Présentation – Introduction au Jee

Max Devulder

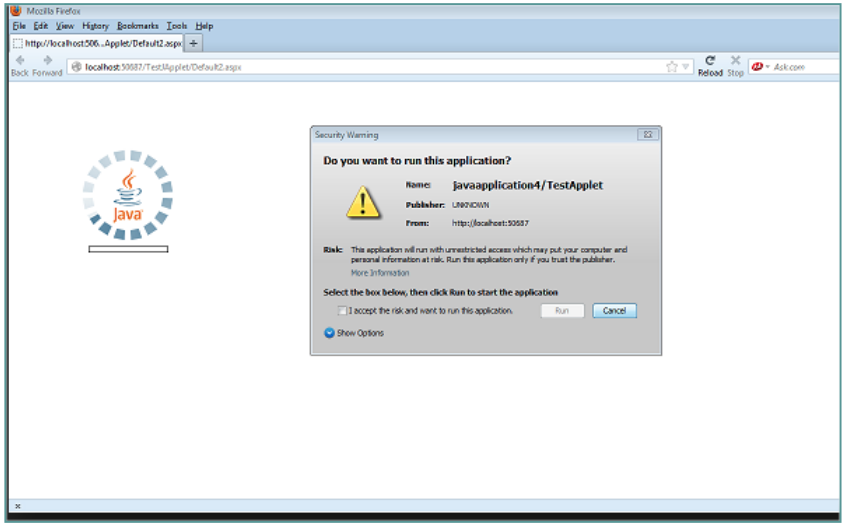
Découverte Java EE

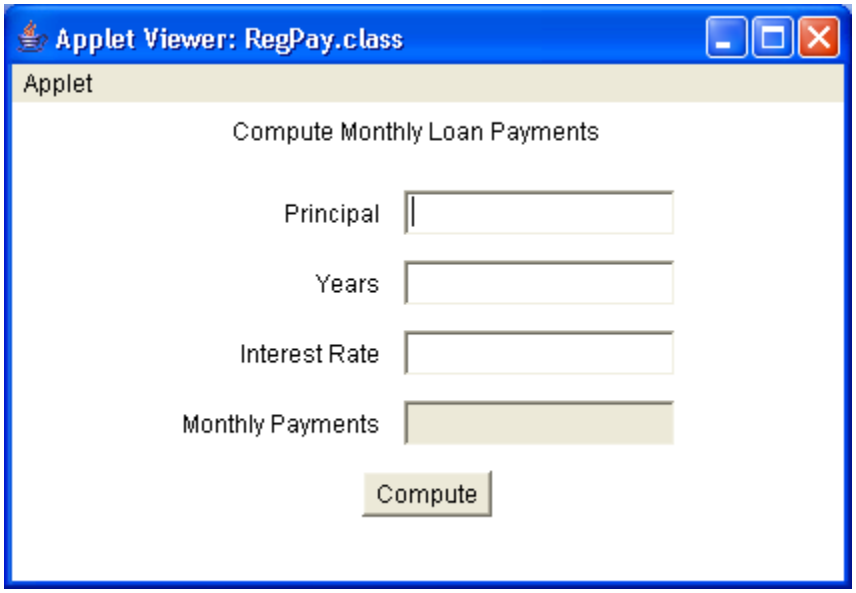
- Norme proposée par la société Sun ™, portée par un consortium de sociétés internationales
- Visant à définir un standard de développement d'applications d'entreprises
- Basé sur le langage Java

- Racheté par **Oracle**, le 20 avril 2009 pour 7,4 milliards de dollars.

- Procès contre Google (Android) en 2010 pour l'utilisation de 37 API (11k LOC) pour 8,8 milliards de dollars.
  Toujours en cours.

- Ne pas confondre avec les Applets et le SWING.



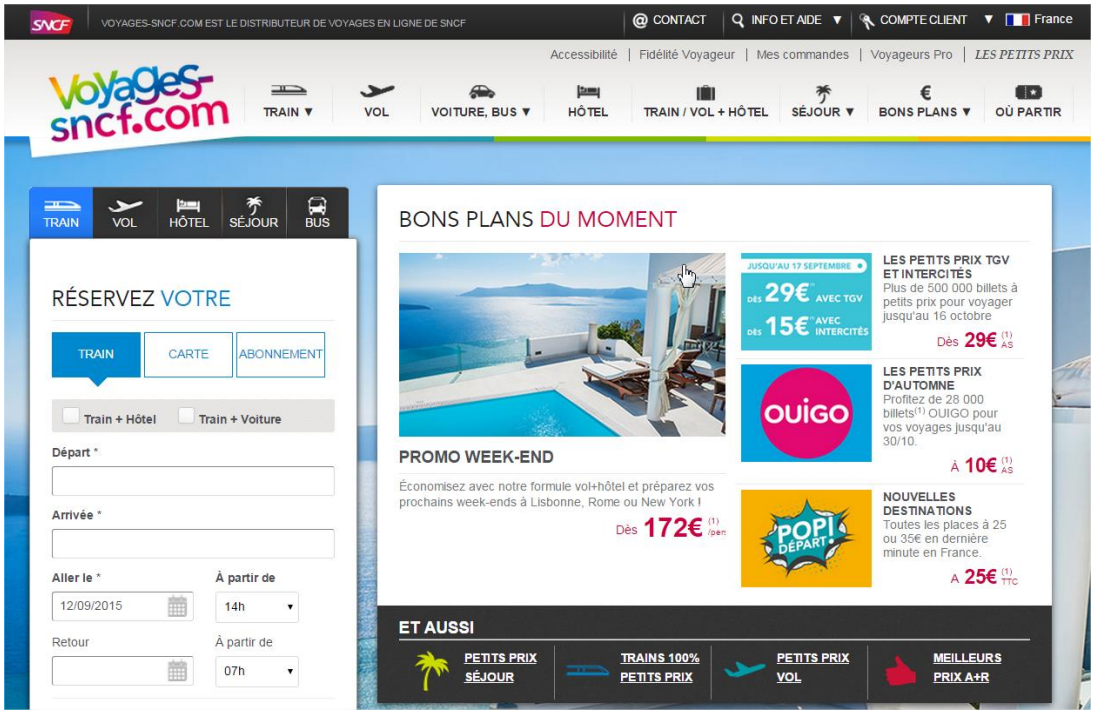


Une application d'entreprise ?

Page web dynamiques

Authentification

Outils de supervision



Envoi mail

Flux

Connexion BDD

Découverte Java EE

Java Entreprise Edition comprend

- Les spécifications du serveur d'application
- Des services, au travers de composants **(API)**

Les composants

Les composants web : JSP (Java Server Pages) DEPRECATED

Les servlets (Contrôleur)

Les composants métier : EJB (Enterprise Java Beans)

Les services

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

JavaMail est une API permettant l'envoi de courrier électronique.

JMS (Java Message Service) fournit des fonctionnalités de communication asynchrone entre applications

Les servlet

Bon ok … j'ai pas tout compris, concrètement ?

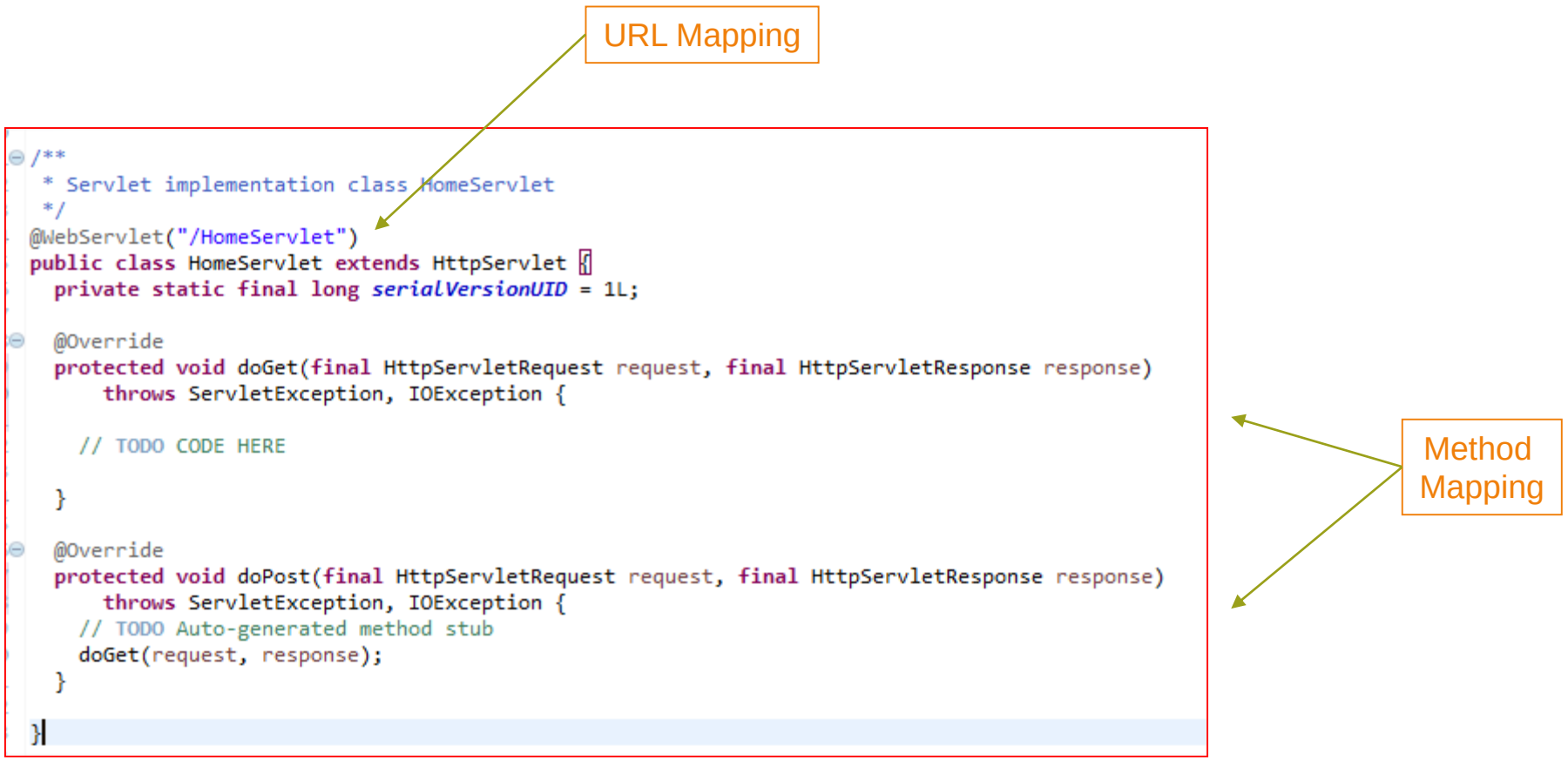
- Focus sur la **Servlet :**
  Composant natif fourni par l'API JEE.
  Ecoute le protocole de communication HTTP
  Principe : Etendre HttpServlet et surcharger de ses méthodes :

  - **init() :** appelée une seule fois par le serveur lors du new(), c'est bien le serveur qui va instancier notre servlet.
  - **doGet() :** appelée automatiquement lors d'une requête de type GET.
  - **doPost() :** appelée automatiquement lors d'une requête de type POST.

Les servlet

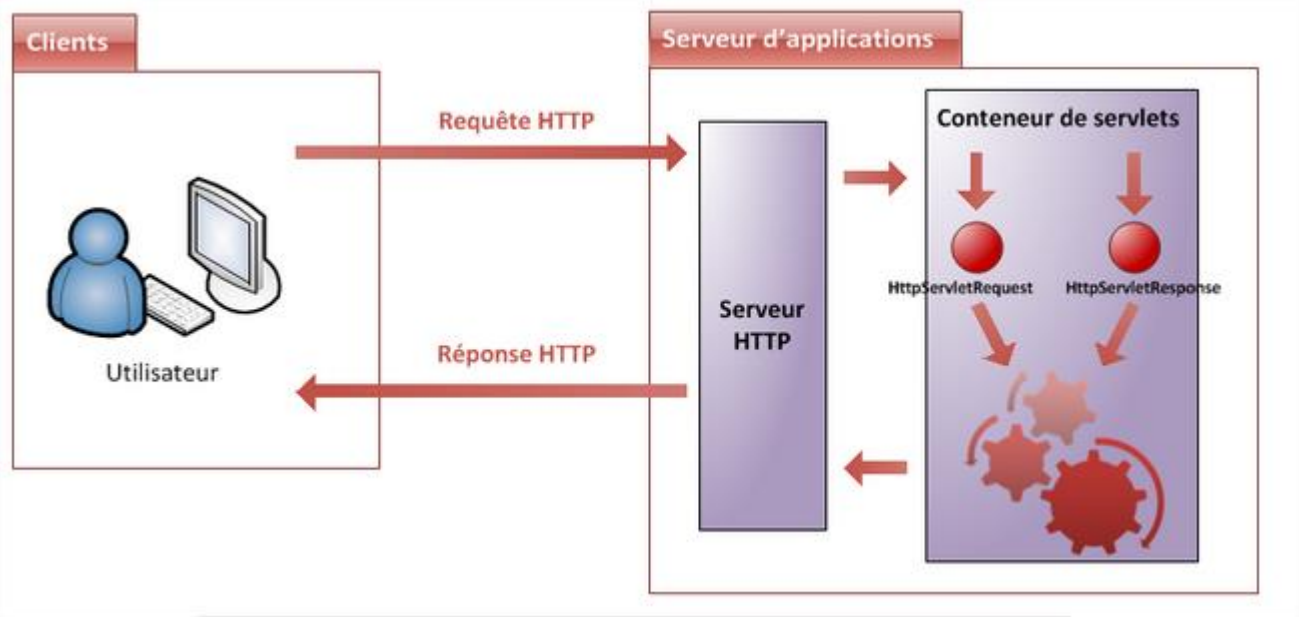
```
import java.io.* ;
import java.text.* ;
import java.util.* ;
import javax.servlet.* ;
import javax.servlet.http.* ;

public Bonjour extends HttpServlet {

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

    response.setContentType("text/html") ;
    PrintWriter out = response.getWriter() ;
    out.println("<html>") ;
    out.println("<head>") ;
    out.println("<title>Bonjour le monde !</title>") ;
    out.println("</head>") ;
    out.println("<body>") ;
    out.println("<h1>Bonjour le monde !</h1>") ;
    out.println("</body>") ;
    out.println("</html>") ;
   }

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

Les servlet


URL Mapping
/**
* Servlet implementation class HomeServlet
*/
@WebServlet("/HomeServlet")
public class HomeServlet extends HttpServlet {
private static final long serialVersionUID = 1L;
@Override
protected void doGet(final HttpServletRequest request, final HttpServletResponse response)
throws ServletException, IOException {
// TODO CODE HERE
}
@Override
protected void doPost(final HttpServletRequest request, final HttpServletResponse response)
throws ServletException, IOException {
// TODO Auto-generated method stub
doGet(request, response);
}
}
Method Mapping

- `HttpServletRequest` : Contient la requête HTTP, et donne accès à toutes ses informations, telles que les en-têtes (*headers*) et le corps de la requête.

- `HttpServletResponse` : Initialise la réponse HTTP qui sera renvoyée au client, et permet de la personnaliser, en initialisant par exemple les en-têtes et le corps (nous verrons comment par la suite).



Les JSP

Ok pour les servlet, maintenant les JSP

JSP : Java Server Page

- Assez peut pratique d'utiliser
- Page html dans laquelle il est possible d'écrire du JAVA

    - En PHP : <?php echo "Hello"; ?>
    - Avec un JSP : <% System.out.println("Hello"); %>

Un première JSP

Simple

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta   http-equiv="Content-Type"   content="text/html; charset=UTF-8">
        <title>Une première JSP</title>
    </head>
    <body>
        <h1>Bonjour le monde !</h1>
    </body>
</html>
```

Un première JSP

Dynamique

```
<p>Trois fois <br>
<%  for (int i =  0 ; i <  3 ; i++) { %>
        Bonjour le monde ! <br>
<% } %>
sans effort !</p>
```

```
Trois fois
Bonjour le monde !
Bonjour le monde !
Bonjour le monde !
sans effort !
```

La Servlet associée :

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

      final Marin myMarin = new Marin();
      myMarin.setNom("Bart");
      myMarin.setPrenom("Jean");
      request.getSession().setAttribute("marin", myMarin);

      response.sendRedirect("bonjour.jsp");

}
```

Pour rediriger vers ma jsp

Afficher des objets

- Taglig : <jsp:useBean
- Evaluation d'une expression : ${}

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC  "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<jsp:useBean id="marin" beanName="marin" scope="session"
             type="org.paumard.cours.model.Marin"/>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <p>Nom = ${marin.nom}</p>
    </body>
</html>
```