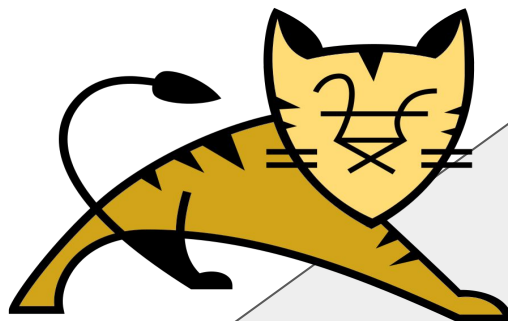




Programmation Java EE



JAKARTA EE

Description de la formation

- Séance 1: Introduction au Java EE
- Séance 2: Servlet (partie 1)
- Séance 3: Servlet (partie 2)
- Séance 4: JSTL et JDBC
- Séance 5: JPA (1er partie)
- Séance 6: JPA (2eme partie)
- Séance 7: Premier pas vers Spring
- Séance 8: Spring et Architecture SOA

Plan du cours

- La Spécification Java EE
 - Architecture Java EE
 - Composants
- Architecture WEB
 - Architecture n-tier
 - Serveur web et Protocol HTTP
 - Serveur d'Application Java
 - Conteneur de Servlet
- Java Bean



C'est quoi
Java EE ?

Java EE

- Java EE, ou JEE ou encore Java Enterprise Edition :
 - ◆ est un ensemble de spécifications destinées aux applications d'entreprise.
 - ◆ peut être vu comme une extension du langage Java afin de faciliter la création d'applications réparties, robustes, performantes et à haute disponibilité.
 - ◆ est basée sur des composants.

- Les API de Java EE peuvent se répartir en trois grandes catégories :
 - ◆ Les composants
 - ◆ Les services d'infrastructures
 - ◆ Les services de communication

Java EE (Composants)

→ Les composants web :

- ◆ Servlet : Elle permet de traiter les données envoyées par l'utilisateur et de choisir la Vue à retourner à celui-ci. On appelle cette partie : Contrôleur.
- ◆ JSP : Les JSP sont les pages servant à générer l'ensemble du code HTML de l'interface utilisateur. On l'appelle généralement : Vue.
- ◆ JSP et Servlets constituent les solutions techniques de base pour les applications Web en Java.

→ Les composants métiers : EJB

- ◆ EJB (Enterprise Java Bean) : Il est chargé du traitement des données propres à un secteur d'activité (on parle de logique métier ou de logique applicative) et de l'interfaçage avec les bases de données. On parle de la partie : Modèle.

Java EE (Services d'infrastructures)

- JDBC (Java DataBase Connectivity) est une API d'accès aux bases de données relationnelles.
- JNDI (Java Naming and Directory Interface) est une API d'accès aux services de nommage et aux annuaires d'entreprises tels que DNS, NIS, LDAP, etc.
- JTA/JTS (Java Transaction API/Java Transaction Services) est une API définissant des interfaces standard avec un gestionnaire de transactions.
- JCA (JEE Connector Architecture) est une API de connexion au système d'information de l'entreprise tels les ERP.
- JPA (Java Persistence API) fournit un langage de requête (également appelé JPQL), que vous pouvez utiliser pour manipuler des objets sans écrire de requêtes SQL spécifiques à la base de données avec laquelle vous travaillez.

Java EE (Services de communication)

- JAAS (Java Authentication and Authorization Service) est une API de gestion de l'authentification et des droits d'accès.
- JavaMail est une API permettant l'envoi de courrier électronique.
- JMS (Java Message Service) fournit des fonctionnalités de communication asynchrone (appelées MOM pour Middleware Object Message) entre applications.
- RMI-IIOP est une API permettant la communication synchrone entre objets.

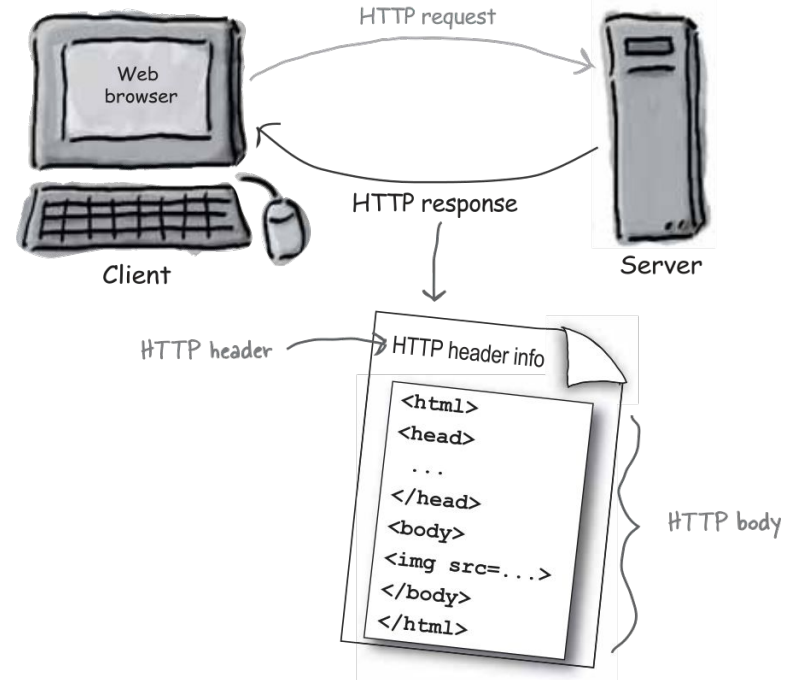
Architecture WEB

Java EE (Serveur WEB)

- Une machine qui héberge un logiciel capable de fournir du contenu web.
- Une machine peut héberger plusieurs logiciels serveur de toute nature (serveur de fichier, serveur web, serveur DNS etc...)
- Une machine peut être à la fois serveur et client.

Java EE (Protocole HTTP)

- HTTP (Hyper Text Transfer Protocol) est un protocole de communication client serveur développé pour le Web et fonctionne sous TCP/IP.
- Les serveurs Web communiquent bien évidemment sur ce protocole.



Java EE (Protocole HTTP)

- La demande de la ressource à un serveur HTTP se fait via un URL.
- URL (Uniform Resource Locator) : Procédé d'adressage permettant d'identifier sans ambiguïté une ressource sur le Web. Il prend la forme:

protocole://emplacement_serveur:port_ecoute/ressource

- Exemple:

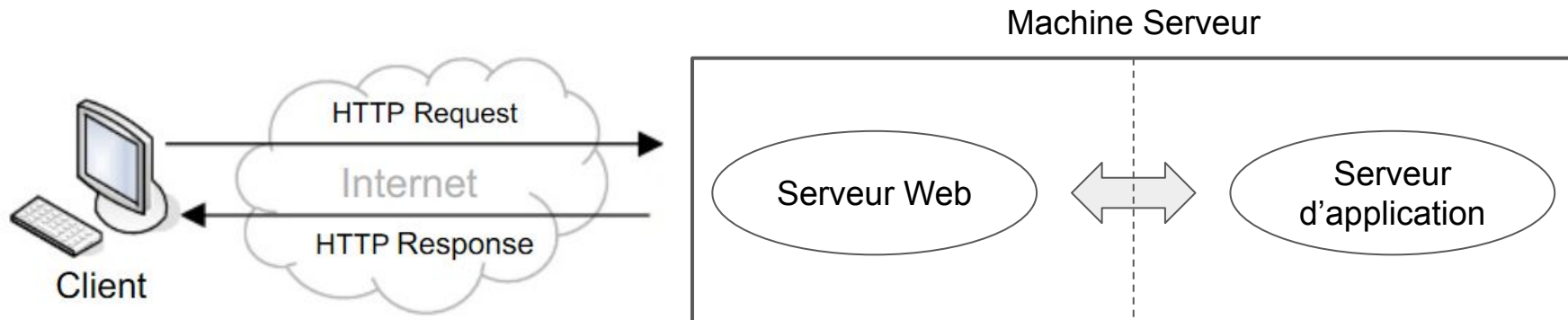
<http://173.194.67.138:80/index.html>

Java EE (Protocole HTTP)

- Un serveur Web est - par convention- ouvert sur le port 80.
- Exemple:
 - ◆ <http://192.198.0.1:80> équivalant a <http://192.198.0.1>
- L'emplacement du serveur est matérialisé par son adresse IP mais pour simplifier la vie de l'utilisateur, cette IP est généralement masquée derrière un nom de domaine attribué à cette adresse IP (via un serveur DNS).

Java EE (Serveur d'application)

- Un serveur d'application utilise un langage de programmation pour créer à la volée du contenu et dans le même temps effectuer des opérations.
- Ce dernier va en réalité interroger le serveur web qui va être configuré pour demander certains contenus à un serveur applicatif. Une transaction va donc s'effectuer entre le serveur web et le serveur applicatif.

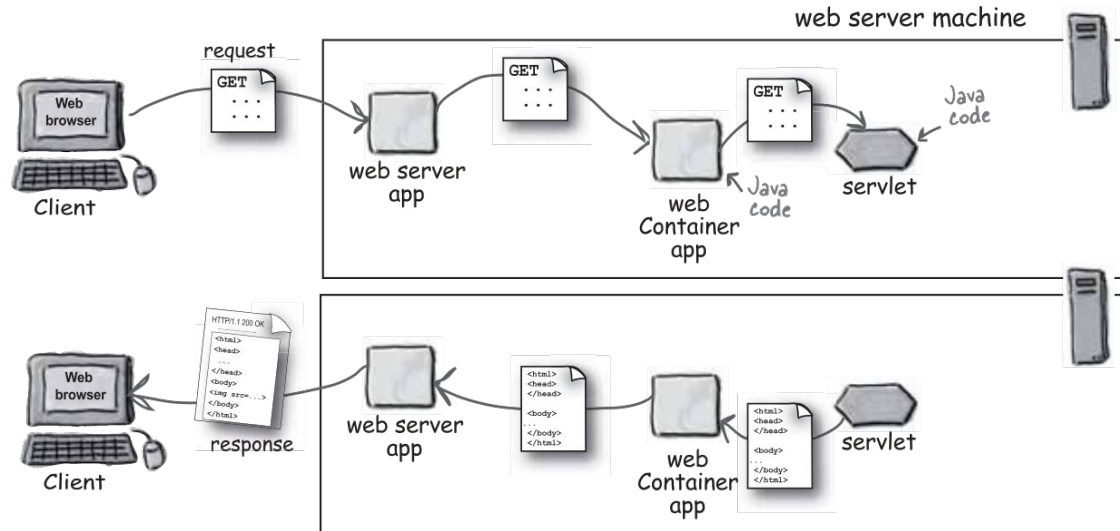


Java EE (Serveur d'application)

- Un serveur applicatif Java est un logiciel capable de servir des applications Java.
- on distingue deux catégories de serveur applicatif:
 - ◆ Serveur d'application Web Java (conteneur Web)
 - ◆ Serveur d'application Java Enterprise Edition

Java EE (Serveur d'application Web Java)

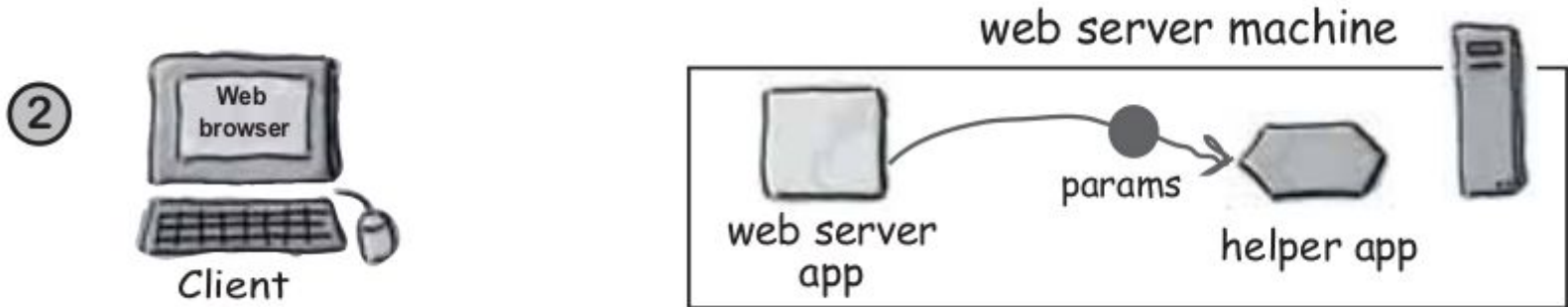
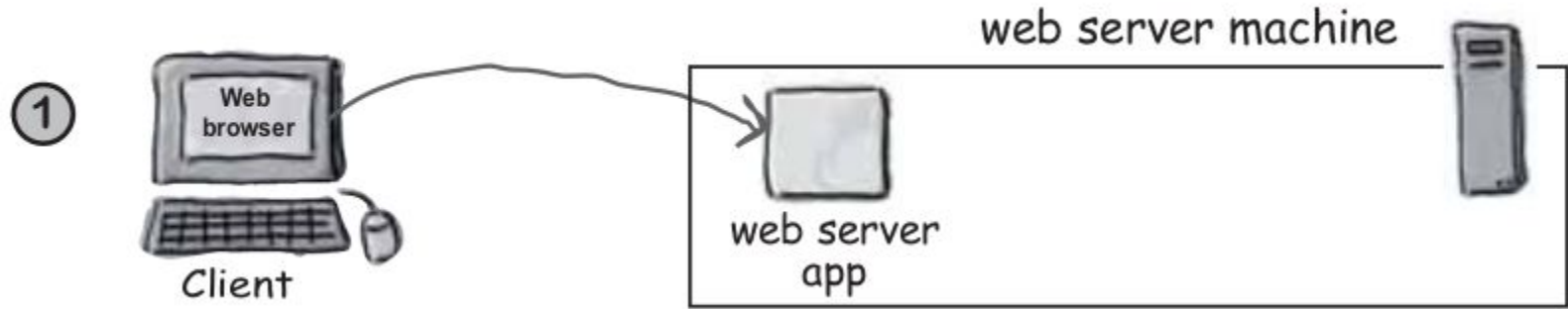
- Un serveur dite d'application Web est en mesure de fournir du contenu Web.
- Or ce contenu Web est généré grâce à la technologie Servlet ou JSP qui est donc un sous ensemble de Java Entreprise. c'est pourquoi on appelle les serveurs d'application Web Java des Conteneurs de Servlet



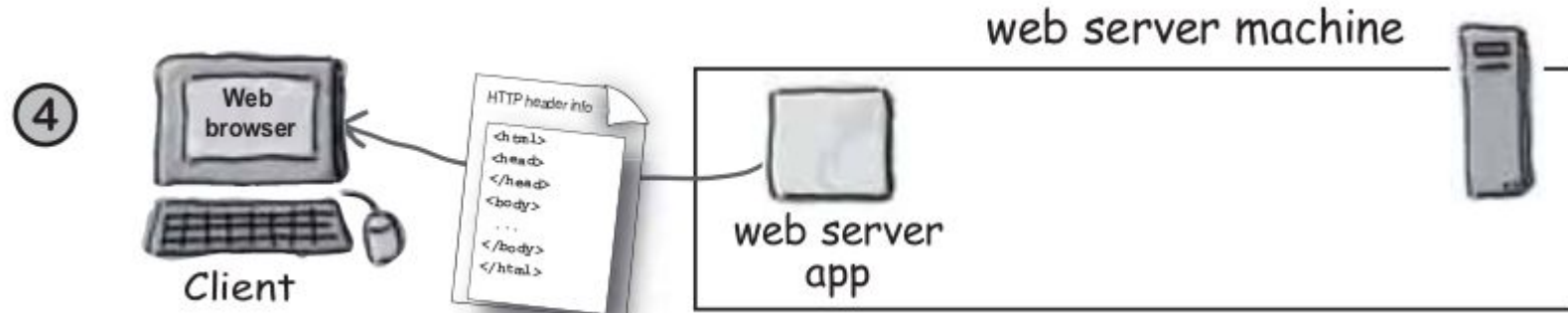
Java EE (Serveur d'application Java EE)

- Les serveurs dits d'application Java EE doivent quant à eux théoriquement pouvoir mettre en œuvre toutes les technologies Client-Serveur issu de la norme Java EE.
 - ◆ Servlet/JSP
 - ◆ mais aussi d'autres technologies EJB, JMS, etc.
- Dans certains cas, le fonctionnement interne du serveur d'applications Java EE consiste à déléguer le traitement des Servlet/JSP à un conteneur de Servlet embarqué.

Java EE (Architecture WEB)



Java EE (Architecture WEB)



Java EE (Application Web)

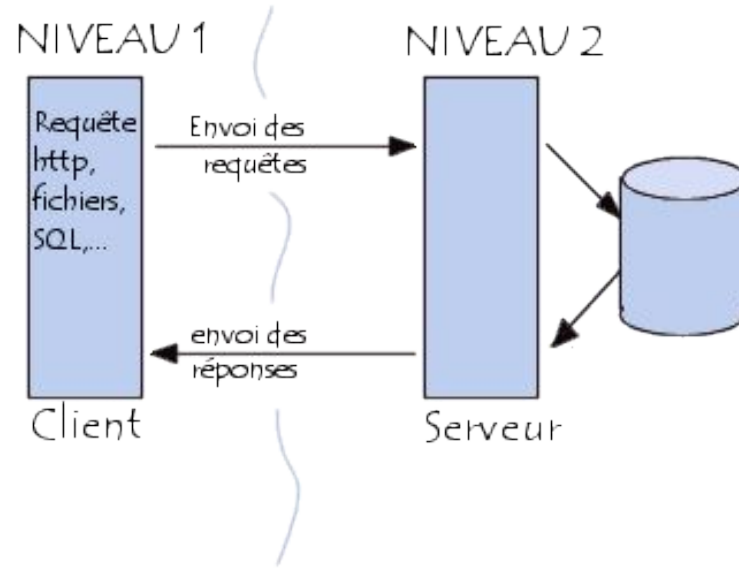
- Une application web n'est ni plus ni moins qu'un répertoire qui contient un fichier WEB-INF/web.xml
- Ce fichier s'appelle descripteur de déploiement et indique au conteneur de servlet comment il doit prendre en charge l'application.
- Exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  id="WebApp_ID" version="3.0">
  ....
</web-app>
```

Java EE (Architecture WEB)

→ L'architecture 2tiers

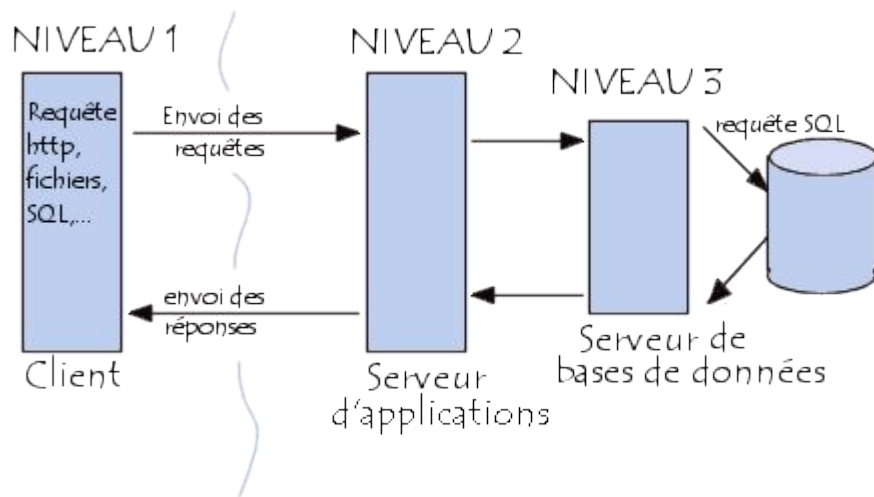
- ◆ composée de deux éléments, un client et un serveur.
- ◆ peut aussi être représentée avec un serveur de base de données (SGBD)



Java EE (Architecture WEB)

→ L'architecture 3tiers est :

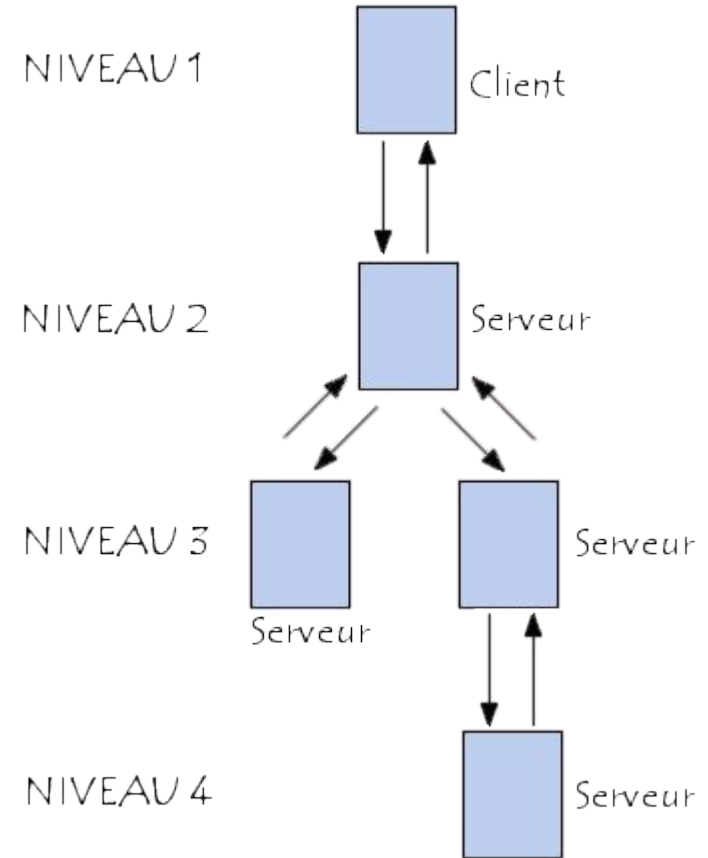
- ◆ La couche présentation (de premier niveau)
- ◆ La couche métier (de second niveau)
- ◆ La dernière couche (de troisième niveau)



Java EE (Architecture WEB)

→ L'architecture n-tiers

- ◆ a été pensée pour pallier les limitations des architectures 3tiers et concevoir des applications puissantes et simples à maintenir



Java Bean

Java Bean

- l'un des caractéristiques de java est de faciliter la réutilisation des des objets par des personnes et ou des logiciels.
- par exemple:

```
public class ReUtiliser {  
    public static void main(String args[]) {  
        try {  
            Class ClassReUtilisable = Class.forName("Eshop.Utilisateur");  
            Object instance = ClassReUtilisable.newInstance();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Java Bean

- Standard pour la réécriture des composants réutilisables.
- Java Beans sont des classes qui encapsulent plusieurs objets en un seul (Bean).
- Il respecte les convention suivantes:
 - ◆ Doit implémenter "Serializable"
 - ◆ Constructeur par défaut sans paramètre.
 - ◆ Tous les attributs doivent être privés.
 - ◆ doit avoir des setters et des getters.

Java Bean

Java.io.Serializable

- Serializable est une interface “Drapeau” qui ne demande la réécriture d’aucune méthode.
- Elle indique simplement que les objets de cette classe peuvent être transportés dans un flux de lecture ou d’écriture.

Constructeur

- Toute classe qui ne possède explicitement aucun constructeur possède un constructeur par défaut qui est le constructeur sans paramètre, il n’est donc pas nécessaire d’en définir un.
- Si on décide de créer un ou plusieurs constructeurs avec paramètres, il sera alors nécessaire d’ajouter explicitement le constructeur sans paramètres.

Mettons cela en oeuvre

—