


**La plateforme Java Entreprise Edition**



**Beginning  
Java EE 7**

**Mahmoud EL HAMLAOUI**  
Enseignant chercheur  
IT architecture and Model driven Systems development  
ADMIR, ENSIAS







"To persuade me of the merit of your language,  
you must show me how to construct *models* in it."  
1978 Turing Award Lecture



---

---

---

---

---

---

---

---

**Pourquoi utiliser une plateforme**

- Une plateforme est une base générique qui
  - fournit un ensemble de fonctionnalités utiles pour une majorité d'applications
  - se construit sur la base d'un ensemble de besoins génériques partagés entre plusieurs applications
- Parmi les avantages
  - facilité de prise en main des fonctionnalités de cette plateforme
  - pas d'intérêt à recoder des fonctionnalités communes et récurrentes
  - gain de temps et d'argent

EL HAMLAOUI Mahmoud 59

---

---

---

---

---

---

---

---

**La nébuleuse JAVA**

- Java Framework est composé de trois éditions, destinées à des usages différents
  - Java Micro Edition
    - développement d'applications embarquées
  - Java Standard Edition
    - développement d'applications classiques
  - Java Entreprise Edition
    - développement d'application d'entreprise

EL HAMLAOUI Mahmoud 60

---

---

---

---

---

---

---

---

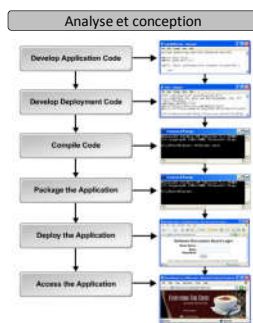
## Application Entreprise

- Un logiciel qui fournit des services à un grand nombre d'utilisateurs sur un réseau
- Permet de fournir des solutions de conception et de mise en œuvre pour des domaines d'application spécifiques
  - ❑ planification de ressources
  - ❑ gestion de relation client
  - ❑ gestion de la chaîne de production dans les organisations



EL HAMILAOUTI Mahmoud

## Application Entreprise



EL HAMILAOUTI Mahmoud

62

## Vue d'ensemble

- La plateforme JEE offre une approche composant à la conception, le développement, l'assemblage et le déploiement d'applications distribuées
- Elle Offre
  - ❑ un modèle d'application distribuées multi-tiers
  - ❑ un modèle de composants réutilisables
  - ❑ un modèle de sécurité unifié
  - ❑ un système de transaction flexible
  - ❑ un support de service Web
  - ❑ un Framework de développement de site Web

EL HAMILAOUTI Mahmoud

63

## Vue d'ensemble

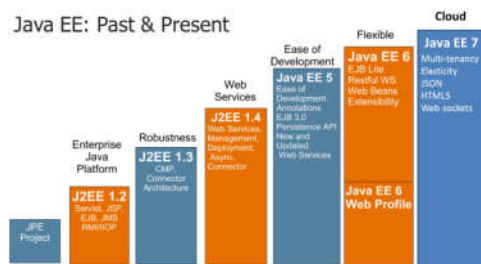
- La base du modèle d'application JEE est le langage de programmation Java et la JVM
  - portabilité
  - sécurité
  - productivité des développeurs
- Le modèle d'application définit une architecture pour la mise en œuvre des applications multi-tiers
  - l'évolutivité
  - l'accessibilité
  - la facilité de gestion

EL HAMILAOUTI Mahmoud

64

## Historique

Java EE: Past & Present

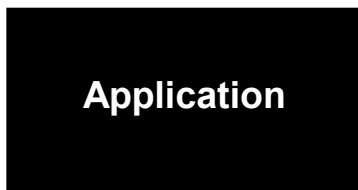


EL HAMILAOUTI Mahmoud

65

## Approche nostalgique

- Noyau monolithique
- Mélange entre présentation, règles métier et données



BD

EL HAMILAOUTI Mahmoud

66

## Approche nostalgique

```
// application logic
if (!empty($_POST["userId"])) {

    // data base access
    $c = DB::Connect("mysql://user:passwd@host/db");
    $res = $conn->Query(
        "SELECT * FROM user WHERE id = "
        + $_POST["userId"] + "");
    $row = $res->FetchRow(DB_FETCHMODE_ASSOC);

    // business logic
    $displayName = $row["display_name"];

    // presentation
    echo "Hello " . $row["display_name"] . "!";
}
```

EL HAMILAOUTI Mahmoud

67

## Approche 2 tiers

- Version 1 :
  - Client : Présentation
  - Serveur : Services métiers et Persistance
- Version 2 :
  - Client : Présentation et Services métiers
  - Serveur : Persistance
- Version 3 :
  - Client : Présentation minimale
  - Serveur : Présentation, Services métiers et Persistance
- Version 4 :
  - Client : Présentation et Service métiers
  - Serveur : Services métiers et Persistance



EL HAMILAOUTI Mahmoud

68

## Approche 3 tiers

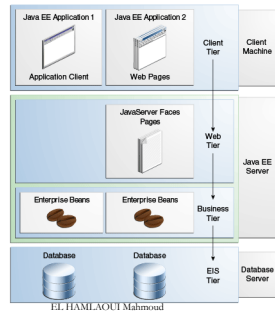
- Les 3 principaux tiers s'exécutent sur des machines différentes :
  - Machine Client : Présentation,
  - Serveur d'application : Services métiers,
  - Serveur de base de données : Persistance.



EL HAMILAOUTI Mahmoud

69

## Approche multi-tiers



70

## Composants JAVA EE

- La spécification JEE (JSR 342) définit 3 types de composants
  - ❑ les applications clients et applets qui sont des composants
    - fonctionnant chez le client
  - ❑ les Servlets, les JavaServer Pages (JSP) et les pages JavaServer Face (JSF) qui sont des composants Web
    - fonctionnant sur le serveur
  - ❑ les Enterprise JavaBeans (EJB) qui sont des composants métiers (*Business*)
    - fonctionnant sur le serveur.
  - ❑ les JavaBeans sont souvent utilisés dans les applications JEE mais ne sont pas considérés comme des composants JEE par la spécification.

EL HAMILAOUI Mahmoud

71

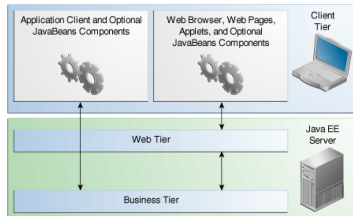
## Clients JAVA EE

- Un client JEE peut être un client fonctionnant sur un navigateur Internet ou une application cliente
  - ❑ Clients Web : pages écrites en HTML ou XML (générées par des « composants Web » s'exécutant sur le tiers Web) interprétées par un navigateur Internet
  - ❑ Applets : petite application client écrite en Java et s'exécutant dans la zone sécurisée du navigateur Internet. Elles nécessitent l'installation de la machine virtuelle Java sur le poste du client
  - ❑ Application Client : application écrite en Java s'exécutant directement sur le poste client. Elles nécessitent aussi l'installation de la machine virtuelle Java sur le poste du client mais permettent généralement la mise en œuvre d'interfaces graphiques sophistiquées (écrites en Swing) ainsi que l'accès aux ressources du poste client (disque dur, réseau, etc.)

EL HAMILAOUI Mahmoud

72

## Clients JAVA EE

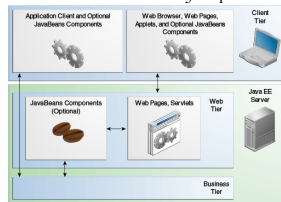


EL HAMILAOUTI Mahmoud

73

## Composant Web

- Les composants Web sont des servlets ou des pages JSP ou des pages JSF qui exécutent dynamiquement des requêtes et construisent des réponses
- Les composants Web peuvent utiliser des JavaBeans afin de gérer l'interaction entre l'utilisateur et les EJB qui s'exécutent sur le tiers métier

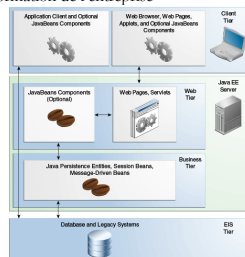


EL HAMILAOUTI Mahmoud

74

## Composant métier

- Ces composants appelés Entreprise JavaBeans permettent de répondre à la logique du domaine d'application
  - ❑ ne contiennent pas le système d'information de l'entreprise
  - ❑ existe en 2 types



EL HAMILAOUTI Mahmoud

75

## Le tiers du système d'information de l'entreprise

- Ce tiers correspond aux infrastructures propres à l'entreprise telles que les bases de données, le système de planification des ressources, mainframe de traitement des transactions, etc.
- Par exemple
  - JDBC et JPA pour la persistance des données

EL HAMILAOUTI Mahmoud

76

## Conteneurs JAVA EE

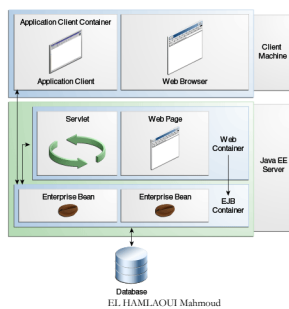
- Un conteneur JEE est l'interface entre un composant et tous les éléments de bas niveaux nécessaire à l'exécution
- Un module JEE (Servlet, EJB, etc.) doit être déployé dans son container afin d'être exécuté
- Le paramétrage d'un conteneur peut agir sur
  - le modèle de sécurité JAVA EE
  - le modèle de transaction JAVA EE
  - le système de gestion des noms JNDI
  - le modèle de communications distantes transparentes entre clients et EJB

EL HAMILAOUTI Mahmoud

77

## Conteneurs JAVA EE

- Répartition des conteneurs sur les serveurs



EL HAMILAOUTI Mahmoud

78

## Serveurs JAVA EE

### ■ Un serveur d'application JAVA EE fournit

- ❑ un conteneur WEB
- ❑ et/ou Un conteneur EJB

### ■ Exemple de serveurs d'application

- ❑ Tomcat : Module WEB
- ❑ Jboss : Module WEB et Module EJB
- ❑ WildFly : Module WEB et Module EJB
- ❑ GlassFish : Module WEB et Module EJB
- ❑ Jonas: Module WEB et Module EJB
- ❑ Websphere: Module WEB et Module EJB
- ❑ WebLogic: Module WEB et Module EJB
- ❑ Etc.



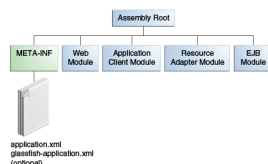
EL HAMILAOUTI Mahmoud

79

## Emballage d'applications

### ■ Une application JAVA EE peut être emballée dans

- ❑ fichier JAR (Java Archive)
- ❑ fichier WAR (Web Archive)
- ❑ fichier EAR (Enterprise Archive)



EL HAMILAOUTI Mahmoud

80

## Emballage d'applications

### ■ Le descripteur de déploiement est un fichier de code XML qui informe l'environnement d'exécution sur un composant.

En-tête du fichier web.xml

Balise principale

Balise de description de l'application WEB

Balise de description d'une Servlet

Nom de la Servlet "Identification"

Classe de la Servlet

Définition d'un chemin virtuel

Nom de la Servlet considéré "Identification"

Définition du chemin virtuel

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
  version="2.4">
  <display-name>Application WEB</display-name>
  <servlet>
    <servlet-name>HelloWorldServlet</servlet-name>
    <servlet-class>com.helloWorldServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HelloWorldServlet</servlet-name>
    <servlet-class>com.helloWorldServlet</servlet-class>
    <url-pattern>/helloWorldServlet</url-pattern>
  </servlet-mapping>
</web-app>

```

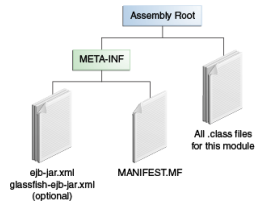
EL HAMILAOUTI Mahmoud

81



## Emballage d'applications

- Il est possible de déployer un EJB emballé dans un JAR qui ne figure pas dans un EAR

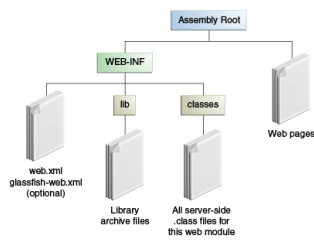


EL HAMILAOUTI Mahmoud

82

## Emballage d'applications

- Un module Web peut être déployé
  - sans être emballé
  - ou bien emballé dans un WAR

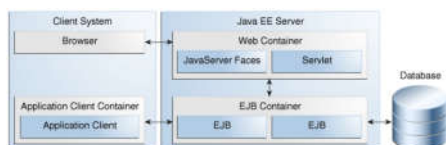


EL HAMILAOUTI Mahmoud

83

## Bibliothèques JAVA EE

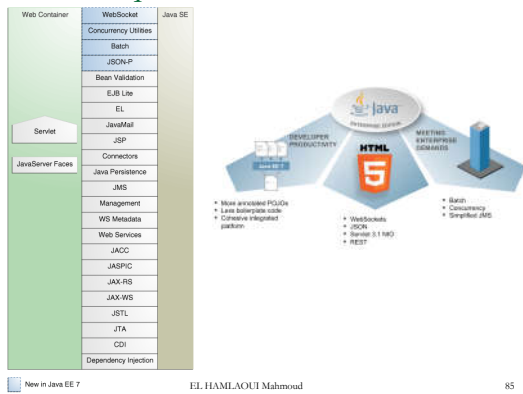
- JAVA EE fournit un ensemble de bibliothèque pour le développement d'applications d'entreprises



EL HAMILAOUTI Mahmoud

84

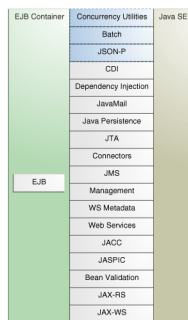
## Bibliothèques du conteneur web



EL HAMILAOUTI Mahmoud

85

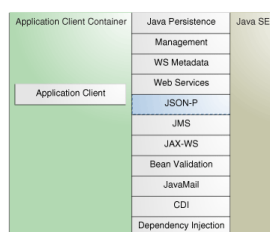
## Bibliothèques du conteneur EJB



EL HAMILAOUTI Mahmoud

86

## Bibliothèques du conteneur de l'application cliente



EL HAMILAOUTI Mahmoud

87

## Servlets/JavaServer Pages



EL HAMILAOUTI Mahmoud

88

## Introduction

- Le web est constitué d'innombrables clients et serveurs connectés par une liaison filaire ou sans fils.
- L'objectif de ce cours est de vous permettre de construire une application web accessible aux clients du globe.

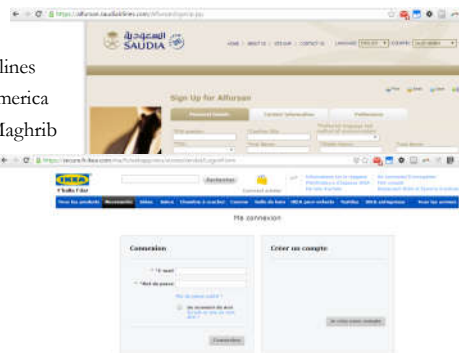


EL HAMILAOUTI Mahmoud

89

## JSP/Servlets dans le monde réel

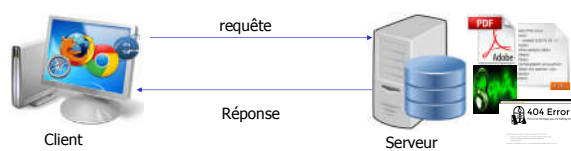
- Saudia Airlines
- Bank of America
- Bank Al-Maghrib
- Hertz
- Nike
- Ikea
- Nascar
- INWI
- ...



EL HAMILAOUTI Mahmoud

90

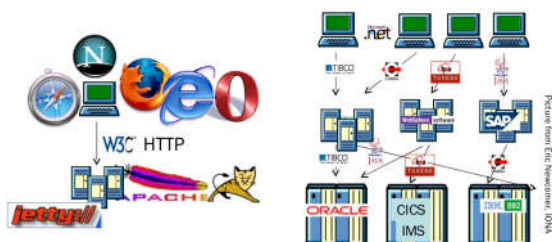
## Communication client/Serveur



EL HAMLAOUI Mahmoud

91

## Communication Client/Serveur



EL-HAMLAOUI Mahmoud

92

## URL (Uniform Resource Locator)

- Permet de localiser une ressource sur le web
- *protocole://domaine:port/cheminDeLaRessource*
- URI VS. URL VS. URN
  - Exemple :
    - URI : EL HAMLAOUI Mahmoud
    - URN : EL HAMLAOUI Mahmoud + numéro passeport
    - URL : EL HAMLAOUI Mahmoud + numéro passeport + Coordonnées GPS

EL HANLAOU Mahmoud

93

## URL (Uniform Resource Locator)

### ■ Chaîne de requête :

- *protocol://domaine:port/cheminDeLaRessource/méthode?nomParam1=valeur1&nomParam2=valeur2*

### ■ Fragment :

- *protocol://domaine:port/cheminDeLaRessource#fragment*

EL HAMILAOUTI Mahmoud

94

## Protocole HTTP (HyperText Transfer Protocol)

- Protocole de communication développé pour le web
- Utilisé pour transmettre les ressources entre le client (navigateur) et le serveur web

EL HAMILAOUTI Mahmoud

95

The screenshot displays the ENSIAS website interface with the header 'LA GRANDE ÉCOLE DU GÉNIE INFORMATIQUE'. Below the header, there is a navigation menu and a main content area. The developer tools are open, showing a 404 Not Found error for the URL `http://www.ensias.ma/modules/whic_menu/js/nootools.js`. The response headers indicate the content type is `text/html` and the status is `404 Not Found`. The request headers show the user agent as `Mozilla/5.0 (Windows NT 6.1; WOW64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36)`.

## Requête HTTP



EL HAMILAOUTI Mahmoud

97

## Requête HTTP

### ■ Méthodes HTTP :

- ❑ GET : Récupère une ressource. Cette méthode n'est sans effet sur la ressource
- ❑ HEAD : Demande uniquement des informations sur la ressource
- ❑ POST : Envoie une requête de taille illimitée. Elle est utilisée lorsqu'une requête modifie la ressource.
- ❑ PUT : Ajoute une ressource sur le serveur
- ❑ TRACE : Effectue un diagnostic sur la connexion en demandant au serveur de retourner ce qu'il a reçu
- ❑ OPTION : Permet d'obtenir les options de communication de la ressource
- ❑ DELETE : Supprime une ressource du serveur
- ❑ CONNECT : établit un tunnel de communication

EL HAMILAOUTI Mahmoud

98

## Requête HTTP (GET VS. POST)

### ■ Exemple :

```
GET /fichier.ext?variable=valeur&variable2=valeur2 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

```
POST /fichier.ext HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: text/xml; charset=utf-8
Content-Length: 88
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

```
variable=valeur&variable2=valeur2
```

## Réponse HTTP

HTTP/1.1 200 OK	Status Line
Date: Thu, 20 May 2004 21:12:58 GMT	General Headers
Connection: close	
Server: Apache/1.3.27	Response Headers
Accept-Ranges: bytes	
Content-Type: text/html	Entity Headers
Content-Length: 170	
Last-Modified: Tue, 18 May 2004 10:14:49 GMT	Message Body
<html>	
<head>	HTTP Response
<title>Welcome to the Amazing Site!</title>	
</head>	
<body>	
<p>This site is under construction. Please come back later. Sorry!</p>	
</body>	
</html>	

EL HAMILAOUTI Mahmoud

100

## Réponse HTTP

- Code de statut sont divisé en 5 catégories :
  - ❑ 1XX (information) : indique que la requête est reçue et que le serveur poursuit le processus
    - 100 : Continue
    - 101 : Switching Protocols
  - ❑ 2XX (succès) : indique que la requête est reçue avec succès, comprise et acceptée
    - 200 : OK
    - 201 : created
  - ❑ 3XX (Redirection) : indique que d'autres mesures doivent être prises afin de compléter la demande
    - 300 : Multiple Choices
    - 301 : Moved permanently

EL HAMILAOUTI Mahmoud

101

## Réponse HTTP

- Code de statut sont divisé en 5 catégories :
  - ❑ 4XX (Erreur du client web) : indique que la requête n'a pas été comprise
    - 400 : Bad Request
    - 401 : *Unauthorized*
  - ❑ 5XX (Erreur du serveur) :
    - 500 : Internal Server Error
    - 501 : Not Implemented

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

EL HAMILAOUTI Mahmoud

102

## Réponse HTTP

- MIME (Multipurpose Internet Mail Extensions) est standard pour typer les documents transférés :
  - ❑ Content-type:TYPE-PRINCIPAL/SOUS-TYPE
- Exemples :
  - ❑ Content-type: image/gif
  - ❑ Content-type: text/html
  - ❑ Content-type: application/json
  - ❑ Content-type: application/pdf

EL HAMILAOUI Mahmoud

103

---

---

---

---

---

---

---

---

EL HAMILAOUI Mahmoud

104

---

---

---

---

---

---

---

---

## Logiciels requis

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>■ Tomcat               <ul style="list-style-type: none"> <li>❑ Installer Java</li> <li>❑ Installer un IDE</li> <li>❑ Télécharger Tomcat</li> <li>❑ Récupérer les jars en fonction du besoin JSF 2_2.jar</li> <li>❑ Ajouter les fichiers de configuration                   <ul style="list-style-type: none"> <li>■ En fonction du Framework</li> </ul> </li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>■ Java EE 7 server               <ul style="list-style-type: none"> <li>❑ Installer Java</li> <li>❑ Installer un IDE</li> <li>❑ Télécharger un serveur supportant java JEE 7                   <ul style="list-style-type: none"> <li>■ GlassFish 4, Jboss 8, wildfly 8, JEUS 8, webSphere</li> </ul> </li> <li>❑ Pas besoin de fichiers JAR</li> <li>❑ Ajouter les fichiers de configuration                   <ul style="list-style-type: none"> <li>■ En fonction du Framework</li> </ul> </li> </ul> </li> </ul> |
|---|---|

EL HAMILAOUI Mahmoud

105

---

---

---

---

---

---

---

---



## Exercice : Manipulation de l'environnement

EL HAMILAOUI Mahmoud

106

## Exercice

- Suivre la présentation d'Eclipse (Workspace, Plugins, Editors, Views, Perspective, Preferences)
- Démarrer la version d'Eclipse fourni et lancer le serveur d'application wildfly
- Connecter-vous au serveur sur <http://localhost:8080>
- Lancer la console d'administration (port 4848)
- Créer une application Web (en gardant pour l'instant par défaut tous les paramètres)
- Observer l'architecture générée
- Créer deux pages html avec un lien entre eux
- Déployer le projet sur le serveur et visualiser le résultat

EL HAMILAOUI Mahmoud

107

## Servlet

EL HAMILAOUI Mahmoud

108

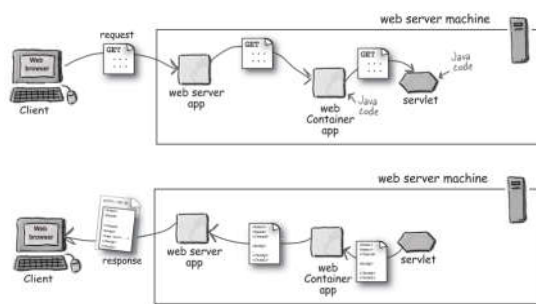
## Présentation

- Une Servlet est un composant web conçu à partir d'une classe Java déployée au sein d'une application
- Elle interagit avec un client Web par l'intermédiaire du protocole HTTP
- Elle apporte un contenu dynamique en réponses à des requêtes client

EL HAMILAOUTI Mahmoud

109

## Présentation



EL HAMILAOUTI Mahmoud

110

## Avantages

- Les servlets possèdent de nombreux avantages :
  - ❑ Elles sont portables et évolutives
  - ❑ Elles sont performantes et rapides
  - ❑ Elles sont disponibles (contexte multitâche)

EL HAMILAOUTI Mahmoud

111

## Servlet 3.0

- La spécification de la Servlet 3.0 est détaillée dans la JSR-315.
- Parmi les nouveautés cette version de la Servlet :
  - ❑ Utilisation des annotations pour la définition
    - des Servlet (@WebServlet),
    - des filtres (@ServletFilter),
    - ainsi que des écouteurs (@WebListener)
  - ❑ Définition des Servlets, filtres et écouteurs par programmation
  - ❑ Définition des Servlet Asynchrones
  - ❑ Définition des web-fragments

[1 http://download.oracle.com/otndocs/jsp/servlet-3.0-fr-eval-oth-JSpec/](http://download.oracle.com/otndocs/jsp/servlet-3.0-fr-eval-oth-JSpec/)

EL HAMELAOUI Mahmoud

112

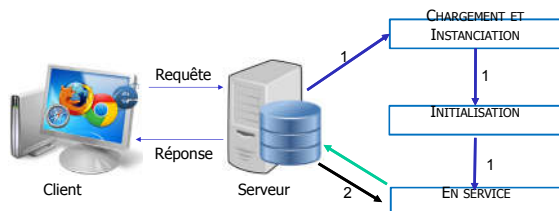
## Servlet 3.0

- ❑ Gestion de l'initialisation de bibliothèques partagées
- ❑ Définition de ressources statiques dans les archives JAR
- ❑ Gestion personnalisée des sessions utilisateurs
- ❑ Gestion optimisée de l'upload de fichiers
- ❑ Gestion du processus d'authentification par me conteneur Web

EL HAMELAOUI Mahmoud

113

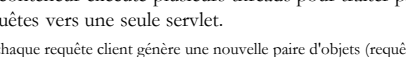
## Cycle de vie

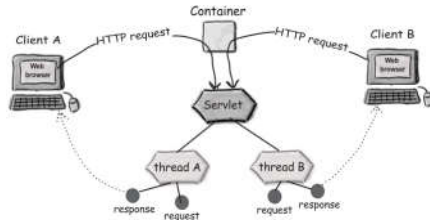


EL HAMELAOUI Mahmoud

114

## Cycle de vie

- Le conteneur exécute plusieurs threads pour traiter plusieurs requêtes vers une seule servlet.
    - chaque requête client génère une nouvelle paire d'objets (requête et réponse).
- 
- The diagram shows a central 'Container' box at the top. Below it is a 'Servlet' box. To the left is 'Client A' (a laptop) and to the right is 'Client B' (a laptop). Arrows labeled 'HTTP request' point from both clients to the 'Container'. Arrows labeled 'HTTP response' point from the 'Container' back to both clients. A dashed arrow points from the 'Servlet' box to the 'Container' box, indicating the servlet is managed by the container.
- Est-ce un thread par client ou un thread par demande ?



EL. HAMLAOUI Mahmoud

115

## Hiérarchie de servlet

- La hiérarchie de la servlet est composée d'une :
  - Interface Servlet
  - Une classe abstraite *GenericServlet*
  - Une classe abstraite *HttpServlet*



EL HAMLAOUI Mahmoud

## Hiérarchie de servlet

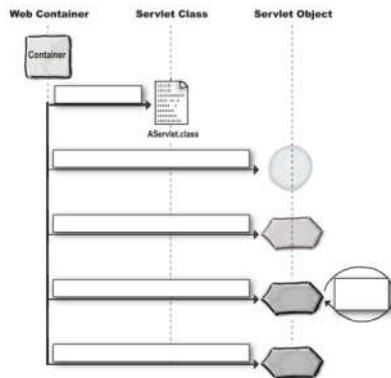
- Les méthodes qui correspondent aux cycles de vie d'une Servlet sont :
  - La méthode `init()`
    - `ServletConfig` : permet de charger les paramètres de configuration.
    - En cas d'erreur l'exception `ServletException` est levée
  - La méthode `service()`
  - La méthode `destroy()`



EL HAMLAOUI Mahmoud

117

## Pause



EL HAMILAOUTI Mahmoud

118

## Hiérarchie de servlet

- Les méthodes qui correspondent aux cycles de vie d'une Servlet sont :
  - La méthode `init()`
  - La méthode `service()`
    - `HttpServletRequest`
    - `HttpServletResponse`
    - En cas d'erreur l'exception `ServletException` est levée
  - La méthode `destroy()`



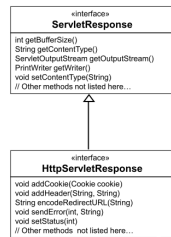
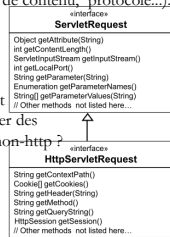
EL HAMILAOUTI Mahmoud

119

## HttpServletRequest Vs. HttpServletResponse

- `javax.servlet.ServletException` fournit des méthodes permettant de récupérer ou d'ajouter des données dans la requête ainsi que des méthodes pour les métadonnées de la requête (infos clients, taille, type de contenu, protocole...).

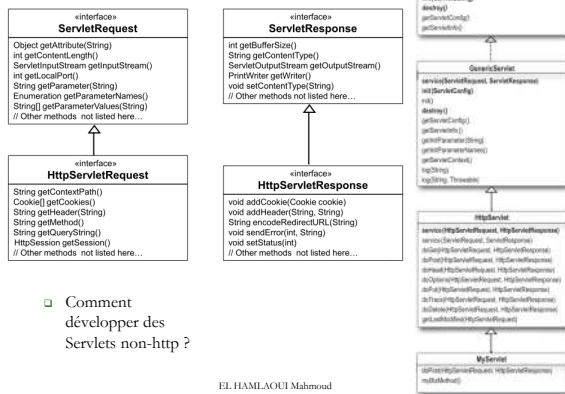
Comment développer des Servlets non-http ?



- `javax.servlet.HttpServletResponse` fournit des méthodes pour insérer des données dans la réponse renvoyée au client.

EL HAMILAOUTI Mahmoud

## HttpServletRequest Vs. HttpServletResponse



- Comment développer des Servlets non-http ?

EL. HAMLAOUI Mahmoud

---

---

---

---

---

---

## Exemple Hello v1

```
package testPackage; // Always use packages.
import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
    @Override
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        out.println("Hello World!");
    }
}
```

EL HAMLAOUI Mahmoud

122

[illegible]

## Déploiement de de la Servlet

- Maintenant qu'on sait comment écrire une Servlet, comment peut-on l'utiliser ?
  - Descripteur de déploiement
  - Annotations Servlet >= version 3.0

EL HAMLAOUI Mahmoud

123

---

---

---

---

---

---

## Configuration

### ■ D.D

```
<servlet>
  <servlet-name>myServlet</servlet-name>
  <servlet-class>
    my.server.side.stuff.MyAwesomeServlet
  </servlet-class>
  <init-param>
    <param-name>configFile</param-name>
    <param-value>config.xml</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>myServlet</servlet-name>
  <url-pattern>/path/to/my/servlet</url-pattern>
</servlet-mapping>
```

EL HAMILAOUTI Mahmoud

124

## Configuration

### ■ D.D

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="...">

  <servlet>
    <servlet-name>myServlet</servlet-name>
    <servlet-class>
      my.server.side.stuff.MyAwesomeServlet
    </servlet-class>
    <init-param>
      <param-name>configFile</param-name>
      <param-value>config.xml</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>myServlet</servlet-name>
    <url-pattern>/path/to/my/servlet</url-pattern>
  </servlet-mapping>
```

http://hostname/appName/path/to/my/servlet

EL HAMILAOUTI Mahmoud

125

## Configuration

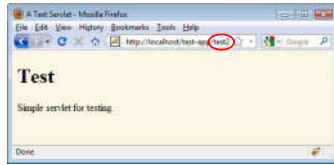
### ■ Annotation

```
@WebServlet(name="myServlet",
  urlPatterns={"/path/to/my/servlet"},
  initParams={@InitParam(name="configFile",
    value="config.xml")})
public class MyAwesomeServlet extends HttpServlet { ... }
```

EL HAMILAOUTI Mahmoud

126

## Defining Custom URLs: Result (à faire)



### ■ Eclipse details

- Name of Eclipse project is "test-app"
- Servlet is in src/testPackage/TestServlet.java
- Deployed by right-clicking on Tomcat, Add and Remove Projects, Add, choosing test-app project, Finish, right-clicking again, Start (or Restart)

EL HAMILAOUTI Mahmoud

127

## URL patterns

Règle	URL pattern	URL possibles
Exact	/any	/any
Chemin	/any/*	/any/ /any/other /any/index.htm
extension	*.jsp	/index.jsp /any.index.jsp
Par défaut	/	N'importe quelle url

EL HAMILAOUTI Mahmoud

128

## URL patterns (exercices)

URL pattern	Exemple	OK/KO ?
<url-pattern>/status/* </url-pattern>	http://example.com/examples/status/synopsis	
	http://example.com/examples/server/status	
	http://example.com/examples/status/complete?date=today	
<url-pattern>*.map </url-pattern>	http://example.com/examples/US/Oregon/Portland.map	
	http://example.com/examples/US/Washington/Seattle.map	
	http://example.com/examples/Paris.France.map	
	http://example.com/examples/US/Oregon/Portland.MA P	
	http://example.com/examples/interface/description/mail. map	

EL HAMILAOUTI Mahmoud

129



## URL patterns (exercices)

URL pattern	Exemple	OK/KO ?
<url-pattern>/status/* </url-pattern>	http://example.com/examples/status/synopsis	OK
	http://example.com/examples/server/status	KO
	http://example.com/examples/status/complete?date=today	OK
<url-pattern>*.map </url-pattern>	http://example.com/examples/US/Oregon/Portland.map	OK
	http://example.com/examples/US/Washington/Seattle.map	OK
	http://example.com/examples/Paris.France.map	OK (maj)
	http://example.com/examples/US/Oregon/Portland.MA.p	KO (maj)
	http://example.com/examples/interface/description/mail.mapi	KO (mapi)

EL HAMLAOUTI Mahmoud

130

## URL patterns (encore ?!!)

- Quel pattern sera-t-il utilisé avec la requête suivante :
  - http://example.com/servlet/Catalog/search/index.jsp

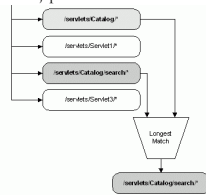
Web.xml

```

<servlet-mapping>
  <servlet-name>catalogBrowse</servlet-name>
  <url-pattern>/Catalog/*</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>catalogSearch</servlet-name>
  <url-pattern>/Catalog/search/*</url-pattern>
</servlet-mapping>

```



EL HAMLAOUTI Mahmoud

131

## Exercices *content type*

- En utilisant l'entête HTTP, écrire les Servlets permettant d'envoyer au client un contenu :
  - format HTML
  - format XML
  - dynamique sous la forme d'un flux d'octets (image par exemple)
    - transformer l'image en BufferedImage
  - Télécharger un JAR
    - BufferedInputStream ()
    - getResourceAsStream ()

EL HAMLAOUTI Mahmoud

132

## Idempotence d'une requête

EL HAMILAOUI Mahmoud

133

---

---

---

---

---

---

---

---

## Partage d'informations

- Les composants Web, comme la plupart des objets, travaillent généralement avec d'autres objets pour accomplir leurs tâches.
- Les composants Web peuvent le faire en :
  - Utilisation d'objets helper privés (par exemple, composants JavaBeans),
  - Partage des objets qui sont des attributs d'une portée publique,
  - Utilisation d'une base de données,
  - Invocation d'autres ressources Web (inclusion, délégation).

EL HAMILAOUI Mahmoud

134

---

---

---

---

---

---

---

---

## Portée des objets

- La portée d'un attribut indique la disponibilité de cet attribut à être utilisé.
- Une application Web répond à de multiples demandes des clients lorsqu'elle est opérationnelle. Ces demandes peuvent provenir du même client ou de clients différents.

EL HAMILAOUI Mahmoud

135

---

---

---

---

---

---

---

---

## Portée des objets

- La portée Application est représentée par l'interface `javax.servlet.ServletContext`.
- La portée Request est représentée par l'interface `javax.servlet.http.HttpServletRequest`.
- La portée Session est représentée par l'interface `javax.servlet.http.HttpSession`.
- La portée Scope est représentée par la classe abstraite `javax.servlet.jsp.PageContext`.

EL HAMILAOUI Mahmoud

136

## Portée des objets

- Chacun portée peut gérer les attributs avec les méthodes :
  - ❑ `void setAttribute(String nom, Object value)`
  - ❑ `Object getAttribute(String nom)`
  - ❑ `Enumeration getAttributeNames()`
  - ❑ `void removeAttribute(String nom)`

EL HAMILAOUI Mahmoud

137

## L'interface ServletContext

- Un objet de type `javax.servlet.ServletContext` représente les informations de configuration d'une application Web.
  - ❑ À l'inverse de l'objet `javax.servlet.ServletConfig` qui représente les informations de configuration d'un Servlet au sein d'une application Web
- Permet la communication entre les servlets et le conteneur de servlets
  - ❑ Chaque Servlet de la même application Web aura donc accès à ces paramètres,
  - ❑ Une seule instance par application Web.
- Accès en lecture et en écriture dans le fichier de configuration.

EL HAMILAOUI Mahmoud

138

## Exemple

### ■ Web.xml

```
<web-app>
  <!-- paramètres globaux -->
  <context-param>
    <param-name>emailAdministrateur</param-name>
    <param-value>admin@betaboutique.fr</param-value>
  </context-param>
  <servlet>
    <servlet-name>servletauthentification</servlet-name>
    <servlet-class>
      ma.ensias.Servlet Authentification
    </servlet-class>
    <init-param>
      <param-name>defaultIdentifiant</param-name>
      <param-value>monidentifiant</param-value>
    </init-param>
    <init-param>
      <param-name>defaultMotDePasse</param-name>
      <param-value>monmotdepasse</param-value>
    </init-param>
  </servlet>
</web-app>
```

EL HAMILAOUTI Mahmoud

139

## Exemple

### ■ Récupérer des paramètres

```
public class ServletAuthentification extends HttpServlet {
  //variables de classe
  String ident=null;
  String mdp=null;
  String emailadministrateur=null;
  public void init()
  {
    //récupération des paramètres d'initialisation
    de la Servlet dans le fichier web.xml
    config=getServletConfig();

    ident=(String)config.getInitParameter("defaultIdentifiant");
    mdp=(String)config.getInitParameter("defaultMotDePasse");
    //récupérer l'email de l'administrateur
    ServletContext servletContext=getServletContext();
    emailadministrateur=servletContext.getInitParameter
    ("emailAdministrateur");
  }
}
```

EL HAMILAOUTI Mahmoud

140

## Exemple

```
public void doGet(HttpServletRequest request, HttpServletResponse
response)throws ServletException, IOException
{
  //récupération de l'identifiant/login dans la requête
  String identifiant=request.getParameter("identifiant");
  //récupération du mot de passe dans la requête
  String motdepasse=request.getParameter("motdepasse");
  //flux de sortie
  PrintWriter out=response.getWriter();
  //pas d'identifiant
  if(identifiant==null)
  { out.println("Authentification incorrecte !");}
  //pas de mot de passe
  if(motdepasse==null)
  { out.println("Authentification incorrecte !");}
  //vérifier l'égalité des valeurs
  if( (identifiant!=null && identifiant.equals(ident))
  && (motdepasse!=null && motdepasse.equals(mdp)) )
  { out.println("Authentification correcte, bienvenue :
  "+identifiant); }
  else
  { out.println("Authentification incorrecte, mauvaise
  saisie des informations !");
  out.println("Veuillez contacter : "+emailadministrateur);
  }
}
```

EL HAMILAOUTI Mahmoud

141

## Exemple

### ■ Ajouter des paramètres

```
//créer l'objet JavaBean Client
Client client1=new Client();
client1.setIdentifiant(identifiant);
client1.setMotdepasse(motdepasse);
//sauvegarder l'objet client dans le contexte de l'application
ServletContext servletContext=getServletContext();
servletContext.setAttribute("client1", client1);
```

EL HAMILAOUI Mahmoud

142

## Traitement des requêtes

### ■ Récupérer des paramètres transmis par le client

- ❑ Si le paramètre n'existe pas, la valeur null est retournée,
- ❑ `getParameter(nom)` permet de récupérer sous la forme d'une chaîne de caractères,

```
...
public void doGet(HttpServletRequest request,
HttpServletResponse
response)throws ServletException, IOException
{ //récupérer l'identifiant/login dans la requête
String identifiant=request.getParameter("identifiant");
...
}
```

- ❑ La méthode `getParameterValues(nom)` permet de récupérer sous la forme d'un tableau de chaînes de caractères un ensemble de valeurs.

EL HAMILAOUI Mahmoud

143

## Traitement des requêtes

- ❑ `getParameterNames()` permet de récupérer sous la forme d'une énumération l'ensemble des noms des paramètres contenus dans la requête.

```
...
public void doGet(HttpServletRequest request,
HttpServletResponse
response)throws ServletException, IOException
{
Enumeration enume=request.getParameterNames();
while(enume.hasMoreElements())
{
String nomParametre=(String)enume.nextElement();
String valeurParametre=request.getParameter(nomParametre);
out.println("parametre : "+nomParametre+" - valeur : "+valeurParametre);
}
...
}
```

- ❑ `getParameterMap()` permet de récupérer sous la forme d'un objet de type `Map` l'ensemble des paramètres contenus dans la requête

EL HAMILAOUI Mahmoud

144

## Gestion des entrées dans plusieurs langues

- Convertir de l'Anglais vers le Japonais

```
String firstNameWrongEncoding = request.getParameter("firstName");
String firstName = new String(firstNameWrongEncoding.getBytes(),
    "Shift_JIS");
```

- Prendre en compte l'Anglais ou le Japonais

```
request.setCharacterEncoding("JISAutoDetect");
String firstName = request.getParameter("firstName");
```

EL HAMILAOUTI Mahmoud

145

## Traitement des requêtes



EL HAMILAOUTI Mahmoud

146

## Traitement des requêtes

```
private void showPreview(HttpServletRequest request,
    PrintWriter out) {
    String headingFont = request.getParameter("headingFont");
    headingFont = replaceIfMissingOrDefault(headingFont, "");
    ...
    String name = request.getParameter("name");
    name = replaceIfMissing(name, "Arnaud");
    String title = request.getParameter("title");
    title = replaceIfMissing(title, "Looser");
    String languages = request.getParameter("languages");
    languages = replaceIfMissing(languages, "<I>None</I>");
    String languageList = makeList(languages);
    String skills = request.getParameter("skills");
    skills = replaceIfMissing(skills, "Pas Assez.");
    ...
}
```

EL HAMILAOUTI Mahmoud

147

## Suivi des sessions

- Une session est une conversation entre le serveur et un client.
- le protocole HTTP est un protocole sans état, ce qui signifie qu'aucune information utilisateur n'est concernée et que le serveur considère chaque demande comme une nouvelle requête.
- Il existe quatre techniques qui peuvent être utilisées pour identifier une session utilisateur.
  - Cookies,
  - Champs cachés,
  - Réécriture d'URL,
  - API de suivi des sessions.

EL HAMILAOUTI Mahmoud

148

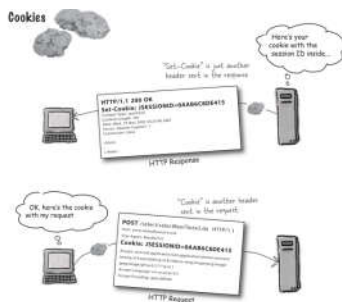
## Cookies

- Un cookie est un élément de données qui peut être envoyé par le serveur à destination du navigateur client.
- A chaque connexion du client, le navigateur transmet également le cookie (ou les cookies) que notre serveur a précédemment dép osé(s)
- L'utilisateur peut configurer son navigateur pour refuser les cookies ou les supprimer directement manuellement.
- Un cookie est composé d'une paire nom/valeur et de plusieurs propriétés pour renseigner le domaine, la durée de vie, la sécurité, etc.

EL HAMILAOUTI Mahmoud

149

## Cookies



EL HAMILAOUTI Mahmoud

150

## Méthodes Cookies

Method Summary	
<code>java.lang.Object</code>	<code>clone()</code> Overrides the standard <code>java.lang.Object.clone</code> method to return a copy of this cookie.
<code>java.lang.String</code>	<code>comment()</code> Returns the comment describing the purpose of this cookie, or null if the cookie has no comment.
<code>java.lang.String</code>	<code>comment()</code> Returns the domain-unique set for this cookie.
<code>int</code>	<code>expires()</code> Returns the maximum age of the cookie, specified in seconds. By default, -1, indicating the cookie will persist until browser shutdown.
<code>java.lang.String</code>	<code>getName()</code> Returns the name of the cookie.
<code>java.lang.String</code>	<code>getPath()</code> Returns the path in the server to which the browser returns this cookie.
<code>boolean</code>	<code>secure()</code> Returns true if the browser is sending cookies only over a secure protocol, or false if the browser can send cookies using any protocol.
<code>java.lang.String</code>	<code>setValue()</code> Returns the value of the cookie.
<code>int</code>	<code>version()</code> Returns the version of the protocol this cookie complies with.
<code>java.lang.String</code>	<code>comment(java.lang.String comment)</code> Specifies a comment that describes a cookie's purpose.
<code>java.lang.String</code>	<code>comment(java.lang.String domain)</code> Specifies the domain within which this cookie should be presented.
<code>int</code>	<code>expires(int maxAge)</code> Sets the maximum age of the cookie in seconds.
<code>java.lang.String</code>	<code>getPath(java.lang.String path)</code> Specifies a path for the cookie to which the client should return the cookie.
<code>boolean</code>	<code>secure(boolean flag)</code> Indicates to the browser whether the cookie should only be sent using a secure protocol, such as HTTPS or SSL.
<code>java.lang.String</code>	<code>setValue(java.lang.String newValue)</code> Replaces a cookie value to a cookie after the cookie is created.
<code>int</code>	<code>version(int ver)</code> Sets the version of the cookie protocol that the cookie complies with.

151

## Cookies

```

public static String buildCookieString(Cookie cookie)
{
    StringBuffer buffer = new StringBuffer();
    int version = cookie.getVersion();
    if (version != -1) {
        buffer.append("$Version=");
        buffer.append(cookie.getVersion());
        buffer.append("; ");
    }
    // cookie name/value
    buffer.append(cookie.getName());
    //buffer.append("=");
    buffer.append(cookie.getValue());
    //buffer.append("; ");
    // cookie path
    String path = cookie.getPath();
    if (path != null) {
        //buffer.append("path=");
        buffer.append(path);
        //buffer.append("; ");
    }
    String cookieHeader = buffer.toString();
    return cookieHeader;
}

```

<http://www.programcreek.com/2014/05/java-servlet-http-cookie/>

## Caractères restreints

- Certains caractères qui ne peuvent pas être utilisés dans le nom et la valeur d'un cookie :

,  
 =  
 (  
 )  
 "  
 ;  
 [  
 ]  
 /  
 ?  
 :  
 @

EL-HAMLAOUI Mahmoud

153



## Exemple

```
public class CookieTest extends HttpServlet {
    public void doPost(HttpServletRequest request,
        HttpServletResponse response) throws IOException, ServletException
    {
        response.setContentType("text/html");
        String name = request.getParameter("username");
        Cookie cookie = new Cookie("username", name);
        //cookie.setMaxAge(24*60*60);
        response.addCookie(cookie);
        PrintWriter out = response.getWriter();
        out.println("<html>\n" + "<head><title>" + title +
            "</title></head>\n" + " </body> ... </body></html>");
        /*RequestDispatcher view =
        request.getRequestDispatcher("cookieresult.jsp");
        view.forward(request, response);*/
    }
}
```

EL-HAMLAOUTI Mahmoud

154

## Exemple

```
public class CheckCookie extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse
        response) throws IOException, ServletException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Cookie[] cookies = request.getCookies();
        if (cookies != null) {
            for (int i = 0; i < cookies.length; i++) {
                Cookie cookie = cookies[i];
                if (cookie.getName().equals("username")) {
                    String userName = cookie.getValue();
                    out.println("Hello" + userName);
                    break;
                }
            }
        }
    }
}
```

EL-HAMLAOUTI Mahmoud

155

## Champs cachés

- Les champs cachés sont les champs de saisie qui ne sont pas affichés sur la page mais sa valeur est envoyée à la servlet comme d'autres champs de saisie.

```
<INPUT TYPE="HIDDEN" NAME="session" VALUE="...">
```

- La valeur du champ peut être récupérée en utilisant `request.getParameter("sessionId")` dans servlet.

EL-HAMLAOUTI Mahmoud

156

## Réécriture d'URL

- Le client ajoute quelques données supplémentaires à la fin de chaque URL qui identifie la session
- Le serveur associe cet identifiant aux données qu'il a stockées sur cette session
  - Par exemple, `http://host/path/file.html;jsessionid=1234`



157

## Réécriture d'URL

- Avantages :
  - Fonctionne même si les cookies sont désactivés ou non pris en charge
- Inconvénients :
  - Beaucoup de traitement fastidieux
  - Doit coder toutes les URL qui se rapportent à votre propre site
  - Les liens provenant d'autres sites et signets peuvent échouer

```
HttpSession session = request.getSession();
out.println("<html><body>");
out.println("<a href=\"\" + response.encodeURL(\"/ServletTest.do\") + \"\">click here</a>");
out.println("</body></html>");
```

### ■ Avec SendRedirect

```
...
response.sendRedirectURL("/coffee.do")
....
```

EL-HAMLAOUI Mahmoud

158

## API de suivi des sessions

- Le suivi des sessions implique les étapes suivantes :
  - Obtention de l'objet de session associé (HttpSession) en utilisant la méthode `request.getSession()`,
  - Obtention de la valeur spécifique de l'objet session, en appelant la méthode `getAttribute (String)` sur l'objet HttpSession.
  - Stockage des informations dans un appel de session en appelant la méthode `setAttribute (key, object)` sur un objet de session,
  - Suppression des données de session en appelant la méthode `removeAttribute (key)`
  - Destruction de la session en appelant la méthode `invalidate ()` sur l'objet session. Ceci permet de se déconnecter de l'utilisateur connecté.

EL-HAMLAOUI Mahmoud

159

## Méthodes HttpSession

[illegible]

EL. HAMLAQUI Mahmoud

160

## Destruction d'une session

- Une session peut être détruite de 3 façons :
  - Expiration

```
</servlet>
<session-config>
  <session-timeout>15</session-timeout> 
```

- Appel de la méthode `invalidate()` sur l'objet `session`

```
session.invalidate();
```

- ❑ Application tombe en panne elle n'est pas déployée.

- Il est possible de définir un délai d'attente d'une session spécifique

```
session.setMaxInactiveInterval(20*60); // en secondes
```

EL. HAMLAOUI Mahmoud

161

## Méthodes HttpSession

[illegible]

EL HANLAOU Mahmoud

162

## Enchaînement

- La plupart du temps, la programmation de Servlets consiste à récupérer des données en paramètre, à traiter ces données et à renvoyer une réponse adaptée au client.
  - `sendRedirect(String)` (Client)
  - `RequestDispatcher` (Serveur)

EL HAMILAOUI Mahmoud

163

## La méthode `sendRedirect`

- La méthode `sendRedirect(String)`, effectue une redirection de la requête en désactivant complètement la Servlet.
  - `response.sendRedirect("http://www.ensias.ma");`
- Elle effectue le même traitement que les deux instructions :
  - `response.setStatus(HttpServletResponse.SC_MOVED_TEMPORARILY);`
    - Status code : 302 → Une nouvelle URL doit être interprétée comme un remplacement temporaire, et non permanent
  - `response.setHeader("Location", "http://www.monserveur.com/autreUrl.html");`
- Possibilité d'utiliser des URLs relatives.
- Jamais un `sendRedirect` après écriture sur le flux de sortie.
  - `IllegalStateException`

EL HAMILAOUI Mahmoud

164

## L'interface `RequestDispatcher`

- Délégation de la requête à une autre servlet
  - effectuer un traitement continu vers une autre ressource Web,
  - Le contrôle est passé à la seconde ressource.
- Inclusion de contenu
  - inclure le contenu d'une autre ressource Web dans la réponse HTTP,
  - la première Servlet conserve le contrôle.
- Possibilité d'utiliser les attributs pour échanger les informations.

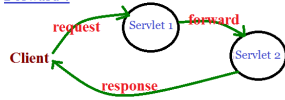
EL HAMILAOUI Mahmoud

165

## L'interface RequestDispatcher : Délégation

- La délégation consiste à transmettre la requête HTTP courante à une autre ressource Web qui sera chargée de son traitement et qui peut également transmettre la requête à une autre ressource.

Forward :



```

RequestDispatcher view =
    request.getRequestDispatcher("/result.jsp");
/*RequestDispatcher view =
    getServletContext().getRequestDispatcher("/result.jsp");*/
view.forward(request,response);
out.println("Ce message ne sera pas affiché");
  
```

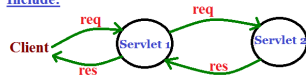
EL HAMILAOUTI Mahmoud

166

## L'interface RequestDispatcher : Inclusion

- L'inclusion consiste à incorporer le contenu d'une autre ressource Web dans la réponse HTTP qui sera envoyée au client.

Include:



```

RequestDispatcher view =
    request.getRequestDispatcher("result.jsp");
view.include(request,response);
out.println("Ce message sera affiché");
  
```

EL HAMILAOUTI Mahmoud

167

## Filtres

- Les filtres permettent de traiter :
  - les requêtes qui viennent des clients avant qu'elles ne soient traitées par les Servlets,
  - les réponses venant des Servlets avant qu'elles ne soient envoyées aux clients.
- Il est possible par exemple de :
  - décrypter des requêtes envoyées aux Servlets, traiter les données avec les Servlets et crypter les réponses pour les clients,
  - gérer l'authentification des clients,
  - convertir des formats d'images, appliquer des transformations XSLT sur des données XML...

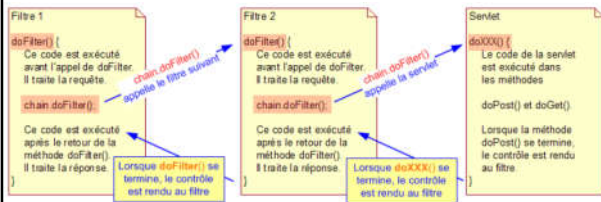
```

log("-----")
log("Route Note client : "+ request.getRemoteHost());
log("Adresse IP du client : "+ request.getRemoteAddr());
log("Paramètres envoyés : "+ request.getQueryString());
log("Numéro de port du client : "+ request.getRemotePort());
log("Nom du client : "+ request.getRemoteUser());
log("-----")
  
```

EL HAMILAOUTI Mahmoud

168

## Filtres



EL HAMILAOUTI Mahmoud

169

## Filtres

- Créer une classe implémentant l'interface javax.servlet.Filter
- Définir les méthodes

### Method Summary

void	destroy()	Called by the web container to indicate to a filter that it is being taken out of service.
void	doFilter(ServletRequest request, ServletResponse response, FilterChain chain)	The doFilter method of the Filter is called by the container each time a request/response pair is passed through the chain due to a client request for a resource at the end of the chain.
void	init(FilterConfig filterConfig)	Called by the web container to indicate to a filter that it is being placed into service.

EL HAMILAOUTI Mahmoud

170

## Filtres

- DD :

```
...
<!-- definition du filtre -->
<filter>
  <filter-name>nom du filtre pour l'application web</filter-name>
  <filter-class>nom qualifié de la classe filtre </filter-class>
</filter>
...
```

```
...
<filter-mapping>
  <filter-name>Filtre 1</filter-name>
  <servlet-name>servletaauthentification</servlet-name>
</filter-mapping>
<filter-mapping>
  <filter-name>Filtre 2</filter-name>
  <servlet-name>servletaauthentification</servlet-name>
</filter-mapping>
...
```

EL HAMILAOUTI Mahmoud

171

## Filtres

```
public class FiltreJournalisation implements Filter{
    private FilterConfig filterConfig=null;
    public void init(FilterConfig filterConfig)
    {
        this.filterConfig=filterConfig;
        System.out.println("Initialisation du filtre :
        "+this.filterConfig.getFilterName());
    }
    public void doFilter(ServletRequest request,
    ServletResponse response, FilterChain chain)
    {
        //traitement de la requête avant la Servlet
        System.out.println("----- REQUETE -----");
        System.out.println("Encodage : "+request.getCharacterEncoding());
        System.out.println("Type de contenu : "+request.getContentType());
        System.out.println("Taille du contenu :
        "+request.getContentLength());
        System.out.println("Nom machine distante :
        "+request.getRemoteHost());
        System.out.println("Adresse IP machine distante :
        "+request.getRemoteAddr());
        try
        {
            chain.doFilter(request, response);
        }
        catch(Exception e)
        {System.out.println("Erreur dans le filtre FiltreJournalisation");}
    }
}
```

EL HAMILAOUI Mahmoud

172

## Filtres

```
//traitement de la requête avant la Servlet
System.out.println("----- REPONSE -----");
System.out.println("Encodage : "+response.getCharacterEncoding());
System.out.println("Type de contenu : "+response.getContentType());
//exemple d'écriture dans le fichier de log de l'application
ServletContext context=this.filterConfig.getServletContext();
context.log("Encodage de la réponse :
"+response.getCharacterEncoding());
}

public void destroy()
{
    System.out.println("Destruction du filtre :
    "+this.filterConfig.getFilterName());
}
}
```

```
----- REQUETE -----
Encodage : null
Type de contenu : null
Taille du contenu : -1
Nom machine distante : 127.0.0.1
Adresse IP machine distante : 127.0.0.1
----- REPONSE -----
Encodage : ISO-8859-1
Type de contenu : null
```

EL HAMILAOUI Mahmoud

173

## Filtres

```
@WebFilter (filterName="FiltreJournalisation", urlPatterns={"*"},
initParams={@WebInitParam(name="parametere", value="valeurParam")})

public class FiltreJournalisation implements Filter{
    private FilterConfig filterConfig=null;
    public void init(FilterConfig filterConfig) {
        this.filterConfig=filterConfig;
        System.out.println("Initialisation du filtre :
        "+this.filterConfig.getFilterName());
        System.out.println("Lecture d'un param:
        "+this.filterConfig.getInitParameter("parametere"));
    }
    public void doFilter(ServletRequest request,
    ServletResponse response, FilterChain chain)
    {
        System.out.println("----- REQUETE -----");
        System.out.println("Encodage : "+request.getCharacterEncoding());
        System.out.println("Type de contenu : "+request.getContentType());
        System.out.println("Taille du contenu :
        "+request.getContentLength());
        System.out.println("Nom machine distante :
        "+request.getRemoteHost());
        System.out.println("Adresse IP machine distante :
        "+request.getRemoteAddr());
        try
        {
            chain.doFilter(request, response);
        }
        catch(Exception e)
        {System.out.println("Erreur dans le filtre FiltreJournalisation");}
    }
}
```

EL HAMILAOUI Mahmoud

174

## Génération dynamique des servlets

- La spécification Servlet 3.0 permet d'ajouter des Servlets, filtres et écouteurs durant l'exécution du programme.

```
@WebListener
public class MonListener implements ServletContextListener {
    public void contextInitialized(ServletContextEvent sce)
    {
        System.out.println("Listener");
        ServletContext sc=sce.getServletContext();
        ServletRegistration sr=sc.addServlet("MaServletDynamique",
            "MaServletDynamique");
        sr.setInitParameter("unparametre", "Valeur du paramètre");
        sr.addMapping("/maservletdynamique/*");
    }
    public void contextDestroyed(ServletContextEvent sce)
    {
    }
}
```

EL HAMILAOUI Mahmoud

175

## Génération dynamique des servlets

- Listener est chargé au démarrage de l'application et génère la Servlet à la volée.

```
public class MaServletDynamique extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    public void doGet(HttpServletRequest request, HttpServletResponse
response)throws
        ServletException, IOException
    {
        response.setContentType("text/html");
        ...
        String paramInitialise =
getServletConfig().getInitParameter("unparametre");
        out.println("Paramètre :"+ paramInitialise);
        ...
    }
}
```

EL HAMILAOUI Mahmoud

176

## Utilisation des web-fragments

EL HAMILAOUI Mahmoud

177



## Injection des ressources JNDI

EL HAMILAOUI Mahmoud

178

---

---

---

---

---

---

---

---

## Gestion de la sécurité

EL HAMILAOUI Mahmoud

179

---

---

---

---

---

---

---

---

## Conclusion

- With servlets, it is easy to
  - ❑ Read form data
  - ❑ Read HTTP request headers
  - ❑ Set HTTP status codes and response headers
  - ❑ Use cookies and session tracking
  - ❑ Share data among servlets
  - ❑ Remember data between requests
  - ❑ Get fun, high-paying jobs
- But, it sure is a pain to
  - ❑ Use those println statements to generate HTML
  - ❑ Maintain that HTML

EL HAMILAOUI Mahmoud

180

---

---

---

---

---

---

---

---

## Conclusion

- Avec les servlets, il est facile de :
  - ❑ Lire les données du formulaire
  - ❑ Lire les en-têtes de requête HTTP
  - ❑ Définir les codes d'état HTTP et les en-têtes de réponse
  - ❑ Utiliser les cookies et le suivi des sessions
  - ❑ Partager les données entre les servlets
  - ❑ Garder les données entre les demandes
- Mais, il est difficile de :
  - ❑ Utiliser ces instructions `println` pour générer du HTML

EL HAMILAOUTI Mahmoud

181

## JDBC



EL HAMILAOUTI Mahmoud

182