Java & JEE Training

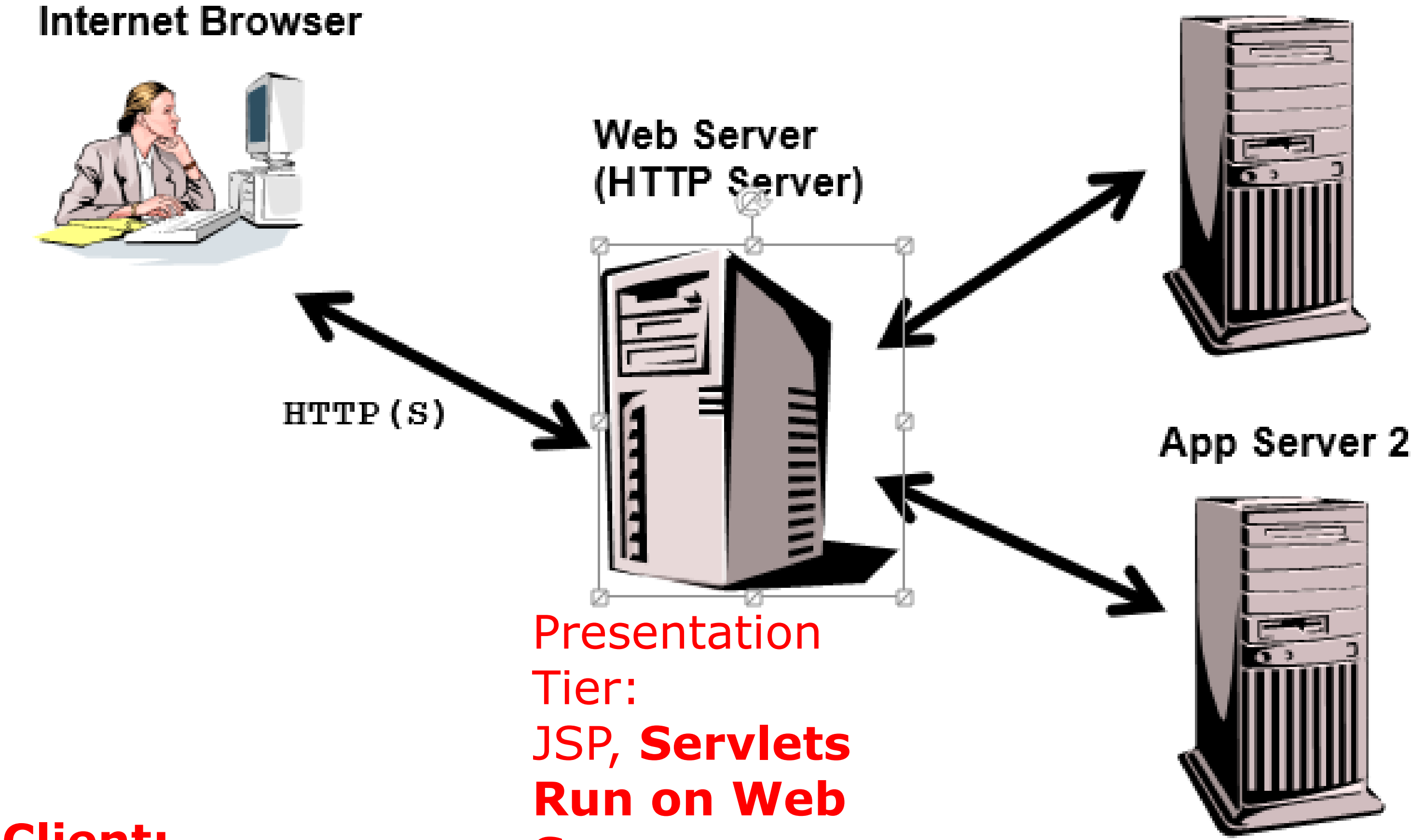
Day 32 – Session Management using Servlets Cookies and Hidden form fields

MindsMapped Consulting

Quick review of previous session…

- Web.xml
- Basic web application – end to end
- Parameters and Attributes

Web Server and Application Server



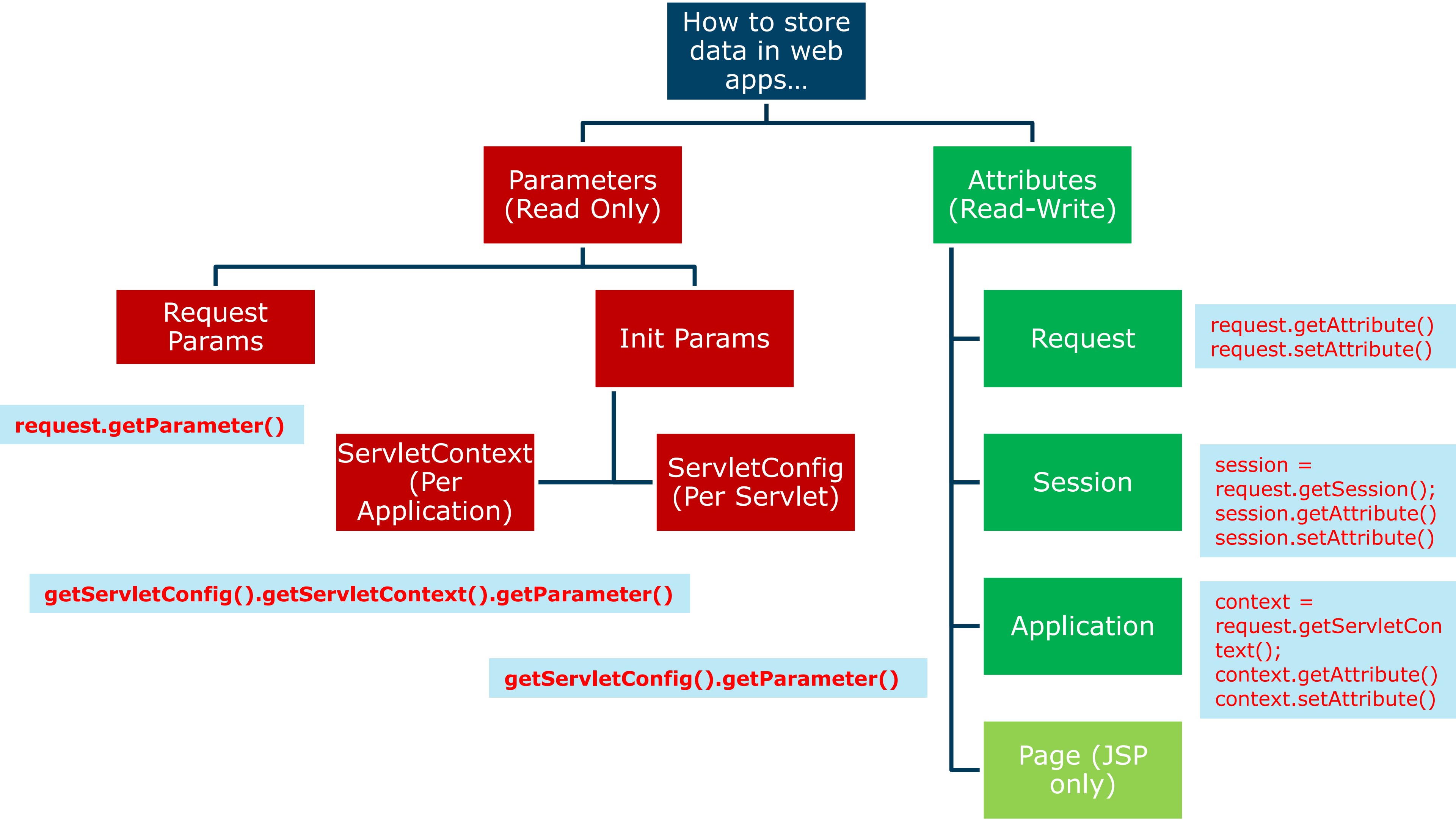
Client:
HTML JS CSS
(Run on client machine on the browser)

Presentation Tier:
JSP, **Servlets Run on Web Server or container**

JDBC on web server or app server

Database

Parameters and Attributes



How to store data in web apps…

Parameters (Read Only)

Request Params

request.getParameter()

Init Params

ServletContext (Per Application)

ServletConfig (Per Servlet)

getServletConfig().getServletContext().getParameter()

getServletConfig().getParameter()

Attributes (Read-Write)

Request

request.getAttribute()
request.setAttribute()

Session

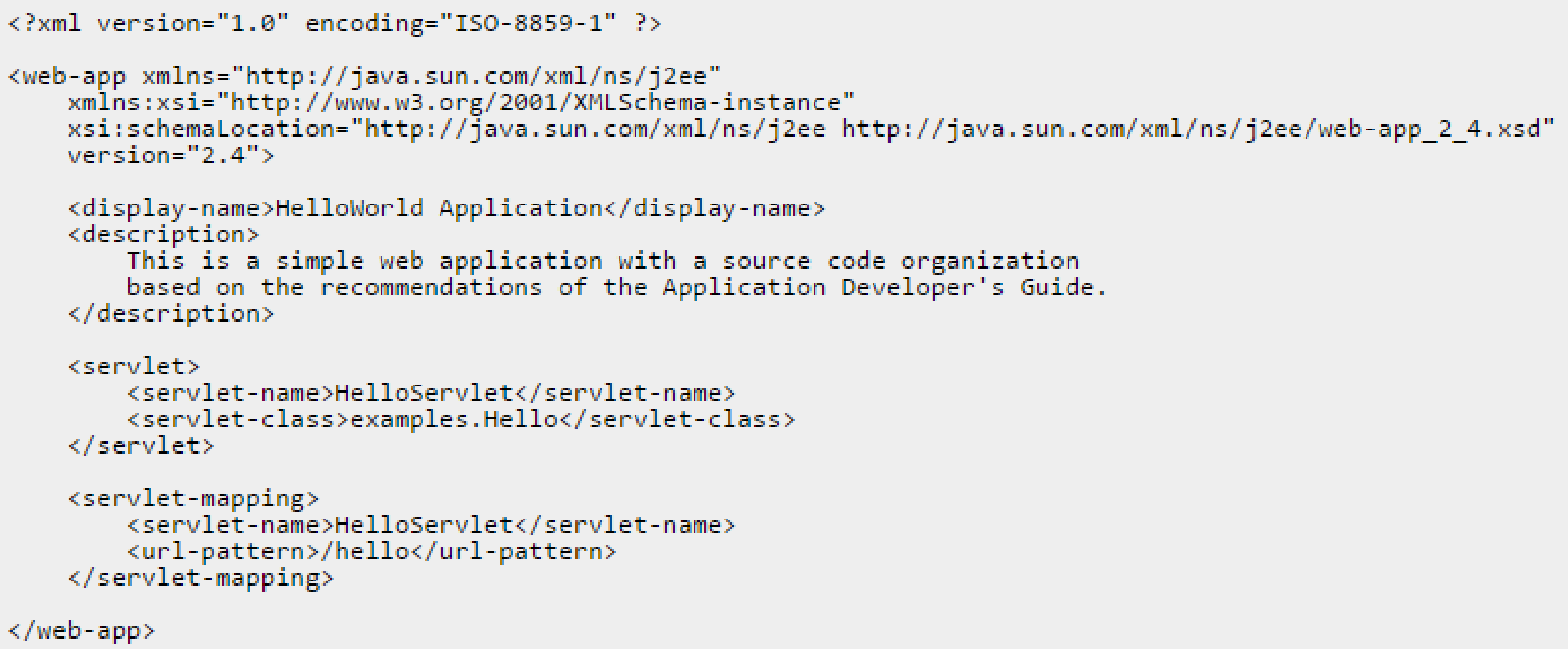
session = request.getSession();
session.getAttribute()
session.setAttribute()

Application

context = request.getServletContext();
context.getAttribute()
context.setAttribute()

Page (JSP only)

Web.xml file: Servlet mapping to URL



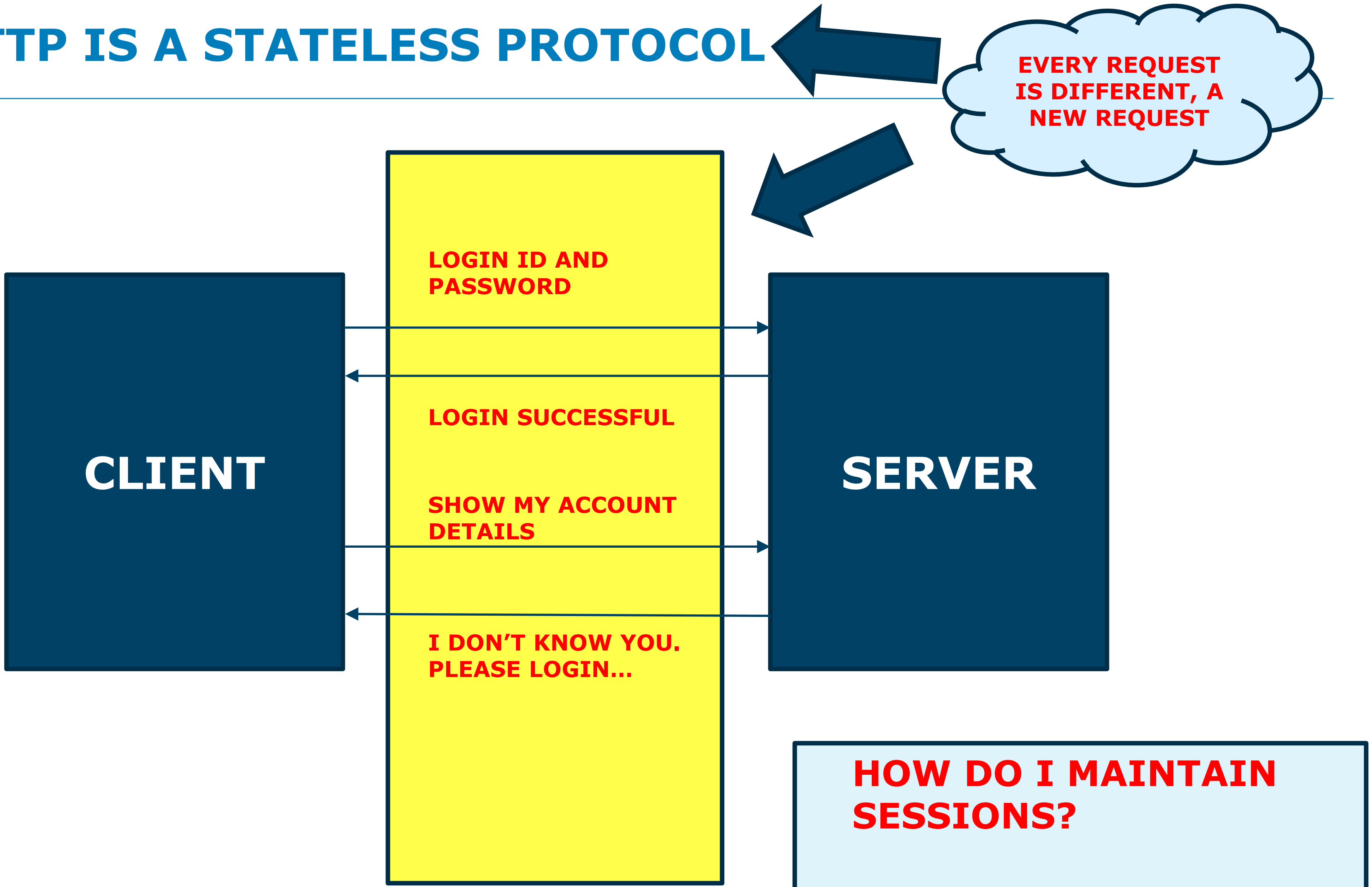
```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
    version="2.4">

    <display-name>HelloWorld Application</display-name>
    <description>
        This is a simple web application with a source code organization
        based on the recommendations of the Application Developer's Guide.
    </description>

    <servlet>
        <servlet-name>HelloServlet</servlet-name>
        <servlet-class>examples.Hello</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>

</web-app>
```

Java & JEE Training

Session tracking with servlets

MindsMapped Consulting

HTTP IS A STATELESS PROTOCOL



EVERY REQUEST IS DIFFERENT, A NEW REQUEST

CLIENT

SERVER

LOGIN ID AND PASSWORD

LOGIN SUCCESSFUL

SHOW MY ACCOUNT DETAILS

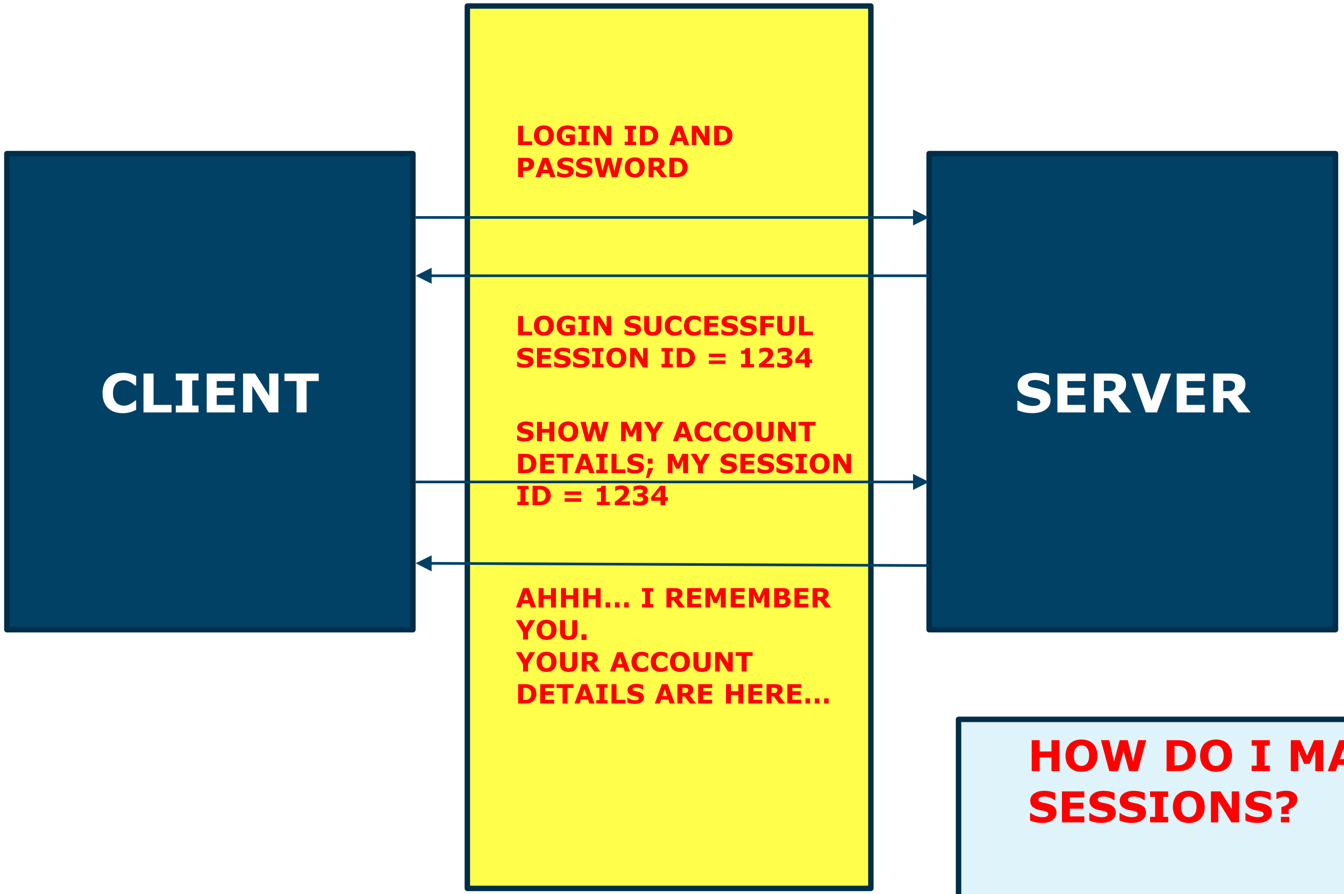
I DON'T KNOW YOU. PLEASE LOGIN…

SESSION – SEQUENCE OF RELATED REQUESTS COMING IN FROM THE CLIENT

HOW DO I MAINTAIN SESSIONS?

SERVER SHARES AN "ID" WITH THE CLIENT TO REMEMBER THE SESSION

HTTP IS A STATELESS PROTOCOL



CLIENT

SERVER

LOGIN ID AND PASSWORD

LOGIN SUCCESSFUL
SESSION ID = 1234

SHOW MY ACCOUNT DETAILS; MY SESSION ID = 1234

AHHH… I REMEMBER YOU.
YOUR ACCOUNT DETAILS ARE HERE…

SESSION – SEQUENCE OF RELATED REQUESTS COMING IN FROM THE CLIENT

HOW DO I MAINTAIN SESSIONS?

SERVER SHARES AN ID WITH THE CLIENT TO REMEMBER THE SESSION

Why track sessions?

- HTTP is stateless protocol
- So we need to maintain state across multiple requests of a session using session tracking techniques.

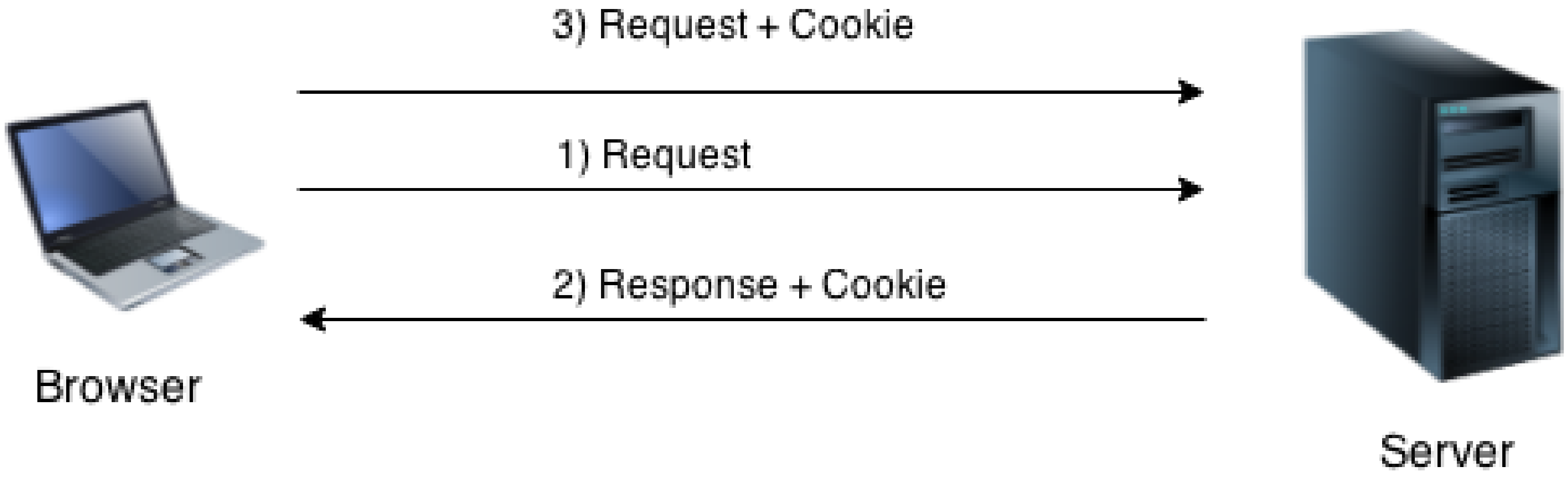
- Session tracking techniques:
  - **Cookies – key-value pair is stored on the browser… used for maintaining session information. Cookies are used (read and written) by the server**
  - **Hidden Form Field**

  - URL Rewriting
  - HttpSession

Demo of using cookies for session management…

- Student should be able to login by providing username and password.
- Once he logs in, he should be able to view more information about himself viz. course, college, score.
- Logout.

How cookies work?



Using cookies

- **javax.servlet.http.Cookie** class

- Creating a cookie:

```
Cookie ck=new Cookie("user",“pawan");//creating cookie object
response.addCookie(ck);//adding cookie in the response
```

- Deleting a cookie

```
Cookie ck=new Cookie("user","");//deleting value of cookie
ck.setMaxAge(0);//changing the maximum age to 0 seconds
response.addCookie(ck);//adding cookie in the response
```

- Get all cookies from request:

```
Cookie ck[]=request.getCookies();
for(int i=0;i<ck.length;i++){
 out.print("<br>"+ck[i].getName()+" "+ck[i].getValue());//printing cookies info
}
```

Example

- Demo of using cookies