Note:

GenericSevlet is an best example for "Adapter class design pattern".

init() is overloaded in GenericServlet.

Eg: FourthApp

Note:

By default response type/content type is "text/html"

There is no need to write html head and body tag , by default every thing you write will be in the html body tag.

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns="/test")

public class FourthServlet extends GenericServlet{

public void service(ServletRequest req , ServletResponse res)throws ServletException,IOException{

PrintWriter out = res.getWriter();

out.println("<h1 style = 'color : blue';> writing servlet using generic servlet </h1>");

out.close();

}

}

Behind the scenes

=================

In the above code 2 .class files will be used

a. FourthServlet.class

b. GenericServlet.class

=> Loading:: Container will load FourthServlet.class file for the url pattern ("/disp")

=> Instantiation:: Container will create an Object for FourthServlet.class ( for abstract class object cannot be created GenericServlet (Ac) ).

=> Initialization:: Container will call init(), First it will check in FourthServlet.class if not, it would check in GenericServlet

init() is available inside GenericServlet but it has 2 methods with the name

init(SC config)

init()

Can we override the init logic?

We can override, but it is a good practise to override only init(),but not init(SC config) becoz config is local variable in init(SC config),and the config variable memory would be gone once the control comes out of the init(SC config)

so better override init() but not init(SC config).

Internally implementation, in predefined code even though config is a local variable, it is reused as a instance variable as shown below.

If we try to override int( SC config ) connection w.r.t instance variable config will be broken, which is used again in int() method.

Code in GenericServlet

=====================

public abstract class GenericServlet implements Servlet,ServletConfig,Serializable{

private transient SC config;

public void init(SC config)throws SE{

this.config=config;

init();

}

public void init() throws SE{

;;;;;;

}

}

=> RequestProcessing phase :: Container will call service(req,resp) to provide response to the client.

First it will check in FourthServlet.class if not, it would check in GenericServlet.

service(req,resp) is available inside GenericServlet as abstract and we need to give the body of this method inside FourthServlet as shown in the above program.

=> ServletDelnstantion=> Contianer will call destroy() to perform De-Instantion action.

First it will check in FourthServlet.class if not, it would check in GenericServlet.

destroy() is not avaialble in FourtServlet.class so it would take from GenericServlet.class and it

will execute.

Note:

1. If our servlet class does not contains init() method

a. GenericServlet: init(SC)

b. GS: init()

c. User: service(req,resp)

2. If our servlet class contains init(SC) method

a. US: init(SC)

b. US: service(req,resp)

3. If our servlet class contains init() method

a. GS: init(SC)

b. US: init(SC)

c. US: service(req,resp)

why 2 init(),init(SC config) in GenericServlet?

init(SC config) -> container

init() -> developer

Which init method is best suited for developer?

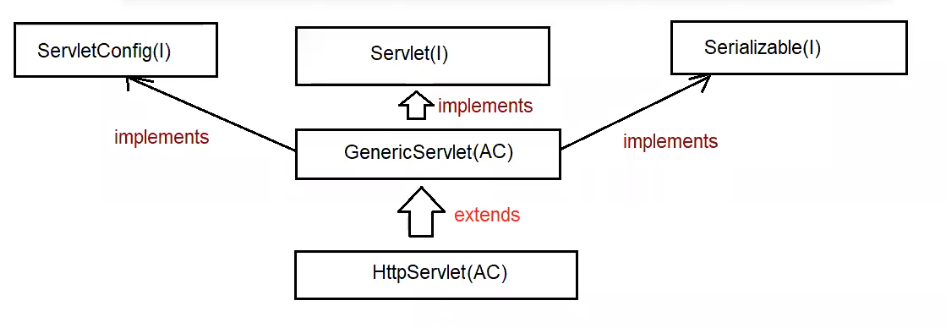
init() => best suited for writing initialziation logic.

Different ways of Creating a Servlet:

================================

1. Servlet(I) ========== > 5 abstract methods

2. GenericServlet(AC) == > one abstract method (pvs(ServletRequest request, ServletResponse response)



Note:

When we build webapplications, internally httpprotocol is used and while sending the request, the request type can be

a. POST

b. GET

We can build Servlet in a easier way with the help of GenericServlet, then y need HttpServlet(AC)?

Ans. In case of GenericServlet(AC), to process the request we have only one method(pvs(req,resp)) which is generic for any type of request like GET, POST, ....

Because there is only one method available which is generic for any type of request, Debugging the application becomes difficult.

We can build Servlet in a easier way with the help of GenericServlet, then y need HttpServlet(AC)?

Ans. In case of GenericServlet(AC), to process the request we have only one method(pvs(req,resp)) which is generic for any type of request like GET,POST, ....

Becoz there is only one method available which is generic for any type of request, Debugging the application becomes difficult.

henceforth to deal with only Httpprotocol, we have a special approach to create a servlet called "HttpServlet".

Note:

C:\Users\nitin>javap javax.servlet.http.HttpServlet

Compiled from "HttpServlet.java"

public abstract class HttpServlet extends GenericServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException;

protected void doPost(HttpServletRequest request,HttpServletResponse response) throws ServletException,IOException;

protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException;

public void service(ServletRequest request, ServletResponse response) throws ServletException,IOException;

}

request method type is POST === > doGet(request,response)

request method type is GET === > doPost(request,response)

Note:

For a webapplication how to send GET request?

a. type url in the address bar of browser and hit the request.

b. by clicking the hyperlink in a webpage.

c. using <form method="GET">

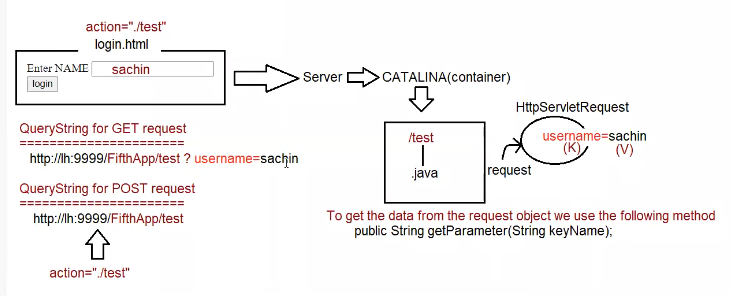
d. default <form> method attribute is GET only.

Eg: FifthApp

For a webapplication how to send POST request?

1. using <form method="POST">

Eg: SixthApp



Eg: SeventhApp

Life Cycle of HttpServlet :

=====================

1. when we submit the form browser prepares HttpRequest and sends to server.

2. WebServer checks the request is for static/dynamic information.

3. If the request is for Static information then webserver provides required information (copy and paste)

if available otherwise 404 Status Code(Saying the requested resource is not available).

4. If the request is for dynamic information, then webserver hands over the control to "catalina" container.

5. wecontainer identifies the request based on "web.xml" or through "annotation".(/test)

6. webcontainer will check whether the ServletObject(TestServlet) is available or not (/test ==== > TestServlet.class)

7. If the Servletobject is not available, then it will perform the following action

a. Loading === > static block

b. instantiation === > constructor

c. initialization === > init()[same as GenericServlet lifecyle]

8.RequestProcessing phase

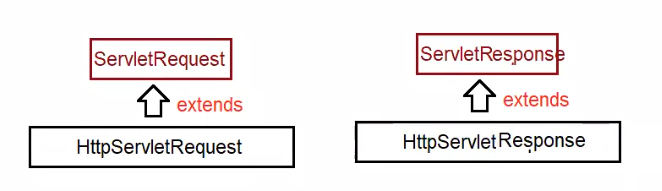
a. webcontainer will create ServletRequest,ServletResponse object by invoking

public void service(ServletRequest request, ServletResponse response) throws ServletException,IOException;

container will check in our class service (ServletRequest,ServletResponse) is available or not, if it is avaialble it will execute our service() only and provides the response.

if our servlet class does not contain service(), then container will execute HttpServlet service method, which is overridden method of parent GenericServlet in HttpServlet.

public void service (ServletRequest request, ServletResponse response)



Internal code

HttpServlet:

==========

public void service(ServletRequest request,ServletResponse response){

HttpServletRequest hreq = (HttpServletRequest)request;

HttpServletResponse hresp = (HttpServletResponse)response;

service(hreq,hresp); //protected service(hreq,hresp)

}

webcontainer will call protected service (HttpServletRequest hreq,HttpServletResponse hresp) throws SE,IOE

if our class contains protected void service (HttpServletRequest hreq, HttpServletResponse hresp) throws SE, IOE then container will call our class service() only.

if our class does not contains protected void service (HttpServletRequest hreq, HttpServletResponse hresp) throws SE, IOE then container will call HttpServlet class service()

HttpServlet: ( internal code )

protected void service (HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException {

String requestType = request.getMethod();

if(requestType.equals("GET")){

doGet(request,response);//protected/public doGet(request,response);

}

else if (requestType.equals("POST")){

doPost(request,response);//protected/public doPost(request,response);

}

;;;;;;;

;;;;;;;

else

return 501 status code saying Http method not implemented

}

webcontainer will check whether our servlet class contains doXXXX(req,resp).

if it contains doXXXX(req,resp) it will be executed and it provides the response.

if our servlet class does not contain doXXXX(req,resp) then HttpServlet class doXXXX(req,resp) will be called.

HttpServlet

=============

protected void doXXXX(HttpServlet request,HttpServletResponse response ) throws SE,IOE{

return 405 | 400 status code saying HttpMethod GET is not supported by this URL.

}

hierachy of calling the methods

a. public service(SRreq,SResp)

b. protected service(HSReq,HSResp)

c. public void doXXXX(HSReq,HSResp)

case1:

If our servlet class contains public void service(SReq,SResp) then for every type of request(POST,GET) same method will be executed.

case2:

If our servlet class contains public void service(SReq,SResp) and protected void servcie(HSReq,HSResp) then for every type of request(POST,GET) public void service(SReq,SResp) same method will be executed.

case3:

If our servlet class contains protected void servcie(HSReq,HSResp) and doGet() then for every type of request(POST,GET) protected void service(HSReq,HSResp) same method will be executed.

case4:

we are sending GET request,but our servlet doesnot contain doGet(),it contains doPost() then which method would be called?

it calls HttpServlet doGet() which would send 405 status code.

case5:

we are sending POSt request,but our servlet doesnot contain doPost(),it contains doGet() then which method would be called?

it calls HttpServlet doPost() which would send 405 status code.

case6:

Assume we need to give common response for both the request type, then how to code?

Eg: EighthApp

Eg: NinthApp

Playing with request Object

========================

To retrieve only one value from request object

public abstract java.lang.String getParameter(java.lang.String);

To retrive multiple values from request object

public abstract java.lang.String[] getParameterValues(java.lang.String);

To know the type of request from request object

public abstract java.lang.String getMethod();

Eg: TenthApp

Different ways of Creating a Servlet

1. Servlet(I)

2. GenericServlet (AC)

3. HttpServlet (AC)

Dynamic response will be generated by Servlet.

Mapping a resource to particular url-pattern for the web-container can be configured in 2 ways

a. XML (legacy approach)

b. Annotation (Available from Servlet3.0V)

Note:

user input will be sent in the form of QueryString from the browser to protocol, and container will store in request object

ServletRequest(I)

| extends

HttpServletRequest(I)

HttpRequest Structure

a.RequestLine(request type,resourcename,protocolversion)

b.RequestHeader(information about the client)

c.RequestBody(actual data(QueryString))

methods

=======

public abstract java.util.Enumeration<java.lang.String> getHeaderNames();

public abstract java.lang.String getHeader(java.lang.String);

Eg: RequestHeader