

Les Servlets

servlet = ?

- Une servlet est un programme (plug-in) à ajouter à un serveur (quel qu'il soit).
- Ce cours a trait à la programmation Java coté serveur (J2EE)
- Pour l'instant les serveurs acceptant des servlets sont plutôt des serveurs Web.
- Contre-exemple : serveur de mail qui détruit les mails contenant des virus.

Motivation et historique

- Nécessité d'avoir des pages HTML dynamiques i.e. pages créées lors de la requête (météo, cours de la bourse, vente aux enchères, etc.)
- Technologie des scripts CGI.
- Le serveur Web demande à lancer un programme par le protocole CGI
- Inconvénient : nécessite de créer un process (sauf technique propriétaire)

CGI : la technique

- Le serveur Web, lorsqu'une URL de script CGI est demandée, passe la main au programme adéquat qui exécute le script.
- Ce programme génère la partie dynamique en HTML.
- La page HTML est complétée par le serveur Web et retournée au client Web.

Servlets

- La technique des CGI en Java
- **MAIS**
- Sans créer de processus + toute la puissance de Java (accès aux divers domaines de l'informatique : BD, multimédia, réseau, objets distribués, composants, etc.)
- + indépendance de la plate-forme et du serveur

Comment ça marche ?

- Le serveur (Web) possède désormais un interpréteur Java
- => il n'y a pas de processus créé lors de l'exécution de code Java
- Cf. les clients Web possèdent un interpréteur Java permettant de lancer des applets.
- D'où le nom de servlets.

Moteurs de servlets

- Pour exécuter des servlets, il faut un moteur de servlets dans le serveur Web.
- Ces moteurs sont des plug-in pour des serveurs Web existants
- ou bien des serveurs Web aux mêmes
- Plug-in : deux candidats : JRun
(www.allaire.com), tomcat
(jakarta.apache.org)

Serveurs Web et servlets

- Il existe des serveurs Web qui traitent les servlets (et JSP) :
 - IBM WebSphere
 - iPlanet Enterprise 4.x (ex Netscape)
- Voir à `java.sun.com/products/servlet`

Tomcat

- Développé par la communauté qui implémente les spécifs servlets.
- Téléchargeable (en version d'utilisation élémentaire) gratuitement à <http://jakarta.apache.org/downloads/bin/index.html>
- Plug-in de Apache v1.3 ou plus, Microsoft IIS v4.0 ou plus, ...
- Est aussi un mini-serveur Web.

Programmation des servlets

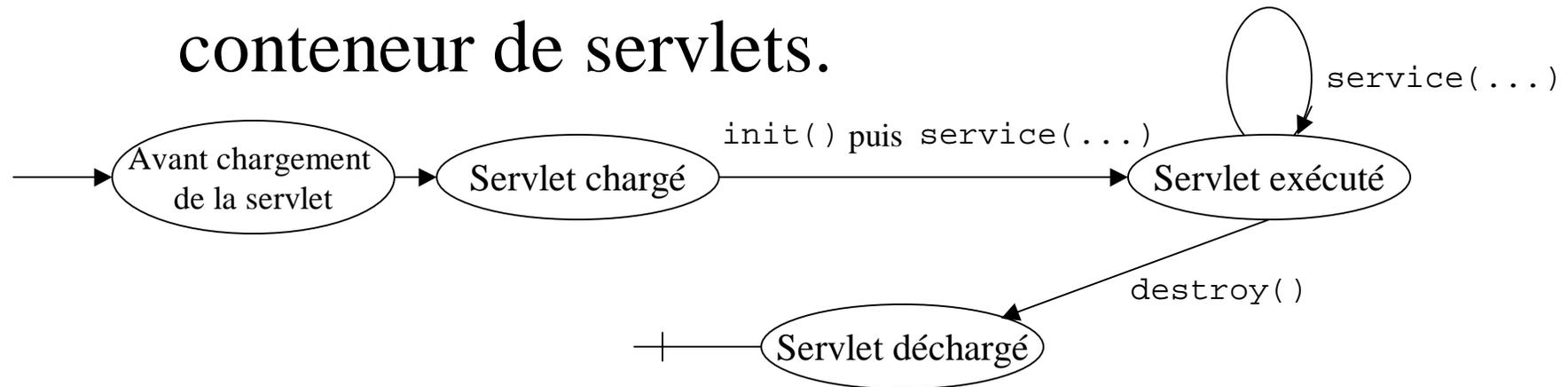
- Utilise deux paquetages :
 - `javax.servlet` : paquetage générique
 - `javax.servlet.http` : paquetage pour serveurs Web
- Ces paquetages **ne** sont **pas** dans J2SE 1.4
- Sont des paquetages supplémentaires.
- Il sont intégrés dans J2EE voir à <http://java.sun.com/j2ee/>

Documentation et tutorial

- Documentation à
`http://java.sun.com/j2ee/j2sdk-ee/techdocs/api/index.html`
- Tutorial à :
 - `http://java.sun.com/docs/books/tutorial/servlets/index.html`

Les états d'une servlet

- Cf. les états d'une applet. Le passage d'un état à un autre est automatique fait par le conteneur de servlets.



- Chargement = chargement en mémoire du moteur de servlets.
- Le conteneur de servlets lance la méthode `service(...)` à chaque requête.

Méthodes fondamentales d'une servlet

- Les trois méthodes
 - `public void init()`,
 - `public void service(...)`,
 - `public void destroy()`
 - sont définies dans la classe abstraite de base `javax.servlet.GenericServlet`.
- `service(...)` contient deux paramètres qui modélisent la requête et la réponse

Servlet pour serveur Web

- On a de nouveau les trois méthodes
 - `public void init()`,
 - `public void service(...)`,
 - `public void destroy()`
- spécialisées dans la classe abstraite de base `javax.servlet.http.HttpServlet` qui sont lancées automatiquement.

Construire une servlet pour serveur Web

- Cf. construction des applets
- On construit une classe qui hérite de la classe `javax.servlet.http.HttpServlet`
- On spécialise les méthodes qui nous intéressent
- On dépose cette classe « au bon endroit » du serveur Web

`service(...)` dans `HttpServlet`

- Contient deux paramètres :
`protected void
service(HttpServletRequest req,
HttpServletResponse resp) throws
ServletException, java.io.IOException`
- Cette méthode écrite dans `HttpServlet` redirige la requête suivant son type http.
- Méthode HTTP GET => `doGet(...)`
- Méthode HTTP POST => `doPost(...)`

doGet (...), doPost (...)

- **Ont les mêmes paramètres que `service(...)` de `HttpServlet` :**

```
public void doXXX(HttpServletRequest request, HttpServletResponse response) throws ServletException, java.io.IOException
```
- **En général on spécialise l'une des deux méthodes et la seconde appelle la première.**

Récupérer les données de la requête

- La requête envoie ses données par http
- Souvent des champs d'un formulaire
- On récupère ces données par
`String getParameter(String nomComposantFormulaire)`
- sur `request`
- Cf. les applets

Envoyer une réponse

- On positionne le type MIME par :
`void setContentType(String
type)`
- On récupère le canal de sortie texte par :
`PrintWriter getWriter()`
- sur `response`

Une servlet : code complet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MaPremiereServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // Etape 1. Spécifier le type MIME du contenu de la réponse
        response.setContentType("text/html");

        // Etape 2. Récupère le PrintWriter pour envoyer des données au client
        PrintWriter out = response.getWriter();

        // Step 3. Envoyer l'information au client
        out.println("<html>");
        out.println("<head><title>Bonjour</title></head>");
        out.println("<body>");
        out.println("<h1> Bonjour à tous </h1>");
        out.println("Il est : " + new java.util.Date());
        out.println("</body></html>");
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}
```

Démonstration

Bibliographie

- **Tutorial servlets :**
`http://java.sun.com/docs/books/tutorial/servlets/index.html`
- **Page de départ de la technologie servlets :**
`http://java.sun.com/products/servlet/index.html`