



Création de services Web et d'un squelette EJB à partir d'un fichier WSDL

Table des matières

Création de services Web et d'un squelette EJB à partir d'un fichier WSDL 1

Leçon 1.1 : Configuration de l'espace de travail et création des projets requis	1
Créer un serveur WebSphere Application Server v6.1	1
Définition du niveau de compatibilité WS-I	1
Création du projet EJB de service Web	2
Création du projet routeur de service Web	2
Ajout de projets au serveur	3
Point de contrôle de la leçon	3
Leçon 1.2 : Importation et validation du fichier WSDL	3
Validation du document WSDL	4

Point de contrôle de la leçon	4
Leçon 1.3 : Création du service Web	4
Création d'un service Web à partir d'un fichier WSDL	5
Point de contrôle de la leçon	7
Leçon 1.4 : Implémentation des méthodes de conversion des températures	7
Point de contrôle de la leçon	8
Leçon 1.5 : Validation de la compatibilité WS-I pour le trafic du service Web	8
Routage du trafic et vérification de la compatibilité WS-I	9
Point de contrôle de la leçon	10
Récapitulatif	10

Création de services Web et d'un squelette EJB à partir d'un fichier WSDL

Objectifs

Vous allez apprendre à créer un service Web compatible WS-I à partir d'un fichier WSDL. L'assistant génère un squelette EJB qui contient un ensemble de méthodes correspondant aux opérations décrites dans le document WSDL. Lorsque l'EJB est créé, chaque méthode dispose d'une implémentation générique que vous pouvez facilement remplacer en modifiant l'EJB.

Temps requis

L'exécution de ce tutoriel prend **1 heure et 30 minutes** environ. Si vous décidez d'explorer d'autres facettes des services Web ou des EJB dans le cadre de ce tutoriel, le délai d'exécution peut être plus long.

Conditions requises

Pour exécuter ce tutoriel, vous devez maîtriser les concepts suivants :

- Concepts de base des services Web, tels que SOAP et WSDL
- Notions de base du langage XML
- Programmation des EJB

Lorsque vous êtes prêt, commencez la leçon 1.1 : Configuration d'un espace de travail et création des projets requis

[Retour d'informations](#)

Information associée



[Afficher la version PDF](#)

Leçon 1.1 : Configuration de l'espace de travail et création des projets requis

Créer un serveur WebSphere Application Server v6.1

Pour créer un serveur WebSphere Application Server, effectuez les opérations suivantes :

1. Dans le menu **Fichier**, sélectionnez **Nouveau** → **Autre** → **Serveur** → **Serveur** → **Suivant**.
2. Sélectionnez **Serveur WebSphere v6.1** comme type de serveur. Cliquez sur **Suivant**.
3. Si cette phase d'exécution n'a toujours pas été créée dans votre espace de travail, vous serez invité à sélectionner le répertoire d'installation pour le serveur. Cliquez sur **Suivant**.
4. Acceptez le port et le nom de serveur définis par défaut. Cliquez sur **Terminer**.
5. Attendez le démarrage du serveur. Une fois le serveur démarré, la console affiche le message `Serveur serveur1 prêt pour l'e-business;`

Définition du niveau de compatibilité WS-I

WS-I désigne l'interopérabilité des services Web, y compris l'interopérabilité des plateformes, des systèmes d'exploitation et des langages de programmation.

L'organisme WS-I définit des standards regroupés dans des documents appelés Profils, qui établissent les règles à respecter pour assurer l'interopérabilité d'un service Web. Les produits Rational Developer

valident les services Web conformément à WS-ISSBP (WS-I Simple SOAP Binding Profile 1.0) et WS-I AP (WS-I Attachments Profile 1.0). Pour plus d'informations sur WS-I, visitez le site <http://www.ws-i.org/>

Par défaut, le niveau de compatibilité SSBP WS-I est défini sur **Ignorer**. Avec ce paramètre, aucun avertissement ne sera émis si des choix non compatibles sont effectués. Ce niveau de compatibilité est utilisé par les assistants de services Web et l'outil de validation WSDL. Comme cet exemple génère un service Web compatible WS-I, le niveau de compatibilité WS-I doit être défini sur **Requis**.

Vous pouvez changer le niveau de compatibilité WS-I en procédant comme suit :

1. Dans la barre de menus principale, cliquez sur **Fenêtre** → **Préférences**. La boîte de dialogue Préférences s'affiche.
2. Développez la branche **Services Web** et sélectionnez **WebSphere** → **Compatibilité BSP WS-I**.
3. Sélectionnez l'option **Requis** dans la liste déroulante située en regard de WS-I SSBP.
4. Cliquez sur **OK**.

Création du projet EJB de service Web

Les étapes suivantes de ce tutoriel sont exécutées dans la perspective J2EE. Si le système vous demande si vous souhaitez changer de perspective après avoir exécuté une tâche, cliquez sur **Non**.

Le projet EJB contiendra la logique métier du service Web ainsi que le fichier WSDL.

1. Dans la barre de menus principale, cliquez sur **Fichier** → **Nouveau** → **Projet** → **EJB** → **Projet EJB**. Cliquez sur **Suivant**.
2. Saisissez TempEJB dans la zone de saisie Nom. Sous Phase d'exécution cible, assurez-vous que le serveur cible est WebSphere Application Server v6.1. Dans la zone **Nom de projet EAR**, saisissez TempEJB_EAR comme nom EAR. Cliquez sur **Suivant**.
3. Par défaut, les valeurs correctes pour ce type de projet seront sélectionnées. Cliquez sur **Suivant**.
4. Décochez la case pour la création d'un module JAR client. L'assistant de services Web créera ce module pour vous. Cliquez sur **Terminer**.

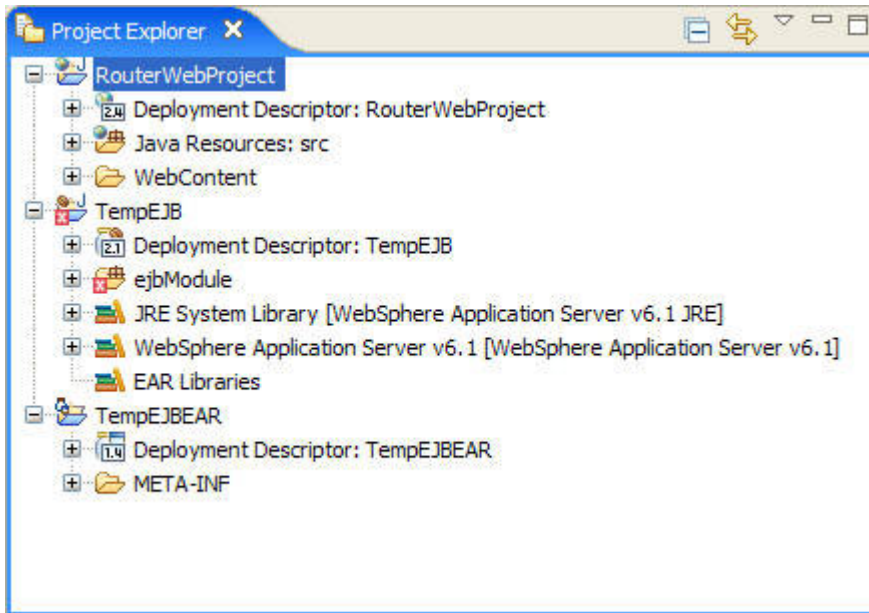
Le projet EJB qui contiendra la logique de service Web et l'EAR associé sera créé. Une erreur sera associée au projet EJB car il ne contient pas de bean enterprise. Le bean sera généré par l'assistant de services Web.

Création du projet routeur de service Web

Les services Web EJB nécessitent un projet routeur. Ce projet contient le servlet routeur qui agit comme noeud final pour le service et qui fera appel à l'EJB. Si vous utilisez SOAP via JMS comme méthode de transport, le projet routeur doit être un projet EJB. Si vous utilisez SOAP via HTTP comme c'est le cas dans ce tutoriel, le projet routeur doit être un projet Web. Le projet que vous créez doit être ajouté au même EAR que le projet EJB qui contiendra le bean enterprise. Ce projet ne doit pas contenir aucune logique métier pour votre service Web.

Vous pouvez créer un projet Web en procédant comme suit :

1. Dans la barre de menus principale, cliquez sur **Fichier** → **Nouveau** → **Projet** → **Web** → **Projet Web dynamique**. Cliquez sur **Suivant**.
2. Saisissez RouterWebProject dans la zone de saisie Nom. Sous Phase d'exécution cible, assurez-vous que le serveur cible est WebSphere Application Server v6.1. Dans la zone **Nom de projet EAR**, assurez-vous que TempEJB_EAR est sélectionné. Cette opération permet de s'assurer que le bean enterprise que vous allez créer et le projet de routeur sont tous les deux référencés dans le même projet EAR. Cliquez sur **Terminer**.
3. Vous avez maintenant créé votre projet routeur et votre espace de travail devrait ressembler à ceci :



Ajout de projets au serveur

Vous pouvez associer le projet au serveur sur lequel le service Web sera exécuté en procédant comme suit :

1. A l'aide du bouton droit de la souris, cliquez sur le serveur dans la vue Serveurs et sélectionnez **Ajouter et supprimer des projets**. Si la vue Serveurs n'est pas ouverte dans l'espace de travail, ouvrez-la à partir du menu **Fenêtre** en sélectionnant **Afficher la vue → Serveurs**.
2. Dans la fenêtre qui s'affiche, sélectionnez TempEJB_EAR qui contient votre routeur et les projets EJB puis cliquez sur **Ajouter**.
3. Cliquez sur **Terminer**.

Point de contrôle de la leçon

Vous pouvez à présent commencer la leçon 1.2 : Importation et validation du fichier WSDL.

[Retour d'informations](#)

Leçon 1.2 : Importation et validation du fichier WSDL

Le document WSDL Conversion des températures est mis à votre disposition pour effectuer l'exercice. Le fichier WSDL que vous devez utiliser dans ce tutoriel permet de convertir les degrés Fahrenheit en degrés Celsius et inversement.

Avant de commencer, vous devez effectuer la leçon 1.1 : Configuration de l'espace de travail et création des projets requis.

Vous pouvez créer un nouveau document WSDL ou importer un document existant. Le document WSDL Conversion des températures utilisé dans ce tutoriel est mis à votre disposition dans un projet simple. Pour importer le document WSDL Conversion des températures dans le plan de travail, procédez comme suit :

1. Dans la vue Explorateur de projets, développez **TempEJB**, cliquez avec le bouton droit de la souris et cliquez sur **Nouveau → Dossier** pour créer un dossier appelé **wsdl**. Cliquez sur **Terminer**.
2. Importez le projet simple contenant le fichier WSDL

3. Développez le projet simple importé appelé TempConversionWSDL et développez le dossier WebContent. Le fichier ConvertTemperature.wsdl s'y trouvera.
4. Cliquez avec le bouton droit de la souris sur le fichier ConvertTemperature.wsdl et sélectionnez **Déplacer...**
5. Sélectionnez le dossier wsdl que vous avez créé dans le projet TempEJB en tant que destination et cliquez sur **OK**.

Validation du document WSDL

Le valideur WSDL peut valider la sémantique WSDL et la compatibilité WS-I.

Vous pouvez valider le document WSDL Convertisseur de températures en procédant comme suit :

1. Sélectionnez le document ConvertTemperature.wsdl dans l'Explorateur de projets.
2. Cliquez avec le bouton droit de la souris et cliquez sur **Valider**. Les erreurs éventuelles s'affichent dans la vue Tâches.

Si aucune erreur ne se produit pendant la validation, vous pouvez continuer à créer le service Web.

Point de contrôle de la leçon

Vous pouvez à présent commencer la leçon 1.3 : Création du service Web.

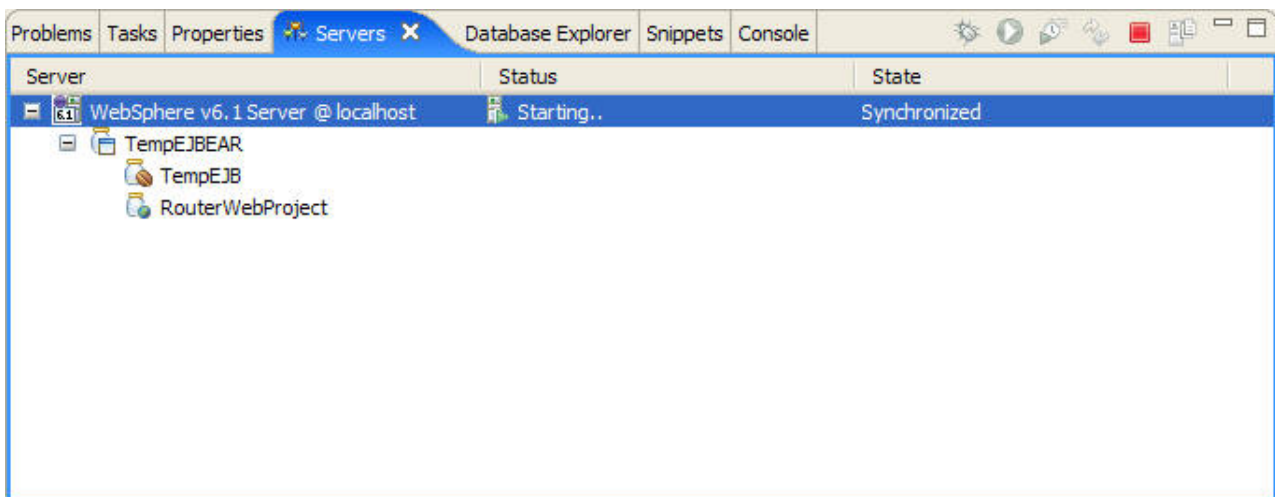
[Retour d'informations](#)

Leçon 1.3 : Création du service Web

Avant de commencer, vous devez effectuer la leçon 1.2 : Importation et validation du fichier WSDL.

Avant de tenter de créer un service Web, il est vivement conseillé de démarrer le serveur WebSphere Application Server sur lequel le service Web doit s'exécuter. Il est possible de démarrer le serveur en utilisant les assistants de services Web mais cette opération peut prendre plusieurs minutes en fonction de la vitesse de votre système. Le démarrage du serveur avant la procédure permet d'accélérer l'exécution de l'assistant et de limiter le risque d'erreurs générées par l'assistant lorsque le démarrage du serveur prend trop de temps.

Pour démarrer le serveur, cliquez sur celui-ci à l'aide du bouton droit de la souris dans la vue Serveurs et sélectionnez **Démarrer** :



Si la vue Serveurs n'est pas ouverte dans l'espace de travail, ouvrez-la à partir du menu **Fenêtre** en sélectionnant **Afficher la vue** → **Serveurs**.

Création d'un service Web à partir d'un fichier WSDL

L'assistant de services Web permet de créer, de configurer pour le déploiement et de déployer un service Web sur un serveur. Une fois que le service Web est déployé, l'assistant vous aide à générer le proxy client et l'exemple d'application pour tester le service Web.

Une fois les tests terminés, vous pouvez publier le service Web dans un registre d'entités UDDI à l'aide de l'assistant d'exportation.

1. Dans la vue Explorateur de projets, sélectionnez le document **ConvertTemperature.wsdl** dans votre projet EJB.
2. Cliquez sur **Fichier** → **Nouveau** → **Autre**. Sélectionnez **Services Web** pour afficher les différents assistants de services Web. Sélectionnez l'assistant **Service Web**. Cliquez sur **Suivant**.
3. Sélectionnez les options suivantes sur la première page de l'assistant :
 - Type de service Web : service Web EJB descendant
 - Définition de service : assurez-vous que le fichier **ConvertTemperature.wsdl** que vous avez importé est sélectionné.
 - Règle de génération de niveau de service : déplacez la règle sur **Tester le service**.
 - Configuration de service : assurez-vous que **WebSphere v6.1 Server** et l'environnement d'exécution **IBM WebSphere JAX-RPC** sont sélectionnés. Cliquez sur **Projet de service** et entrez **TempEJB** comme nom de projet de service. **TempEJB** doit être sélectionné comme projet EAR de service. Ne pas
 - Règle de génération de niveau de client : déplacez la règle sur **Tester le client**.
 - Configuration de service : assurez-vous que **WebSphere v6.1 Server** et l'environnement d'exécution **IBM WebSphere JAX-RPC** sont sélectionnés. L'assistant créera un client et un projet EAR client. Vous pouvez accepter les noms par défaut ou entrer un nom différent.
 - Surveiller le service Web.

Une fois les options correctes sélectionnées, l'assistant devrait ressembler à ceci :

Web Service

Web Services

Review your Web service options and make any necessary changes before proceeding to the next page.

Web service type: Top down EJB Web Service

Service definition: /WebProject/WebContent/wsdl/ConvertTemperature.wsdl [Browse...](#)

Test service

Configuration:

- [Server: WebSphere v6.1 Server](#)
- [Web service runtime: IBM WebSphere JAX-RPC](#)
- [Service project: TempEJB](#)
- [Service EAR project: TempEJBEAR](#)

Client type: Java Proxy

Test client

Configuration:

- [Server: WebSphere v6.1 Server](#)
- [Web service runtime: IBM WebSphere JAX-RPC](#)
- [Client project: TempEJBClient](#)
- [Client EAR project: TempEJBClientEAR](#)

☐ Publish the Web service

☒ Monitor the Web service

☐ Do not show me this dialog box again.

[? < Back](#) [Next >](#) [Finish](#) [Cancel](#)

Cliquez sur **Suivant**.

- Dans la page Configuration EJB d'un squelette EJB de service Web, sélectionnez RouterWebProject comme projet de routeur s'il n'est pas déjà sélectionné.
- La page Test du service Web permet de sélectionner le dispositif à utiliser pour tester le service Web avant le développement d'un client ou d'un proxy. Sélectionnez l'Explorateur de services Web comme dispositif de test pour le service Web et cliquez sur **Lancer**. Plusieurs secondes peuvent s'écouler avant que le serveur WebSphere Application Server ne démarre.
- L'Explorateur de services Web s'affiche dans un navigateur Web. Sélectionnez **fahrenheitToCelsius** ou **celsiusToFahrenheit** dans la liste des opérations. Entrez un nombre dans la zone Valeur et cliquez sur **Exécuter**. Une implémentation générique de chacune de ces opérations est fournie et la valeur par

défaut -3 est renvoyée. Si ces deux opérations s'exécutent correctement, fermez la fenêtre du navigateur et cliquez sur **Suivant** dans l'assistant de services Web.

7. Dans la page Proxy de service Web, conservez la sélection Configuration de sécurité sur Aucune sécurité pour assurer la compatibilité WS-I. Cliquez sur **Suivant**.
8. Dans la page Test du client de service Web, vérifiez que les options **Tester le proxy généré** et **Exécuter le test sur le serveur** sont toutes les deux sélectionnées. Dans la section Méthodes, assurez-vous que toutes les méthodes sont sélectionnées ou cliquez sur **Sélectionner tout** pour sélectionner toutes les méthodes. Si vous souhaitez publier le service Web dans un registre UDDI, cliquez sur **Suivant** pour configurer les options de publication du service Web. Toutefois, cette étape n'est pas traitée dans ce tutoriel. Sinon, cliquez sur **Terminer**.
9. L'exemple d'application est lancé dans un navigateur Web. Vous pouvez utiliser cette application pour tester le service Web en sélectionnant une méthode dans la section Méthodes, en entrant une valeur dans la section Entrées et en cliquant sur **Appeler** pour afficher les résultats dans la section Résultat. Ne fermez pas encore la fenêtre du navigateur contenant le fichier TestClient.jsp. Elle doit être utilisée ultérieurement dans ce tutoriel pour tester la compatibilité WS-I du trafic de service Web.

Point de contrôle de la leçon

Vous pouvez à présent commencer la leçon 1.4 : Implémentation des méthodes de conversion des températures.

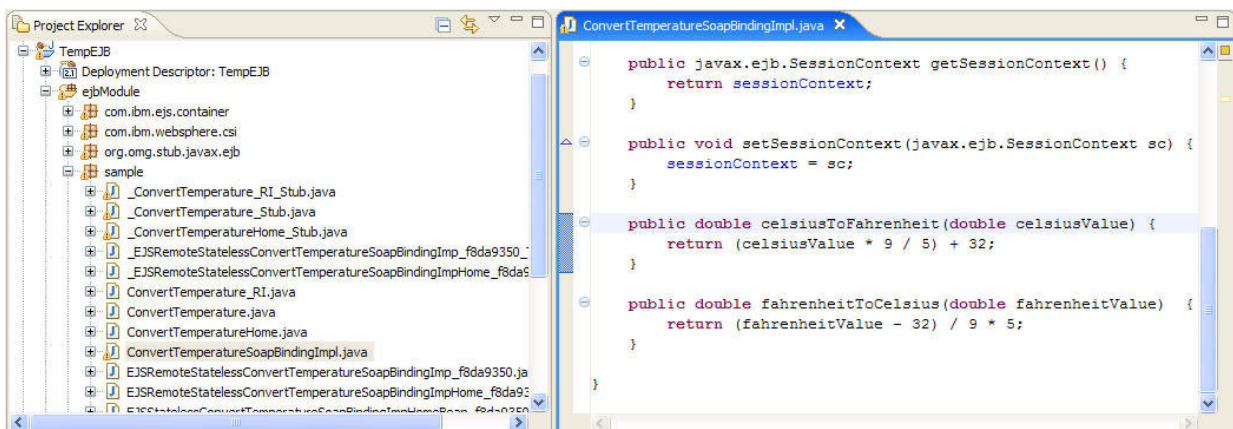
[Retour d'informations](#)

Leçon 1.4 : Implémentation des méthodes de conversion des températures

Avant de commencer, vous devez effectuer la leçon 1.3 : Création d'un service Web.

Des implémentations génériques des méthodes **fahrenheitToCelsius** et **celsiusToFahrenheit** ont été automatiquement générées lorsque vous avez créé un service Web à partir du document WSDL. Cette section vous explique comment remplacer ces implémentations génériques par du code pertinent et comment tester les nouvelles méthodes.

1. Dans la vue Explorateur de projets, sélectionnez **ConvertTemperatureSoapBindingImpl.java** sous **TempEJB** → **ejbModule** → **sample**.
2. Recherchez la méthode **fahrenheitToCelsius** et remplacez l'implémentation actuelle par : `return (fahrenheitValue - 32) / 9 * 5;`
3. Recherchez la méthode **celsiusToFahrenheit** et remplacez l'implémentation actuelle par : `return (celsiusValue * 9 / 5) + 32;`



4. Sélectionnez **Fichier** → **Sauvegarder** pour sauvegarder vos modifications.
5. Redémarrez l'EAR en développant **WebSphere v6.1 Server** dans la vue Serveurs et en cliquant avec le bouton droit de la souris sur **TempEJB EAR** → **Redémarrer TempEJB EAR**.
6. Cliquez sur **Exécuter** → **Lancer** l'explorateur de services Web dans la barre de menus principale et répétez les instructions de la section précédente pour tester les méthodes **fahrenheitToCelsius** et **celsiusToFahrenheit**.

Point de contrôle de la leçon

Vous pouvez à présent commencer la leçon 1.5 : Validation de la compatibilité WS-I pour le trafic de service Web.

[Retour d'informations](#)

Leçon 1.5 : Validation de la compatibilité WS-I pour le trafic du service Web

Avant de commencer, vous devez effectuer la leçon 1.4 : Implémentation des méthodes de conversion des températures.

Pour vous assurer que les paires demandes/réponses de l'enveloppe SOAP sont compatibles avec WS-I, vous devez faire transiter le trafic du service Web par le moniteur TCP/IP :

Lorsque vous créez un service Web à l'aide des assistants Service Web ou Client de service Web, vous pouvez décider de configurer et d'exécuter automatiquement le moniteur TCP/IP. Comme vous avez sélectionné cette option lors de la création du service Web, la vue Moniteur TCP/IP doit apparaître dans l'espace de travail. Dans le cas contraire, vous pouvez l'ouvrir en sélectionnant **Fenêtre** → **Afficher la vue** → **Autre** → **Débogage** → **Moniteur TCP/IP** .

Vous pouvez également configurer le moniteur TCP/IP manuellement en effectuant les opérations suivantes :

1. Dans l'exemple d'application, appelez la méthode `getEndPoint`. Enregistrez ce noeud final.
2. Créez un serveur qui assurera la fonction de moniteur TCP/IP :
 - a. Dans le menu **Fenêtre**, sélectionnez **Préférences**.
 - b. Dans la fenêtre Préférences, développez **Exécuter/Déboguer** puis sélectionnez **Moniteur TCP/IP**.
 - c. Cochez la case **Afficher la vue Moniteur TCP/IP en cas d'activité**.
 - d. Dans la liste Moniteurs TCP/IP, cliquez sur **Ajouter**. La boîte de dialogue Nouveau moniteur s'affiche.
 - e. Définissez les paramètres suivants :

Option	Description
Port de surveillance local	Indiquez un numéro de port unique sur le système local.
Nom d'hôte	Indiquez le nom d'hôte ou l'adresse IP du système où le serveur s'exécute.
Port	Indiquez le numéro de port du serveur distant.

Option	Description
Type	Indiquez si le type de requête émis par le navigateur Web est envoyé via HTTP ou TCP/IP. Si l'option HTTP est sélectionnée, les demandes du navigateur Web sont modifiées afin que l'en-tête HTTP indique le système distant et elles sont séparées si plusieurs requêtes HTTP sont reçues sur la même connexion. Si l'option TCP/IP est sélectionnée, toutes les requêtes sont envoyées telles quelles.

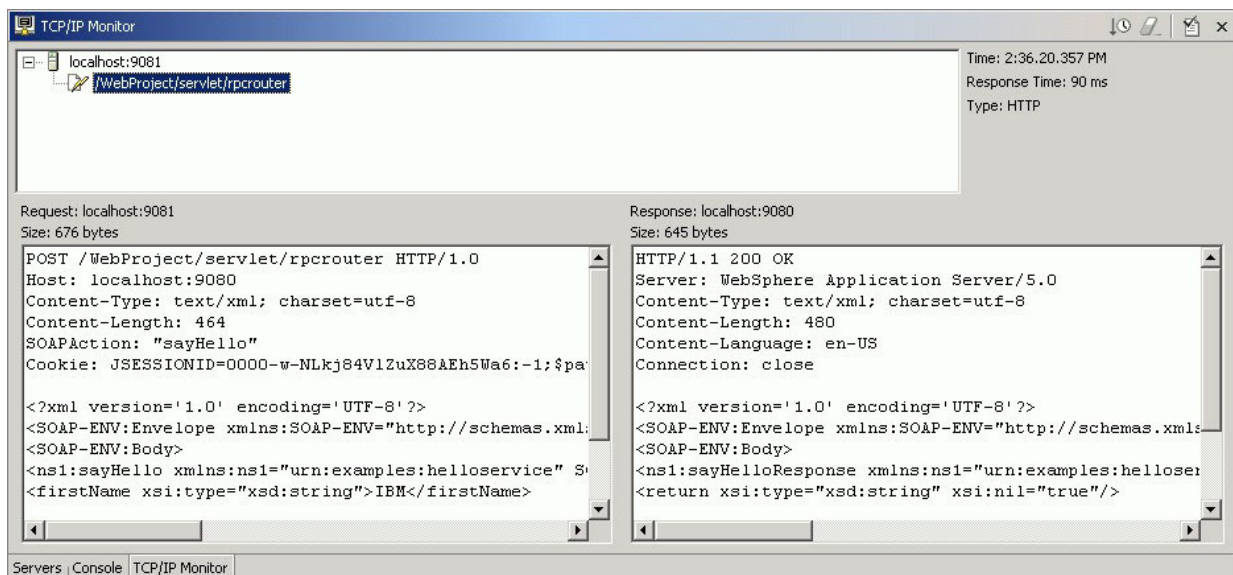
- Pour que le service Web transite par le moniteur, vous devez modifier le noeud final du client de service Web. Le moniteur TCP/IP écoute le port 9081. Dans la fenêtre de navigateur Web utilisée à l'étape 1, appelez la méthode `setEndPoint` et modifiez le noeud final afin de l'associer au port 9081. Par exemple, la valeur par défaut est `http://localhost:9081/racine_contexte_module_web/servlet/rpcrouter`. Appelez la méthode `getEndPoint` une nouvelle fois pour vérifier que la modification a été prise en compte.


Description ou affichage des résultats des étapes précédentes.

Routage du trafic et vérification de la compatibilité WS-I

Vous pouvez faire transiter le trafic par le moniteur TCP/IP et tester sa compatibilité WS-I en procédant comme suit :

- Sélectionnez la méthode de service Web dans la sous-fenêtre Méthodes. Appelez cette méthode.
- Accédez à la vue Moniteur TCP/IP en cliquant sur l'onglet Moniteur TCP/IP dans la vue Serveurs. Cette opération permet d'afficher les paires demandes/réponses qui transitent par le moniteur TCP/IP. Le résultat obtenu sera de la forme suivante :



- Pour vérifier que le trafic SOAP est compatible WS-I, vous pouvez générer un fichier journal en cliquant sur l'icône . Dans la boîte de dialogue qui s'affiche, sélectionnez un nom pour le fichier journal et indiquez où vous souhaitez le sauvegarder. Ce fichier journal sera validé pour la compatibilité WS-I. Vous pouvez ouvrir le fichier journal dans un éditeur XML pour examiner son contenu.

Point de contrôle de la leçon

Terminez l'exécution du tutoriel en consultant les éléments de la rubrique Récapitulatif.

[Retour d'informations](#)

Récapitulatif

Vous avez créé un service Web de conversion des températures compatible WS-I et un squelette EJB à partir d'un fichier WSDL, et appris à implémenter des méthodes EJB de base.

Leçons acquises

Si vous avez effectué tous les exercices, vous êtes en mesure de :

- Définir le niveau de compatibilité WS-I.
- Créer un projet Web appelé WebProject.
- Importer le document WSDL de conversion des températures.
- Valider le document WSDL et sa compatibilité WS-I.
- Créer le service Web et le squelette EJB et tester les méthodes incluses dans le service Web à l'aide des services Web
- Implémenter les méthodes fahrenheitToCelsius et celsiusToFahrenheit (facultatif)
- Déployer le service Web sur le serveur WebSphere Application Server.
- Tester le trafic de service Web pour vérifier sa compatibilité WS-I.

Informations complémentaires

Pour plus d'informations sur les services Web, WSDL, SOAP et l'environnement d'exécution WebSphere v6, reportez-vous à l'aide en ligne de WebSphere Studio en cliquant sur **Aide** → **Table des matières**. Pour consulter des articles techniques plus approfondis sur les services Web, consultez DeveloperWorks

[Retour d'informations](#)

(C) Copyright IBM Corporation 2000, 2004. All Rights Reserved.