

# J2EE Basics Revision

What is Java EE/Jakarta EE ? Specification for prim services , necessary for building enterprise apps.

eg of prim services : Servlet specs , JSP, Conn pooling , security ,JPA , Java Mail , web services....

Why J2EE 1.8/Java EE 8 /Jakarta EE 8

1. Java EE server independent
2. Supports variety of clnts -- thin clnts/ thick clnt /smart clnts
3. Prim services are imple in rdy manner : by web / app server --so that prog can focus only B.L

Who provides Specs of prim services in J2EE / Java EE : Oracl/Sun / Eclipse (Jakarta EE)

Implementation : Server vendors

Popular vendors & their products

(Apache : Tomcat / Tomee, Oracle --Glassfish , Jboss , weblogic....)

Request response flow(Layers)

URL --- http://www.abc.com:9090/day2.1

URL : Uniform resource locator

Revise request response flow

http : scheme(protocol : app layer)

www.abc.com : DNS qualified host name (IP adr) ---IP layer resolves port no --req reaches host

9090 : port no --resolved by TCP -- req reaches web server

web server --> chks path(URI) --->if itcontains any dyn web app --delegates it to WC

/day2.1 : ctx path (web app name)

WC --routes request to the specific web app

web.xml --- Dep descriptor meant for WC per web app

what are contents ? -- XML tags

welcome-file : index.html (Location : / : root of web =>webapp) => public

Resp : SC 200 | Header/s | body : index page ---> clnt side

What is WC ?

Server side JVM

Manages web apps.

Provides run time env for server side dynamic web comps(servlets, JSP,filter)

1. Server side JVM residing within web server.

Its run-time environment for dynamic web components(Servlet & JSP,Filter) .  
Jobs ---

1. Creating Http Request & Http response objects

2. Controlling life-cycle of dyn web comps (manages life cycle of servlet,JSP,Filters)

3. Giving ready-made support for services --- Naming,security,Conn pooling .

4. Handling concurrent request from multiple clients .

5. Managing session tracking...

2. What is web.xml? --- Deployment descriptor one per web appln

populated by -- prog (with help IDE)

who reads it -- WC

when ---once at the web app deployment time

what does it consist of --- XML tags based instructions meant for WC

Mandatory till Java EE 1.5 , optional later.

J2EE compliant folder structure of dynamic web application

refer to a diagram.("day1\_data\day1\_help\web programming related diags\j2ee compliant web appln folder structure.png")

3. Why servlets??? Adds dyn nature

Job list

1. Request processing

2. B.L

3. Dynamic response generation

4. Data access logic(DAO class --managing DAO layer)

5. Page navigation

What is a servlet ?

-- Java class (with NO main method) -- represents dynamic web component - whose life cycle will be managed by WC(web container : server side JVM)

life cycle methods --- init , service,destroy

Servlet API details --refer to diag servlet-api.png (from above folder) : IMPORTANT

<Tomcat\_Home>\lib --- servlet-api.jar => specs only !

imple left to vendors --catalina.jar => imple classes

API

javax.servlet.Servlet i/f ---- init , service,destroy

imple class : javax.servlet.GenericServlet --abstract

abstract method : service

concrete : init ,destroy

sub class -- javax.servlet.http.HttpServlet : abstract class , NO abstract methods

public void service(ServletRequest rq ,ServletResponse rs) throws ServletExc, IOExc

--> protected void service(HttpServletRequest rq ,HttpServletResponse rs) throws

ServletExc, IOExc  
--> dispatching  
get --> doGet  
post --> doPost ...

Steps of creating the servlet

```
public class MyServlet extends HttpServlet
{
    @Override
    public void doGet(HttpServletRequest rq ,HttpServletResponse rs) throws
ServletExc, IOExc
    {
        rs.setContentType("text/html");
        try(PW pw=rs.getWriter())
        {
            pw.print(.....);
        } //pw.close -->pw.flush--> committing the resp
    }
}
```

Deployment of the servlet

```
@WebServlet(value="/hello",loadOnStartup=1)
```

.....

WC creates a map : Key -- /hello

value : F.Q servlet cls name

OR

XML Tags

For HTTP response status codes :

refer : <https://www.restapitutorial.com/httpstatuscodes.html>

For primer in TCP/IP in web programming

[https://www.techtarget.com/searchnetworking/definition/TCP-IP#:~:text=TCP%2FIP%20stands%20for%20Transmission,\(an%20intranet%20or%20extranet\).](https://www.techtarget.com/searchnetworking/definition/TCP-IP#:~:text=TCP%2FIP%20stands%20for%20Transmission,(an%20intranet%20or%20extranet).)

How does WC store the mapping info of the servlet ???

In the map created @ web app dep time

key :

value :

Trace Servlet Life cycle

URL request after Client clicks on the link : "Invoke A Servlet"

<http://localhost:8080/day2.1/hello>

WC -- chks for the key(/hello) ---found --1st request ---WC starts life cycle

Key (URL pattern) not found --- HTTP 404

class loading ---singleton instance ---init => init seq

init --success ---???

fail --ServletExc --- WC ABORTS life cycle

At the end :

public void destroy() : inherited from G.S --> servlet instance is marked for GC  
Triggers : server shut down / reload (re deploy) /un deploy  
-----

Deployment of the servlet

1. Via annotation

eg : @WebServlet(value="/validate")  
public class LoginServlet extends H.S {...}  
Map :  
key -- /validate  
value -- F.Q servlet cls name  
URL : http://host:port/day1.1/validate?....

OR

2. Using XML tags

How to deploy a servlet w/o annotations --- via XML tags  
web.xml

```
<servlet>
  <servlet-name>abc</servlet-name>
  <servlet-class>pages.SecondServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>abc</servlet-name>
  <url-pattern>/test2</url-pattern>
</servlet-mapping>
WC : map
key : /test2
value : pages.SecondServlet
```

eg URL --http://host:port/test\_web/hello

At the time of web app deployment ---WC tries to populate map of url patterns ,  
from XML tags (from web.xml). Later ---it will check for @WebServlet annotation

Objective 3 .

What is default loading policy of WC for servlets ? lazy (i.e WC will start life  
cycle of the servlet :only after clnt sends the 1st request to the servlet)

Can you change it to eager ? Yes

Use Case : Typically for time consuming(heavy weight) inits.

eg : setting up DB conn, setting up spring framework....

HOW ?

```
@WebServlet (value="/test",loadOnStartup=1)
```

```
.....
```

OR

xml tag :

```
<load-on-startup>1</....>
```

Centralized err handling in Servlets

How ?

```
@Override
```

```
public void init() throws ServletException
```

```
{
```

```
try {
```

```
    open connection
```

```
    create dao inst
```

```
} catch(Exception e)
```

```
{
```

```
//re throw the exception back to WC , so that WC doesn't continue to service.
```

```
    throw new ServletException("err in init" +getClass().getName(),e);
```

```
}
```

```
}
```

API of javax.servlet.ServletException

Constructor :

```
public ServletException(String message,Throwable rootCause)
```