

J2EE

A.-E. Ben Salem

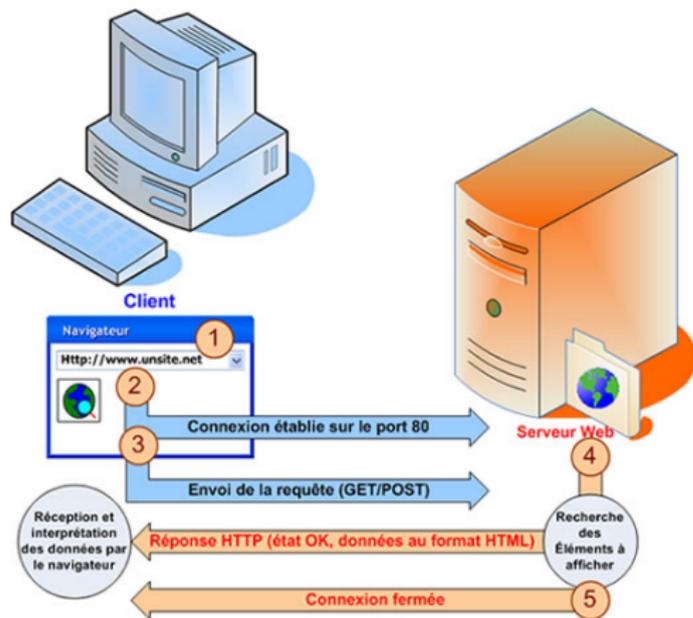
LRDE and LIP6

09 Octobre 2011

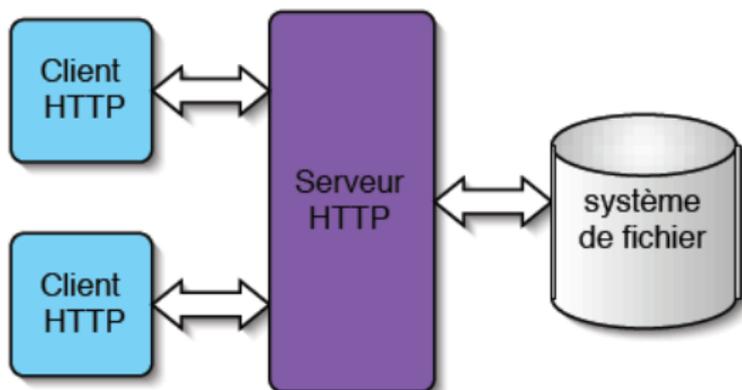
- ① J2EE
- ② Architecture Client/Serveur HTTP
- ③ Différence entre Web Statique et Web Dynamique
- ④ Web Dynamique avec un Serveur d'applications J2EE
- ⑤ Les pages JSP
- ⑥ Déploiement d'une application dans les Conteneurs J2EE

- ▶ J2EE est l'acronyme de Java 2 Entreprise Edition
- ▶ renommée Java EE, depuis sa version 5.
- ▶ basée sur J2SE : les API de base de Java
- ▶ c'est une plate-forme pour le développement et l'exécution d'applications distribuées :
 - ▶ Développement : à l'aide d'un ensemble de bibliothèques logicielle (API : Servlet 3.0, JSP 2.2, EJB 3.1, JDBC, ...).
 - ▶ Exécution : déploiement dans un serveur d'application (Tomcat, JBoss WebSphere, Weblogic, ...).
- ▶ orientée client/serveur

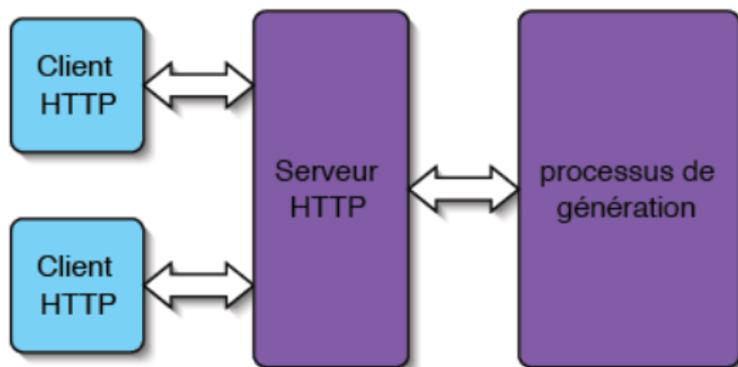
Architecture Client/Serveur HTTP



- ▶ Client : navigateur web
- ▶ Serveur : serveur HTTP
- ▶ Protocole de communication : HTTP
- ▶ Requête : URL pour désigner une ressource
- ▶ Port : 80 (8080 Tomcat)
- ▶ Réponse : page HTML

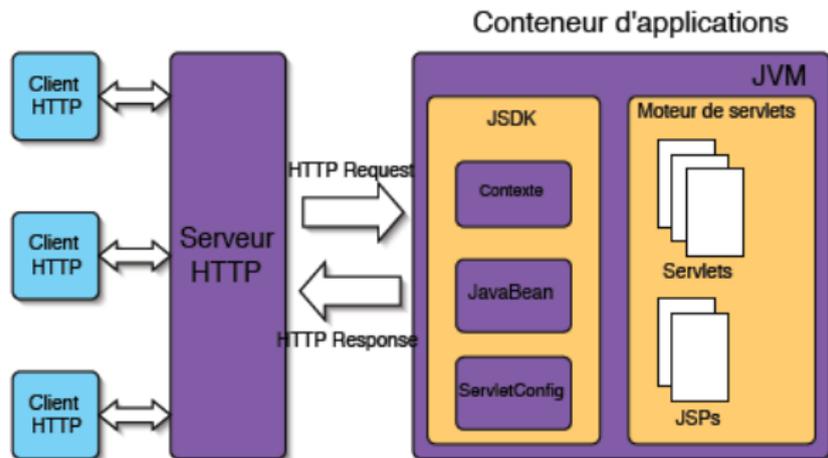


- ▶ Serveur HTTP statique (Apache, IIS, ...)
- ▶ Contenu statique : la ressource demandée est un fichier (page html statique, fichier txt, pdf, images, ...)
- ▶ un site est associé à un dossier dans le système de fichiers



- ▶ Contenu dynamique :
 - ▶ Page HTML généré dynamiquement,
 - ▶ éventuellement à partir d'informations provenant d'une BD.
- ▶ Java (J2EE), PHP, Perl, ASP, C# (.NET)

Web Dynamique avec un Serveur d'applications J2EE



- ▶ Serveur HTTP + Serveur d'applications (processus de génération)
- ▶ Serveur d'applications J2EE contient :
 - ▶ éventuellement un serveur HTTP interne (Tomcat embarque un serveur HTTP Apache)
 - ▶ Conteneur Web : exécution des Servlets et/ou JSPs permettant de générer dynamiquement des pages HTML
 - ▶ Conteneur EJB...

- ▶ Une page JSP est un fichier au format XHTML
- ▶ Une JSP mélange deux types de code :
 - ▶ Partie statique : code HTML
 - ▶ Partie dynamique : code Java exécuté par le serveur dont le résultat est injecté dans la page html transmise au client
- ▶ L'écriture d'une JSP est basée sur les balises suivantes :
 - ▶ tags HTML (<p>,
, <h1>...)
 - ▶ tags JSP : transformés en code Java par le Serveur (par exemple le tag "<c :forEach>" pour écrire une boucle)
 - ▶ scriptlets : code Java écrit dans un tag spécial "<%>"

- ▶ Du code Java : `<% code Java %>`
- ▶ Des évaluations d'expression : `<%= expression %>`, Par exemple : `<%= nom_de_variable %>` affiche le contenu de la variable
- ▶ Des variables prédéfinies

```
<%@ page language="Java" %>
<html><head><title>First.jsp</title>
</head><body>
<h1>Nombres de 1 à 10</h1>
<% for(int i=1; i<=10; i++) { %>
    <%= i %> <br/>
<% } %>
</body>
</html>
```

Variables prédéfinies

```
HttpServletRequest request
HttpServletResponse response
HttpSession session
ServletContext application
PrintWriter out
Object page
ServletConfig config
javax.servlet.jsp.PageContext pageContext
Throwable exception
```

Envoi de paramètres dans une Requête HTTP

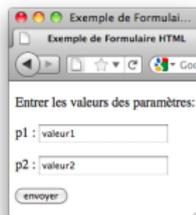
Deux méthodes pour ajouter des paramètres dans une Requête HTTP :

- ▶ Requêtes de type **GET** : paramètres inclus dans l'URL

```
http://serveur/contexte/ressource?param1=valeur1&param2=valeur2
```

- ▶ Requêtes de type **POST** : paramètres inclus dans les entêtes HTTP, nécessite l'utilisation d'un formulaire :

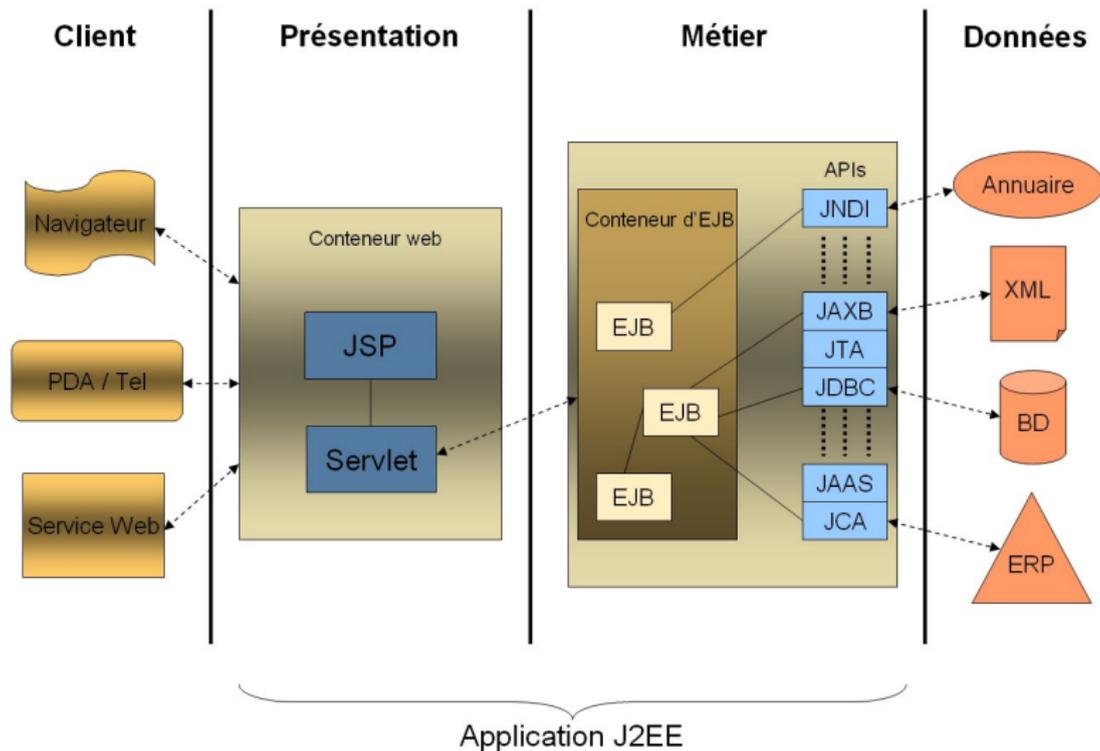
```
<FORM METHOD=POST ACTION="ressource">  
<p>Entrer les valeurs des paramètres :</p>  
<p>p1 : <INPUT TYPE=TEXT NAME="param1"></p>  
<p>p2 : <INPUT TYPE=TEXT NAME="param2"></p>  
<INPUT TYPE=SUBMIT VALUE="envoyer">  
</FORM>
```



- ▶ Reception des paramètres côté Serveur J2EE :

```
Code Java à ajouter à une JSP : request.getParameter("param1");
```

Architecture de J2EE



L'architecture d'une application se découpe en :

- ▶ Couche Client (dialogue avec l'utilisateur) :
 - ▶ Léger : navigateur Web (html-HTTP), ...
 - ▶ Lourd : Application java (Swing), Applet, ...
 - ▶ client Web Service (xml-HTTP)
- ▶ Couche Présentation (Conteneur WEB) :
 - ▶ Servlet
 - ▶ JSP (Struts, JSF, ...)
- ▶ Couche Métier (Conteneur EJB) :
 - ▶ POJO (Hibernate, Spring)
 - ▶ EJB
 - ▶ Web Service
- ▶ Couche Données :
 - ▶ Base de Données (Oracle, DB2, ...)
 - ▶ XML
 - ▶ ERP (Cobol, ...)

- ▶ Conteneur Web, composé de :
 - ▶ un moteur de Servlets pour exécuter les Servlets
 - ▶ un moteur de JSP pour exécuter les JSPs
 - ▶ un serveur HTTP (ou un "plug in" d'un serveur HTTP externe : Apache, IIS, ...)
- ▶ conteneur d'EJB : pour exécuter les EJB
- ▶ Un serveur d'application peut fournir :
 - ▶ uniquement un conteneur Web : Tomcat
 - ▶ uniquement un conteneur d'EJB : JBoss, Jonas, ...
 - ▶ les deux Web et EJB : Websphere, Weblogic, ...

Déploiement d'une application dans les Conteneurs

Pour déployer une application dans un conteneur, il faut lui fournir dans une archive deux éléments :

- ▶ les fichiers de l'application : Servlets, JSPs, classes compilées (.class), ressources statiques (pages HTML statiques, images, CSS, pdf, txt, ...), ...
- ▶ un fichier XML descripteur de déploiement : précise au conteneur la configuration (les options) pour exécuter l'application

Chaque conteneur possède son propre format d'archive :

Conteneur	Contenu de l'archive	Extension de l'archive	Descripteur de déploiement
Web	Servlets, JSP, ressourceswar	web.xml
EJB	EJB et leurs classes	.jar	ejb-jar.xml
les 2 : Web + EJB	.war et .jar	.ear	application.xml

Archives à déployer dans les Conteneurs

