

# JEE : introduction générale

**Achref El Mouelhi**

Docteur de l'université d'Aix-Marseille  
Chercheur en Programmation par contrainte (IA)  
Ingénieur en Génie logiciel

`elmouelhi.achref@gmail.com`

# Plan

- 1 Introduction
- 2 Fonctionnement
- 3 Installation et configuration
- 4 Création d'un premier projet web (avec Eclipse)
- 5 Structure d'un projet JEE réalisé sous Eclipse
- 6 Un premier Hello world

# Introduction

## JEE, c'est quoi ?

- JEE : Java Enterprise Edition (anciennement J2EE)
- Une technologie qui facilite le développement d'applications d'entreprise distribuées
- Développée par **Sun** puis **Oracle**
- JEE = JSE + plusieurs autres API
  - JSE : Java Standard Edition (anciennement J2SE)
  - API : Application Programming Interface

# Introduction

## La plateforme JSE est composée de plusieurs API

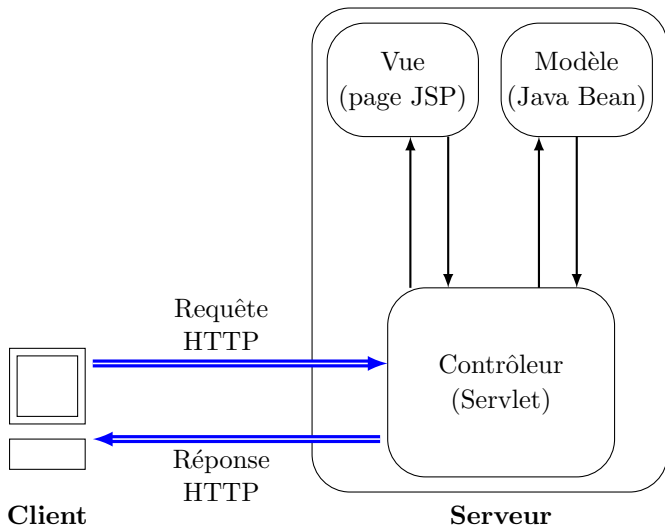
- `Swing` : une bibliothèque graphique
- `JAXP` : Java API for XML Processing
- `JDBC` : Java Database Connectivity
- ...

# Introduction

## Autres composants de JEE

- JSP : Java Server Pages (pour les vues)
- Servlet : les contrôleurs du modèle MVC
- JPA : Java Persistence API
- EJB : Enterprise Java Beans
- ...

# JEE : schématisation



# JEE : explication

## Déroulement

- Le client et le serveur doivent se parler via HTTP (requête - réponse)
- Lorsque l'utilisateur saisit l'adresse d'une page de notre site, cette dernière sera envoyée sous forme de requête HTTP au contrôleur
- Le contrôleur demande au modèle de lui fournir certaines données
- Ensuite il renvoie ces données à la vue pour qu'elle construise la page HTML
- Enfin le client reçoit la réponse sous forme de page JSP

# Installation et configuration

## Deux choses

- IDE : Environnement de Développement Intégré
- Un serveur HTTP



# Installation et configuration

## Environnement de Développement Intégré


- **Eclipse**
- Netbeans
- JDeveloper
- ...


# Installation et configuration


Eclipse : choisir Eclipse IDE for Enterprise Java Developers

```
https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2019-06/R/eclipse-inst-win64.exe
```


# Installation et configuration

 by Oomph


type filter text 




**Eclipse IDE for Java Developers**  
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration




**Eclipse IDE for Enterprise Java Developers**  
Tools for Java developers creating Enterprise Java and Web applications, including a Java IDE, tools for Enterprise Java, JPA, JSF, Mylyn, Maven, Git and...



**Eclipse IDE for C/C++ Developers**  
An IDE for C/C++ developers with Mylyn Integration.





**Eclipse IDE for JavaScript and Web Developers**  
The essential tools for any JavaScript developer, including JavaScript, HTML, CSS, XML languages support, Git client, and Mylyn.




**Eclipse IDE for PHP Developers**  
The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.


# Installation et configuration

 by Oomph


type filter text 




**Eclipse IDE for Java Developers**  
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration




**Eclipse IDE for Enterprise Java Developers**  
Tools for Java developers creating Enterprise Java and Web applications, including a Java IDE, tools for Enterprise Java, JPA, JSF, Mylyn, Maven, Git and...



**Eclipse IDE for C/C++ Developers**  
An IDE for C/C++ developers with Mylyn integration.



**Eclipse IDE for JavaScript and Web Developers**  
The essential tools for any JavaScript developer, including JavaScript, HTML, CSS, XML languages support, Git client, and Mylyn.



**Eclipse IDE for PHP Developers**  
The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.

# Installation et configuration

## Modifier l'encodage par défaut

- Aller dans le menu `Window` et cliquer sur `Preferences`
- Dans la zone de recherche, écrire `encoding`
- Pour toutes les sections, mettre `UTF-8` à la place de l'encodage par défaut (généralement `Cp1252` ou `ISO-8859-1`)
- Ensuite appliquer et fermer

# Installation et configuration

## Serveur HTTP

- **Apache Tomcat**
- WebLogic Server (Serveur payant d'oracle utilisé par JDeveloper)
- JBoss
- GlassFish (Open Source de Oracle)
- ...

## Pourquoi Apache Tomcat ?

- gratuit
- multi-plateforme
- léger

# Installation et configuration

## Mise en place d'Apache Tomcat

- Pour télécharger, aller sur `https://tomcat.apache.org/download-90.cgi`
- Aller dans la section `Core`
- Cliquer sur `32-bit/64-bit Windows Service Installer`
- Attendre la fin du téléchargement puis lancer l'installation
- Installer Apache Tomcat à la racine de votre disque dur (`C:` pour Windows) et vérifier que le nom du dossier destination ne contient pas d'espace
- Décocher la case de la dernière fenêtre d'installation proposant de démarrer Apache Tomcat

# Création d'un premier projet web (Eclipse)

## Les étapes

- Aller dans le menu `File`, sous-menu `New` et cliquer sur `Project` (ou bien directement via le raccourci `CTRL + N`)
- Ensuite choisir `Dynamic Web Project` situé dans le répertoire `Web`
- Saisir le nom du projet dans `Project name`:
- Ensuite cliquer sur le bouton `New Runtime`, choisir la dernière version d'Apache Tomcat (ici `v9.0`) et cocher la case juste en-dessous `create a new local server`
- Cliquer sur `next` et préciser le répertoire d'installation de Tomcat
- Cliquer sur `Finish` ensuite deux fois sur `Next`, puis cocher la case `Generate web.xml deployment descriptor`



# Création d'un premier projet web (avec Eclipse)

## Les étapes

- Faire un clic droit sur le projet et choisir `Run` As ensuite `Run on Server`
- Sélectionner le serveur Tomcat et cocher la case `Always use this server when running this project`
- Cliquer sur `Next` et vérifier que notre projet figure dans la liste `Configured`
- Et enfin valider en cliquant sur `Finish`

# Structure d'un projet JEE réalisé sous Eclipse

## Structure légèrement différente d'une structure JEE standard

- Le nom de notre projet définit la racine de l'arborescence
- La racine contient principalement trois répertoires `src`, `build`, `WebContent`
  - `src` contient les fichiers `.java`
  - `build` contient les fichiers `.class` générés à partir des `.java` de `src`
  - `WebContent` (répertoire propre à eclipse) comporte deux dossiers `WEB-INF` et `META-INF`
    - `WEB-INF` a un dossier `lib` (pour les `.jar`) et un fichier `web.xml`. Il peut également contenir les fichiers relatifs aux vues (`.css`, `.js`, `.jsp`, `.html...`).
    - `META-INF` contient les méta-données du projet.

# Un premier Hello world

## Création d'une page HTML

- Clic droit sur le projet, aller dans le menu `new` et choisir `HTML File`
- Placer le fichier directement dans `WebContent` (pas dans `WEB-INF`)
- Nommer le fichier `index.html` et générer la page en cliquant sur `Finish`
- Mettre `Hello world` entre les balises `<body>`
- Démarrer le serveur
- Aller à l'adresse  
`http://localhost:8080/nomProjet/index.html`