

ING2
Semestre 1

Développement d'applications Web

Éric Pipard

Travail dirigé n° 4 **Développement d'une JSP**

Objectifs : connaître le synopsis de développement d'une jsp

Mots-clés : jsp, bean, scriptlet

Table des matières

1 – Introduction aux JSP.....	3
2 – Création de la vue.....	3
3 – Cycle de vie d'une JSP.....	4
4 – Création d'une page JSP.....	5
5 – Passage d'information via l'URL.....	5
6 – Import de classes personnalisées.....	6
7 – Déclaration de méthodes dans une page JSP.....	7
8 – Balises ou actions personnalisées.....	7
9 – Inclusion de ressources externes dans une page.....	8
10 – Utilisation d'un bean.....	9

1 – Introduction aux JSP

Le modèle MVC indique de placer tout ce qui touche à l'affichage (texte, mise en forme, etc.) dans une couche à part : la vue. La technologie pour réaliser une vue est la page JSP.

Une page JSP est un fichier qui :

- possède l'extension .jsp ;
- peut contenir des balises HTML et des balises JSP qui appellent de manière transparente du code Java ;
- est exécutée côté serveur après avoir été transformée en servlet.

L'intérêt est de rendre possible la création de pages dynamiques : puisqu'il y a une étape de génération sur le serveur, il devient possible de faire varier l'affichage et d'interagir avec l'utilisateur, en fonction notamment de la requête et des données reçues.

Il est possible de générer n'importe quel type de format avec une page JSP : du HTML, du CSS, du XML, du texte brut, etc. Dans la très grande majorité des cas, il s'agira de pages HTML destinées à l'affichage des données de l'application sur le navigateur du client.

Le langage JSP combine à la fois les technologies HTML, XML, servlet et JavaBeans permettant de créer des vues dynamiques.

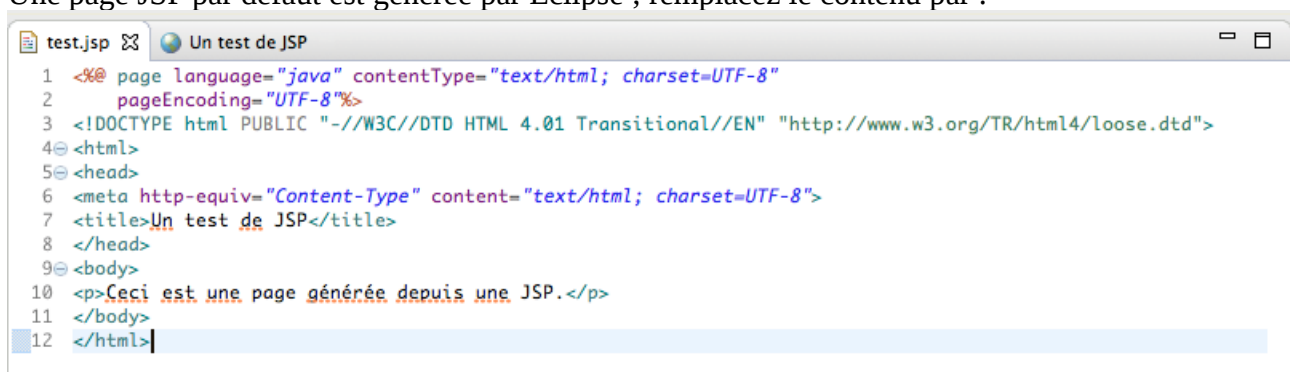
Remarque :

- quand une page JSP est placée sous /WEB-INF, il n'est pas possible d'y accéder directement depuis le navigateur, il faut passer par une servlet ;
- pour pouvoir y accéder sans servlet, alors il suffit de placer les JSP sous /WebContent et pas sous /WEB-INF.

2 – Création de la vue

Depuis Eclipse faire un clic droit sur le dossier WebContent du projet SimpleJsp, puis choisir New>JSP File, et dans la fenêtre qui apparaît renseignez les informations demandées (nom de la page : test).

Une page JSP par défaut est générée par Eclipse ; remplacez le contenu par :

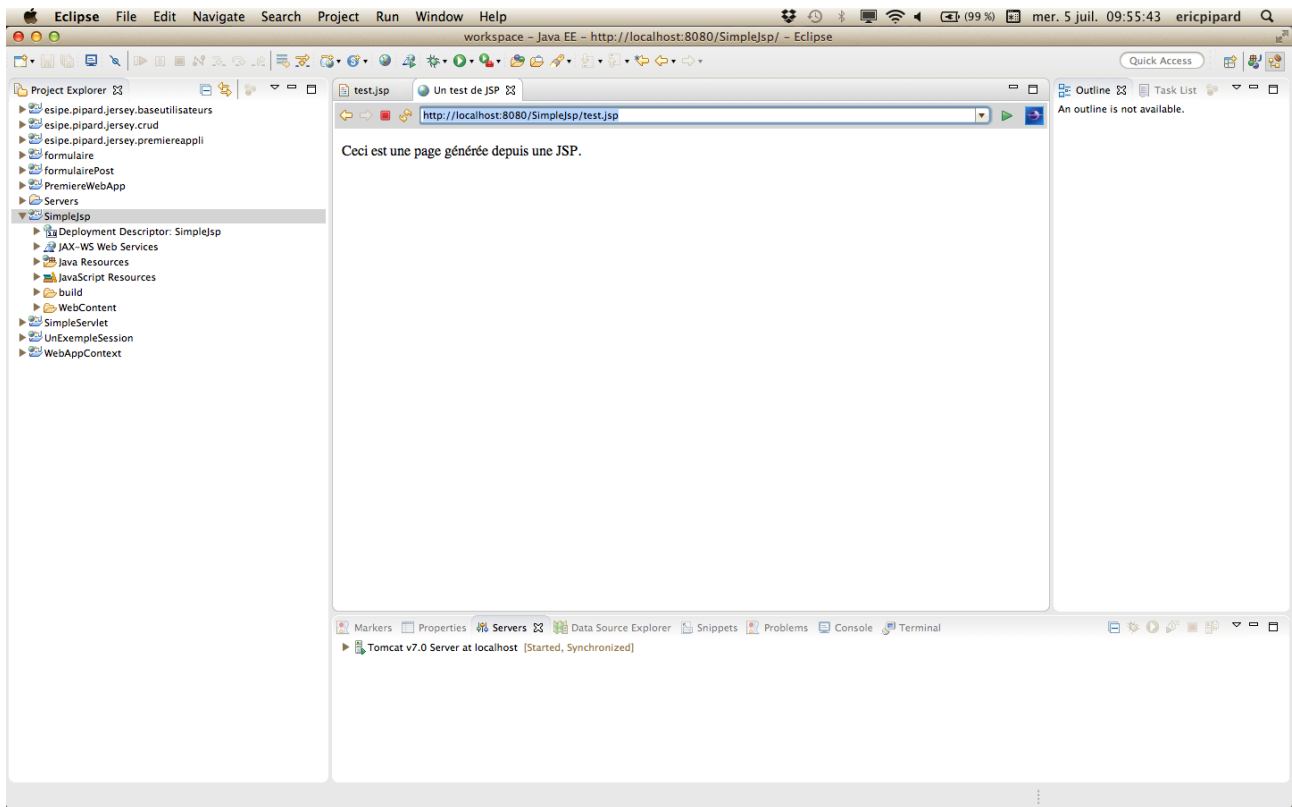


```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7 <title>Un test de JSP</title>
8 </head>
9 <body>
10 <p>Ceci est une page générée depuis une JSP.</p>
11 </body>
12 </html>
```

Entrez l'URL :

<http://localhost:8080/SimpleJsp/test.jsp>

Vous devriez obtenir ceci :



3 – Cycle de vie d'une JSP

Quand une JSP est demandée pour la première fois, ou quand l'application web démarre, le conteneur de servlets va vérifier, traduire puis compiler la page JSP en une classe héritant de `HttpServlet`.

Le moteur de servlets va exécuter la classe JSP auparavant traduite et compilée et envoyer la sortie générée (typiquement, une page HTML/CSS/JS) depuis le serveur vers le client à travers le réseau, sortie qui sera alors affichée dans son navigateur.

Voici ce qui se passe au sein du conteneur de servlets lorsqu'une requête HTTP est destinée à une JSP :

- le conteneur vérifie si la JSP a déjà été traduite et compilée en une servlet :
 - si non, il vérifie la syntaxe de la page, la traduit en une servlet (du code Java) et la compile en une classe exécutable prête à l'emploi ;
 - si oui, il vérifie que l'âge de la JSP et de la servlet est identique :
 - si non, cela signifie que la JSP est plus récente que la servlet et donc qu'il y a eu modification, le conteneur effectue alors à nouveau les tâches de vérification, traduction et compilation ;
- il charge ensuite la classe générée, en crée une instance et l'exécute pour traiter la requête.

Le code généré ressemble à ceci :

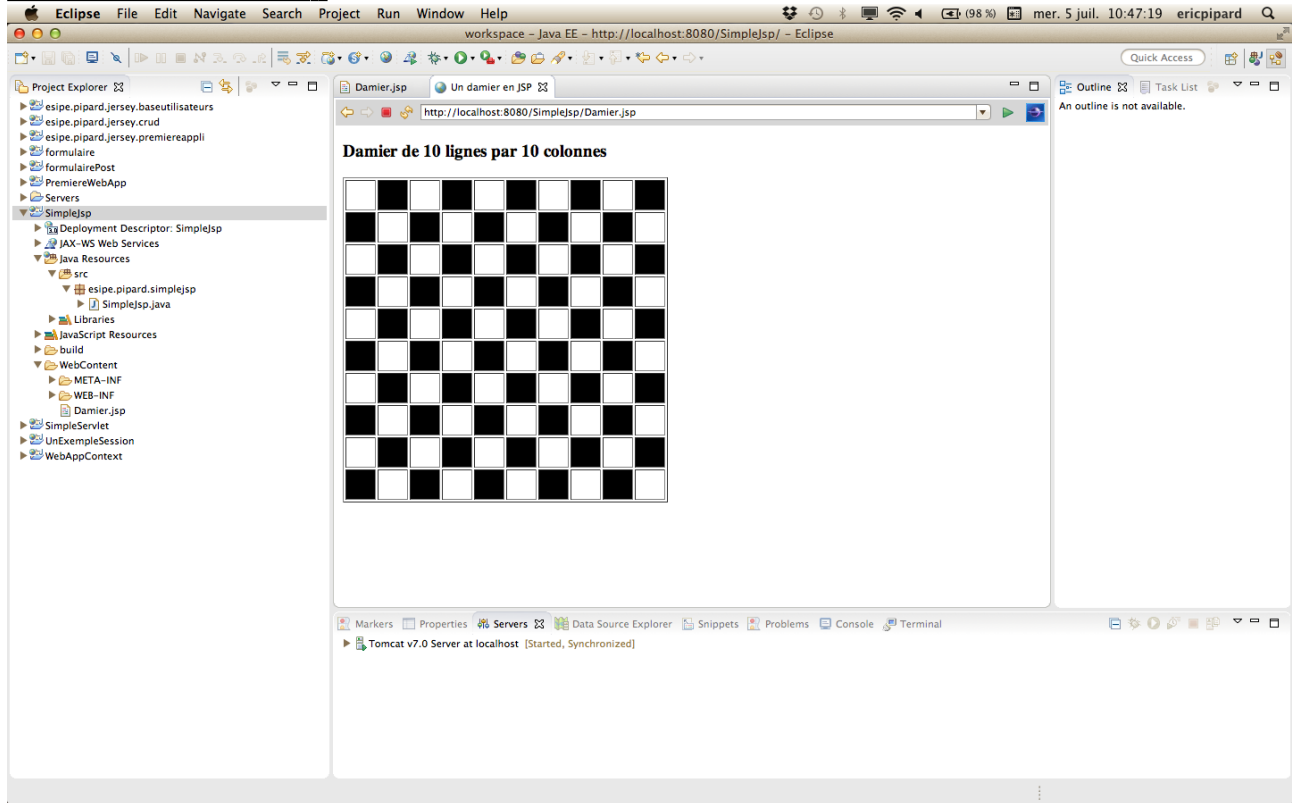
```
out.write("<html>\r\n");
out.write("    <head>\r\n");
out.write("        <meta charset=\"utf-8\" />\r\n");
out.write("        <title>Test</title>\r\n");
out.write("    </head>\r\n");
out.write("    <body>\r\n");
out.write("        <p>Ceci est une page générée depuis une
        JSP.</p>\r\n");
out.write("    </body>\r\n");
```

```
out.write("</html>");
```

4 – Création d'une page JSP

Créer une page nommée damier.jsp permettant d'afficher un damier de N lignes X N colonnes. L'affichage sera réalisé sous forme d'un tableau html de N lignes par N colonnes dont les cellules sont alternativement noires ou blanches. On choisira N=10. Le projet se nomme SimpleJsp.

Voici le résultat attendu :



5 – Passage d'information via l'URL

Créer une page nommée ArithmTable.jsp permettant d'afficher des tables arithmétiques. Le but est de visualiser dans le navigateur du client la table de N pour un opérateur précis ie la table de 7 pour la multiplication. L'opérateur et l'entier N seront récupérés via l'URL comme ceci :

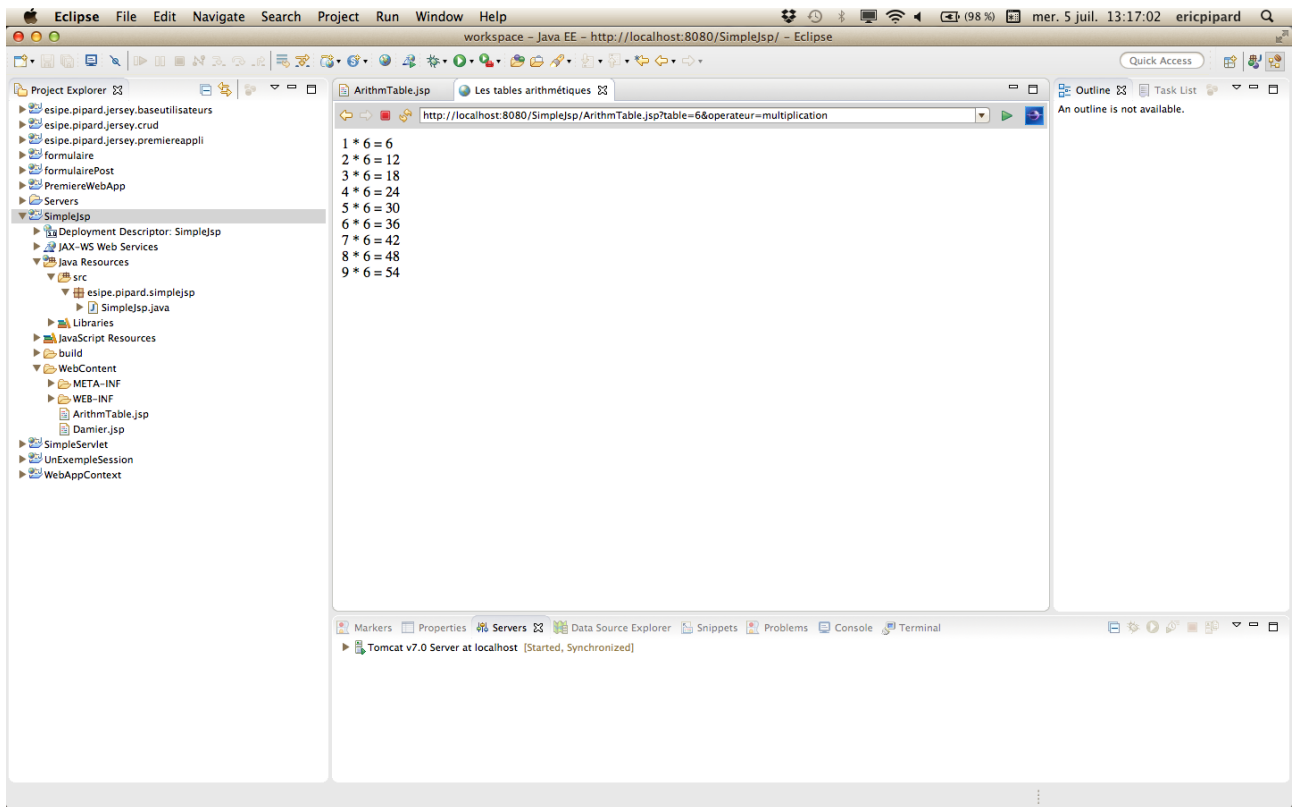
<http://localhost:8080/SimpleJsp/ArithmTable.jsp?table=6&operateur=multiplication>

Pour récupérer des informations via une URL il faut utiliser l'objet implicite `request`.

La page devra tester si N est bien un entier (`parseInt()`) compris entre 1 et 10. En cas d'erreur un booléen `erreurTable` sera positionné à `true` et un message d'erreur sera retourné au client.

La page devra tester si l'opérateur mentionné dans l'URL est une chaîne appartenant à l'ensemble {"addition", "soustraction", "multiplication", "division"}. En cas d'erreur un booléen `erreurOperateur` sera positionné à `true` et un message d'erreur sera retourné au client. Le projet se nomme SimpleJsp.

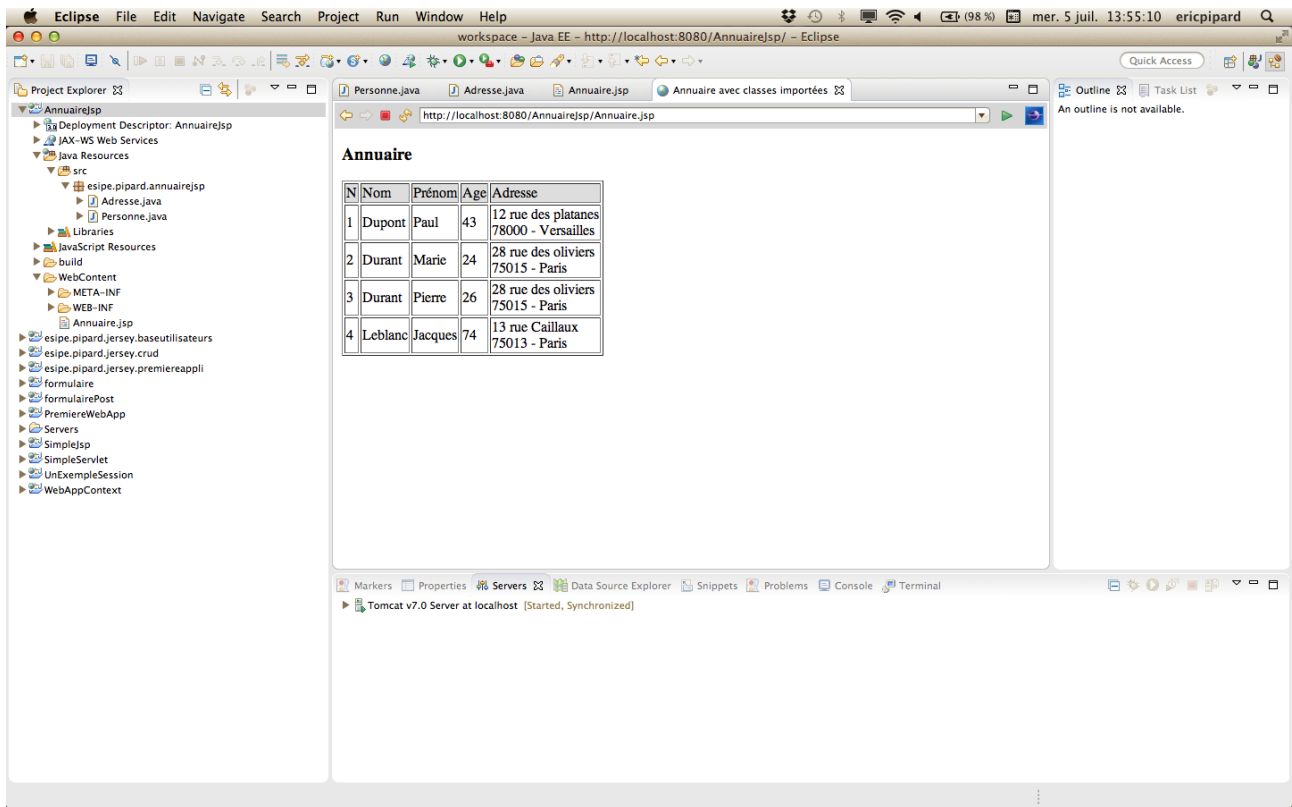
Résultat attendu :



6 – Import de classes personnalisées

Créer une page JSP permettant d'afficher un annuaire composé des nom, prénom, âge et adresse de plusieurs personnes. Vous devez réaliser 2 classes java pour modéliser les notions de personne et d'adresse et importer ces classes dans la JSP. Le projet se nomme Annuairejsp.

Résultat attendu :



7 – Déclaration de méthodes dans une page JSP

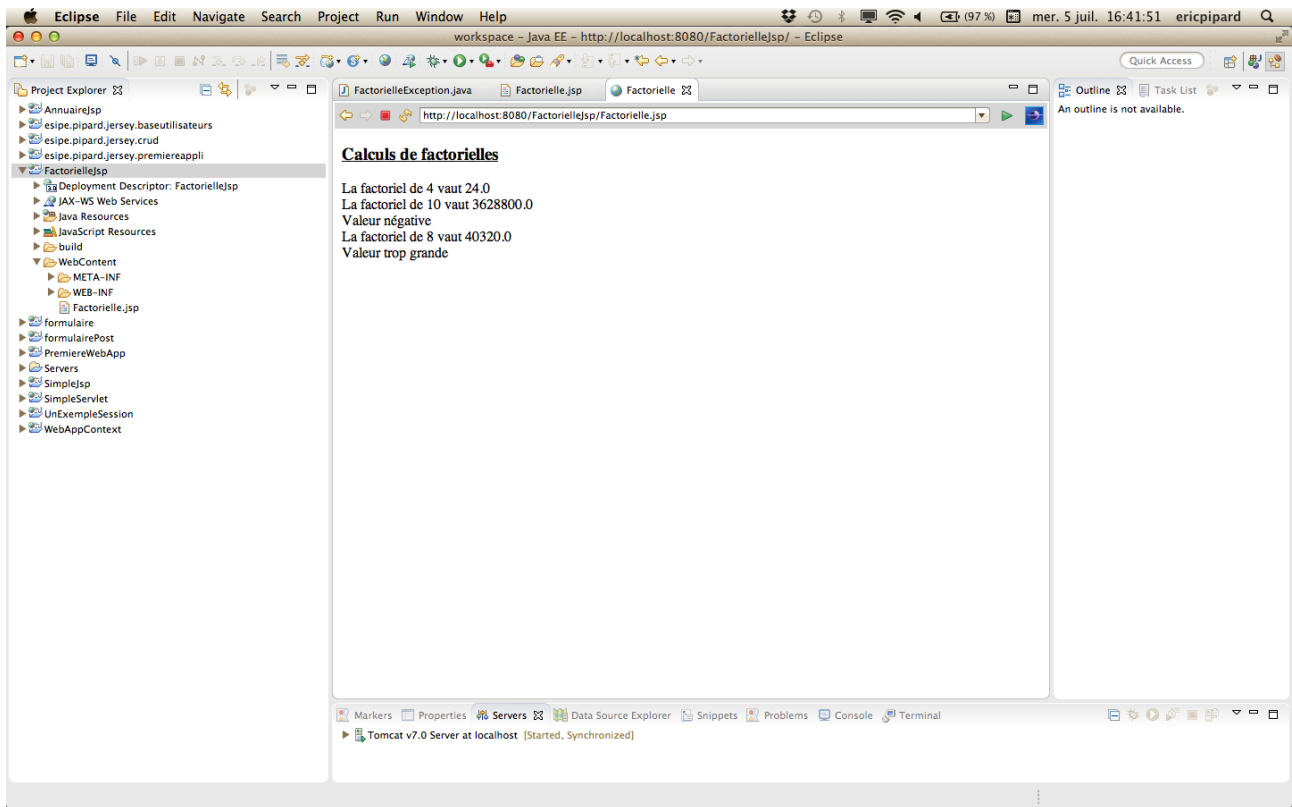
Créer une page JSP permettant de calculer la factorielle d'un entier n. Il faudra tester si n est un entier positif et inférieur à 75.

Vous devez réaliser une classe `FactorielleException` dérivant de `java.lang.Exception` ; cette classe possède uniquement un constructeur appelant la constructeur de la classe parente.

La page JSP comprendra une méthode factorielle qui retourne la factorielle de son argument et utilise `FactorielleException`.

La page JSP comprendra une seconde méthode `evaluerFactorielle` ; cette méthode appelle la méthode factorielle et récupère le résultat afin de fournir au client une phrase explicite ie "La factorielle de 4 vaut 24.0".

Résultat attendu :



8 – Balises ou actions personnalisées

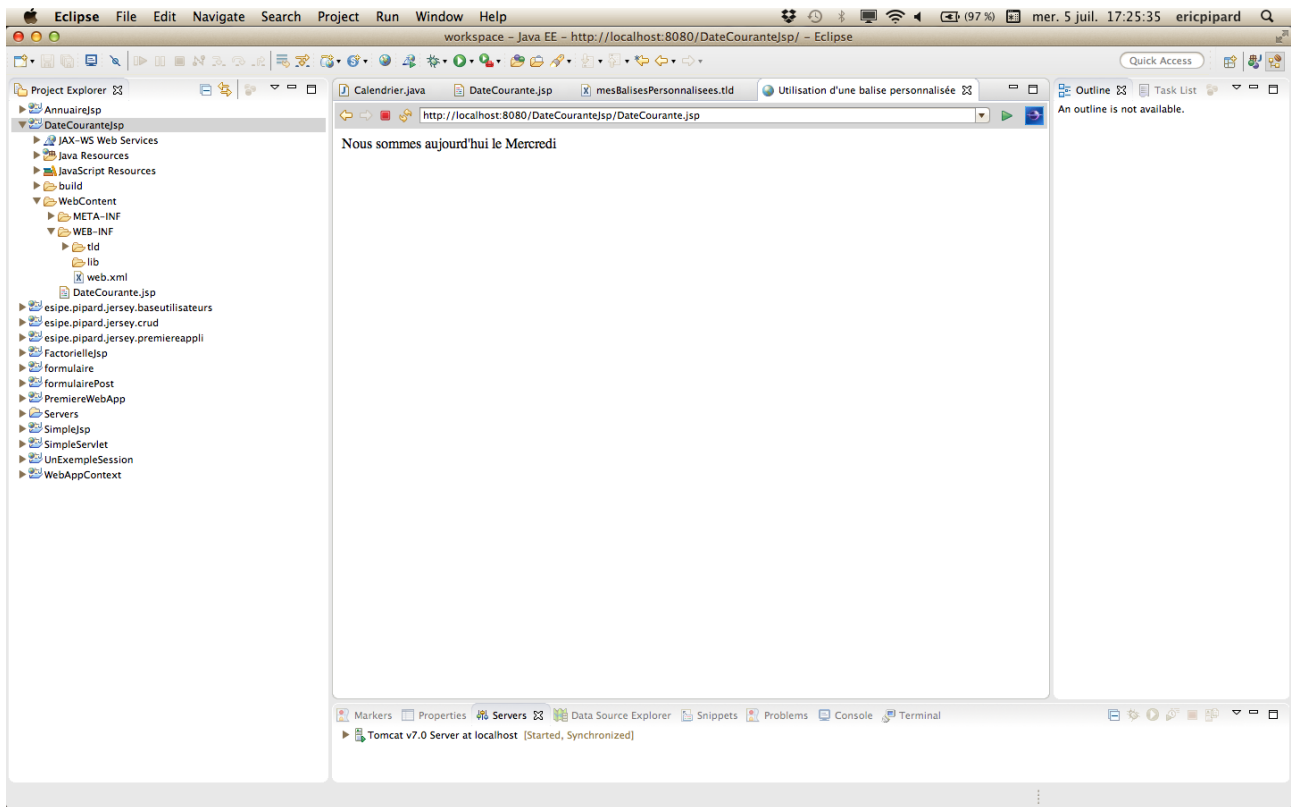
Il faudra utiliser dans une page JSP une balise personnalisée `<maBalise : dateCourante />` dans le but d'afficher le nom du jour de la semaine actuelle dans le navigateur du client.

Une balise est associée à une classe ; il faudra donc créer une classe `Calendrier`. Cette classe dérivera de `javax.servlet.jsp.tagext.TagSupport` et implémentera la méthode `doStartTag()`.

Ensuite il faudra créer un fichier d'extension `tld` (Tag Library Descriptor) chargé d'indiquer au moteur de servlets la correspondance entre le nom de l'action et la classe associée.

Créer une page JSP utilisant cette action personnalisée.

Résultat attendu :



9 – Inclusion de ressources externes dans une page

Créer un répertoire dans lequel vous placerez les ressources partagées.

Toutes les pages JSP de l'application doivent présenter un en-tête et un pied de page communs.

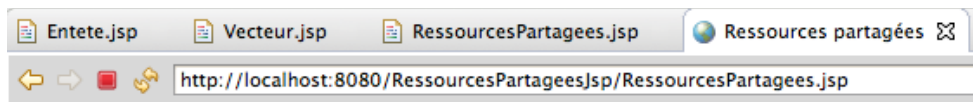
Placer dans le répertoire les ressources suivantes :

- un fichier css précisant que tout texte situé entre `<h3>` et `</h3>` doit être affiché en bleu
- une page JSP représentant l'en-tête commun des pages de l'application. Celle-ci fait appel à la feuille de style css et affiche le titre de l'application.
- une page JSP représentant le pied de page commun des pages de l'application. Celle-ci affiche la date du jour.

Créer une dernière ressource jsp partagée qui regroupe un ensemble de méthodes communes à l'application. Vous déclarerez une méthode retournant tous les multiples d'un entier n compris entre 1 et 1000.

Créer une page présentant la liste des multiples d'un entier n sous la forme d'un tableau html. Les paramètres sont passés via l'URL et la page utilise les ressources partagées.

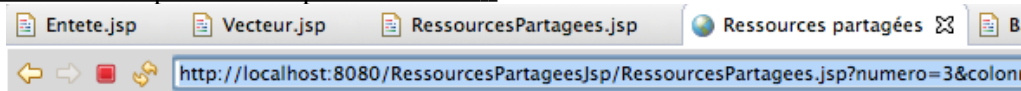
Résultat attendu sans paramètre passé à l'URL :



Bienvenue dans l'application "Ressources partagées"

Affichage des multiples de 2 dans un tableau de 5 colonnes.				
2	4	6	8	10
12	14	16	18	20
22	24	26	28	30
32	34	36	38	40
42	44	46	48	50
52	54	56	58	60
62	64	66	68	70
72	74	76	78	80
82	84	86	88	90
92	94	96	98	100
102	104	106	108	110
112	114	116	118	120
122	124	126	128	130
132	134	136	138	140

Résultat attendu avec paramètres passé à l'URL :



Bienvenue dans l'application "Ressources partagées"

Affichage des multiples de 3 dans un tableau de 4 colonnes.			
3	6	9	12
15	18	21	24
27	30	33	36
39	42	45	48
51	54	57	60
63	66	69	72
75	78	81	84
87	90	93	96

10 – Utilisation d'un bean

Vous devez dans une page html saisir un nombre puis afficher ses décalages <<, >> et >>>.

La page html enverra le formulaire à une page jsp qui utilisera un bean pour les calculs et pour afficher le résultat.

Une page jsp d'erreur indiquera que la valeur saisie n'est pas dans le bon format.

Résultats attendus :

