

Atelier Framework Spring-01

Maven-Partie-04

Objectifs

Gestion de dépendances avec Maven

1. Déclarer une dépendance

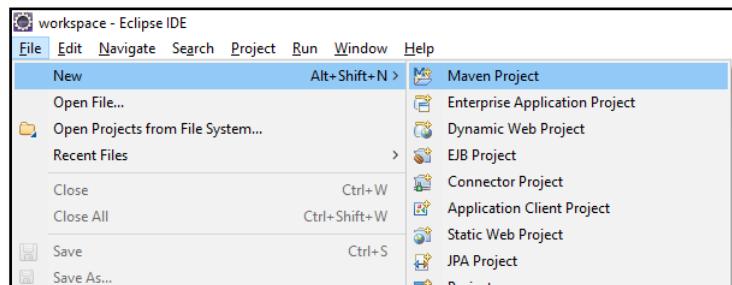
- Créer un projet Maven Avec « eclipse »
- Editer le fichier « pom.xml »
- Réaliser une dépendance à partir du dépôt central (distant)
- Réaliser une dépendance avec un projet livré dans le dépôt local

2. Exécuter et déployer une application WEB

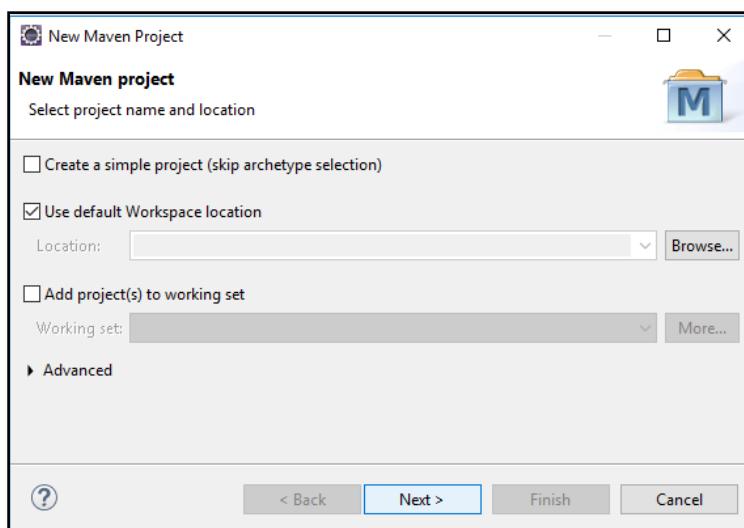
- Exécuter une application WEB
- Déployer une application WEB dans un serveur externe

A. Utiliser le fichier jar livré par un autre projet web

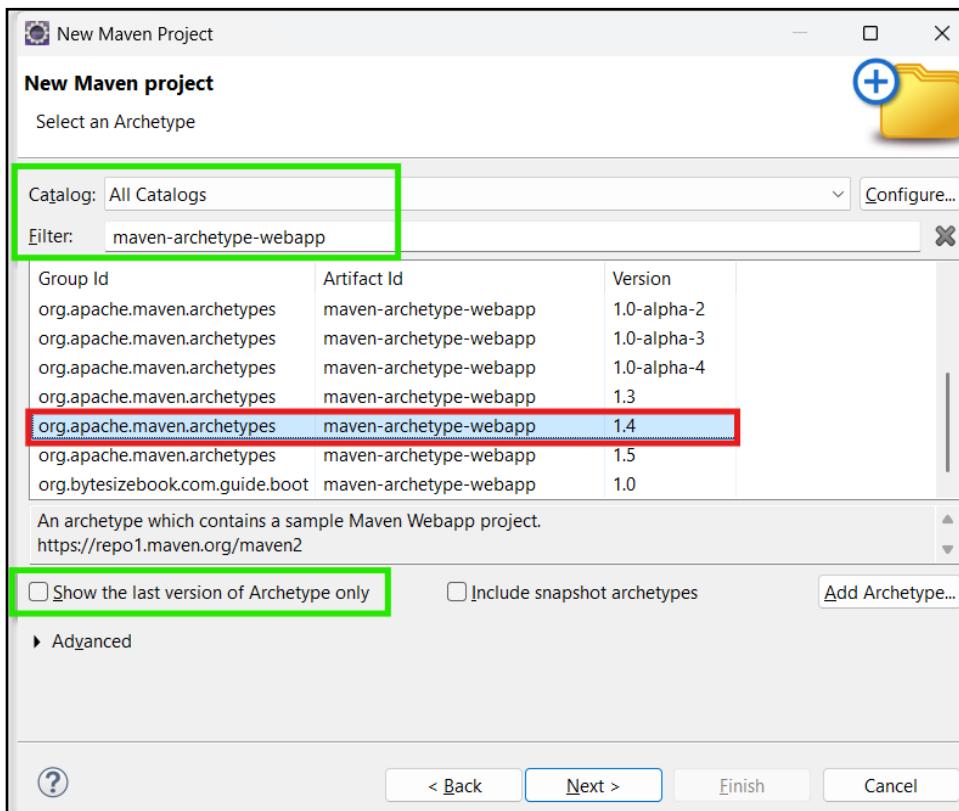
1. Créer un projet de type « Maven Project » en utilisant l'EDI « eclipse » :



2. Un assistant de construction de projet « Maven » s'affiche :



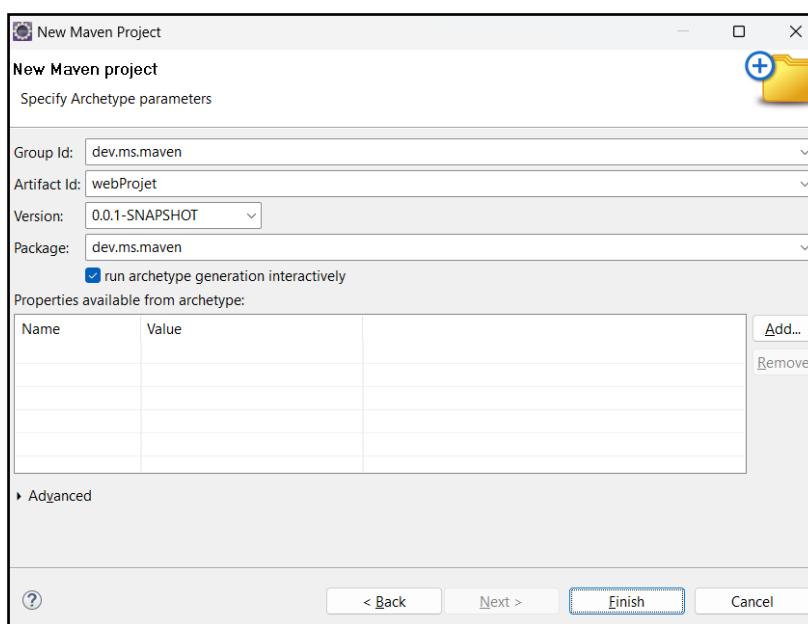
3. Appuyer sur « **Next** » pour choisir un archetype (**modèle**). Cette fois, choisir un archetype « **webapp** » pour créer une application web : **maven-archetype-webapp (version 1.4)**:



Remarquer le téléchargement des plugins par Maven

4. Appuyer sur « **Next** » pour spécifier les paramètres du projet :

- **groupId** : **dev.ms.maven**
- **artifactId** : **webProjet**
- **version** : **0.0.1-SNAPSHOT**
- **package** : **dev.ms.maven** (même valeur que « **groupId**)



- Appuyer sur « **Finish** » pour lancer la construction du projet (**nouvelle application web**)
- Dans le volet « **Console** », taper « **ENTREE** » pour confirmer les paramètres de construction et achever le processus de création du projet « **webProjet** » :

```
[INFO] Generating project in Interactive mode
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-webapp:1.4] found in catalog remote
[INFO] Using property: groupId = dev.ms.maven
[INFO] Using property: artifactId = webProjet
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = dev.ms.maven
confirm properties configuration:
groupId: dev.ms.maven
artifactId: webProjet
version: 0.0.1-SNAPSHOT
package: dev.ms.maven
Y: : |
```

- Pour indiquer que ce projet dépend de notre premier projet « **premierProjet** » (càd utilise des fonctionnalités définies dans une classe existante dans le projet « **premierProjet** »), ouvrir le fichier « **pom.xml** » du projet « **webProjet** » pour ajouter la dépendance suivante à l'intérieur des deux balises **<dependencies>** et **</dependencies>** suite à la dépendance de « **junit** » :

```
<dependency>
  <groupId>dev.ms.maven</groupId>
  <artifactId>premierProjet</artifactId>
  <version>1.0-SNAPSHOT</version>
</dependency>
```

Voici la nouvelle liste de dépendances dans le fichier « **pom.xml** » :

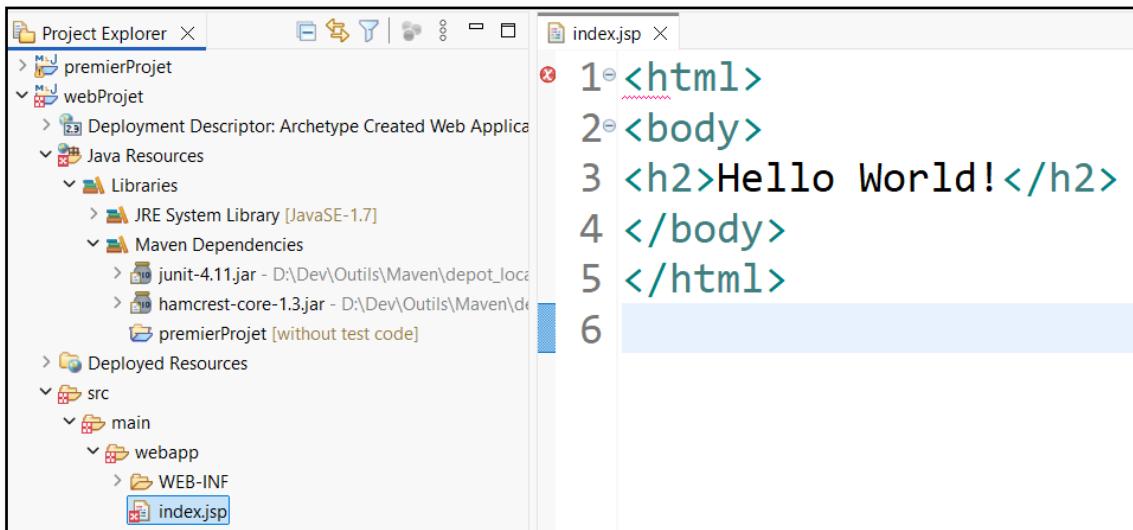
```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>dev.ms.maven</groupId>
    <artifactId>premierProjet</artifactId>
    <version>1.0-SNAPSHOT</version>
  </dependency>
</dependencies>
```

- Enregistrer et remarquer l'existence du jar « **premierProjet** » dans la liste des dépendances MAVEN :



B. Configurer le projet web

9. Remarquer qu'une erreur est signalée au niveau du projet et plus particulièrement au niveau de la page « **index.jsp** » due à la non existence de la classe « **javax.servlet.http.HttpServlet** » :

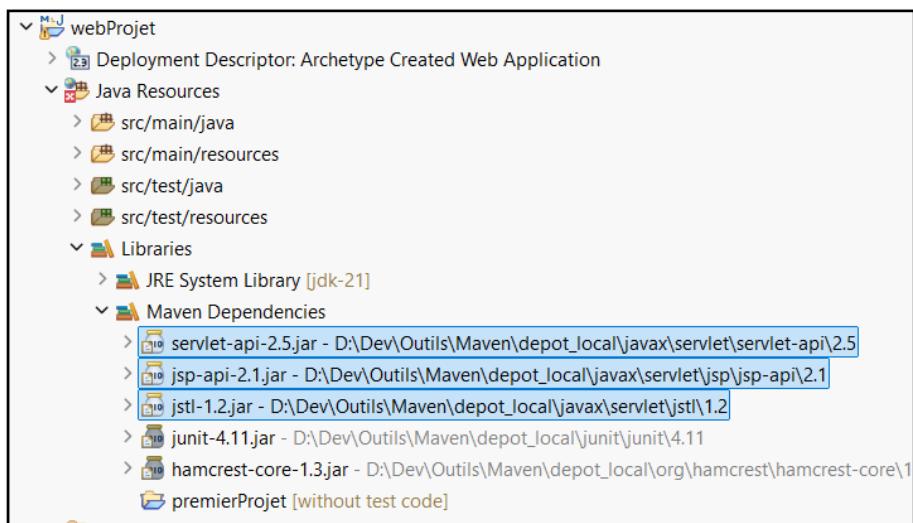


10. Ouvrir le fichier « **pom.xml** » pour ajouter les dépendances suivantes à l'intérieur de la balise **<dependencies>** :

```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>2.5</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.1</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
    <scope>compile</scope>
</dependency>
```

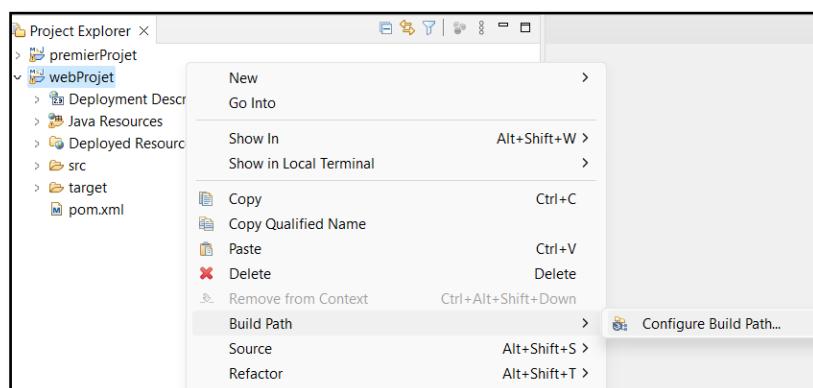
NB : Ne pas oublier d'enregistrer le fichier « **pom.xml** »

Remarquer que l'outil « **Maven** » lance la mise à jour des dépendances et se met à télécharger les nouvelles bibliothèques signalées pour les ajouter aux dépendances de **Maven** :

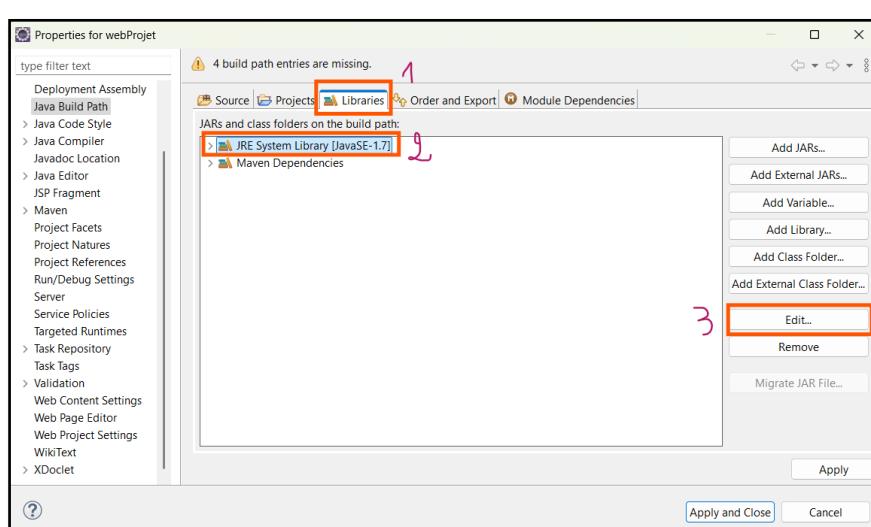


Dans le cas où l'erreur persiste, vérifier la version du JDK :

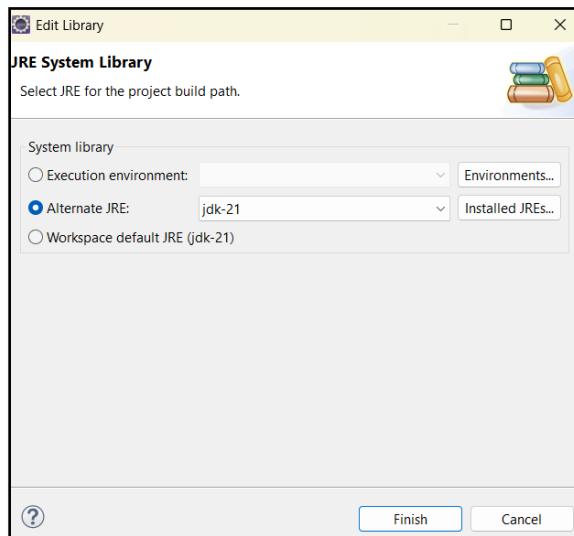
a. Choisir la commande « **Java Build Path** » :



b. Aller à l'onglet « **Librairies** » et éditer le JDK :



c. Spécifier JDK 21 :

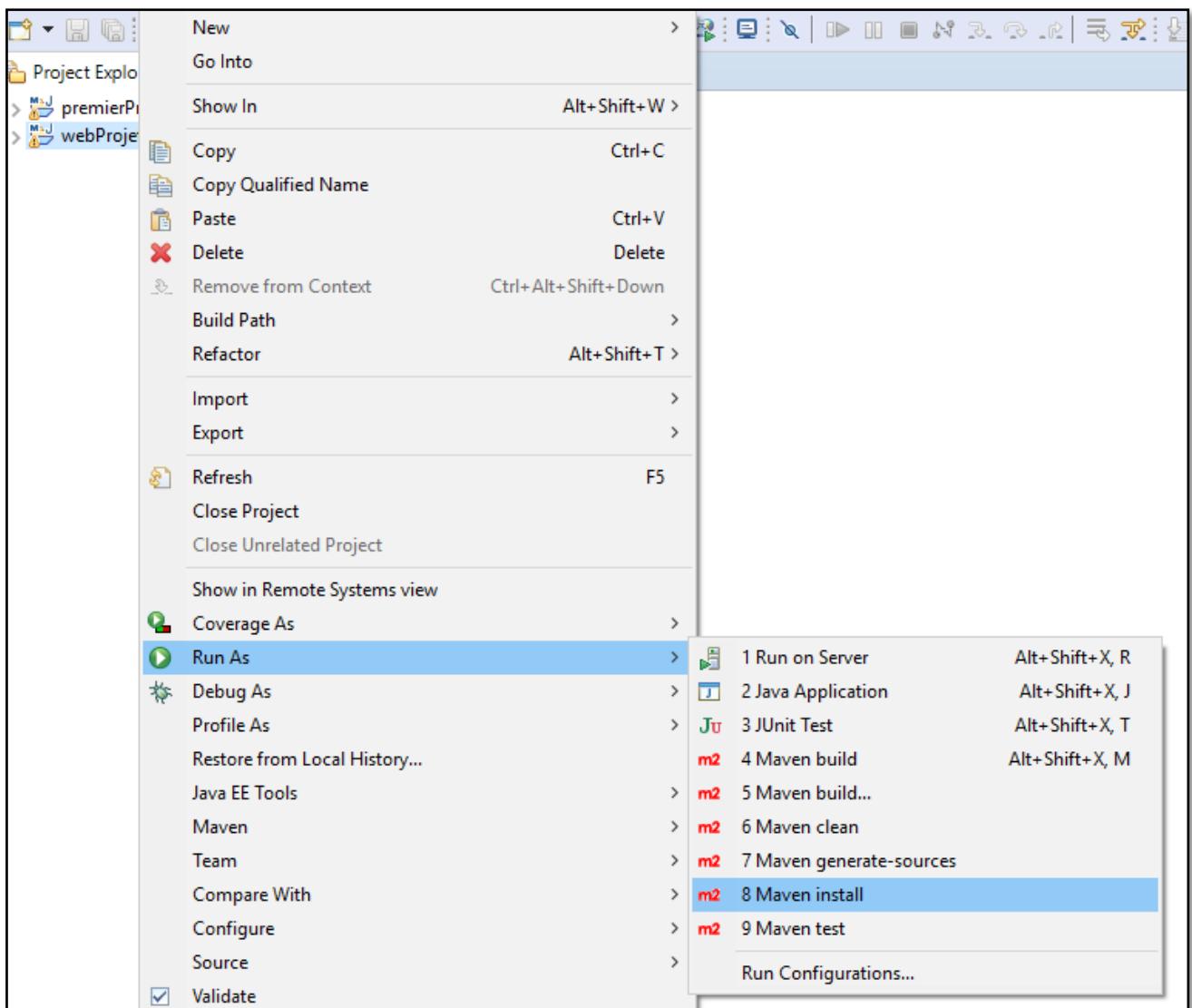


C. Exécuter et déployer l'application WEB

11. Changer le code du fichier «**index.jsp**» qui existe sous «**src/main/webapp**» comme suit :

```
<%@page import="dev.ms.maven.CalculMetier" %>
<% double a= 0.0; double b= 0.0; double res= 0.0;
String action = request.getParameter("action");
if (action!= null) {
    a = Double.parseDouble(request.getParameter("a"));
    b = Double.parseDouble(request.getParameter("b"));
    CalculMetier metier = new CalculMetier();
    if (action.equals ("somme")){
        res= metier.somme(a,b);
    }
    else {
        res= metier.produit(a,b);
    }
}
%>
<html>
<body>
<form action ="index.jsp" method ="post">
<table>
<tr>
    <td> A: </td> <td> <input type ="text" name ="a" value ="<%=a%>" /></td>
    <td> B: </td> <td> <input type ="text" name ="b" value ="<%=b%>" /></td>
</tr>
<tr>
    <td> <input type ="submit" name ="action" value ="somme" /></td>
    <td> <input type ="submit" name ="action" value ="produit"/> </td>
</tr>
<tr> <td> Résultat : </td> <td> <%=res%></td> </tr>
</table>
</form>
</body>
</html>
```

12. Maintenant, utiliser l'assistant « **eclipse** » , pour installer (livrer) le projet « **webProjet** » en sélectionnant le projet puis choisissant la commande :



13. Remarquer la génération d'un fichier « **webProjet.war** » dans le dossier « **target** » à être déployer avec le serveur web « **Tomcat** ».

Ce PC > DATA (D:) > Dev > Outils > Maven > depot_local > dev > ms > maven > webProjet > 0.0.1-SNAPSHOT			
Nom	Type	Taille	
_remote.repositories	Fichier REPOSITORY...	1 Ko	
maven-metadata-local.xml	xmlfile	1 Ko	
webProjet-0.0.1-SNAPSHOT.pom	Fichier POM	3 Ko	
webProjet-0.0.1-SNAPSHOT.war	Fichier WAR	365 Ko	

14. Passons maintenant à configurer un serveur web «**Tomcat**» (version 9). Accéder à l'url suivante :

https://tomcat.apache.org/download-90.cgi

The screenshot shows the 'Binary Distributions' section of the Tomcat download page. It lists several options under 'Core': zip (pgp, sha512), tar.gz (pgp, sha512), 32-bit Windows zip (pgp, sha512), 64-bit Windows zip (pgp, sha512), and 32-bit/64-bit Windows Service Installer (pgp, sha512). The 'zip (pgp, sha512)' link is highlighted with a red box.

15. Télécharger l'archive (zip) du « **Tomcat** » dans le dossier :

D:\Dev\Spring\Outils

16. Extraire l'archive du « **Tomcat** » dans le dossier :

D:\Dev\Spring\Serveurs

17. Accéder au dossier « **bin** » de Tomcat et lancer programme « **startup.bat** » :

The screenshot shows a Windows File Explorer window with the path: Ce PC > DATA (D:) > Dev > Spring > Serveurs > apache-tomcat-9.0.108 > bin. The 'bin' folder contains several files and subfolders. The 'startup' file is selected and highlighted with a blue border.

Nom	Type	Taille
commons-daemon-native.tar	Dossier d'archive compressé	211 Ko
tomcat-native.tar	Dossier d'archive compressé	339 Ko
bootstrap	Executable Jar File	35 Ko
commons-daemon	Executable Jar File	27 Ko
tomcat-juli	Executable Jar File	52 Ko
catalina	Fichier de commande Windows	18 Ko
ciphers	Fichier de commande Windows	3 Ko
configtest	Fichier de commande Windows	2 Ko
digest	Fichier de commande Windows	3 Ko
makebase	Fichier de commande Windows	4 Ko
setclasspath	Fichier de commande Windows	4 Ko
shutdown	Fichier de commande Windows	2 Ko
startup	Fichier de commande Windows	2 Ko
tool-wrapper	Fichier de commande Windows	5 Ko
version	Fichier de commande Windows	2 Ko
catalina-tasks	Microsoft Edge HTML Document	2 Ko
catalina.sh	sh_auto_file	26 Ko

18. Accéder à travers le navigateur web au serveur par le lien :

http://localhost:8080

Voici la page d'accueil de Tomcat :

19. Accéder au volet d'administration « **Manager App** ». Tomcat nécessite une authentification avec un nom d'utilisateur et un mot de passe :

20. Pour se faire, accéder au sous dossier « **conf** » de Tomcat et éditer le fichier « **tomcat-users.xml** » :

Nom	Type	Taille
Catalina	Dossier de fichiers	
catalina.policy	Fichier POLICY	13 Ko
catalina	Fichier source Properties	8 Ko
context	Microsoft Edge HTML Document	2 Ko
jaspic-providers	Microsoft Edge HTML Document	2 Ko
jaspic-providers	XML Schema	3 Ko
logging	Fichier source Properties	4 Ko
server	Microsoft Edge HTML Document	9 Ko
tomcat-users	Microsoft Edge HTML Document	3 Ko
tomcat-users	XML Schema	3 Ko
web	Microsoft Edge HTML Document	175 Ko

21. Ajouter, **entre les deux balises <tomcat-users> et </tomcat-users>**, les instructions suivantes pour définir un utilisateur (**admin**, **admin**) ayant les droits nécessaires pour gérer les applications web :

```
<role rolename="manager-gui"/>

<role rolename="admin-gui"/>

<role rolename="manager-script"/>

<user username="admin" password="admin" roles="manager-gui,admin-gui, manager-script"/>
```

- ✓ Enregistrer le fichier puis Arrêter le serveur « **Tomcat** » en cliquant sur le fichier « **shutdown.bat** » (dans le sous-dossier « **bin** »)
- ✓ Redémarrer « **Tomcat** » (en cliquant sur le fichier « **startup.bat** »)
- ✓ Ré-accéder à l'interface web d'administration de tomcat pour s'authentifier avec les deux paramètres (« **admin** », « **admin** ») et gérer les applications déployées :

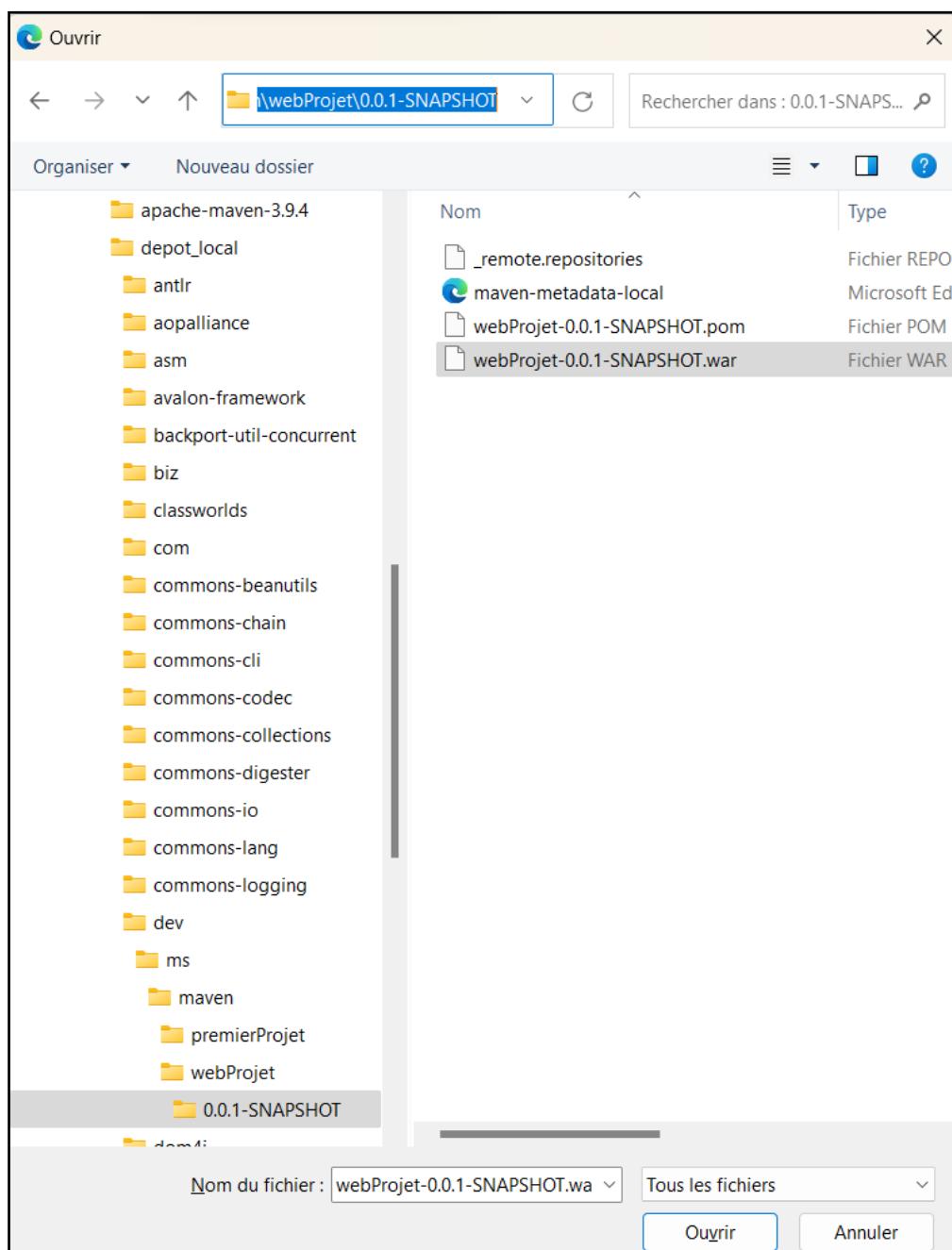
The screenshot shows the Tomcat Manager application running at localhost:8080/manager/html. The main page displays a table of deployed applications under the 'Applications' section. The table includes columns for Chemin (Path), Version (Version), Nom d'affichage (Display Name), Fonctionnelle (Functional), Sessions (Sessions), and Commandes (Commands). The applications listed are: Welcome to Tomcat, Tomcat Documentation, Servlet and JSP Examples, Tomcat Host Manager Application, and Tomcat Manager Application. Each application row has buttons for Démarrer (Start), Arrêter (Stop), Recharger (Reload), Retirer (Remove), and options to Expirer les sessions (Inactives depuis ≥ 30 minutes).

- ✓ Accéder dans l'interface web ainsi ouverte au volet « **Déployer \Fichier WAR à déployer** » pour déployer l'application web qui est déjà livrée dans le repository local.

The screenshot shows the 'Deployer' interface with the following sections:

- Emplacement du répertoire ou fichier WAR de déploiement sur le serveur**: Contains fields for 'Chemin de contexte (requis)', 'Version (pour les déploiements en parallèle)', 'URL du fichier XML de configuration', and 'URL vers WAR ou répertoire'.
- Fichier WAR à déployer**: Contains a button 'Choisir le fichier WAR à téléverser' and a 'Deployer' button. A red box highlights the 'Choisir le fichier WAR à téléverser' button and its associated text input field.

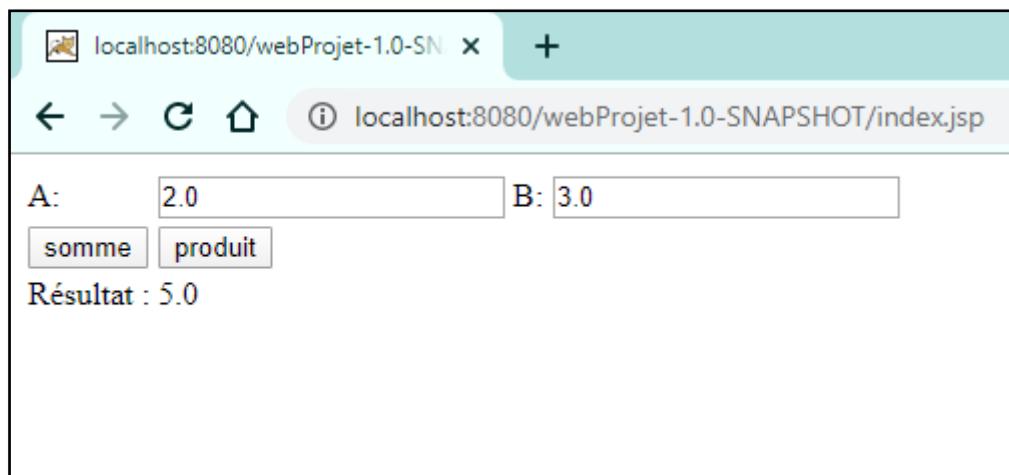
- ✓ Sélectionner le fichier WAR du projet « **webProjet** » :



- ✓ Cliquer ensuite sur « **Déployer** » pour avoir le projet web déployé dans la liste des applications prises en compte par le serveur Tomcat :

Applications					
Chemin	Version	Nom d'affichage	Fonctionnelle	Sessions	Commandes
/	Aucun spécifié	Welcome to Tomcat	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/docs	Aucun spécifié	Tomcat Documentation	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/examples	Aucun spécifié	Servlet and JSP Examples	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/host-manager	Aucun spécifié	Tomcat Host Manager Application	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/manager	Aucun spécifié	Tomcat Manager Application	true	1	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/webProjet-0.0.1-SNAPSHOT	Aucun spécifié	Archetype Created Web Application	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes

- ✓ Ainsi le projet WEB est déployé il suffit de cliquer sur le lien correspondant pour lancer son exécution.
- ✓ Voici le résultat :



Exercice

Réaliser les modifications nécessaires pour que la page **index.jsp** affiche aussi un troisième bouton « **Division** » pour réaliser une opération **a / b**