

Dt : 19/12/2023

3.FilterConfig:

=>FilterConfig is an interface from javax.servlet package and which is instantiated

automatically when filter-program is loaded for execution.

=>This FilterConfig Object is loaded with Filter_name.

=>To access FilterConfig object reference we must use init()-method

=>we use <init-param> subtag part of <filter> tag to initialize parameter-values

to FilterConfig Object.

syntax:

<web-app>

...

<filter>

<filter-name> name </filter-name>

<filter-class> class </filter-class>

<init-param>

<param-name> name </param-name>

<param-value> value </param-value>

</init-param>

</filter>

<filter-mapping>

`<filter-name> name </filter-name>`

`<url-pattern> /url </url-pattern>`

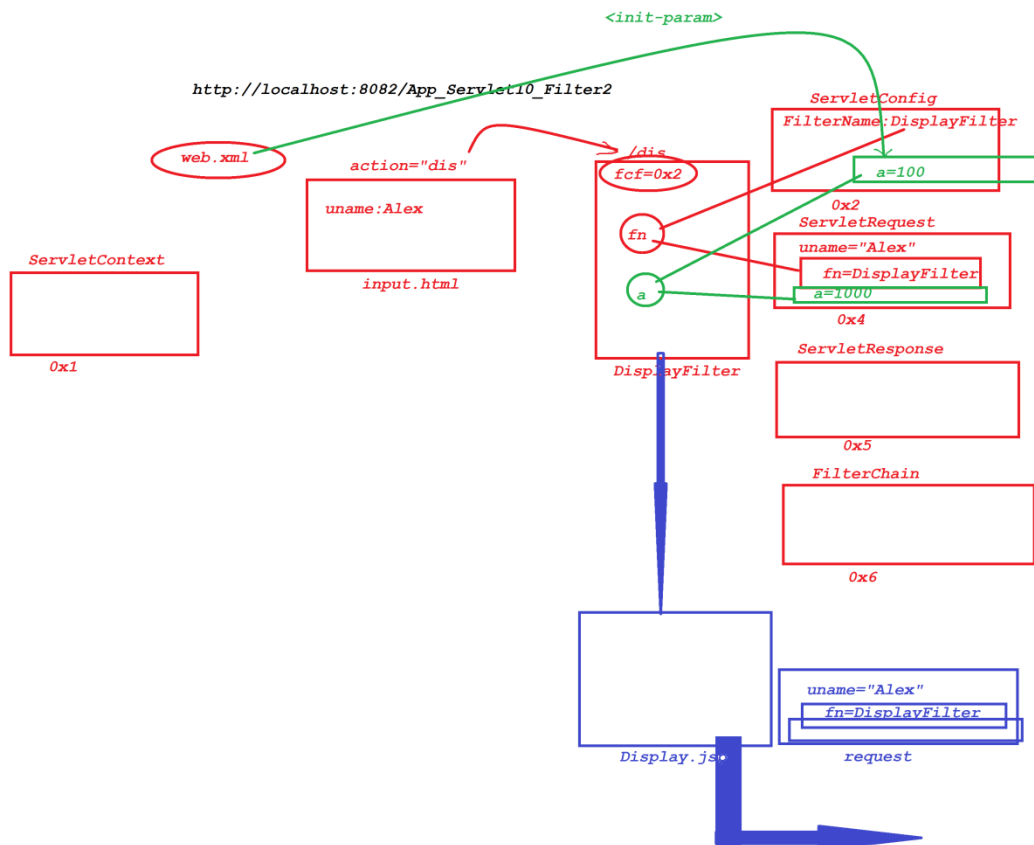
`</filter-mapping>`

...

`</web-app>`

Ex-Application:(Demonstrating FilterConfig)

Layout:



input.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="dis" method="post">
UserName:<input type="text" name="uname"><br>
<input type="submit" value="Display">
</form>
</body>
</html>
```

DisplayFilter.java

```
package test;
```

```
import java.io.*;
```

```
import javax.servlet.*;
```

```
public class DisplayFilter implements Filter{
```

```
    public FilterConfig fcf=null;
```

```
        @Override
```

```
    public void init(FilterConfig fcf)throws ServletException{
```

```
        this.fcf=fcf;
```

```
    }
```

```
        @Override
```

```
    public void doFilter(ServletRequest req,ServletResponse res,FilterChain
chain)throws
```

```
        ServletException,IOException{
```

```
        req.setAttribute("fn",fcf.getFilterName());
```

```

    req.setAttribute("a", fcf.getInitParameter("a"));

    req.getRequestDispatcher("Display.jsp").forward(req, res);

}

}

```

Display.jsp

```

<%@ page language="java"
    contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
String uName = request.getParameter("uname");
String fn = (String)request.getAttribute("fn");
out.println("Welcome UserName : "+uName+"<br>");
out.println("====FilterConfig====");
out.println("<br>FilterName: "+fn);
out.println("<br>FilterConfig
Val: "+request.getAttribute("a"));
%>
</body>
</html>

```

web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <filter>
        <filter-name>DisplayFilter</filter-name>
        <filter-class>test.DisplayFilter</filter-class>
        <init-param>
            <param-name>a</param-name>

```

```

        <param-value>1000</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>DisplayFilter</filter-name>
    <url-pattern>/dis</url-pattern>
</filter-mapping>
    <welcome-file-list>
        <welcome-file>input.html</welcome-file>
    </welcome-file-list>
</web-app>

```

=====

=====

faq:

wt is the diff b/w

(i)ServletConfig

(ii)FilterConfig

(iii)JSP-Config

(i)ServletConfig:

=>ServletConfig is instantiated automatically when Servlet-program loaded for

execution and which holds Servlet_name.

(ii)FilterConfig:

=>FilterConfig is instantiated automatically when Filter-program is loaded for

execution and which holds Filter_name

(iii)JSP-Config:

=>JSP-Config means ServletConfig and which is instantiated automatically when

JSP-program loaded for execution.

=>This JSP-Config is accessed with implicit name "config"

=====

***imp**

Events and Listeners in Servlet Programming:

define Events?

=>The actions which are performed are known as Events.

define Listeners?

=>Listeners are executed background to Servlet-Objects and record the actions performed on Servlet-Objects.

=>The Listeners in Servlet programming can be added to the following Servlet-Objects

1.ServletContext

2.ServletRequest

3.HttpSession

1.ServletContext:

=>we add 'ServletContextListener' to ServletContext Object

Listener_name : ServletContextListener

public void contextInitialized(javax.servlet.ServletContextEvent);

public void contextDestroyed(javax.servlet.ServletContextEvent);

Event_name : ServletContextEvent

public javax.servlet.ServletContext getServletContext();

Attribute_Lisnter_name : ServletContextAttributeListener

public void attributeAdded(javax.servlet.ServletContextAttributeEvent);

public void attributeRemoved(javax.servlet.ServletContextAttributeEvent);

public void attributeReplaced(javax.servlet.ServletContextAttributeEvent);

Attribute_Event_name : ServletContextAttributeEvent

public java.lang.String getName();

public java.lang.Object getValue();

2.ServletRequest:

=>we add 'ServletRequestListener' to ServletRequest Object

Listener_name : ServletRequestListener

public void requestInitialized(javax.servlet.ServletRequestEvent);

public void requestDestroyed(javax.servlet.ServletRequestEvent);

Event_name : ServletRequestEvent

public javax.servlet.ServletRequest getServletRequest();

public javax.servlet.ServletContext getServletContext();

Attribute_Listener_name : ServletRequestAttributeListener

public void attributeAdded(javax.servlet.ServletRequestAttributeEvent);

public void attributeRemoved(javax.servlet.ServletRequestAttributeEvent);

public void attributeReplaced(javax.servlet.ServletRequestAttributeEvent);

Attribute_Event_name : ServletRequestAttributeEvent

public java.lang.String getName();

public java.lang.Object getValue();

3.HttpSession:

=>we add 'HttpSessionListenter' to HttpSession object

Listener_name : HttpSessionListener

public void sessionCreated(javax.servlet.http.HttpSessionEvent);

public void sessionDestroyed(javax.servlet.http.HttpSessionEvent);

Event_name : HttpSessionEvent

public javax.servlet.http.HttpSession getSession();

Attribute_Listener_name : HttpSessionAttributeListener

public void attributeAdded(javax.servlet.http.HttpSessionBindingEvent);

public void attributeRemoved(javax.servlet.http.HttpSessionBindingEvent);

public void attributeReplaced(javax.servlet.http.HttpSessionBindingEvent);

Attribute_Event_name : HttpSessionBindingEvent

public java.lang.String getName();

public java.lang.Object getValue();

=====