

SPPU-TE-COMP-CONTENT - KSKA Git

Q1. What is JSP? Explain life cycle of JSP.

Ans. JSP (Java server pages) is technology used to develop dynamics, platform-independent web applications.

- It allows embedding Java code directly into HTML pages using special JSP tags.

→ JSP life cycle:

- JSP pages can be processed using JSP container:-

1) Client makes request for required JSP page server.

⊖ The server must have JSP container as request processed.

2) On receiving request JSP container searches and then read desired JSP page.

- This JSP page is straightforward converted to corresponding servlet.

- Every JSP element is converted into corresponding Java code.

⊖ 3) This servlet is then compiled to generate servlet class file.

- Using this response is generated.

4) The JSP container executes servlet class file.

- The compilation process involves three steps:-

1) Parsing the JSP.

2) Translating the JSP into a servlet

3) Compiling the servlet.

SPPU-TE-COMP-CONTENT - KSKA Git

Q2. What are various elements of JSP pages?

Ans. A JSP (Java Server Pages) page consists of several elements that allow dynamic content generation and interaction with Java-based backend logic.

The key elements of a JSP page are:-

1. Directives:

Directives provide global information about the JSP page to the container.

They are enclosed within `<%@ ... %>` tags.

eg:

```
<%@ page language = 'java' contentType = 'text/html';  
charset = UTF-8' pageEncoding = 'UTF-8' %>
```

2. Declarations:

Used to declare variables and methods that can be used within the JSP page.

Enclosed within `<%! ... %>` tags.

3. Scriptlets:

Contain Java code that is executed during request processing.

Enclosed within `<% ... %>` tags.

```
<% int number = 5; out.println("Number: " + number); %>
```

4. Expressions:

Used to output values directly into the response.

Enclosed within `<%= ... %>` tags.

eg:

```
<p> The counter is: <%= getCounter() %> </p>
```


SPPU-TE-COMP-CONTENT - KSKA Git

Q3. Discuss different ~~for~~ JSP actions in brief.

Ans. JSP action tags are the special tags that can be used to provide instructions to the JSP container on how to manage the server-side actions.

- They can enable the developers to perform specific tasks including the other files, forwarding the requests, and manipulating the session data within the JSP documents.

1. Include tag

- It can be used to include the content of the other resources like the JSP, HTML, or servlet in the current JSP page.

eg:

```
<jsp:include page = 'header.jsp' />
```

2. Forward tag

- This tag can be used to forward the current request to another resource like the JSP, HTML, or servlet without the client's knowledge.

eg:

```
<jsp:forward page = 'success.jsp' />
```

3. getPropertyTag

- This tag can be used to get the properties of the JavaBean component of the JSP pages.

eg:

```
<jsp:getPropertyTag name = 'user' property = 'username'  
value = 'John' />
```

→ Syntax:

```
<jsp:getProperty name = 'beanName' property = 'propertyName'  
value = 'propertyValue' />
```


SPPU-TE-COMP-CONTENT - KSKA Git

Q4. Discuss different directives in JSP with Example.?

Ans. • JSP (Java Server Pages) directives provide global instructions to the JSP container about how the page should be processed.

- These directives do not produce output directly but affect the entire JSP page.
- There are three main types of directives in JSP:

1. Page directive

- The page directive provides information about the JSP page, such as importing packages, defining error pages, session control, buffer size, etc.

eg:

```
<%@ page attribute = 'value' %>
```

2. Include directive

- The include directive is used to include the content of another JSP or HTML file at compilation time.
- This is useful for code reusability, such as including headers, footers, or menus.

eg:

```
<%@ include file = 'filename.jsp' %>
```

3. Taglib directive

- The taglib directive is used to import custom tag libraries in JSP.
- It is essential for using JSP standard Tag Library (JSTL) or custom tags.
- Syntax:

```
<%@ taglib prefix = 'prefix' uri = 'tag-library-uri' %>
```


SPPU-TE-COMP-CONTENT - KSKA Git

Q6. Explain implicit objects in JSP.

Ans. Implicit objects are the Java objects that the JSP container makes available to the developers on each page and the developer can call them directly without being explicitly declared.

- JSP implicitly objects are also called pre-defined variables.
- Following table lists out the none implicit objects that JSP supports:-

1. request

- This is the `HttpServletRequest` object associated with the request.

2. response

- This is the `HttpServletResponse` object associated with the response to the client.

3. out

- This is the `PrintWriter` object used to send output to the client.

4. session

- This is the `HttpSession` object associated with the request.

5. application

- This is the `ServletContext` object associated with the application context.

6. config

- This is the `ServletConfig` object associated with the application context.