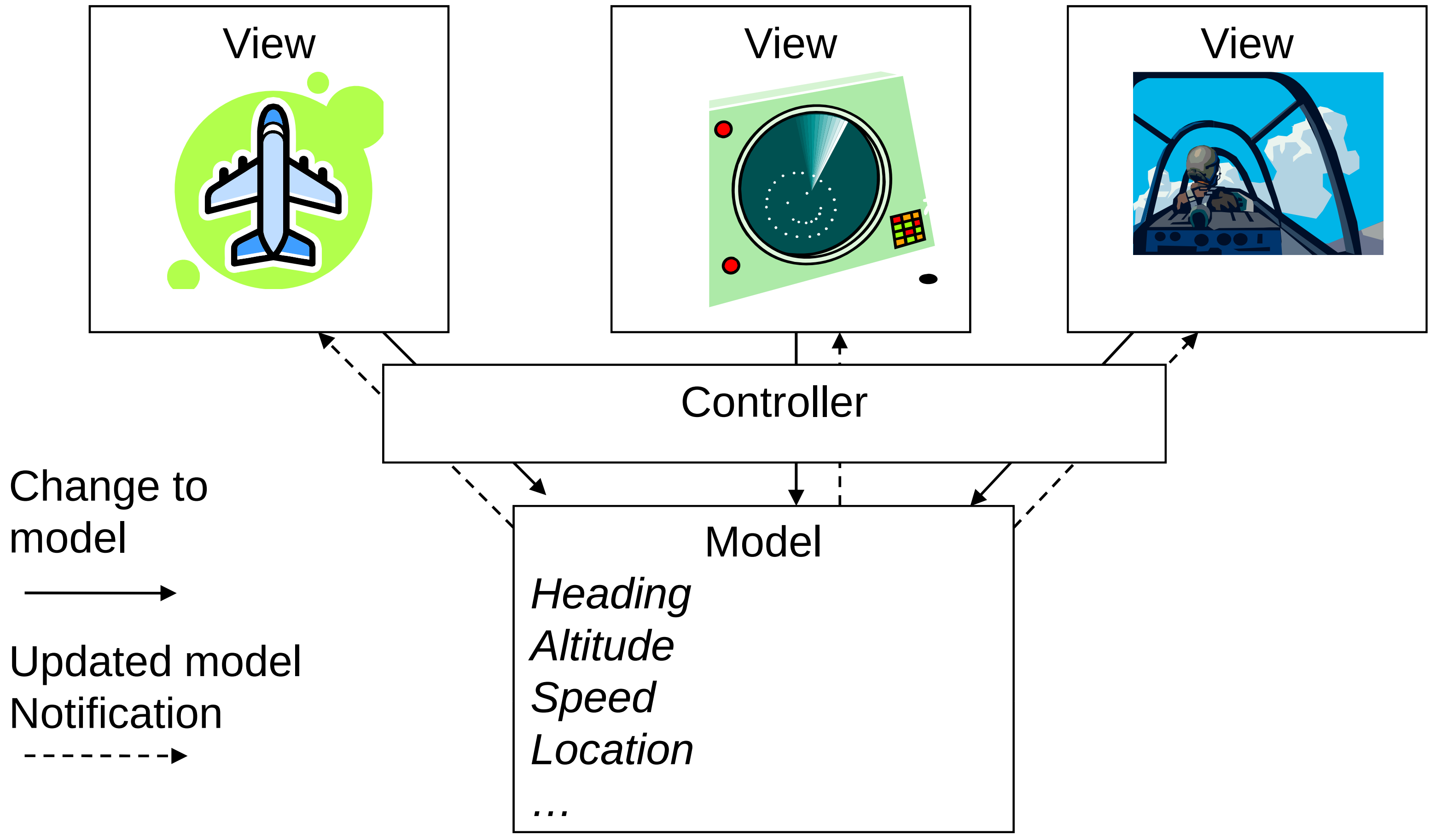
Books: See all 2275 items

Buy new: ~~$59.99~~ $37.79 Used & new from $10.98 Usually ships in 1 to 4 weeks

Excerpt from page 678: "... *final Object[] [] data {"Core web Programming", "$ 0.99", "0", "$0.00"}, {"Core ...*"

Local intranet

Model-View-Controller (MVC) Architecture

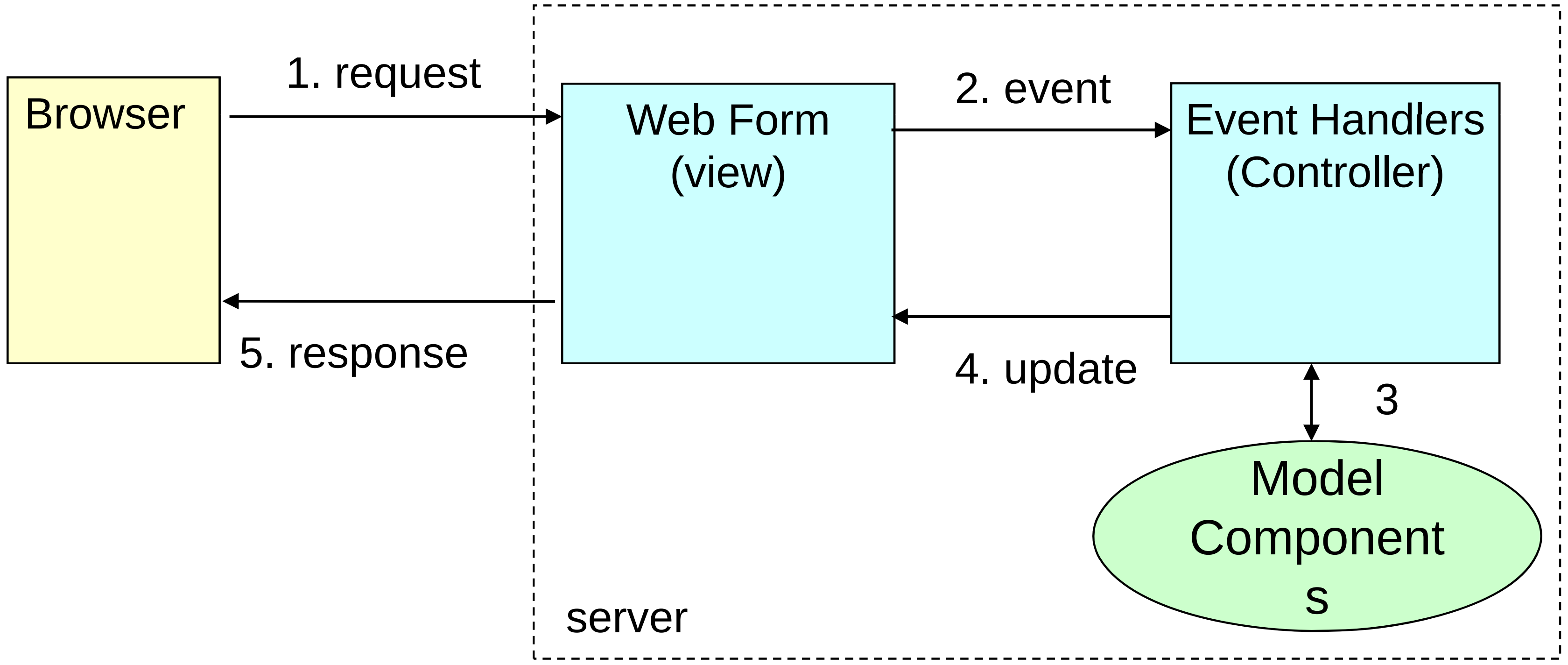


MVC Roles

- Model
  - The underlying data
- View
  - What is presented to the user
- Controller
  - The handling component that manages user interaction and triggers appropriate updates to the model
- Web application context
  - Multiple views will be different clients with separate browsers looking at the same application in different ways

Web Application Architecture

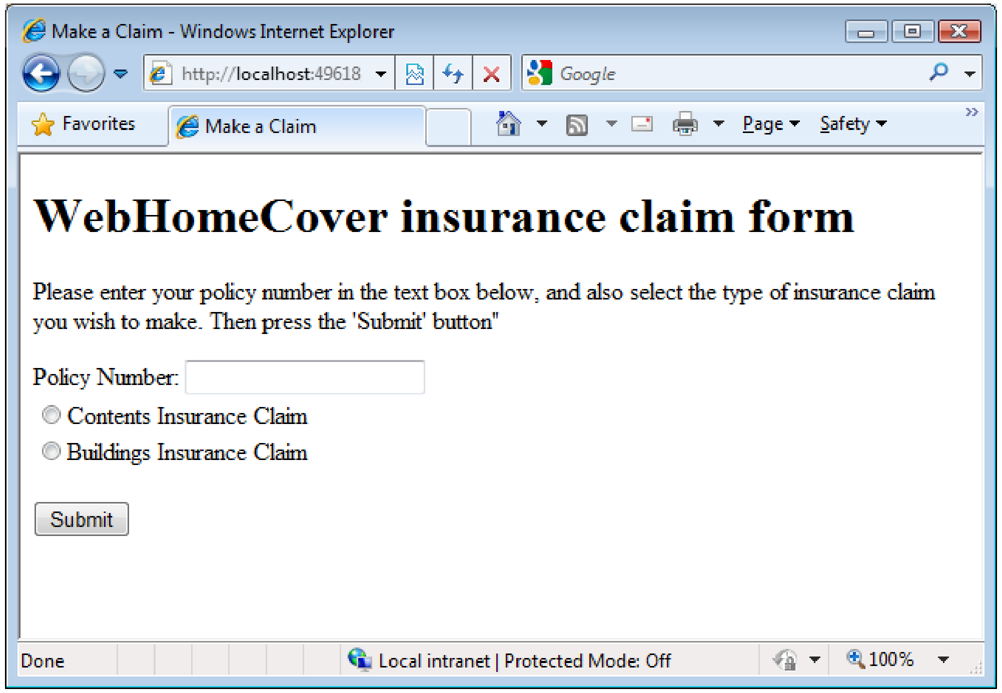
- A .NET web application uses Web Forms for the view and event handlers as the controller layer



Building a Simple Webflow

- We will create a simple multi-form webflow from one of our insurance system use cases
  - a customer lodging an insurance claim
- Two separate web forms for data collection
  1. Policy number and policy type
  2. Claim amount and description

The Two Web Forms



Make a Claim - Windows Internet Explorer

http://localhost:49618

WebHomeCover insurance claim form

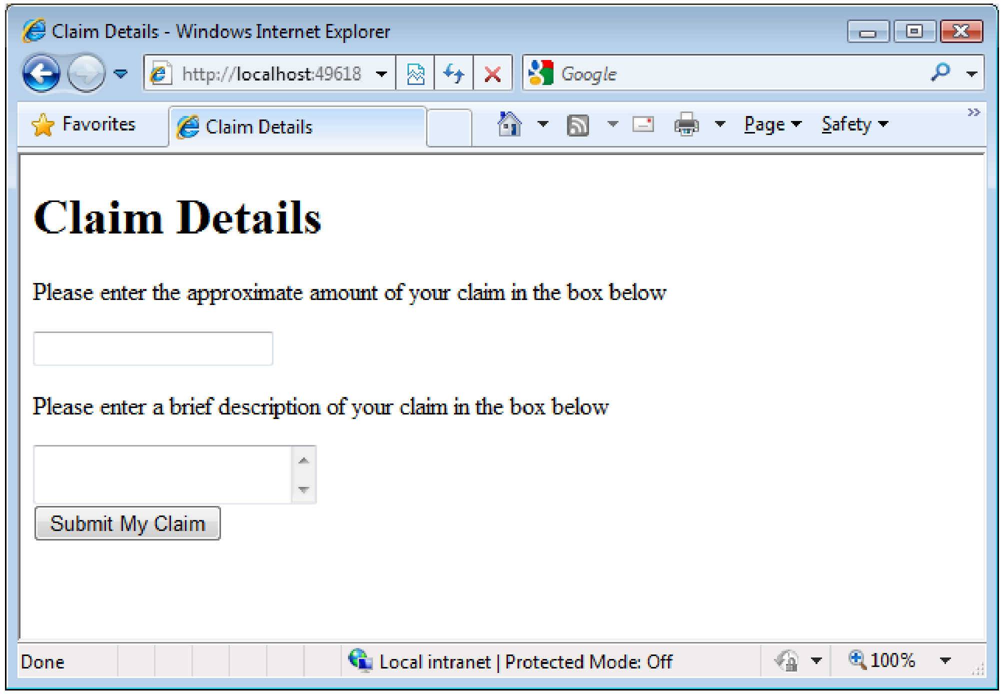
Please enter your policy number in the text box below, and also select the type of insurance claim you wish to make. Then press the 'Submit' button"

Policy Number:

○ Contents Insurance Claim

○ Buildings Insurance Claim

ClaimForm1.aspx



Claim Details - Windows Internet Explorer

http://localhost:49618

Claim Details

Please enter the approximate amount of your claim in the box below

Please enter a brief description of your claim in the box below

ClaimForm2.aspx

Linking Pages in a Webflow

- At the moment each form posts back to itself
- We can change this behavior by using the Server.Transfer method in an event handler

```
Server.Transfer(“webform.aspx"")
```

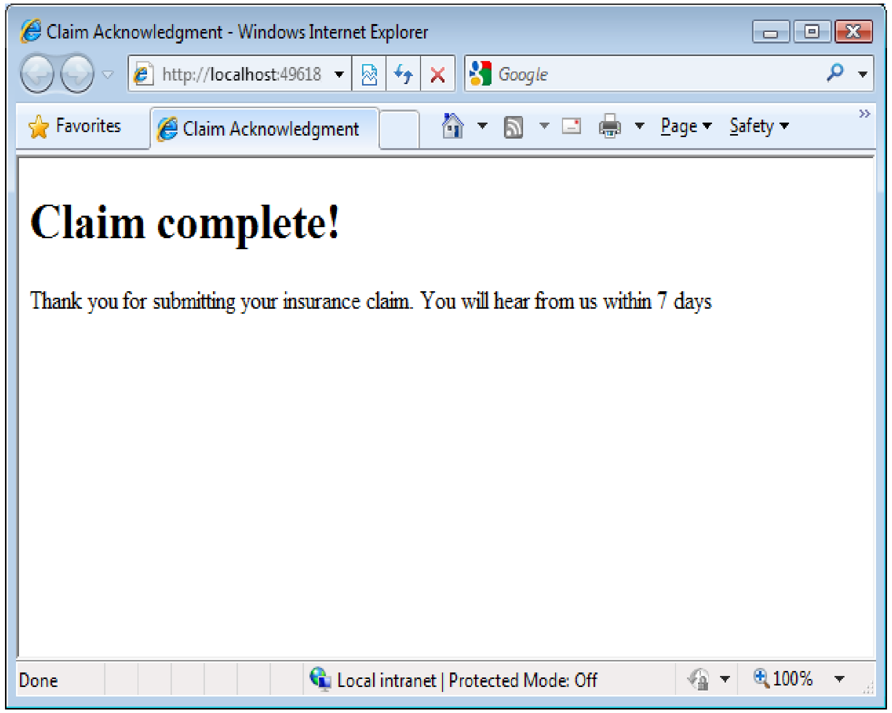
- In this example we transfer to ClaimForm2.aspx in the button event handler for ClaimForm1.aspx

```
Server.Transfer("ClaimForm2.aspx")
```

Final Page in the WebFlow

- We will also add a final page, ClaimForm3.aspx, to acknowledge receipt of the claim
- The event handler for ClaimForm2.aspx will transfer to this page

```
Server.Transfer("ClaimForm3.aspx")
```


Claim Acknowledgment - Windows Internet Explorer
http://localhost:49618
Google
Favorites
Claim Acknowledgment
Page
Safety
Done
Local intranet | Protected Mode: Off
100%

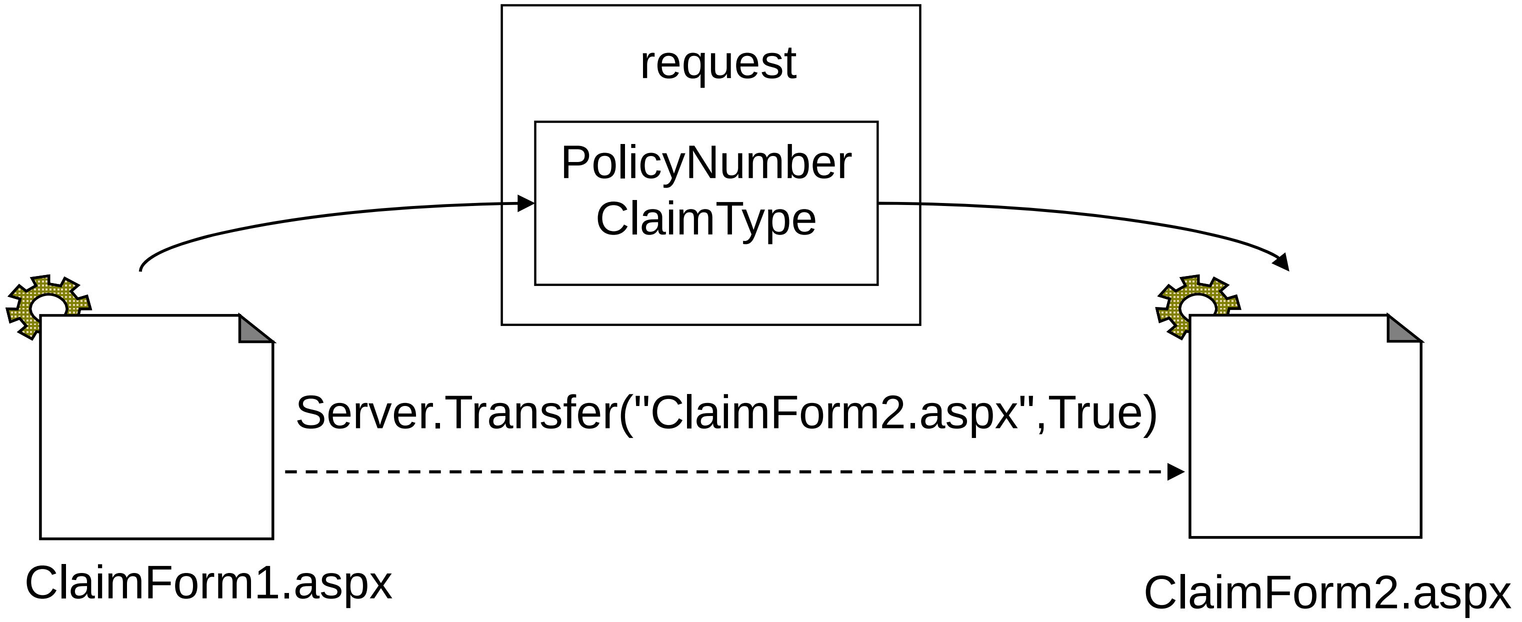
Claim complete!

Thank you for submitting your insurance claim. You will hear from us within 7 days

Request Parameters

- The second Boolean parameter to the Transfer method ensures that request data is maintained if set to ‘true’

```
Server.Transfer("ClaimForm2.aspx", True)
```



Redirecting

- Instead of transferring to another page, we can redirect
- Redirecting creates a new request to the page so it is not possible to maintain request parameters using this method

```
Response.Redirect("ClaimForm2.aspx")
```

Page Event Handlers

- The ‘Page Load’ event occurs when the web page is first loaded
- Can be used to write dynamic content to the page when it is first displayed
- We will use the Page Load event handler display the user’s policy number and policy type, from ClaimForm1, on ClaimForm2

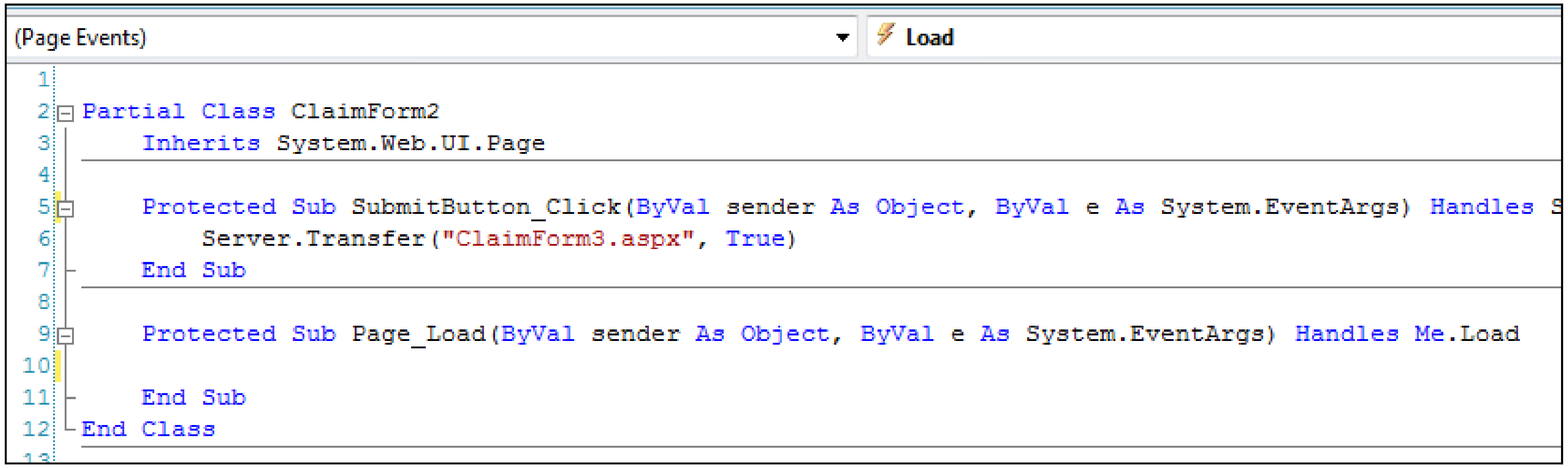
Labels on ClaimForm2

- These labels (‘Policy’ and ‘Type’) will be populated by the Page Load event in ClaimForm2

```
<p>Your policy number: <strong>
<asp:Label ID="Policy" runat="server" Text=""></asp:Label>
</strong>
<br />
Your policy type: <strong>
<asp:Label ID="Type" runat="server" Text=""></asp:Label>
</strong>
</p>
```

C# and VB Page Event Handlers

- In C#, the page event handler is automatically provided
- In VB, you need to select '(Page Events)' from the list on the top left, and then select the 'Load' event from the list on the right



```

Partial Class ClaimForm2
    Inherits System.Web.UI.Page

    Protected Sub SubmitButton_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles S
        Server.Transfer("ClaimForm3.aspx", True)
    End Sub

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

    End Sub
End Class
```

Retrieving Request Parameters

- In the page load event handler, we need to retrieve the request parameters and populate the labels
- In VB, use the Request.Item method

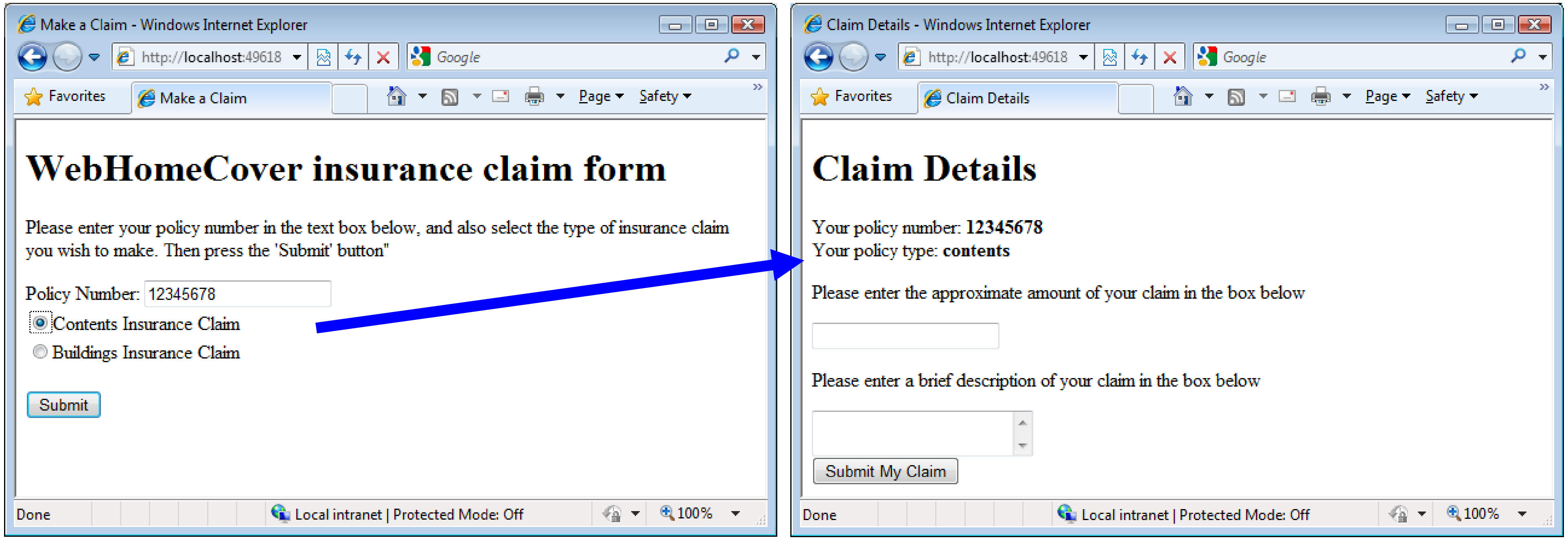
```
Policy.Text = Request.Item("PolicyNumber")
Type.Text = Request.Item("PolicyType")
```

- In C#, use this array-like syntax

```
Policy.Text = Request["PolicyNumber"];
Type.Text = Request["PolicyType"];
```

Webflow with Request Parameters

- The data from the first page is displayed when the second form loads
- A vast majority of web apps rely on multi-page logic and flows to achieve business processes



WebHomeCover insurance claim form

Please enter your policy number in the text box below, and also select the type of insurance claim you wish to make. Then press the 'Submit' button"

Policy Number: 12345678

[✓ Contents Insurance Claim]

Buildings Insurance Claim

Claim Details

Your policy number: **12345678**
Your policy type: **contents**

Please enter the approximate amount of your claim in the box below

Please enter a brief description of your claim in the box below

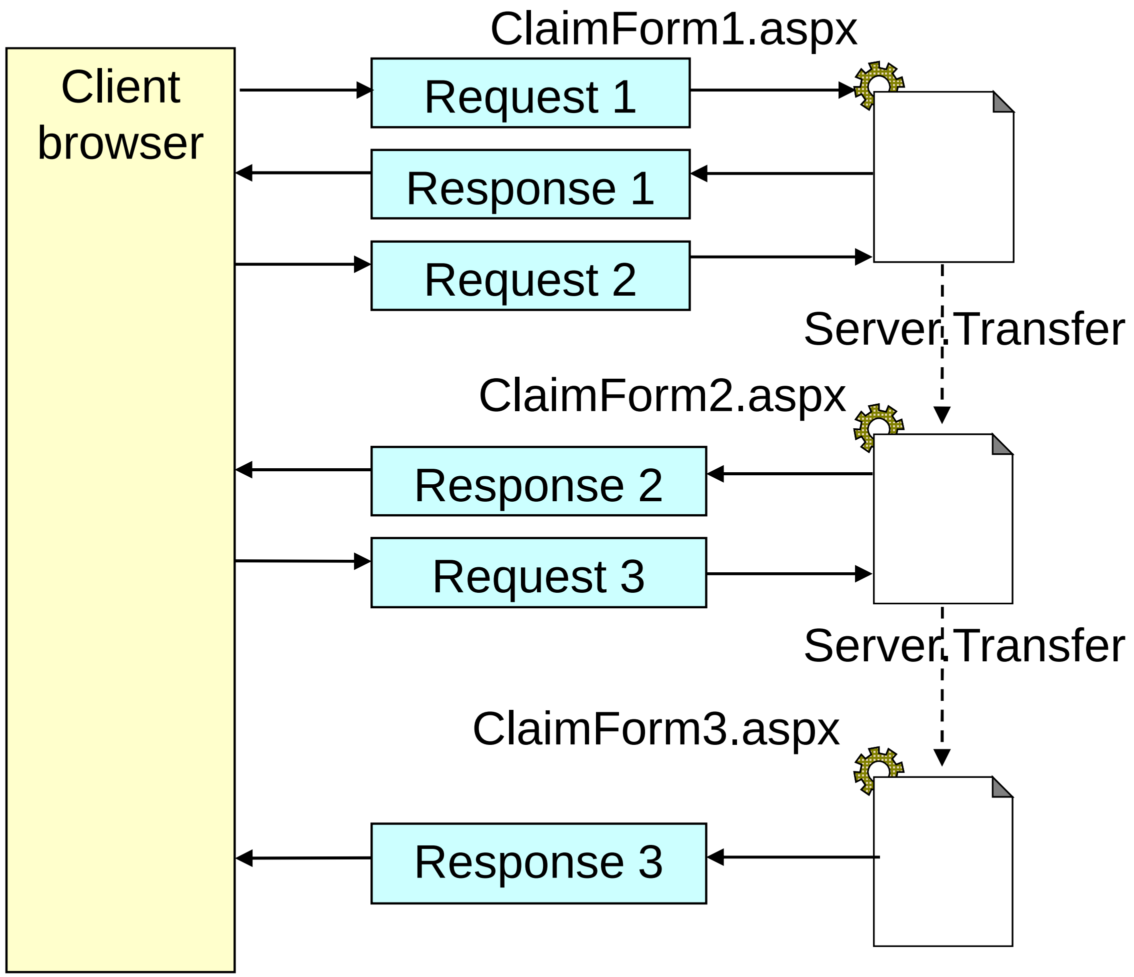
Processing Form Data in the Controller Layer

- There are one or two processes that may take place in the controller layer, such as data conversion, and validation
- Since all the parameters are returned from the 'request' object as Strings, we sometimes need to perform conversion processes to change these Strings into more suitable data types

```
double valueOfClaim = 0.0;
try
{
 valueOfClaim = Double.parseDouble(claimValue);
}
catch(NumberFormatException e)
{
 e.printStackTrace();
}
```

The Insurance Claim Webflow

- The initial request goes to 'ClaimForm1.aspx'
- The first response is simply the page generated by the web form
- The next request is also to 'ClaimForm1.aspx'
- After that, the button event handlers transfer between pages
- This type of webflow requires some session management

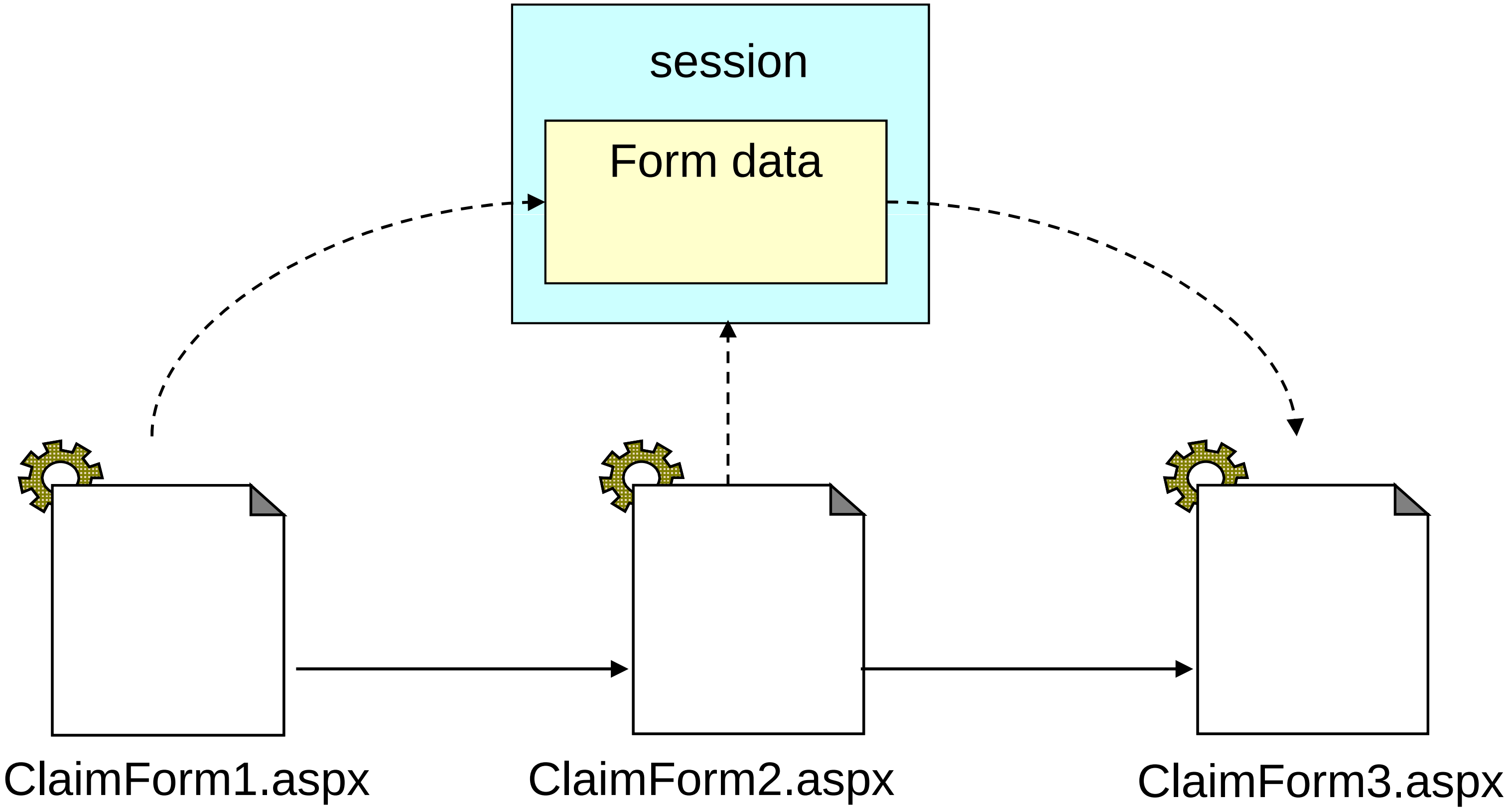


Session Management

- HTTP is a stateless protocol
  - Does not maintain the connection after a request/response cycle
- We need to store client session state on the server
- Use the ‘Session’ object in event handlers
- A session is effectively a conversation between a client and a server that includes multiple request / response cycles
- A session could be a shopping process at an on-line store, a series of interactions with a share trading system, or any other application where the client needs to maintain some state over a series of different web pages

Sessions

- A ‘Session’ can be used to maintain client form data over a series of pages in a webflow
- Essentially it is a distributed variable



The Session Object

- A ‘Session’ object is basically like a Hash table in that it can contain a collection of key-value pairs, where the keys are Strings and the values are Objects
- This is very similar to the way that data is stored in the Request object
- A session object will be created for a client automatically when it is accessed by an event handler
- It can also be useful to use the unique session id as an identifier for records stored on a database (such as items in a shopping cart)

Setting Session Attributes

- In VB, set the items in a session using the ‘Item’ method
- To set an item, assign a value to a String key

```
Session.Item("PolicyNumber") = PolicyNumber.Text
Session.Item("PolicyType") = PolicyType.Text
```

- C# uses the same array-like syntax used with the Request

```
Session["PolicyNumber"] = PolicyNumber.Text;
Session["PolicyType"] = PolicyType.Text;
```

Getting Session Attributes

- In VB, get the items in a session using the ‘Item’ method, passing the key as a parameter

```
Policy.Text = Session.Item("PolicyNumber")
```

- C# again uses the array-like syntax
- Type casting is required on the object returned from the session

```
Policy.Text = (String)Session["PolicyNumber"];
```

Session Management

- One important aspect of session management is freeing up resources that are no longer required

```
Session.Abandon()
```

- At the beginning of a webflow we may want to check if the session that we have acquired from the request is new
- We can send the client back to start the webflow from the beginning by forwarding or redirecting

```
if(Session.IsNewSession)
{
Response.Redirect("ClaimForm1.aspx");
}
```

```
If Session.IsNewSession() Then
Response.Redirect("ClaimForm1.aspx")
End If
```

Other Session Methods

- Getting the value of a session item leaves the item in the session, while abandoning the session destroys it completely
- Sometimes we need to actually remove an individual item from the session

```
Session.Remove("itemKey")
```

- Similarly we can remove all of the keys and values from a session (both have the same effect)

```
Session.RemoveAll()
```

or

```
Session.Clear()
```

- These methods empty the session but do not abandon it

Sessions for User Management

- As the previous example illustrated, sessions can be used to store and keep an number of values
- In any system that requires users to log in, sessions can be used to give them access to members-only areas of a web application / site
- Their name, userid and access level can all be stored upon login and follow the user around the site
- Records of what the user does and where they go in the site can also be recorded

Sessions for Security

- The first line of security in web applications that use sessions is to check if a session exists before any information is shown
- If no session exists for a client trying to load a page, then that user can be automatically re-directed to the login page
- In ASP.Net the Login Controls can be used to manage this automatically, including user-level management
- For novice developers, session and multi-level user management can be quite frustrating at first

Sessions for Security

- Sessions are often used to store record id values for a logged in user
- Instead of the value being passed as a hidden form field or as part of the url, it can be stored in the session where only the application can see it
- That record id can then be used on every page where information about the user is required – with the record id being used to query the user database(s)
- It is a trade off between extracting the data during login and having several sessions running at once with all the necessary user data in them, versus just a single session, but more calls to the database system

Logging Users

- Because of the stateless nature of the web, it can be difficult to track users once they are in a site
- Using sessions, we can assign each person within a website a session id, and as they move around the site, we can record this session and the details about it
- For example;
  - The web page name
  - The time of the visit
  - Time spent on a page
  - Items looked at within a page (such as products)

Logging users

- The logging of users for statistical purposes is sometimes referred to in the literature as the Clickstream or Clickstreaming
- The difference between web server logs and Clickstream data is that the latter allows site operators to follow an anonymous user from their entry point into a site through to their exit
- Also, web server logs typically record large amounts of info that go into linear text files, whist Clickstream data is much more specific and goes directly into a database for rapid extraction and manipulation
- Obviously, a busy site could generate extremely large data files and require a lot of concurrent sessions

Clickstreaming

- So why record Clickstream data?
  - To help optimize navigation
  - Put the items / products that people look at most often, closer to the most common site entry point
  - See which pages people are spending the most time on
  - In some cases, a cookie can be left on the client machine, and the Clickstream data for the visitor stored in the database as a profile
  - When the user returns, if the cookie on their machine still exists, their profile is extracted and products relevant to their previous browsing history placed first in any search results
- Many large commercial sites make heavy user of Clickstream data for client profiling

Clickstreaming

- In online education, Clickstream can be used to profile student usage of course materials
  - What items have been looked at / downloaded
  - For how long
  - Have bulletin board entries been made
  - Quiz scores / time spent on quizzes
  - Log-ins and Logouts
- Can be useful for staff to see which materials are being utilised and which are not
- To see how students are performing in ongoing quizzes based on the learning materials

Clickstream Analysis

- Collecting Clickstream data is only the first part of the process
- To understand the implications of what the data is saying, advanced statistical analysis is required
- Often, one is looking for a complex result from relatively simple data
- A number of tools are available online that can be integrated with web applications to automatically track the Clicksteam of users
- These tools can then pump out charts, graphs and numerical calculations according to the requirements of the site administrators

Auditing

- Whilst Clickstream data can indicate how a user moves through a site and what they look at, auditing data records what they do
- In the context of this discussion, auditing relates to registered users within a web application
- Things that can be recorded as part of an audit log include;
  - Successful logins
  - Failed logins
  - User details in either of the above cases
  - Time
  - IP address
  - Events (add / edit / delete)
  - Logouts

Logins

- When a user logs-in is critical to knowing if a user was in the system when an event occurs
- Where that person logged in from (IP) address can be useful for backtracking in cases of stolen / fraudulent account details (hence why freemail systems are not truly anonymous)
- How often a person logs-in and at what times can be useful (does it vary or is it consistent – does it vary from the norm)

Failed Logins

- Instances of failed log-ins need to be recorded in order to detect unauthorized attempts at system access
- Again, time and IP address are required
- The username and password entered on each attempt need to be recorded
- This raises two other issues;
  - Do you implement a try limit – that is, three failed attempts using the same username, or from the same IP address and then that IP address / username is locked out for a set period of time
  - Do you allow concurrent (multiple) connections with the same account details. Theoretically, if a user is logged in, they should not be able to log-in again, especially from another address (as this causes issues in repudiation)

Event Logging

- As we have seen over the course of the weekly lectures, most of the things we do in web applications development involves interacting with data in databases
- Event logging (aside from those already discussed) is critical when data is being entered into a database, deleted from a database or update within the database
- Typically, the system should record who added, deleted, edited the data, what the event was, what the data was, and at what time it happened

Event Logging

- The challenge with data logging is to decide how much or how little information to record
- In the previous example we have enough information to see who added a record, that it was a record addition, who the new user was (to the point where we could locate the rest of their details at least) and what time the record was added
- We can sort and sort for events according to any of these data points
- Remember, a web application could have tens, hundreds or thousands of users per day – such activity could lead to unmanageable log table sizes and server activity
- The greater the fidelity of the logging environment, the greater the load on the system, and thus a reduced application performance

Event Logging

- In the end, the point of logging is to provide administrators with enough information to discover who did what to which data and when
- Sessions are used to Clickstream anonymous users within a system, whilst database logs are used to record all user-centered events within a system
- No piece of data should be add, deleted or edited without the admins being able to identify that user
- If people are trying to log into a system and keep failing, there should be a record of it

Logging and Auditing

- Auditing is not just about finger pointing when things go wrong
- It can be used to check that data is reliable and up to date
- That workers are actually working (a contentious issue in modern workplaces)
- That incorrectly added/deleted/edited data can be identified and corrected
- That possible security issues can be identified and dealt with
- To indicate when a system is at maximum and minimum usage load, so that planning for maintenance and backup can be planned for with minimal disruption

Auditing Issues

- Who does the auditing?
  - Can users see their own logs?
  - Can only the admins see all logs
  - Do they need a specific reason to review user event logs?
  - Is privacy an issue?
- Auditing Policy
  - Should staff know what is being audited?
  - How long are audit records kept?
  - Do audit records provide absolute proof of the events that took place?
- As potential developers, these are just some of the issues you will need to be aware of when developing for yourself or clients
- Not all these questions have easy answers
- An application with no auditing is like a bank without locks or cameras
- However, whilst auditing can be critical to problem resolution, it creates problems of access, privileges, system capacity and system performance

Conclusion

- Session management is a crucial skill to master in any web dev environment
- For multi-page and multi-level user applications, they are essential
- Sessions are also crucial to security, user management and site usage analysis