Java Servlets 3.0

Lesson 10: Session Tracking

Capgemini

Lesson Objectives

In this lesson, we will learn:

- Introduction and Need for Session Management
- Different Techniques of Session Tracking
- Best Practices

What is a Session?

A session is the duration from which a client connects to a server till the client disconnects from that server where the user might access/view multiple pages.

- A session is specific to an application as well as a user
- Session begins with either of the following:
  - The first connection to an application by a client
  - The client logs-in for authenticated sessions
- Session ends after either of the following:
  - That client's last connection
  - That client logs-out
  - A time-out period of inactivity

What is Session Tracking?

Session tracking implies maintaining client specific information on the server across multiple requests during the session

For example: Any Online Shopping application saves the state of a user’s shopping cart across the requests

Why Session Tracking?

Session tracking is desirable due to the following reasons:

- HTTP is stateless protocol.
- In HTTP communication, client makes a connection to the server, sends the request, gets the response and closes the connection.

Session Tracking Techniques

There are several techniques of session tracking in JEE :

- Hidden Form Fields
- URL Rewriting
- Cookies
- Session Tracking API

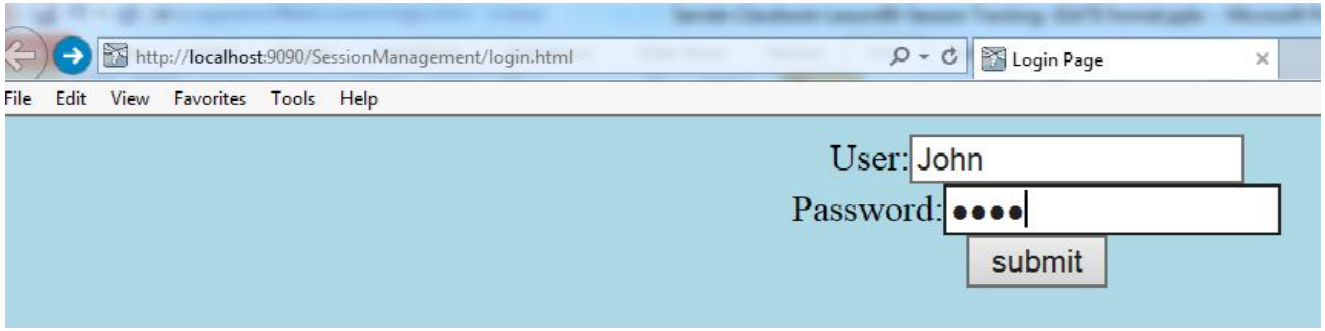
Hidden Form Fields

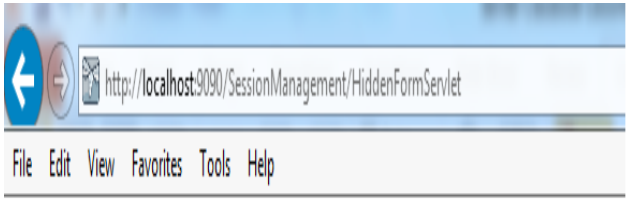
Hidden Form fields are fields added to an HTML form that are not displayed in the client's browser.

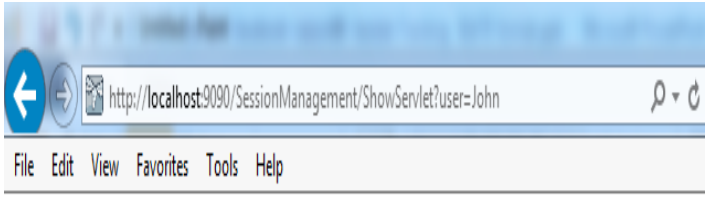
- They are sent back to the server when the form that contains them is submitted
- Hidden fields are supported in all the popular browser, they demand no special server requirements, and they can be used with clients that haven't registered or logged.
- The major disadvantage of this technique is that it works only for a sequence of dynamically generated forms.

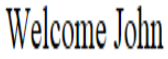
Demo

Execute the following login.html, HiddenFormServlet, ShowServlet









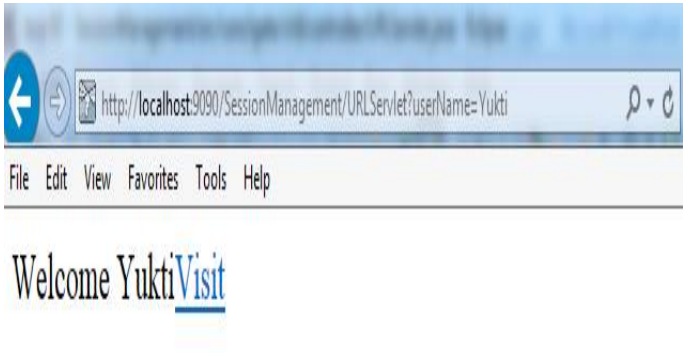
URL Rewriting

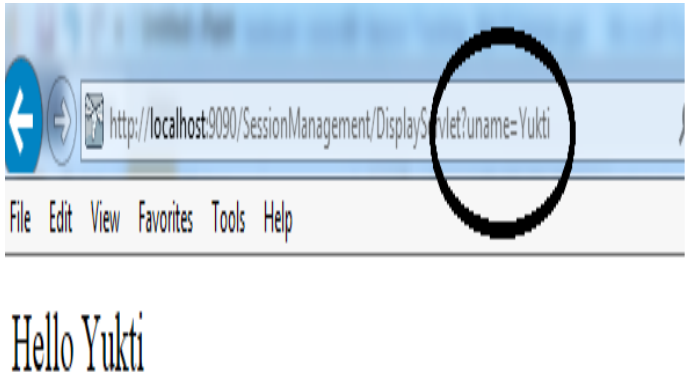
URL rewriting is another way to support anonymous session tracking.

- With URL rewriting, every local URL the user might click on is dynamically modified, or rewritten, to include extra information.
- The extra information can be in the form of extra path information, added parameters, or some custom, server-specific URL.
- Due to the limited space available in rewriting a URL, the extra information is limited to a unique session ID

Demo

Execute the following input.html, URLServlet, DisplayServlet





Cookies

Cookie is a small text file containing client information sent by a web server to a browser that can later be read back from the browser.

- When a browser receives a cookie, it saves the cookie and thereafter sends the cookie back to the server each time it accesses a page on that server, subject to certain rules.
- Since cookie’s value can uniquely identify a client, cookies are often used for session tracking.

Working with Cookies

Cookie can be created using Cookie class which is in the package javax.servlet.http.Cookie.

- To create cookie use Cookie class constructor:
  - public Cookie(String name, String Value)
- Once the cookie is created, send the cookie to the browser using the following method:
  - HttpServletResponse.addCookie(Cookie cookie)
- Cookies can be retrieved by servlet from a request by using the following method:
  - HttpServletRequest.getCookies()

Cookie Methods

Let us discuss some prominent cookie methods:

- public void setComment(java.lang.String purpose)
- public java.lang.String getComment()
- public void setDomain(java.lang.String pattern)
- public java.lang.String getDomain()
- public void setMaxAge(int expiry)
- public int getMaxAge()
- public void setPath(java.lang.String uri)
- public java.lang.String getPath()
- public void setSecure(boolean flag)

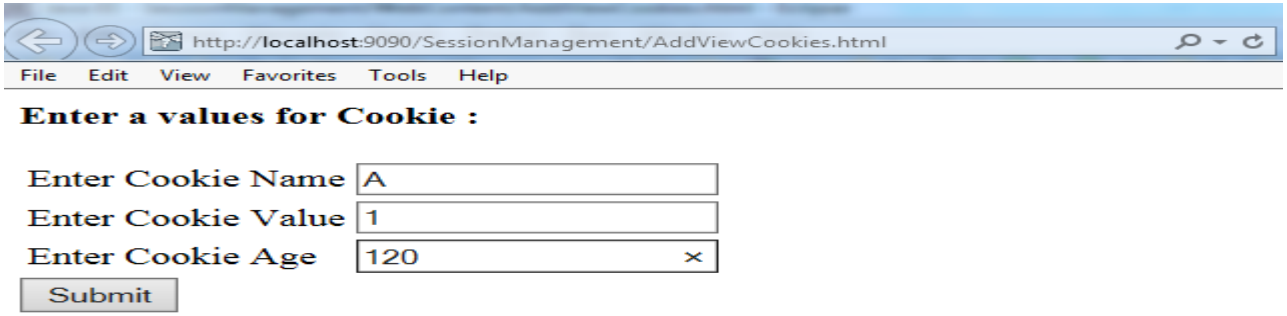
Cookie Methods

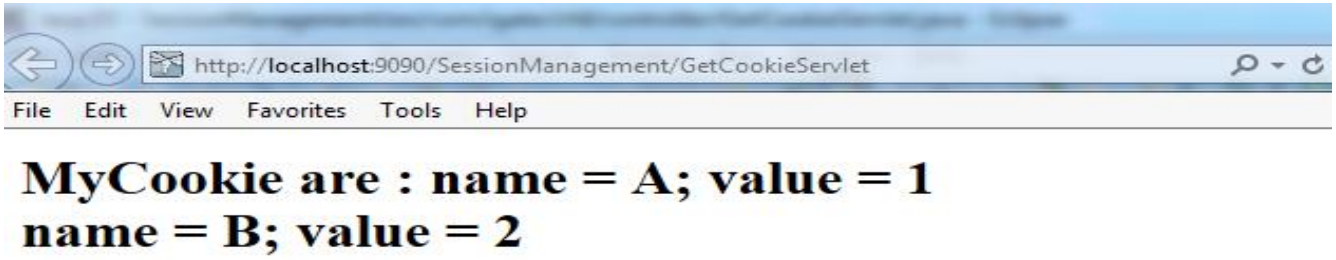
- public java.lang.String getName()
- public void setValue(java.lang.String newValue)
- public java.lang.String getValue()
- public int getVersion()
- public void setVersion(int v)

Demo

Creating and retrieving cookies:

- AddCookieServlet Code
- GetCookieServlet Code
- AddViewCookies html page



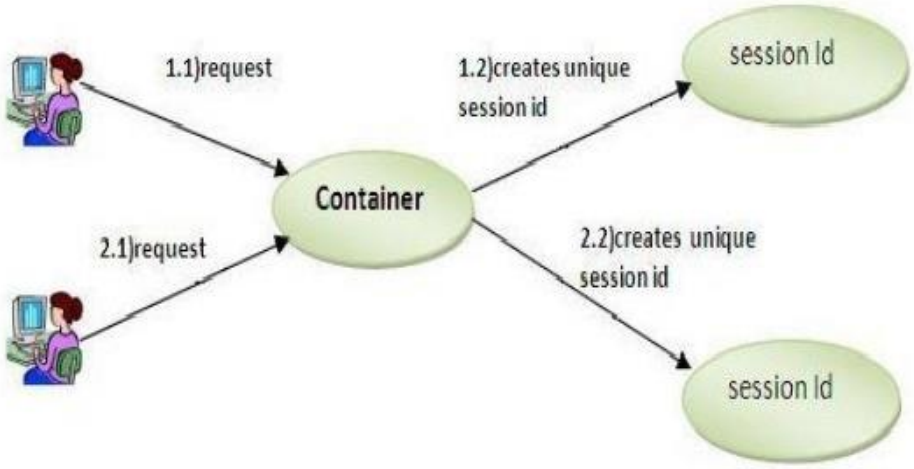


Session Tracking API

Servlet API provides HttpSession interface in javax.servlet.http package to track and manage sessions

HttpSession interface provides a way to identify a user across more than one page request or visit to a Web site and to store information about that user

A session usually corresponds to one user, who may visit a site many times



Session Tracking API - Using HttpSession

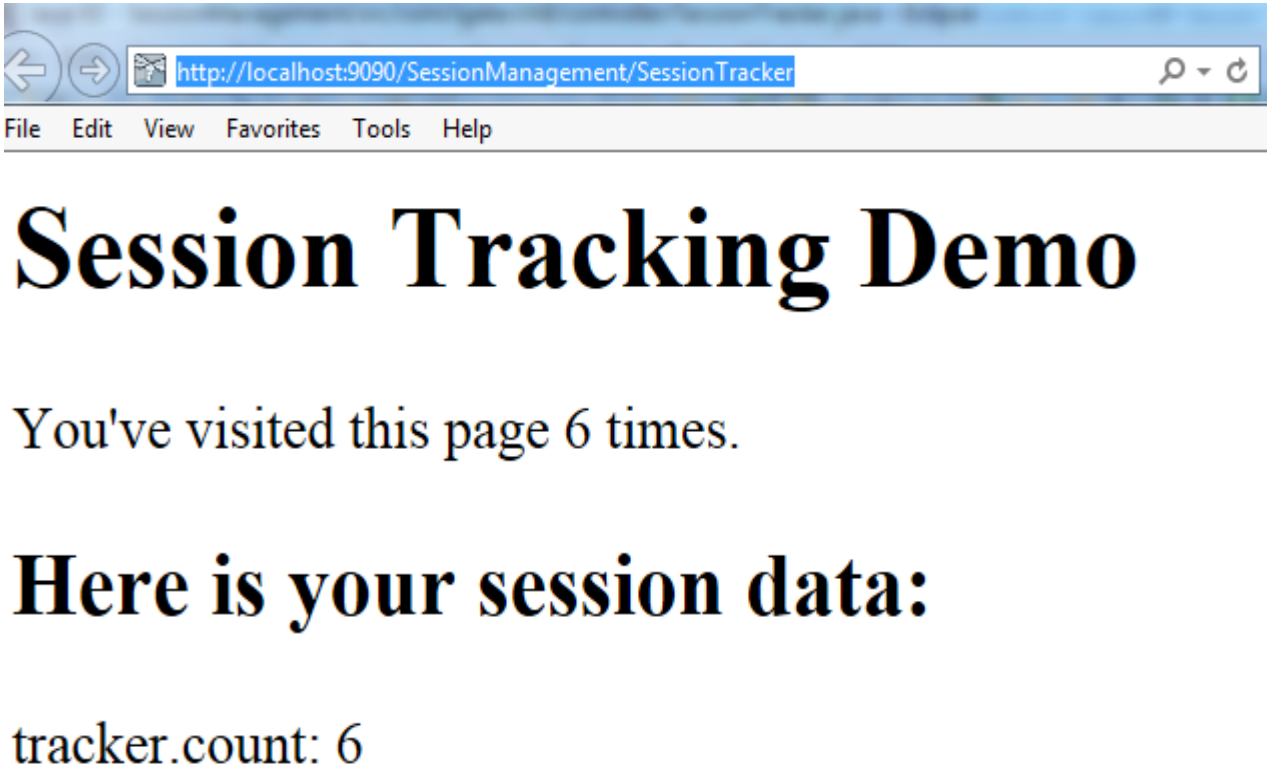
HttpSession interface provides methods for session tracking. They are:

- public HttpSession HttpServetRequest.getSession(boolean create)
- public void HttpSession.setAttribute(String name, Object value)
- java.lang.Object HttpSession.getAttribute(String name)
- public void HttpSession.removeAttribute(String name)
- public String HttpSession.getId()

Demo

Session tracking to count the number of times a client has accessed the site:

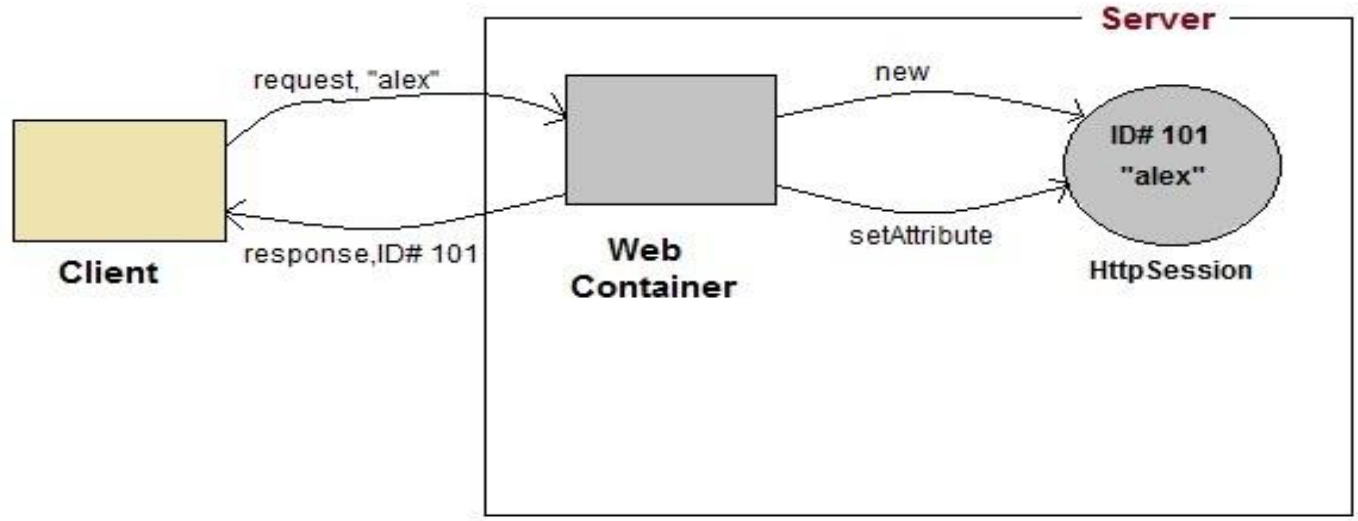
- SessionTracker.java servlet



Session Tracking API – Using Session ID

When a user first accesses the site, is assigned a new HttpSession object and a unique session ID.

- Session ID: It identifies the user and is used to associate the user with HttpSession object in subsequent requests.
- Session id is appended to the URL for session tracking using URL rewriting.



Session Tracking API – Using Session ID

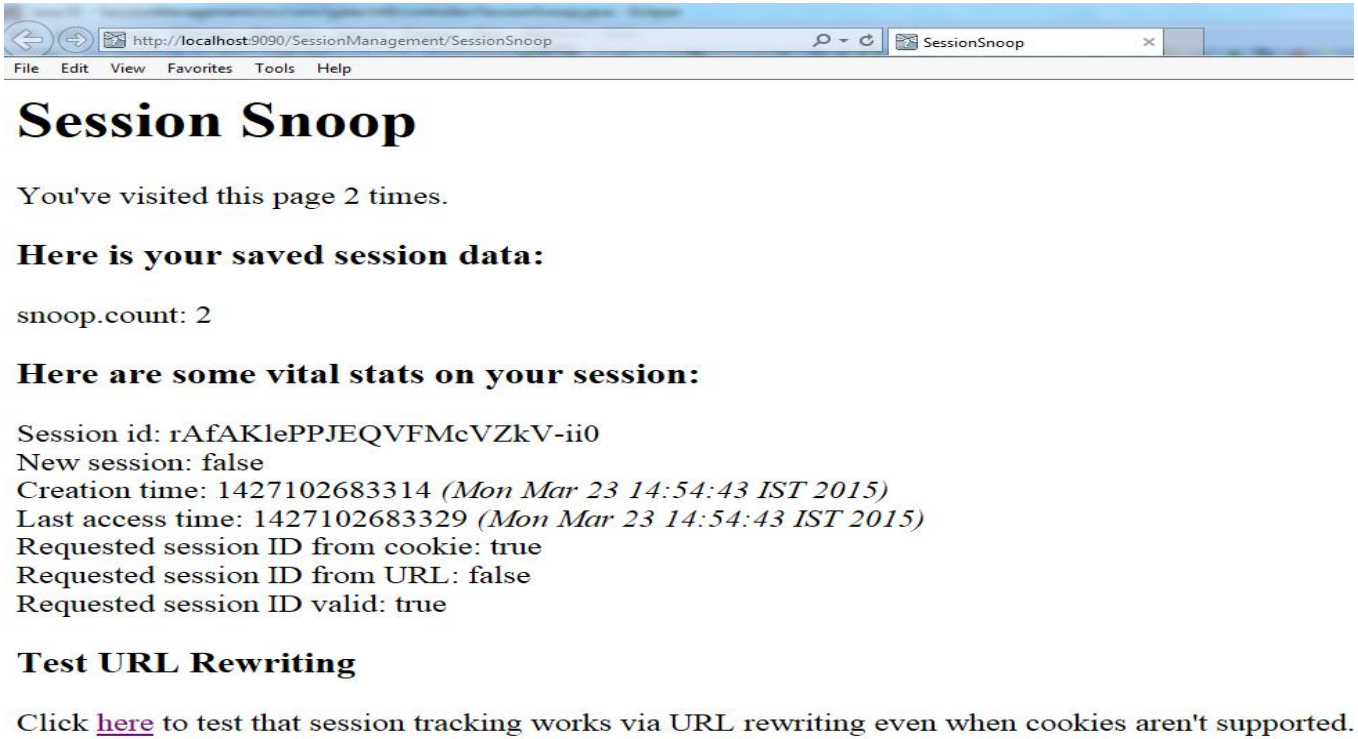
Two methods to encode URL:

- public java.lang.String encodeURL(java.lang.String url)
- public java.lang.String encodeRedirectURL(java.lang.String url)

Demo

Execute the servlet which gets Session Information using HttpSession & URL Rewriting

- Sessionsnoop servlet



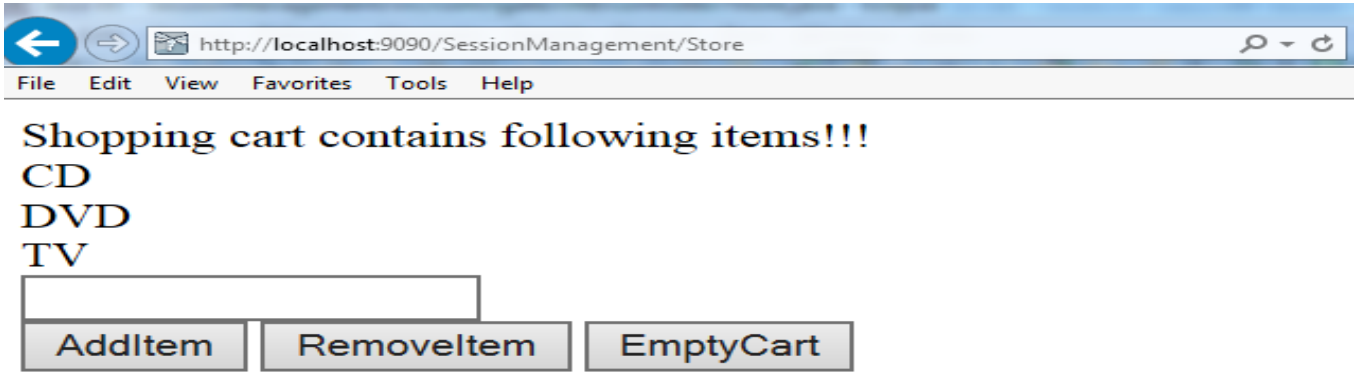
Demo

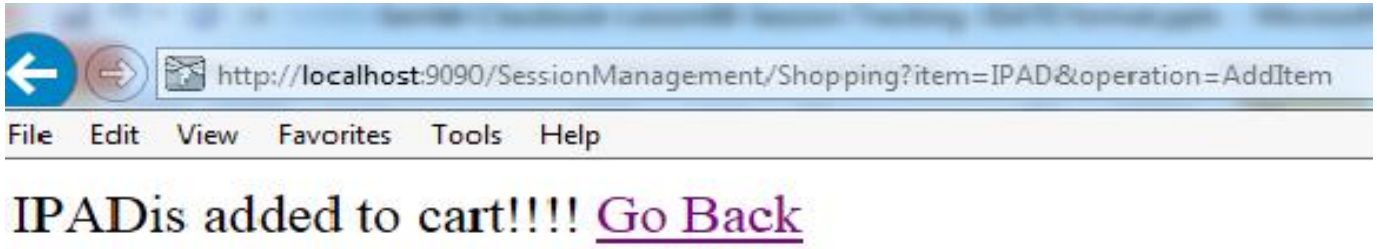
Execute a servlet that manually invalidates a session if it is more than a day old or has been inactive for more than an hour.

- Manual Invalidate servlet

Demo

This is a simple shopping cart example: Store.java and Shopping.java





Best Practices in Session Tracking

An application that uses URL rewriting to track sessions must adhere to certain programming guidelines as URL rewriting has high security risks.

The application developer needs to:

- Program session servlets to encode URLs
- Supply a servlet or JSP file as an entry point to the application
- Avoid using plain HTML files in the application
- All emitted links must be consistently rewritten

Lab: Session Management

- Lab 6.1

Summary

In this lesson, we have learnt:

- The concept of Session Tracking
- Session Tracking Techniques

Review Questions

Question 1: Web Application developer needs to maintain client's state on server because:

- Option 1: To remind client of his information
- Option 2: Http is a stateless protocol
- Option 3: Http is unreliable protocol
- Option 4: No need as its taken care of by the Http Server

Question 2: Browsers required to accept ____ cookies per site:

- Option 1: 1
- Option 2: 40
- Option 3: 20
- Option 4: unlimited

Review Questions

Question 3: ____ method is used to end the session

- Option 1: invalidate()
- Option 2: logout()
- Option 3: invalidSession()

Question 4: Global timeout for all the sessions is maintained by.

- Option 1: HttpSession.setMaxInactiveInterval(600)
- Option 2: iHttpSession.logout(600)
- Option 3: mapping in web.xml
- Option 4: All Options are true