



Informatique & Mathématiques Appliquées

Création de tags personnalisés sous NetBeans

Philippe Genoud (<u>Philippe.Genoud@imag.fr</u>) lundi 21 janvier 2008

Ce tutorial vous guide dans l'utilisation de Netbeans (version 6.0) pour créer vos propres bibliothèques de balises personnalisées (custom tags).

Créer un projet d'application Web sous Netbeans

1) Créez un nouveau projet (menu File>New Project)

2) Sélectionnez la catégorie Web et le type de projet Web Application

3) Choisissez un nom de projet (TPCustomTags par exemple), un répertoire de stockage (Project Location), un serveur d'application (Apache Tomcat), une version de JEE (Java EE 5) et validez la création (bouton Finish).



Page 1/12 MCours.com Un projet d'application web est crée, sa hiérarchie est la suivante. Il contient une seule page index.jsp:



4) lancez l'exécution de ce projet (c'est-à-dire sa compilation et son déploiement dans le serveur Tomcat intégré à Netbeans) simplement en cliquant sur le bouton de la barre d'outil de Netbeans.



Si vous travaillez sur les terminaux SunRay et que vous vous connectez sur les serveurs imablade pour lancer Netbeans, il faut configurer le serveur Tomcat de Netbeans pour fixer des numéros de ports différents selon les utilisateurs. Donc avant de poursuivre référez vous à l'annexe *Configuration de Tomcat/Netbeans pour les serveurs imablade*.

	N	etBea	ans ID	E 6.0									
I	File	Edit	View	Navigate	Source	Refactor	Build	Run	Profile	Versioning	Tools	Window	Help
	P	Ľ) 🔡	5	% 🖣) (1	6 12		- 🤄) •	
	Pro	jects	4 0	× Files		Ser	vices		Profi	ler Run	Main Pro	oject (F6)	sp 🗙

Le serveur Tomcat est automatiquement lancé, et une l'application lancée dans votre navigateur. (Remarque : vous pouvez choisir le navigateur ouvert (Mozilla, IE, ...) à l'aide du menu **Tools>Options**).

C Options						
B	P	{ <mark>@</mark> }		a v A S		
General	Editor	Java Code	Fonts & Colors	Keymap	Miscellaneous	
Web Browser:	<default< td=""><td>System Browser</td><td>></td><td>onts & Colors Keymap Miscellaneous</td><td>Edit</td></default<>	System Browser	>	onts & Colors Keymap Miscellaneous	Edit	
	<default 9<="" td=""><td>5ystem Browser:</td><td>></td><td></td><td></td><td></td></default>	5ystem Browser:	>			
	Internet E	xplorer				
Proxy Settings:	Firefox	ол,				

Votre navigateur affiche la page index.jsp (qui ne propose pas grand-chose ... pour le moment).



Créer un premier tag personnalisé : HelloTag

Vous allez maintenant créer un premier tag personnalisé qui affichera tout simplement le texte *Bonjour Monde Cruel !!!* en gras. Tout d'abord il vous faut définir la librairie de tags à laquelle sera associé votre nouveau tag. Pour cela procéder comme suit :

1) Créez un nouveau fichier de type Tag Library Descriptor (TLD). Soit par le menu File>New File, le bouton New File de la barre d'outils ou un clic droit sur le projet dans le navigateur. Dans la catégorie Web sélectionnez le type de fichier Tag Library Descriptor.

New File		X
Steps 1. Choose File Type	Choose File Type Project:	~
2	Categories: File Types: Web Struts JavaServer Face: JavaScript File Java JavaScript File Junit Tag Handler Tag Library Descriptor Tag File Description: Creates a Tag Library Descriptor (TLD) file, which can be used with JavaServer Pages (JSP) files. A tag library descriptor is an XML document that contains information about a library as a whole and	
	< Back Next > Finish Cancel Help	

Un assistant (wizard) de création de fichier TLD s'affiche. Renseignez le nom de la bibliothèque de tags (par exemple avec maTagLib) et validez (bouton Finish les autres valeurs proposées par défaut par NetBeans sont suffisantes).

Steps	Name and Location					
Choose File Type Name and Location	TLD Name: maTagLib Project: TPCustomTags]				
	Location: Web Pages Folder: WEB-INF/tlds/	Browse				
TPCustomTags Web Pages Web INF WEB-INF Uds Index.jsp Configuration Files	Created File: 4tdCustomTags\TPCustomTags\web\WEB-IN URI: /WEB-INF/tlds/maTagLib Prefix: mataglib	ıF\tlds\ma⊺agLib.				
MANIFEST.MF	<back next=""> Finish Cance</back>	Help				

Un fichier maTagLib.tld est créé dans le répertoire Web-INF/tlds de votre application et est ouvert dans l'éditeur.

2) Il vous faut maintenant créer votre gestionnaire de tag (*tag handler*). Créez un nouveau fichier de type Tag Handler... (comme précédemment vous passerez par le menu File>New File, le bouton New File de la barre d'outils ou un clic droit sur le projet dans le navigateur).

L'assistant de création de gestionnaire de tag s'affiche. Dans le premier panneau vous préciserez le nom de votre gestionnaire de tags (par exemple HelloTag) et son package (par exemple mestags).

🗊 New Tag Handler		
Steps	Name and L	ocation
 Choose File Type Name and Location 	Class Name:	HelloTag
3. TLD Information	Project:	ne and Location is Name: iect: TPCustomTags ation: Source Packages ation: Source Packages kage: mestags ated File: ursJAVA\tds\sujets\tdCustomTags\TPCustomTags\src\java\mestags\Hel Support Class to Extend: SimpleTagSupport SimpleTagSupport (J2EE 1.4) BodyTagSupport seates a tag handler that extends javax.servlet.jsp.tagext.SimpleTagSupport.
	Location:	Source Packages
	Package:	mestags 💌
	Created File:	$urs JAVA \ tds \ sujets \ tdC us to mTags \ TPC us to mTags \ src \ java \ mestags \ Hel$
	Tag Support (Class to Extend: jSupport (J2EE 1.4) Support
	Description:	
	Creates a tag	handler that extends javax.servlet.jsp.tagext.SimpleTagSupport.
		< Back Next > Finish Cancel Help

Passez à l'étape suivante (bouton Next) et renseigner les différents champs pour le descripteur de votre tag personnalisé dans le fichier .tld de votre bibliothèque :

- Le nom du tag (par exemple hellotag),
- Le type de corps (*body*), ici empty,
- Le fichier TLD, vous sélectionnerez le fichier maTagLib.tld créé précédemment (bouton Browse).

Un package mestags et un fichier HelloTag.java sont crée dans votre projet et un éditeur est ouvert sur le fichier HelloTag.java.

Steps	TLD Information
Choose File Type Name and Location TLD Information	Add Corresponding Tag to the Tag Library Descriptor Specify the tag library descriptor that the tag should be added to: TLD File: Browse
	Tag Name: heliotag 🕕 Browse Files
	Tag Handler Class: mestags.HelloTag
TPCustomTags	Body Content: empty scriptless ta Attributes: Name Type Required Web Pages META-INF WEB-INF WEB-INF WEB-INF WEB-INF META-INF META-INF
index.jsp Configuration Files MANIFEST.MF Context.xml	The Tag Library Descriptor (TLD file) need to be spe one must be created first. The Tag Library Descripto purpose. Select File Cancel
Server Resources	< Back Next > Finish Cancel Help
Source Packages	
mestags	

3) Modifiez comme suit le code de la méthode doTag de votre classe HelloTag.java afin que celle-ci écrive Bonjour Monde Cruel !!! sur la sortie de votre page JSP.

```
public void doTag() throws JspException {
    JspWriter out=getJspContext().getOut();
    try {
        out.println("<strong>Bonjour Monde Cruel !!!</strong>");
    } catch (java.io.IOException ex) {
        throw new JspException(ex.getMessage());
    }
}
```

Sauvegardez le fichier !

4) Modifiez la page index.jsp. Ajoutez la directive taglib pour déclarer la bibliothèque de tag et l'invocation de votre nouveau tag hellotag.



5) Sauvegardez le fichier index.jsp et relancez votre application (bouton Run de la barre d'outils).

NetBeans IDE	6.0								
File Edit View N	lavigate Source Refactor Build Run Profile Versioning Tools Window Help								
🖀 🔚 😫									
Projects 🔍 🛪	Projects 🔍 × 🥹 JSP Page - Mozilla Firefox 📃 🗖 🗙								
Eichier Édition <u>A</u> ffichage <u>H</u> istorique <u>M</u> arque-pages <u>O</u> utils <u>?</u>									
 									
ISP Page ISP Page									
	Hello World! Bonjour Monde Cruel !!!								
	Terminé	-							

Ajout d'un attribut au tag HelloTag.

Dans ce qui suit vous allez modifier votre gestionnaire de tag de manière à ce que l'utilisateur puisse paramétrer le message afficher à l'aide d'un nom.

1) Modifiez le fichier maTagLib.tld pour rajouter la description d'un attribut pour le tag HelloTag.



Sauvegardez le fichier maTagLib.jsp

2) Modifiez le gestionnaire de Tag (HelloTag.java) pour y ajouter le support de l'attribut name.



Sauvegardez le fichier HelloTag.jsp

3) Modifiez la page index.jsp pour y ajouter un appel à votre tag avec un paramètre.

MCours.com



4) Sauvegardez le fichier index.jsp et relancez l'application (bouton Run de la barre d'outils).



Exercices

Les exercices suivants reprennent les exemples présentés à la fin du cours (voir les transparents en ligne <u>http://www.inrialpes.fr/helix/people/genoud/ENSJAVA/cours/supportsPDF/JSP_2pp.pdf</u>).

1) Rajoutez à votre bibliothèque de tags un nouveau tag qui admet un corps. Ce tag admet un paramètre (obligatoire) name de type chaîne de caractères. Le résultat produit est l'encadrement du texte situé à l'intérieur du corps du tag par HELLO name et BYE name comme le montre la figure ci-dessous.

	😻 Exemple d'une Page JSP avec un 🔳 🗖 🔀				
 <mestags:helloavecbodytag name="TITI"> Ceci est un texte pour le corps (body) du TAG pour montrer comment cela marche </mestags:helloavecbodytag> 	Eichier Édition Affichage Historique Marque-pages				
	Ceci est un texte pour le corps (body) du TAG pour montrer comment cela marche BYE TITI				
	Terminé 🕥 .;				

2) Après avoir tester votre tag précédent modifiez le, de manière à ce que le texte affiché soit transformé en majuscule.

<pre> <mestags:helloavecbodytag name="TITI"> Ceci est un texte pour le corps (body) du TAG pour montrer comment cela marche</mestags:helloavecbodytag></pre>	Exemple d'une Page JSP avec un tag personalisé Image: Chier Édition Affichage Historique Marque-pages Outils ? Image: Chier Édition Affichage Historique Marque-pages Qutils ? Image: Chier Affichage Historique Marque-pages Qutils ? Image: Chier Affichage Historique Marque-pages Qutils ? Image: Chier Affichage Historique Marque-pages Qutils ? Image: Chier Affichage Historique Marque-pages Page: Chier Affichage Historique Marque-pages Qutils ? Image: Chier Affichage Historique Marque-pages Qutils ?
	HELLO TITI CECI EST UN TEXTE POUR LE CORPS (BODY) DU TAG POUR MONTRER COMMENT CELA MARCHE
	BYE TITI

Indication : on utilisera un objet StringWriter comme paramètre dans le message invoke envoyé au corps du tag pour récupérer la chaîne correspondant au corps, chaîne qui sera ensuite transformée en majuscules (méthode toUpperCase() de la classe String) avant d'être envoyé sur la sortie de la page Jsp.

3) Ecrire deux tags switchtag et casetag qui lorsqu'ils sont imbriqués reproduisent le fonctionnement d'une instruction switch case des langages de programmation.



Annexe

Configuration de Tomcat/Netbeans pour les serveurs imablade.

Lorsque vous utilisez le serveur Tomcat de Netbeans, celui-ci utilise par défaut les ports 8084 et 8025. Si plusieurs Netbeans tournent sur une même machine (ce qui est le cas lorsque vous travaillez sur les serveurs d'application de l'UFR IMA), il faut attribuer des numéros de port différents pour chacun des serveurs Tomcat. Pour cela procéder comme suit :

1) Sélectionnez l'onglet Services (si il n'est pas présent, vous pouvez l'afficher par le menu Window>Services).

2) Développez le nœud Servers (en cliquant dessus) et selectionnez le nœud Apache Tomcat.

3) Ouvrez le fenêtre de configuration des propriétés du serveur (clic droit sur le noeud Tomcat et sélection de l'item **Properties** dans le menu contextuel). Dans celle-ci fixez les numéros de ports (Server Port **ET** Shutdown Port) en choisissant des numéros qui ne sont pas utilisés.



Au cas où les numéros que vous auriez choisis serait déjà utilisés un message d'erreur est affiché sur la console au lancement de Tomcat.

Outp	out
\square	Java DB Database Process x Glass Fish V2 x Apache Tomcat 6.0.14 Log x Apache Tomcat 6.0.14 x WebApplication2
	<pre>Inbrary-inclusion-in-manifest: compile: compile-jsps: Starting of Tomcat failed, the server port 8084 is already in use. Deployment error: Starting of Tomcat failed, the server port 8084 is already in use. See the server log for details. at org.netbeans.modules.j2ee.deployment.devmodules.api.Deployment.deploy(Deployment.java:163) at org.netbeans.modules.j2ee.ant.Deploy.execute(Deploy.java:104) at org.apache.tools.ant.UnknownElement.execute(UnknownElement.java:288) at sun.reflect.GeneratedMethodAccessor65.invoke(Unknown Source) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:597) at org.apache.tools.ant.Task.perform(Task.java:348) d org.apache.tools.ant.Task.perform(Task.java:348)</pre>

Par ailleurs, sur les terminaux SunRay il faut configurer le navigateur qui sera ouvert lorsqu'une application Web est lancée. Pour cela sélectionnez le menu **Tools>Options** et dans la fenêtre de configuration qui apparaît sélectionnez le navigateur Mozilla.

$\widehat{\mathbf{O}}$		Optio	ns				8
General Editor Iav	المن المن المن المن المن المن المن المن	Fonts & Colors	Keymap	C/C++		Miscellaneous	
Web Browser:	Mozilla		(c) map	.,		Edit]
Proxy Settings: O <u>N</u> o Proxy							
	Manual Pro HTTP Prox More	y:				Po <u>r</u> t:]
Advanced Opt	ions			ок	Canc	el <u>H</u> elp	



Page 12/12