

## Java EE

TP n°1.2 : Utilisation des API : EJB 3.0 et JSF 2.2

Création d'un projet de type WAR.

## Sommaire

1. Objet.....	3
2. Pré requis.....	3
3. Développement de l'application WEB.....	3
4. Déploiement et test de l'application WEB .....	15

## 1. Objet

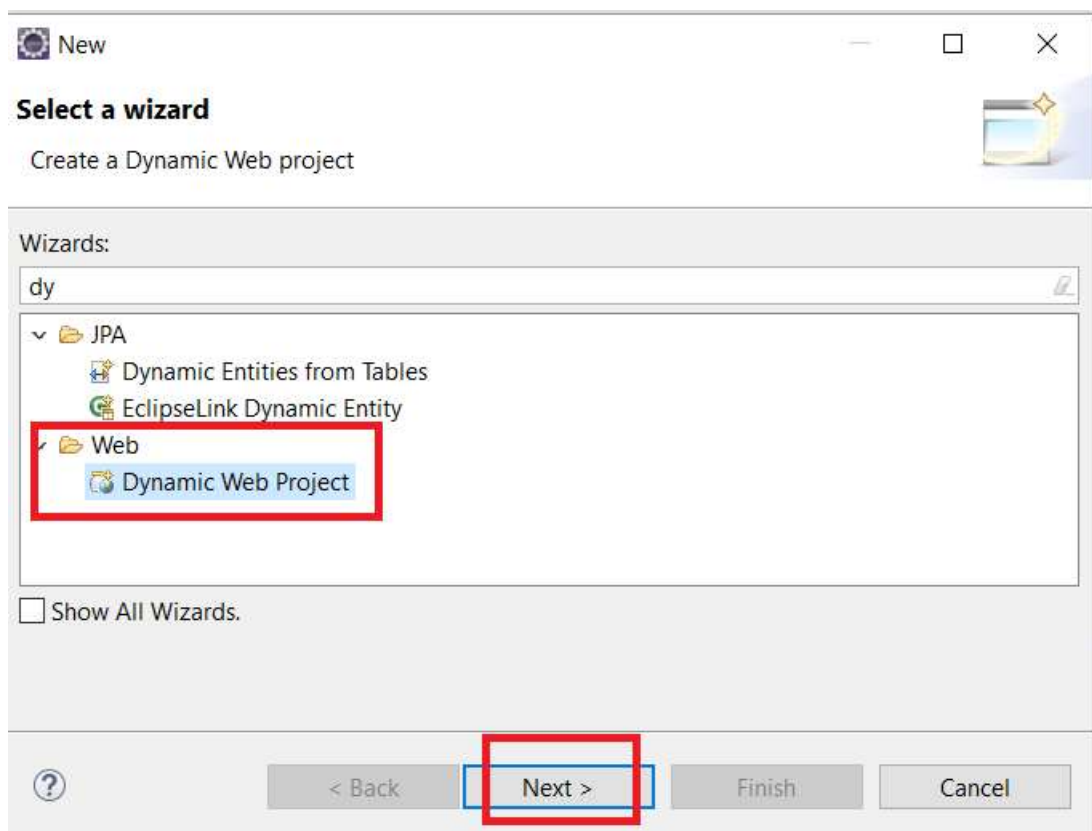
- ✓ Développer un projet web (de type WAR) pour administrer les articles d'une application eCommerce.
- ✓ Développer la couche présentation avec JSF 2.x : les V en utilisant les tags JSF (fichier .xhtml), le C en utilisant la Servlet offerte par JSF et le M en utilisant les Managed Bean proposés par JSF.
- ✓ Utiliser la couche métier développée dans le TP n°1.1 avec EJB 3.0.
- ✓ Déployer le projet web au niveau de WildFly 8.2.0.
- ✓ Tester les différents cas de test (CRUD).

## 2. Pré requis

- ✓ Il faut réaliser tout d'abord le TP n°1.1.
- ✓ JDK 1.7.
- ✓ Eclipse Luna.
- ✓ WildFly 8.2.0.
- ✓ MySQL 5.0 (driver mysql-connector-java-5.1.17-bin.jar).

## 3. Développement de l'application WEB

Créer le projet web **ecommercetp** comme illustre l'écran suivant :



Cliquer sur **Next>** :

New Dynamic Web Project

### Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:  <sup>1</sup>

Project location

☒ Use default location

Location:

Target runtime

Dynamic web module version

<sup>3</sup> <sup>4</sup>

Configuration

Hint: Get started quickly by selecting one of the pre-defined project configurations.

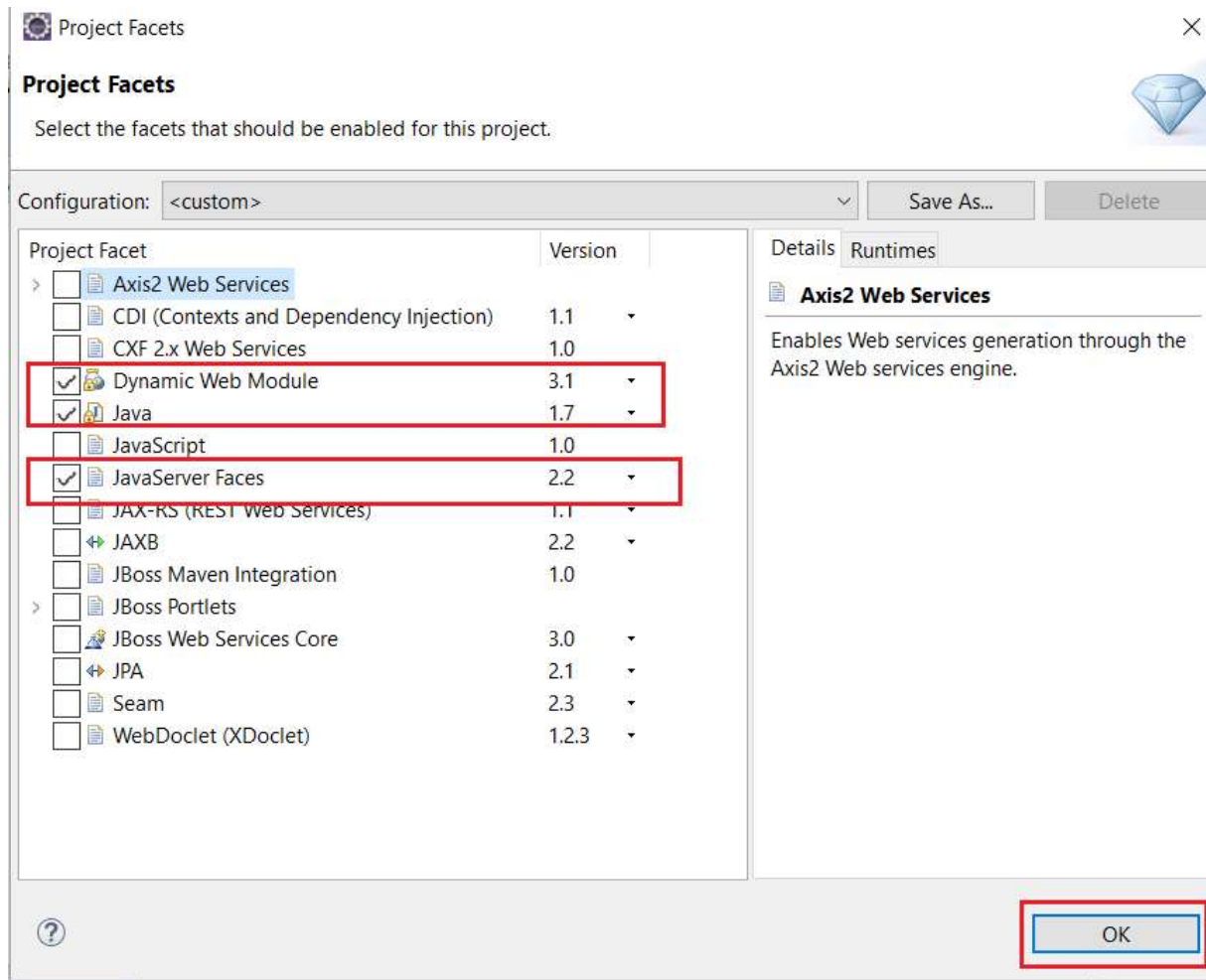
EAR membership

☐ Add project to an EAR

EAR project name:

- 1- Saisir le nom du projet (par exemple : ecommercetp).
- 2- Sélectionner la **Target Runtime**.
- 3- Sélectionner 3.1.

Cliquer sur le bouton **Modify...**



- 1- Cocher Dynamic Web Module version 3.1.
- 2- Cocher Java version 1.7.
- 3- Cocher JSF version 2.2.

Cliquer ensuite sur le bouton **OK** :

## Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.



Project name:

Project location

☒ Use default location

Location:

Target runtime

Dynamic web module version

Configuration

Hint: Get started quickly by selecting one of the pre-defined project configurations.

EAR membership

☐ Add project to an EAR

EAR project name:


Cliquer sur **Next>** :

## Java

Configure project for building a Java application.



Source folders on build path:

 src

Add Folder...

Edit...

Remove

Default output folder:

build\classes

?

< Back

Next >

Finish

Cancel

Cliquer sur **Next>** :

New Dynamic Web Project

**Web Module**

Configure web module settings.

Context root:

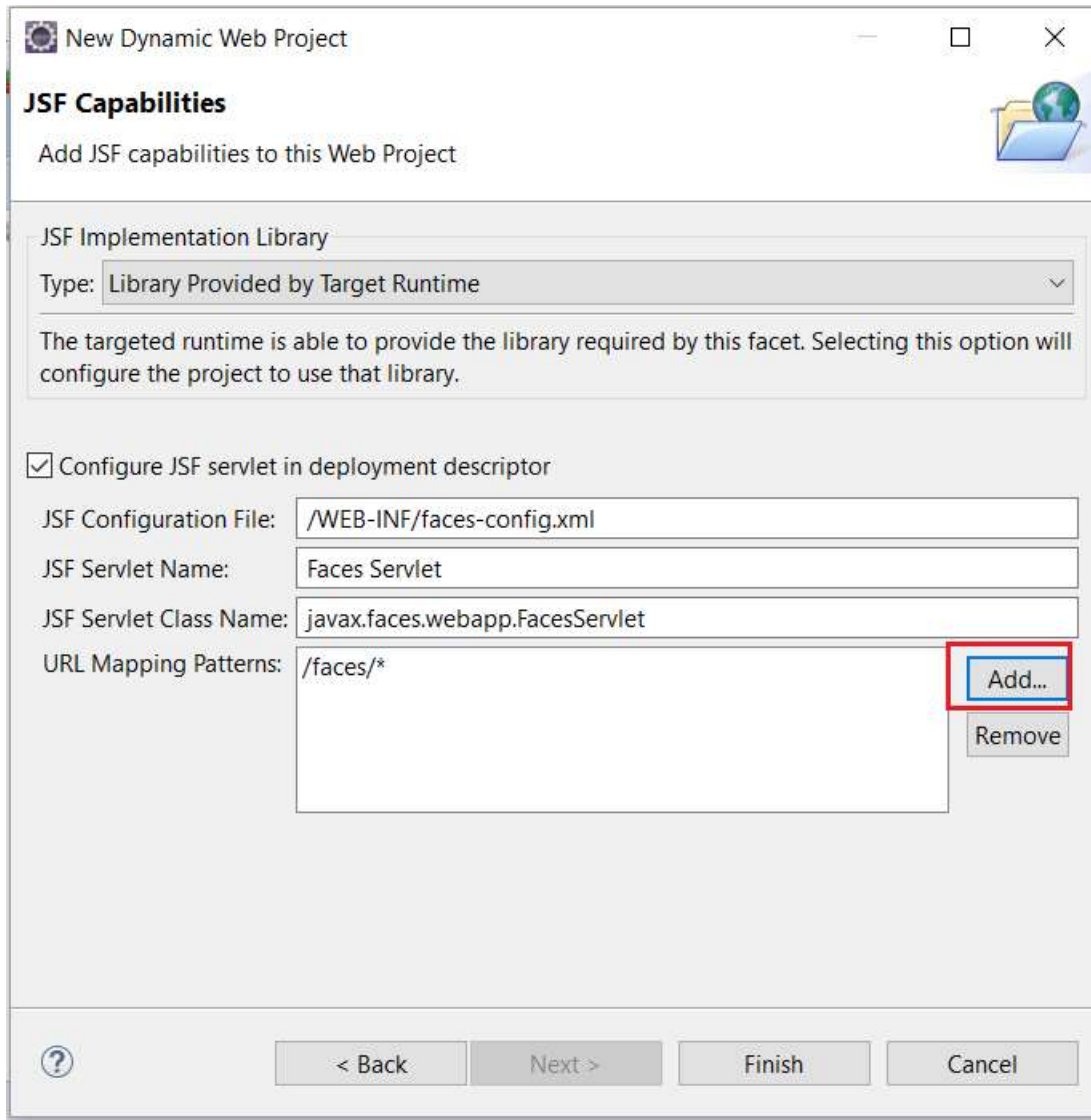
Content directory:

☒ Generate web.xml deployment descriptor

? < Back Next > Finish Cancel

Cocher "**Generate web.xml**" et cliquer sur le bouton **Next>** :





**New Dynamic Web Project**

**JSF Capabilities**  
Add JSF capabilities to this Web Project

JSF Implementation Library  
Type: Library Provided by Target Runtime

The targeted runtime is able to provide the library required by this facet. Selecting this option will configure the project to use that library.

☒ Configure JSF servlet in deployment descriptor


JSF Configuration File: /WEB-INF/faces-config.xml

JSF Servlet Name: Faces Servlet

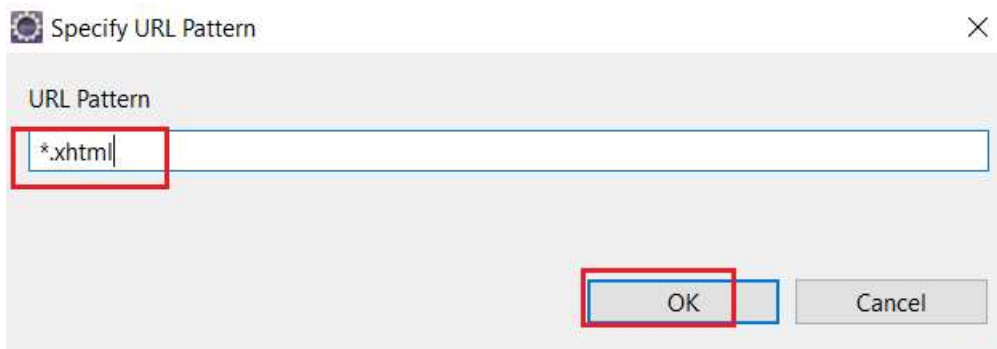
JSF Servlet Class Name: javax.faces.webapp.FacesServlet

URL Mapping Patterns: /faces/\*

**Add...**  
Remove

 < Back Next > Finish Cancel

Cliquer sur **Add...** :




**Specify URL Pattern**

URL Pattern

\*.xhtml

**OK** Cancel

Saisir **\*.xhtml** et cliquer sur **OK** :

 New Dynamic Web Project

### JSF Capabilities

Add JSF capabilities to this Web Project

JSF Implementation Library

Type: Library Provided by Target Runtime

The targeted runtime is able to provide the library required by this facet. Selecting this option will configure the project to use that library.

☒ Configure JSF servlet in deployment descriptor

JSF Configuration File: /WEB-INF/faces-config.xml

JSF Servlet Name: Faces Servlet

JSF Servlet Class Name: javax.faces.webapp.FacesServlet

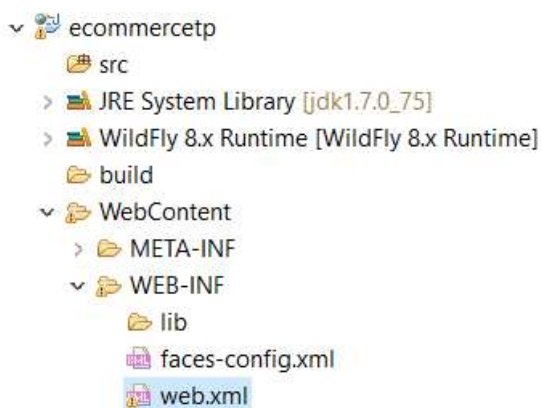
URL Mapping Patterns:

- /faces/
- \*.xhtml

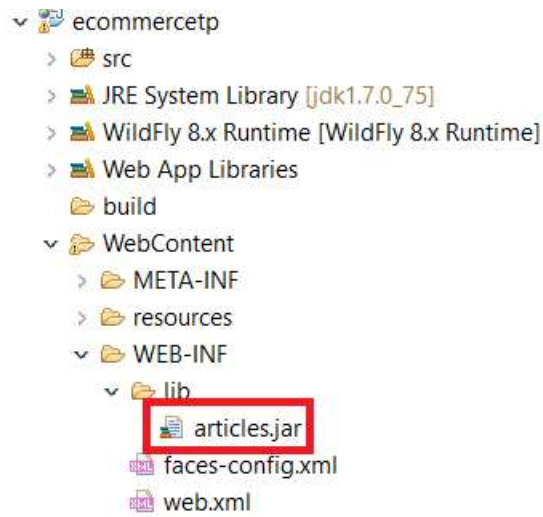
Add... Remove

? < Back Next > **Finish** Cancel

Cliquer sur **Finish**. L'arborescence du projet ci-après sera créée :



Copier le fichier JAR du TP n°1 (articles.jar) dans WEB-INF/lib :



Créer la classe `ArticleBean` suivante :

```
package ma.gov.formation.cigma.presentation;

import java.util.ArrayList;
import java.util.List;

import javax.ejb.EJB;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;

import ma.gov.formation.cigma.service.Article;
import ma.gov.formation.cigma.service.IService;

@ManagedBean(name = "articleBean")
@SessionScoped
public class ArticleBean {
    @EJB
    private IService service;
    private List<Article> articles = new ArrayList<>();
    private Article article = new Article();

    public Article getArticle() {
        return article;
    }

    public void setArticle(Article article) {
        this.article = article;
    }

    public void setArticles(List<Article> articles) {
        this.articles = articles;
    }

    public IService getService() {
        return service;
    }

    public void setService(IService service) {
        this.service = service;
    }
}
```

```

    }

    public List<Article> getArticles() {
        articles = service.getAll();
        return articles;
    }

    public String addArticle() {
        service.save(article);
        return null;
    }
}

```

- Configurer la page de bienvenue (articles.xhtml) au niveau du fichier **web.xml** :

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    id="WebApp_ID" version="3.1">
    <display-name>ecommercetp</display-name>
    <welcome-file-list>
        <welcome-file>articles.xhtml</welcome-file>
    </welcome-file-list>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>*.xhtml</url-pattern>
    </servlet-mapping>
</web-app>

```

- Créer la page **articles.xhtml** suivante (la page à mettre dans le chemin WebContent/WEB-INF/lib) :

```

<html lang="en" xmlns="http://www.w3.org/1999/xhtml"
    xmlns:f="http://xmlns.jcp.org/jsf/core"
    xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
    xmlns:a="http://xmlns.jcp.org/jsf/passthrough"
    xmlns:h="http://xmlns.jcp.org/jsf/html">

    <h:head>
        <title>TP n°2 EJB 3.0 et JSF 2.2</title>
        <h:outputStylesheet library="css" name="styles.css" />
    </h:head>

    <h:body>
        <h:form>
            <h:panelGrid columns="2" border="1" cellpadding="10"
                cellspacing="1">
                <f:facet name="header">
                    <h:outputText value="Ajouter un nouvel article :" />
                </f:facet>
            </h:panelGrid>
        </h:form>
    </h:body>
</html>

```

```

        <h:outputLabel value="Désignation de L'article" />
        <h:inputText
value="#{articleBean.article.designation}"></h:inputText>
        <h:outputLabel value="Prix" />
        <h:inputText
value="#{articleBean.article.prix}"></h:inputText>
        <h:outputLabel value="Quantité en stock" />
        <h:inputText
value="#{articleBean.article.quantiteEnStock}"></h:inputText>
        <f:facet name="footer">
            <h:panelGroup style="display:block; text-align:center">
                <h:commandButton value="Ajouter"
action="#{articleBean.addArticle}"></h:commandButton>
            </h:panelGroup>
        </f:facet>
    </h:panelGrid>

    <h2>Liste des articles :</h2>
    <h:dataTable value="#{articleBean.articles}" var="article"
        styleClass="articleTable" headerClass="articleTableHeader"
        rowClasses="articleTableOddRow,articleTableEvenRow">

        <h:column>
            <f:facet name="header">Identifiant</f:facet>

            #{article.id}

        </h:column>

        <h:column>
            <f:facet name="header">Désignation</f:facet>
            #{article.designation}

        </h:column>

        <h:column>
            <f:facet name="header">Prix</f:facet>
            #{article.prix}

        </h:column>

        <h:column>
            <f:facet name="header">Quantité en stock</f:facet>
            #{article.quantiteEnStock}

        </h:column>
    </h:dataTable>
</h:form>

</h:body>
</html>

```

- Créer la feuille de style **styles.css** suivante (le fichier à créer dans le chemin : WebContent/resources/css) :

```

.articleTable {
    border-collapse:collapse;

```

```

    border:1px solid #000000;
}

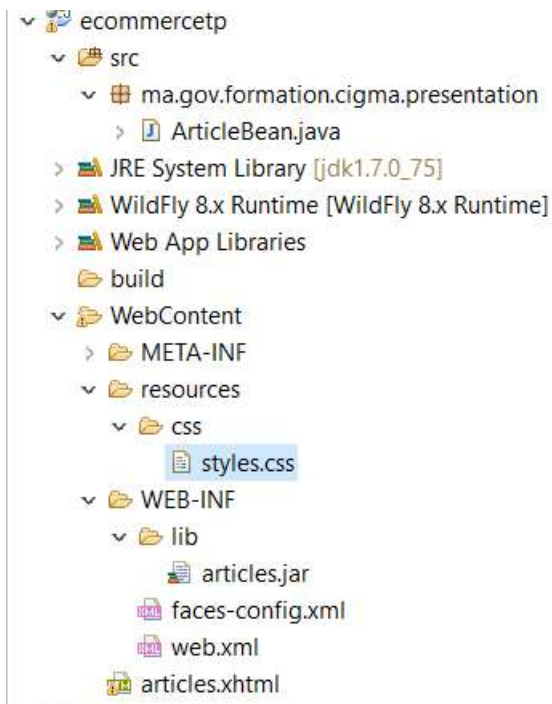
.articleTableHeader {
    text-align:center;
    background:none repeat scroll 00#B5B5B5;
    border-bottom:1px solid #000000;
    padding:2px;
}

.articleTableOddRow {
    text-align:center;
    background:none repeat scroll 00#FFFFFF;
}

.articleTableEvenRow {
    text-align:center;
    background:none repeat scroll 00#D3D3D3;
}

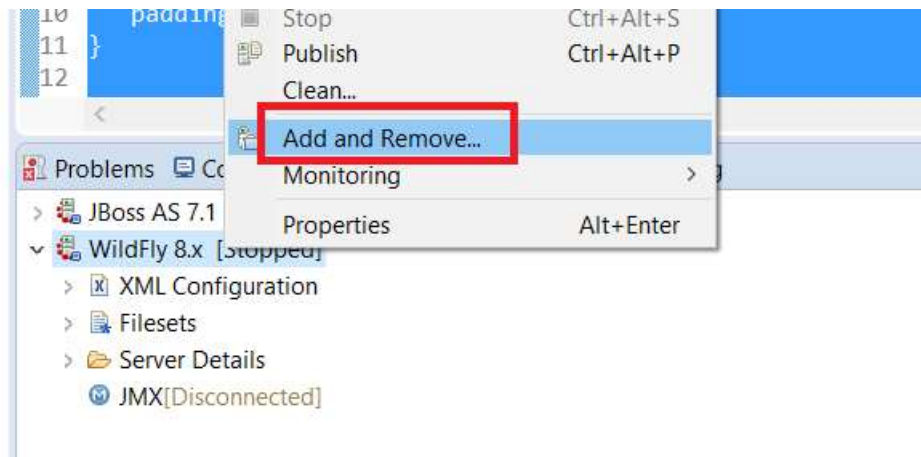
```

L'arborescence finale de votre projet web est la suivante :

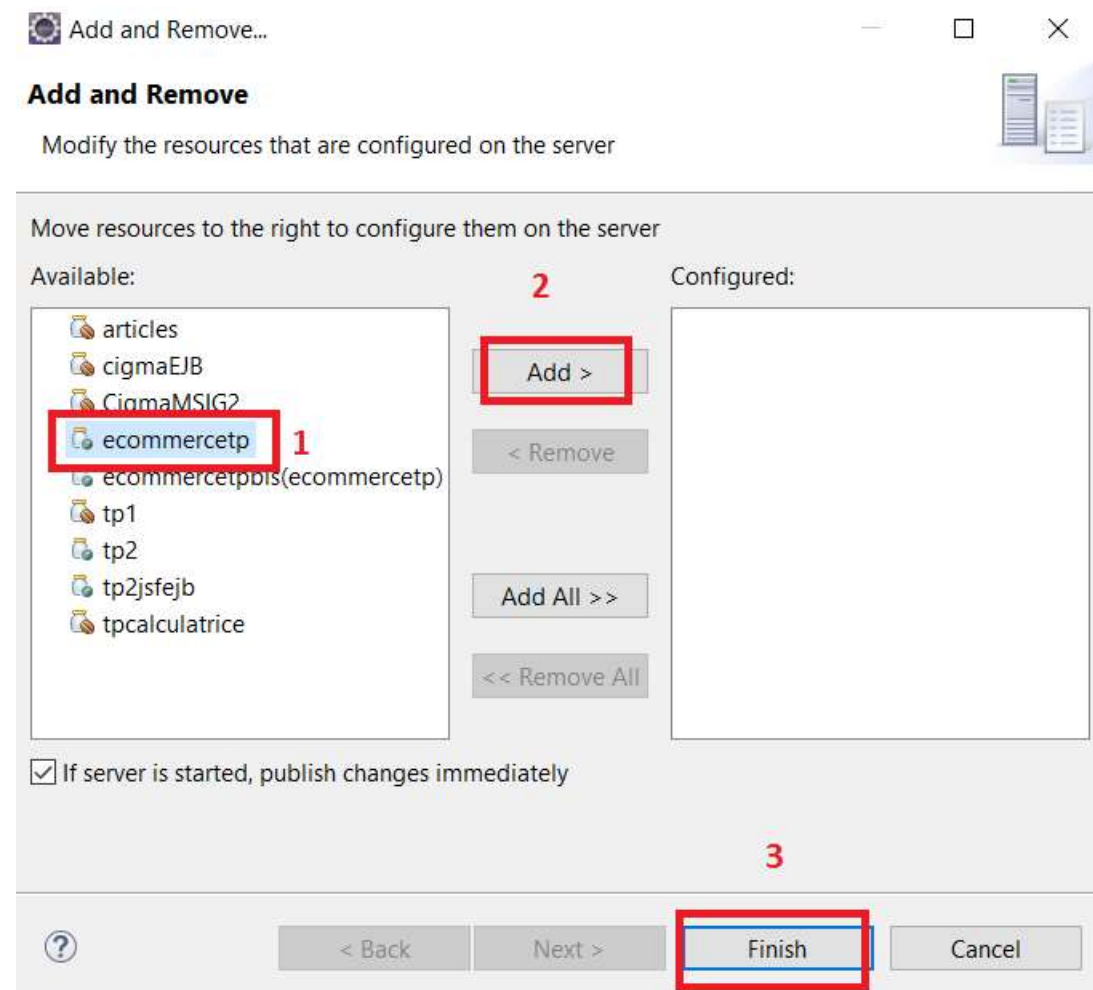


## 4. Déploiement et test de l'application WEB

Cliquer à droite de la souris sur le serveur WildFly comme illustre l'écran suivant :

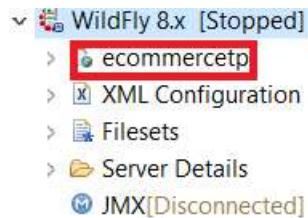


Cliquer sur **Add and Remove ...**



- 1- sélectionner votre application web.
- 2- Cliquer sur le bouton **Add**.

Cliquer sur le bouton **Finish**. L'application est déployée au niveau de WildFly comme illustre l'écran suivant :



- Démarrer votre serveur et lancer le lien : <http://localhost:8080/ecommercetp>

la page **articles.xhtml** devrait être exécutée :

Ajouter un nouvel article :	
Désignation de l'article	<input type="text"/>
Prix	<input type="text"/>
Quantité en stock	<input type="text"/>
Ajouter	

## Liste des articles :

Identifiant	Désignation	Prix	Quantité en stock
23	TV LG	7000.0	20
24	TV Sony	10000.0	15
25	PC DELL	1500.0	70
26	Switch	1000.0	100
29	pc sony	15000.0	100

Pour ajouter un nouvel article, saisir la désignation, le prix et la quantité et cliquer sur Ajouter, comme illustre l'écran suivant :



TP n°2 EJB 3.0 et JSF 2.2 x

localhost:8080/ecommercetp/

**Ajouter un nouvel article :**

Désignation de l'article	<input type="text" value="TV LED"/>
Prix	<input type="text" value="6000"/>
Quantité en stock	<input type="text" value="20"/>
<input type="button" value="Ajouter"/>	

### Liste des articles :

Identifiant	Désignation	Prix	Quantité en stock
23	TV LG	7000.0	20
24	TV Sony	10000.0	15
25	PC DELL	1500.0	70
26	Switch	1000.0	100
29	pc sony	15000.0	100

Cliquer sur Ajouter, le nouvel article est ajouté comme illustré ci-dessous :



Ajouter un nouvel article :	
Désignation de l'article	<input type="text" value="TV LED"/>
Prix	<input type="text" value="6000.0"/>
Quantité en stock	<input type="text" value="20"/>
<input type="button" value="Ajouter"/>	

### Liste des articles :

Identifiant	Désignation	Prix	Quantité en stock
23	TV LG	7000.0	20
24	TV Sony	10000.0	15
25	PC DELL	1500.0	70
26	Switch	1000.0	100
29	pc sony	15000.0	100
33	TV LED	6000.0	20

### Questions :

- ✓ Développer la fonctionnalité permettant de modifier un article.
- ✓ Développer la fonctionnalité permettant de supprimer un article.