

Module : JEE

Servlet



Chapitre II: Servlet

I- Introduction

II-Servlet API

III- Cycle de vie

IV- Étapes à suivre pour créer un exemple de servlet

Chapitre II: Servlet

I- Introduction

- Servlet est une technologie qui est utilisée pour créer une application web.
- Servlet est une API qui fournit de nombreuses interfaces et classes, y compris de la documentation.
- Servlet est un composant Web qui est déployé sur le serveur pour créer une page Web dynamique,

Chapitre II: Servlet

II- Servlet API

Les packages `javax.servlet` et `javax.servlet.http` représentent des interfaces et des classes pour l'api des servlets.

- Le package `javax.servlet` contient de nombreuses interfaces et classes qui sont utilisées par les servlets ou les conteneurs web. Elles ne sont pas spécifiques à un protocole.
- Le package `javax.servlet.http` contient des interfaces et des classes qui sont responsables des demandes http uniquement.

Chapitre II: Servlet

II- Servlet API

`javax.servlet`

Interfaces de package `javax.servlet`

1. `Servlet`
2. `ServletRequest`
3. `ServletResponse`
4. `RequestDispatcher`
5. `ServletConfig`
6. `ServletContext`
7. `SingleThreadModel`
8. `Filter`
9. `FilterConfig`
10. `FilterChain`
11. `ServletRequestListener`
12. `ServletRequestAttributeListener`
13. `ServletContextListener`
14. `ServletContextAttributeListener`

Chapitre II: Servlet

II- Servlet API

`javax.servlet`

Classes de package `javax.servlet`

1. `GenericServlet`
2. `ServletInputStream`
3. `ServletOutputStream`
4. `ServletRequestWrapper`
5. `ServletResponseWrapper`
6. `ServletRequestEvent`
7. `ServletContextEvent`
8. `ServletRequestAttributeEvent`
9. `ServletContextAttributeEvent`
10. `ServletException`
11. `UnavailableException`

Chapitre II: Servlet

II- Servlet API

`javax.servlet.http`

Interfaces de package `javax.servlet.http`

1. `HttpServletRequest`
2. `HttpServletResponse`
3. `HttpSession`
4. `HttpSessionListener`
5. `HttpSessionAttributeListener`
6. `HttpSessionBindingListener`
7. `HttpSessionActivationListener`
8. `HttpSessionContext` (deprecated now)

Chapitre II: Servlet

II- Servlet API

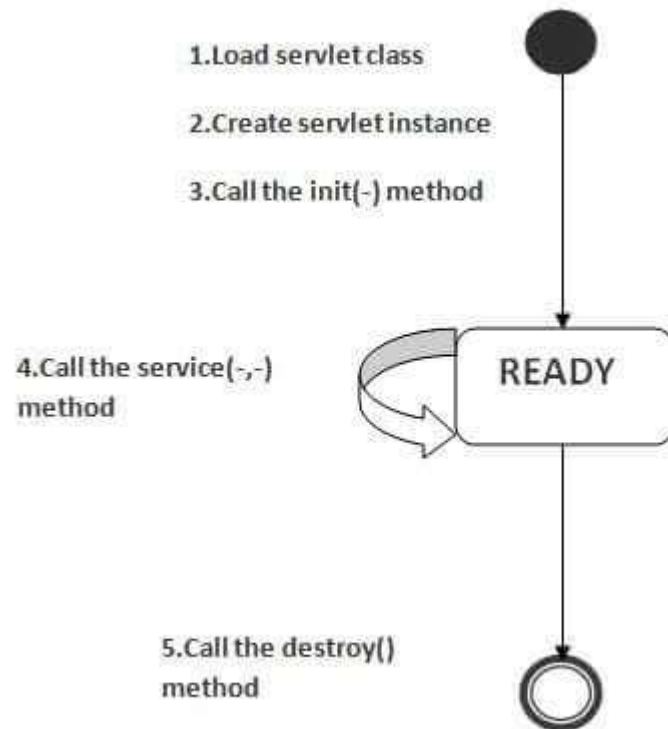
`javax.servlet.http`

Classe de package `javax.servlet.http`

1. `HttpServlet`
2. `Cookie`
3. `HttpServletRequestWrapper`
4. `HttpServletResponseWrapper`
5. `HttpSessionEvent`
6. `HttpSessionBindingEvent`

Chapitre II: Servlet

III- Cycle de vie



Chapitre II: Servlet

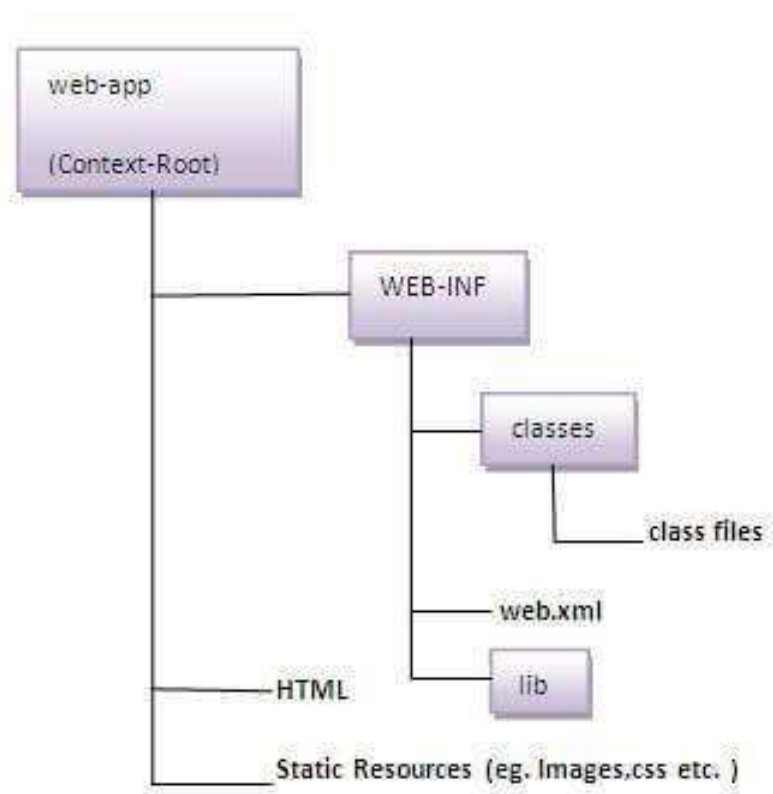
IV- Étapes à suivre pour créer un exemple de servlet

1. Créer une structure de répertoire
2. Créer une servlet
3. Compiler la servlet
4. Créer un descripteur de déploiement
5. Accéder à la servlet

Chapitre II: Servlet

IV- Étapes à suivre pour créer un exemple de servlet

1. Créer une structure de répertoire



Chapitre II: Servlet

IV- Étapes à suivre pour créer un exemple de servlet

2. Créer une servlet

Il existe trois façons de créer une servlet.

1. En implémentant l'interface Servlet
2. En héritant de la classe GenericServlet
3. En héritant de la classe HttpServlet

Chapitre II: Servlet

IV- Étapes à suivre pour créer un exemple de servlet

2. Créer une servlet

```
1 import javax.servlet.http.*;
2 import javax.servlet.*;
3 import java.io.*;
4 public class exempleServlet extends HttpServlet{
5
6     public void doGet(HttpServletRequest req,HttpServletResponse res)
7         throws ServletException,IOException {
8         res.setContentType("text/html");//setting the content type
9         PrintWriter pw=res.getWriter();//get the stream to write the data
10
11         //writing html in the stream
12         pw.println("<html><body>");
13         pw.println("Welcome to servlet");
14         pw.println("</body></html>");
15
16         pw.close();//closing the stream
17     }
18 }
```

Chapitre II: Servlet

IV- Étapes à suivre pour créer un exemple de servlet

3. Compiler la servlet

Pour compiler la servlet, le fichier jar doit être chargé. Les différents serveurs fournissent différents fichiers jar :

Jar file	Server
servlet-api.jar	Apache Tomcat
weblogic.jar	Weblogic
javaee.jar	Glassfish
javaee.jar	JBoss

Chapitre II: Servlet

IV- Étapes à suivre pour créer un exemple de servlet

4. Créer un descripteur de déploiement

Le descripteur de déploiement est un fichier **xml**, à partir duquel le conteneur Web obtient les informations sur la servlet à invoquer.

```
1  <web-app>
2
3  <servlet>
4    <servlet-name>servlet_1</servlet-name>
5    <servlet-class>exempleServlet</servlet-class>
6  </servlet>
7
8  <servlet-mapping>
9    <servlet-name>servlet_1</servlet-name>
10   <url-pattern>/ma_page</url-pattern>
11 </servlet-mapping>
12
13 </web-app>
```


Chapitre II: Servlet

WebApplication1 - NetBeans IDE 6.5

File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help

WebApplication1

- Web Pages
 - WEB-INF
 - index.html
 - index.jsp
 - Configuration Files
 - MANIFEST.MF
 - sun-web.xml
 - web.xml
 - Server Resources
 - Source Packages
 - <default package>
 - DemoServlet.java
 - Test Packages
 - Libraries
 - Test Libraries

web.xml - Navigator

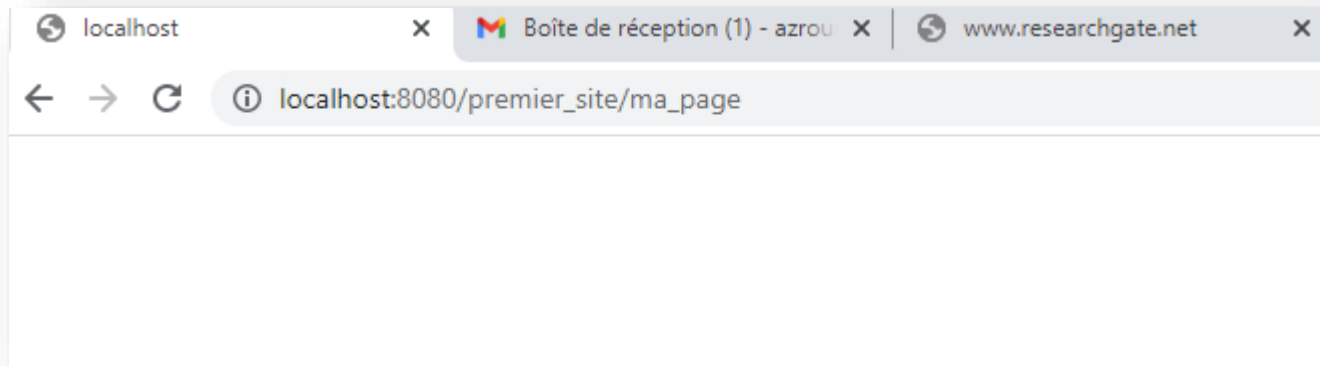
- <?xml version="1.0" encoding="UTF-8">
- <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee">
- <servlet>
- <servlet-name>DemoServlet</servlet-name>
- <servlet-class>DemoServlet</servlet-class>
- </servlet>
- <servlet-mapping>
- <servlet-name>DemoServlet</servlet-name>
- <url-pattern>/DemoServlet</url-pattern>
- </servlet-mapping>
- <session-config>
- <session-timeout>
- 30
- </session-timeout>
- </session-config>
- <welcome-file-list>
- <welcome-file>index.html</welcome-file>
- </welcome-file-list>
- </web-app>

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee">
  <servlet>
    <servlet-name>DemoServlet</servlet-name>
    <servlet-class>DemoServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>DemoServlet</servlet-name>
    <url-pattern>/DemoServlet</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
</web-app>
```


Chapitre II: Servlet

IV- Étapes à suivre pour créer un exemple de servlet

5. Accéder à la servlet



Chapitre II: Servlet

V- Récupère les valeurs de la requête (de client)

Un objet de **ServletRequest** est utilisé pour fournir les informations de la requête du client à une servlet, telles que le type de contenu, la longueur du contenu, les noms et valeurs des paramètres, les informations d'en-tête, les attributs, etc.

Chapitre II: Servlet

V- Récupère les valeurs de la requête (de client)

Méthodes utiles

Methode	Description
<code>public String getParameter(String name)</code>	Renvoie la valeur d'un paramètre par son nom.
<code>public String[] getParameterValues(String name)</code>	Renvoie un tableau de String contenant toutes les valeurs du nom du paramètre donné. Il est principalement utilisé pour obtenir les valeurs d'une liste de sélection multiple.
<code>java.util.Enumeration getParameterNames()</code>	Renvoie une énumération de tous les noms des paramètres de la demande.
<code>public int getContentLength()</code>	Renvoie la taille des données d'une requête (-1 si inconnue)
<code>public String getCharacterEncoding()</code>	Renvoie l'encodage des caractères d'une requête.
<code>public String getContentType()</code>	Renvoie le type de média des données (texte/ image..) ou null si non connu.
<code>public ServletInputStream getInputStream() throws IOException</code>	Returns an input stream for reading binary data in the request body.
<code>public abstract String getServerName()</code>	Renvoie le nom de serveur
<code>public int getServerPort()</code>	Renvoie le numéro du port sur lequel cette requête a été reçue.
<code>public int getRemoteAddr()</code>	Renvoie l'adresse de client (utilisateur)

Chapitre II: Servlet

V- Récupère les valeurs de la requête (de client)

Exemples

```
/////recuperer les en-tête HTTP
String cont_enc=req.getHeader("Accept-Encoding");
String cont_lan=req.getHeader("Accept-Language");
String cont_lon=req.getHeader("Authorization");

//recupérer de parametre 'name' envoyer par le formulaire ou une URL
String nom=req.getParameter("name");

///recuperer les en-tête HTTP
int  taille=req.getContentLength();
String cont_type=req.getCharacterEncoding();
```

Chapitre II: Servlet

V- Récupère les valeurs de la requête (de client)

Exemples

```
Enumeration en = req.getHeaderNames(); //import java.util.*;
while (en.hasMoreElements()) {
    String headerName = (String) en.nextElement();
    String headerValue = req.getHeader(headerName);
    out.print("<b>" + headerName + "</b>: " + headerValue + "<br>");
}
```

Chapitre II: Servlet

Tp_2_Creation d'une servlet simple