

De l'esprit à la machine
L'approche Professo-Académique


Frameworks J2EE

Abdelahad SATOUR

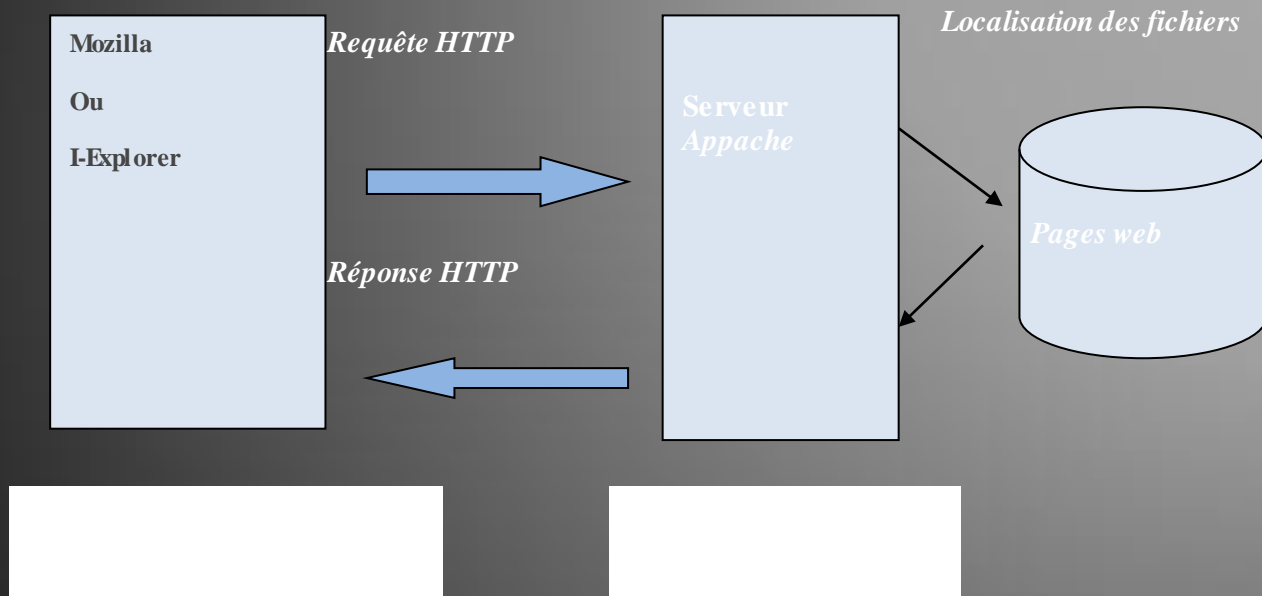
Séances 5–8

Un pas vers j2EE
Servlet

1. HTTP

- ▶ Le protocole HTTP (HyperText Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990.
 - ▶ Destiné à transférer des données sur Internet (en particulier des pages Web écrites en HTML).
 - ▶ Quelle est la différence avec FTP?
 - ▶ Les données sont localisées grâce à une chaîne de caractères appelée *URL* (Uniform Resource Locator) entre un navigateur (le client) et un serveur Web.
- 

1. HTTP



La communication entre le navigateur et le serveur se fait en deux temps :

- Le navigateur effectue une **requête HTTP**
- Le serveur traite la requête puis envoie une **réponse HTTP**

1.HTTP

- ▶ Une requête HTTP est destinée à une adresse de la forme (URL)

www. Machine. Domaine

- ▶ Exemple: **www.google.com**
- ▶ Cette adresse doit être spécifiée dans la barre adresse du navigateur

1.HTTP

- ▶ Voir plus sur HTTP.PDF

Commandes

Commande	Description
GET	Requête de la ressource située à l'URL spécifiée
HEAD	Requête de l'en-tête de la ressource située à l'URL spécifiée
POST	Envoi de données au programme situé à l'URL spécifiée
PUT	Envoi de données à l'URL spécifiée
DELETE	Suppression de la ressource située à l'URL spécifiée

En-têtes

Nom de l'en-tête	Description
Accept	Type de contenu accepté par le browser (par exemple <i>text/html</i>). Voir types MIME
Accept-Charset	Jeu de caractères attendu par le browser
Accept-Encoding	Codage de données accepté par le browser
Accept-Language	Langage attendu par le browser (anglais par défaut)
Authorization	Identification du browser auprès du serveur
Content-Encoding	Type de codage du corps de la requête
Content-Language	Type de langage du corps de la requête
Content-Length	Longueur du corps de la requête

Content-Type	Type de contenu du corps de la requête (par exemple <i>text/html</i>). Voir types MIME
Date	Date de début de transfert des données
Forwarded	Utilisé par les machines intermédiaires entre le browser et le serveur
From	Permet de spécifier l'adresse e-mail du client
From	Permet de spécifier que le document doit être envoyé s'il a été modifié depuis une certaine date
Link	Relation entre deux URL
Orig-URL	URL d'origine de la requête
Referer	URL du lien à partir duquel la requête a été effectuée
User-Agent	Chaîne donnant des informations sur le client, comme le nom et la version du navigateur, du système d'exploitation

En-têtes de réponse

Nom de l'en-tête	Description
Content-Encoding	Type de codage du corps de la réponse
Content-Language	Type de langage du corps de la réponse
Content-Length	Longueur du corps de la réponse
Content-Type	Type de contenu du corps de la réponse (par exemple <i>text/html</i>). Voir types MIME
Date	Date de début de transfert des données
Expires	Date limite de consommation des données
Forwarded	Utilisé par les machines intermédiaires entre le browser et le serveur
Location	Redirection vers une nouvelle URL associée au document
Server	Caractéristiques du serveur ayant envoyé la réponse

Les codes de réponse

Ce sont les codes que vous voyez lorsque le navigateur n'arrive pas à vous fournir la page demandée. Le code de réponse est constitué de trois chiffres : le premier indique la classe de statut et les suivants la nature exacte de l'erreur.

Code	Message	Description
10x	Message d'information	Ces codes ne sont pas utilisés dans la version 1.0 du protocole
20x	Réussite	Ces codes indiquent le bon déroulement de la transaction
200	OK	La requête a été accomplie correctement
201	CREATED	Elle suit une commande POST , elle indique la réussite, le corps du reste du document est sensé indiquer l' URL à laquelle le document nouvellement créé devrait se trouver.
202	ACCEPTED	La requête a été acceptée, mais la procédure qui suit n'a pas été accomplie
203	PARTIAL INFORMATION	Lorsque ce code est reçu en réponse à une commande GET , cela indique que la réponse n'est pas complète.
204	NO RESPONSE	Le serveur a reçu la requête mais il n'y a pas d'information à renvoyer
205	RESET CONTENT	Le serveur indique au navigateur de supprimer le contenu des champs d'un formulaire

206	PARTIAL CONTENT	Il s'agit d'une réponse à une requête comportant l'en-tête <i>range</i> . Le serveur doit indiquer l'en-tête <i>content-Range</i>
30x	Redirection	Ces codes indiquent que la ressource n'est plus à l'emplacement indiqué
301	MOVED	Les données demandées ont été transférées à une nouvelle adresse
302	FOUND	Les données demandées sont à une nouvelle URL, mais ont cependant peut-être été déplacées depuis...
303	METHOD	Cela implique que le client doit essayer une nouvelle adresse, en essayant de préférence une autre méthode que GET
304	NOT MODIFIED	Si le client a effectué une commande GET conditionnelle (en demandant si le document a été modifié depuis la dernière fois) et que le document n'a pas été modifié il renvoie ce code.
40x	Erreur due au client	Ces codes indiquent que la requête est incorrecte
400	BAD REQUEST	La syntaxe de la requête est mal formulée ou est impossible à satisfaire
401	UNAUTHORIZED	Le paramètre du message donne les spécifications des formes d'autorisation acceptables. Le client doit reformuler sa requête avec les bonnes données d'autorisation
402	PAYMENT REQUIRED	Le client doit reformuler sa demande avec les bonnes données de paiement
403	FORBIDDEN	L'accès à la ressource est tout simplement interdit
404	NOT FOUND	Classique! Le serveur n'a rien trouvé à l'adresse spécifiée. Parti sans laisser d'adresse... :)

400	BAD REQUEST	La syntaxe de la requête est mal formulée ou est impossible à satisfaire
401	UNAUTHORIZED	Le paramètre du message donne les spécifications des formes d'autorisation acceptables. Le client doit reformuler sa requête avec les bonnes données d'autorisation
402	PAYMENT REQUIRED	Le client doit reformuler sa demande avec les bonnes données de paiement
403	FORBIDDEN	L'accès à la ressource est tout simplement interdit
404	NOT FOUND	Classique! Le serveur n'a rien trouvé à l'adresse spécifiée. Parti sans laisser d'adresse... :)
50x	Erreur due au serveur	Ces codes indiquent qu'il y a eu une erreur interne du serveur
500	INTERNAL ERROR	Le serveur a rencontré une condition inattendue qui l'a empêché de donner suite à la demande (comme quoi il leur en arrive des trucs aux serveurs...)
501	NOT IMPLEMENTED	Le serveur ne supporte pas le service demandé (on ne peut pas tout savoir faire...)
502	BAD GATEWAY	Le serveur a reçu une réponse invalide de la part du serveur auquel il essayait d'accéder en agissant comme une passerelle ou un proxy
503	SERVICE UNAVAILABLE	Le serveur ne peut pas vous répondre à l'instant présent, car le trafic est trop dense (toutes les lignes de votre correspondant sont occupées veuillez rappeler ultérieurement)
504	GATEWAY TIMEOUT	La réponse du serveur a été trop longue vis-à-vis du temps pendant lequel la passerelle était préparée à l'attendre (le temps qui vous était imparti est maintenant écoulé...)

2.HTTP Java

- ▶ TP

2.HTTP Java

- ▶ <http://docs.oracle.com/javase/7/docs/api/java/net/URLConnection.html>

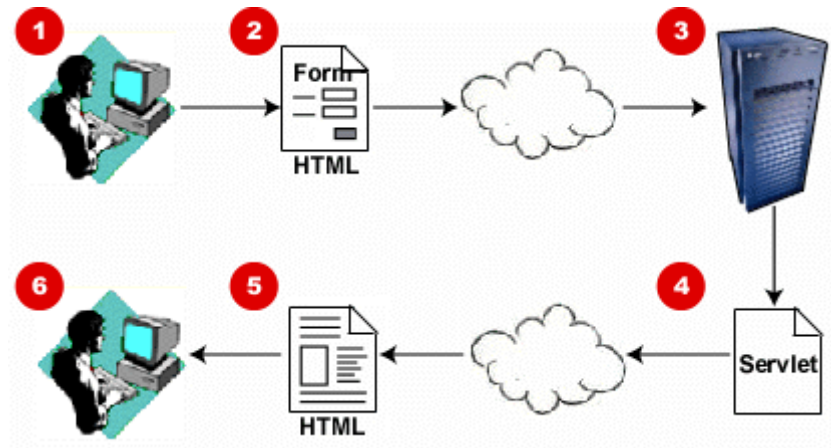
2.HTTP Java

- ▶ HTTP Query Discussion

3. C'est quoi un application web?

Cycle de vie d'une application web

- ▶ Une application web est constituée de 2 niveaux, le client émettant une requête et le serveur traitant cette requête.



3. C'est quoi un application web?

Cycle de vie d'une application web

- ▶ Un serveur web soumet une forme d'un document à compléter (une requête),
- ▶ Le client complète la forme et la retourne au serveur,
- ▶ Le serveur la réceptionne et la redirige vers la servlet invoquée dans la page html. En exécutant cette requête, le programme peut communiquer avec une ou plusieurs bases de données si c'est nécessaire. Ainsi donc, une servlet analyse les données soumises, peut éventuellement sauvegarder celles-ci ou collecter d'autres pour construire dynamiquement une page HTML.
- ▶ ces programmes retournent les résultats au serveur qu'il se charge de les transmettre au client.

3. C'est quoi un application web?

Cycle de vie d'une application web

► Le client peut utiliser :

- un formulaire HTML, permettant de saisir les champs ou bien il peut y avoir validation via des scripts (javaScript) ;
- des applets et communiquer avec le serveur via des sockets/RMI (IFT3880).
- Les requêtes http vers le serveur contiennent :
 - l'url de la ressource à accéder,
 - la requête GET pour extraire des informations sur le serveur,
 - la requête POST pour modifier les données sur le serveur.

3. C'est quoi un application web?

Cycle de vie d'une application web

- ▶ Le serveur identifie avec la requête le type d'environnement d'exploitation à charger en fonction de :
 - l'extension du fichier (.cgi, .jsp etc.) ou
 - le répertoire où il se trouve (cgi-bin/, servlet/ etc.).

3. C'est quoi un application web?

Cycle de vie d'une application web

- ▶ Le serveur charge par la suite l'environnement d'exécution Perl (cgi-perl), JVM (servlets) etc.
- ▶ Le script ou programme précise le type du contenu (HTML, images etc.) et intègre la réponse dans un flot de sortie.
- ▶ Le navigateur définit le type MIME (l'encodage utilisé pour le transfert de documents multimédias à travers le réseau) texte/html audio/basic image/gif etc. et affiche les données en fonction.

4. Servlet

Définition d'une Servlet

- ▶ Servlet : Server-side applet
- ▶ Les servlets sont aux serveurs ce que sont les applets aux browsers mais sans interface graphique utilisateur ...
- ▶ Elles sont limitées à la puissance du langage HTML ... par contre, elles ne sont pas astreintes aux mêmes règles de sécurité que les *applets*
- ▶ Les servlets correspondent à des programmes Java normaux qui utilisent des modules supplémentaires (ainsi que les classes et les méthodes associées) figurant dans l'API des servlets Java.
- ▶ Les servlets s'exécutent sur une machine de serveur Web à l'intérieur d'un serveur compatible avec Java. Elles permettent l'extension des fonctions du serveur.
- ▶

4. Servlet

Définition d'une Servlet

- ▶ Une servlet peut être chargée :
 - ▶ automatiquement lors du démarrage du serveur Web,
 - ▶ lorsque le premier client demande les services de la servlet.
- ▶ Une fois chargées, les servlets restent actives dans l'attente d'autres requêtes du client.

4. Servlet

Définition d'une Servlet

- ▶ Les servlets permettent l'extension des fonctions du serveur grâce à la création d'un environnement de prestation de services de requête/réponse via le Web :
 - Un client envoie une requête au serveur. Ce dernier transmet au servlet les informations relatives à la requête. Le servlet crée ensuite une réponse que le serveur renvoie au client.

4. Servlet

Définition d'une Servlet

- ▶ Dans la mesure où il s'agit d'un programme Java :
 - La servlet peut utiliser toutes les fonctions du langage Java lors de la création de la réponse.
 - Elle peut également communiquer avec des ressources externes tels que des fichiers ou des bases de données, ou avec d'autres applications (également écrites en langage Java ou dans d'autres langages), afin de créer la réponse et éventuellement de sauvegarder des informations relatives à l'interaction requête/réponse.

4. Servlet

Définition d'une Servlet

- ▶ La réponse envoyée au client peut donc être une réponse dynamique et unique conçue pour une interaction particulière et non une page HTML statique existante.

4. Servlet

A quoi servent les servlets?

- ▶ Les servlets exécutent un grand nombre de fonctions, par exemple:
 - – Une servlet peut créer et renvoyer une page Web HTML complète dont le contenu dynamique dépend de la nature de la requête du client.
 - – Une servlet peut simplement créer une partie d'une page Web HTML qui est intégrée à une page HTML statique existante.
 - – Une servlet peut communiquer avec d'autres ressources du serveur, y compris des bases de données, d'autres applications Java et des applications écrites dans d'autres langages.
 - – Une servlet peut traiter les connexions avec plusieurs clients en acceptant les données en entrée de plusieurs clients et en diffusant à ces derniers des résultats. Une servlet peut, par exemple, correspondre à un serveur de jeux électroniques faisant intervenir plusieurs joueurs.

4. Servlet

Quels sont les avantages et les inconvénients des servlets?

► Avantages

- Les servlets sont indépendantes des OS (Unix ou NT) et des serveurs Web (Apache, IIS etc.) ;
- Peuvent produire de l'HTML côté client (notamment pour la consultation de la base), sur la base d'http;
- Peuvent dialoguer avec des applets Java côté client avec un protocole à objets distribués de type RMI;
- S'appuient sur un langage vraiment standard : Java (et non pas Java script ou Visual Basic);
- Par rapport aux applets, le client est « allégé » ;
- Par rapport aux CGI, les servlets prennent en charge les connexions des utilisateurs en multi-thread, qui n'est pas le cas des CGI (même avec FastCGI).

4. Servlet

Quels sont les avantages et les inconvénients des servlets?

▶ Inconvénients

- Par rapport aux applets, interface graphique utilisateur limitée à HTML
- Par rapport au CGI, voir plus loin par la suite...

4. Servlet

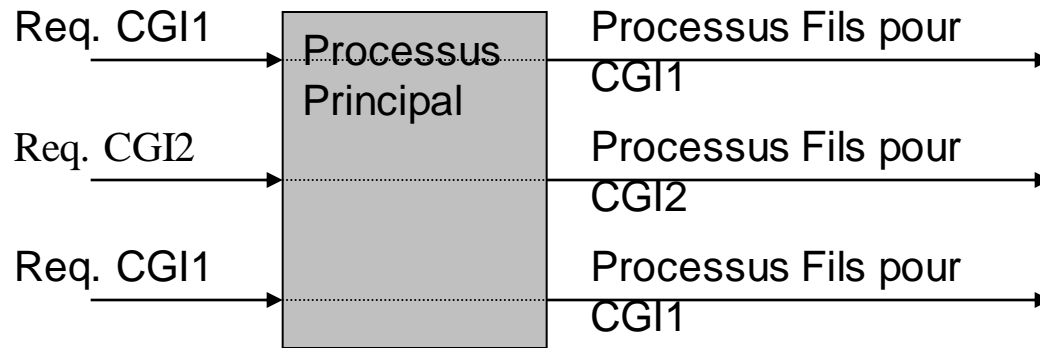
C'est quoi les différences entre les scripts CGI et les servlets?

- ▶ La plus grosse différence entre les CGI et les servlets est la performance.
- ▶ Il n'y a qu'une seule machine virtuelle Java qui tourne sur le serveur.
- ▶ La servlet est placée en mémoire une fois qu'elle est appelée. Elle n'est pas remise en mémoire jusqu'à ce que la servlet change.
- ▶ Une servlet dont le code a été modifié peut être réactivée (c'est à dire remplacée en mémoire) sans redémarrer le serveur ou l'application.

4. Servlet

C'est quoi les différences entre les scripts CGI et les servlets?

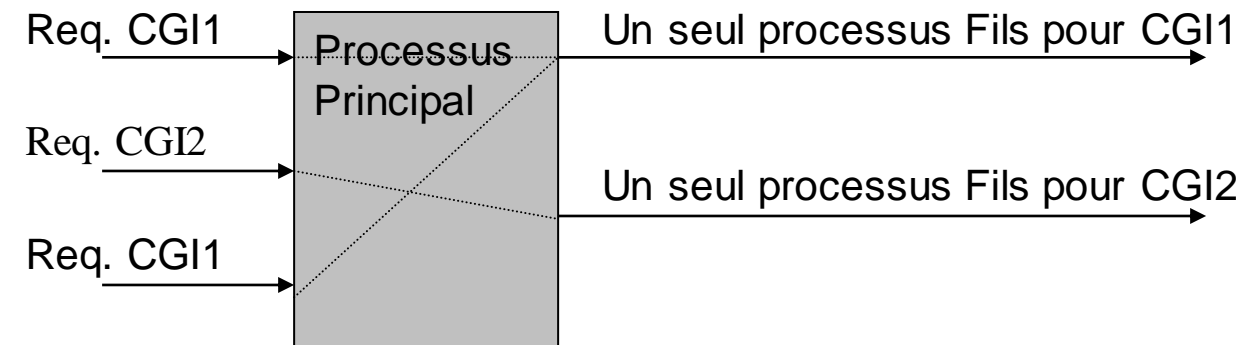
Common Gateway Interface (CGI)



4. Servlet

C'est quoi les différences entre les scripts CGI et les servlets?

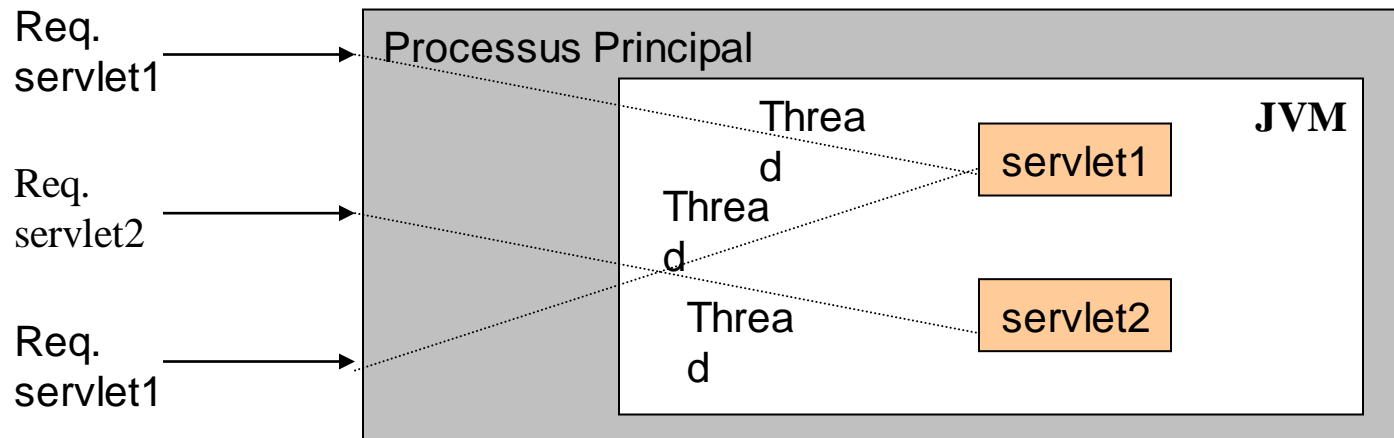
FastCGI



4. Servlet


C'est quoi les différences entre les scripts CGI et les servlets?

Les servlets



4. Servlet

C'est quoi les différences entre les scripts CGI et les servlets?

- ▶ Les servlets résident en mémoire, de ce fait leur exécution est très rapide. L'information statique peut être partagée par plusieurs invocations de la servlet, vous autorisant donc de partager cette information entre plusieurs utilisateurs.
 - ▶ Les servlets sont modulaires, chaque servlet peut accomplir une tâche spécifique et ainsi vous pouvez les rendre communicantes.
 - ▶ Moins il y a de calcul à faire côté client, plus l'approche *servlet* est intéressante.
- 


5. Web Static HTTP Java

- ▶ Web Static Jetty TP
- ▶ Embeded Jetty (sur le web)

5. Web Dynamic


- ▶ Premier pas vers JOnAs
- ▶ <http://localhost:9000/>
- ▶ <http://localhost:9000/jonasAdmin/>
- ▶ Architecture dossier d'un serveur web ou un serveur applicatif

6. Conteneurs de Servlet

- ▶ Tomcat 4.1
 - ▶ IronFlare Orion 1.5
 - ▶ Jetty 4.1
 - ▶ Caucho Resin 2.1
 - ▶ Sun ONE 7.0
 - ▶ IBM WebSphere 4.0
 - ▶ BEA WebLogic 7.0
- 

6. Conteneurs de Servlet

Tomcat

- ▶ Tomcat 4.1 (Catalina)
 - ▶ Projet Apache (Apache \neq Apache Httpd)
 - ▶ Open source
 - ▶ Implantation de référence de la spécification
 - ▶ Spécification Servlet 2.3 et JSP 1.2
(bientôt Servlet 2.4 et JSP 2.0)
- 

6. Conteneurs de Servlet

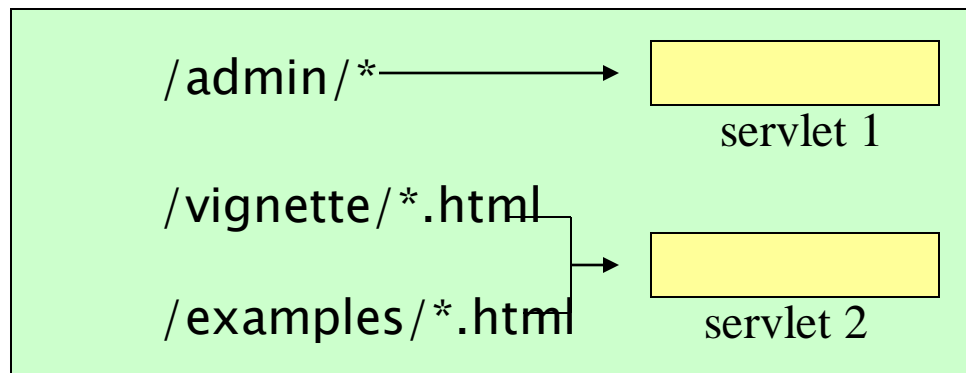
Servlet & Conteneur

Le conteneur de servlets associe à des URLs virtuels une servlet

Browser HTTP



Requête
Réponse

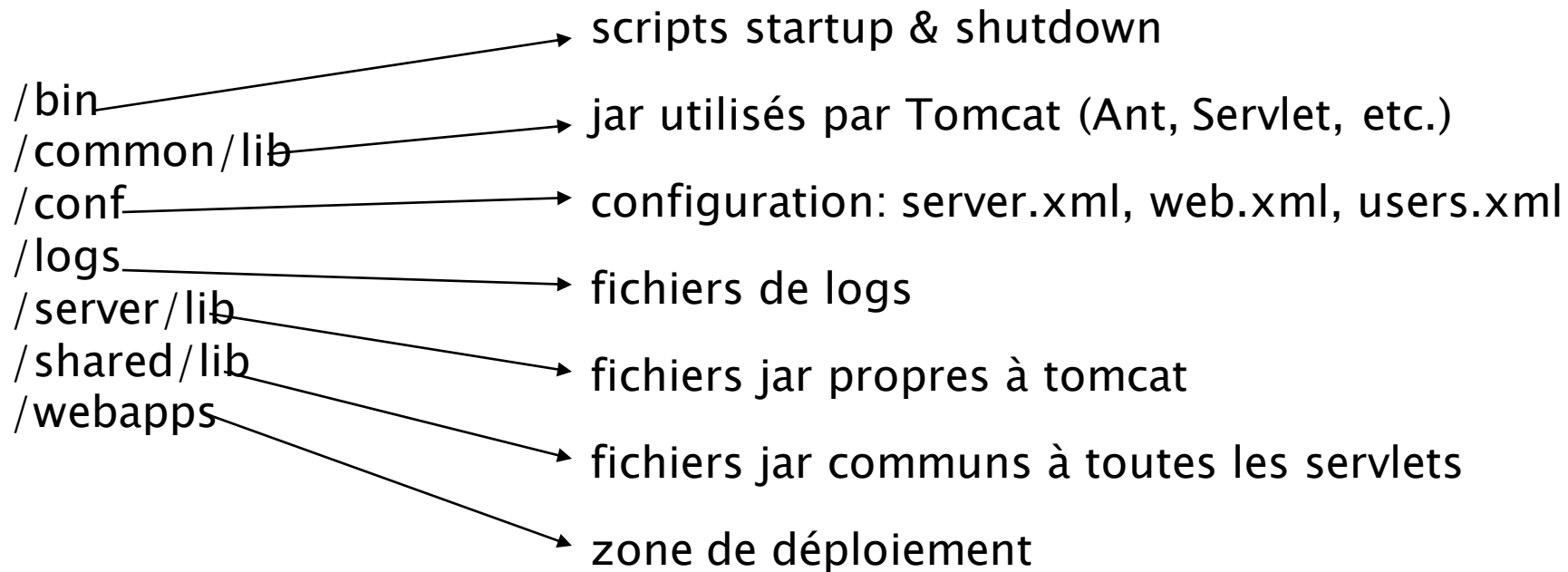


Conteneur de Servlets

6. Conteneurs de Servlet

Répertoire de Tomcat

Organisation des répertoires de Tomcat



6. Conteneurs de Servlet

Configuration de Tomcat

Le fichier server.xml

- **Server**
Racine, spécifie le port de shutdown.
- **Service**
Associe des connecteurs à un engine.
- **Connector**
Ils correspondent à un point d'accès à un service, soit via un serveur soit en connexion directe.
- **Engine**
correspond au conteneur de servlet en lui-même.
- **Logger**
Ils effectuent la journalisation.
- **Host**
Déclare où sont stockées les servlets pour un nom de machine.
- **Context**
Chaque Context représente la configuration associée à un chemin dans la hiérarchie

6. Conteneurs de Servlet

Configuration de Tomcat

port d'écoute

Un **Connector** point d'accès utilisable par un client :

- port="8080"
- minProcessors="5" nombre de threads minimum
- maxProcessors="75" nombre de threads maximum
- enableLookups="true" DNS inverse
- acceptCount="100" nombre de connections pendante
- connectionTimeout="20000"

Le conteneur **Engine**

- name="Standalone"
- defaultHost="localhost" Nom de l'host si pas HTTP 1.1

6. Conteneurs de Servlet

Configuration de Tomcat

Le **Logger** effectue la journalisation des requêtes

- prefix="catalina_log. "
- suffix=".txt"
- timestamp="true"

Le tag **Host** définit les paramètres pour un host virtuel

- name="localhost"
- appBase="webapps"
- unpackWARs="true"
- autoDeploy="true"

6. Conteneurs de Servlet

Configuration de Tomcat

Gestion des associations entre un URI et un chemin sur le disque

Un **Context** représente l'association entre un chemin sur le serveur (URI) et un chemin sur le disque

- `path= "/examples"`
- `docBase="examples"`
- `reloadable="true"`

URI d'accès

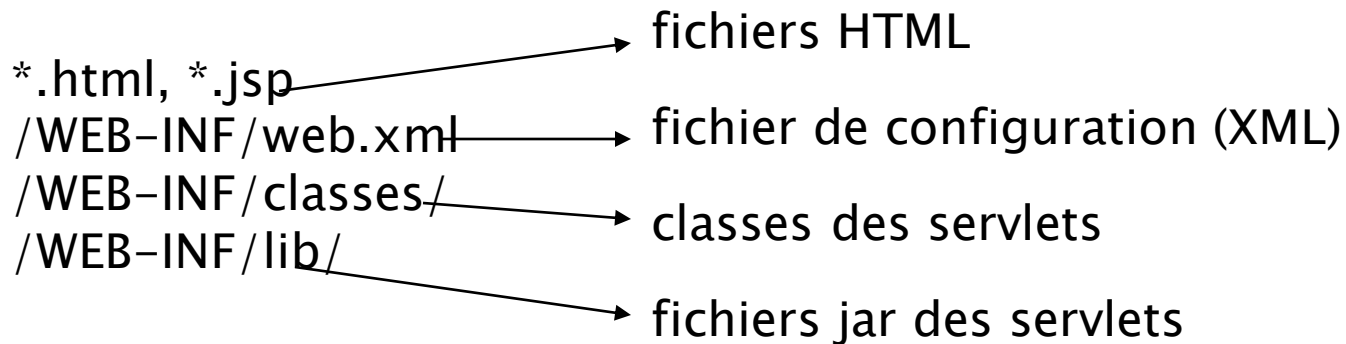
Chemin d'accès des fichiers
(relatif ou absolu par rapport à webapps)

Détection automatique des changements
et rechargement si besoin

6. Conteneurs de Servlet

Configuration de Tomcat

Une application Web possède **dans un repertoire** lui-même dans *webapps* une architecture spécifique



L'ensemble des fichiers et répertoire peut être mis dans un war (Web Archive) grâce à la commande `jar`. Le war est automatiquement *dé-jarré* s'il est placé dans le répertoire *webapps*.

7. Generic vs HTTP Servlet

Il existe deux types de servlets

- ▶ Les **GenericServlet** qui ne pré-suppose pas d'un protocole
- ▶ Les **HttpServlet** qui répondent à des clients par le protocole HTTP

GenericServlet est une classe du package `javax.servlet` tandis que
HttpServlet est une classe du package `javax.servlet.http`

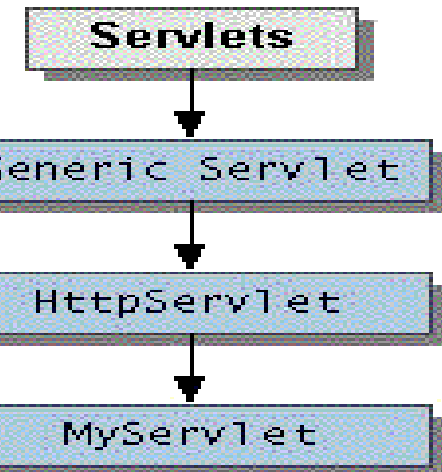
7. Generic vs HTTP Servlet

- ▶ `java.lang.Object`

- ▶ `javax.servlet.GenericServlet`


|

- ▶ `javax.servlet.http.HttpServlet`



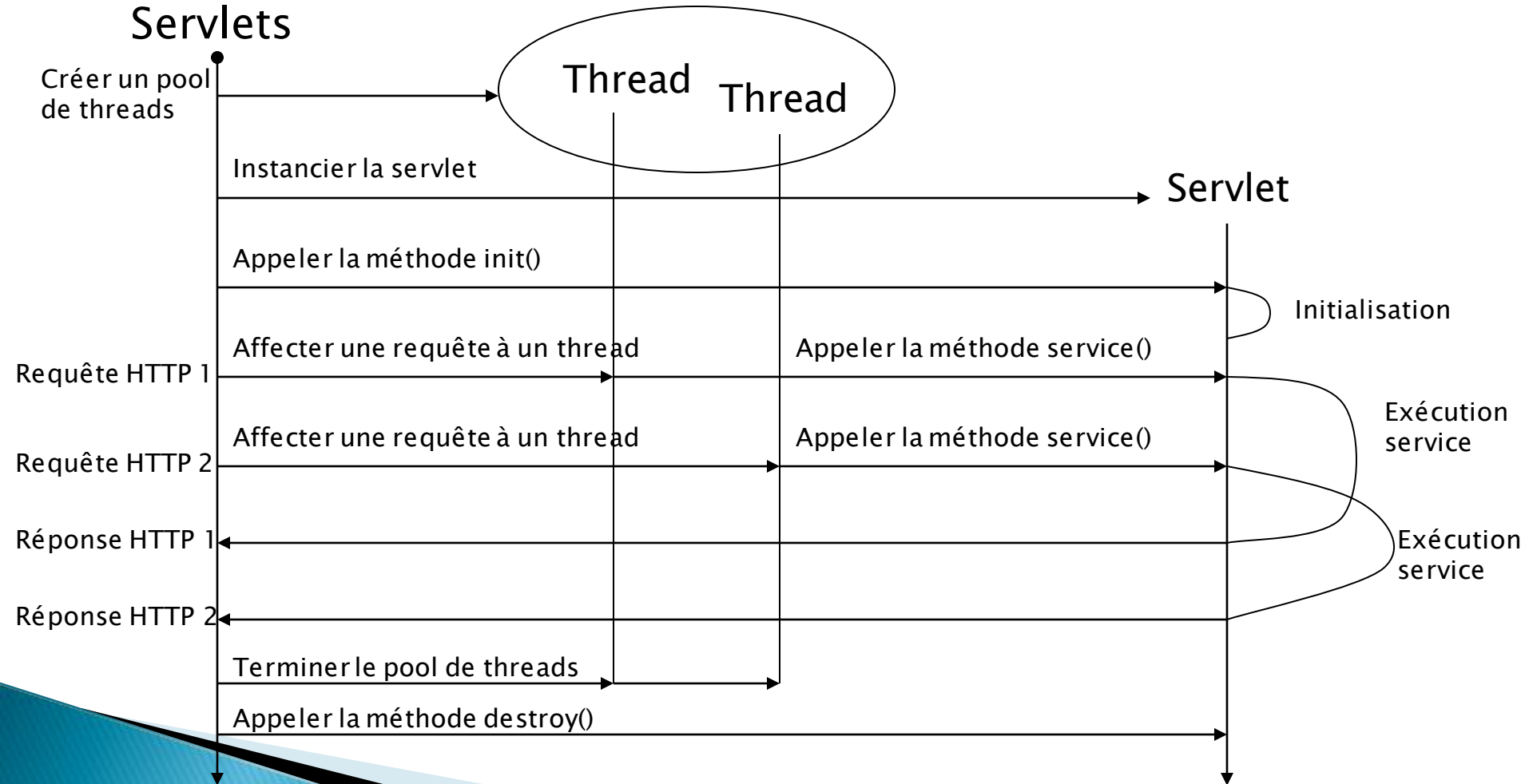
7. Cycle de vie d'une servlet

Le cycle de vie d'une servlet :

1. la méthode **init()** est appelée après le chargement ;
 2. une méthode **service()** est appelée à chaque requête dans **une nouvelle thread**.
 3. la méthode **destroy()** est appelée pour le déchargement.
- 

7. Cycle de vie d'une servlet

Moteur de Servlets



7. TP Servlet Generic



7. L'interface Request

L'interface **ServletRequest** permet de récupérer les paramètres de la requête :

```
public abstract int getContentLength()
public abstract String getContentType()
public abstract String getProtocol()
public abstract String getScheme()
public abstract String getServerName()
public abstract int getServerPort()
public abstract String getRemoteAddr()
public abstract String getRemoteHost()
public abstract ServletInputStream getInputStream() throws IOException
public abstract String getParameter(String name)
public abstract String[] getParameterValues(String name)
public abstract Enumeration getParameterNames()
public abstract Object getAttribute(String name)
```

Description du serveur

Description du client

Il est possible de rajouter des attributs (non HTTP)

7. L'interface Response

L'interface **ServletResponse** permet de renvoyer une réponse :

public abstract void **setContentLength**(int length)

public abstract void **.setContentType**(String type)

public abstract ServletOutputStream **getOutputStream**() throws IOException

Taille de la réponse
(peut être omis)

Type de contenu au
format MIME

L'objet réponse permet d'obtenir
le flux de sortie en écriture

7. L'interface Response

L'interface **ServletResponse** permet de renvoyer une réponse :

public abstract void **setContentLength**(int length)

public abstract void **.setContentType**(String type)


public abstract ServletOutputStream **getOutputStream**() throws IOException

Taille de la réponse
(peut être omis)

Type de contenu au
format MIME

L'objet réponse permet d'obtenir
le flux de sortie en écriture

7. TP Servlet HTTP

- ▶ La classe qui hérite de `HttpServlet` doit redéfinir au moins l'une des méthodes suivantes:
 - ▶ `doGet`, pour les requêtes HTTP GET (récupérer des informations du serveur, un fichier html, une image etc.)
 - ▶ `doPost`, pour les requêtes HTTP POST (envoyer des informations au serveur, des données par exemple etc.)
 - ▶ `doPut`, pour les requêtes HTTP PUT (déposer un fichier sur un serveur)
 - ▶ `doDelete`, pour les requêtes HTTP DELETE (effacer un fichier du serveur)
 - ▶ `init` et `destroy`, (étapes appelées lors du chargement et resp. lors de la destruction de la servlet)
 - ▶ `getServletInfo` (des informations sur la servlet: auteur, version etc.)
- 

7. TP Servlet HTTP

Manipulation rapide(Admin)
Formulaire



7. TP Servlet HTTP

Redirection

Utilisation de

- ▶ `response.sendRedirect("/index.html");`
- ▶ `response.sendError(HttpServletResponse.SC_NOT_FOUND);`