Response(HttpResponse)

HttpResponse

- The ***HttpServletResponse*** object generates a response to return to the requesting client. Its methods allow you to set the response header and the response body.
- Responseheader ***(response.setContentType("text/html");)*** identifies the MIME type of the response.
- ***getWriter()*** method to return a ***PrintWriter*** object. The print() and println() methods of the ***PrintWriter*** object write the servlet response back to the client.

Most Common Methods

- **Writing HTML**
  - PrintWriter writer = response.getWriter();
  - writer.write("<html><body>GET/POST response</body></html>");
- **Headers**
  - response.setHeader("Header-Name", "Header Value");
- **Content-Type**
  - response.setHeader("Content-Type", "text/html");
- **Writing Text**
  - response.setHeader("Content-Type", "text/plain");
  - PrintWriter writer = response.getWriter();
  - writer.write("This is just plain text");

Most Common Methods

- **Content-Length**
  - response.setHeader("Content-Length", "31642");
- **Writing Binary Data**
  - OutputStream outputStream = response.getOutputStream(); outputStream.write(...);
- **Redirecting to a Different URL**
  - response.sendRedirect("https://www.google.com");
- **Writing HTML**

Cookies

Cookie

- A cookie is a small piece of information as a text file stored on client’s machine by a web application.

How cookie works?

- As HTTP is a stateless protocol so there is no way to identify that it is a new user or previous user for every new request.
- In case of cookie a text file with small piece of information is added to the response of first request.
- They are stored on client's machine. Now when a new request comes cookie is by default added with the request.
- With this information we can identify that it is a new user or a previous user.

Types of cookies

- Session cookies/Non-persistent cookies: These types of cookies are session dependent i.e. they are accessible as long as session is open and they are lost when session is closed by exiting from the web application.
- Permanent cookies/Persistent cookies: These types of cookies are session independent i.e. they are not lost when session is closed by exiting from the web application. They are lost when they expire.

Cookie Class

- Cookie class
  - Cookie class provides the methods and functionality for session management using cookies.
  - Cookie class is in javax.servlet.http
- Commonly used constructor of Cookie class:
  - Cookie(String name,String value): Creates a cookie with specified name and value pair.
  - Syntax: public Cookie(String name,String value)

Commonly used Method

- setMaxAge(int expiry):Sets the maximum age of the cookie.
  - Syntax: public void setMaxAge(int expiry)
- getMaxAge(): Returns the maximum age of the cookie. Default value is -1.
  - Syntax: public int getMaxAge()
- setValue(String newValue): Change the value of the cookie with new value.
  - Syntax: public void setValue(String newValue)
- getValue(): Returns the value of the cookie.
  - Syntax: public String getValue()
- getName(): Returns the name of the cookie.
  - Syntax: public String getName()

Create cookie

```
//create cookie object
Cookie cookie=new Cookie(“cookieName”,”cookieValue”);
//add cookie object in the response
response.addCookie(cookie);
```

How to get cookie?

- HttpServletRequest interface's getCookies() method is used to get the cookies from request object.
  - Syntax: public Cookie[] getCookies()
- Example:

```
//get all cookie objects.
Cookie[] cookies = request.getCookies();
//iterate cookies array to get individual cookie objects.
for(Cookie cookie : cookies){
out.println("Cookie Name: " + cookie.getName());
out.println("Cookie Value: " + cookie.getValue());
}
```

How to remove or delete cookies

- Cookies can be removed by setting its expiration time to 0 or -1. If expiration time set to 0 than cookie will be removed immediately. If expiration time set to -1 than cookie will be removed when browser closed.

How to remove or delete cookies

```
//Remove value from cookie
Cookie cookie = new Cookie(“cookieName”, “”);
//Set expiration time to 0.
cookie.setMaxAge(0);
//add cookie object in the response.
response.addCookie(cookie);
```

Hidden field

Hidden field

- Hidden field is an input text with hidden type. This field will not be visible to the user.
- Syntax:<input name=”fieldName” value=”fieldValue” type=”hidden”/>

Get hidden field value in servlet

- HttpServletRequest interface's getParameter() method is used to get hidden field value in servlet.
- Syntax: String value = request.getParameter("fieldName");
- Note: Hidden field only works in case of form submission so they will not work in case of anchor tag as no form submission is there.

Advantages

- Advantages of hidden field:
  - All browsers support hidden fields.
  - Simple to use.
- Disadvantages of hidden fields:
  - Not secure.
  - Only work in case of form submission.

URL Rewriting

URL Rewriting

- URL rewriting is a way of appending data at the end of URL. Data is appended in name value pair form. Multiple parameters can be appended in one URL with name value pairs.
  - Syntax:URL?paramName1=paramValue1& paramName2=paramValue2
- HttpServletRequest interface's getParameter() method is used to get parameter value from url in servlet.
  - Syntax: String value = request.getParameter("fieldName");

Sessions

HttpSession

- HttpSession is an interface that provides a way to identify a user in multiple page requests. A unique session id is given to the user when first request comes. This id is stored in a request parameter or in a cookie.

Session Methods()

Method	Description
public HttpSession getSession()	Will cause one session to be created.
public HttpSession getSession(boolean)	true = will cause one to be created; false = will return null (no session)
public String getRequestedSessionId()	Gets the ID assigned by the server to the session
public Boolean isRequestedSessionIdValid()	Returns true if the request contains a valid session ID
public Boolean isRequestedSessionIdFromCookie()	Returns true if the session ID was sent as part of a cookie
public Boolean isRequestedSessionIdFromURL()	Returns true if the session ID was sent through URL rewriting

Default technique for session tracking is to use cookies.

Cookies are sent in the header part of an HTTP message, so they must be set in the response prior to writing any data to the response.

Session Tracking with URL Rewriting

Method	Description
public String encodeURL(String)	Encodes the specified URL by including the session ID in it, or, if encoding is not needed, returns the URL unchanged.
public String encodeRedirectURL(String)	Encodes the specified URL by including the session ID in it, or, if encoding is not needed, returns the URL unchanged.

Session is Useful for persisting information about a client and a client's interactions with an application.

Interfaces

Method (lifecycle types)	Description
public long getCreationTime()	Returns the time when this session was created.
public String getId()	Returns a string containing the unique identifier assigned to this session.
public long getLastAccessedTime()	Returns the last time the client sent a request associated with this session.
public boolean isNew()	Returns true if the client does not yet know about the session or if the client chooses not to join the session.
public void setMaxInactiveInterval(int interval)	Specifies the time, in seconds, between client requests before the servlet container will invalidate this session.
public int getMaxInactiveInterval()	Returns the maximum time interval, in seconds, that the servlet container will keep this session open between client accesses.
public void invalidate()	Invalidates this session then unbinds any objects bound to it.

Get session object

- HttpServletRequest interface's getSession() method is used to get the session object.

- Syntax: HttpSession session = request.getSession();

Set attribute in session object

- HttpSession interface’s setAttribute() method is used to set attribute in session object.

- Syntax: public void setAttribute(String name,Object value);

- Example: session.setAttribute(“attName”, “attValue”);

Get attribute from session object

- HttpSession interface's getAttribute() method is used to get attribute from session object.

- Syntax: public Object getAttribute(String name);

- Example: String value = (String) session.getAttribute("attName");