

# Table des Matières

Développement Applications Web.....	2
Représentation d'Applications Web dans l'EDI.....	3
Vue Project des Applications Web.....	3
Vue des Fichiers d'Applications Web.....	3
Structure Application Web.....	4
Ajout de Fichiers et Bibliothèques à Votre Application Web.....	5
Création de Pages Web et Autres Fichiers Publics.....	7
Création des Classes, Servlets et Autres Fichiers Privés.....	7
Ajout de Ressources Externes à Votre Projet.....	8
Ajout de Tags et Bibliothèques de Tag.....	9
Edition et Refactoring Fichiers Applications Web.....	10
Completion de Tags.....	10
Extension des Abréviations pour les fichiers JSP.....	12
Edition Manuelle du Descripteur de Déploiement.....	13
Refactoring Composants Web.....	14
Déploiement d'une Application Web.....	15
Personnalisation Contenu d'un Fichier WAR.....	15
Non-Déploiement d'une Application Web.....	16
Redéployer une Application Web.....	16
Création d'un Fichier WAR.....	16
Déploiement vers un Autre Serveur Tomcat.....	16
Test et Débogage de Votre Application Web.....	17
Vérification des Erreurs JSP.....	18
Visualiser la Servlet d'un Fichier JSP.....	18
Visualiser un Fichier dans un Navigateur Web.....	18
Passage de Paramètres de Requête à une Application Web.....	19
Débogage JSP et Fichiers Tag.....	19
Création et Déploiement d'Applettes.....	19
Création d'une Applette.....	20
Exécution et Débogage d'une Applette dans l'Applet Viewer.....	20
Exécution d'une Applette dans un Navigateur Web.....	21
Mettre l'Applet dans un Fichier JAR.....	21
Mettre l'Applet dans un Fichier WAR.....	21
Définition des Permissions de l'Applette.....	22
Modification du Navigateur Web par Défaut de l'EDI.....	22
Monitoring Transactions HTTP.....	23
Mise En Place du Moniteur HTTP.....	23
Mise En Place du Moniteur HTTP Pour les Serveurs Démarrés En Dehors de l'EDI.....	24
Analyse des Données Récoltées.....	26
Rejouer des Requêtes HTTP.....	29

# Développement Applications Web

L'EDI NetBeans est un environnement idéal pour le développement d'applications Web. L'EDI élimine tout un tas de nuisances que vous auriez normalement rencontrés, particulièrement dans la définition de l'application et dans les étapes entre le codage, déploiement, débogage et redéploiement de votre application. Et parce que Ant est à la base de l'automatisation, il n'y a pas de mystères propriétaires que vous devez refaire si vous désirez faire que votre projet fonctionne sans l'EDI comme intermédiaire.

Voici une liste de choses que l'EDI fait pour simplifier le développement d'Applications Web:

- [nl] Fournit un serveur Web Tomcat pour déployer, tester et déboguer vos applications.
- [nl] Définit le fichier et la structure de dossier d'une application web pour vous.
- [nl] Génère et maintient le contenu des descripteurs de déploiement, y compris l'enregistrement de toutes servlettes que vous rajoutez à votre projet.
- [nl] Génère et maintient un script Ant avec les targets (commandes) pour la compilation, nettoyage, test, création de fichier WAR et déploiement sur un serveur. Ce script vous évite d'avoir à copier manuellement les fichiers dans le serveur Web.
- [nl] S'assure que les fichiers de configuration qui apparaissent dans le dossier WEB-INF de votre application ne sont pas effacés lorsque vous exécutez la commande Clean pour enlever les résultats des builds précédents.
- [nl] Fournit une coloration syntaxique, completion de code, et d'autres aides pour l'édition de servlettes, JSP, HTML et fichiers de bibliothèque de tag.
- [nl] Fournit la commande Compiler JSP, qui vous permet de détecter les erreurs de syntaxe dans les fichiers JSP avant de déployer vers votre serveur, que l'erreur survienne à la compilation ou durant la traduction du fichier JSP en une servlette.
- [nl] Fournit un support de débogage compréhensif, incluant le mode pas à pas dans les fichiers JSP et le traçage des requêtes HTTP.

Most of the topics in this chapter assume that you are using the Tomcat Web server, but it is also possible to use the Sun Java System Application Server, which supports enterprise-tier applications and provides broader support for Web services. See Chapter XXX for more information on working with the Sun Java System Application Server and XXX for information on developing, exposing, and consuming Web services.

Ce chapitre se focalise sur les problèmes spécifiques aux applications Web, comme la création et l'édition de composants Web, le débogage des transactions HTTP, etc., mais ne fournit aucune information sur la création de projet. Consultez le Chapitre 3 pour obtenir des informations quant à la création de projets.

---

## Représentation d'Applications Web dans l'EDI

Les applications Web sont basées sur une architecture quelque peu complexe où le layout des fichiers au moment du développement diffère de celui de l'application compilée. L'EDI vous aide à contrôler ce processus en :

- [lb] Fournissant une vue orientée développement de votre projet dans la fenêtre Projects. Cette vue vous donne un accès aisé à vos sources et informations concernant votre classpath, mais cache les résultats du build et les méta-données du projet. Peut-être le point le plus important, le fait de travailler dans la fenêtre Projects assure qu'aucun des fichiers que vous créez ne seront effacés par inadvertance lorsque vous exécutez la commande Clean sur votre projet.
- [lb] Fournissant une vue orientée Fichier de votre projet dans la fenêtre Files. Cette fenêtre est particulièrement utile pour accéder et personnaliser vos scripts de déploiement et pour naviguer parmi ce qui résulte de votre projet, comme le fichier WAR de votre projet et son contenu.
- [lb] Créant et maintenant un script Ant, qui est utilisé lorsque vous exécutez des commandes typiques comme Build Project et Run Project. Entre autres choses, le script Ant automatise le placement de vos fichiers dans l'application build, la packaging de ces fichiers dans un fichier WAR, et le déploiement vers le serveur spécifié.

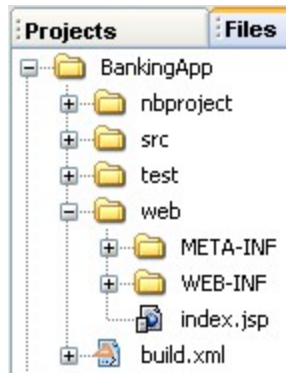
### Vue Project des Applications Web

La fenêtre Projects fournit une représentation “logique” de la structure de source de l'application, avec des noeuds pour les objets suivants:

- [lb] Web Pages (pour les fichiers HTML, JSP, et images auxquels les utilisateurs de l'application pourront directement accéder via leur navigateur Web)
- [lb] Source Packages (pour les paquetages des sources Java, qui contiendra les servlets et d'autres classes Java).
- [lb] Test Packages (pour les tests unitaires)
- [lb] Configuration Files (pour vos descripteurs de déploiement et d'autres fichiers)
- [lb] Web Services (où vous pouvez créer et enregistrer des services Web – voir le chapitre XXX pour plus de détails sur le développement de Web Services)
- [lb] Libraries (où vous pouvez ajouter des bibliothèques ou inclure les résultats d'autres projets de l'EDI)
- [lb] Test Libraries (où vous pouvez ajouter toutes bibliothèques nécessaires à l'exécution des tests unitaires de votre application)

### Vue des Fichiers d'Applications Web

Si vous ouvrez la fenêtre Files, vous verrez l'organisation “physique” des fichiers sur le disque (comme montré sur l'illustration 6-1) et trouverez que cette organisation est alignée sur les recommandations d'Apache pour l'organisation des fichiers sources (voir Structure d'Application Web à la page 4). L'EDI ajoute également les dossiers `nbproject` (pour accueillir les méta-données du projet) et `test` (pour les tests unitaires), mais ils ne sont pas inclus dans le résultat final de l'application.

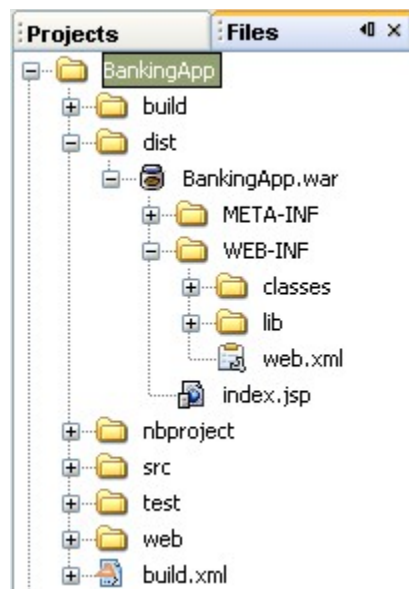


### Illustration 6-1

*Fenêtre Files avec une structure de dossiers d'une application Web*

Lorsque vous faites un build de l'application, que ce soit via l'EDI ou en exécutant directement Ant, un dossier `build` est créé pour accueillir les classes compilées et le dossier `dist` est créé pour accueillir le fichier WAR généré comme montré dans l'illustration 6-2.

### Illustration 6-2



*Fenêtre Files montrant la structure du fichier WAR créé*

## Structure Application Web

Apache Jakarta fournit un "guidelines" de comment structurer vos applications Web pour s'assurer qu'elles fonctionnent proprement avec le serveur Tomcat. Lorsque vous créez un projet dans l'EDI, cette structure est respectée.

Ce qui suit est un rapide survol des éléments importants de la structure de l'application Web construite:

[lb] Le dossier racine (connu comme "document base"), qui contient tous les

autres fichiers et dossiers de l'application.

- [lb] Fichiers qui sont directement accessibles aux utilisateurs de l'application via leur navigateur web, comme les fichiers HTML, les images et les fichiers JSP.
- [lb] Le dossier `WEB-INF`, qui contient le fichier de descripteur de déploiement (`web.xml`) et les dossiers `classes`, `lib`, `tags`, ainsi que d'autres dossiers et fichiers. Le contenu de `WEB-INF` comprend le "bulk" de l'application est n'est pas directement accessible aux utilisateurs.
- [lb] Le dossier `classes` contient les fichiers des classes et servletes compilées avec leur hiérarchie de paquetage se reflétant par les sous-dossiers.

Vous pouvez trouver plus de détails sur la structure de source Tomcat à la page <http://jakarta.apache.org/tomcat/tomcat-5.0-doc/appdev/source.html>.

Voyez le Tableau 6-1 pour savoir comment les nombreux éléments de source d'une Application Web correspondent avec leur représentation dans l'EDI et où ils se situeront finalement dans l'application déployée.

Tableau 6-1: Matrice des Elements d'Application Web et Leur Représentation dans l'EDI

Contenu	Représentation dans la fenêtre Projects	Représentation dans la fenêtre Files	Emplacement dans le fichier WAR généré (situé dans le dossier dist)
Pages Web	Noeud Web Pages	Dossier <code>web</code>	Racine du fichier
Fichiers Sources Java, Servletes, ...	Noeud Source Packages	Dossier <code>src</code>	Dossier <code>WEB-INF/classes</code>
Tests Unitaires	Noeud Test Packages	Dossier <code>test</code>	N/A
Descripteurs de déploiement ( <code>web.xml</code> )	Noeud Configuration Files	Dossier <code>web/WEB-INF</code>	Dossier <code>WEB-INF</code>
Fichier de configuration de contexte Tomcat ( <code>context.xml</code> )		Dossier <code>web/META-INF</code>	Dossier <code>META-INF</code>
Bibliothèques	Noeud Libraries	Dossier <code>web/WEB-INF/lib</code>	Dossier <code>WEB-INF/lib</code>
Entrées classpath de Test	Noeud Test Libraries	Dossier <code>test</code>	N/A
Méta-données du projet y compris le script Ant	Boîte de dialogue Project Properties, que vous pouvez ouvrir en cliquant-droit sur le noeud du projet et choisir Properties.	<code>build.xml</code> file, <code>nbproject</code> folder	N/A

---

## Ajout de Fichiers et Bibliothèques à Votre Application Web

Une fois que vous avez créé un projet Web via l'assistant New Project, vous pouvez commencer à y mettre des pages Web et du code.

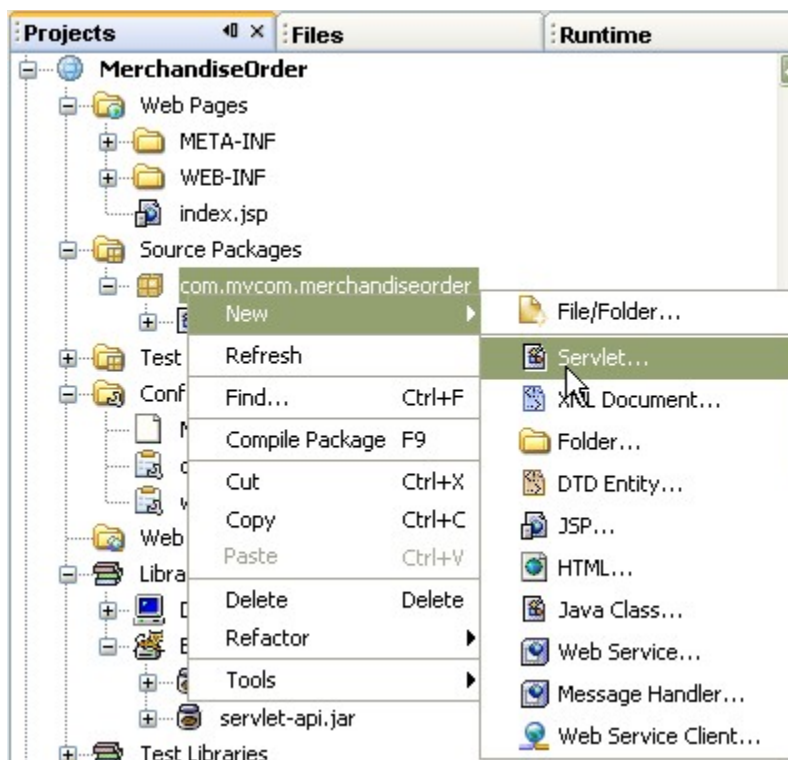
Le moyen le plus direct de créer des fichiers est d'ouvrir la fenêtre Projects et de cliquer-droit sur le noeud du projet ou le dossier spécifique où vous désirez placer le fichier et de choisir New et

ensuite un modèle depuis le sous-menu. Voir Illustration 6-3. Un assistant apparaît pour le modèle vous permettant d'indiquer son nom et d'autres caractéristiques du fichier. En général, l'assistant vous guide pour s'assurez que les fichiers soient placés dans un répertoire approprié, pour suivre la structure d'une application Web bien conçue.

### Conseil EDI NetBeans

Le sous-menu New du menu contextuel d'un noeud affiche directement une sélection de modèles couramment utilisés. Si vous désirez voir tous les modèles disponibles, choisissez New | File/Folder.

Les modèles sont groupés sous différentes catégories, comme Java Classes, Web, Database. En plus des modèles pour les servlets et pages JSP, la catégorie Web contient des modèles pour les filtres, Web application listeners, fichiers tag, et tag handlers, qui contient des exemples de code utiles pour ces types de fichier spécifiques.



**Illustration 6-3**

*Création d'une nouvelle servlete depuis le menu contextuel d'un paquetage*

Conceptuellement, les fichiers dans une application Web se groupent en différents types de fichiers, dont sa fonction détermine l'endroit où vous allez placer le fichier dans l'application. Les types principaux de fichiers que vous avez besoin de rajouter au projet sont:

- [lb] Les pages Web et autres fichiers publics, c'est-à-dire les fichiers auxquels les utilisateurs de l'application peut directement accéder via leurs navigateur Web. Typiquement, cela inclut les fichiers JSP, les fichiers HTML, et les fichiers images.
- [lb] Les fichiers privés, c'est-à-dire les fichiers qui ne sont pas directement visibles à l'utilisateur final et qui fait le "bulk" du traitement dans l'application. Cela inclut

les classes Java, les fichiers JSP, les servletes et les bibliothèques de Tag et aboutissent dans le dossier `web/WEB-INF/classes` de l'application Web compilée.

- [lb] Les ressources externes, les fichiers créés en dehors du projet dont dépendent les fichiers dans le projet. Cela inclut les bibliothèques de tags, les fichiers JAR résultant des autres projets de l'EDI, et d'autres fichiers JAR et sont gardés dans le dossier `WEB-INF/lib` de l'application Web compilée.

De plus, il y a des fichiers de configuration comme le descripteur de déploiement (`web.xml`) et les fichiers spécifiques au serveur que vous déployez, mais l'EDI génère et maintient ces fichiers pour vous. Par exemple, si vous créez une servlet dans l'EDI, la servlet est enregistrée dans le fichier `web.xml` automatiquement.

### **Conseil EDI NetBeans**

Les bibliothèques de Tag peuvent être rajoutée à votre application Web comme bibliothèque ou comme source. Voir Ajout Tags et Bibliothèques de Tag à la page XXX pour plus d'informations sur la façon de travailler avec les bibliothèques de tags.

## **Création de Pages Web et Autres Fichiers Publics**

Généralement, vous ajoutez des pages Web et d'autres fichiers publics directement dans le noeud Web Pages de la fenêtre Projects ou dans le dossier que vous avez créé sous ce noeud. Lorsque l'application est construite, ces fichiers sont placés par le script Ant du projet dans le dossier `web` de l'application.

Pour ajouter un fichier HTML visible publiquement, cliquez-droit sur le noeud Web Pages et choisissez New | HTML.

Pour ajouter un fichier JSP visible publiquement, cliquez-droit sur le noeud Web Pages et choisissez New | JSP. L'assistant vous permet de spécifier si le fichier JSP utilise la syntaxe standard (et a l'extension `.jsp`) ou utilise la syntaxe XML (a l'extension `.jspx`). Vous pouvez également créer le fichier en tant que fragment (en utilisant l'extension `.jspxf`), que vous référencerez depuis d'autres pages avec les instructions `include`.

## **Création des Classes, Servletes et Autres Fichiers Privés**

Comme pour les projets Java généraux, les classes sont organisées dans les paquetages sous le noeud Source Packages. Pour la plupart des projets, ce noeud correspond avec un dossier sur votre disque nommé `src`. Lorsque l'application est générée, ces fichiers sont placés par le script ant dans le répertoire `WEB-INF/classes` de l'application.

Pour ajouter une classe à votre projet, cliquez-droit sur le noeud Source Packages ou le noeud d'un paquetage spécifique et choisissez New | Java Class. Si vous n'avez pas créé de paquetage pour la classe que vous rajouter, vous pouvez encore le faire dans l'assistant au moment où vous créez la classe.

Pour ajouter une servlet à votre projet, cliquez-droit sur le noeud Source Packages ou le noeud du paquetage spécifique et choisissez New | Servlet. L'assistant pour la création de la servlet vous guide également dans l'enregistrement de la servlet dans le descripteur de déploiement de l'application (fichier `web.xml`).

Si vous désirez rajouter un fichier basé sur un modèle plus spécifique, cliquez-droit sur un noeud de paquetage et choisissez New | File/Folder pour obtenir une liste plus étendue de modèles, y compris pour les composants JavaBeans, les fichiers `.properties`, les fichiers XML, et les

composants Web spécialisés. De plus, la plupart des modèles Web ont un squelette très utile et des suggestions dans les commentaires pour vous aider à développer ces types d'objets.

Voyez le tableau 6-2 pour une liste de modèles que vous pouvez retrouver dans la catégorie Web de l'assistant New File.

*Tableau 6-2: Modèles Web*

Modèles	Description
JSP	Vous permet de créer un fichier JSP (syntaxe standard), un document JSP (syntaxe XML), ou un fragment JSP (qui peut être référencé statiquement depuis un autre fichier).
Servlet	Crée une classe Java qui étend la classe <code>HttpServlet</code> . Vous permet également d'enregistrer la servlet dans le fichier de descripteur de déploiement ( <code>web.xml</code> ) du projet.
Filter	Crée une classe Java qui implémente l'interface <code>javax.servlet.Filter</code> . Les filtres vous permet de modifier les requêtes HTTP pour une servlet et les réponses de la servlet. Dans l'assistant, vous pouvez créer soit un filtre élémentaire, ou un qui englobe les objets <code>ServletRequest</code> et <code>ServletResponse</code> .  Dans l'assistant du modèle, vous pouvez enregistrer le filtre dans le fichier descripteur de déploiement ( <code>web.xml</code> ) du projet.
Web Application Listener	Crée une classe Java qui implémente un ou plusieurs des interfaces listeners disponibles pour les servlets, comme <code>ServletContextListener</code> et <code>HttpSessionListener</code> . Selon les interfaces que vous sélectionnez dans l'assistant, la classe créée sera à l'écoute des événements, comme lorsque les contextes de servlets est initialisé ou détruit, la session de servlet est créé ou détruite, ou lorsque les attributs sont rajoutés ou supprimés du contexte ou de la session.  Dans l'assistant du modèle, vous pouvez enregistrer le listener dans le fichier descripteur de déploiement ( <code>web.xml</code> ) du projet.
Tag Library Descriptor	Crée un descripteur pour une bibliothèque de tag personnalisée. Vous pouvez alors enregistrer les fichiers tag et les tag handlers dans ce fichier manuellement, ou lorsque vous utilisez l'assistant New File pour créer de nouveaux fichiers tag et des tag handlers.
Tag File	Crée un fichier <code>.tag</code> vide avec des commentaires suggérant les éléments de syntaxe JSP que vous pouvez utiliser pour créer un tag personnalisé.
Tag Handler	Crée une classe Java pour les tags JSP personnalisés. Le modèle inclut des commentaires de code avec des exemples de code et des suggestions pour vous aider dans la création de tags personnalisés.
HTML	Crée un HTML avec les tags élémentaires déjà introduits.
Web Service	Crée un simple service Web. Voyez XXX pour plus d'informations sur l'extension de Services Web avec des applications Web.
Message Handler	Crée un gestionnaire de messages basé sur SOAP pour les services Web. Voyez XXX pour plus d'information sur la consommation de Services Web avec des applications Web.
Web Service Client	Crée un client de service Web basé sur la JSR 109. Voyez XXX pour plus d'informations sur la consommation de services Web avec des applications Web.

## Ajout de Ressources Externes à Votre Projet

Si votre application Web a besoin d'être livrée avec d'autres bibliothèques, vous pouvez les rajouter dans le noeud Libraries. Si votre projet a été configuré pour fonctionner avec le serveur



Tomcat, les bibliothèques JSP et servlet sont incluses automatiquement et listées sous un sous-noeud pour le serveur Tomcat.

Vous pouvez rajouter des ressources externes à votre projet selon l'une des trois méthodes qui suivent:


- [lb] Un dossier ou fichier JAR individuel.
- [lb] Un ensemble de ressources (ou “bibliothèque”). Ce ensemble peut inclure de multiples fichiers JAR, les sources pour les fichiers JAR (qui sont nécessaires pour la completion de code, ou si vous désirez y passer avec le débogueur), et la documentation Javadoc. Vous pouvez créer de tel ensemble dans le Gestionnaire de Bibliothèques (menu Tools).
- [lb] Les archives d'un autre projet de l'EDI.

Pour rajouter un dossier ou fichier JAR isolé à votre projet, cliquez-droit sur le noeud Libraries de votre projet, choisissez Add JAR/Folder, et recherchez le dossier ou fichier Jar sur votre disque et sélectionnez le.

Pour ajouter un ensemble de ressources à votre projet, cliquez-droit sur le noeud Libraries, choisissez Add Library, et sélectionnez la bibliothèque depuis la liste de la boîte de dialogue Add Library. Si la bibliothèque que vous recherchez n'est pas présente, vous pouvez la rajouter à la liste en choisissant Manage Libraries.

#### **Conseil EDI NetBeans**

En plus de fournir un moyen de regrouper des ressources, les Gestionnaire de Bibliothèques simplifie l'accès aux ressources régulièrement utilisées. Même si vous n'avez pas besoin de regrouper les ressources, vous pouvez toujours rajouter des ressources individuelles au Gestionnaire de Bibliothèques pour vous éviter d'avoir à plonger dans les entrailles de vos répertoires pour rajouter les ressources aux autres projets.

Pour ajouter les archives d'un autre projet, cliquez-droit sur le noeud Libraries du projet en cours, et naviguez jusqu'au dossier du projet sur votre disque. Les dossiers de projets NetBeans sont indiqués par l'icône .

#### **Conseil EDI NetBeans**

Si vous désirez ajouter un fichier JAR ou le résultat du projet au fichier WAR sans qu'il ne fasse partie du classpath de la compilation, vous pouvez le faire via le noeud Packaging dans la boîte de dialogue Project Properties. Voyez Personnalisation Contenu du Fichier WAR à la page XXX.

### **Ajout de Tags et Bibliothèques de Tag**

Des Tags et bibliothèques de tag peuvent être rajoutés à votre projet Web selon l'une de ces façons :

- [lb] Fournit dans un fichier JAR contenant un fichier de descripteur de bibliothèque de tag (TLD) et les tags associés, sous la forme de fichier tag (en utilisant une syntaxe JSP) et/ou gestionnaire de tag (écrit en Java). De telles bibliothèques de tag apparaissent dans le répertoire `WEB-INF/lib` de l'application Web ou dans le dossier des bibliothèques partagées du serveur. Vous pouvez ajouter une bibliothèque de tag à un projet Web via le noeud Libraries du projet Web (voir Ajout Ressources Externes à Votre Projet à la page XXX).

Si vous développez une bibliothèque de tag “from scratch”, vous pouvez créer un projet Java Library pour la bibliothèque et ensuite rajoutez cette bibliothèque au projet Web en cliquant-droit sur le noeud Libraries du projet Web et en choisissant Add Project. Voyez le Chapter 3 pour plus d'information quant à la création de projets Java Library.

- [lb] En tant que fichier tag (en utilisant soit la syntaxe standard, soit la syntaxe document) inclut dans le dossier `WEB-INF/tags` de l'application Web. Vous pouvez ajouter de nouveaux fichiers tag à votre projet en cliquant-droit sur le noeud du projet et en choisissant New | File/Folder en ensuite en sélectionnant le modèle Tag File dans la catégorie Web de l'assistant.
- [lb] En tant que fichier TLD situé dans le répertoire `WEB-INF/tlds` et gestionnaires de tag (écrits en fichiers Java) dans le noeud Source Packages. Vous pouvez ajouter de nouveaux fichiers TLD et tag handlers à votre projet en cliquant-droit sur le noeud du projet et en choisissant New | File/Folder et ensuite sélectionnait les modèles dans la catégorie Web de l'assistant.

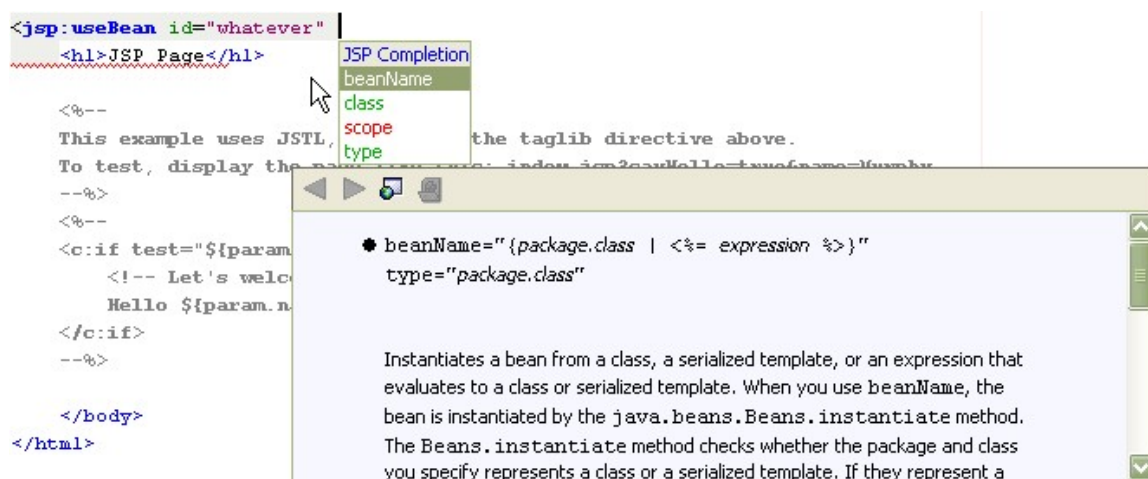
---

## *Edition et Refactoring Fichiers Applications Web*

L'Éditeur de Source de l'EDI fournit tout un tas de fonctionnalités qui vous simplifie l'encodage et les changements de code à faire dans vos nombreux composants d'application Web. Cette section va survoler quelques unes de ces fonctionnalités qui sont particulièrement utiles pour les applications Web. Voyez le Chapitre 4 pour plus d'information sur les fonctionnalités d'édition de l'EDI et les fonctionnalités de Refactoring.

### **Completion de Tags**

L'EDI vous permet d'avoir les tags et leurs attributs complétés automatiquement dans les fichiers JSP, HTML, et XML. La fonctionnalité de completion de tag, non seulement réduit le nombre de frappe clavier, mais vous fournit également une documentation contextuelle sur les différentes façons possibles de compléter le tag ou l'attribut que vous êtes en train d'encoder, comme le montre l'illustration 6-4.



#### Illustration 6-4

##### *La fonctionnalité de completion de code dans un fichier JSP*

Pour activer la fenêtre de completion de tag, il vous suffit de taper quelques lettres commençant le tag et de presser Ctrl-Espace, ou d'attendre une seconde pour que la completion de code s'active automatiquement. S'il n'existe qu'une seule façon de compléter le mot que vous avez tapé, la fin du mot sera complété automatiquement. S'il existe plusieurs façons de compléter le tag, une liste des possibilités vous est alors proposées. Vous pouvez continuer à coder pour réduire la liste ou alors sélectionnez le texte que vous désirez en utilisant la souris ou les touches fléchées pour sélectionner le mot désiré.

Par exemple, si vous désirez ajouter les instructions suivantes dans un fichier JSP:

```
<jsp:useBean id="hello" scope="page"  
class="org.mondomaine.monpaquetage.MaClasse" />
```

vous pouvez le faire par les étapes suivantes:

- [nl] Tapez `<jsp:u`
- [nl] Pressez Ctrl-Espace (`seBean` est rajouté).
- [nl] Tapez `i`
- [nl] Pressez Ctrl-Espace (`d=""` est rajouté avec le point d'insertion mis entre les guillemets).
- [nl] Tapez `hello`
- [nl] Pressez la touche flèche droite deux fois.
- [nl] Tapez `s`
- [nl] Pressez Entrée (`cope=""` est rajouté, avec le point d'insertion mis entre les guillemets).
- [nl] Tapez `p`
- [nl] Pressez Ctrl-Espace (`age` est rajouté).
- [nl] Pressez la touche flèche droite deux fois.
- [nl] Tapez `c`
- [nl] Pressez Entrée (`lass=""` est rajouté avec le point d'insertion mis entre les guillemets).
- [nl] Tapez `o`
- [nl] Pressez Ctrl-Espace (`rg` est rajouté, en supposant que la classe fasse partie de votre projet).
- [nl] Tapez un point (`.`) et pressez Ctrl-Espace (`mondomaine` est rajouté).
- [nl] Tapez un point (`.`) et pressez Ctrl-Espace (`monpaquetage` est rajouté).
- [nl] Tapez un point (`.`) et pressez Ctrl-Espace (`MaClasse` est rajouté).
- [nl] Pressez la touche flèche droite deux fois et tapez `>`

Voyez le Chapitre 4: Génération Bout de Code pour avoir des informations sur la completion d'expressions Java et la configuration de completion de code.

## Extension des Abréviations pour les fichiers JSP

Pour les bouts de code JSP couramment utilisés, vous pouvez tirer profit des abréviations dans l'Éditeur de Source pour réduire le nombre de frappes au clavier. Les abréviations sont alors étendues lorsque vous tapez l'abréviation et que vous pressez la barre d'espace.

Voyez le Tableau 6-3 pour une liste d'abréviations pour les fichiers JSP. Vous pouvez également étendre les abréviations pour les autres types de fichiers. Voyez le Chapitre 4, Tableau 4-1 pour une liste d'abréviations pour les classes Java.

Si une abréviation correspond au texte que vous désirez taper ('est-à-dire que vous ne désirez pas l'étendre en autre chose), pressez Maj-Espace pour empêcher sa transformation.

Vous pouvez modifier la liste des abréviations dans la boîte de dialogue Abréviations pour un type de fichier. Voyez le Chapitre 4, Ajout, Modification, et Suppression des Abréviations.

Tableau 6-3: Abréviations JSP dans l'Éditeur de Source

Abréviation	Devient
ag	application.getValue("
ap	application.putValue("
ar	application.removeValue("
cfgi	config.getInitParameter("
oup	out.print("
oupl	out.println("
pcg	pageContext.getAttribute("
pcgn	pageContext.getAttributeNamesInScope(
pcgs	pageContext.getAttributesScope("
pcr	pageContext.removeAttribute("
pcs	pageContext.setAttribute("
rg	request.getParameter("
sg	session.getValue("
sp	session.putValue("
sr	session.removeValue("
jspf	<jsp:forward page="
jg	<jsp:getProperty name="
jspg	<jsp:getProperty name="
jspi	<jsp:include page="
jspp	<jsp:plugin type="
jsps	<jsp:setProperty name="
jspu	<jsp:useBean id="
pg	<%@ page
pga	<%@ page autoFlush="
pgb	<%@ page buffer="
pgc	<%@ page contentType="
pgerr	<%@ page errorPage="
pgex	<%@ page extends="

Abbréviation	Devient
pgie	<%@ page isErrorPage="
pgim	<%@ page import="
pgin	<%@ page info="
pgit	<%@ page isThreadSafe="
pgl	<%@ page language="
pgs	<%@ page session="
tglb	<%@ taglib uri="

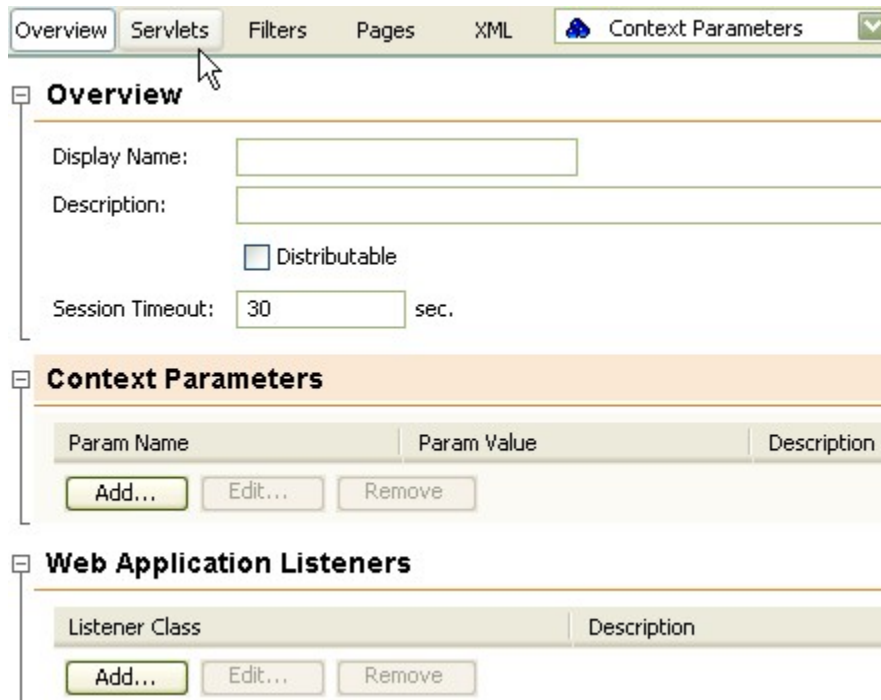
## Edition Manuelle du Descripteur de Déploiement

Bien que l'EDI vous guide pour l'ajout d'entrées dans le descripteur de déploiement (fichier `web.xml`) lorsque vous rajouter des servletes, filtres, et listeners à votre projet, vous pouviez avoir l'occasion d'éditer le fichier à la main.

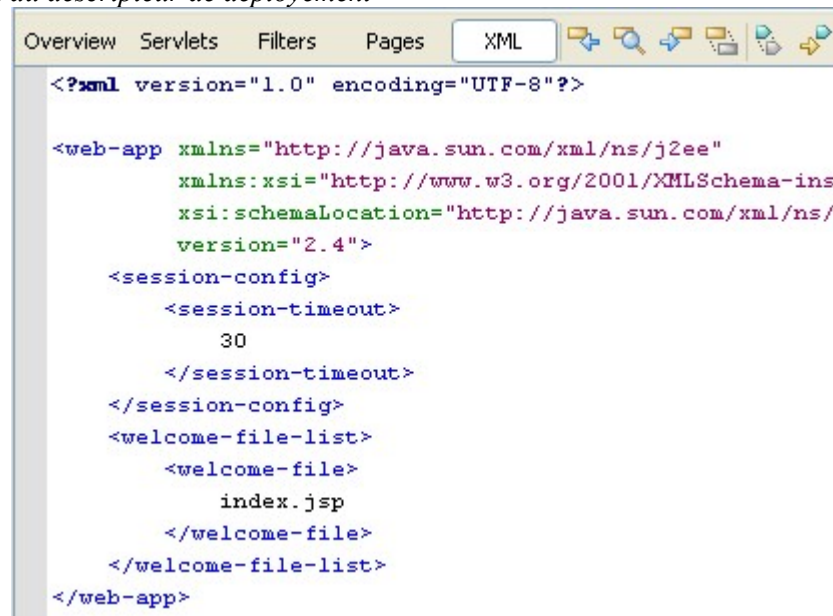
Pour ouvrir le descripteur de déploiement dans l'Éditeur de Source, ouvrez la fenêtre Projects, étendez le noeud Configuration Files (ou le noeud Web Pages | Web-INF) et double-cliquez sur le fichier `web.xml`. Le fichier s'ouvre en tant que document multi-onglets dans l'Éditeur de source avec l'onglet Overview ouvert (comme montré sur l'Illustration 6-5). Vous pouvez éditer différentes parties du fichier `web.xml` avec les éditeurs visuels pour les éléments différents du fichier (Overview, Servlets, Filters, et Pages) ou vous pouvez cliquez sur l'onglet XML pour éditer directement la source XML du fichier (voir Illustration 6-6).

### Illustration 6-5

*Editeur visuel du descripteur de déploiement*



**Illustration 6-6**  
*Editeur XML du descripteur de déploiement*



## Refactoring Composants Web

Le support de refactoring dans l'EDI NetBeans s'étend aux applications Web et aux applications entreprises. Par exemple, vous pouvez faire ce qui suit:

- [1b] Renomme les classes, méthodes, et champs des servletes, gestionnaires de tag, et autres composants Web. Voir le Chapitre 4, Renommer Toutes Les Occurrences de

Classe, Méthode ou Champ Actuellement Sélectionné

- [lb] Déplacer des classes dans d'autres paquetages ou projets (ce qui a pour résultat que la classe sera livrée dans un différent fichier WAR). Voir le Chapitre 4, Déplacement d'une Classe dans un Paquetage Différent.
- [lb] Modification des paramètres de méthodes y compris les noms des paramètres, les types de paramètres, et les visibilité de méthodes, l'ajout de paramètres à une méthode ou la suppression de paramètres à une méthode. Voir le Chapitre 4, Modification de la Signature d'une Méthode.
- [lb] Modification de la visibilité des champs et rajout des méthodes setter et getter. Voir le Chapitre 4, Génération des Propriétés Read/Write.

Lorsque vous renommez une classe de composant Web (comme une servlette ou un gestionnaire de tag), toutes les entrées correspondantes dans le descripteur de déploiement (fichier `web.xml`) et/ou le descripteur de bibliothèque de tag (TLD) sont également mises à jour. Lorsque vous déplacez une classe et que le descripteur de déploiement est affecté, vous êtes invité, avec la boîte de dialogue Confirm Changes à confirmer les modifications au descripteur de déploiement.

#### **Important**

Attention de ne pas renommer ou modifier les paramètres d'une méthode de servlette qui doit être implémentée avec un nom donné selon les spécifications des Servlets ou des EJB.

---

## ***Déploiement d'une Application Web***

Par défaut, une application Web est déployée sur le serveur que vous avez spécifié lorsque vous exécutez ce projet.

Pour les applications que vous déployez dans le serveur Tomcat, l'application est déployée “sur place”, signifiant que l'EDI crée un fichier XML qui est placé dans le répertoire `conf/Catalina/localhost/` du serveur et fait pointer Tomcat vers le répertoire build du projet où se trouvent les fichiers de l'application.

Lorsque vous générez un projet Web dans l'EDI, un fichier WAR est également créé, que vous pouvez déployer manuellement vers un serveur.

### **Personnalisation Contenu d'un Fichier WAR**

Par défaut, un fichier WAR généré par un projet Web inclut:

- [lb] Tous les fichiers affichés dans le noeud Web Pages du projet, y compris les fichiers `web.xml` et `context.xml`.
- [lb] Les fichiers de classes compilées depuis les sources situées dans le noeud Java Sources plus les autres fichiers qui y sont placés, à l'exception de ceux qui ont l'extension `.java` et `.form`.
- [lb] Toutes bibliothèques que vous avez rajoutées dans le noeud Libraries du projet.

Vous pouvez également rajouter des fichiers JAR au fichier WAR et écarter du contenu qui apparaîtrait normalement.

Pour personnaliser le contenu d'un fichier WAR:

- [nl] Cliquez-droit sur le noeud du projet dans la fenêtre Projects et choisissez Properties.

- [nl] Sélectionnez le noeud Build | Packaging.
- [nl] Si vous désirez écarter des fichiers du fichier WAR généré, modifier l'expression régulière dans le champ Exclude From WAR File.
- [nl] Si vous désirez ajouter des dossiers ou fichiers supplémentaires, faites le via les boutons Add JAR/Folder, Add Library, ou Add Project.

Les boutons Add JAR/Folder vous permettent de rajouter des fichier JAR ou dossiers, qu'ils proviennent ou non de projets de l'EDI.

Le bouton Add Library vous permet de rajouter tout fichier JAR ou groupe de fichiers JAR que vous avez désigné dans le Gestionnaire de Bibliothèque de l'EDI.

Le bouton Add Project vous permet de rajouter le fichier JAR qui est généré par un autre projet de l'EDI. Lorsque vous ajoutez un projet de l'EDI à un fichier WAR, le fichier JAR de ce projet est régénéré à chaque fois que vous générez l'application Web.

### **Non-Déploiement d'une Application Web**

Lorsque vous stoppez une application Web que vous avez exécutée via l'EDI, l'application reste déployée via la référence à l'application sous la forme d'un fichier XML dans le répertoire `conf/Catalina/localhost` de Tomcat.

Pour ne plus déployer de telle applicaton depuis Tomcat:

- [nl] Ouvrez la fenêtre Runtime.
- [nl] Étendez le noeud Servers, et étendez le noeud pour le serveur Tomcat et le noeud Web Applications pour le serveur.
- [nl] Cliquez-droit sur le noeud de l'application Web et choisissez Undeploy.

### **Redéployer une Application Web**

Pour enlever votre application du serveur et la redéployer, cliquez-droit sur le noeud du projet et choisissez Redeploy Project.

### **Création d'un Fichier WAR**

Lorsque vous exécute la commande Build Project sur un projet Web dans l'EDI, un fichier WAR est automatiquement créé et placé dans le dossier `dist` du projet. Vous pouvez accéder à ce fichier et visualiser son contenu dans la fenêtre Files.

### **Déploiement vers un Autre Serveur Tomcat**

L'EDI comprend un serveur Tomcat, ce qui facilite le développement et test d'application Web. Si vous avez une installation différente de Tomcat que vous désirez tester et/ou déployer vers, vous pouvez enregistrer cette installation dans l'EDI. Vous pouvez facilement basculer votre application pour travailler avec différentes installations de serveur. Cela est particulièrement utile si vous désirez développer et tester sur une installation et ensuite déployer vers un environnement de production.

Pour que l'EDI puisse reconnaître une autre installation de Tomcat:

- [nl] Choisissez Tools | Server Manager .
- [nl] Cliquez sur le bouton Add Server.



- [nl] Dans la page Choose Server de l'assistant qui s'ouvre, sélectionnez un serveur depuis la liste déroulante Server, introduisez un nom d'affichage identifiant ce serveur dans l'EDI dans le champ Name, et cliquez sur Next.
- [nl] Sur la page Tomcat Server Instance Properties, spécifiez le répertoire d'installation de Tomcat (et le répertoire de base si c'est une installation partagée) et remplissez un identifiant et un mot de passe.  
Vous pouvez également déterminer d'activer le Moniteur HTTP de l'EDI. Si le Moniteur HTTP est activé, vous pouvez monitorer les requêtes serveur de votre application, les cookies etc, ce qui simplifie le débogage de votre application. Cependant, cette option ralentit le serveur. Vous désirez certainement que cette option soit désactivée si vous l'utilisez pour votre serveur de production. Voyez Monitoring Des Transactions HTTP à la page XXX pour plus d'informations sur l'utilisation du Moniteur HTTP.
- [nl] Vérifiez que le serveur n'utilise pas un numéro de port déjà utilisé. Dans l'EDI, vous pouvez voir le numéro de port du serveur en plaçant le curseur de la souris sur le noeud du serveur pour voir l'aide contextuelle du noeud.  
Si une autre instance de serveur utilise le même port, vous devez arrêter un des serveurs, et changer le port qu'il utilise. Arrêter le serveur en cliquant-droit sur le noeud du serveur, en choisissant Start/Stop Server et en cliquant sur Stop Server. Ensuite, cliquez-droit sur le noeud du serveur, choisissez Properties, et modifiez la propriété Server Port. Redémarrez le serveur en cliquant-droit sur le noeud, choisissant Start/Stop Server et en cliquant sur Start Server.
- [nl] Si vous avez d'autres applications Web existantes dans l'EDI que vous désirez exécuter sur le serveur nouvellement rajouté, modifiez les propriétés de chaque projet pour utiliser ce serveur. Vous pouvez le faire en cliquant-droit sur le noeud du projet dans la fenêtre Projects, choisissant Properties, sélectionnant le noeud Run, et finalement sélectionnant le serveur dans la liste déroulante.

#### **Conseil EDI Netbeans**

Si vous désirez modifier plus tard la configuration du serveur, vous pouvez accéder aux propriétés du serveur en ouvrant la fenêtre Runtime, en étendant le noeud Servers node, cliquant-droit sur le noeud du serveur spécifique pour y choisir Properties.

---

## ***Test et Débogage de Votre Application Web***

L'EDI NetBeans fournit un environnement riche pour le “troubleshooting” et l'optimisation de vos applications Web. Quelques unes des fonctionnalités pour faciliter le test de vos applications Web incluent :

- [lb] La commande Compile JSP, qui vous permet de vérifiez les fichier JSP individuellement pour les erreurs avant de les déployer vers le serveur.
- [lb] Intégration du Débogueur avec les fichiers JSP, ce qui signifie que vous pouvez rajouter des points d'arrêt dans les fichiers JSP et avancez pas à pas dans le JSP lors du débogage (plutôt que d'avoir à vous promenez dans le code de la servlette générée).
- [lb] Capacité d'avancer pas à pas dans les fichiers tag.
- [lb] Capacité d'évaluer les expressions de Expression Language (EL) dans les fichiers

JSP durant une session de débogage (en plaçant le curseur de la souris sur l'expression ou en rajoutant un “watch”, technique consistant à rajouter une surveillance sur cette expression).

- [lb] Le Moniteur HTTP, qui trace la communication HTTP entre les servletes et le serveur. Cette fonctionnalité est couverte en détail dans Monitoring Transactions HTTP à la page .

Voir le Chapitre 5 pour plus d'informations sur les fonctionnalités de débogage générale de l'EDI que ces fonctionnalités étendent.

## Vérification des Erreurs JSP

Les fichiers JSP ne sont pas compilés comme des fichiers Java avant d'être déployés. Au lieu de cela, ils sont compilés par le serveur après qu'ils aient été déployés (où ils sont en fait tout d'abord transformés en servletes, qui sont alors compilées. Cela complique la tâche pour corriger les erreurs puisque vous devez déployer l'application découvrir l'erreur, arrêter l'application, corriger l'erreur, et redéployer.

Cependant, l'EDI NetBeans vous permet de compiler les fichiers JSP pour vérifiez les erreurs avant que vous n'emballiez et déployez l'application. Vous pouvez soit compiler les fichiers JSP manuellement, soit spécifier qu'ils soient compilés lorsque vous générez le projet.

Pour compiler un fichier JSP manuellement, sélectionnez le fichier dans la fenêtre Projects ou dans l'Éditeur de Source et sélectionnez Build | Compile File ou pressez F9.

Pour avoir tous les fichiers JSP compilés lorsque vous exécutez la commande Build Project, cliquez-droit sur le noeud du projet, choisissez Properties, sélectionnez le noeud Compiling, et cochez la case Test Compile All JSP Files During Builds.

Le résultat de la compilation est reporté dans la fenêtre Output, où vous pouvez découvrir toutes les erreurs, qu'elles se soient produites durant la traduction en servlete ou durant la compilation de la servlete.

Les fichiers compilés eux-mêmes sont placés dans le dossier `build/generated` du projet, que vous pouvez visualiser depuis la fenêtre Files. Ces fichiers ne sont pas utilisés lorsque vous générez et préparez l'application pour le déploiement.

## Visualiser la Servlete d'un Fichier JSP

La génération d'une servlete depuis un fichier JSP se produit dynamiquement sur le serveur où l'application Web est déployée. Vous pouvez visualiser cette servlete générée une fois que vous avez exécuté le projet ou la JSP spécifique qui lui est associée en cliquant-droit sur le fichier et en choisissant View Servlet.

### Conseil EDI NetBeans

Si vous désirez voir le code de la servlete qui est générée lorsque vous exécutez la commande Compile JSP, ouvrez la fenêtre Files, ouvrez le dossier `build/generated/src`, et recherchez le fichier ayant le même nom que celui de la page JSP avec avec un suffixe `_jsp` et une extension `.java`.

## Visualiser un Fichier dans un Navigateur Web

Vous pouvez ouvrir les composants d'une Application Web dans un navigateur Web depuis l'EDI.

Pour voir une page JSP spécifique dans un navigateur Web, vous devez exécuter ce fichier individuellement (en cliquant-droit sur le fichier dans l'Éditeur de Source ou en pressant Maj-F6).

Pour ouvrir un fichier HTML dans le navigateur Web, cliquez-droit sur le noeud de la page HTML dans la fenêtre Projects et choisissez View.

### **Conseil EDI NetBeans**

La commande View pour les fichiers HTML n'est pas disponible depuis l'Éditeur de Source. Si vous désirez voir le fichier HTML que vous éditez actuellement dans l'Éditeur de Source sans que vos doigts ne quittent le clavier, pressez Maj-Ctrl-1 pour vous positionnez sur le noeud du fichier dans la fenêtre Projects, pressez Maj-F10 pour ouvrir le menu contextuel du noeud, et pressez la touche flèche bas pour sélectionner View, et pressez Entrée.

## **Passage de Paramètres de Requête à une Application Web**

Vous pouvez tester manuellement la façon dont l'application Web répondra à un certain input en exécutant l'application avec certains paramètres définis à l'avance.

Pour passer les paramètres de requête à la page JSP:

- [nl] Cliquez-droit sur le noeud du fichier JSP et choisissez Properties.
- [nl] Dans la propriété Request Parameters, introduisez les paramètres sous le format de URL query string (où l'expression commence avec une URL, continue avec un point d'interrogation (?) pour indiquer le début de la requête, et complétée avec les paramètres sous paires nom=valeur, ou les paires sont séparées par le et commercial (&)).

Pour passez les paramètres de requête à une servlet, cliquez-droit sur le noeud de la servlet dans la fenêtre Projects et choisissez Tools | Set Servlet Execution URI.

- [nl] Dans la boîte de dialogue, ajoutez un point d'interrogation plus les paires nom=valeur, avec chaque paires séparées par un et commercial.

## **Débogage JSP et Fichiers Tag**

L'une des fonctionnalités de l'EDI qui a longtemps rendu l'EDI NetBeans favoris auprès des développeurs Web est la capacité du débogueur d'avancer pas à pas dans les fichiers JSP. Vous pouvez définir des points d'arrêt dans les fichiers JSP et avancer pas à pas dans le JSP tout en monitorant les valeurs des variables et autres aspects du programme qui s'exécute.

L'EDI NetBeans 4.1 rajouter la capacité d'avancer pas à pas dans les fichiers tag.

Pour définir un point d'arrêt dans un fichier JSP ou tag, sélectionnez la ligne où vous aimeriez suspendre l'exécution et pressez Ctrl-F8. Voir le Chapitre 5 pour plus d'information concernant le débogage.

---

## ***Création et Déploiement d'Applettes***

L'EDI NetBeans ne possède pas de projet spécifique pour les applettes, ce qui fait que le cycle de développement des applettes est quelque peu différent que pour les autres types de projets. Vous ne pouvez désigner une applette comme étant une classe principale pour le projet, ce qui signifie que des commandes relatives au projet (comme Run Project) ne s'appliquent pas aux applettes.

Cependant, vous pouvez toujours créer, tester et déployer des applets assez facilement. Les grandes lignes pour le développement d'une applette sont les suivantes:

- [nl] Créez un projet Java Library qui contiendra l'applette.
- [nl] Créez une applette depuis l'un des modèles dans l'assistant New File et codez l'applette.
- [nl] Testez l'applette dans l'applet viewer du JDK en cliquant droit sur le noeud de l'applette dans la fenêtre Projects et en choisissant Run File.
- [nl] Créez un fichier JAR pour l'applette en cliquant-droit sur le noeud du projet de l'applette et en choisissant Build Project.
- [nl] Si vous désirez rajouter l'applette à une application Web, ajoutez le projet de l'applette (ou juste le fichier JAR de l'applette) via le noeud Libraries de l'application Web.

### **Création d'une Applette**

Pour créer une applette:

- [nl] Choisissez New Project, sélectionnez la catégorie General, sélectionnez le modèle Java Library, et cliquez sur Next.
- [nl] Introduisez le nom et l'emplacement du projet et cliquez sur Finish pour quitter l'assistant.
- [nl] Dans la fenêtre Projects, étendez le noeud du projet que vous venez de créer. Ensuite, cliquez-droit sur le noeud Source Packages et choisissez New | File/Folder.
- [nl] Dans l'assistant New File, sélectionnez un des modèles d'applette disponibles. Il y en a 4 disponibles:
  - [lb] Catégorie Java Classes, modèle JApplet. Ce modèle étend `javax.swing.JApplet` et est recommandé par rapport au modèle Applet template, qui est basé sur une classe moins flexible `java.applet.Applet`.
  - [lb] Catégorie Java GUI Forms, modèle JApplet. Ce modèle étend `javax.swing.JApplet` et vous permet d'utiliser l'Éditeur Visuel pour concevoir graphiquement votre applette. Ce modèle est recommandé par rapport au modèle Applet de la sous-catégorie AWT Forms.
  - [lb] catégorie Java Classes, modèle Applet. Ce modèle étend `java.applet.Applet`.
  - [lb] Catégorie Java GUI Forms | AWT Forms, modèle Applet. Ce modèle étend `java.applet.Applet` et vous permet d'utiliser l'Editeur Visuel de l'EDI pour concevoir graphiquement votre applette.  
Cliquez sur Next, spécifiez un nom et un paquetage pour l'applette et ensuite cliquez sur Finish.

Vous pouvez alors coder l'applette, soit à la main ou avec l'assistantce de l'Éditeur Visuel.

### **Exécution et Débogage d'une Applette dans l'Applet Viewer**

Lorsque vous développez l'applette, vous pouvez utiliser l'applet viewer du JDK pour tester les fonctionnalités de l'applette. Lorsque vous exécutez les commandes Run File et Debug File, l'applette est automatiquement affichée dans l'applet viewer.

Pour exécuter une applette, cliquez-droit sur le noeud d'une applette dans la fenêtre Projects et choisissez Run File.

Pour déboguer une applette:

- [nl] Définissez un point d'arrêt dans le code en sélectionnant la ligne où vous désirez suspendre l'exécution et pressez Ctrl-F8.
- [nl] Cliquez-droit sur le noeud de l'applette dans la fenêtre Projects et choisissez Debug File.

## Exécution d'une Applette dans un Navigateur Web

Si vous désirez voir comment votre applette se comporte dans un navigateur Web, vous pouvez ouvrir un lanceur HTML pour l'applette.

Pour exécuter une applette dans un navigateur Web:

- [nl] Ouvrez la fenêtre Files et étendez le répertoire `build` du projet.
- [nl] Cliquez-droit sur le fichier de lanceur HTML (il devrait avoir le même nom que la classe de l'applette, mais avec une extension HTML) et choisissez View.

L'applette s'ouvre dans le navigateur Web par défaut spécifié dans l'EDI. Voir Modification du Navigateur Web par Défaut de l'EDI à la page XXX si vous désirez modifier le navigateur web par défaut de l'EDI.

### Conseil EDI NetBeans

Si vous désirez personnaliser le fichier lanceur HTML, vous pouvez copier le fichier de lanceur généré dans le dossier qui contient le code source de l'applette. Cela empêchera au lanceur d'être écrasé à chaque fois que vous exécutez l'applette.

Lorsque vous exécutez l'applette ou que vous générez le projet de l'applette, le fichier HTML est copié depuis dans le dossier avec la classe compilée de l'applette. Si vous ne désirez pas que ce fichier soit inclut dans le JAR qui est créé lorsque vous générez le projet, vous pouvez modifier le filtre pour le contenu du fichier JAR. Dans la fenêtre Projects, cliquez-droit sur le noeud du projet et choisissez Properties. Dans la boîte de dialogue, sélectionnez le noeud Packaging et modifier l'expression régulière dans le champ Exclude From JAR File. Par exemple, vous pouvez ajouter une virgule plus une expression comme `**/Myapplet.html` pour vous assurez que votre fichier de lanceur (quelque soit le répertoire où il se trouve) est exclut du fichier JAR généré.

## Mettre l'Applet dans un Fichier JAR

Si vous désirez mettre une applette dans un fichier JAR, vous pouvez le faire en cliquant-droit sur le noeud du projet de l'applette dans la fenêtre Projects et en choisissant Build Project.

L'applette est alors compilée et les fichiers `.class` compilées sont placées dans un fichier JAR dans le répertoire `dist`, que vous pouvez voir dans la fenêtre Files.

## Mettre l'Applet dans un Fichier WAR

Pour rajouter une applette à une application Web:

- [nl] Mettre l'applette dans un fichier JAR. Voir Mettre une Applette dans un Fichier JAR ci-dessous pour savoir comment le faire dans l'EDI.

- [nl] Cliquez-droit sur le noeud du projet de l'application Web dans la fenêtre Projects et choisissez Properties.
- [nl] Dans la boîte de dialogue Project Properties, sélectionnez le noeud Build | Packaging.
- [nl] Cliquez sur le bouton Add Project, naviguez jusqu'au dossier du projet de l'applette, et cliquez sur Add Project JAR Files.

### **Définition des Permissions de l'Applette**

Lorsque vous créez et exécutez une applette via l'EDI, une `applet.policy` est créée avec toutes les permissions accordées et est placée dans le répertoire racine du projet (qui peut être vu dans la fenêtre Files). Vous pouvez modifier ce fichier en double-cliquant sur son noeud pour l'ouvrir dans l'Éditeur de Source.

Vous pouvez également spécifier une police différente pour l'applette. Pour spécifier un fichier `policy` différent:

- [nl] Cliquez-droit sur le noeud du projet de l'application Web dans la fenêtre Projects et choisissez Properties.
- [nl] Dans la boîte de dialogue Project Properties, sélectionnez le noeud Run.
- [nl] Dans le champ VM Options, modifiez la valeur de l'option `-Djava.security.policy` pour pointer vers le fichier `policy`.

---

### ***Modification du Navigateur Web par Défaut de l'EDI***

Pour modifier le navigateur Web par défaut de l'EDI, choisissez Tools | Setup Wizard et sélectionnez un navigateur depuis la liste déroulante Web Browser.

Si l'EDI ne trouve pas un navigateur Web sur votre système, vous devez spécifier l'exécutable pour ce navigateur Web.

Pour faire pointer l'EDI vers un navigateur Web qui est repris dans la liste de l'assistant Setup:

- [nl] Choisissez Tools | Options et étendez le noeud IDE Configuration | Server and External Tools Settings | Web Browsers.
- [nl] Sélectionnez le sous-noeud pour le navigateur dont vous désirez modifier la propriété Browser Executable.

Pour rajouter un navigateur à la liste de l'EDI:

- [nl] Choisissez Tools | Options et étendez le noeud IDE Configuration | Server and External Tools Settings | Web Browsers.
- [nl] Cliquez-droit sur le noeud Web Browsers, choisissez New | External Browser, introduisez le nom du navigateur Web et cliquez sur Finish.
- [nl] Sélectionnez le sous-noeud du navigateur rajouté et modifiez la propriété Browser Executable pour pointer vers l'exécutable du navigateur Web.

---

## Monitoring Transactions HTTP

L'EDI NetBeans fournit un Moniteur HTTP intégré, pour aider à isoler les problèmes de flots de données lors d'exécution de servlets et JSP sur un serveur Web. Il n'est pas nécessaire de rajouter de logique à votre application Web pour tracer les requêtes HTTP et les informations d'état associées. Le Moniteur HTTP intégré à l'EDI NetBeans peut faire cela pour vous.

Lorsque l'EDI est configuré avec un conteneur Web où une application Web est déployée avec une servlet filtre Moniteur HTTP NetBeans et un "filter mapping", le moniteur HTTP va automatiquement enregistrer toutes les requêtes HTTP faites au conteneur Web. Pour chaque requête HTTP qui est traitée par le conteneur web, le Moniteur HTTP non seulement enregistre la requête, mais il enregistre également les informations d'état maintenue dans le conteneur web.

En utilisant le Moniteur HTTP, vous pouvez analyser les requêtes HTTP et stockez les requêtes HTTP GET et HTTP POST pour de futures sessions d'analyse. Vous pouvez également éditer ces requêtes stockées et les rejouer. C'est une fonctionnalité puissante pour aider à isoler le flot des données et les informations d'état passée dans une requête HTTP vers un conteneur web. Les requêtes HTTP sont stockées jusqu'à ce que vous quittiez l'EDI, à moins que vous ne les sauvegardez explicitement.

Voici une liste des possibilités que vous pouvez faire grâce au Moniteur HTTP de l'EDI:

- [lb] Analyser les enregistrements des requêtes HTTP.
- [lb] Enregistrer les enregistrements des requêtes HTTP.
- [lb] Éditer les enregistrements des requêtes HTTP.
- [lb] Rafraîchir les enregistrements des requêtes.
- [lb] Trier les enregistrements des requêtes HTTP.
- [lb] Supprimer les enregistrements des requêtes HTTP.
- [lb] Rejouer les enregistrements de requêtes HTTP.

Dans les sections qui suivent, vous allez apprendre comment configurer le moniteur HTTP, analyser les données que le Moniteur HTTP collecte, et rejouer les requêtes HTTP enregistrées.

### Mise En Place du Moniteur HTTP

Lorsque vous exécutez une application Web, le Moniteur HTTP s'affiche automatiquement. Si le Moniteur HTTP n'apparaît pas en bas de l'EDI, vérifiez alors que le Moniteur HTTP est bien activé, en effectuant les tâches suivantes:

- [nl] Étendez le noeud Servers de la fenêtre Runtime pour montrer les serveurs web enregistrés. Ensuite, cliquez-droit sur le noeud du serveur et choisissez Properties.
- [nl] Cochez la case Enable HTTP Monitor.
- [nl] Si le serveur web enregistré est actuellement démarré, arrêtez-le et redémarrez-le en cliquant droit sur le noeud du serveur et en sélectionnant Start / Stop Server.

L'illustration 6-montre la propriété 6 HTTP Monitor dans la boîte de dialogue Sun Java System Application Server 8 Properties.



**Illustration 6-6:**

*Boite de dialogue de propriétés pour le Sun Java System Application Server*

## Mise En Place du Moniteur HTTP Pour les Serveurs Démarrés En Dehors de l'EDI

Vous pouvez également utiliser le Moniteur HTTP sur des serveurs Web démarrés en dehors de l'EDI NetBeans. Pour faire cela, exécutez les tâches suivantes:

- [nl] Allez dans le répertoire où l'EDI NetBeans est installé et copier les fichiers `modules/org-netbeans-modules-schema2beans.jar` et `modules/org-netbeans-modules-web-httpmonitor.jar` dans le répertoire de votre module web `WEB-INF/lib`.
- [nl] Ajoutez une déclaration de filtre appropriée à votre version de servlet en haut du fichier `WEB-INF/web.xml` de votre module Web.

Les entrées Filters et filter mapping doivent être spécifiées au début du descripteur de déploiement. Voyez les exemples ci-dessous pour les filtres selon les versions 2.3 et 2.4 des spécifications de Servlet.

Une déclaration de filtre selon les spécifications 2.4 ressemble à ceci:

```
<filter>
  <filter-name>HTTPMonitorFilter</filter-name>
  <filter-class>
    org.netbeans.modules.web.monitor.server.MonitorFilter
  </filter-class>
  <init-param>
    <param-name>
      netbeans.monitor.ide
    </param-name>
    <param-value>
      name-of-host-running NetBeans IDE:http-server-port
    </param-value>
  </init-param>
</filter>
```



```

    </param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>
    HTTPMonitorFilter
  </filter-name>
  <url-pattern>
    /*
  </url-pattern>
  <dispatcher>
    REQUEST
  </dispatcher>
  <dispatcher>
    FORWARD
  </dispatcher>
  <dispatcher>
    INCLUDE
  </dispatcher>
  <dispatcher>
    ERROR
  </dispatcher>
</filter-mapping>

```

---

Une déclaration de filtre selon les spécifications 2.3 ressemble à ceci:

```

<filter>
  <filter-name>HTTPMonitorFilter</filter-name>
  <filter-class>
    org.netbeans.modules.web.monitor.server.MonitorFilter
  </filter-class>
  <init-param>
    <param-name>
      netbeans.monitor.ide
    </param-name>
    <param-value>
      name-of-host-running NetBeans IDE:http-server-port
    </param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>
    HTTPMonitorFilter
  </filter-name>
  <url-pattern>
    /*
  </url-pattern>
</filter-mapping>

```

---

Une application Web peut être monitorée avec le Moniteur HTTP de l'EDI depuis plusieurs IDE NetBeans en rajoutant des entrées `init-param` supplémentaires à la déclaration de filtre de la serviette dans le descripteur de déploiement web. Par exemple, vous devriez rajouter une entrée `init-param` comme celle montrée ci-dessous:

```

<init-param>
  <param-name>

```

```
    netbeans.monitor.ide
</param-name>
<param-value>
    name-of-2nd-host-running NetBeans IDE:http-server-port
</param-value>
</init-param>
```

---

### IMPORTANT

Lorsque vous déployez le module web que vous avez monitorer avec le Moniteur HTTP vers un serveur de production, souvenez-vous de supprimer les déclarations de mapping Servlet filter and filter mapping du descripteur de déploiement du module web. Autrement le module web sera ouvert pour le Monitoring HTTP depuis les EDI NetBeans spécifiés dans la(es) section(s) `init-param` section(s) du servlet de filtre dans le descripteur de déploiement de module web.

### Analyse des Données Récoltées

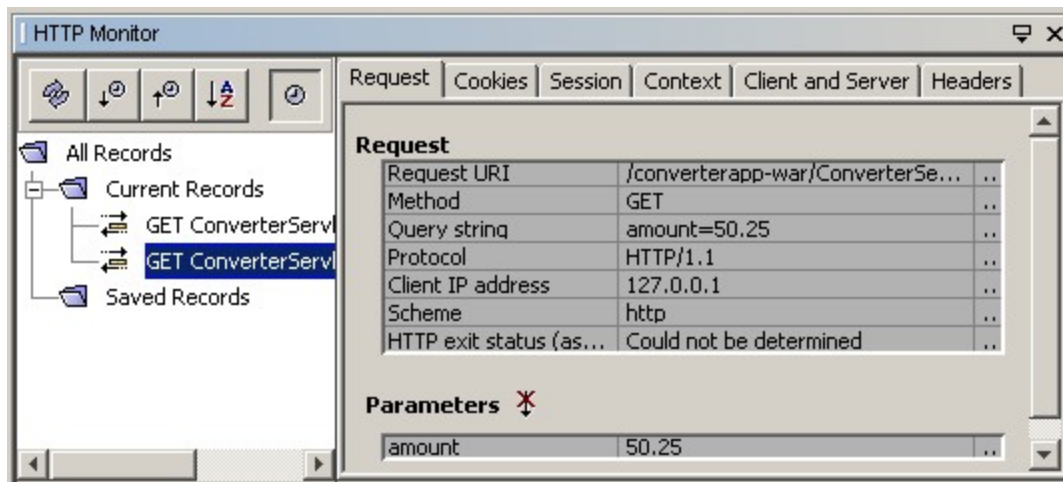
Après avoir mis en place le Moniteur HTTP, vous pouvez utiliser le Moniteur HTTP pour déboguer votre application web en observant les flots de données de vos pages JSP et servletes sur votre serveur web. Le Moniteur HTTP enregistre les données de chaque requête rentrante. Le Moniteur HTTP est automatiquement affiché au bas de l'EDI. Une copie d'écran du Moniteur HTTP est montrée à l'illustration 6-7.



**Illustration 6-7**  
*Moniteur HTTP*

Le Moniteur HTTP est constitué de deux panneaux principaux. A gauche se trouve la vue arborescente des enregistrements de requêtes HTTP. Chaque requête HTTP fait au serveur HTTP est enregistrée dans le Moniteur HTTP. Les requêtes résultant d'un dispatch interne sont reflétée par des noeuds imbriqués sous ces conteneurs web qui le supporte. De plus, les requêtes de type « Forward » ou « Include » sont imbriquées sous le noeud correspondant à la requête principale.

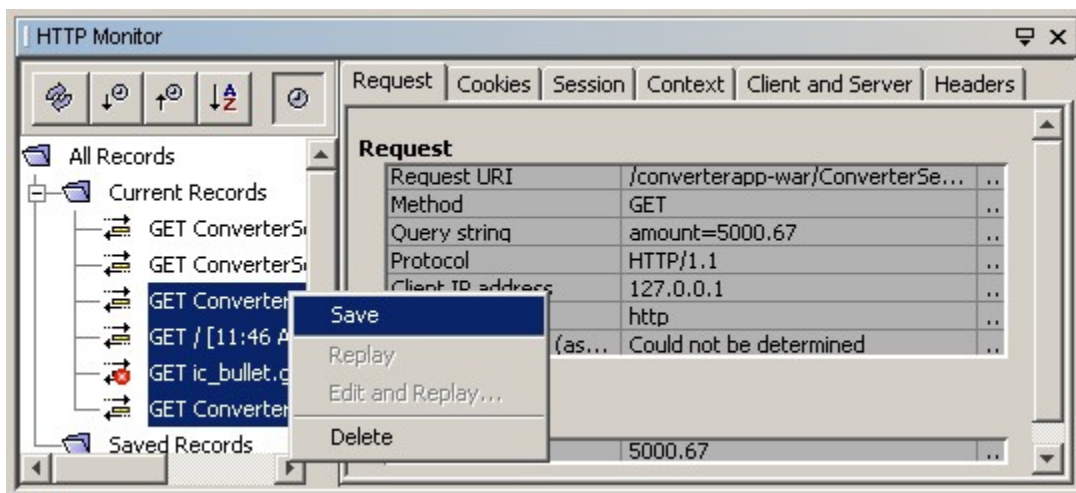
Dans le panneau de droite du Moniteur HTTP sont affichées les données supplémentaires pour un enregistrement de requête HTTP du panneau de gauche. Lorsque vous sélectionnez une requête HTTP dans le panneau de gauche, les données de session correspondantes à l'enregistrement sélectionné sont affichées dans le panneau de droite. Les informations disponibles dans le panneau de droite incluent: les informations de requête détaillées, les paires nom du cookie/ valeur, les données de la session, du contexte de servlet, les attributs de contextes, les paramètres d'initialisation, les informations clients/serveurs comme le protocole client, l'adresse IP du client, la plateforme du serveur, le host name du serveur, ainsi que les informations de l'entête de la requête HTTP. Le panneau de droite vous permet de voir les données spécifiques pour chacune de ces catégories en sélectionnant l'onglet correspondant à l'information que vous désirez voir. L'illustration 6-7 ci-dessous vous montre les informations de session supplémentaire pour un enregistrement de requête HTTP.



### Illustration 6-7

*HTTP Monitor with a record selected and its request data displayed*

Dans la vue arborescente du panneau de gauche du Moniteur HTTP, il y a deux catégories d'enregistrements qui peuvent être visualisés. Les enregistrements actuels ou les enregistrements sauvegardés. Les Enregistrements Actuels représentent les enregistrements de requêtes HTTP collectée depuis que l'EDI NetBeans a été démarré. Les Enregistrements Actuels vont persister aux redémarrage des serveurs Web, mais ils ne vont pas persister au redémarrage de l'EDI NetBeans. Pour rendre les informations persistantes aux redémarrage de l'EDI NetBeans, ils doivent être sauvegardés quelque part. Les Enregistrements Actuels peuvent être sauvegardés individuellement en sélectionnant l'enregistrement d'une requête HTTP, en cliquant droit dessus et en choisissant Save depuis le menu contextuel. Vous pouvez sélectionner plusieurs enregistrements de requêtes HTTP en pressant les touches Maj ou Ctrl. L'illustration 6-8 vous illustre la sélection et la sauvegarde de multiples requêtes HTTP.



### Illustration 6-8

#### *Saving multiple records in the HTTP Monitor*

Remarquez que lorsque vous sauvegardez les requêtes HTTP, les enregistrements sélectionnés sont déplacés dans la catégorie Saved Records. Les enregistrement sélectionnés ne sont pas copiés. Vous devez en tenir compte lorsque vous désirez rejouter une séquence de requêtes HTTP.

Le panneau de gauche du Moniteur HTTP fournit également plusieurs options de visualition des requêtes HTTP dans la partie supérieur du panneau, sous la forme de cinq boutons. Par exemple, vous pouvez recharger tous les enregistrements de requêtes HTTP, les trier par heure en ordre ascendant ou descendant, les trier alphabétiquement, et montrer ou cacher la date et heure de chaque requête HTTP. Le Tableau 6-4 résume l'action de chacun des boutons.

Tableau 6-4: boutons barre d'outils Moniteur HTTP

Bouton	Action
reload	Recharge tous les enregistrements de requêtes HTTP actuellement stoqués
descending sort	Trie les requêtes HTTP enregistrée selon la date et heure en ordre descendant
ascending sort	Trie les requêtes HTTP enregistrée selon la date et heure en ordre ascendant
alphabetically sort	Tie les enregsitremets de requêtes HTTP alphabétiquement
time stamp	Cache ou affiche la date/heure dans la liste des requêtes HTTP

En plus de sauvegarder les enregistrements de requêtes HTTP, les enregsitremets peuvent également être supprimés. L'EDI NetBeans fournit beaucoup de flexibilité lors de l'effacement des enregsitremets HTTP. Par exemple, vous pouvez effacer un ou plusieurs ou tous les enregsitremets actuels. Vous pouvez également supprimer un ou plusieurs ou tous les enregsitremets sauvegardés.

Pour supprimer un Enregistrement Actuel individuel, cliquez droit sur l'enregistrement à effacer et choisissez l'option Delete du menu contextuel. Pour supprimer plusieurs Enregistrements Actuels, sélectionnez plusieurs enregistrement à l'aide des touches Maj et Ctrl, et ensuite cliquez-droit sur les enregistrements sélectionnés et choisissez l'option Delete du menu contextuel. Pour supprimer tous les Enregistrements Actuels, cliquez droit sur le dossier Current Records et

choisissez l'option Delete. Procédez de la même façon pour les enregistrements situés dans la catégorie Saved Records.

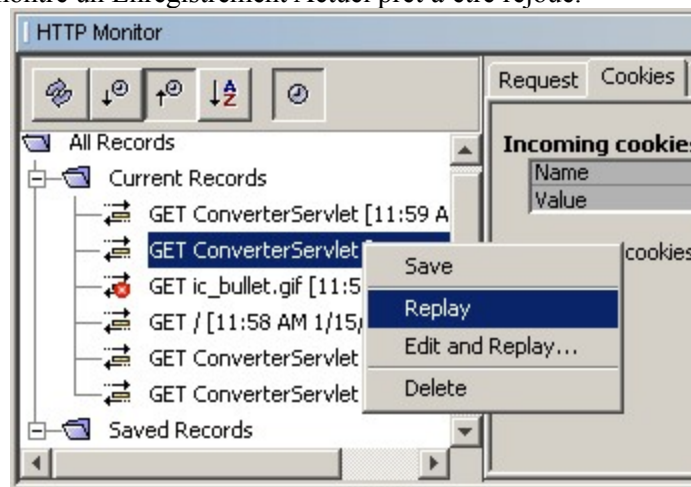
## Rejouer des Requêtes HTTP

La fonctionnalité la plus puissante du Moniteur HTTP est sa capacité d'édition et de rejouer des requêtes HTTP. En éditant et rejouant des requêtes HTTP, vous pouvez rapidement et facilement tracer et diagnostiquer des problèmes avec le flot de données depuis l'exécution des servlets et pages JSP sur le serveur web. Lorsque vous rejouez une requête HTTP, la réponse apparaît dans votre navigateur web. Donc, vous pouvez visuellement voir le résultat d'une requête HTTP qui est rejouée dans votre navigateur web.

### Conseil EDI NetBeans

L'EDI ouvre votre navigateur web par défaut lorsque vous rejouez les requêtes HTTP. Si l'EDI ne peut trouver le navigateur web par défaut du système d'exploitation, vous pouvez configurer le navigateur web manuellement. Voyez Modification du Navigateur Web par défaut de l'EDI à la page XXX.

Aussi bien les Enregistrements Actuels que les Enregistrements Sauvegardés peuvent être rejoués. Pour rejouer une requête HTTP, sélectionnez la requête HTTP à rejouer dans l'arborescence du panneau de gauche et choisissez l'option Replay dans le menu contextuel. L'Illustration 6-9 montre un Enregistrement Actuel prêt à être rejoué.



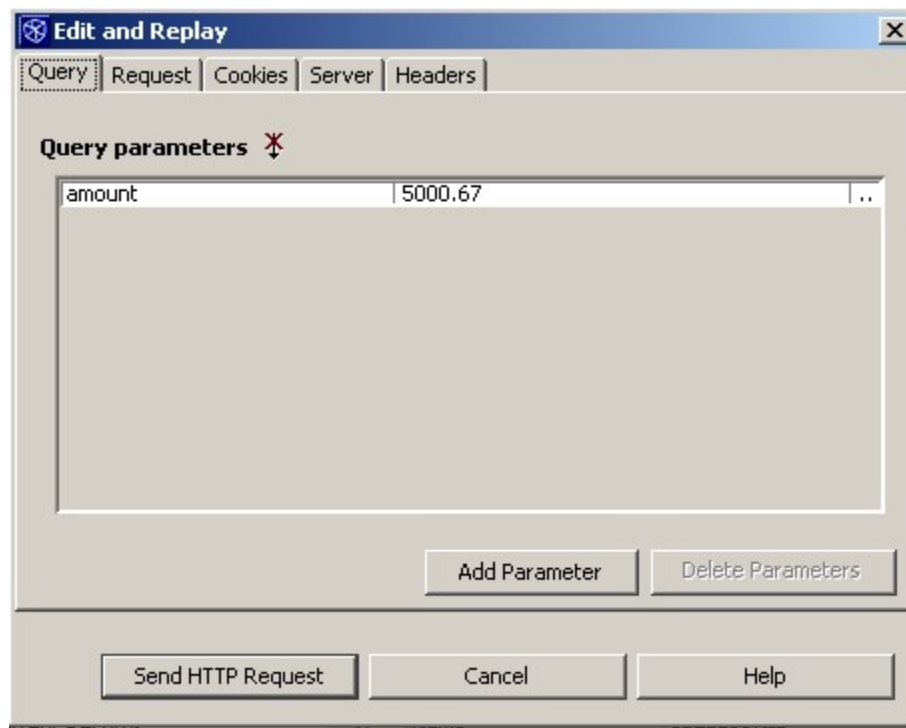
### Illustration 6-9

*Sélection d'un enregistrement pour être rejoué dans le Moniteur HTTP*

Notez que l'enregistrement sélectionné est rejoué dans votre navigateur après que vous ayez choisi Replay dans le menu contextuel.

En plus de rejouer les requêtes HTTP, vous pouvez également éditer une requête HTTP avant de la rejouer. Pour éditer une requête HTTP avant de la rejouer, sélectionnez une requête HTTP dans l'arborescence du panneau de gauche, que ce soit un enregistrement Actuel ou un Enregistrement Sauvegardé, cliquez-droit sur l'enregistrement et choisissez l'option Edit and Replay du menu contextuel. REMARQUE: Vous ne pouvez sélectionner plusieurs enregistrements dans le but de les éditer et les rejouer. Après avoir choisi l'option Edit and Replay, une boîte de dialogue est affichée où vous pouvez faire de nombreuses modifications.

Les modifications supportées dans le Edit and Replay d'une requête HTTP inclut des options pour éditer un paramètre dans la query, modifier les paramètres URI de la requête, modifier les cookies, modifier les propriétés d'exécution du serveur et modifier l'entête de la requête HTTP.



### Illustration 6-10

*Moniteur HTTP, boîte de dialogue Edit and Replay*

Dans l'onglet Query, sur l'illustration Illustration 6-10, vous pouvez ajouter un paramètre de requête ou supprimer un paramètre de requête et modifier les paramètres de requête de l'URI. Dans l'onglet Request, vous pouvez modifier l'URI de la requête en sélectionnant le bouton (...) situé près de la valeur du paramètre de requête. Vous pouvez modifier la méthode de la requête d'un GET vers un POST ou un PUT, en sélectionnant la liste déroulante située à droite du type Request Method dans la colonne de gauche. Vous pouvez également modifier le Protocole de Requête en sélectionnant le bouton (...) situé dans la colonne de droite du Protocole de Requête. Dans l'onglet Cookies, vous pouvez ajouter, modifier ou supprimer des cookies associés à la requête HTTP. Dans l'onglet Server, vous pouvez modifier les valeurs d'exécution du serveur comme le host name, et le port où la requête HTTP doit être exécutée. Vous pouvez ajouter, modifier ou supprimer des en-têtes HTTP dans l'onglet Headers.

Après avoir fait les modifications que vous désiriez à la requête HTTP, vous pouvez rejouer la requête modifiée en pressant le bouton Send HTTP Request. La requête HTTP qui en résulte sera alors envoyée vers votre navigateur web où le résultat sera affiché.