



# Application de scénario de magasin point de vente iSeries

*Version 5.1 pour Windows*





# Application de scénario de magasin point de vente iSeries

*Version 5.1 pour Windows*



# Table des matières

<b>Avis aux lecteurs canadiens . . . . .</b>	<b>v</b>
<b>Chapitre 1. Introduction à l'application de scénario de magasin point de vente iSeries . . . . .</b>	<b>1</b>
Scénarios basés sur des projets . . . . .	2
<b>Chapitre 2. Concepts technologiques . . . . .</b>	<b>5</b>
<b>Chapitre 3. Exécution du scénario . . . . .</b>	<b>9</b>
Avant de commencer . . . . .	9
Installation des fichiers exemple . . . . .	9
Restauration des fichiers .savf . . . . .	10
Modification de la liste des bibliothèques . . . . .	11
Importation et exécution de l'application dans Development Studio . . . . .	12
Extraction des fichiers .zip sur votre système local . . . . .	13
Création des projets Web . . . . .	13
Importation de fichiers de projet Web . . . . .	14
Importation de fichiers de projet WebFacing et création du projet WebFacing . . . . .	14
Configuration du serveur . . . . .	15
Configuration du projet WebFacing . . . . .	17
Définition des informations relatives au serveur . . . . .	17
Exécution de l'application dans l'atelier . . . . .	18
Exécution de l'application en tant que client dans l'atelier . . . . .	18
Exécution de l'application en tant qu'administrateur dans l'atelier . . . . .	19
Déploiement de l'application de scénario sur WebSphere Application Server . . . . .	19
Configuration de WebSphere Application Server . . . . .	20
Sécurisation de la page de l'administrateur . . . . .	21
Création de fichiers EAR pour le déploiement iSeries WebSphere Application Server . . . . .	22
Déploiement des fichiers EAR sur iSeries WebSphere Application Server . . . . .	22
Exécution de l'application sur WebSphere Application Server . . . . .	23
<b>Chapitre 4. Module d'initiation 1 : Création d'un service Web pour afficher le prix des produits (SV000514) . . . . .</b>	<b>25</b>
Introduction . . . . .	25
Avant de commencer . . . . .	25
Création d'un nouveau projet Web . . . . .	26
Définition des informations relatives au serveur iSeries . . . . .	26
Création du programme de service RPG . . . . .	26
Création des paramètres et génération du bean Java . . . . .	27
Création d'un service Web à partir du bean Java . . . . .	28
Test du modèle . . . . .	29
<b>Chapitre 5. Module d'initiation 2 : Création d'une interface pour afficher le stock et les articles commandés (SV000501) . . . . .</b>	<b>31</b>
Introduction . . . . .	31
Avant de commencer . . . . .	32
Création d'un projet WebFacing . . . . .	32
Conversion de la source DDS . . . . .	33
Configuration UTF-8 dans l'atelier — Pour les utilisateurs WAS version 4.0 uniquement . . . . .	34
Configuration UTF-8 pour WebSphere Application Server . . . . .	35
Création d'une feuille de style . . . . .	36
(Facultatif) Personnalisation manuelle de la feuille de style en cascade . . . . .	36
Extension et amélioration de votre projet WebFacing avec une interaction Web . . . . .	36
Liaison du projet avec une interaction Web . . . . .	39
Publication des fichiers et redémarrage du serveur . . . . .	40
Test de l'interface . . . . .	41
<b>Chapitre 6. Module 1 avancé : Création de code HTML, de servlets et de fichiers JSP pour passer des commandes client sur un serveur iSeries (SV001585). . . . .</b>	<b>45</b>
Introduction . . . . .	45
Récapitulatif des grandes étapes . . . . .	46
Avant de commencer . . . . .	46
Création de la page Web, des servlets et des fichiers JSP . . . . .	47
<b>Chapitre 7. Module 2 avancé : Création d'un projet Web qui utilise les services Web SV000514 et SV001586 (SV000618) . . . . .</b>	<b>55</b>
Introduction . . . . .	55
Avant de commencer . . . . .	57
Création des pages Web, des servlets, des fichiers JSP et du code RPG . . . . .	57
Avant le déploiement sur WebSphere Application Server . . . . .	59
<b>Chapitre 8. Remarques . . . . .</b>	<b>61</b>
LICENCE DE COPYRIGHT . . . . .	62
Informations relatives à l'interface de programmation . . . . .	63
Marques . . . . .	64



---

## Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

### Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

### Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

### Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








### OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

### Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

<b>France</b>	<b>Canada</b>	<b>Etats-Unis</b>
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

### **Brevets**

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

### **Assistance téléphonique**

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.



---

## Chapitre 1. Introduction à l'application de scénario de magasin point de vente iSeries

Ce module de scénario est un modèle d'application développé à l'aide d'IBM WebSphere Development Studio Client for iSeries, la technologie Eclipse du serveur iSeries. Ce scénario s'adresse aux développeurs qui souhaitent utiliser un serveur iSeries pour effectuer les tâches suivantes :

- Développement Java
- Développement Web
- Gestion et transformation de code RPG en applications Web

Cette application comprend deux fichiers .savf que vous devez restaurer sur votre serveur iSeries et cinq fichiers .zip que vous devez importer dans le produit client. Ils forment une série de pages Web comprenant des URL désignant différents points du processus.

**Remarque importante :** Pour prendre connaissance de tous les modules de mise à jour, correctifs et autres informations, consultez notre page de support : [ibm.com/software/awdtools/wdt400/support/](http://ibm.com/software/awdtools/wdt400/support/).

Ce scénario porte sur différents composants du produit, en particulier sur des composants iSeries spécifiques :

- Outil IBM WebFacing
- Services Web
- Outils de développement Web pour iSeries (assistant d'interaction Web et assistant d'appel de programme inclus)
- Outils de développement Java pour iSeries
- IBM Toolbox for Java

Le scénario illustre la situation de deux entreprises, un grossiste et un magasin de vêtements, qui travaillent ensemble et utilisent leur serveur iSeries pour la logique et les données de gestion. Dans le passé, ces entreprises communiquaient par courrier électronique, par téléphone et par fax pour vérifier le stock, passer et suivre les commandes. A présent, elles envisagent d'utiliser le Web pour traiter leurs transactions commerciales standard.

Le magasin point de vente souhaite disposer d'un site Web où :

- Les clients peuvent acheter des produits
- Les employés peuvent acheter du stock auprès du grossiste

Le grossiste souhaite :

- Recevoir en ligne les commandes du détaillant pour le suivi
- Servir de nombreux clients potentiels

Dans ce scénario, vous jouerez le rôle du consultant en programmation des deux entreprises en les aidant à mettre leur entreprise sur le Web.

L'application possède différents points d'entrée en fonction du type d'utilisateur. Si vous êtes un client, vous commencez par visualiser les produits proposés par le magasin (des vêtements de loisirs, dans notre cas). Si vous souhaitez effectuer un

achat, vous pouvez cliquer sur un lien pour accéder à un écran de commande. Lorsque vous avez passé votre commande, une page récapitulative est générée et vous pouvez poursuivre vos achats, annuler ou confirmer la commande.

Si vous êtes un administrateur, vous disposez d'un ID utilisateur sécurisé pour accéder à cette application. Il est donc nécessaire de définir la stratégie de sécurité de cette application lors du déploiement. Votre point d'entrée est un écran de connexion, qui vous permet de visualiser vos commandes, de visualiser le stock et de commander des produits auprès du grossiste. Vous pouvez sélectionner des articles, vérifier le prix de gros le plus récent et commander des articles dans la taille et la quantité de votre choix. L'application vérifie si le grossiste dispose de la taille et de la quantité que vous souhaitez, puis confirme la commande ou indique qu'il ne peut pas honorer la commande pour le moment.

Dans l'ombre, l'application exécute de nombreuses actions portant sur les différents composants du produit. Le tableau suivant présente le processus et les composants du produit pour chaque partie de l'application. Consultez les informations détaillées sur l'exécution de chaque tâche.

Tableau 1.

Tâches de l'application client	Tâches de l'application administrateur	Processus sous-jacent
Afficher les prix des produits		Utilisez un programme RPG iSeries pour créer un service Web et des outils de développement Web pour visualiser et afficher les prix.
Passer une commande auprès du magasin		Utilisez les servlets et les fichiers JSP avec les outils de développement Java iSeries, les outils de développement Web iSeries et IBM Toolbox for Java pour accéder au stock et l'afficher sur un serveur iSeries, passer des commandes et afficher un récapitulatif des achats.
	Visualiser le stock	Utilisez l'outil IBM WebFacing pour convertir un programme RPG existant en application Web et utilisez les outils de développement pour personnaliser la page Web.
	Commander des produits au grossiste à l'aide de l'ID et de la quantité de marchandises	Créez un service Web qui sera appelé lorsque vous appuyez sur le bouton <b>Acheter</b> .
Visualiser la page Web initiale du magasin	Visualiser la page Web initiale du stock	Utilisez les outils de développement Web iSeries pour créer les deux pages d'accueil.

## Scénarios basés sur des projets

L'application de scénario iSeries comprend cinq projets, appelés SV000501, SV000514, SV000618, SV001585 et SV001586. Le présent guide explique comment exécuter l'application préintégré. Il comprend également des modules d'initiation et des modules avancés, qui vous expliquent comment générer les différents projets. Les modules d'initiation s'adressent aux développeurs qui débutent dans le

développement d'applications et dans Development Studio. Les modules avancés s'adressent aux développeurs qui connaissent bien le développement d'applications et Development Studio Client.

**Remarque :** Il existe cinq projets, mais seuls quatre modules sont fournis car les projets SV001586 et SV000514 sont des modules Services Web combinés.

Ce scénario se présente sous la forme de cinq projets :

**Projet SV000501 : Création d'un projet pour afficher les commandes en attente, le stock et les détails des produits** – Créé à l'aide des Outils de développement Web iSeries et de l'outil IBM WebFacing, ce projet s'adresse aux programmeurs RPG connaissant peu le développement d'applications Web et souhaitant utiliser l'outil IBM WebFacing pour publier leurs applications RPG sur le Web.

**Projet SV001585 : Création de code HTML, de servlets et de fichiers JSP pour passer des commandes client sur un serveur iSeries** – Ce projet utilise les classes SQL et JDBC d'IBM Toolbox for Java, le bean RecordIOManager des Outils de développement Java iSeries et l'assistant d'appel de programme iSeries. Ces éléments présentent différentes méthodes pour accéder et manipuler des données et des programmes résidant sur le serveur iSeries. Ce projet s'adresse aux programmeurs Java et aux développeurs d'applications Web souhaitant développer des pages Web pour accéder aux données et au code iSeries. Par ailleurs, vous devez disposer de connaissances en matière de gestion de serveurs iSeries et de programmation RPG.

**Projet SV000514 : Création d'un service Web iSeries pour afficher les prix des produits** – L'assistant de services Web utilise un bean Java généré par l'assistant d'appel de programme iSeries pour appeler une ou plusieurs procédures de programme sur le serveur iSeries et renvoyer les informations à un navigateur. Ce projet s'adresse aux programmeurs RPG souhaitant utiliser Services Web pour créer des applications modulaires et autonomes pouvant être décrites, publiées, localisées et appelées sur le Web.

**Projet SV001586 : Création d'un service Web pour passer des commandes auprès du grossiste via un serveur iSeries** – Ce service Web accepte un ID marchandise et la quantité demandée, puis effectue la commande auprès du grossiste. Ce projet est un composant du projet SV000514 et s'adresse aux programmeurs RPG souhaitant créer des services Web.

**Projet SV000618 : Création d'un projet Web comme interface entre le bon de commande, le formulaire de stock et l'ordre d'achat générés par l'outil IBM WebFacing** – Ce projet requiert les Outils de développement Web iSeries et comprend la création de fichiers HTML et JSP pour utiliser et connecter les services Web développés dans les projets SV000514 et SV001586. Ce projet s'adresse aux développeurs souhaitant utiliser Services Web et connaissant la programmation RPG et Java.



---

## Chapitre 2. Concepts technologiques

Pour pouvoir utiliser l'application de scénario, vous devez connaître un certain nombre de concepts technologiques, en particulier si vous débutez dans le développement d'applications Web. La liste suivante présente certains concepts que vous rencontrerez lors de l'utilisation de cette application.

### Fichier archive d'entreprise (EAR)

Les fichiers EAR sont des fichiers d'archive Java (JAR) standard portant une extension .ear. Ils peuvent contenir plusieurs projets Web. Vous pouvez utiliser ces fichiers pour regrouper et déployer vos propres applications Web sur le serveur WAS (WebSphere Administrative Server). **Remarque :** Dans l'interface graphique de l'outil de déploiement d'applications J2EE SDK, vous devez créer un fichier EAR et ajouter les fichiers JAR et WAR (Web Archive) au fichier EAR. Si vous utilisez les outils de mise en forme de ligne de commande, vous devez créer dans un premier temps les fichiers JAR et WAR, puis le fichier EAR.

### Outil IBM WebFacing

L'Outil WebFacing permet de convertir des interfaces 5250 existantes en interfaces utilisateur graphiques fonctionnant via un navigateur. En apportant peu ou aucune modification aux applications iSeries d'origine, vous pouvez étendre l'utilisation des programmes à Internet ou à l'intranet.

### Fichier d'archive Java (JAR)

Les fichiers JAR sont des modules compressés de fichiers Java, semblables à des fichiers .zip. Ils permettent de compresser les fichiers de classe, d'images et de son d'une applet Java dans un seul fichier pour un téléchargement plus rapide dans votre navigateur.

### Fichiers JSP (Java Server Page)

Les fichiers JSP permettent d'afficher du contenu dynamique dans des pages HTML statiques. Ces fichiers écrits en Java dépendent du serveur et de la plateforme utilisés. En séparant la présentation et le contenu Web, les fichiers JSP facilitent le travail des développeurs devant modifier rapidement l'apparence et l'affichage de leurs pages Web.

### Bean d'appel de programme

Il s'agit des beans Java générés par l'assistant d'appel de programme. Le premier type de bean est un bean Java ordinaire utilisé par les applications Java. L'autre type peut être utilisé par l'assistant de services Web pour créer un service Web.

### Assistant d'appel de programme

L'assistant d'appel de programme vous aide à créer les beans Java et le fichier PCML associé requis pour appeler un programme ou une procédure iSeries. Il vous invite à entrer des informations concernant des objets programme ou programme de service, ainsi que les paramètres pour ces objets, puis crée les beans Java souhaités (et un fichier PCML).

## **Langage RPG (Report Program Generator)**

Il s'agit d'un langage de programmation procédural utilisé par les programmeurs iSeries. Il vous permet de créer des applications de gestion (programmes de facturation et de saisie de demande, par exemple). La dernière version, ILE RPG IV, étend les fonctionnalités du langage RPG tout en prenant en charge l'expérience des versions précédentes par le programmeur.

## **Servlet**

Il s'agit de programmes côté serveur écrits en Java s'exécutant sur des serveurs disposant de fonctions Java ou sur des serveurs d'applications comme IBM WebSphere Application Server. Les servlets effectuent les tâches spécifiées par le serveur (en générant une réponse HTML pour répondre à des requêtes, par exemple). Par exemple, vous pouvez les utiliser dans une application bancaire en ligne pour répondre à l'utilisateur tout en transmettant des données au serveur.

## **Composants Web**

Les composants Web permettent de définir des objets iSeries, notamment des zones de saisie de données et des boutons de fonction, pour échanger des informations entre des programmes de serveur iSeries et la page Web. Les développeurs peuvent utiliser des composants Web pour enregistrer des événements client (vérification de la syntaxe des zones d'entrée et clic de bouton, par exemple).

## **Assistant d'interaction Web**

Cet assistant fait partie des Outils de développement Web iSeries. Il permet de créer et de gérer les interactions entre les programmes ILE et les pages Web. L'assistant surveille l'affichage des données d'entrée, de sortie et des messages d'erreur, puis dirige les données des zones d'entrée et de sortie vers les programmes ILE. Vous pouvez également utiliser l'assistant d'interaction Web pour mapper des messages d'erreur sur la zone où l'erreur s'est produite pour permettre à l'utilisateur d'identifier facilement la source de l'erreur.

## **Services Web**

Les services Web sont des applications conçues et implémentées pour être utilisées via Internet. Ils sont créés à l'aide de normes ouvertes comme SOAP, WSDL et XML. Les services Web peuvent être utilisés dans des situations professionnelles variées, notamment dans un système de gestion des stocks pour permettre aux clients de vérifier le niveau des stocks via Internet ou de suivre une commande directement depuis un fournisseur.

## **Langage WDSL (Web Services Definition Language)**

Le langage WSDL est un langage XML, qui définit l'interface d'un service Web. Il peut interpréter un service Web et gérer le flux des informations entre le service Web et le programme de serveur. Par exemple, les développeurs utilisent le langage WSDL pour créer l'interface d'un site Web qui affiche le cours des actions.

## **WebSphere Studio Workbench**

IBM WebSphere Development Studio Client for iSeries s'appuie sur WebSphere Studio Workbench, l'implémentation IBM de la plateforme Eclipse. Extensible et universel, l'atelier comprend tous les outils nécessaires pour créer et gérer des applications. Les développeurs peuvent utiliser Development Studio pour intégrer de nouveaux objets à l'environnement de développement grâce à des plug-ins et ajouter en transparence des fichiers Java, des graphiques, de la vidéo, etc.





---

## Chapitre 3. Exécution du scénario

Vous pouvez exécuter les applications *Wholesale* et *Retail store* dans l'atelier Development Studio ou sur WebSphere Application Server pour toutes les plateformes, notamment la plateforme iSeries. Pour obtenir une présentation des applications, consultez le Chapitre 1, «Introduction à l'application de scénario de magasin point de vente iSeries», à la page 1.

A la fin de ce chapitre, vous serez à même d'effectuer les tâches suivantes :

- Restaurer les fichiers exemple et les objets Development Studio Client sur votre serveur iSeries
- Créer des projets Web et un projet WebFacing pour stocker les fichiers
- Importer les fichiers restaurés depuis votre serveur iSeries dans Development Studio Client
- Configurer le serveur WebSphere Application Server (WAS)
- Configurer le projet WebFacing
- Exécuter l'application en tant que client dans l'environnement de test WebSphere
- Exécuter l'application en tant qu'administrateur dans l'environnement de test WebSphere
- Configurer votre serveur WebSphere Application Server
- Sécuriser la page de l'administrateur
- Créer des fichiers EAR pour le déploiement WAS externe
- Déployer l'application sur le serveur WAS
- Exécuter l'application sur le serveur WAS

---

### Avant de commencer

Pour tester les applications à partir de l'atelier, vous devez vérifier les points suivants :

- Vous disposez d'un serveur iSeries V5R1 ou plus récent.
- Vous avez installé les derniers correctifs WebFacing. Consultez notre page de support pour les correctifs :  
<http://www.ibm.com/software/awdtools/wdt400/support/>
- Vous disposez d'un accès NET USE sur le serveur iSeries.

---

### Installation des fichiers exemple

Pour utiliser l'application de scénario iSeries, les fichiers suivants sont nécessaires :

- Wholesale.savf
- Retailstor.savf
- Qdtssfl.savf
- SV000501.zip
- SV000514.zip
- SV000618.zip
- SV001585.zip
- SV001586.zip

Les fichiers .savf contiennent des données iSeries et des programmes RPG. Les fichiers .zip contiennent les applications Web qui interagissent avec les programmes iSeries pour manipuler les données iSeries. Dans un premier temps, vous devez restaurer les fichiers .savf sur votre serveur iSeries. Vous devez ensuite importer les fichiers .zip dans Development Studio Client et exécuter l'application dans l'atelier.

## Restauration des fichiers .savf

Pour utiliser les exemples de ce guide, vous devez restaurer les bibliothèques WHOLESale et RETAILSTOR sur votre serveur iSeries. Vous devez le faire même si vous avez déjà restauré les bibliothèques d'une version précédente du produit, car leur contenu est différent.

**Remarque :** Les fichiers .savf utilisés pour installer l'exemple de bibliothèque doivent être utilisés avec un serveur iSeries V5R1 ou ultérieur. Pour les besoins de ce scénario, les deux bibliothèques sont restaurées sur le même serveur iSeries. Si vous développez cette application pour une entreprise réelle, vous restaureriez les deux bibliothèques sur deux systèmes serveur iSeries différents. Vous restaureriez la bibliothèque WHOLESale sur le serveur iSeries qui fournit les services Web et la bibliothèque RETAILSTOR sur le serveur iSeries du magasin point de vente.

Pour restaurer le fichier Wholesale.savf, procédez comme suit :

1. Connectez-vous à votre serveur iSeries via un émulateur.
  - a. Créez une bibliothèque pour stocker les fichiers de sauvegarde. Pour créer une nouvelle bibliothèque dans l'émulateur, tapez CRTLIB.
  - b. Nommez votre bibliothèque SCENARIO.
  - c. Passez à la ligne suivante, indiquez \*TEST comme type de bibliothèque et appuyez sur Entrée pour enregistrer les modifications.
  - d. Créez deux fichiers de sauvegarde à l'aide de la commande CRTSAVF, en appuyant sur Entrée entre les deux lignes :

```
CRTSAVF FILE(SCENARIO/WHOLESale)
CRTSAVF FILE(SCENARIO/RETAILSTOR)
```

Ces lignes indiquent que vous souhaitez créer des fichiers de sauvegarde dans votre bibliothèque Scenario.
2. Sur le poste de travail, ouvrez une fenêtre d'invite.
  - a. Vous devez vous placer dans le répertoire contenant les fichiers .savf. Par défaut, vous devez entrer `cd c:\wdsc\wdscsampl`. Si vous avez installé le produit dans un répertoire différent ou si vous avez choisi de ne pas utiliser "wdsc" comme répertoire principal pour Development Studio Client, placez-vous dans le répertoire wdscsampl où vous avez installé le produit.
  - b. Sur la ligne de commande, entrez : `ftp nom_hôte`, où *nom\_hôte* représente le nom de votre serveur iSeries (PROD400, par exemple).
  - c. Entrez votre ID utilisateur et votre mot de passe pour le serveur iSeries.
  - d. Sur la ligne de commande, entrez `cd /qsys.lib/scenario.lib` pour vous placer dans votre bibliothèque Scenario.
  - e. Entrez les lignes suivantes :

```
bin
put WHOLESale.savf WHOLESale.savf
put RETAILSTOR.savf RETAILSTOR.savf
quit
```

Ces lignes indiquent que vous souhaitez déplacer les fichiers de sauvegarde de votre système local dans le serveur iSeries.

3. De retour dans la console iSeries, restaurez votre bibliothèque Wholesale :
  - a. Entrez RSTLIB et appuyez sur F4 pour définir le mode de restauration de la bibliothèque.
  - b. Dans la zone **Bibliothèque sauvegardée**, tapez WHOLESale et appuyez sur la touche de tabulation.
  - c. Dans la zone **Unité**, tapez \*savf et appuyez sur la touche de tabulation.
  - d. Dans la zone suivante, appuyez sur Entrée pour afficher les autres valeurs et appuyez sur la touche de tabulation pour passer à la zone **Fichier de sauvegarde**.
  - e. Dans la zone **Fichier de sauvegarde**, tapez WHOLESale et appuyez sur la touche de tabulation.
  - f. Dans la zone **Bibliothèque**, supprimez la valeur et tapez scenario.
  - g. Appuyez sur Entrée pour restaurer la bibliothèque WHOLESale sur votre serveur iSeries.
4. Répétez cette procédure pour le fichier Retailstor.savf :
  - a. Entrez RSTLIB et appuyez sur F4 pour définir le mode de restauration de la bibliothèque.
  - b. Dans la zone **Bibliothèque sauvegardée**, tapez RETAILSTOR et appuyez sur la touche de tabulation.
  - c. Dans la zone **Unité**, tapez \*savf et appuyez sur la touche de tabulation.
  - d. Dans la zone suivante, appuyez sur Entrée pour afficher les autres valeurs et appuyez sur la touche de tabulation pour passer à la zone **Fichier de sauvegarde**.
  - e. Dans la zone **Fichier de sauvegarde**, tapez RETAILSTOR et appuyez sur la touche de tabulation.
  - f. Dans la zone **Bibliothèque**, supprimez la valeur et tapez scenario.
  - g. Appuyez sur Entrée pour sauvegarder votre action et restaurer la bibliothèque RETAILSTOR sur votre serveur iSeries.

## Modification de la liste des bibliothèques

Même si vous avez restauré les bibliothèques RETAILSTOR et WHOLESale sur votre serveur iSeries, vous devez vérifier qu'elles figurent dans la liste des bibliothèques que vous utilisez lorsque vous vous connectez au serveur. Dans le cas contraire, vos applications Web risquent de ne pas trouver les bibliothèques et vous risquez de rencontrer des problèmes lors de l'exécution ou de la compilation des applications. Vous devez également vérifier que la bibliothèque QGPL figure dans la liste des bibliothèques.

Pour ce faire, vous devez mettre à jour la description de travail et votre profil utilisateur pour que les modifications apportées à votre liste des bibliothèques deviennent permanentes.

Les méthodes utilisées pour modifier les listes des bibliothèques dépendent des utilisateurs. Si vous ne savez pas quelle méthode utiliser, voici un exemple :

1. Sur votre serveur iSeries, tapez dsplibl et appuyez sur Entrée pour afficher la liste de vos bibliothèques.

2. Vérifiez que les bibliothèques RETAILSTOR, WHOLESale et QGPL figurent dans votre liste des bibliothèques. Si c'est le cas, vous pouvez fermer la fenêtre de l'émulateur. Si ce n'est pas le cas :
  - a. Appuyez sur F12 pour quitter l'écran.
  - b. Recherchez le nom de la description de travail. Pour ce faire, tapez `dspusrprf nom_utilisateur` et appuyez sur Entrée pour afficher votre profil utilisateur (*nom\_utilisateur* est l'ID que vous utilisez pour vous connecter au serveur iSeries). La description de travail figure sur l'une des pages de votre profil utilisateur (faites défiler l'écran vers le bas pour y accéder). Lorsque vous l'avez trouvée, notez son nom et appuyez sur F12 pour quitter l'écran.
  - c. Sur la ligne de commande, tapez `chgjobd` et appuyez sur F4.
  - d. Dans la zone **Description de travail**, tapez le nom de votre description de travail et appuyez sur F10 pour afficher les autres paramètres.
  - e. Descendez jusqu'à **Liste des bibliothèques initiale**.
  - f. Dans la zone située sous **Liste des bibliothèques initiale**, la mention "+ si autres valeurs" apparaît. Tapez le signe plus et appuyez sur Entrée.
  - g. Dans la page **Spécifiez d'autres valeurs**, supprimez les entrées éventuelles et tapez RETAILSTOR, WHOLESale et QGPL, en séparant chaque valeur par une tabulation. Vous devez taper les noms des bibliothèques dans cet ordre car les applications doivent d'abord trouver la bibliothèque RETAILSTOR. Appuyez sur Entrée.
  - h. Appuyez à nouveau sur Entrée pour enregistrer vos modifications.
  - i. Sur la ligne de commande, tapez `chgusrprf` et appuyez sur Entrée.
  - j. Tapez votre nom d'utilisateur et appuyez sur Entrée.
  - k. Appuyez sur F10 pour afficher les autres paramètres.
  - l. Descendez jusqu'à **Description du travail**.
  - m. Si le nom de votre description de travail ne correspond pas à la description de tâche que vous avez trouvée à l'aide de la commande `dspusrprf`, tapez le nom de la description de travail trouvée par cette commande et appuyez sur Entrée pour sauvegarder vos modifications.

---

## Importation et exécution de l'application dans Development Studio

Les fichiers du programme installé comprennent cinq fichiers zip pour Development Studio Client, qui se trouvent par défaut dans le répertoire `c:\wdsc\wdscsampl`. Ces fichiers sont les suivants :

- SV000514.zip
- SV000618.zip
- SV001585.zip
- SV001586.zip
- SV000501.zip

Vous devrez les importer un par un dans l'atelier Development Studio Client et spécifier un fichier EAR (Enterprise Archive) pour chacun. Les fichiers EAR sont des fichiers d'archive Java (JAR) standard portant une extension `.ear`. Ils peuvent contenir plusieurs projets Web. Vous pouvez utiliser ces fichiers pour regrouper et déployer vos propres applications Web sur le serveur WAS (WebSphere Administrative Server).

Les quatre premiers fichiers correspondent aux projets Web et le cinquième fichier correspond à un projet WebFacing. Vous devez d'abord créer les projets Web, puis le projet WebFacing. Ensuite, importez le contenu des fichiers zip dans les cinq projets.

## Extraction des fichiers .zip sur votre système local

Avant de pouvoir importer le contenu des fichiers .zip dans l'atelier Development Studio Client, vous devez les extraire dans des répertoires séparés, de façon à pouvoir les importer sous forme de systèmes de fichiers au lieu de fichiers .zip.

**Remarque :** Si vous utilisez un programme autre que WinZip, effectuez les tâches comme à votre habitude avec ce programme.

1. Sur votre système local, placez-vous dans c:\wdsc\wdscsampl. Le répertoire c:\wdsc peut être différent si vous avez choisi d'installer le programme sur un lecteur ou dans un répertoire de base différent. Si c'est le cas, placez-vous dans ce répertoire et recherchez \wdscsampl.
2. Cliquez deux fois sur le fichier SV000501.zip.
3. Cliquez sur le bouton Extraire et placez-vous dans le dossier où se trouvent les fichiers zip, par défaut, c:\wdsc\wdscsampl.
4. Sélectionnez le bouton d'option **Tous les fichiers** puis cliquez sur **Extraire**.
5. Si vous regardez dans votre répertoire wdscsampl, vous verrez qu'un dossier SV000501 est créé, comportant tous les fichiers nécessaires.
6. Maintenant que tous les fichiers sont extraits dans c:\scenario\SV000501, supprimez le fichier SV000501.zip pour éviter de le sélectionner accidentellement lorsque vous importerez votre système de fichiers.
7. Répétez la même procédure pour les quatre autres fichiers .zip (SV000618.zip, SV001585.zip, SV001586.zip et SV000514.zip).

## Création des projets Web

Dans cette section, vous allez créer les projets Web pour SV000514, SV000618, SV001585 et SV001586. Les procédures de création de chaque projet sont présentées individuellement. Même si ces procédures semblent répétitives, notez que les quelques différences mineures sont importantes lors de la création des quatre projets (notamment pour savoir quels fichiers EAR sont associés à quels projets).

Pour créer les projets Web :

1. Dans le menu Démarrer, ouvrez IBM WebSphere Development Studio Client for iSeries.
2. Dans Development Studio Client, ouvrez la perspective Web à partir de la barre de menus en cliquant sur **Fenêtre > Ouvrir une perspective > Autre > Web** et cliquez sur **OK**.
3. Cliquez sur **Fichier > Nouveau > Projet Web dynamique**.
  - a. Dans la zone **Nom du projet**, tapez SV000514.
  - b. Cochez la case **Configurer les options avancées** et cliquez sur **Suivant**.
  - c. Cliquez sur le bouton **Nouveau** en regard de la zone **Projet EAR**.
  - d. Dans la zone **Nom du projet**, tapez SVWholeSaleEAR et cliquez sur **Fin** dans les deux boîtes de dialogue.
4. Cliquez sur **Fichier > Nouveau > Projet Web dynamique**.
  - a. Dans la zone **Nom du projet**, entrez SV001586 et cliquez sur **Suivant** (la case **Configurer les options avancées** doit se cocher automatiquement).

- b. Vérifiez que SWholeSaleEAR est sélectionné dans la zone déroulante **Projet EAR**.
  - c. Cliquez sur **Fin**.
5. Cliquez sur **Fichier > Nouveau > Projet Web dynamique**.
  - a. Dans la zone **Nom du projet**, tapez SV000618 et cliquez sur **Suivant**.
  - b. Cliquez sur le bouton **Nouveau** en regard de la zone **Projet EAR**.
  - c. Dans la zone **Nouveau nom de projet**, tapez SVStoreEAR et cliquez sur **Fin** dans les deux boîtes de dialogue.
6. Cliquez sur **Fichier > Nouveau > Projet Web dynamique**.
  - a. Dans la zone **Nom du projet**, tapez SV001585 et cliquez sur **Suivant**.
  - b. Vérifiez que SVStoreEAR est sélectionné dans la zone déroulante **Projet EAR**.
  - c. Cliquez sur **Fin**.

## Importation de fichiers de projet Web

Pour importer les systèmes de fichiers dans votre espace de travail, exécutez la procédure suivante pour importer les fichiers SV000514. La procédure est identique pour les trois autres systèmes de fichiers du projet Web (SV001586, SV001585 et SV000618). La section qui suit celle-ci décrit comment créer et importer le projet WebFacing.

Pour importer les fichiers :

1. Cliquez avec le bouton droit de la souris sur le dossier **SV000514** et sélectionnez **Importer**.
2. Dans l'assistant **Importation**, cliquez sur **Système de fichiers** et sur **Suivant**.
3. Placez-vous dans le répertoire où vous avez extrait les fichiers de scénario ; par défaut, ce devrait être c:\wdsc\wdscsampl.
4. Cliquez sur **SV000514**, puis sur **OK**.
5. Cliquez sur **Sélectionner tout** pour sélectionner l'ensemble des composants du système de fichiers SV000501.
6. Dans la zone **Dans le dossier**, entrez **SV000514** si ce numéro n'apparaît pas par défaut.
7. Cochez la case **Remplacer les ressources existantes sans avertissement**.
8. Si des boîtes de dialogue vous demandent si vous souhaitez modifier la racine de contexte, cliquez sur **Fin** et sur **Oui**.
9. Répétez les étapes 2 à 9 pour les dossiers SV001586, SV001585 et SV000618 en veillant à utiliser les différents noms de projet à mesure que vous exécutez les étapes.

## Importation de fichiers de projet WebFacing et création du projet WebFacing

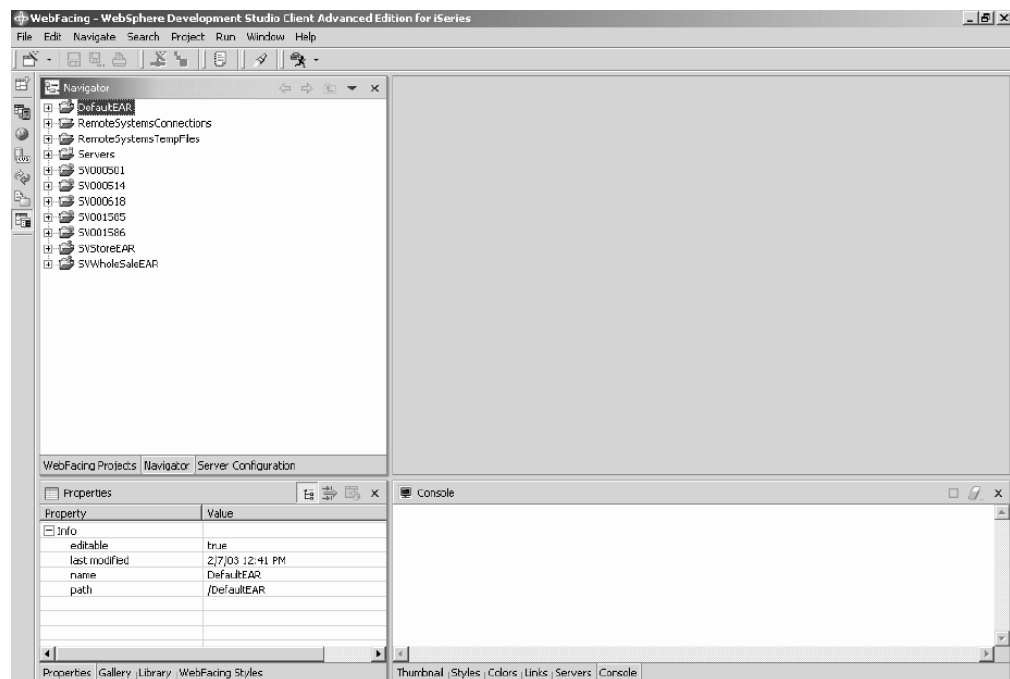
Le projet SV000501 est différent des quatre projets Web que vous venez de créer car SV000501 est un projet WebFacing. Les projets WebFacing sont différents car ils comprennent des fichiers différents et possèdent une structure unique, spécialement conçue pour le traitement de fichiers DDS sur votre serveur iSeries.

Pour importer les fichiers et créer un projet WebFacing :

1. Si vous ne vous trouvez pas dans la vue du navigateur, cliquez sur l'onglet **Navigateur de projet**.
2. Cliquez avec le bouton droit de la souris sur la vue et sélectionnez **Importer**.

3. Dans la fenêtre Importation, sélectionnez **Projets WebFacing** et cliquez sur **Suivant**.
4. Placez-vous dans le répertoire où vous avez extrait les fichiers de scénario ; par défaut, ce devrait être c:\wdsc\wdscsampl.
5. Sélectionnez **SV000501**, puis cliquez sur **OK**. Cliquez sur **Suivant**.
6. Dans la zone **Projets WebFacing trouvés** de la fenêtre, sélectionnez la case à cocher **SV000501**.
7. Dans la zone **Projet d'application d'entreprise (EAR)**, tapez **SVStoreEAR**.
8. Cliquez sur **Fin**.

Maintenant que vous avez importé tous les fichiers, votre espace de travail doit ressembler à l'écran suivant :



Si des erreurs sont signalées en regard des projets importés (représentées par des marques x rouges), cliquez avec le bouton droit de la souris sur chaque projet et sélectionnez **Reconstruire le projet**. Selon le contenu de votre espace de travail, cela devrait résoudre la plupart des erreurs et permettre à l'application de fonctionner correctement. Si des erreurs persistent, cela peut être dû au code, qui n'a pas entièrement migré vers la version 5.1 de Development Studio Client. Vous pouvez poursuivre les exercices de ce chapitre pour vous familiariser avec le produit, ou bien passer au chapitre suivant pour essayer de compiler les applications à partir de zéro.

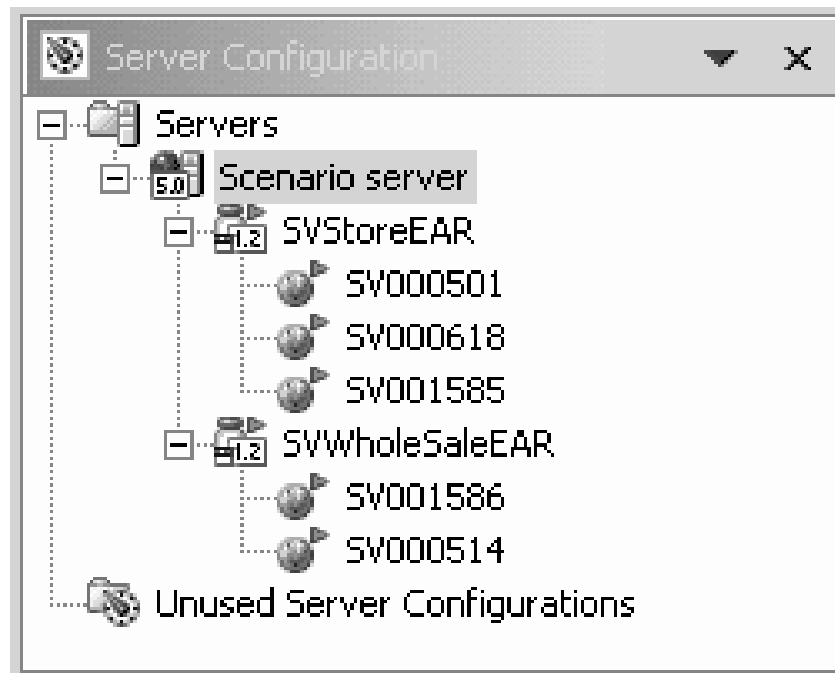
## Configuration du serveur

Maintenant que vous avez importé tous les fichiers source, vous devez configurer le serveur de l'environnement de test WebSphere pour reconnaître les applications SVWholeSaleEAR et SVStoreEAR.

Pour configurer le serveur de l'environnement de test WebSphere :

1. Ouvrez la perspective du serveur. Dans la barre de menus de l'atelier, cliquez sur **Fenêtre > Ouvrir une perspective > Autre > Serveur** et sur **OK**.

2. Dans la vue Configuration de serveur, cliquez avec le bouton droit de la souris sur **Serveurs** et sélectionnez **Nouveau > Serveur et configuration de serveur**.
3. Dans la zone **Nom de serveur**, tapez Scenario Server.
4. Dans la zone **Dossier**, tapez Scenario folder.
5. Dans la zone **Type de serveur**, développez **WebSphere version 5.0** et cliquez sur **Environnement de test**, si cette option n'est pas sélectionnée par défaut.
6. Cliquez sur **Fin** et sur **Oui** si vous recevez des messages.
7. Pour ajouter des fichiers EAR à la configuration du serveur :
  - a. Développez **Serveurs**.
  - b. Cliquez avec le bouton droit de la souris sur **Scenario Server** et sélectionnez **Ajouter > SVWholeSaleEAR**.
  - c. Développez **Serveurs**.
  - d. Cliquez de nouveau avec le bouton droit de la souris sur **Scenario server** et sélectionnez **Ajouter et supprimer des projets**.
  - e. Cliquez sur le bouton **Ajouter tout** pour déplacer SVStoreEAR et SVWholeSaleEAR sur la droite de la fenêtre.
  - f. Cliquez sur **Fin**.
8. Dans la vue Configuration du serveur, développez **Scenario Server**, **SVStoreEAR** et **SVWholeSaleEAR** pour afficher toutes les applications :



Pour vérifier que le serveur a sélectionné les projets :

1. Dans la vue Serveurs, à droite de la vue Configuration du serveur, cliquez avec le bouton droit de la souris sur **Scenario server** et sélectionnez **Publier**. Si la vue Serveurs n'apparaît pas, cliquez sur l'onglet **Serveurs**. La boîte de dialogue **Publication** indique la progression du processus de publication.
2. Cliquez sur OK pour vos applications dans l'environnement de test WebSphere Application Server.
3. Une fois la publication terminée, cliquez sur **OK** dans la boîte de dialogue.
4. Dans la même vue, cliquez à nouveau avec le bouton droit de la souris sur **Scenario Server** et cliquez sur **Démarrer**. La vue Console affiche les activités



du serveur. Le serveur est lancé lorsque la mention "Serveur *nom\_serveur* ouvert pour e-business" apparaît en bas de la vue. Faites défiler l'écran pour afficher l'ensemble des messages.

## Configuration du projet WebFacing

Avant d'exécuter l'application ou d'utiliser le module SV000501, vous devez démarrer le serveur WebFacing associé à l'application à exécuter et configurer le projet pour qu'il utilise le serveur iSeries approprié.

Pour lancer le serveur WebFacing :

1. Ouvrez un émulateur 5250, puis connectez-vous à l'aide de votre ID utilisateur et de votre mot de passe.
2. Sur la ligne de commande, tapez `strtcpsvr *webfacing`.
3. Le message "Démarrage du serveur WEBFACING" apparaît en bas de l'écran.

Vous devez maintenant modifier les propriétés WebFacing en revenant à l'atelier Development Studio Client :

1. Ouvrez la perspective Web (vous pouvez passer d'une perspective à l'autre en cliquant sur les icônes regroupées à gauche de l'écran).
2. Cliquez sur l'onglet **Navigateur de projet** pour afficher les structures des projets.
3. Développez **SV000501** et cliquez deux fois sur **Descripteur de déploiement Web**.
4. Cliquez sur l'onglet **Paramètres**.
5. Cliquez sur **WFDefaultHost**.
6. Dans la zone **Valeur**, supprimez la valeur existante et tapez le nom de votre serveur iSeries.
7. Dans la barre de menus de l'atelier, cliquez sur l'icône de sauvegarde ou sur **Fichier > Sauvegarder le descripteur de déploiement Web**. Si des messages apparaissent, cliquez sur **OK**. Fermez le fichier.

## Définition des informations relatives au serveur

Vous devez maintenant définir les informations du serveur iSeries pour l'ensemble des cinq projets, en vérifiant qu'ils sont configurés pour être exécutés sur votre serveur iSeries avec votre ID utilisateur et votre mot de passe. Pour définir les informations relatives au serveur :

1. Vérifiez que vous vous trouvez bien dans la perspective Web.
2. Dans la vue Navigateur de projet, cliquez avec le bouton droit de la souris sur **SV000501** et sélectionnez **Spécification de la configuration d'exécution des outils Web iSeries**.
3. Dans la zone **Nom de serveur iSeries**, supprimez la valeur existante et tapez le nom du serveur iSeries.
4. Supprimez les autres valeurs, puis tapez votre ID utilisateur et votre mot de passe.
5. Cliquez sur **Fin** (cliquez à nouveau sur **Fin**, si nécessaire).
6. Répétez les étapes 2 à 5 pour les autres projets (SV000514, SV000618, SV001585 et SV001586).

## Exécution de l'application dans l'atelier

A ce stade, vous pouvez exécuter l'application dans l'environnement de test WebSphere de l'atelier. Si vous êtes satisfait de son exécution, vous pouvez exporter le fichier EAR vers le serveur et le déployer sur le serveur iSeries WebSphere Application Server.

## Exécution de l'application en tant que client dans l'atelier

En tant que client, vous arrivez sur la page Web d'achat du magasin point de vente, vous parcourez les articles et vous commandez une quantité et une taille.

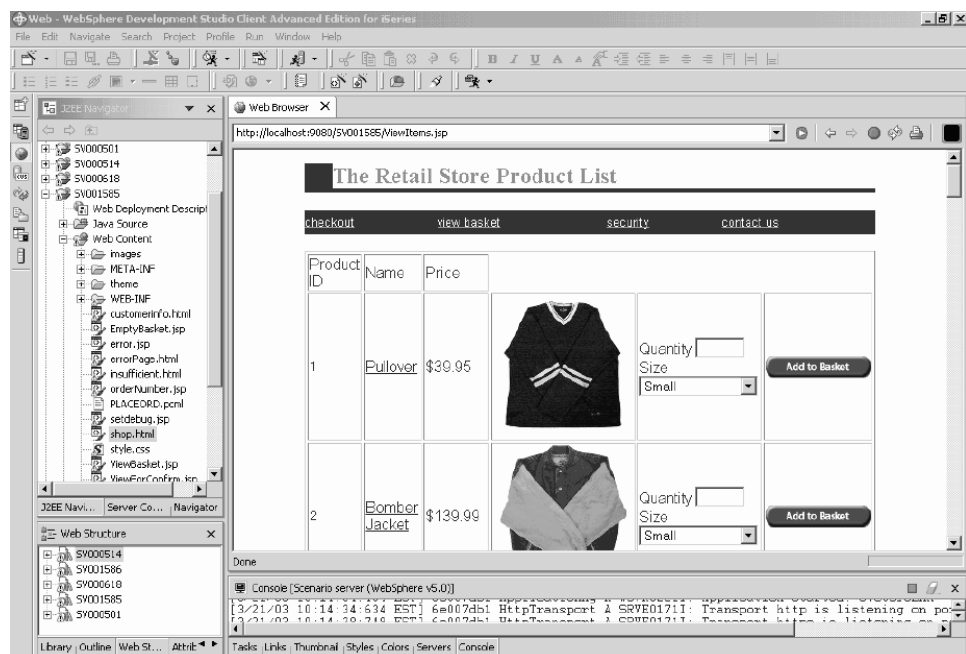
**Remarque :** Vous risquez de rencontrer des problèmes si vous tentez d'exécuter l'application derrière un pare-feu, car votre fichier web.xml recherche le fichier suivant : `http://java.sun.com/j2ee/dtds/web-app_2_2.dtd`. Pour résoudre ce problème, remplacez l'instruction DOCTYPE des fichiers web.xml par la valeur suivante avant d'exécuter l'application :

```
!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"x:/wdsc/eclipse/plugins/com.ibm.etools.j2ee/dtds/web-app_2_2.dtd"
```

Où `x:/wdsc` représente le répertoire dans lequel vous avez installé le produit. En général, les fichiers web.xml figurent dans *Nom de projet* > **Contenu Web** > **WEB-INF**. Une fois le fichier ouvert, cliquez sur l'onglet **Source** pour modifier directement le fichier.

Pour exécuter l'application en tant que client dans l'atelier :

1. Dans la perspective Web, développez **SV001585** > **Contenu Web**.
2. Cliquez avec le bouton droit de la souris sur **shop.html** et sélectionnez **Exécuter sur le serveur**. L'application est lancée dans le navigateur de l'atelier.
3. Cliquez sur **Fin**.
4. Ouvrez l'application en cliquant sur l'image des tee-shirts.
5. Essayez d'entrer des valeurs sur la page suivante. Dans cet exemple, vous allez commander un pull-over ou un blouson d'aviateur, choisir la taille et la quantité, et ajouter des articles à votre panier :

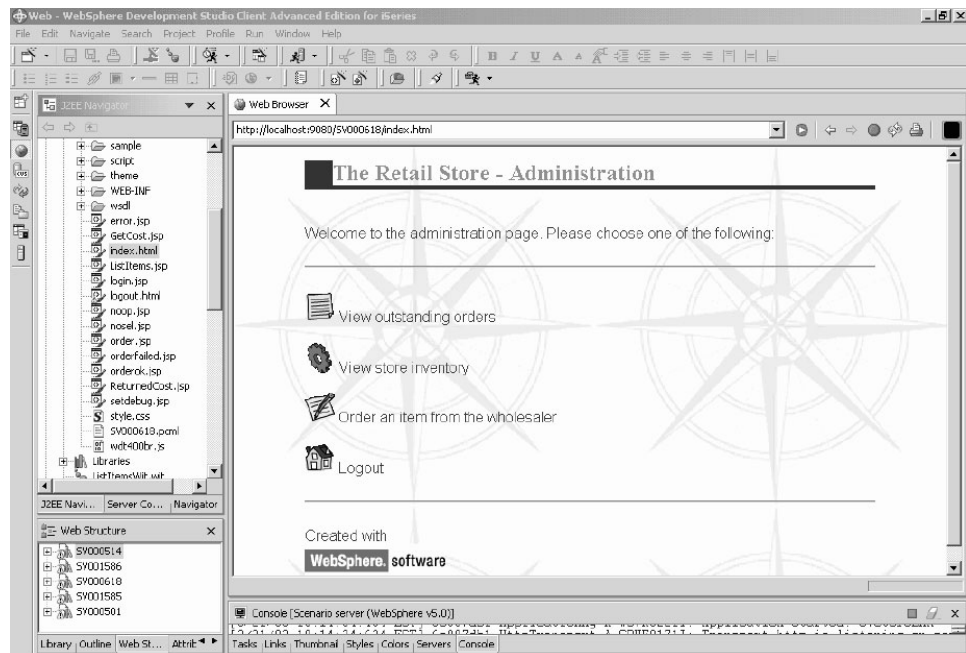


La page qui affiche le texte "your order has been processed" correspond à la dernière page de l'exemple. Dès que vous êtes satisfait de l'exemple, vous pouvez fermer le navigateur.

## Exécution de l'application en tant qu'administrateur dans l'atelier

En tant qu'administrateur, vous arrivez sur la page Web d'administration du magasin point de vente, vérifiez les commandes ou le stock. Pour exécuter l'application en tant qu'administrateur dans l'atelier :

1. Dans la perspective Web, développez **SV000618 > Contenu Web**.
2. Cliquez avec le bouton droit de la souris sur **index.html** et sélectionnez **Exécuter sur le serveur**. L'application est lancée dans le navigateur de l'atelier.
3. Cliquez sur l'icône à gauche de **Visualiser le stock** pour afficher la page suivante, comme si vous étiez un administrateur chargé des achats pour votre magasin :



Essayez de cliquer sur les icônes pour afficher les pages d'administration.

---

## Déploiement de l'application de scénario sur WebSphere Application Server

Maintenant que vous avez exécuté l'application de scénario iSeries dans l'environnement de test de l'atelier, vous pouvez déployer l'application sur WebSphere Application Server comme si vous exécutiez réellement l'application. Avant de déployer l'application, vous devez toutefois effectuer quelques ajustements, notamment sécuriser la page de l'administrateur et modifier l'URL d'un service Web, pour permettre à l'application de désigner la page appropriée (comme indiqué dans les sections suivantes).

**Remarque :** Le déploiement sur WebSphere Application Server est facultatif ; vous pouvez passer au chapitre suivant et terminer les modules sans tester l'application sur WebSphere Application Server.

## Configuration de WebSphere Application Server

Au début du chapitre, vous avez consulté la section «Exécution de l'application dans l'atelier» à la page 18. Maintenant que vous avez déployé les fichiers sur le serveur iSeries, vous pouvez exécuter l'application sur votre serveur iSeries à l'aide de WebSphere Application Server.

Pour déployer l'application sur WebSphere Application Server (facultatif), vous devez vérifier les points suivants :

- Vous avez créé ou avez accès à l'instance WebSphere Application Server version 4.0, version 5.0 ou WebSphere Application Server Express sur le serveur iSeries, et l'instance est en cours d'exécution (uniquement si vous souhaitez tester le déploiement sur WebSphere Application Server).
- Vous connaissez les numéros de port des instances HTTP et WebSphere Application Server sur le serveur iSeries.
- La console d'administration WebSphere version 4.0 ou 5.0 est installée sur votre poste de travail.

**Remarque :** Pour la version 5.0 de WAS, la console utilise un navigateur. Vous n'avez donc pas besoin d'installer une console sur votre poste de travail. Pour plus d'informations sur les différentes versions de WAS, consultez la documentation en ligne.

Vos applications iSeries compatibles Web utilisent WebSphere Application Server pour exécuter les servlets Java et les fichiers JSP (JavaServer Page), qui assurent la communication entre le navigateur Web de l'utilisateur et les données ou programmes iSeries. Pour prendre en charge vos pages HTML et vos fichiers JSP à partir du même système iSeries, un serveur HTTP doit également être installé sur ce système. Nous vous conseillons d'utiliser le serveur IBM HTTP motorisé par Apache. La documentation concernant l'utilisation de ce serveur est disponible sur le site : IBM HTTP Server for iSeries Documentation Center à l'adresse [http://publib.boulder.ibm.com/pubs/html/series\\_http/v5r1/index.htm](http://publib.boulder.ibm.com/pubs/html/series_http/v5r1/index.htm).

WebSphere Application Server exécute des pages JSP, des beans Java et des servlets Java générés pour différents processus. Les principales ressources de documentation pour IBM WebSphere Application Server for iSeries et la console d'administration IBM WebSphere pour iSeries sont disponibles sur les sites suivants :

- WebSphere Application Server IBM version 4.0 Advanced Edition pour iSeries à l'adresse <http://publib.boulder.ibm.com/was400/40/AE/french/docs/>
- WebSphere Application Server IBM version 4.0 Advanced Single Server Edition pour iSeries à l'adresse <http://publib.boulder.ibm.com/was400/40/AEs/french/docs/>

Nous vous recommandons vivement de vous familiariser avec la documentation IBM WebSphere Application Server, notamment les sections *Modules J2EE*, *Installation de WebSphere Application Server* et *Configuration de plusieurs instances du serveur d'administration WebSphere*. Vous devez au minimum effectuer les étapes du lien *Installation*.

Utilisez le plan du site pour trouver des informations sur l'installation, la configuration et les correctifs requis pour la console d'administration WebSphere.

## Sécurisation de la page de l'administrateur

Seules les personnes autorisées doivent pouvoir accéder à la page de l'administrateur, `index.html`. Cette page doit donc être correctement sécurisée. La sécurisation peut être effectuée par programmation dans la logique d'application Web ou via la fonction de sécurité de WebSphere. Dans ce scénario, nous avons utilisé la fonction de sécurité de WebSphere pour sécuriser la page. Vous noterez que nous avons utilisé WebSphere Application Server V4.0 Advanced Edition. Pour consulter les informations relatives à WebSphere Application Server, consultez les sites Web suivants :

- WebSphere Application Server IBM version 4.0 Advanced Edition pour iSeries à l'adresse <http://publib.boulder.ibm.com/was400/40/AE/french/docs/>
- WebSphere Application Server IBM version 4.0 Advanced Single Server Edition pour iSeries à l'adresse <http://publib.boulder.ibm.com/was400/40/AEs/french/docs/>
- Page RedBooks pour WebSphere : Recherchez des RedBooks pour WebSphere Application Server version 5.0 et WebSphere Application Server Express Edition version 5.0 à l'adresse <http://publib-b.boulder.ibm.com/redbooks.nsf/portals/WebSphere>

Si vous utilisez une version différente de WebSphere Application Server, consultez la documentation relative à cette version en matière de sécurisation de ressources Web.

Vous pouvez configurer la sécurité des ressources Web, notamment des pages Web et des servlets, dans Development Studio ou dans l'outil d'assemblage de l'application. Pour ce scénario, nous allons utiliser Development Studio.

Pour passer en revue la configuration et les propriétés de sécurité de cette application Web :

1. Dans Development Studio, ouvrez la perspective Web.
2. Dans la vue Navigateur, développez **SV000618 > Contenu Web > WEB-INF**.
3. Cliquez deux fois sur **web.xml** pour ouvrir la vue `web.xml`.
4. Cliquez sur l'onglet **Security**.
5. Pour sécuriser la page d'administration (`index.html`), une contrainte de sécurité a également été définie. Dans la partie supérieure de la vue, cliquez sur **Security Constraints**.
6. Dans la liste, cliquez sur la première instance de **SecurityConstraint**.
7. A droite, cliquez sur **AdminPage**.
8. Cliquez sur **Edit** pour appeler la boîte de dialogue **Web Resource Collections**. Les méthodes GET et POST du fichier `index.html` sont déjà sélectionnées.
9. Cliquez sur **OK**.

Dans la section **Security roles**, le rôle de sécurité défini est "Administrator". Au cours du déploiement, des personnes sont chargées de ce rôle et disposent donc d'un accès à la page `index.html`. Dans la section **Authorized roles**, nous avons attribué le rôle d'accès Administrator à cette contrainte de sécurité. Une fois la sécurité activée, seuls les utilisateurs disposant du rôle Administrator ont accès à la page `index.html` à condition d'avoir fourni les informations d'identification appropriées (ID utilisateur et mot de passe). Lorsqu'une ressource est sécurisée, WebSphere Application Server tente d'abord d'authentifier l'utilisateur. Le système effectue l'authentification à l'aide de certificats ou demande à l'utilisateur

d'indiquer un ID utilisateur et un mot de passe. L'invite peut être effectuée via une boîte de dialogue d'authentification de base ou un formulaire personnalisé.

Dans ce scénario, nous avons conçu notre propre page de connexion appelée login.jsp. Pour configurer son invite d'authentification, sélectionnez l'onglet **Pages** dans la vue web.xml. Dans la section **Login, Form** est déjà présélectionné comme méthode d'authentification. Vous noterez également que la page de connexion est login.jsp. La page d'erreur s'affiche en cas d'échec de la demande de connexion. Dans ce cas, l'application affiche à nouveau la page login.jsp.

## Création de fichiers EAR pour le déploiement iSeries WebSphere Application Server

Vous devez créer des fichiers EAR pour déployer votre application sur iSeries WebSphere Application Server. Les fichiers EAR sont des fichiers d'archive Java (JAR) standard portant une extension .ear. Ils peuvent contenir plusieurs projets Web. Vous pouvez utiliser ces fichiers pour regrouper et déployer vos propres applications Web sur le serveur WAS (WebSphere Administrative Server).

Pour créer les fichiers EAR :

1. Ouvrez la perspective Web.
2. Dans la vue Navigateur, cliquez avec le bouton droit de la souris sur **SVStoreEAR** et sélectionnez **Exporter**.
3. Dans la fenêtre Exportation, cliquez sur **Fichier EAR** et cliquez sur **Suivant**.
4. Sous **Où souhaitez vous exporter les ressources ?**, cliquez sur **Parcourir** et naviguez jusqu'à un répertoire de votre Système de fichiers intégré iSeries sur lequel vous pouvez stocker les fichiers EAR. Vous devez avoir mappé votre unité réseau sur un système IFS iSeries.
5. Dans la zone **Nom de fichier**, entrez **SVRetailStorEAR.ear** et cliquez sur **Ouvrir**.
6. Cliquez sur **Fin**.
7. Répétez les étapes 2 à 6 pour **SVWholeSaleEAR**.

## Déploiement des fichiers EAR sur iSeries WebSphere Application Server

Maintenant que vous avez créé les fichiers EAR, vous pouvez les déployer sur WebSphere Application Server.

1. Ouvrez la console d'administration WebSphere.
2. Cliquez avec le bouton droit de la souris sur **Applications d'entreprise** et sélectionnez **Installation de l'application d'entreprise**.
3. Sélectionnez le bouton d'option **Installer l'application (\*ear)**.
4. Cliquez sur le bouton **Parcourir** supérieur (le bouton inférieur n'est pas disponible).
5. Naviguez jusqu'au répertoire IFS où vous avez exporté les fichiers EAR.
6. Sélectionnez **SVRetailStorEAR.ear**.
7. Dans la zone **Nom de l'application**, entrez **"RETAILSTOR"**.
8. Cliquez sur **Suivant** et entrez votre ID utilisateur de serveur iSeries du rôle d'administrateur.
9. Cliquez plusieurs fois sur **Suivant** jusqu'à ce que la page **Sélection des hôtes virtuels pour les modules Web** apparaisse.

10. Pour les trois modules Web, cliquez sur **Sélection de l'hôte virtuel** et sélectionnez le serveur virtuel de votre choix dans la boîte à liste déroulante. Si vous ne savez pas quel système hôte sélectionner, choisissez **default** ou **default\_host**.
11. Cliquez sur **Suivant**.
12. Pour les trois modules Web, cliquez sur **Sélectionner un serveur** et sélectionnez le serveur que vous souhaitez utiliser. Si vous ne savez pas quel serveur sélectionner, choisissez **Serveur par défaut**.
13. Cliquez sur **Suivant**.
14. Cliquez sur **Fin** et sur **OK** dans la boîte de dialogue.
15. Cliquez avec le bouton droit de la souris sur **Applications d'entreprise** et sélectionnez **Installation de l'application d'entreprise**.
16. Cliquez sur le bouton **Parcourir** inférieur (le bouton supérieur n'est pas disponible).
17. Naviguez jusqu'au répertoire IFS où vous avez placé les fichiers EAR.
18. Sélectionnez **SVWholesaleEAR.ear**.
19. Dans la zone **Nom de l'application**, entrez "WHOLESALE".
20. Cliquez plusieurs fois sur **Suivant** jusqu'à ce que **Fin** soit activé.
21. Cliquez sur **Fin** et sur **OK** dans la boîte de dialogue.

**Remarque :** Un message de confirmation apparaît au bout de quelques minutes.

## Exécution de l'application sur WebSphere Application Server

Pour prendre le rôle du client et lancer le point d'entrée du magasin point de vente, entrez l'URL suivante dans un navigateur Web :

```
http://nom_hôte_iSeries:numéro_port_HTTP/SV001585/shop.html
```

Pour prendre le rôle de l'administrateur et lancer le point d'entrée WholeSale, entrez l'URL suivante dans un navigateur Web :

```
http://nom_hôte_iSeries:numéro_port_HTTP/SV000618/index.html
```

Si vous ne connaissez pas le numéro de port HTTP, contactez votre administrateur WebSphere Application Server.





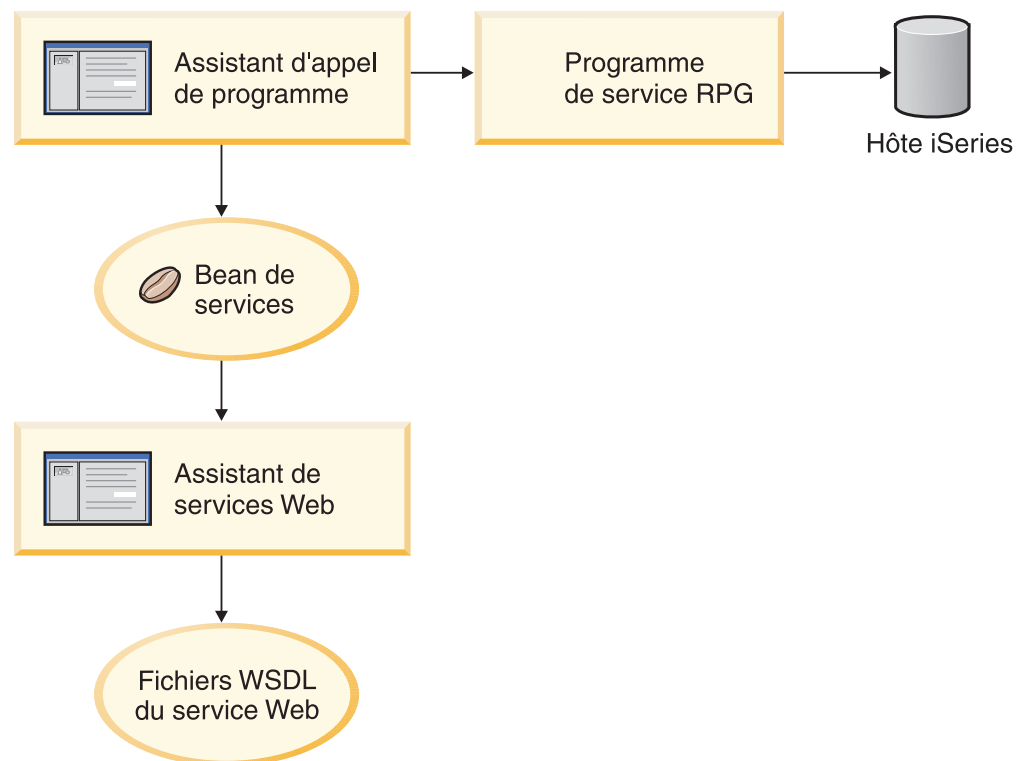
---

## Chapitre 4. Module d'initiation 1 : Création d'un service Web pour afficher le prix des produits (SV000514)

---

### Introduction

Dans ce module, vous allez créer un service Web à partir d'un programme RPG résidant sur le serveur iSeries pour afficher le prix des produits de votre stock. Vous devez commencer par créer un programme de service RPG comprenant une procédure, qui permet d'extraire dans la base de données iSeries le prix d'un article en fonction de sa référence. Utilisez l'assistant d'appel de programme des Outils de développement Java iSeries pour appeler le programme RPG et créer un bean de services. Vous pouvez ensuite utiliser l'assistant de services Web pour créer un service Web et le vérifier à l'aide de l'exemple généré.



---

### Avant de commencer

Vous ne pourrez effectuer les exercices que si les conditions ci-dessous sont remplies. Ces conditions prérequis sont détaillées dans le Chapitre 3, «Exécution du scénario», à la page 9.

- Vous disposez d'un accès TCP/IP à un serveur iSeries.
- Vous avez installé les derniers correctifs pour le serveur iSeries. Consultez notre page de support pour obtenir des informations sur les correctifs et les modules de mise à jour : <http://www.ibm.com/software/awdtools/wdt400/support/>
- Vous avez démarré les serveurs iSeries à l'aide de la commande STRTCPSVR \*ALL

- Vous avez démarré le serveur WebFacing à l'aide de la commande STRTCPSVR \*WEBFACING.
- Vous avez effectué toutes les tâches du Chapitre 3, «Exécution du scénario», à la page 9 (à l'exception des tâches facultatives WebSphere Application Server, qui ne sont pas obligatoires pour tester l'application dans l'atelier).

---

## Création d'un nouveau projet Web

Pour créer ce service Web, la première étape consiste à créer un nouveau projet Web pour stocker vos informations.

1. Dans l'atelier IDE, ouvrez la perspective Web en cliquant sur **Fenêtre > Ouvrir une perspective > Autre > Web > OK**.
2. Cliquez sur **Fichier > Nouveau > Projet Web dynamique**.
3. Dans la zone **Nom du projet**, tapez Project514.
4. Cochez la case **Configurer les options avancées** et cliquez sur **Suivant**.
5. Cliquez sur le bouton **Nouveau** en regard de la zone **Projet EAR**.
6. Dans la zone **Nom du projet**, tapez Project514EAR et cliquez sur **Fin** dans les deux boîtes de dialogue.

Les projets Project514 et Project514EAR ont été ajoutés à votre espace de travail dans la vue Navigateur de projet.

---

## Définition des informations relatives au serveur iSeries

Une fois le projet Web créé, vous devez définir le serveur iSeries utilisé par le projet pour obtenir des informations.

1. Cliquez avec le bouton droit de la souris sur le projet **Project514** et sélectionnez **Spécification de la configuration d'exécution des outils Web iSeries**.
2. Indiquez le nom du serveur iSeries hébergeant la bibliothèque WHOLESale restaurée (PROD400, par exemple).
3. Entrez votre ID utilisateur et votre mot de passe pour le serveur iSeries.
4. Tapez Wholesale dans la zone **Bibliothèque** et cliquez sur **Ajouter**.
5. Cliquez sur **Fin**.

---

## Création du programme de service RPG

Vous souhaitez que votre application récupère le prix d'un article selon sa référence. Cette opération est effectuée par un programme de service RPG contenant une procédure appelée QryProdCost. La bibliothèque WHOLESale contient un programme de service RPG appelé CWWSSRV. Ce programme comprend une procédure QryProdCost, qui peut accepter les références d'article en entrée, ouvrir le fichier de stock à partir de la bibliothèque WHOLESale, récupérer le prix de la base de données du stock et renvoyer le prix. Pour effectuer cette tâche, l'interface comprend deux paramètres, un pour la référence et l'autre pour le prix de l'article. Si la référence ou le prix est introuvable, le programme RPG renvoie un message à l'interface.

Pour créer ce service Web, vous devez d'abord créer un bean Java à l'aide de l'assistant d'appel de programme pour appeler la procédure RPG QryProdCost. Utilisez ensuite les services Web pour activer la procédure RPG en tant que service Web via le bean Java.

Pour créer le bean Java :

1. Dans la perspective Web, cliquez avec le bouton droit de la souris sur **Project514** et sélectionnez **Nouveau > Autre**.
2. Sélectionnez **iSeries > Java** dans la partie gauche de la fenêtre, puis **Bean d'appel de programme** dans la partie droite.
3. Cliquez sur **Suivant** pour appeler l'assistant d'appel de programme.
4. Dans la zone **Nom de bean Java** sous le titre **Ajouter un programme**, tapez **Inventory**.
5. Dans la zone **Objet de programme**, tapez le nom du programme de service **RPG CWWSSRV**.
6. Dans la zone **Bibliothèque**, tapez **WHOLESALE**.
7. Dans la boîte à liste déroulante **Type de programme**, sélectionnez **\*SRVPGM**.
8. Dans la zone **Point d'entrée**, tapez **QryProdCost**.
9. Cliquez sur **OK** pour ajouter la définition de programme.

## Création des paramètres et génération du bean Java

Une fois le programme spécifié, vous pouvez ajouter des paramètres. Le programme **RPG CWWSSRV** comprend les deux paramètres suivants :

- Paramètre de référence de l'article - Le programme utilise ce nombre pour rechercher l'article dans la base de données.
- Paramètre de prix de l'article - Le programme utilise ce nombre pour rechercher le prix de l'article dans la base de données.

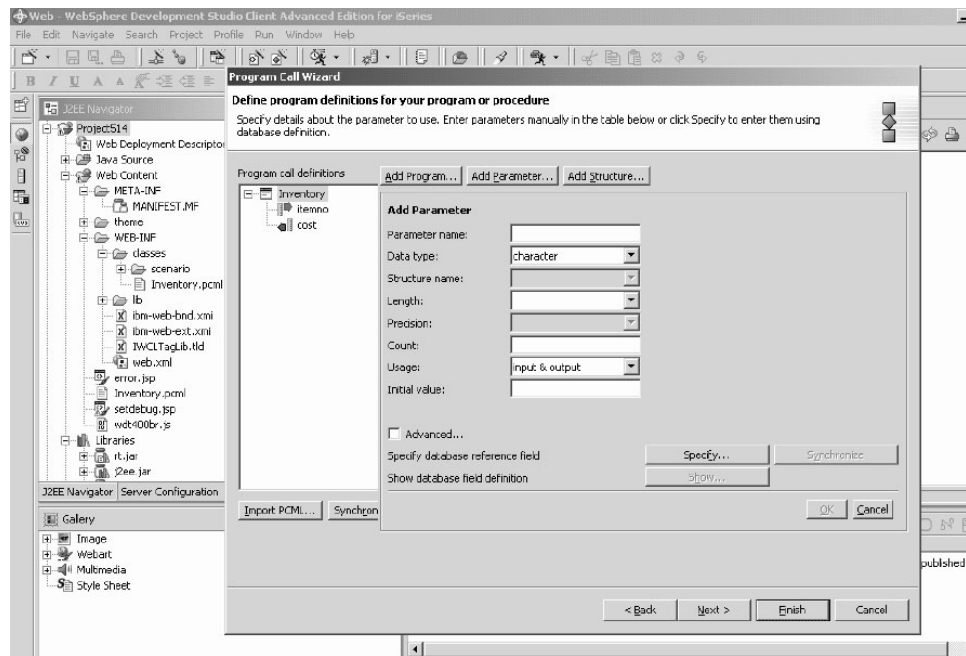
Pour ajouter le paramètre de référence de l'article :

1. Dans le panneau de gauche de l'assistant d'appel de programme, cliquez sur la définition d'appel de programme **Inventory** pour la sélectionner. Les zones à droite de la page sont remplies.
2. Cliquez sur **Ajouter un paramètre**.
3. Dans la zone **Nom de paramètre**, tapez **itemno**.
4. Dans la boîte à liste déroulante **Type de données**, sélectionnez **Décimal condensé**.
5. Dans la zone **Longueur**, tapez **5**.
6. Dans la zone **Précision**, tapez **0**.
7. Dans la boîte à liste déroulante **Usage**, sélectionnez **entrée**.
8. Cliquez sur **OK** pour ajouter ce paramètre. **itemno** apparaît sous **Inventory** dans le panneau de gauche. Vous pouvez maintenant ajouter le second paramètre.

Pour entrer le paramètre de prix de l'article :

1. Dans le panneau de gauche de l'assistant d'appel de programme, cliquez de nouveau sur la définition d'appel de programme **Inventory** pour la sélectionner. Les zones à droite de la page sont remplies.
2. Cliquez sur **Ajouter un paramètre**.
3. Dans la zone **Nom de paramètre**, tapez **cost**.
4. Dans la boîte à liste déroulante **Type de données**, sélectionnez **Décimal condensé**.
5. Dans la zone **Longueur**, tapez **7**.
6. Dans la zone **Précision**, tapez **2**.
7. Dans la boîte à liste déroulante **Usage**, sélectionnez **sortie**.

8. Cliquez sur **OK** pour ajouter ce paramètre. **cost** apparaît sous **Inventory** dans le panneau de gauche. A ce stade, l'assistant doit ressembler à la figure suivante. L'icône à gauche des paramètres peut être de type entrée, entrée et sortie, ou sortie :



Pour définir le nom de package de ces paramètres dans l'assistant d'appel de programme :

1. Cliquez sur **Suivant**.
2. Dans la zone **Package**, indiquez **scenario** comme nom de package. Conservez les valeurs par défaut dans les autres zones.
3. Décochez la case **Application Java**.

**Remarque :** Dans la liste des fichiers située sous la mention "Ces fichiers seront générés par l'assistant", le nom du bean Java généré est **InventoryServices.java**.

4. Cliquez sur **Fin** pour générer les fichiers.

## Création d'un service Web à partir du bean Java

Une fois que vous avez créé le bean Java qui appelle le programme RPG, vous devez convertir le bean en service Web pour permettre aux autres programmes d'accéder à ce même programme RPG via Internet.

Lors de la création du service Web, un assistant de services Web génère les fichiers WSDL qui sont distribués aux utilisateurs devant utiliser le service Web. Pour créer le service :

1. Dans la vue Navigateur de la perspective Web, développez **Project514 > Source Java > scenario**.
2. Cliquez avec le bouton droit de la souris sur **InventoryServices.java** et sélectionnez **Nouveau > Autre**.
3. Dans la fenêtre Nouveau, cliquez sur **Services Web**, sur **Service Web** dans la partie droite de la fenêtre, puis sur **Suivant**.
4. Cochez les cases **Générer un proxy** et **Tester le proxy généré**.

5. Cliquez sur **Suivant** trois fois de suite et notez au passage la génération des fichiers nécessaires. Lorsque la fenêtre d'identité du bean Java de service Web apparaît, cliquez sur **Fin**. Vous devrez probablement cliquer une seconde fois sur **Fin** si vous constatez des erreurs. Si vous ne pouvez pas mettre fin à l'assistant, cliquez simplement sur **Annuler** ; la plupart des fichiers nécessaires seront créés de toute façon.

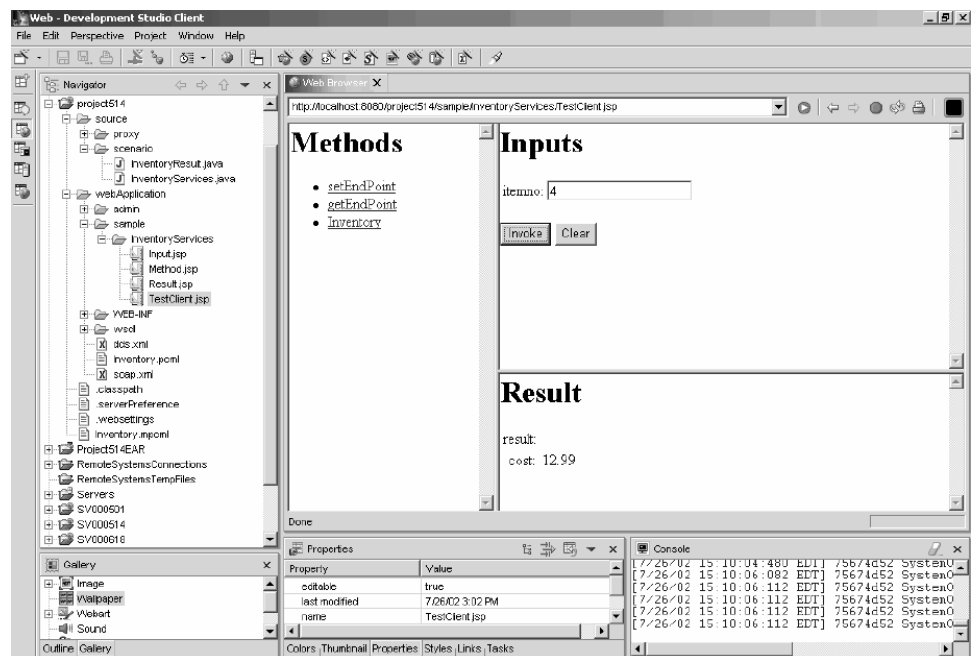
## Test du modèle

Lorsque vous avez créé le service Web, on vous a demandé de générer un modèle. L'assistant de services Web a donc créé un nouveau projet contenant des pages de test, que vous pouvez utiliser pour tester le service Web. Dans la vue **Navigateur**, ce nouveau nom de projet est **Project514Client**. Vous noterez également que l'outil a automatiquement ouvert le fichier **TestClient.jsp** dans la partie droite de l'atelier. Ce fichier figure sous **Project514Client > Contenu Web > sample > InventoryServices**.

Pour tester le modèle :

1. Dans la page chargée, faites défiler vers le bas le panneau des méthodes. Cliquez sur la méthode **Inventory (java.math.BigDecimal)**.
2. Dans la zone **itemno**, tapez 4, puis cliquez sur **Appeler**.
3. Vérifiez que le résultat renvoyé est :

result:  
cost: 12.99



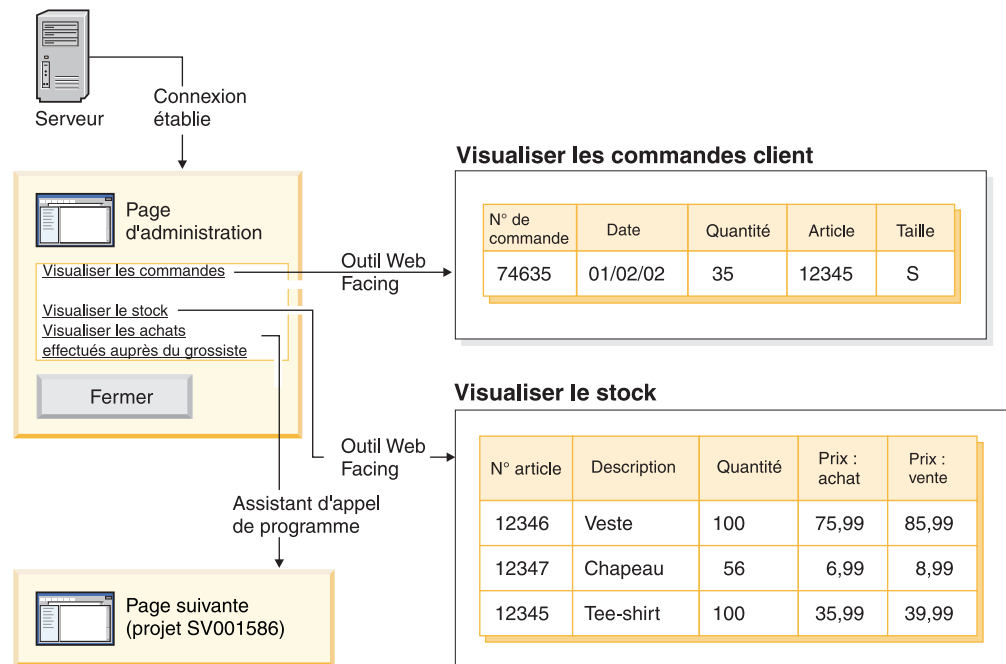
Une fois le résultat confirmé, vous avez terminé le module ! Vous venez de créer un service Web qui affiche le prix des produits dans l'interface d'application.



# Chapitre 5. Module d'initiation 2 : Création d'une interface pour afficher le stock et les articles commandés (SV000501)

## Introduction

Dans ce module, vous allez créer une interface qui affiche les options d'opération d'un administrateur à l'aide de l'Outil WebFacing. Une fois que vous êtes connecté au serveur iSeries, l'interface affiche le stock disponible et vous pouvez vérifier les commandes existantes. En tant qu'administrateur, vous pouvez acheter des marchandises auprès du grossiste.



Dans ce projet, vous utiliserez deux programmes et deux fichiers écran restaurés sur votre serveur iSeries. Les noms de programme sont ViewInventory et ViewOrder. Ils utilisent les deux fichiers écran suivants : ORDERDSP et QUERY. Ces deux fichiers contiennent des paramètres Web personnalisés pour les fichiers JSP générés par l'Outil WebFacing et sont utilisés pour les images et les liens hypertexte. Le paramètre Web d'image permet d'utiliser le contenu d'une zone pour générer le nom du fichier image et afficher l'image dans le fichier JSP. Le paramètre Web de lien hypertexte permet d'appeler une autre application lorsque vous cliquez sur l'image dans le fichier JSP. Vous pouvez utiliser CODE Editor ou CODE Designer pour vérifier le code source du fichier écran en vue de déterminer les paramètres Web.

---

## Avant de commencer

Vous ne pourrez effectuer les exercices que si les conditions ci-dessous sont remplies. Ces conditions prérequis sont détaillées dans le Chapitre 3, «Exécution du scénario», à la page 9.

- Vous disposez d'un accès TCP/IP à un serveur iSeries.
- Vous avez démarré les serveurs iSeries à l'aide de la commande STRTCPSVR \*ALL
- Vous avez démarré le serveur WebFacing à l'aide de la commande STRTCPSVR \*WEBFACING.
- Vous avez effectué toutes les tâches du Chapitre 3, «Exécution du scénario», à la page 9 (à l'exception des tâches facultatives WebSphere Application Server, qui ne sont pas obligatoires pour tester l'application dans l'atelier).

---

## Création d'un projet WebFacing

La première étape consiste à créer un projet WebFacing et à définir les commandes CL appropriées. Pour créer le projet WebFacing :

1. Dans l'atelier, ouvrez la perspective WebFacing en cliquant sur une icône de perspective à gauche de l'espace de travail ou sur **Fenêtre > Ouvrir une perspective > Autre > WebFacing** et sur **OK**.
2. Créez un nouveau projet WebFacing en cliquant sur **Fichier > Nouveau > Projet WebFacing**.
3. Nommez le projet Project501.
4. Cliquez sur le bouton d'option **Existant** en regard de **Projet d'application d'entreprise**.
5. Entrez SVStoreEAR dans la zone **Nom de projet existant** (cette valeur respecte les majuscules/minuscules). Cliquez sur **Suivant**.
6. Dans la boîte à liste déroulante **Niveau J2EE**, sélectionnez **1.3**. Cliquez sur **Suivant**.
7. Dans la zone **Connexion**, le nom de votre serveur iSeries devrait apparaître automatiquement. Dans le cas contraire, cliquez sur **Nouveau** et tapez le nom du serveur iSeries dans la boîte de dialogue, puis cliquez sur **Fin**.
8. Cliquez sur **Régénérer la liste DDS**, puis entrez votre mot de passe et votre ID utilisateur dans la boîte de dialogue pour régénérer la liste.
9. Dans la liste générée de bibliothèques, développez **RETAILSTOR**.
10. Cliquez sur **QDDSSRC**, puis cliquez sur la flèche droite (>>) pour déplacer les fichiers.
11. Cliquez deux fois sur **Suivant** jusqu'à ce que la page **Entrée de commandes CL** apparaisse.

Vous devez maintenant ajouter une commande et un intitulé de commande pour l'administrateur :

12. Dans la zone **Commande CL**, entrez call viewinvent.
13. Dans la zone **Intitulé de la commande**, supprimez toutes les valeurs et entrez Visualiser le stock.
14. Sélectionnez le bouton d'option **Ouverture de session avec valeurs spécifiées**.
15. Cliquez sur **Ajouter**. La partie inférieure de la fenêtre indique que l'élément a été ajouté à la liste.

Vous devez maintenant ajouter une deuxième commande et un deuxième intitulé de commande pour le client :

16. Dans la zone **Commande CL**, supprimez la valeur et tapez call vieworder.



17. Dans la zone **Intitulé de la commande**, supprimez la valeur et entrez Visualiser les commandes.
18. Sélectionnez le bouton d'option **Invite pour ouverture de session**.
19. Cliquez sur **Ajouter**. La partie inférieure de la fenêtre indique que l'élément a été ajouté à la liste. Cliquez sur **Suivant**.
20. Dans la fenêtre **Sélection d'un style Web**, parcourez la liste des styles disponibles. Comme nous devons récupérer ultérieurement les informations de ce style, sélectionnez **avenue** pour ce module. Cliquez sur **Suivant**.
21. Vérifiez que la case **Non. Je veux simplement créer le projet pour l'instant** est cochée et cliquez sur **Fin**.
22. Si un message demande si vous souhaitez réparer la configuration du serveur, cliquez sur **OK**. Ce message sert à confirmer que vous souhaitez ajouter le projet au fichier SVStoreEAR.

Dans la mesure où vous construisez Project501 comme miroir du projet SV000501, vous devez maintenant copier certains des fichiers image de SV000501 dans Project501 pour afficher Project501 correctement. Si nécessaire, vous pouvez toutefois ajouter vos propres images à l'interface. Pour copier les images :

1. Ouvrez la perspective Web.
2. Dans la vue Navigateur, développez **SV000501 > Contenu Web > images**.
3. (Facultatif) Développez **generated** pour noter toutes les images que vous importez dans le projet.
4. Cliquez avec le bouton droit de la souris sur **generated** et sélectionnez **Copier**.
5. Faites défiler la vue Navigateur jusqu'à ce que **Project501** apparaisse à nouveau.
6. Développez **Project501 > Contenu Web**.
7. Cliquez avec le bouton droit de la souris sur le dossier **images** et sélectionnez **Coller** dans le menu en incrustation.
8. Le dossier **generated** existe pour le Project501. Développez **generated** pour noter les images que vous avez ajoutées.

---

## Conversion de la source DDS

Maintenant que vous avez créé le projet, vous pouvez convertir les fichiers écran DDS en fichiers JSP pour votre page Web. Lorsque vous convertissez vos fichiers écran DDS, l'outil WebFacing génère des fichiers JSP et XML pour remplacer le code DDS et fournir un accès au Web. Les fichiers générés contiennent les données du format d'enregistrement ou contrôlent sa présentation, affichent la version Web de l'écran, demandent des données et traitent les erreurs d'entrée. L'assistant génère également une page d'accueil pour l'application, qui permet de lancer la version Web de votre programme.

Toutefois, vous devez modifier votre source pour désigner le serveur iSeries approprié. Plus précisément, vous devez apporter une modification à l'un de vos fichiers écran DDS pour que le lien vers l'application fonctionne. Vous devez remplacer le nom par défaut, SV000501, par votre nom de projet, Project501 :

1. Revenez dans la perspective WebFacing et ouvrez la vue Projets WebFacing si elle n'est pas déjà ouverte par défaut.
2. Développez **Project501 > DDS**.
3. Cliquez deux fois sur la seconde entrée,  
`<serveur_iSeries>RETAILSTOR/QDDSSRC(QUERY)`

4. Faites défiler l'écran jusqu'aux environs de la ligne 16, lorsque le message suivant apparaît :  
A\*%WB 12 FLD 1 next ('/SV000501/DetailPage.do?PRODNO=&{PRODNO}')
5. Supprimez SV000501 et entrez Project501.
6. Enregistrez et fermez le fichier.

**Remarque :** Si vous devez exécuter à nouveau le projet SV000501, vous devez remplacer la valeur Project501 par SV000501 car ce fichier réside sur le serveur iSeries et est accessible par les deux projets.

Pour convertir la source DDS :

1. Dans la perspective WebFacing, cliquez sur l'onglet **Projets WebFacing** pour ouvrir la vue correspondante si elle n'est pas déjà ouverte par défaut.
2. Développez **Project501 > DDS**.
3. Sélectionnez *serveur\_iSeries > RETAILSTOR/QDDSSRC(ORDERDSP)* et *serveur\_iSeries > RETAILSTOR/QDDSSRC(QUERY)* en cliquant sur le premier, puis en maintenant la touche Maj enfoncée tout en cliquant sur le second.
4. Cliquez avec le bouton droit de la souris et sélectionnez **Convertir** pour lancer la conversion. Vous devrez probablement entrer votre ID utilisateur et votre mot de passe si vous êtes déconnecté de votre serveur iSeries.

## Configuration UTF-8 dans l'atelier — Pour les utilisateurs WAS version 4.0 uniquement

Les applications de l'Outil WebFacing prennent en charge l'affichage de plusieurs langues à l'écran. Etant donné que les langues utilisent des jeux de caractères différents, les flux de données entre le navigateur et WebSphere Application Server sont codés en UTF-8. Pour que l'Outil WebFacing fonctionne correctement, vous devez configurer la prise en charge UTF-8 dans le fichier de propriétés de l'application dans l'atelier.

**Remarque :** La présente section concerne uniquement les utilisateurs WAS 4.0. WAS version 5.0 effectue cette tâche automatiquement.

Pour configurer la prise en charge UTF-8 :

1. Ouvrez la perspective Web (vous pouvez passer d'une perspective à l'autre en cliquant sur les icônes regroupées à gauche de l'écran).
2. Cliquez sur l'onglet **Navigateur de projet** pour afficher les structures des projets.
3. Développez **Project501** et cliquez deux fois sur **Descripteur de déploiement Web**.
4. Cliquez sur l'onglet **Environnement**.
5. Cliquez sur le bouton **Ajouter**.
6. Ouvrez la perspective Serveur. Pour ce faire, cliquez sur son icône dans le coin supérieur gauche de la barre des tâches ou sur **Perspective > Ouvrir > Autre > Serveur** dans la barre de menus, puis cliquez sur **OK**.
7. Dans la vue Navigateur, développez le dossier **Serveurs**.
8. Cliquez deux fois sur **defaultInstance.wsi** pour l'ouvrir dans l'éditeur par défaut.
9. Cliquez sur l'onglet **Environnement**, puis sur le bouton **Ajouter**.
10. Une valeur modifiable apparaît "(nouvelle variable)". Supprimez cette chaîne par défaut et entrez : `client.encoding.override`.

11. Dans la zone **Valeur**, tapez UTF-8.
12. Cliquez sur l'icône de sauvegarde ou sur **Fichier > Sauvegarder le descripteur de déploiement Web**.

## Configuration UTF-8 pour WebSphere Application Server

(Facultatif) Si vous souhaitez déployer votre application iSeries sur WebSphere Application Server, vous devez également configurer la prise en charge UTF-8 sur WebSphere Application Server et dans l'atelier.

### Configuration UTF-8 sur WebSphere Application Server 4.0 Advanced Edition

1. Démarrez la console d'administration WebSphere.
2. Développez l'icône **Noeuds** et **Nom du noeud > Serveurs d'applications > Serveur par défaut**.
3. Sélectionnez l'onglet **Paramètres JVM** et cliquez sur le bouton **Paramètres JVM avancés** pour ouvrir la boîte de dialogue **Paramètres JVM avancés**.
4. Dans la zone **Arguments de ligne de commande**, entrez :  
`-Dclient.encoding.override=UTF-8`
5. Cliquez sur **OK**, puis sur **Appliquer** dans l'onglet **Paramètres JVM**.
6. Pour appliquer cette modification à vos applications WebSphere, arrêtez le serveur par défaut, puis redémarrez-le. Pour arrêter le serveur, cliquez avec le bouton droit de la souris sur **Serveur par défaut** et sélectionnez **Arrêt**. Une fois le processus terminé, cliquez avec le bouton droit de la souris sur **Serveur par défaut** et sélectionnez **Démarrer**.

### Configuration UTF-8 sur WebSphere Application Server 4.0 Advanced Single Server Edition

1. Démarrez la console d'administration WebSphere.
2. Dans la console d'administration basée sur le navigateur, développez l'icône **Noeuds** et **Nom du noeud > Serveurs d'applications > Serveur par défaut > Définition du processus > Paramètres JVM**.
3. Ouvrez la section Paramètres avancés de la page Paramètres JVM, puis cliquez sur le lien Propriétés système. La page Propriétés système apparaît.
4. Cliquez sur **Nouveau** pour ajouter une nouvelle propriété système.
5. Dans la zone **Nom**, entrez `client.encoding.override`.
6. Dans la zone **Valeur**, tapez UTF-8.
7. Cliquez sur **OK**. Si le message **La configuration doit être sauvegardée** apparaît et qu'un lien figure dans la partie supérieure de la page **Paramètres JVM**, cliquez sur le lien pour ouvrir la page **Sauvegarde de la configuration**. Sélectionnez **Enregistrer**, puis cliquez sur **OK**.
8. Pour appliquer cette modification à vos applications, arrêtez Application Server, puis redémarrez-le. La méthode d'arrêt et de démarrage du serveur d'applications peut varier en fonction de la plateforme sur laquelle vous avez installé WebSphere Application Server. Pour plus d'informations sur l'arrêt et le démarrage du serveur d'applications, consultez la documentation WebSphere Application Server relative à votre plateforme.

---

## Création d'une feuille de style

Si vous souhaitez intégrer des pages supplémentaires avec une feuille de style en cascade (CSS), vous devez personnaliser le style du projet WebFacing ou la feuille de style en cascade pour leur donner un aspect similaire. Une fois la feuille de style personnalisée, vous pouvez utiliser l'assistant d'interaction Web pour créer une page Web détaillée utilisant la feuille de style en vue d'afficher des informations sur les articles du magasin (prix et couleur, par exemple). Pour les besoins de ce module, vous allez intégrer la feuille de style `DetailPageResults.jsp` à partir du projet SV000501. Pour votre information, vous pouvez également personnaliser le fichier CSS manuellement (voir section suivante).

### (Facultatif) Personnalisation manuelle de la feuille de style en cascade

Comme indiqué précédemment, vous pouvez personnaliser vos propres feuilles de style au lieu d'intégrer la feuille de style `DetailPageResults.jsp` du projet SV000501 dans le Project501 :

1. Ouvrez la vue Navigateur de projet et développez **Project501 > Contenu Web > styles > apparea**.
2. Cliquez deux fois sur **apparea.css** pour l'ouvrir dans CSS Designer. Cette feuille de style contrôle l'apparence de votre zone d'application.
3. Modifiez et enregistrez le fichier.
4. De retour dans la vue Navigateur de projet, placez-vous dans **Project501 > Contenu Web > styles > chrome**.
5. Cliquez deux fois sur **avenue.css** pour l'ouvrir dans CSS Designer. Cette feuille de style contrôle l'apparence de la page entière.
6. Modifiez et enregistrez le fichier.

---

## Extension et amélioration de votre projet WebFacing avec une interaction Web

Dans cette section, vous allez améliorer le projet WebFacing. Si la liste d'éléments est affichée, cliquez sur une image représentant un article pour obtenir des informations supplémentaires sur l'article correspondant. Pour ce faire, vous pouvez utiliser l'assistant d'interaction Web pour appeler un programme RPG, récupérer les détails de l'article et les afficher sur une autre page Web. Vous allez effectuer les tâches suivantes :

- Définir les informations relatives au serveur
- Copier la feuille de style appropriée
- Créer l'interaction
- Ajouter des programmes et des paramètres à l'interaction
- Modifier l'usage des paramètres

Vous devez d'abord définir les informations relatives au serveur :

1. Ouvrez la perspective Web.
2. Dans la vue Navigateur, cliquez avec le bouton droit de la souris sur **Project501** et sélectionnez **Spécification de la configuration d'exécution des outils Web iSeries**.
3. Indiquez le nom de votre serveur iSeries, votre ID utilisateur et votre mot de passe.
4. Tapez `Retailstor` dans la zone **Bibliothèque** et cliquez sur **Ajouter**.

5. Cliquez sur **Fin**. Si nécessaire, cliquez à nouveau sur **Fin**.

Avant de créer l'interaction Web, vous devez copier les styles et la feuille de styles appropriés pour que Project501 affiche le format de fichier JSP adéquat. Cette tâche n'est pas obligatoire si vous avez effectué l'exercice «(Facultatif) Personnalisation manuelle de la feuille de style en cascade» à la page 36. Copiez d'abord le dossier de styles, puis `DetailPageResults.jsp` :

1. Dans la vue Navigateur, développez **SV000501 > Contenu Web**.
2. Cliquez avec le bouton droit de la souris sur **Styles**, puis sélectionnez **Copier**.
3. Faites défiler la vue Navigateur vers le haut jusqu'à ce que **Project501** apparaisse à nouveau.
4. Développez le projet, cliquez avec le bouton droit de la souris sur **Contenu Web** et sélectionnez **Coller** dans le menu en incrustation. Cliquez sur **Oui** pour remplacer les styles existants.
5. Revenez dans **SV000501 > Contenu Web**.
6. Cliquez avec le bouton droit de la souris sur **DetailPageResults.jsp**, puis sélectionnez **Copier**.
7. Faites défiler la vue vers le haut jusqu'à ce que **Project501** apparaisse à nouveau.
8. Développez le projet, cliquez avec le bouton droit de la souris sur **Contenu Web** et sélectionnez **Coller** dans le menu en incrustation.

Les nouveaux éléments sont ajoutés sous le dossier Contenu Web de Project501.

Vous pouvez maintenant créer l'interaction Web :

1. Cliquez sur **Fichier > Nouveau > Autre**.
2. Dans la boîte de dialogue **Nouveau**, cliquez sur **Web** dans la partie gauche de la fenêtre et sur **Interaction Web** dans la partie droite. Cliquez sur **Suivant**.
3. Dans la zone **Nom de l'interaction Web**, entrez `DetailPage` et cliquez sur **Suivant**.
4. Sélectionnez le bouton d'option **Générer le JSP d'entrée**.
5. Sélectionnez le bouton d'option **Utiliser les pages de sortie** (si le bouton d'option n'est pas sélectionné) et cliquez sur **Ajouter**.
6. Dans la boîte de dialogue **JSP de sortie**, développez **Contenu Web**, sélectionnez la page de sortie préformatée **DetailPageResults.jsp** et cliquez sur **OK**.
7. Cliquez sur **Suivant**.

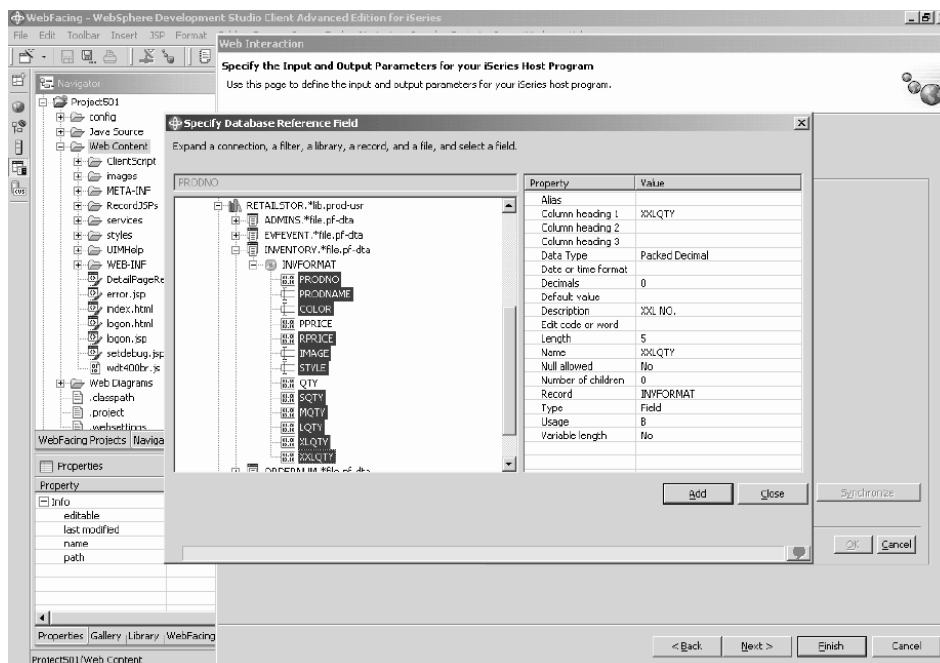
Vous pouvez maintenant ajouter des programmes et des paramètres à votre interaction. Vous devez ajouter 11 paramètres au même programme. Au lieu d'ajouter manuellement chaque paramètre, vous pouvez les ajouter plus rapidement :

1. Sélectionnez **Utiliser un programme ILE iSeries** si cette option n'est pas déjà sélectionnée.
2. Cliquez sur **Ajouter un programme**.
3. Dans la zone **Alias de programme**, entrez `DetailPage`.
4. Dans la zone **Objet de programme**, cliquez sur **Parcourir**.
  - a. Développez `serveur_iSeries > *LIBL > RETAILSTOR`.
  - b. Cliquez sur **DETAILPAGE.\*pgm.rpgle** (premier fichier après **RETAILSTOR**) et sur **OK**.

- De retour dans l'assistant d'interaction Web, cliquez sur **OK** dans le coin inférieur droit.

Dans la partie gauche de l'assistant, DetailPage est ajouté sous la section Définitions d'appel de programme.

- Dans la section Définitions d'appels de programme, cliquez sur **DetailPage**.
- Cliquez sur **Ajouter un paramètre**.
- En regard de la mention **Indiquez la zone de référence de la base de données** dans la partie inférieure de l'assistant, cliquez sur **Spécifier**.
- Développez *serveur\_iSeries* > \*LIBL > RETAILSTOR > INVENTORY.\*file.pf-dta > INVFORMAT pour afficher une liste de 13 paramètres. Vous devez ajouter les 11 paramètres suivants. Pour ce faire, cliquez sur chaque paramètre et sur **Ajouter**. Vous pouvez également maintenir la touche CTRL enfoncée tout en cliquant sur chaque paramètre et cliquer sur **Ajouter** : PRODNO, PRODNAME, COLOR, RPRICE, IMAGE, STYLE, SQTY, MQTY, LQTY, XLQTY, XXLQTY. En fait, vous sélectionnez toutes les zones à l'exception des zones PPRICE et QTY.

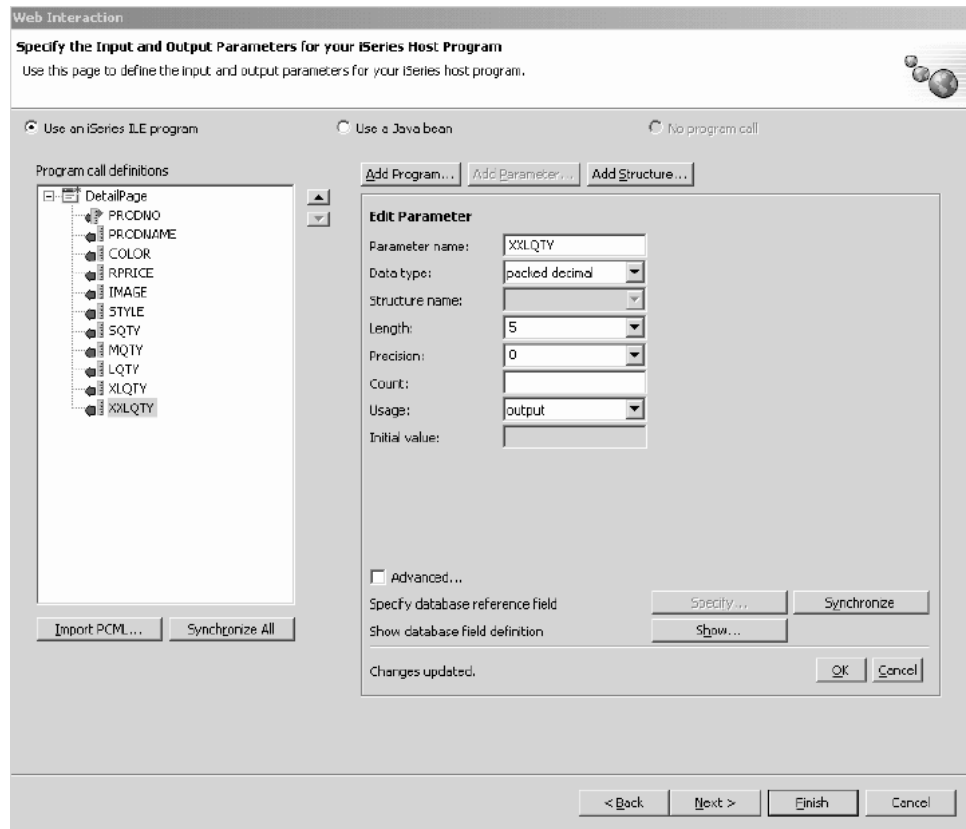


- Cliquez sur **Fermer**.

Vous devez maintenant modifier l'usage de 10 des 11 paramètres à "sortir" (tous les paramètres à l'exception de PRODNO).

- Dans l'assistant d'interaction Web, cliquez sur **PRODNAME** pour le sélectionner.

2. Dans la zone de liste **Usage**, définissez la valeur sélectionnée sur **sortie** et cliquez sur **OK**. Répétez la procédure pour tous les paramètres (à l'exception du premier paramètre PRODNO). Une fois tous les paramètres ajustés, votre espace de travail doit ressembler à la figure suivante. L'icône à gauche des paramètres peut être de type entrée, entrée et sortie, ou sortie.



3. Cliquez sur **Suivant** pour afficher l'aperçu **Formulaire d'entrée**.
4. Cliquez sur **Fin** pour créer l'interaction Web.
5. Sélectionnez **Oui global** ou **OK** et **Fin** à nouveau si vous recevez des messages.

## Liaison du projet avec une interaction Web

Maintenant que vous avez créé un fichier JSP d'interaction Web qui utilise les paramètres d'entrée et de sortie, vous devez le personnaliser pour qu'il fonctionne avec le composant WebFacing. Vous devez entrer du code pour lier votre application WebFacing et appeler cette interaction Web. Pour ce faire, vous devez ajouter une fonction JavaScript dans le fichier `webface.js` de manière à pouvoir appeler le servlet `DetailPageServlet` avec le paramètre `PRODNO` dans une nouvelle fenêtre.

Pour créer la liaison :

1. Dans la vue Navigateur, développez **Project501 > Contenu Web > ClientScript**.
2. Cliquez deux fois sur **webface.js** pour l'ouvrir dans un éditeur.
3. Placez-vous à la fin du fichier et entrez les lignes suivantes :

```
var mywindow
function next(app)
{
mywindow = window.open(app,"Details","RESIZABLE=YES, HEIGHT=700, WIDTH=800");
}
```

4. Cliquez sur l'icône de sauvegarde ou sur **Fichier > Enregistrer webface.js**.

Si vous souhaitez créer une application similaire à SV000501 dans le futur, vous devez également modifier un paramètre Web dans votre source DDS pour activer l'image que vous avez ajoutée et activer le lien de fermeture de fenêtre pour la fonction JavaScript.

Le code RPG inclus dans cette application est modifié pour indiquer la modification. Toutefois, vous devez effectuer manuellement la modification pour les applications futures. Par ailleurs, vous devez reconvertir la source DDS après avoir modifié les paramètres Web.

Vous pouvez vérifier le code pour dupliquer le résultat.

Pour afficher la source DDS :

1. Ouvrez la perspective WebFacing.
2. Dans la vue Projets WebFacing, développez **Project501 > DDS**.
3. Cliquez avec le bouton droit de la souris sur **<serveuriSeries > RETAILSTOR/QDDSSRC(QUERY)** et sélectionnez **Ouvrir avec > CODE Designer**.
4. Une fois CODE Designer ouvert, développez **SCREEN1 > ITEMSUB**.
5. Cliquez sur **IMAGESRC** pour le sélectionner.
6. Cliquez sur l'onglet **Source**.
7. Cliquez sur l'onglet **Paramètres Web** dans le coin inférieur droit de la fenêtre.

**Remarque :** Vérifiez les propriétés des paramètres Web, notamment la largeur en pixels et le nom du fichier. Dans le futur, vous devrez apporter les mêmes modifications à votre source DDS et reconvertir la source.

8. Notez la présence des lignes suivantes dans la source :

```
A   PRODNO R   0 5 6
A   PRODNAME R   0 5 16
A   IMAGESRC   19A 0 5 33
A*%WB 13 FLD 100|100|&{IMAGESRC}
A*%WB 12 FLD 1 javascript:next
      ('/Project501/DetailPageServlet?PRODNO=&{PRODNO}')
```

Vous noterez que Project501 figure à la dernière ligne. Comme indiqué précédemment, vous devez remplacer la valeur Project501 par SV000501 si vous souhaitez exécuter l'application SV000501 à nouveau.

9. Enregistrez le fichier en cliquant sur l'icône de sauvegarde ou sur **Fichier > Enregistrer** dans la barre de menus.
10. Fermez le fichier et CODE Designer.

Pour plus d'informations sur l'utilisation de la source DDS, consultez l'aide de l'atelier et la documentation de l'outil IBM WebFacing.

---

## Publication des fichiers et redémarrage du serveur

Dans cette section, vous allez publier vos fichiers et redémarrer le serveur pour appliquer toutes les modifications dans votre application Project501.

Avant de redémarrer le serveur, vous devez toutefois vérifier que Project501.war a été ajouté au fichier SVStoreEAR.



Pour vérifier votre configuration :

1. Dans la vue Navigateur, développez le dossier **Scenario folder**.
2. Cliquez deux fois sur **Scenario server.wsi**.  
Une boîte de dialogue peut vous demander si vous souhaitez que l'éditeur corrige automatiquement les entrées du projet. Cliquez sur **Oui**. Si aucune boîte de dialogue n'apparaît, aucune action n'est nécessaire. Dans tous les cas, le fichier est automatiquement modifié lorsqu'il est ouvert.
3. Enregistrez et fermez le fichier.

Pour redémarrer le serveur :

1. Ouvrez la perspective du serveur.
2. Dans la vue Configuration de serveur, dans le coin inférieur gauche, développez **Serveurs** et cliquez deux fois sur **Scenario server**.
3. Cliquez sur l'onglet **Serveurs** dans le coin inférieur droit de l'écran pour afficher l'état de votre serveur dans la vue Serveurs.
4. Cliquez avec le bouton droit de la souris sur **Scenario server** et sélectionnez **Publier**. Une fois la publication terminée, cliquez sur **OK**.
5. Dans la vue Serveurs, cliquez avec le bouton droit de la souris sur votre serveur et sélectionnez **Démarrer** ou **Redémarrer** (selon l'option activée). Si des messages indiquent que des fichiers ne sont pas enregistrés, fermez la boîte de dialogue, enregistrez tous les fichiers ouverts et recommencez.
6. Vérifiez les informations consignées dans le journal de la console (ouvert automatiquement). Le serveur est lancé lorsque la mention "Serveur *nom\_serveur* ouvert pour e-business" apparaît en bas du journal.

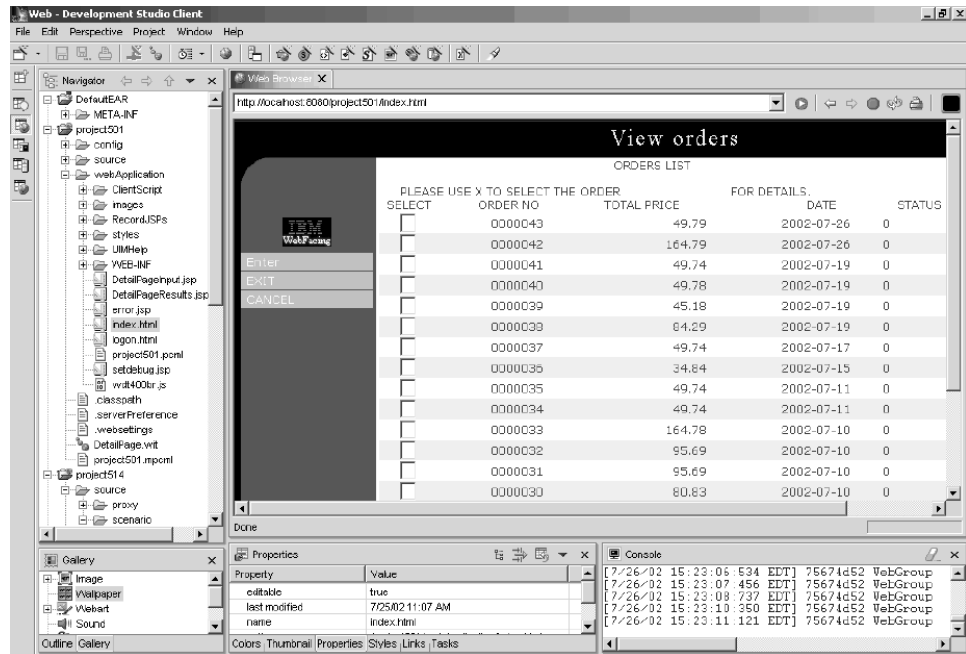
---

## Test de l'interface

Vous avez terminé la procédure de création d'une interface pour afficher les commandes et le stock. Pour tester votre interface :

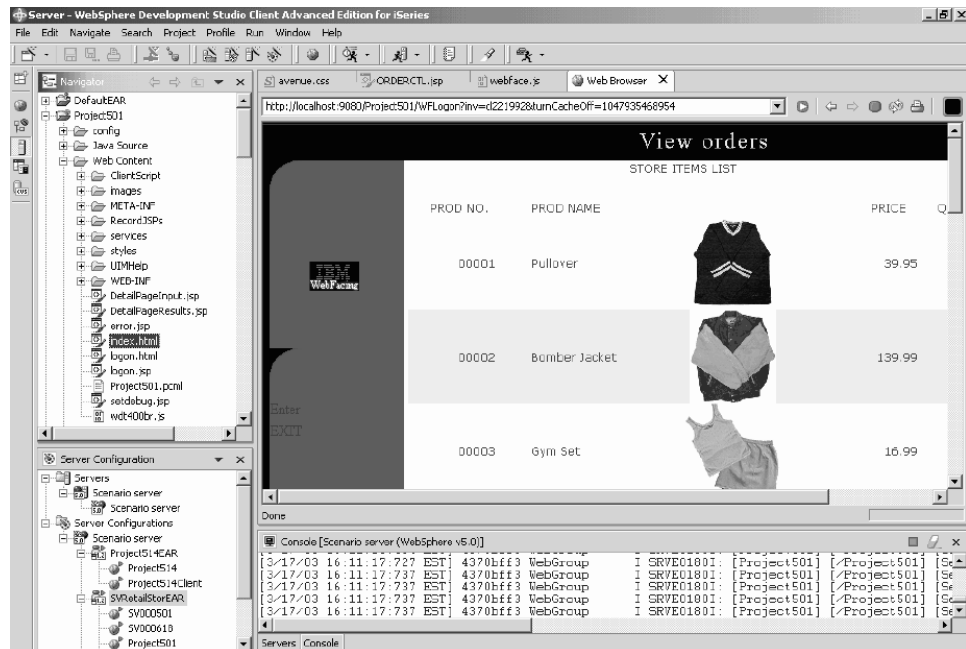
1. Ouvrez la perspective Web.
2. Développez **Project501 > Contenu Web**.
3. Cliquez avec le bouton droit de la souris sur **index.html** et sélectionnez **Exécuter sur le serveur**.
4. La boîte de dialogue **Sélection du serveur** apparaît. Vérifiez que **Scenario server** est sélectionné par défaut et cliquez sur **Fin**.

5. Cliquez sur **Visualiser les commandes – Lancer** dans la fenêtre principale du navigateur pour appeler l'application administrateur. Une fois connecté avec votre ID utilisateur et votre mot de passe iSeries (pour le serveur iSeries que vous avez utilisé au cours du développement de ce projet), le programme vous renvoie sur la page :



Tapez X en regard d'une référence produit pour afficher des détails sur ce produit.

6. Cliquez sur la flèche de retour pour aller à la page index.html et cliquez sur **Visualiser le stock – Lancer** dans la fenêtre principale du navigateur pour appeler l'application client. Le programme vous renvoie sur la page suivante (si vous êtes connecté à votre serveur iSeries avec votre ID utilisateur et votre mot de passe) :



Cliquez sur une image pour afficher des détails sur un élément spécifique.

Vous avez terminé le module ! Vous venez de créer une interface utilisateur pour afficher le stock et les articles commandés.



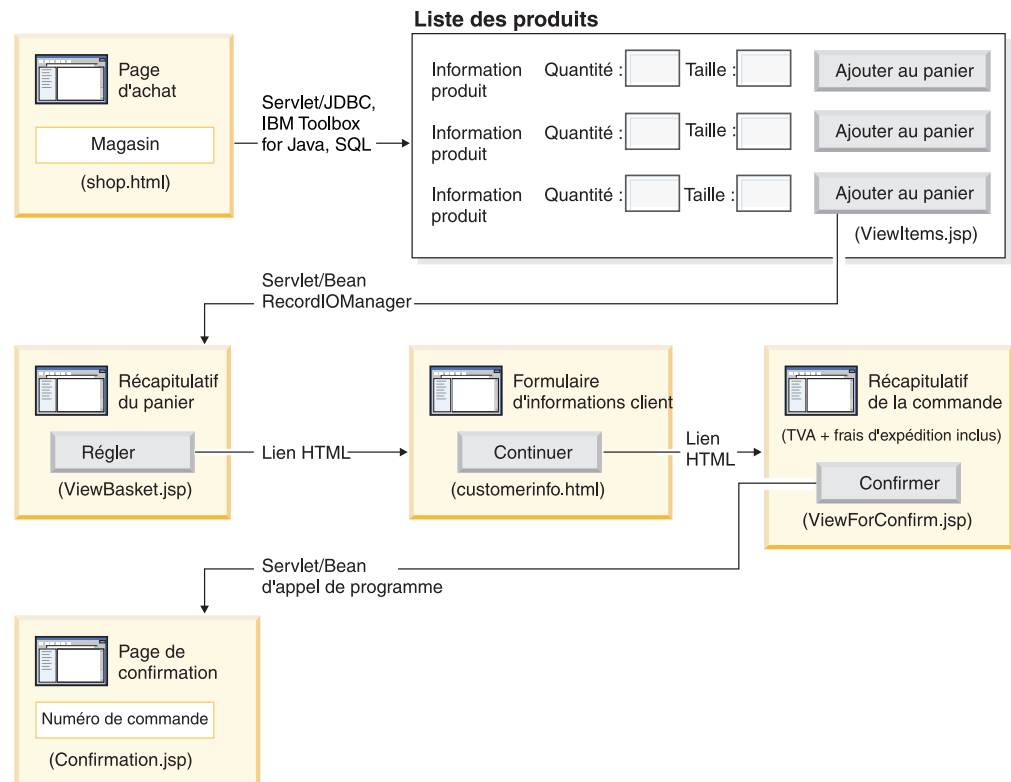
---

# Chapitre 6. Module 1 avancé : Création de code HTML, de servlets et de fichiers JSP pour passer des commandes client sur un serveur iSeries (SV001585)

---

## Introduction

Ce projet s'adresse aux utilisateurs connaissant bien la programmation Java et disposant de compétences en matière de gestion de données iSeries et de programmes RPG. Le projet vous explique comment utiliser les classes d'accès aux données IBM Toolbox for iSeries, le bean RecordIOManager et le bean d'appel de programme pour créer du code HTML, des servlets et des fichiers JSP afin de passer des commandes client sur un serveur iSeries. En tant qu'utilisateur, vous accédez à la page d'achat pour consulter les produits disponibles et ajouter des articles à votre panier. Après avoir ajouté les articles de votre choix (qui apparaissent dans le récapitulatif du panier), cliquez sur le bouton de règlement pour afficher un formulaire d'informations client. Une fois le formulaire rempli, le projet affiche un récapitulatif de commande doté d'un bouton de confirmation. Ce bouton vous renvoie à une page de confirmation, qui affiche votre commande et le numéro correspondant.



---

## Récapitulatif des grandes étapes

Ce module est un module avancé. Les instructions ne détaillent donc pas toutes les étapes de création du projet, mais présentent les étapes de développement spécifiques à l'iSeries pour la création d'un projet de ce type. Les grandes étapes sont les suivantes :

1. Ecrivez une page Web d'achat HTML.
2. Ecrivez un servlet alimentant un bean Java (à l'aide de JDBC et SQL) avec des articles pouvant être achetés par le client.
3. Ecrivez un fichier JSP pour afficher les articles à vendre et permettre aux clients d'indiquer la quantité et la taille de l'article désiré, et sélectionner l'article en cliquant sur un bouton **Ajouter au panier**.
4. A l'aide du bean RecordIOManager, écrivez un servlet appelé via le bouton **Ajouter au panier**, qui met à jour la base de données iSeries INVENTORY en soustrayant la quantité et la taille requises pour l'article et en ajoutant cette sélection à un bean Java appelé "basket". Si l'opération réussit, le servlet envoie la réponse dans le fichier ViewBasket.jsp. Dans le cas contraire, le servlet affiche une page d'erreur.
5. Ecrivez un formulaire HTML permettant au client d'entrer des informations personnelles.
6. Ecrivez une page JSP de confirmation d'achat affichant le contenu du panier, la TVA applicable, ainsi que les frais d'expédition et de manutention. Cette page doit également comprendre un bouton de confirmation pour le client.
7. Ecrivez un servlet appelé lorsque le client clique sur le bouton de confirmation, qui utilise un bean Java créé à l'aide de l'assistant d'appel de programme de l'iSeries. Une des méthodes de bean Java appelle un programme RPG pour créer une nouvelle commande dans la base de données ORDERS du serveur iSeries correspondant au panier du client. Le servlet renvoie ensuite un numéro de commande, place le bean Java sur la session de l'application Web et charge un fichier JSP de confirmation de commande contenant le numéro de commande.

---

## Avant de commencer

Vous ne pourrez effectuer les exercices que si les conditions ci-dessous sont remplies. Ces conditions prérequis sont détaillées dans le Chapitre 3, «Exécution du scénario», à la page 9.

- Vous disposez d'un accès TCP/IP à un serveur iSeries.
- Vous avez démarré les serveurs iSeries à l'aide de la commande STRTCPSVR \*ALL
- Vous avez démarré le serveur WebFacing à l'aide de la commande STRTCPSVR \*WEBFACING.
- Vous avez restauré les bibliothèques WHOLESale et RETAILSTOR sur votre serveur iSeries.
- Les bibliothèques WHOLESale, RETAILSTOR et QGPL figurent dans votre liste des bibliothèques.
- Vous avez effectué toutes les tâches du Chapitre 3, «Exécution du scénario», à la page 9 (à l'exception des tâches facultatives WebSphere Application Server, qui ne sont pas obligatoires pour tester l'application dans l'atelier).

---

## Création de la page Web, des servlets et des fichiers JSP

Pour construire les composants du projet SV001585 :

1. Créez un projet Web.
2. Dans Page Designer, écrivez une page shop.html contenant un lien qui appelle un servlet GetItems.
3. Importez le fichier jt400.jar correspondant aux classes iSeries Toolbox for Java dans le dossier *lib* du projet Web. Ce fichier jar figure dans le répertoire `x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime`, où *x* représente le répertoire dans lequel vous avez installé Development Studio.

**Remarque :** Consultez les fichiers `GetItems.java` et `ViewItems.jsp` du projet SV001585 pour connaître les principaux composants JDBC et SQL d'iSeries Toolbox for Java du servlet et du fichier JSP. In the Navigator view of the Web perspective, you can find `GetItems.java` by expanding **SV001585 > source**, and you can find `ViewItems.jsp` by expanding **SV001585 > webApplication**.

4. Ecrivez un servlet `GetItems`, qui utilise les composants JDBC et SQL d'iSeries Toolbox for Java pour récupérer les vêtements de la base de données `iSeriesINVENTORY` qui :
  - a. Place le bean `ResultSet` contenant le résultats de la requête SQL sur la session.
  - b. Redirige la requête vers `ViewItems.jsp`.

Exemple de code du fichier `GetItems.jsp` :

```
public void init() {
    .
    .
    .
    // Chargement du pilote JDBC IBM Toolbox for Java
    DriverManager.registerDriver(new com.ibm.as400.access.AS400JDBCdriver());
    // Nous avons extrait le nom, l'ID utilisateur et le mot de passe as400
    // à partir du fichier web.xml à l'aide d'un analyseur XML.
    as400conn =
    DriverManager.getConnection(
        "jdbc:as400://" + as400 + ";naming=sql;errors=full",
        userid,
        password);

    dmd = as400conn.getMetaData();
    .
    .
    .
}

public void service(HttpServletRequest request, HttpServletResponse response){
    .
    .
    .
    Statement select =
    as400conn.createStatement(
        ResultSet.TYPE_SCROLL_INSENSITIVE,
        ResultSet.CONCUR_READ_ONLY);
    ResultSet rs =
    select.executeQuery(
        "SELECT PRODNO, PRODNAME, RPRICE, IMAGE FROM "
        + retailLibrary
        + dmd.getCatalogSeparator()
        + inventoryFile);

    HttpSession session = request.getSession(true);
```

```

        session.setAttribute("resultset", rs);

        response.sendRedirect("/ViewItems.jsp");

        .
        .
        .
    }

```

5. Ecrivez un fichier JSP appelé ViewItems, qui extrait les vêtements du bean ResultSet obtenu dans l'étape précédente, pour afficher les vêtements dans un tableau. Le fichier JSP doit également contenir un formulaire distinct pour chaque article. Ces formulaires vous permettront de choisir la taille et la quantité requises, puis d'ajouter l'article à votre panier. Pour écrire le fichier JSP, vous pouvez utiliser Page Designer dans les Outils de développement Web iSeries. Plus précisément, vous pouvez organiser la page dans la vue Conception et ajouter le code approprié dans la vue Source. Exemple de code du fichier ViewItems.jsp :

```

<!--Obtention de l'objet ResultSet à partir de la session--><%
    int columnCount = 0;
    ResultSet rs = (ResultSet)session.getAttribute("resultset");
    if(rs !=null)
%>
<%
{
    rs.beforeFirst();
        ResultSetMetaData rsmd = rs.getMetaData ();
        columnCount = rsmd.getColumnCount ();

%>
<TABLE border="1">
<TBODY>
<TR>
    <TD>ID produit</TD>
    <TD>Nom</TD>
    <TD width="551">Prix</TD>
    <TD colspan="2"></TD>
</TR>
<%while (rs.next ()) {
<TR>
<!--Création d'un formulaire pour cette ligne (ou cet article)-->
<FORM name="myform" action="/SV001585/AddtoBasket"
onsubmit="return errorChecking(this);">

<!--Obtention des données de colonnes à partir de cette ligne
de l'objet ResultSet-->
<!--Le traitement des données est une méthode définie par l'utilisateur
pour modifier les données à afficher si nécessaire-->
        <%
            for (int i = 1; i <= columnCount; ++i){
                String value = rs.getString(i);
                if (rs.isNull ())
                    value = "<null>";
                else{
                    if(i==1)
                        prodID=value;
                    value = processData(i,value);

                }
            }
        <%>
        <TD><%=value%></TD>
        <%
            }
        <%>

<!--Création d'une zone de saisie pour la quantité et

```



```

d'un menu déroulant pour la taille-->
<!--Nous utilisons l'ID produit comme nom de la zone-->
  <TD width="290">Quantité
    <INPUT size="5" type="text" name='<%=prodID+"Q"%>' ><BR>
    Taille <SELECT name='<%=prodID+"S"%>'>
      <OPTION value="s" selected>Small</OPTION>
      <OPTION value="m" selected>Medium</OPTION>
      <OPTION value="l" selected>Large</OPTION>
      <OPTION value="XL" selected>Extra Large</OPTION>
      <OPTION value="XXL" selected>Extra Extra Large</OPTION>
    </SELECT>
  </TD>
<TD><INPUT type="image" name="submit" src="images/Add_to_basket.gif"></TD>
</FORM>
</TR>
<%
}
%>
</TBODY>
</TABLE>
<%
}

```

6. Importez le fichier iseriesut.jar des Outils de développement Java iSeries dans le dossier *lib* de votre projet Web. Ce fichier JAR figure dans le répertoire `x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime`, où *x* représente le répertoire dans lequel vous avez installé le produit. Consultez les fichiers `AddtoBasket.java` et `ViewBasket.jsp` du projet `SV001585` pour voir la mise en oeuvre. Dans la vue Navigateur de la perspective Web, développez **SV001585 > source** pour accéder à `AddtoBasket.java` et **SV001585 > webApplication** pour accéder à `ViewBasket.jsp`.
7. Utilisez le bean `RecordIOManager` des Outils de développement Java iSeries pour écrire un servlet `AddtoBasket` appelé par le bouton **Ajouter au panier**, qui met à jour la base de données iSeries `INVENTORY` en soustrayant la quantité demandée par le client et en ajoutant les articles à un bean Java appelé `Basket` dans la session. Exemple de code du fichier `AddtoBasket.jsp` :

```

public class AddtoBasket extends HttpServlet {

    // Classe interne de AddtoBasket
    public class MyRecordIOManager extends RecordIOManager {
        .
        .
        .
        public MyRecordIOManager(
            String hostInfo1,
            String hostInfo2,
            String hostInfo3,
            String file,
            String lib)throws Exception{
            super(hostInfo1, hostInfo2,hostInfo3,file,lib);
            setFileType(RecordIOManager.FILEACCESS_KEYED);
            setCommitLockLevel(RecordIOManager.COMMITLOCKLEVEL_ALL);
            // Le journal a le même nom que le fichier de base de données
            setJournal(file);
            // Le journal figure dans la même bibliothèque que le fichier de
            base de données etJournalLibrary(lib);
        }
        .
        .
        .
        public synchronized String updateDBFile(
            String id,
            String size,
            String quantity
        ) {

```

```

.
.
.
// Ouverture du fichier
try {
    if (openFile()) {
        record = readRecord(key);
        quantityAvailable = ((BigDecimal)
            record.getValueAt(0,sizeColumn)).intValue();
        totalQuantityAvailable = ((BigDecimal)
            record.getValueAt(0, 8)).intValue();
        if (quantityRequested <= quantityAvailable) {
            newQuantity =
                new BigDecimal(quantityAvailable - quantityRequested);
            totalNewQuantity =
                new BigDecimal
                    (totalQuantityAvailable - quantityRequested);
            record.setValueAt(newQuantity, 0, sizeColumn);
            record.setValueAt(totalNewQuantity, 0, 8);
            // Nous mettons à jour l'enregistrement sans le valider
            // au cas où le client voudrait
            // vider le panier
            // Il faut alors appeler la méthode rollBack.
            updateRecord(record);
            status = success;
        } else {
            status = notEnough;
        }
        } else
            status = accessError;
    } catch (Exception e) {
e.printStackTrace();
        status = accessError;
    }

    // Fermeture du fichier et ajout
    try {
        closeFile();
    } catch (Exception e) {
        // en cas d'annulation suite à une erreur
        try {
            rollback();
        } catch (Exception e1) {
            e1.printStackTrace();
        }
        status = accessError;
    }

    return status;
}

// Méthode d'initialisation du servlet AddtoBasket
public void init() {
    hostInfo = GetItems.getHostInfo();
}

public void doGet(HttpServletRequest req, HttpServletResponse res) {
.
.
.
    Basket basket = (Basket) session.getAttribute("basket");
    MyRecordIOManager recIO = (MyRecordIOManager) session.getAttribute("recIO");
    if (basket == null) {
        basket = new Basket();
        session.setAttribute("basket", basket);
    }
}

```

```

    }

    if(recIO == null){
    if (recIO == null) {
    try {
    recIO =
    new MyRecordIOManager(
    hostInfo[0],
    hostInfo[1],
    hostInfo[2],
    GetItems.getInventoryFile(),
    GetItems.getRetailLibrary());
    } catch (Exception e) {
    try {
    res.sendRedirect("errorPage.html");
    return;
    } catch (Exception e1) {
    e1.printStackTrace();
    }
    }
    }
    id = req.getParameter("id");
    size = req.getParameter(id + "S");
    quantity = req.getParameter(id + "Q");

    status = recIO.updateDBFile(id, size, quantity);
    session.setAttribute("recIO", recIO);

    if (status.equals("SUCCESS")) {
    basket.addItem(id, quantity, size);
    try {
    res.sendRedirect("ViewBasket.jsp");
    return;
    } catch (Exception e) {
    e.printStackTrace();
    }
    } else {
    if (status.equals("NOT_ENOUGH")) {
    try {
    res.sendRedirect("insufficient.html");
    return;
    } catch (Exception e) {
    e.printStackTrace();
    }
    } else
    if (status.equals("ACCESS_ERROR")) {
    try {
    res.sendRedirect("errorPage.html");
    return;
    } catch (Exception e) {
    e.printStackTrace();
    }
    }
    }
    }
}

```

8. Ecrivez un fichier JSP ViewBasket affichant le contenu du panier.
9. Ecrivez un formulaire customerinfo.html permettant de saisir les informations relatives au règlement. Ce formulaire doit comprendre un bouton **Continuer**, qui appelle ViewForConfirm.jsp.

10. Ecrivez un fichier ViewForConfirm.jsp, qui affiche l'ensemble du contenu du panier plus le solde total. Vous pouvez développer ViewForConfirm.jsp de la même manière que ViewBasket.jsp, en incluant en plus les frais d'expédition et un bouton **Confirmer** pour placer la commande dans la base de données iSeries ORDERS.
11. Utilisez l'assistant d'appel de programme iSeries des Outils de développement Java iSeries pour créer un bean PLACEORD.java, qui accède au programme de service PLACEORD dans la bibliothèque RETAILSTOR. L'assistant crée des beans que les applications Java ou l'assistant de services Web utilisent pour accéder aux programmes ILE iSeries.
  - a. Pour ouvrir l'assistant, cliquez avec le bouton droit de la souris sur SV001585 dans la vue Navigateur, puis sélectionnez **Nouveau > Autre**.
  - b. Dans la fenêtre **Nouveau**, cliquez sur **iSeries > Java > Bean d'appel de programme**.
  - c. Dans l'assistant d'appel de programme, indiquez les informations requises sur le programme iSeries ILE (nom, bibliothèque, type de programme, paramètres d'entrée et de sortie).
  - d. La dernière fenêtre de l'assistant vous permet de créer un bean pour une application Java, un service Web ou les deux. Pour ce projet, vous devez seulement en créer un pour une application Java.

**Remarque :** Le programme de service PLACEORD prend une table de structures et place chaque élément dans un enregistrement d'une base de données ORDERS, générant ainsi un numéro de commande en tant que donnée de sortie de chaque table.

12. Ecrivez un servlet PlaceOrder appelé par le bouton **Confirmer** de ViewForConfirm.jsp.
  - Le servlet utilise le bean généré par l'assistant d'appel de programme iSeries pour accéder à un serveur iSeries et place les commandes dans la base de données ORDERS de la bibliothèque RETAILSTOR.
  - Les commandes sont les articles du panier envoyés au programme ILE en tant que table de structures.
  - Chaque structure de la table est un article du panier.
  - Le programme de service RPG PLACEORD appelé par le bean renvoie le numéro de commande en tant que paramètre de sortie et le place sur la session.

Le segment de code suivant illustre l'utilisation du bean PLACEORD par le servlet PlaceOrder :

```

.
.
.
public void init() throws ServletException {
    hostInfo = GetItems.getHostInfo();
    super.init();

    try {
        /* Création d'une instance du bean PLACEORD créé
           par l'assistant Bean d'appel de programme iSeries */
        orderBean = new PLACEORD();
        orderBean.setConnectionData(hostInfo[0], hostInfo[1], hostInfo[2]);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
.

```

```

.
.

public void doPost(HttpServletRequest request, HttpServletResponse response) {

...

PLACEORD.Orditems_Struct inputStruct = null;

// Extraction des articles du panier
Basket basket = (Basket) request.getSession().getAttribute("basket");
AddtoBasket.MyRecordIOManager recIO = (AddtoBasket.MyRecordIOManager)
    request.getSession().getAttribute("recIO");
if (basket == null || basket.size() == 0 || recIO == null) {
    try {
        response.sendRedirect("errorPage.html");
    } catch (IOException e) {
e.printStackTrace();
    }
} else {
    items = basket.elements();
    // Définition d'une table d'éléments de structure
    while (items.hasMoreElements()) {
        item = (String[]) items.nextElement();
        inputStruct = orderBean.getOrdItemAr(j);
        inputStruct.setItemNo(new BigDecimal(item[0]));
        inputStruct.setQuantity(new BigDecimal(item[1]));
        inputStruct.setSizeOrd(item[2]);
        j = j + 1;
    }
    // Attribution de valeurs fictives aux autres éléments de la table
    for (int i = j - 1; i < 100; i++) {
        inputStruct = orderBean.getOrdItemAr(i);
        inputStruct.setItemNo(new BigDecimal(0));
        inputStruct.setQuantity(new BigDecimal(0));
        inputStruct.setSizeOrd("s");
    }

    // Définition des deux autres paramètres d'entrée du bean
    orderBean.setNumOfItems(new BigDecimal(j));
    orderBean.setBalance((BigDecimal)
        request.getSession().getAttribute("balance"));
    try {
        // Appel du programme iSeries
        orderBean.invoke();

        // Extraction du numéro de commande à partir du bean PLACEORD
        orderNumber = (orderBean.getRetCode()).toString();
        request.getSession().setAttribute("orderNumber", orderNumber);
        basket.empty();
        // Validation immédiate de la commande
        recIO.commit();
        response.sendRedirect("orderNumber.jsp");
        return;
    } catch (Exception e) {
        response.sendRedirect("errorPage.html");
e.printStackTrace();
    }
}
}
}
}

```

13. Ecrivez un servlet OrderNumber, qui récupère le numéro et le présente au client dans un message de confirmation. Si le client n'a pas ajouté d'articles à son panier, une page d'erreur apparaît.



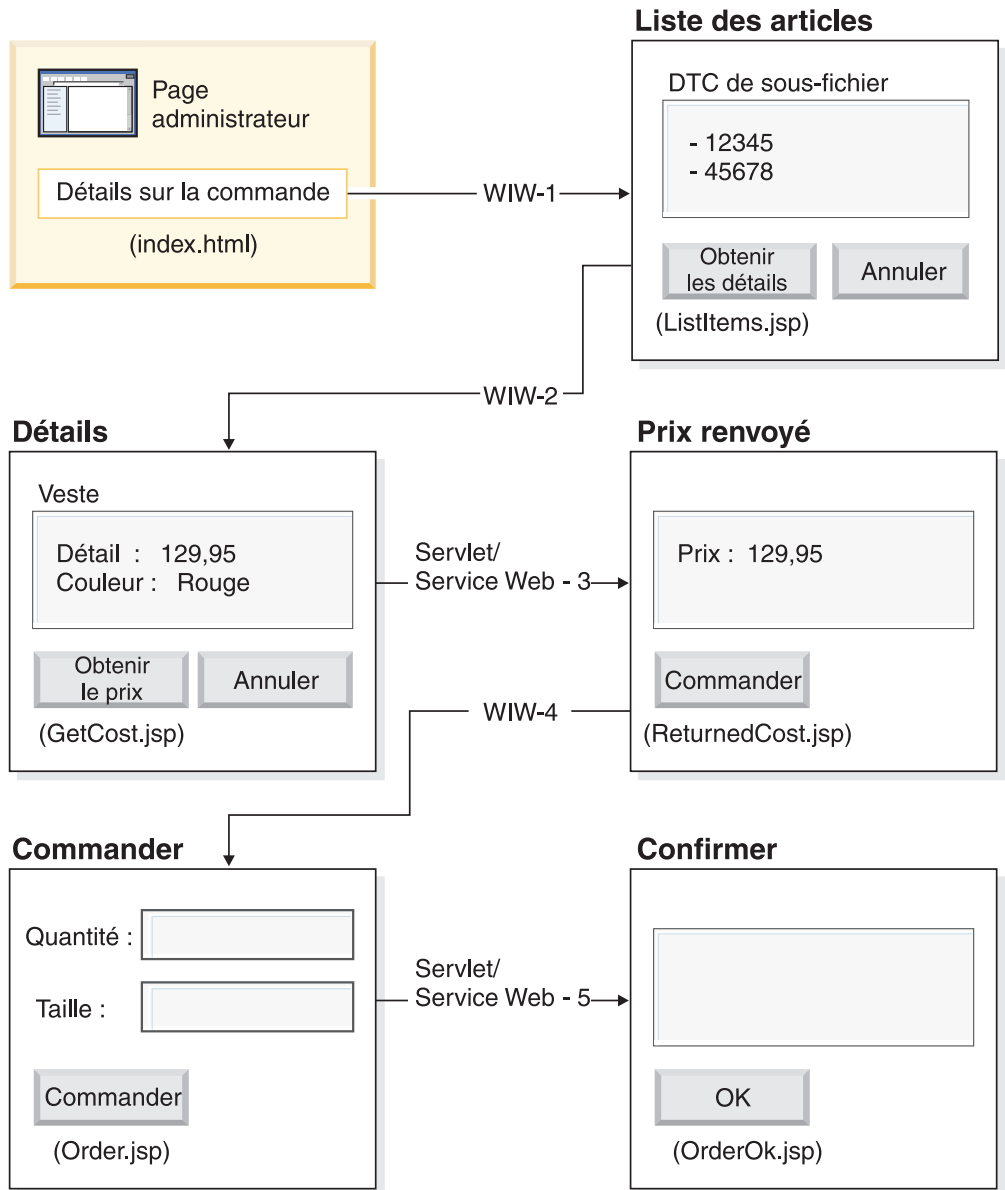
---

## **Chapitre 7. Module 2 avancé : Création d'un projet Web qui utilise les services Web SV000514 et SV001586 (SV000618)**

---

### **Introduction**

Ce projet explique comment utiliser vos connaissances en programmation RPG pour créer des clients Web pour les services Web iSeries et les programmes RPG. Dans ce projet, vous jouerez le rôle d'un administrateur qui parcourt plusieurs pages Web pour déterminer la quantité en stock et commander des articles supplémentaires pour votre magasin point de vente auprès de votre grossiste. Vous entrez le numéro d'article, affichez les détails de l'article, commandez l'article dans les quantités et la taille de votre choix, puis acceptez la confirmation.



(WIW = Assistant d'interaction Web)

Ce module est un module avancé. Les instructions ne détaillent donc pas toutes les étapes de création du projet, mais présentent les étapes de développement pour la création d'un projet de ce type. Ce projet utilise les composants suivants de Development Studio :

- Les Outils de développement Web iSeries pour créer les pages Web avec Page Designer, en intégrant la sortie de l'assistant d'interaction Web et de différents composants Web
- L'explorateur de systèmes éloignés pour créer un programme de service RPG TNLSTITM, qui renvoie les informations relatives aux articles



- L'assistant de services Web pour générer le code proxy du servlet, qui recherche les prix des articles et commande les articles
- Les Outils de développement Java iSeries pour créer les servlets nécessaires
- L'environnement de test WebSphere pour vérifier l'application avant son déploiement sur le serveur iSeries via WebSphere Application Server

---

## Avant de commencer

Vous ne pourrez effectuer les exercices que si les conditions ci-dessous sont remplies. Ces conditions prérequis sont détaillées dans le Chapitre 3, «Exécution du scénario», à la page 9.

- Vous disposez d'un accès TCP/IP à un serveur iSeries.
- Vous avez démarré les serveurs iSeries à l'aide de la commande STRTCPSVR \*ALL
- Vous avez démarré le serveur WebFacing à l'aide de la commande STRTCPSVR \*WEBFACING.
- Vous avez restauré les bibliothèques WHOLESale et RETAILSTOR sur votre serveur iSeries.
- Les bibliothèques WHOLESale, RETAILSTOR et QGPL figurent dans votre liste des bibliothèques.
- Vous avez effectué toutes les tâches du Chapitre 3, «Exécution du scénario», à la page 9 (à l'exception des tâches facultatives WebSphere Application Server, qui ne sont pas obligatoires pour tester l'application dans l'atelier).

---

## Création des pages Web, des servlets, des fichiers JSP et du code RPG

Pour construire les composants du projet SV000618 :

1. Créez un projet Web pour stocker tous les fichiers que vous allez créer.
2. Ecrivez un fichier JSP ListItems, qui répertorie les articles dans la base de données de stock iSeries. Pour écrire le servlet, vous pouvez utiliser Page Designer dans les Outils de développement Web iSeries. Plus précisément, vous pouvez organiser la page dans la vue Conception et ajouter le code approprié dans la vue Source. Vous devez également insérer un sous-fichier DTC (Design Time Control) pour interagir avec le programme de service RPG TNLSTITM, en remplissant le sous-fichier avec les enregistrements de base de données. Vous pouvez définir le programme de service dans les paramètres de contrôle DTC.

Vous devez ensuite utiliser l'assistant d'interaction Web pour créer la page d'entrée :

- Indiquez ListItems.jsp comme page de sortie pour répertorier les articles en stock. Ainsi, l'assistant d'interaction Web crée un fichier ListItems.wit.
- Veillez à ne pas définir d'appels de programme dans l'assistant d'interaction Web car le sous-fichier DTC appelle automatiquement le programme de service RPG TNLSTITM. L'assistant génère également un servlet ListItemsWitServlet, qui sert de lien pour appeler la page ListItems.jsp.
- Pour consulter le fichier ListItemsWit.wit généré par l'assistant d'interaction Web :
  1. Développez SV001618 et cliquez deux fois sur ListItems.wit pour afficher l'assistant d'interaction du fichier.
  2. Dans l'assistant, cliquez sur **Suivant** pour passer en revue les valeurs définies pour l'interaction.

Ensuite, vous devez écrire une page de sortie JSP GetCost dans Page Designer, qui reçoit les entrées de la page d'entrée ListItems.jsp. Si un utilisateur clique sur un article dans la page ListItems.jsp, une page GetCost.jsp affiche les détails correspondants.

Une fois la page GetCost.jsp créée, utilisez l'assistant d'interaction Web pour créer une interaction WitOrder entre ListItems.jsp (que vous sélectionnez comme page d'entrée) et GetCost.jsp (que vous sélectionnez comme page de sortie) :

- Dans la page Appel de programme de l'assistant, indiquez un appel de la procédure GetDetail et un paramètre du programme de service RPG TNLSTITM.
- Dans la procédure, les API du sous-fichier DTC sont intégrés pour déterminer quel enregistrement du sous-fichier a été sélectionné. La procédure utilise ces informations pour récupérer l'enregistrement sélectionné à partir de la base de données INVENTORY et affiche les détails (image) de l'article sélectionné dans le fichier GetCost.jsp.
- Pour consulter le fichier WitOrderWit.wit généré par l'assistant d'interaction Web :
  1. Développez **SV001618** et cliquez deux fois sur **WitOrder.wit** pour afficher l'assistant d'interaction du fichier.
  2. Dans l'assistant, cliquez sur **Suivant** pour passer en revue les valeurs définies pour l'interaction.

Le paramètre *flow* est défini comme contrôleur de flux sur la page de sortie. La valeur du paramètre garantit l'affichage des fichiers JSP appropriés.

Ensuite, vous devez importer les fichiers WSDL (Web Services Definition Language) du projet SV000514 pour permettre à l'administrateur de récupérer le coût d'un article chez le grossiste en appuyant sur le bouton **Get cost**.

- Le bouton **Get cost** appelle QryProdCostServlet.jsp et le service Web correspondant à partir du projet SV000514.
- Utilisez l'assistant de services Web et les fichiers WSDL importés pour générer le code proxy requis pour appeler le service Web.
- Le fichier QryProdCostServlet.jsp prend l'entrée de la page GetCost.jsp, utilise le code proxy Java pour appeler le service Web SV000514 afin de trouver le coût de l'article sélectionné et affiche le coût dans une page appelée ReturnedCost.jsp.
- Pour afficher le code QryProdCostServicesProxy.java et QryProdCostServlet.java :
  1. Développez **SV001618** > **source** > **proxy** > **soap**.
  2. Cliquez deux fois sur **QryProdCostServicesProxy.java**.
  3. Pour QryProdCostServlet.java, cliquez deux fois sur **QryProdCostServlet.java** sous **SV001618** > **source** et observez comment le code proxy Java est instancié.

Ensuite, vous devez utiliser l'assistant d'interaction Web pour lier le fichier ReturnCost.jsp comme entrée et le fichier Order.jsp comme sortie pour permettre à l'administrateur de cliquer sur un bouton **Commander** afin de commander l'article sélectionné auprès du grossiste.

- Grâce à cette interaction, vous n'avez pas besoin d'utiliser un appel de programme car la liaison des deux pages suffit pour afficher les informations appropriées.

- Pour afficher `WitPlaceOrder.wit` :
  1. Développez **SV001618**.
  2. Cliquez deux fois sur **WitPlaceOrder.wit** pour ouvrir l'interaction.
  3. Dans l'assistant, cliquez sur **Suivant** pour passer en revue les valeurs définies.

Ensuite, utilisez le service Web SV001586 pour permettre à l'administrateur d'indiquer la taille et la quantité des articles commandés.

- Importez les fichiers WSDL SV001586 dans ce projet, générez le code proxy Java pour appeler le service Web et écrivez un servlet `OrderSupplyServlet` qui sera appelé lorsque l'utilisateur cliquera sur le bouton **Commander** à partir du fichier `Order.jsp`.
- Le servlet regroupe les informations du fichier `Order.jsp`, appelle le code proxy Java du service Web, qui appelle à son tour le service Web SV001586 et commande l'article.
- Le servlet affiche le fichier `OrderOK.jsp` si la commande réussit. Dans le cas contraire, il affiche une page d'erreur.
- Pour voir comment le proxy de service Web généré est instancié et appelé pour utiliser le service Web SV001586 :
  1. Développez **SV000618 > source**.
  2. Cliquez deux fois sur **OrderSupplyServlet.java** et examinez le contenu.

---

## Avant le déploiement sur WebSphere Application Server

Avant de déployer votre application sur WebSphere Application Server, vous devez modifier une URL spécifique dans les fichiers de classe Java SV000514 et SV001586 pour permettre à l'application de fonctionner correctement.

Pour le projet SV000514 :

- Dans la classe proxy des services Web, `QryProdCostServicesProxy`, une variable définie contient l'URL du service Web à appeler.
- Une fois le proxy créé, cette URL est définie sur `http://localhost:9080/SV000514/servlet/rpcrouter`.
- Lorsque la variable est définie sur cette valeur, le service Web du projet SV000514 de l'IDE est appelé comme indiqué dans la section «Exécution de l'application dans l'atelier» à la page 18.
- Avant de déployer cette application, vous devez modifier la valeur de l'URL pour désigner l'emplacement où vous avez déployé le fichier EAR, `SVWholeSale.ear`, dans le répertoire `iSeries IFS`.

Pour le projet SV001586 :

- Dans la classe proxy des services Web, `OrderSupplyServicesProxy`, une variable définie contient l'URL du service Web à appeler.
- Une fois le proxy créé, cette URL est définie sur `http://localhost:9080/SV001586/servlet/rpcrouter`.
- Lorsque la variable est définie sur cette valeur, le service Web du projet SV001586 de l'IDE est appelé comme indiqué dans la section «Exécution de l'application dans l'atelier» à la page 18.
- Avant de déployer cette application, vous devez modifier la valeur de l'URL pour désigner l'emplacement où vous avez déployé le fichier `SVWholeSaleEAR.ear` dans le répertoire `iSeries IFS`.



---

## Chapitre 8. Remarques

Note to U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM EMEA Director of Licensing  
IBM Europe Middle-East Africa  
Tour Descartes  
La Défense 5  
2, avenue Gambetta  
92066 - Paris-La Défense CEDEX  
France

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations  
IBM Canada Ltd.  
3600 Steeles Avenue East  
Markham, Ontario  
L3R 9Z7  
Canada

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

*IBM World Trade Asia Corporation Licensing 2-31 Roppongi 3-chome, Minato-ku Tokyo 106, Japon*

**Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales : LE PRESENT DOCUMENT EST LIVRE EN L'ETAT. IBM DECLINE TOUTE RESPONSABILITE, EXPLICITE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE VALEUR MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.**

Le présent document peut contenir des inexactitudes ou des coquilles. Il est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut modifier sans préavis les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

*Lab Director IBM Canada Ltd. Laboratory 8200 Warden Avenue Markham, Ontario, Canada L6G 1C7*

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

---

## LICENCE DE COPYRIGHT

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Vous avez le droit de

copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation IBM.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

(C) (nom de votre entreprise) (année). Des segments de code sont dérivés des programmes exemples d'IBM Corp. (C) Copyright IBM Corp. 1992, 2002. Tous droits réservés.

---

## Informations relatives à l'interface de programmation

Les informations relatives à l'interface de programmation ont pour but de vous aider à créer un programme d'application utilisant ce logiciel.


Les interfaces de programmation génériques vous permettent d'écrire un programme d'application qui bénéficie des services fournis par les outils de ce logiciel.

Toutefois, ces informations peuvent également comporter des informations de diagnostic, de modification et d'optimisation. Ces informations vous aident à déboguer votre programme d'application.

**Avertissement** : N'utilisez pas ces informations de diagnostic, de modification et d'optimisation comme interface de programmation car elles sont susceptibles d'être modifiées.

---

## Marques

- 400
- AFP
- AIX
- AIX windows
- APPN
- Application System/400
- AS/400
- AS/400e
- BookManager
- C Set ++
- C/400
- CICS
- CICS/400
- CICS/ESA
- COBOL/2
- COBOL/400
- Common User Access
- CUA
- DB2
- DB2 Extenders
- DB2 Universal Database
-  **eServer**
- GDDM
- IBM
- IBMLink
- Integrated Language Environment
- iSeries
- Language Environment
- MQSeries
- Network Station
- Open Class
- Operating System/2
- Operating System/400
- OS/2
- OS/390
- OS/400
- POWER2
- PowerPC
- PROFS
- RPG/400
- RS/6000
- S/390
- SAA
- SQL/400
- System/36
- System/38
- VisualAge
- VTAM
- WebSphere

InstallShield est une marque de InstallShield Corporation.

Intel et Pentium sont des marques de Intel Corporation aux Etats-Unis et/ou dans certains autres pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

Rational est une marque de International Business Machines Corporation et Rational Software Corporation, aux Etats-Unis et/ou dans certains autres pays.

Lotus, Lotus Notes et Domino sont des marques de Lotus Development Corporation aux Etats-Unis et/ou dans certains autres pays.

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT, Win32, Win32s et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et dans certains autres pays.

Netscape Navigator est une marque de Netscape Communications Corporation.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

D'autres sociétés sont propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.







Numéro de programme : 5724-A81