

Jakarta EE : introduction

Achref El Mouelhi

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

`elmouelhi.achref@gmail.com`



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

Jakarta EE

JEE

- 1999 : inclus dans **Java 2** sous le nom **J2EE** pour **Java 2 Platform Enterprise Edition**
- 2006 : renommé depuis **Java 5** en **JEE** pour **Java Platform Enterprise Edition**
- 2017 : **Oracle** confie le projet à **Eclipse Foundation** qui décide de le renommer **Jakarta EE**

JEE

- Plateforme facilitant 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**

JEE

- Plateforme facilitant 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**

JSE est désormais appelé **Java SE (Standard Edition)**.

Quelques API de la Java SE

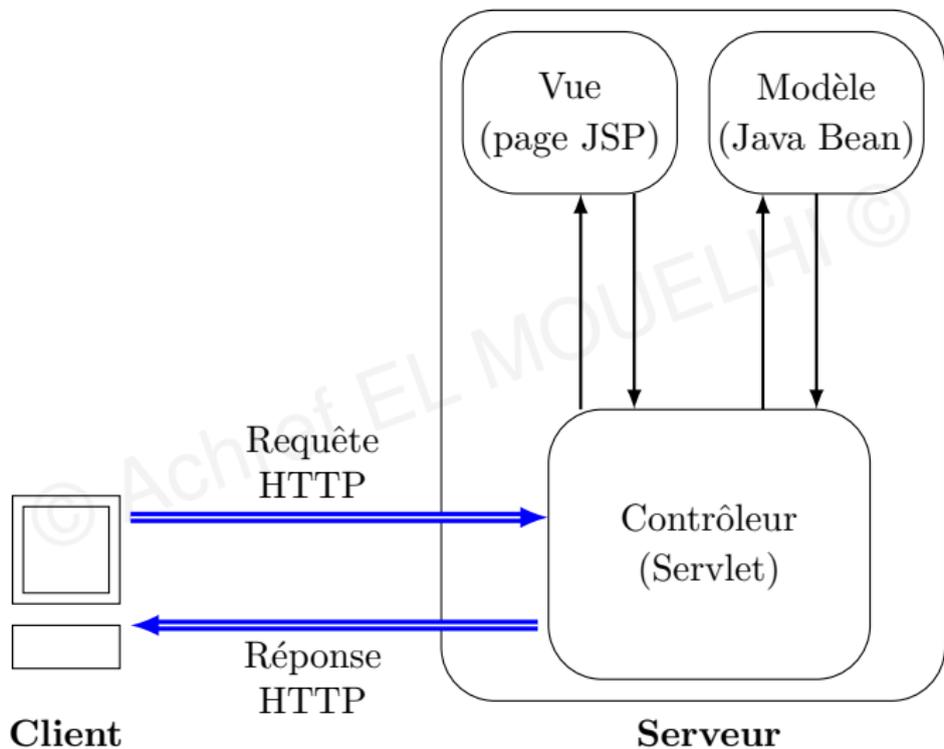
- **Swing** : une bibliothèque pour les applications de bureau
- **JAXP** : **J**ava **A**PI for **X**ML **P**rocessing
- **JDBC** : **J**ava **D**ata**B**ase **C**onnectivity
- ...

Jakarta EE

Quelques composants de la **Jakarta EE**

- **Servlet** : le contrôleur du modèle **MVC**
- **JSP** : **J**ava **S**erver **P**ages (pour les vues dans le Modèle **MVC**)
- **JSTL** : **J**ava **S**tandard **T**ag **L**ibrary (pour les vues dans le Modèle **MVC**)
- **JPA** : **J**akarta **P**ersistence **A**PI
- **EJB** : Enterprise Java Beans
- ...

Jakarta EE



Dérroulement

- L'échange entre le client et le serveur s'effectue via le modèle **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 **HTML** construite par la vue (**JSP**).

Architecture MVC (contexte)

● **Modèle : JavaBean / POJO / Services**

- **JavaBean** : classe **Java** respectant une convention stricte \Rightarrow constructeur vide, attributs privés, getters/setters publics, sérialisable. Utilisé pour transporter des données entre couches (formulaire utilisateur).
- **POJO (Plain Old Java Object)** : classe **Java** ordinaire sans dépendance spécifique (ni interface ni annotation).
- **Service** : composant métier (souvent une classe **Java**) qui contient la logique de traitement et interagit avec les **DAO (Data Access Objects)**. Il ne contient pas de code de présentation.

● **Contrôleur : Servlet** \Rightarrow reçoit la requête **HTTP**, appelle le service, place les données dans la requête (via `setAttribute`) et redirige vers la vue.● **Vue : JSP** \Rightarrow responsable uniquement de l'affichage (**HTML + EL + JSTL**). Éviter le code **Java** direct (scriptlets).

De quoi on a besoin

- **JDK** : **J**ava **D**evelopment **K**it
- **IDE** : **I**ntegrated **D**evelopment **E**nvironment (Environnement de Développement Intégré)
- Un serveur **HTTP**

Jakarta EE

JDK 17 : téléchargement

<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>

JDK 20 : téléchargement

<https://www.oracle.com/java/technologies/javase/jdk21-archive-downloads.html>

JDK 25 : téléchargement

<https://www.oracle.com/fr/java/technologies/downloads/#jdk25-windows>

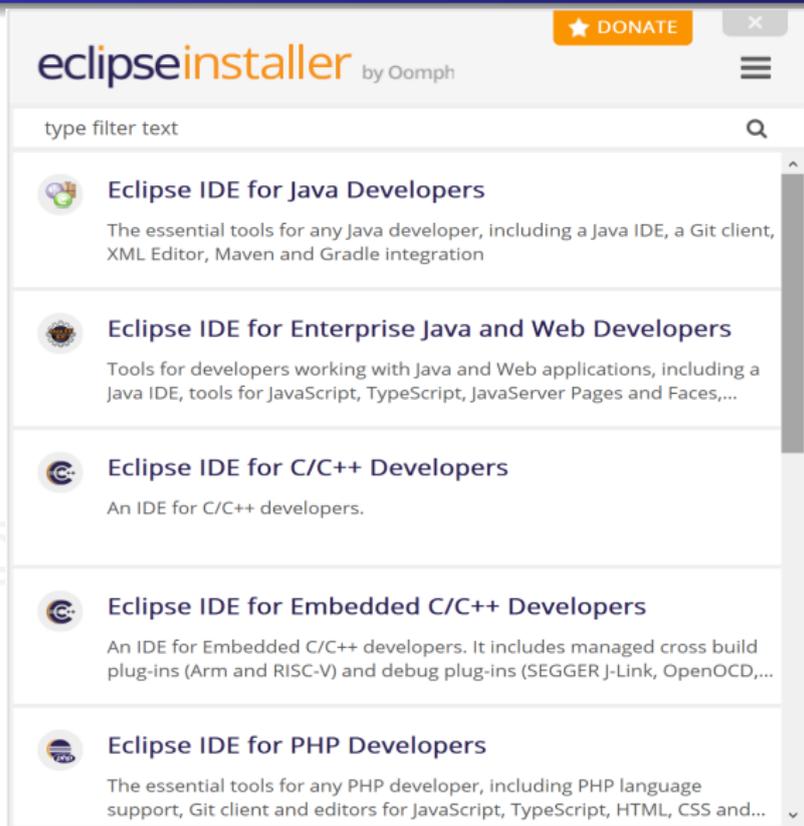
Environnement de Développement Intégré

- **Eclipse**
- Netbeans
- JDeveloper
- IntelliJ IDEA
- JBuilder
- JCreator...
- ...

Eclipse : téléchargement

<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2023-09/R/eclipse-inst-jre-win64.exe>

Jakarta EE

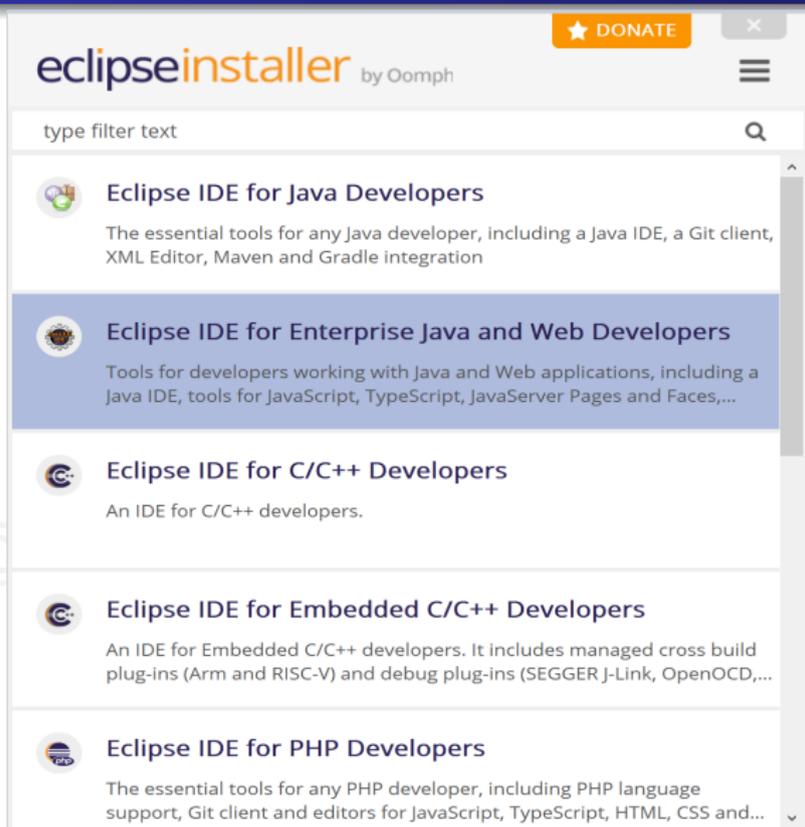


The screenshot shows the Eclipse Installer application window. At the top, the title bar reads "eclipseinstaller by Oomph". To the right of the title bar is an orange "DONATE" button with a star icon, a close button (X), and a hamburger menu icon. Below the title bar is a search bar with the placeholder text "type filter text" and a magnifying glass icon. The main content area displays a list of five Eclipse IDE packages, each with a circular icon, a title, and a brief description:

- Eclipse IDE for Java Developers**
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration
- Eclipse IDE for Enterprise Java and Web Developers**
Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces,...
- Eclipse IDE for C/C++ Developers**
An IDE for C/C++ developers.
- Eclipse IDE for Embedded C/C++ Developers**
An IDE for Embedded C/C++ developers. It includes managed cross build plug-ins (Arm and RISC-V) and debug plug-ins (SEGGER J-Link, OpenOCD,...
- Eclipse IDE for PHP Developers**
The essential tools for any PHP developer, including PHP language support, Git client and editors for JavaScript, TypeScript, HTML, CSS and...

At the bottom right of the application window, there are several navigation icons: a back arrow, a forward arrow, a search icon, and a refresh icon.

Jakarta EE



The screenshot shows the Eclipse Installer website interface. At the top, there is a header with the text "eclipseinstaller by Oomph" and a "DONATE" button. Below the header is a search bar with the placeholder text "type filter text". The main content area displays a list of five IDE options, each with a circular icon and a brief description. The second option, "Eclipse IDE for Enterprise Java and Web Developers", is highlighted with a blue background. The list includes options for Java Developers, Enterprise Java and Web Developers, C/C++ Developers, Embedded C/C++ Developers, and PHP Developers.

eclipseinstaller 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, Maven and Gradle integration
- Eclipse IDE for Enterprise Java and Web Developers**
Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces,...
- Eclipse IDE for C/C++ Developers**
An IDE for C/C++ developers.
- Eclipse IDE for Embedded C/C++ Developers**
An IDE for Embedded C/C++ developers. It includes managed cross build plug-ins (Arm and RISC-V) and debug plug-ins (SEGGER J-Link, OpenOCD,...
- Eclipse IDE for PHP Developers**
The essential tools for any PHP developer, including PHP language support, Git client and editors for JavaScript, TypeScript, HTML, CSS and...

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

Quelques raccourcis **Eclipse**

- `Ctrl` + `Shift` + `:` : commenter/décommenter le code
- `Ctrl` + `Shift` + `f` : formater le code
- `Ctrl` + `Alt` + `↓` ou `Ctrl` + `Alt` + `↑` : dupliquer la ligne sélectionnée
- `Ctrl` + `Shift` + `o` : gérer les importer
- `Ctrl` + `Alt` + `l` : afficher la liste des raccourcis
- `Alt` + `Shift` + `R` : faire une sélection multiple
- `Shift` + `K` : aller à l'occurrence suivante
- `Ctrl` + `Shift` + `K` : aller à l'occurrence précédente

Serveur HTTP

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

© Achref EL MOU...

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
- ...

Jakarta EE

Apache Tomcat

- Conteneur de Servlets
- Serveur d'applications léger qui implémente les spécifications **Jakarta Servlet** et **JavaServer Pages (JSP)**
- Gère les requêtes **HTTP**
- ...

Mise en place d'Apache Tomcat

- Pour télécharger, aller sur `https://tomcat.apache.org/download-11.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'espaces ni de caractères spéciaux (voir slide suivante)
- Décocher la case de la dernière fenêtre d'installation proposant de démarrer **Apache Tomcat**

Bonnes pratiques d'installation d'Apache Tomcat : Emplacement recommandé

- Installer Tomcat à **la racine du disque** (comme `C:\tomcat` ou `D:\apache-tomcat-10`)
- Éviter les dossiers protégés comme `C:\Program Files`
- Cela permet
 - des **chemins plus courts** (moins d'erreurs liées à la longueur)
 - des **droits d'accès simplifiés**
 - une **exécution facilitée des scripts** (`startup.bat`, `shutdown.bat...`)

Jakarta EE

Bonnes pratiques d'installation d'Apache Tomcat : nom du dossier

- Utiliser uniquement des **caractères alphanumériques**
- **Éviter :**
 - les **espaces** : C:\Program Files\Apache Tomcat
 - les **accents** : C:\Tomcaté
 - les **caractères spéciaux** : C:\Tomcat#10
- **Préférer** : C:\tomcat ou D:\apache-tomcat-10.1

Jakarta EE

Bonnes pratiques d'installation d'Apache Tomcat : nom du dossier

- Utiliser uniquement des **caractères alphanumériques**
- **Éviter :**
 - les **espaces** : C:\Program Files\Apache Tomcat
 - les **accents** : C:\Tomcaté
 - les **caractères spéciaux** : C:\Tomcat#10
- **Préférer** : C:\tomcat OU D:\apache-tomcat-10.1

En résumé

Chemin simple, sans espace ni accent = Tomcat stable et sans erreur !

Jakarta EE

Les étapes

- Aller dans le menu `File > New` et cliquer sur `Other` (ou `CTRL` + `N`)
- Ensuite choisir `Dynamic Web Project` situé dans le répertoire `Web`
- Saisir `cours-jee` dans `Project name`
- Ensuite cliquer sur le bouton `New Runtime`, choisir la dernière version d'Apache Tomcat (ici v10.0) et cocher la case juste en-dessous `create a new local server`
- Cliquer sur `Next` et préciser le répertoire d'installation de **Apache Tomcat**
- Cliquer sur `Finish` ensuite deux fois sur `Next`, puis cocher la case `Generate web.xml deployment descriptor`

Les étapes

- Faire un clic droit sur le projet et choisir `Run As` ensuite `Run on Server`
- Sélectionner le serveur **Apache 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`

Jakarta EE

Arborescence typique d'un projet Jakarta EE

```
cours-jee/  
|--src/  
|  |-- main/  
|     |-- java/  
|     |-- webapp/  
|         |-- WEB-INF/  
|             |-- web.xml  
|             |-- lib/  
|             |-- index.html  
|-- build/
```

Structure d'une application JEE

- Le nom du projet définit la racine de l'arborescence
- La racine contient principalement trois répertoires
 - `Java Resources` est réservé aux classes **Java**
 - `src` contient un dossier `main` contenant deux répertoires `java` et `webapp`
 - `java` pointe vers `Java Resources`
 - `webapp` contient tous les fichiers web
 - `build` contient les fichiers compilés (d'extension `.class`)

Contenu de `webapp`

Les fichiers définis directement dans `webapp` sont accessibles aux visiteurs (sans avoir besoin de passer par une **Servlet**). C'est ici qu'on place les fichiers **CSS**, **JavaScript**...

- `WEB-INF` (inaccessible aux visiteurs par **URL**) a un dossier `lib` (pour les librairies externes d'extension `.jar`) et un fichier `web.xml` (premier fichier consulté par le serveur **HTTP**).
- `META-INF` contient les méta-données du projet.

Création d'une page **HTML**

- Clic droit sur le projet, aller dans le menu `new` et choisir `HTML File`
- Placer le fichier directement dans `webapp` (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/cours-jee/` ou `http://localhost:8080/cours-jee/index.html`

Remarque

En cas d'erreur lors du lancement d'**Apache Tomcat**

- Faire double clic sur le serveur
- Aller dans la section **Ports** et vérifier que
 - **Tomcat admin port** : 8005
 - **HTTP/1.1** : 8080
 - **AJP/1.3** : 8009

WEB-INF est inaccessible aux visiteurs

- Déplacer `index.html` dans WEB-INF
- Redémarrer le serveur
- Vérifier que `http://localhost:8080/cours-jee/` et `http://localhost:8080/cours-jee/index.html` ne sont plus accessibles.