Session Management

HttpRequest Methods

Method	Description
Cookie[] getCookies()	Returns an array containing all of the Cookie objects the client sent with this request.
Enumeration getParameterNames()	Returns an Enumeration of String objects containing the names of the parameters contained in this request.
HttpSession getSession()	Returns the current session associated with this request, or if the request does not have a session, creates one.
HttpSession getSession(boolean create)	Returns the current HttpSession associated with this request or, if if there is no current session and create is true, returns a new session.
String getContentType()	Returns the MIME type of the body of the request, or null if the type is not known.
String getContextPath()	Returns the portion of the request URI that indicates the context of the request.
String getParameter(String name)	Returns the value of a request parameter as a String, or null if the parameter doesnot exist.
String getPathInfo()	Returns any extra path information associated with the URL the client sent when it made this request.
String[] getParameterValues(String name)	Returns an array of String objects containing all of the values the given request parameter has, or null if the parameter does not exist.
int getContentLength()	Returns the length, in bytes, of the request body and made available by the input stream, or -1 if the length is not known.

HTTP Status Codes

Code:	Message:	Description:
200	OK	The request is OK
400	Bad Request	The server did not understand the request
401	Unauthorized	The requested page needs a username and a password
403	Forbidden	Access is forbidden to the requested page
404	Not Found	The server can not find the requested page.
408	Request Timeout	The request took longer than the server was prepared to wait.
500	Internal Server Error	The request was not completed. The server met an unexpected condition

HTTPRequest Methods

Method	Description
void sendError(int sc)	Sends an error response to the client using the specified status code and clearing the buffer.
void sendRedirect(String location)	Sends a temporary redirect response to the client using the specified redirect location URL.
void setBufferSize(int size)	Sets the preferred buffer size for the body of the response.
void setContentType(String type)	Sets the content type of the response being sent to the client, if the response has not been committed yet.
void flushBuffer()	Forces any content in the buffer to be written to the client.
public void setStatus (int statusCode)	This method sets an arbitrary status code. The setStatus method takes an int (the status code) as an argument. If your response includes a special status code and a document, be sure to call setStatus before actually

Session Tracking in Servlets

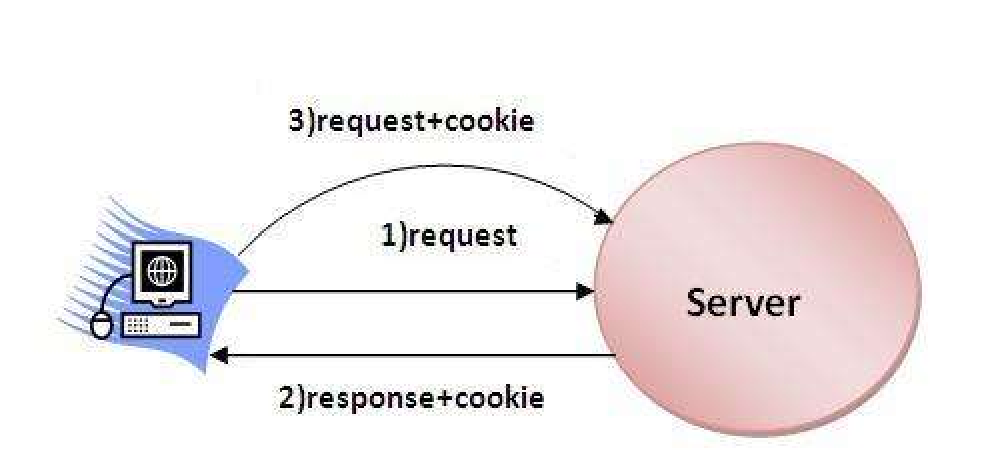
- Session simply means a particular interval of time.
- Session Tracking is a way to maintain state (data) of an user. It is also known as session management in servlet.
- HTTP is stateless that means each request is considered as the new request.
- we need to maintain the state of an user to recognize to particular user.

Session Tracking Techniques

- Cookies
- Hidden Form Field
- URL Rewriting
- HttpSession

Cookies in Servlet

- A cookie is a small piece of information that is persisted between the multiple client requests.
- A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.



Types of Cookie

- Non-persistent cookie
  - It is valid for single session only. It is removed each time when user closes the browser.
- Persistent cookie
  - It is valid for multiple session . It is not removed each time when user closes the browser. It is removed only if user logout or signout.

Pros & Cons

- Advantage of Cookies
  - Simplest technique of maintaining the state.
  - Cookies are maintained at client side.
- Disadvantage of Cookies
  - It will not work if cookie is disabled from the browser.
  - Only textual information can be set in Cookie object.
- Note: Gmail uses cookie technique for login. If you disable the cookie, gmail won't work.

Cookie class

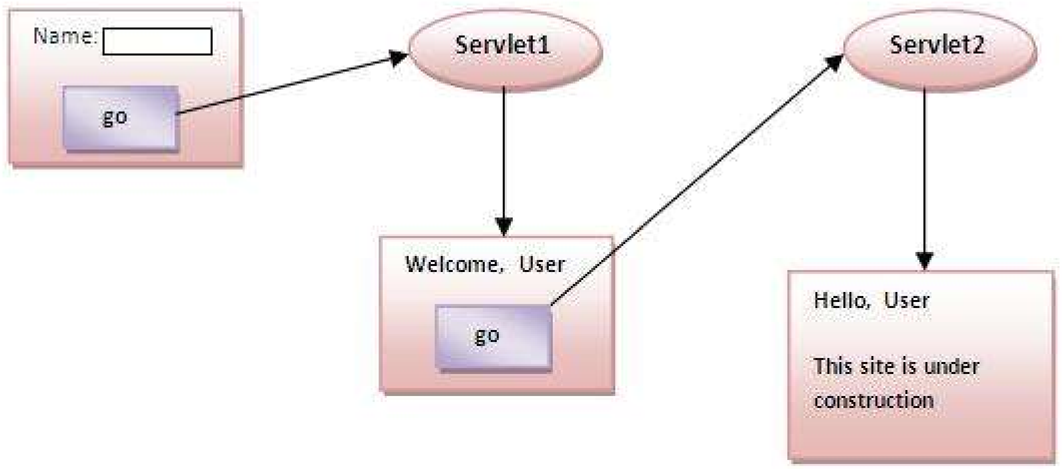
- javax.servlet.http.Cookie class provides the functionality of using cookies. It provides a lot of useful methods for cookies.

Constructor	Description
Cookie()	constructs a cookie.
Cookie(String name, String value)	constructs a cookie with a specified name and value.

Methods to use Cookie

Method	Description
public void setMaxAge(int expiry)	Sets the maximum age of the cookie in seconds.
public String getName()	Returns the name of the cookie. The name cannot be changed after creation.
public String getValue()	Returns the value of the cookie.
public void setName(String name)	changes the name of the cookie.
public void setValue(String value)	changes the value of the cookie.
public void addCookie(Cookie ck)	method of HttpServletResponse interface is used to add cookie in response object.
public Cookie[] getCookies()	method of HttpServletRequest interface is used to return all the cookies from the browser.

Cookie example



Hidden Form Field

- In case of Hidden Form Field a hidden (invisible) textfield is used for maintaining the state of an user.
- We store the information in the hidden field and get it from another servlet.
- This approach is better if we have to submit form in all the pages and we don't want to depend on the browser.

Pros & Cons

- Advantage of Hidden Form Field
  - It will always work whether cookie is disabled or not.
- Disadvantage of Hidden Form Field:
  - It is maintained at server side.
  - Extra form submission is required on each pages.
  - Only textual information can be used.

URL Rewriting

- In URL rewriting, we append a token or identifier to the URL of the next Servlet or the next resource. We can send parameter name/value pairs using the following format:
- url?name1=value1&name2=value2&??

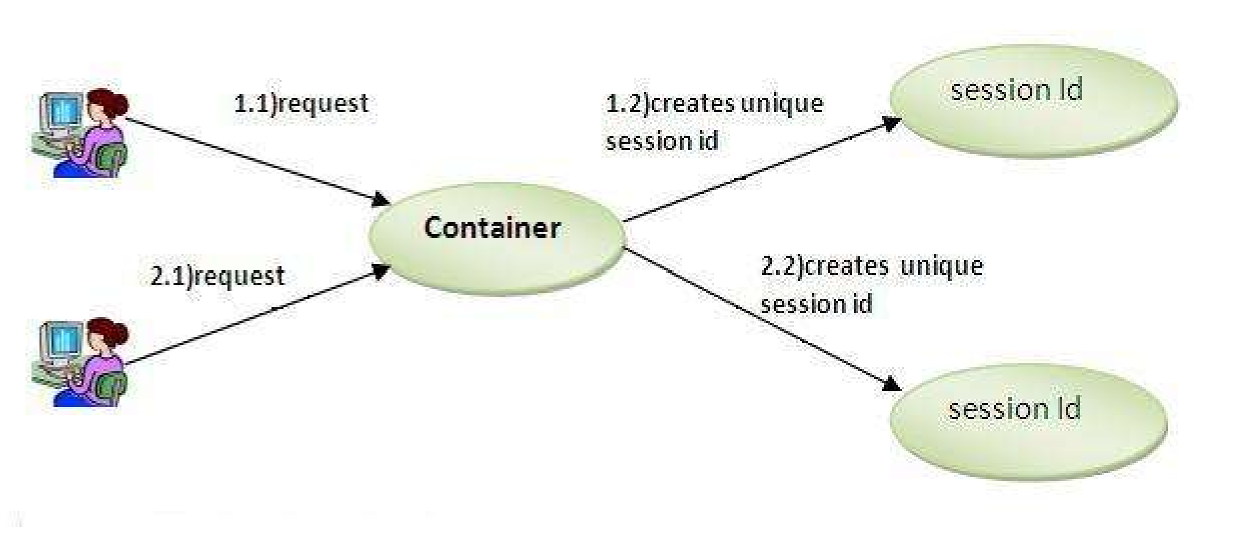
Pros & Cons

- Advantage of URL Rewriting
  - It will always work whether cookie is disabled or not (browser independent).
  - Extra form submission is not required on each pages.
- Disadvantage of URL Rewriting
  - It will work only with links.
  - It can send Only textual information.

HttpSession interface

- Container creates a session id for each user.
- The container uses this id to identify the particular user.
- An object of HttpSession can be used to perform two tasks:
  - bind objects
  - view and manipulate information about a session, such as the session identifier, creation time, and last accessed time.

HttpSession



HttpSession

- The HttpServletRequest interface provides two methods to get the object of HttpSession:
    - public HttpSession getSession():Returns the current session associated with this request, or if the request does not have a session, creates one.
    - public HttpSession getSession(boolean create):Returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session.

Other methos

- public String getId(): Returns a string containing the unique identifier value.
- public long getCreationTime(): Returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.
- public long getLastAccessedTime(): Returns the last time the client sent a request associated with this session, as the number of milliseconds since midnight January 1, 1970 GMT.
- public void invalidate(): Invalidates this session then unbinds any objects bound to it.