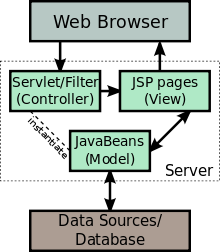
Q1: What is JSP? Discuss the lifecycle of JSP and scripting elements, basic servlet structure,servlet API features and servlet life cycle.

**JSP**

Java Server Pages (**JSP**) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, **JSP** is similar to PHP and ASP, but it uses the Java programming language.



**Life cycle of JSP**

A JSP life cycle can be defined as the entire process from its creation till the destruction which is similar to a **servlet life cycle** with an additional step which is required to compile a JSP into **servlet**.



**JSP scripting elements**

JSP scripting elements enable you insert Java code into the servlet that will be generated from the current JSP page. There are three forms:

* Expressions of the form <%= expression%> that are evaluated and inserted into output,

Script lets of the form <% code %> that are inserted into the servlets service method, and

* Declarations of the form <%! Code %> that are inserted into the body of the servlet class, outside of any existing methods.

**Basic servlet structure**

package testPackage; // Always use packages.

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.annotation.\*;

import javax.servlet.http.\*;

@WebServlet("/hello")

public class HelloWorld extends HttpServlet

{

@Override

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException

{

PrintWriter out = response.getWriter();

out.println("Hello World!");

}

}

Q2: Describe in detail session tracking, why we need session tracking. How we work with cookies, describe the benefits of cookies, its creating, modifying and removing cookies.

**Session Tracking**

There are four techniques which can be used to identify a user session.

a) Cookies

b) Hidden Fields

c) URL Rewriting

d) Session Object

**Cookie**

Cookie is a key value pair of information, sent by the server to the browser and then browser sends back this identifier to the server with every request.

**Hidden Field**

Hidden fields are similar to other input fields with the only difference is that these fields are not displayed on the page but its value is sent as other input fields. For example

**<input type=”hidden” name=”sessionid” value=”unique value”/>**

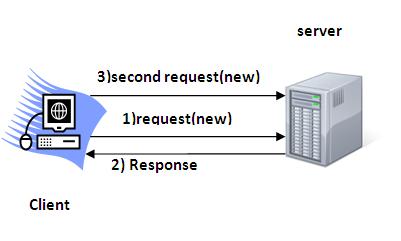
**URL Rewriting**

URL Rewriting is the approach in which a session (unique) identifier gets appended with each request URL so server can identify the user session. For example if we apply URL rewriting on [**http://localhost:8080/jsp-tutorial/home.jsp**](http://localhost:8080/jsp-tutorial/home.jsp)

**Session Object**

Session object is representation of a user session. User Session starts when a user opens a browser and sends the first request to server. Session object is available in all the request (in entire user session) so attributes stored in Http session in will be available in any servlet or in a jsp.

**Session Tracking**

is a way to maintain state (data) of an user. It is also known as **session** management in servlet. Http protocol is a stateless so **we** need to maintain state **using session tracking** techniques. Each time user requests to the server, server treats the request as the new request.

**Benefits of cookies**

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.

**How we work with cookies?**

By default, each request is considered as a new request. In cookies technique, we add cookie with response from the servlet. So cookie is stored in the cache of the browser. After that if request is sent by the user, cookie is added with request by default. Thus, we recognize the user as the old user.

**What are advantages of cookies?**

1. Simplest technique of maintaining the state.
2. Cookies are maintained at client side.
3. How can we delete cookie?
4. Let's see the simple code to delete cookie. It is mainly used to logout or signout the user.
5. Cookie ck=**new** Cookie("user",""); //deleting value of cookie
6. ck.setMaxAge(0); //changing the maximum age to 0 seconds
7. response.addCookie (ck); //adding cookie in the response.

Q3: Discuss JSP and JDBC, Basic steps in using JDBC, Define the connection URL, how to establish the connection also define the concept of creating statements , execute a quarry , process the results, close the connection.

**JDBC (Java Database connectivity)**

is a Java API used for connecting databases to the Java application.

**JSP (Java Server Pages)**

is a technology used for developing web pages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags. JSP is similar to PHP but it uses the Java programming language.

**Basic steps in using JDBC**

The fundamental steps involved in the process of connecting to a database and executing a query consist of the following:

* Import JDBC packages.
* Load and register the JDBC driver.
* Open a connection to the database.
* Create a statement object to perform a query.
* Execute the statement object and return a query result set.
* Process the result set.
* Close the result set and statement objects.
* Close the connection.

**URL**

A URL (Uniform Resource Locator), as the name suggests, provides a way to locate a resource on the [web](http://searchcrm.techtarget.com/definition/World-Wide-Web), the [hypertext](http://searchsoa.techtarget.com/definition/hypertext) system that operates over the [internet](http://searchwindevelopment.techtarget.com/definition/Internet). The URL contains the name of the [protocol](http://searchnetworking.techtarget.com/definition/protocol) to be used to access the resource and a resource name. The first part of a URL identifies what protocol to use. The second part identifies the [IP address](http://searchwindevelopment.techtarget.com/definition/IP-address) or [domain name](http://searchwindevelopment.techtarget.com/definition/domain-name) where the resource is located.

**Creating statements**

Once a connection is obtained we can interact with the database. The JDBC *Statement, Callable Statement,* and *Prepared Statement* interfaces define the methods and properties that enable you to send SQL or PL/SQL commands and receive data from your database. They also define methods that help bridge data type differences between Java and SQL data types used in a database.

try{

//STEP 2: Register JDBC driver

Class.forName("com.mysql.jdbc.Driver");

//STEP 3: Open a connection

System.out.println("Connecting to database...");

conn =DriverManager.getConnection(DB\_URL, USER, PASS);

//STEP 4: Execute a query

System.out.println("Creating database...");

stmt = conn.createStatement();

String sql ="CREATE DATABASE STUDENTS";

stmt.executeUpdate(sql);

System.out.println("Database created successfully...");

}

Q4: What are implicit objects in JSP and discuss its declarations, directives, attributes, and actions?

**Implicit Objects and their corresponding classes**:

|  |  |
| --- | --- |
| **out** | **javax.servlet.jsp.JspWriter** |
| request | javax.servlet.http.HttpServletRequest |
| response | javax.servlet.http.HttpServletResponse |
| session | javax.servlet.http.HttpSession |
| application | javax.servlet.ServletContext |

**JSP Declarations.**

A **JSP declaration** is used to **declare** variables and methods in a page's scripting language. The syntax for a **declaration** is as follows: <%!.

**JSP - Directives**

|  |  |
| --- | --- |
| Directive | Description |
| <%@ page ... %> | Defines page-dependent attributes, such as scripting language, error page, and buffering requirements. |
| <%@ include ... %> | Includes a file during the translation phase. |
| <%@ taglib ... %> | Declares a tag library, containing custom actions, used in the page |

**JSP Attribute**

There can be defined too many attributes for any custom tag. To define the attribute, you need to perform two tasks:

* Define the property in the Tag Handler class with the attribute name and define the setter method
* define the attribute element inside the tag element in the TLD file

**JSP - Actions**

|  |  |
| --- | --- |
| Syntax | Purpose |
| jsp:element | Defines XML elements dynamically. |
| jsp:attribute | Defines dynamically defined XML element's attribute. |
| jsp:body | Defines dynamically defined XML element's body. |
| jsp:text | Use to write template text in JSP pages and documents. |

Question 5 write a program that checks whether the number is a prime number or not ?

|  |  |
| --- | --- |
| Input | Output |
| <html><script>var x;  function isPrime(num) {  if(num < 2){ return 0;}  for (var i = 2; i < num; i++) {  if(num%i==0){ return 0;}  } return 1;}  function myFunction() {  x=parseFloat(num1.value);  var check = isPrime(x)  if(check == 1)  { alert("This Is Prime Number : "+x);  } else{  alert("This Is Not Prime Number : "+x);} num1.value="";}  </script><body><center><h1> Check Prime Number</h1>  Enter Number:<input type="number" id="num1"><br><br>  <input type="button" value="Addition" onClick="myFunction()"><br><br>  </body></html> |  |