Session Tracking

- The Need
  - shopping cart, personalization, ...

- The Difficulty
  - HTTP is a "stateless" protocol
  - Even persistent connections only last seconds

- The Trick??

General Idea

![Diagram showing the flow of requests and responses between client and server]

Three Ways to Implement Session Tracking

- URL Re-writing
- Hidden form field
- Cookies

Example: Shopping Cart (I)

- Using URL re-writing
- Using hidden form field

Cookies

- Issued by the server
  - HTTP Response: Set-Cookie
- Part of the next client request
  - HTTP Request: Cookie
Cookie Attributes
- Name, Value
- Host/Domain, Path
- Require secure connection
- Max age
- Comment (Version 1)

Servlet Cookie API
- Cookie
  - getThis(), setThat() ...
  - setMaxAge( int )
    - 1000??, -1??, 0??
- HttpServletResponse
  - addCookie( Cookie )
- HttpServletRequest
  - Cookie[] get_cookies()

Example: Shopping Cart (II)
- Using Cookie

Cookie or No Cookie?
- Is cookie a potential security problem?
  - Virus??
  - DoS??
- How about privacy?
  - Cookie manager in Mozilla/Firefox(?)
  - Internet Options in IE

It’s Not Easy ...
- ... to generate unique and random session id’s
- ... to associate more information with a session
- ... to tell whether the client has already left

Servlet Session Tracking API
- HttpServletRequest
  - HttpSession getSession()
- HttpSession
  - setAttribute( String, Object )
  - getAttribute( String )
  - setMaxInactiveInterval( int )
    - Tomcat default: 30 seconds
  - invalidate()
Example: Shopping Cart (III)

- Using Session Tracking API

Is Session Shared Across Multiple Servlets?

- **LoginServlet**
  - Validate username and password
  - Failed: redirect to error page
  - Succeeded: set a session attribute "authenticated" to "true", and redirect to DisplayServlet

- **DisplayServlet**
  - Check session attribute "authenticated"
  - null: redirect to LoginServlet
  - "true": display content

Example:

- [LoginServlet]
- [DisplayServlet]