Saturday, June 25, 2022 9:45 AM

Servlet Lifecycle
One servlet object is created!!
When is it created ?

1. Immediately after deployment the container creates the servlet object **}} EAGER INITIALIZATION**
   OR
2. Whenever first request arrives for that servlet the container creates the servlet object **}} LAZY INITIALIZATION**

Who creates the servlet object ? Web container
By default the container uses **LAZY INIT**

Scope in web application data !!! ---- For HOW MANY Requests the same variable/object is available

It could be that data is available for a single request = REQUEST scope data
It could be that data is available for few requests = SESSION scope data
It could be that data is available for ALL requests = APPLICATION scope data

Let us add more servlets to the request scope!!!!!

Servlet Chaining --
Browser -----------------REQ1-----|----->Serv1------>Serv2------>Serv3
REQ1 = single request is sent form serv1 to serv2 to serv3
The response is sent after serv3

Who creates servlets ? Servlet Container ----represented by a class called as **ServletContext**
We ask the ServletContext to give us a RequestDispatcher
Once we get a dispatcher we dispatch the request to the next servlet in two ways
forward dispatch --- the response output of intermediate servlets are washed off/ignored
ONLY the response of the LAST servlet goes to the browser

include dispatch ------ the responses of all the servlets are concatenated and sent to browser----- all responses are included in the final output

Forward	browser ---->S1---->S2----->S3 ----->response ---->browser
Include	**browser**------>S1 ----->S2----->S3 <-------- <------- <-------

Browser ---->VerifyLogin
If succeeded ----------forward to HomeServlet
If failed -----------forward to login servlet

Dispatcher is used when the UI is ready and we want to redirect to it from current servlet with the same request and you may add attributes to it !!!

Session = it is a set of requests and responses between LOGIN and LOGOUT

Http Session ----

PROBLEM --- http is a stateless-protocol
Http server never SAVES data after sending response
Request-----data-----Response ----------> data is discarded

To overcome this problem --- we want the server to remember things till we logout !!!
the technique used is called as token passing ---
On first request ----server generates a token , token is mapped to the shelf of the RACK where client's data will be kept .
Server gives the token to the client
Next time client wants to keep more things then client brings the token shows to server , server finds the shelf in the RACK and stores client data again and returns the token
REPEAT
Finally when the client wants to leave the token is returned for ever and the shelf is cleared

---

Token passing is done in two ways -------

1. URL Rewriting

```
<form  action="verify"   >
<input type="submit"  value ="OK"  />
</form>
```

Action =**verify** means http://localhost:8080/App1/verify is used for the request

```
<form  action="verify? sessionId=12345"   >
  <input type="submit"  value ="OK"  />
   </form>
```

```
< a   href="show"   > </a>  }} Links
```

URL Rewriting

```
<a   href="show" ?sessionId=12345 > </a>
```

---

2. **Cookies** = small 1kb files having Key=value (sessionId=12345)
These cookies are attached to the response header when server sends data to client
Then cookies are also attached to the request header when the client sends request to server

```
Cookie c =  new Cookie("fav-color", "blue");
  response.setCookie(c);   //this will go and sit on client machine
```

Next time server gets a request the request COMES with the cookies on the client
Server knows about client preferences

Repeat this on different browser -------------

first request	http://localhost:8080/App1/cookie?fav-col=blue
second request on click of OK goes to MyServlet	getCookies get value of fav-col and use that in output
next any request to MyServlet from this browser gives o/p in fav-col	