Unit - 4

Servlet

classmate
Date 8/11/16
Page

J2EE is just a specification by java that aims to fullfill software needs of enterprise there are several APIs and technologies such as JSP, EJB etc. which make J2EE a working platform. Specification is a set of rules defined by any company which has to be followed by all the vendors.

For Ex: The specification for 3 pin socket we see in our home defines the space between pin, length of pin and breadth of pin. All three pin socket manufacturer follow the specification

★ Typical Enterprise has following needs.

- Use existing legacy system along with new system.
- Security.
- Easy upgrade to latest technology.
- Stability and reliability of software.
- Availiability of software developers to support.
- Develop complex software such as e-commerce website.

Some advantages are

1. Low cost.
2. good support from community.
3. Open source.
4. Good Documentation.
5. Availability of good J2EE Developers.
6. Low level services are already implemented.
7. J2EE uses mutitier distributed application model.

classmate
Date 10/11/16
Page

★ J2EE APIs & Technologies:

1) JDBC
2) JSP
3) Servlet
4) THREADS
5) G-mail (Java mail)
6) EJB
7) R.M.I.

★ Servelet classes and Interfaces.

1) **HTTP Servelet Request**
Extends the servelet ~~request~~ request interface to provide request information for Http servlets.

2) **HTTPServletResponse**
Extends the servlet response interface to provide http specific functionality in sending ~~up~~ a response.

3) **Http Session:**
It provides a way to identify a user across more than one page request or visit to a website and to store information about the user

4) **HttpSessionActivationListener:**
Object that are bound to an session may listen to container events notifying that sessions will be activated

5) **HttpSessionAttributeListener**
This Listener interface can be implemented in order to get notification of changes to the attribute list of

Date 11/11/16
Page

sessions within the web application.

6) HttpSessionBindingListener:
Causes an object to be notified when it is bound to or unbound from a session.

★ Servlet classes

1) Cookies:
A cookie is a small amount of information send by a servelet to a web-browser, saved by the web browser and later send back to the server.

2) HttpServlet:
It provides an abstract class to be sub classed to create an Http Servlet suitable for a website.

3) HttpServletRequestWrapper:
It provides a convinient implementation of the HttpServletRequest interface that can be sub classed by developers i.e. wish to adapt the request to a Servlet.

4) HttpServletResponseWrapper:
It provides a convinient implementation of the HttpServletResponse

5) HttpSessionBindingEvent:
Events of this type are either send to an object that implements HttpSessionBindingListener when it is bound or unbound from a session, or to a HttpSessionAttributeListener

that has been configured in the deployment descripter when any attribute is bound, unbound or replaced in a session.

6) Http Session Event:
This is the class representing event notifications for change to session within a web application.

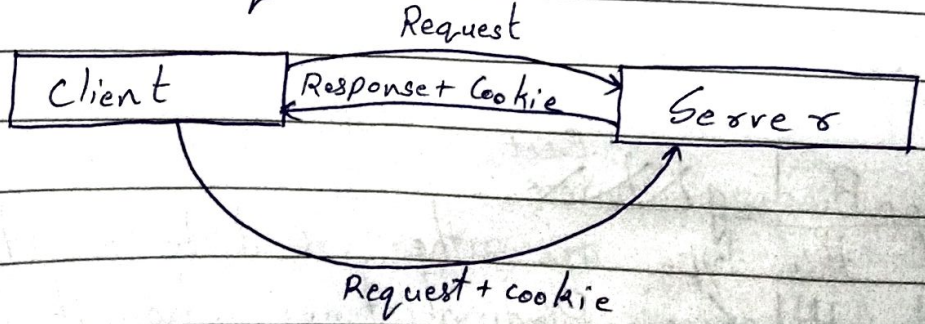
★ Session Tracking in Servlet / JSP:

- Session simply means particular interval of time.
- Session Tracking is a way to maintain ~~session~~ state (Data of the user).
- It is also known as session management in servlet.
- Http is a stateless so we need to maintain state using session tracking techniques

There are four techniques used in session Tracking.

1) Cookies
2) Hidden from fields
3) URL Rewritting
4) HttpSession Interface

1) Cookies:
Cookies is a small piece of information persisted between multiple client request.



How cookie works

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 send by the user, cookie is added with request by default.

"javax.servlet.http.cookie.*;" package:

How to create cookie

★ J2EE Container

A container is a runtime support of a system level entity.

There are four types of container

1) Applet Container
2) Application-client Container
3) Web container
4) EJB Container.

1) Applet Container

- Applet container is a Java program embedded into a web page.
- To use an applet in an html document the tags are used
- They are used to indicate to the browser that a Java Applet should be loaded.
- Applet container manages the execution of applet and contains the web browser.

Q:
Ans

2) Application - Client Container

- Application Client container is a combination of Java class, libraries and other files.
- They are used to distribute along with Java Client programs that execute on there own JVM.

3) Web Container:

- A Web Container is a part of web server.
- It provides the runtime environment to execute web applications such as JSP, Servlet.

- A servlet container translates the URL request into servlet request.
- The JSP implicite objects Request, Response, page, out, page context etc are exposed by JSP container.

4) EJB Container

- It provides runtime environment to execute EJB components such as EJB, Enterprise beans.
- EJB container manages transactions, state management details, multithreading, connection ~~pooling~~ pooling.
- The applications are provided with security using EJB container.
- All database access require by the Entity bean will be handled by the EJB container.

Q:-> What are comman task performed by Servlet container.

Ans:-> Servlet container ~~are~~ is also known as web containers.
For ex. Tomcat server
Some of the important task performed by servlet container ~~are performed by servlet container~~.

1) Communication support.
Servlet container provides easy way of communication between web client (browser) and the servlets and JSP's because of container. We don't need to built a server Socket to listen for any request from web client, ~~parts~~ parse the request and generate response all these important and complex task are done by container and all we need to focus is on bussiness logic for the applications.

2) Life cycle and resource Management:
Servlet container takes care of managing the life cycle of servlet. From the loading of servlets into memory, initializing

servlets, invoking servlet methods and to destroy them. Container provides utility for resource pooling and management

3) Multi threading Support:
Container creates new thread for every request to the servlet and provide them request and response object to process. So servlets are not initialized for each request and saves time and memory.

4) JSP Support:
JSP does not look like normal java classes, but every JSP in the application is compiled by the container and converted to servlet and then container manages them like other servlets

5) Miscellaneous Task:
Servlet container manages the resources pool, perform memory optimization execute garbage collector provide security configuration and support multiple applications.

★ Servlet Config & ServletContext (difference)

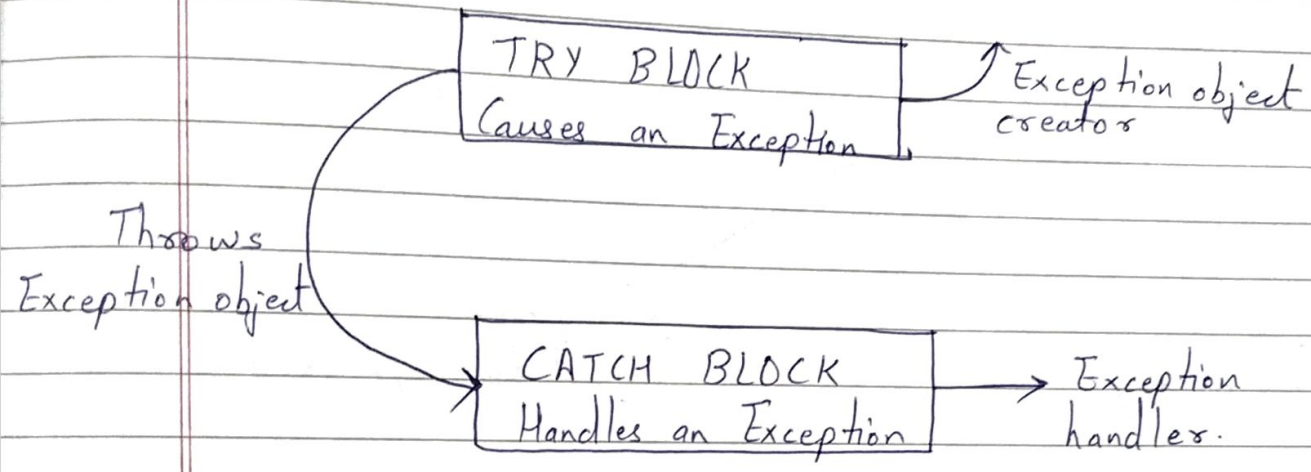
1. Servlet config is a unique object per servlet whereas ServletContext is a unique object for complete application.

2. Servlet config is used to provide init parameters to the servlet whereas Servlet context is used to provide application level init parameters that all other Servlets can use.

3. We can't set attributes in ServletConfig object whereas we can set attributes in ServletContext that other servlets can use in their implementation.

☆ Exception - Handling:

An exception is an condition that is caused by the runtime error in the program. When the java interpreter encounters an error such as dividing by zero, it creates an exception object and throws it



There are five handlers for handling an Exception.

1) Try
2) Catch
3) finally
4) Throw
5) Throws.

Ex :->

```
class A
{
public static void main (String s[])
{
try
{
System.out.println("Hello");
int i = 5/0;
System.out.println("Bye");
```

```
}
catch (Exception e)
{
System.out.println("e");
  }
}
}
```