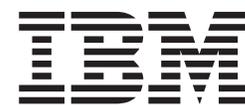




# iSeries-Szenario-Anwendung Retail Store

*Version 5.1 für Windows*





# iSeries-Szenario-Anwendung Retail Store

*Version 5.1 für Windows*



---

# Inhaltsverzeichnis

<b>Kapitel 1. Einführung in die iSeries-Szenario-Anwendung Retail Store</b> . . . . .	<b>1</b>
Projektbasierte Szenarios . . . . .	3
<b>Kapitel 2. Technologiekonzepte.</b> . . . . .	<b>5</b>
<b>Kapitel 3. Szenario ausführen</b> . . . . .	<b>9</b>
Vorbereitende Schritte . . . . .	9
Musterdateien installieren . . . . .	9
.savf-Dateien zurückschreiben . . . . .	10
Bibliotheksliste ändern. . . . .	11
Anwendung in Development Studio Client importieren und ausführen . . . . .	12
Zip-Dateien in Ihrem lokalen System extrahieren	13
Webprojekte erstellen . . . . .	13
Webprojektdateien importieren . . . . .	14
WebFacing-Projektdateien importieren und WebFacing-Projekt erstellen . . . . .	15
Serverkonfiguration . . . . .	16
WebFacing-Projekt konfigurieren . . . . .	18
Serverinformationen definieren . . . . .	18
Anwendung in der Workbench ausführen . . . . .	19
Anwendung als Kunde in der Workbench ausführen . . . . .	19
Anwendung als Administrator in der Workbench ausführen . . . . .	20
Szenario-Anwendung in WebSphere Application Server implementieren. . . . .	21
WebSphere Application Server konfigurieren . . . . .	21
Seite für den Administrator schützen . . . . .	22
EAR-Dateien für die iSeries WebSphere Application Server-Implementierung erstellen . . . . .	24
EAR-Dateien in iSeries WebSphere Application Server implementieren. . . . .	24
Anwendung in WebSphere Application Server ausführen . . . . .	25
<b>Kapitel 4. Modul 1 mit Schrittfolgen: Erstellen eines Web-Service, um Produktpreise zurückzugeben (SV000514)</b> . . . . .	<b>27</b>
Einführung . . . . .	27
Vorbereitende Schritte . . . . .	27
Neues Webprojekt erstellen . . . . .	28
iSeries-Serverinformationen definieren . . . . .	28
RPG-Serviceprogramm erstellen . . . . .	28
Parameter erstellen und JavaBean generieren . . . . .	29
Web-Service aus der JavaBean erstellen . . . . .	30
Muster testen. . . . .	31

<b>Kapitel 5. Modul 2 mit Schrittfolgen: Erstellen einer Schnittstelle, um den Bestand anzuzeigen und Artikel zu bestellen (SV000501).</b> . . . . .	<b>33</b>
Einführung . . . . .	33
Vorbereitende Schritte . . . . .	34
WebFacing-Projekt erstellen . . . . .	34
DDS-Quelle konvertieren . . . . .	35
UTF-8-Unterstützung in der Workbench konfigurieren — Nur für Benutzer von WAS Version 4.0. . . . .	36
UTF-8-Unterstützung für WebSphere Application Server konfigurieren . . . . .	37
Style-Sheet erstellen . . . . .	38
(Optional) Kaskadierendes Style-Sheet manuell anpassen . . . . .	38
WebFacing-Projekt mit einer Webinteraktion erweitern und verbessern. . . . .	38
Projekt mit einer Webinteraktion verlinken . . . . .	41
Dateien publizieren und Server erneut starten . . . . .	43
Schnittstelle testen . . . . .	43
<b>Kapitel 6. Erweitertes Modul 1: HTML-Code, Servlets und JSP-Dateien erstellen, mit denen Kundenaufträge auf einen iSeries-Server gestellt werden (SV001585)</b> . . . . .	<b>47</b>
Einführung . . . . .	47
Zusammenfassung der Hauptschritte . . . . .	48
Vorbereitende Schritte . . . . .	48
Webseite, Servlets und JSP-Dateien erstellen . . . . .	49
<b>Kapitel 7. Erweitertes Modul 2: Webprojekt erstellen, das die Web-Services in Projekt SV000514 und Projekt SV001586 verwendet (SV000618)</b> . . . . .	<b>57</b>
Einführung . . . . .	57
Vorbereitende Schritte . . . . .	59
Webseiten, Servlets, JSPs und RPG-Code erstellen . . . . .	59
Vor der Implementierung in WebSphere Application Server . . . . .	61
<b>Kapitel 8. Bemerkungen</b> . . . . .	<b>63</b>
COPYRIGHTLIZENZ: . . . . .	64
Informationen zur Programmierschnittstelle . . . . .	64
Marken und Dienstleistungsmarken . . . . .	65



---

## Kapitel 1. Einführung in die iSeries-Szenario-Anwendung Retail Store

Das Szenariopaket ist eine Musteranwendung, die mit IBM WebSphere Development Studio Client für iSeries, der Eclipse-basierten Technologie für den iSeries-Server, entwickelt wurde. Zielgruppe des Szenarios sind Anwendungsentwickler, die einen iSeries-Server für Folgendes verwenden möchten:

- Java-Entwicklung
- Webentwicklung
- Verwaltung und Umsetzung von RPG-Code in Webanwendungen

Die Anwendung besteht aus zwei .savf-Dateien, die Sie auf Ihren iSeries-Server zurückschreiben, und fünf .zip-Dateien, die Sie in das Clientprogramm importieren. Sie bilden eine Folge von Webseiten mit URLs für verschiedene Punkte innerhalb des Prozesses.

**Wichtige Anmerkung:** Überprüfen Sie unsere Unterstützungsseite auf alle Service-Packs, PTFs und andere Informationen:  
[ibm.com/software/awdtools/wdt400/support/](http://ibm.com/software/awdtools/wdt400/support/)

Dieses Szenario führt Sie durch verschiedene Teile des Produkts, wobei der Schwerpunkt auf iSeries-spezifischen Komponenten liegt, wie z. B.:

- IBM WebFacing Tool
- Web-Services
- Web Development Tools für iSeries (einschließlich des Assistenten für Webinteraktion und des Assistenten für Programmaufruf)
- Java Development Tools für iSeries
- IBM Toolbox für Java

Das Szenario zeigt zwei Unternehmen, ein Großhandelsunternehmen und ein Einzelhandelsunternehmen für Bekleidung, die in geschäftlicher Beziehung zueinander stehen und beide ihren iSeries-Server für den Geschäftsablauf und Geschäftsdaten verwenden. In der Vergangenheit erfolgte die Kommunikation zwischen den Unternehmen zwecks Überprüfung des Bestands, Übergabe von Aufträgen und Verfolgung von Aufträgen bis zum Auftragsabschluss mit Hilfe von E-Mail, Telefon und Fax. Jetzt besteht für die Unternehmen die Aussicht, das Web zur Ausführung der üblichen Geschäftstransaktionen verwenden zu können.

Das Einzelhandelsunternehmen strebt eine Website an, die

- Kunden zum Kauf von Produkten verwenden können.
- Mitarbeiter zum Bestellen von Waren von dem Großhandelsunternehmen verwenden können.

Das Großhandelsunternehmen möchte

- Bestellungen des Einzelhandelsunternehmens online zur Verfolgung von Bestellungen empfangen.
- mehrere potenzielle Kunden betreuen.

In diesem Szenario übernehmen Sie die Rolle eines Programmierers, der beide Unternehmen berät und sie bei der Umstellung Ihrer Geschäftsabläufe auf das Web unterstützt.

Abhängig vom Benutzertyp hat die Anwendung zwei unterschiedliche Eingangspunkte. Als Kunde ist der Startpunkt die Anzeige der Produkte, die das Unternehmen anbietet, in diesem Fall Freizeitkleidung. Möchten Sie ein Produkt kaufen, können Sie einen Link anklicken, um auf eine Anzeige zum Bestellen zuzugreifen. Nach der Bestellung wird eine Übersichtsseite generiert und Sie können Ihren Einkauf fortsetzen, die Bestellung abrechnen oder die Bestellung übergeben.

Als Administrator haben Sie eine sichere Benutzer-ID für die Anwendung, die die Definition der Sicherheitsstrategie für die Anwendung während der Implementierung erfordert. Ihr Eingangspunkt ist eine Anmeldeanzeige, in der Sie Ihre Bestellungen und den Bestand anzeigen können sowie Produkte bei dem Großhandelsunternehmen kaufen können. Sie könnten Artikel auswählen, prüfen, welchen Preis der Artikel zuletzt bei dem Großhandelsunternehmen hatte, und die gewünschte Größe und Menge bestellen. Die Anwendung prüft, ob der Großhändler die von Ihnen gewünschte Größe und Menge auf Lager hat, und bestätigt entweder den Auftrag oder teilt Ihnen mit, dass er den Auftrag zu diesem Zeitpunkt nicht erfüllen kann.

Unter der Oberfläche der Anwendung finden in den verschiedenen Teilen des Programms viele Aktionen statt. Die folgende Tabelle zeigt den Prozess und die Komponente des Programms, der bzw. die für die einzelnen Teile der Anwendung zuständig ist. Die Vorgehensweise zur Ausführung jeder Task ist in den ausführlichen Informationen beschrieben.

Tabelle 1.

Kundenanwendung	Administratoranwendung	Zugrunde liegender Prozess
Produktpreise anzeigen		Verwenden Sie ein iSeries-RPG-Programm, um einen Webservice zu erstellen, und rufen Sie die Preise mit Web Development Tools auf.
Auftrag des Einzelhändlers übergeben		Verwenden Sie Servlets und JSP-Dateien zusammen mit iSeries Java Development Tools, iSeries Web Development Tools und IBM Toolbox für Java, um auf den Bestand auf einem iSeries-Server zuzugreifen und diesen anzuzeigen und um Aufträge zu erteilen und eine Übersicht über die gekauften Artikel anzuzeigen.
	Bestand anzeigen	Verwenden Sie das IBM WebFacing Tool, um ein vorhandenes RPG-Programm in eine Webanwendung zu konvertieren, und verwenden Sie Web Development Tools, um die Webseite anzupassen.
	Bestellung bei dem Großhandelsunternehmen anhand von Produkt-ID und Menge in Auftrag geben	Erstellen Sie einen Web-Service, der aufgerufen wird, wenn Sie den Knopf <b>Purchase</b> anklicken.
Startwebseite für das Unternehmen anzeigen	Startwebseite zum Bestellen von Bestand anzeigen	Verwenden Sie Web Development Tools für iSeries, um beide Homepages zu erstellen.

---

## Projektbasierte Szenarios

Die iSeries-Szenario-Anwendung besteht aus fünf Projekten, die die Namen SV000501, SV000514, SV000618, SV001585 und SV001586 haben. Dieses Handbuch enthält Anweisungen zur Vorgehensweise bei der Ausführung der Standardsoftware. Dieses Handbuch enthält außerdem Module mit Schrittfolgen und erweiterte Module, die zeigen, wie Sie die verschiedenen Projekte selbst erstellen können. Die Module mit Schrittfolgen sind für Anwendungsentwickler gedacht, die noch nicht viel Erfahrung in der Anwendungsentwicklung und mit Development Studio Client haben. Die erweiterten Module sind für Anwendungsentwickler gedacht, die über Erfahrungen in der Anwendungsentwicklung verfügen und bereits mit Development Studio Client vertraut sind.

**Anmerkung:** Obwohl es fünf Projekte gibt, gibt es nur vier Module, da Projekt SV001586 und SV000514 kombinierte Web-Services-Module sind.

Dieses Szenario besteht aus fünf Projekten:

**Projekt SV000501: Erstellen eines Webprojekts, um offene Aufträge, den Bestand und Produktdetails anzuzeigen** – Dieses Projekt wird mit iSeries Web Development Tools und dem IBM WebFacing Tool erstellt und richtet sich an RPG-Programmierer mit geringen Kenntnissen in der Webanwendungsentwicklung, die das IBM WebFacing Tool verwenden möchten, um Ihre RPG-Anwendungen in das Web zu stellen.

**Projekt SV001585: Erstellen von HTML-Code, Servlets und JSP-Dateien, die Kundenbestellungen auf einen iSeries-Server stellen** – Dieses Projekt verwendet die SQL- und JDBC-Klassen von IBM Toolbox für Java, die Bean RecordIOManager von iSeries Java Development Tools und den iSeries-Assistenten für Programmaufruf. Diese Elemente zeigen verschiedene Möglichkeiten, um auf Daten und Programme auf dem iSeries-Server zuzugreifen und diese zu bearbeiten. Dieses Projekt ist für Java-Programmierer und Webanwendungsentwickler gedacht, die Webseiten für den Zugriff auf iSeries-Daten und -Code entwickeln möchten. Außerdem sollten Sie über Erfahrung in der iSeries-Serververwaltung und der RPG-Programmierung verfügen.

**Projekt SV000514: Erstellen eines iSeries-Web-Service zur Angabe von Produktpreisen** – Der Assistent für Web-Services verwendet eine vom iSeries-Assistenten für Programmaufruf generierte JavaBean, um eine oder mehrere Programmprozeduren auf dem iSeries-Server aufzurufen und die Informationen zurück an einen Browser zu senden. Dieses Projekt richtet sich an RPG-Programmierer, die mit Hilfe von Web-Services eigenständige, modulare Anwendungen erstellen möchten, die über das World Wide Web beschrieben, publiziert, lokalisiert und aufgerufen werden können.

**Projekt SV001586: Erstellen eines Web-Service, um dem Großhandelsunternehmen über einen iSeries-Server Aufträge zu erteilen** – Der Web-Service akzeptiert eine Produkt-ID sowie die erforderliche Menge und erteilt dann dem Großhandelsunternehmen einen Auftrag. Dieses Projekt ist eine Komponente von SV000514 und ist für RPG-Programmierer gedacht, die Web-Services erstellen möchten.

**Projekt SV000618: Erstellen eines Webprojekts, um eine Schnittstelle zu dem/der vom IBM WebFacing Tool generierten Auftragsformular, Bestandsformular und Bestellung zu erstellen** – Dieses Projekt erfordert iSeries Web Development Tools und umfasst das Erstellen von HTML- und JSP-Dateien, um die in SV000514 und SV001586 entwickelten Web-Services verwenden und eine Verbindung zu ihnen herstellen zu können. Das Projekt richtet sich an Anwendungsentwickler, die mit Web-Services arbeiten möchten und über Kenntnisse in der RPG- und Java-Programmierung verfügen.

---

## Kapitel 2. Technologiekonzepte

Um die Szenario-Anwendung durcharbeiten zu können, müssen Sie mit einer Reihe von Technologiekonzepten vertraut sein; dies gilt insbesondere dann, wenn Sie noch nicht über Erfahrungen in der Webanwendungsentwicklung verfügen. Die folgende kurze Liste enthält einige der Begriffe, auf die Sie beim Durcharbeiten der Anwendung treffen.

### **Enterprise Archive-Datei (EAR-Datei)**

Eine EAR-Datei ist eine Standard-JAR-Datei (JAR = Java Archive) mit der Erweiterung .ear. EAR-Dateien können mehrere Webprojekte enthalten und werden zum Packen und Implementieren Ihrer Webanwendungen in WebSphere Administrative Server (WAS) verwendet. **Anmerkung:** In der GUI-Version des J2EE-SDK-Anwendungsimplementierungstools erstellen Sie zunächst eine EAR-Datei und fügen der EAR-Datei JAR- und WAR-Dateien hinzu. Verwenden Sie die Befehlszeilenpackprogramme, erstellen Sie jedoch zunächst die JAR- und WAR-Dateien und dann die EAR-Datei.

### **IBM WebFacing Tool**

Das IBM WebFacing Tool konvertiert vorhandene 5250-Schnittstellen in browserbasierte grafische Benutzerschnittstellen. Mit geringem oder gar keinem Änderungsaufwand an Ihren ursprünglichen iSeries-Anwendungen können Sie die Verwendung Ihrer Programme auf das Internet oder ein Intranet ausdehnen.

### **Java Archive-Datei (JAR-Datei)**

Eine JAR-Datei ist ein komprimiertes Paket von Java-Dateien, ähnlich wie eine .zip-Datei. Sie enthält die Klassen-, Image- und Audiodateien für ein Java-Applet, die in einer einzigen Datei zusammengefasst und komprimiert sind, um den Download in Ihren Browser zu beschleunigen.

### **Java Server Pages (JSP)**

Mit Hilfe von JSPs kann der dynamische Inhalt in statischen HTML-Seiten angezeigt werden. JSPs sind in Java geschrieben und vom Server und der Plattform unabhängig. Da JSPs die Webdarstellung effektiv von den Webinhalten trennen, sind JSPs für Anwendungsentwickler hilfreich, die schnell den Entwurf und die Anzeige ihrer Webseiten ändern müssen.

### **Programmaufruf-Bean**

Dabei handelt es sich um die vom Assistenten für Programmaufruf generierten JavaBeans. Ein Typ ist eine normale JavaBean, die von Java-Anwendungen verwendet wird. Der andere Typ kann vom Assistenten für Web-Services verwendet werden, um einen Web-Service zu erstellen.

### **Assistent für Programmaufruf**

Der Assistent für Programmaufruf unterstützt Sie beim Erstellen der JavaBeans und der zugehörigen PCML-Datei, die zum Aufrufen eines iSeries-Programms oder einer iSeries-Prozedur erforderlich sind. Der Assistent fordert Sie zur Eingabe von

Informationen zu Programm- oder Serviceprogrammobjekten sowie den Parametern für die Objekte auf und erstellt dann die gewünschten JavaBeans (und die PCML-Datei).

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

Eine prozedurale Programmiersprache, die von iSeries-Programmierern verwendet wird. Mit Hilfe von RPG können Sie Geschäftsanwendungen erstellen, wie z. B. Programme für die Rechnungsstellung oder Auftragserfassung. Die letzte Version, ILE RPG IV, erweitert die Funktionalität der Sprache RPG bei gleichzeitiger Unterstützung der Erfahrungen der Programmierer mit vorherigen Versionen.

### **Servlet**

Programme auf Serverseite, die in Java geschrieben sind und in Java-fähigen Servern oder Anwendungsservern, wie z. B. IBM WebSphere Application Server, ausgeführt werden. Servlets führen Tasks aus, die durch den Server angegeben wurden, wie z. B. das Antworten auf Anforderungen durch Generieren einer HTML-Antwort. Sie können Servlets beispielsweise in einer Anwendung für Onlinebanking verwenden, um dem Benutzer zu antworten, während Daten an den Server gesendet werden.

### **Webkomponenten**

Mit Hilfe von Webkomponenten können Sie iSeries-Objekte, wie Dateneingabefelder und Druckknöpfe, definieren, die Daten zwischen iSeries-Serverprogrammen und der Webseite austauschen können. Anwendungsentwickler können Webkomponenten verwenden, um Benutzerereignisse, wie z. B. die Syntaxprüfung von Eingabefeldern und das Anklicken von Schaltflächen, zu erfassen.

### **Assistent für Webinteraktion**

Dieser Assistent ist Teil der iSeries Web Development Tools. Er erstellt und verwaltet die Interaktionen zwischen ILE-Programmen und Webseiten. Der Assistent steuert, wo Eingabe, Ausgabe und Fehlermeldungen angezeigt werden, und überträgt die Daten aus den Eingabe- und Ausgabefeldern in die ILE-Programme. Sie können den Assistenten für Webinteraktion auch verwenden, um Fehlermeldungen dem Bereich zuzuordnen, in dem der Fehler aufgetreten ist, sodass der Benutzer die Quelle des Fehlers problemlos identifizieren kann.

### **Web-Services**

Web-Services sind eigenständige Anwendungen, die für die Verwendung im Internet konzipiert und implementiert wurden. Sie werden mit offenen Standards, wie z. B. SOAP, WSDL und XML, erstellt. Es gibt zahlreiche Situationen in Geschäftsabläufen, in denen Sie einen Web-Service verwenden können, einschließlich eines Bestandsführungssystems, in dem Auftraggeber ihren Lagerbestand über das Internet überprüfen können, oder wenn eine Produktbestellung verfolgt werden soll, die direkt bei einem Lieferanten erfolgt.

### **Web Services Definition Language (WSDL)**

WSDL ist eine XML-basierte Sprache, die die Schnittstelle eines Web-Service definiert. WSDL versteht einen Web-Service und verwaltet den Informationsfluss zwischen dem Web-Service und dem Serverprogramm. Beispielsweise würde ein

Anwendungsentwickler WSDL verwenden, um eine Schnittstelle für eine Website zu erstellen, die aktualisierte Börsennotierungen zeigt.

### **WebSphere Studio Workbench**

IBM WebSphere Development Studio Client für iSeries wurde auf WebSphere Studio Workbench, der IBM Implementierung der Eclipse-Plattform, erstellt. Die erweiterbare universelle Workbench integriert alle Tools, die zum Erstellen und Verwalten von Anwendungen erforderlich sind. Anwendungsentwickler können Development Studio Client verwenden, um neue Objekte über die Verwendung von Plug-ins in die Entwicklungsumgebung einzugliedern und problemlos Java-Dateien, Grafiken, Videos usw. hinzuzufügen.



---

## Kapitel 3. Szenario ausführen

Sie können die Anwendungen *Wholesale* und *Retail store* innerhalb der Development Studio Client Workbench ausführen oder unter WebSphere Application Server für jede Plattform einschließlich der iSeries-Plattform. Eine Übersicht über die Anwendungen befindet sich in Kapitel 1, „Einführung in die iSeries-Szenario-Anwendung Retail Store“, auf Seite 1.

Am Ende dieses Kapitels sollten Sie in der Lage sein, folgende Aufgaben auszuführen:

- Zurückschreiben der Musterdateien und Development Studio Client-Objekte auf Ihren iSeries-Server
- Erstellen von Webprojekten und eines WebFacing-Projekts zum Aufnehmen der Dateien
- Importieren der zurückgeschriebenen Dateien von Ihrem iSeries-Server in Development Studio Client
- Konfigurieren von WebSphere Application Server (WAS)
- Konfigurieren des WebFacing-Projekts
- Ausführen der Anwendung als Kunde in der WebSphere Test Environment
- Ausführen der Anwendung als Administrator in der WebSphere Test Environment
- Konfigurieren Ihres WebSphere Application Servers
- Schützen der Seite für den Administrator
- Erstellen von EAR-Dateien für die externe WAS-Implementierung
- Implementieren der Anwendung in WAS
- Ausführen der Anwendung in WAS

---

### Vorbereitende Schritte

Um die Anwendungen von der Workbench aus zu testen, müssen Sie Folgendes sicherstellen:

- Sie verfügen über einen Server mit iSeries-Version 5 Release 1 oder später
- Sie haben die aktuellen WebFacing-PTFs angelegt. PTFs finden Sie auf unserer Unterstutzungsseite: <http://www.ibm.com/software/awdtools/wdt400/support/>
- Sie verfügen über NET USE-Zugriff auf den iSeries-Server.

---

### Musterdateien installieren

Um die iSeries-Szenario-Anwendung verwenden zu können, müssen Sie mit den folgenden Dateien arbeiten:

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

- SV001586.zip

Die .savf-Dateien enthalten iSeries-Daten und RPG-Programme und die .zip-Dateien enthalten die Webanwendungen, die mit den iSeries-Programmen interagieren, um iSeries-Daten zu bearbeiten. Zunächst müssen Sie die .savf-Dateien auf Ihren iSeries-Server zurückschreiben; dann können Sie die .zip-Dateien in Development Studio Client importieren und die Anwendung in der Workbench ausführen.

## **.savf-Dateien zurückschreiben**

Um mit den Beispielen in diesem Handbuch arbeiten zu können, müssen Sie die Bibliotheken WHOLESale und RETAILSTOR auf Ihren iSeries-Server zurückschreiben. Sie sollten diese Task selbst dann ausführen, wenn Sie die Bibliotheken bereits für ein vorheriges Release des Produkts zurückgeschrieben haben, da ihr Inhalt unterschiedlich ist.

**Anmerkung:** Die .savf-Dateien, die zum Installieren der Musterbibliothek verwendet wurden, sind für die Verwendung mit einem iSeries-Server V5R1 oder höher gedacht. Zum Zweck dieses Szenarios werden beide Bibliotheken auf denselben iSeries-Server zurückgeschrieben; würden Sie diese Anwendung für ein tatsächlich existierendes Unternehmen entwickeln, würden Sie diese zwei Bibliotheken auf zwei unterschiedliche iSeries-Servers zurückschreiben. Sie würden die Bibliothek WHOLESale auf den iSeries-Server zurückschreiben, der die Web-Services zur Verfügung stellt, und die Bibliothek RETAILSTOR auf den iSeries-Server, der zu dem Einzelhandelsunternehmen gehört.

Um die Datei Wholesale.savf zurückzuschreiben, gehen Sie wie folgt vor:

1. Melden Sie sich über einen Green-Screen-Emulator bei Ihrem iSeries-Server an.
  - a. Erstellen Sie eine Bibliothek, die die Sicherungsdateien enthalten soll. Um eine neue Bibliothek in dem Emulator zu erstellen, geben Sie CRTLIB ein.
  - b. Ordnen Sie Ihrer Bibliothek den Namen SCENARIO zu.
  - c. Springen Sie mit der Tabulatortaste in die nächste Zeile; geben Sie \*TEST als Bibliothekstyp ein und drücken Sie die Eingabetaste, um Ihre Änderungen zu sichern.
  - d. Erstellen Sie mit dem Befehl CRTSAVF zwei Sicherungsdateien; drücken Sie zwischen den beiden folgenden Zeilen die Eingabetaste:

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

Diese Zeilen geben an, dass Sie Sicherungsdateien in Ihrer Bibliothek SCENARIO erstellen möchten.

2. Öffnen Sie auf Ihrer Workstation eine Eingabeaufforderung.
  - a. Sie müssen in das Verzeichnis wechseln, in dem sich die .savf-Dateien befinden. Geben Sie standardmäßig `cd c:\wdsc\wdscsampl` ein. Wenn Sie das Produkt auf einem anderen Laufwerk installiert oder nicht "wdsc" als Ausgangsverzeichnis für Development Studio Client ausgewählt haben, müssen Sie in das Verzeichnis wdscsampl an der entsprechenden Position wechseln.
  - b. Geben Sie in die Befehlszeile `ftp Hostname` ein; dabei ist *Hostname* der Name Ihres iSeries-Servers, z. B. PROD400.
  - c. Geben Sie Ihre Benutzer-ID und Ihr Kennwort für den iSeries-Server ein.

- d. Geben Sie in der Befehlszeile `cd /qsys.lib/scenario.lib` ein, um zu Ihrer Bibliothek SCENARIO zu wechseln.
- e. Geben Sie die folgenden Zeilen ein:

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

Diese Zeilen (vor `quit`) geben an, dass die Sicherungsdateien von Ihrem lokalen System auf den iSeries-Server übertragen werden sollen.

3. Zurück in der iSeries-Konsole speichern Sie Ihre Bibliothek WHOLESale zurück:
  - a. Geben Sie `RSTLIB` ein und drücken Sie die Taste F4, um zu definieren, wie die Bibliothek zurückgeschrieben werden soll.
  - b. Geben Sie in das Feld **Gesicherte Bibliothek** den Wert WHOLESale ein und drücken Sie die Tabulatortaste.
  - c. Geben Sie im Feld **Einheit** den Wert `*savf` ein und drücken Sie die Tabulatortaste.
  - d. Drücken Sie im nächsten Feld die Eingabetaste, um weitere Werte anzuzeigen, und springen Sie mit der Tabulatortaste in das Feld **Sicherungsdatei**.
  - e. Geben Sie WHOLESale in das Feld **Sicherungsdatei** ein und drücken Sie die Tabulatortaste.
  - f. Löschen Sie im Feld **Bibliothek** den vorhandenen Wert und geben Sie `scenario` ein.
  - g. Drücken Sie die Eingabetaste, um die Bibliothek WHOLESale auf Ihren iSeries-Server zurückzuschreiben.
4. Wiederholen Sie diese Prozedur für die Datei `Retailstor.savf`:
  - a. Geben Sie `RSTLIB` ein und drücken Sie die Taste F4, um zu definieren, wie die Bibliothek zurückgeschrieben werden soll.
  - b. Geben Sie in das Feld **Gesicherte Bibliothek** den Wert RETAILSTOR ein und drücken Sie die Tabulatortaste.
  - c. Geben Sie im Feld **Einheit** den Wert `*savf` ein und drücken Sie die Tabulatortaste.
  - d. Drücken Sie im nächsten Feld die Eingabetaste, um weitere Werte anzuzeigen, und springen Sie mit der Tabulatortaste in das Feld **Sicherungsdatei**.
  - e. Geben Sie RETAILSTOR in das Feld **Sicherungsdatei** ein und drücken Sie die Tabulatortaste.
  - f. Löschen Sie im Feld **Bibliothek** den vorhandenen Wert und geben Sie `scenario` ein.
  - g. Drücken Sie die Eingabetaste, um Ihre Aktion zu sichern und die Bibliothek RETAILSTOR auf Ihren iSeries-Server zurückzuschreiben.

## Bibliotheksliste ändern

Obwohl Sie die Bibliotheken RETAILSTOR und WHOLESale auf Ihren iSeries-Server zurückgeschrieben haben, müssen Sie sicherstellen, dass sie in Ihrer Bibliotheksliste, die Sie bei der Anmeldung verwenden, vorhanden sind. Andernfalls können Ihre Webanwendungen die Bibliotheken möglicherweise nicht finden und es können Probleme beim Ausführen oder Erstellen der Anwendungen auftreten. Sie müssen außerdem sicherstellen, dass sich die Bibliothek QGPL in Ihrer Bibliotheksliste befindet.

Hierfür müssen Sie Ihre Jobbeschreibung und Ihr Benutzerprofil aktualisieren, damit die Änderungen an Ihrer Bibliotheksliste permanent sind.

Verschiedene Benutzer ändern ihre Bibliotheksliste auf unterschiedliche Art und Weise; das nachfolgende Beispiel zeigt eine Möglichkeit:

1. Geben Sie auf Ihrem iSeries-Server `dspl ibl` ein und drücken Sie die Eingabetaste, um Ihre Bibliotheksliste anzuzeigen.
2. Prüfen Sie, ob die Bibliotheken RETAILSTOR, WHOLESALE und QGPL in Ihrer Bibliotheksliste vorhanden sind. Ist dies der Fall, können Sie beenden und Ihr Emulationsfenster schließen. Ist dies nicht der Fall, gehen Sie wie folgt vor:
  - a. Drücken Sie die Taste F12, um Ihre aktuelle Anzeige abzubrechen.
  - b. Suchen Sie den Namen Ihrer Jobbeschreibung. Geben Sie hierfür `dspusrprf Benutzername` ein und drücken Sie die Eingabetaste, um Ihr Benutzerprofil anzuzeigen. Hierbei steht *Benutzername* für die ID, mit der Sie sich beim iSeries-Server anmelden. Ihre Jobbeschreibung ist auf einer der Seiten Ihres Benutzerprofils angegeben (blättern Sie vor). Wenn Sie Ihre Jobbeschreibung gefunden haben, machen Sie eine Notiz und drücken F12, um die Anzeige zu verlassen.
  - c. Geben Sie `chgjobd` in die Befehlszeile ein und drücken Sie F4.
  - d. Geben Sie in das Feld **Jobbeschreibung** den Namen Ihrer Jobbeschreibung ein und drücken Sie F10, um zusätzliche Parameter aufzurufen.
  - e. Blättern Sie zur **Anfangsbibliotheksliste** vor.
  - f. In dem Feld unter **Anfangsbibliotheksliste** wird "+ für weitere Werte" angezeigt. Geben Sie ein Pluszeichen ein und drücken Sie die Eingabetaste.
  - g. Löschen Sie auf der Seite **Specify more values** (Weitere Werte angeben) alle vorhandenen Einträge und geben Sie RETAILSTOR, WHOLESALE und QGPL ein. Drücken Sie dabei zwischen den einzelnen Werten die Tabulatortaste. Die Eingabe der Bibliotheken in dieser Reihenfolge ist verbindlich, da die Anwendungen zuerst die Bibliothek RETAILSTOR finden müssen. Drücken Sie die Eingabetaste.
  - h. Drücken Sie erneut die Eingabetaste, um Ihre Änderungen zu sichern.
  - i. Geben Sie `chgusrprf` in die Befehlszeile ein und drücken Sie die Eingabetaste.
  - j. Geben Sie Ihren Benutzernamen ein und drücken Sie die Eingabetaste.
  - k. Drücken Sie die Taste F10, um zusätzliche Parameter anzuzeigen.
  - l. Blättern Sie bis zur **Jobbeschreibung** vor.
  - m. Stimmt der Name der Jobbeschreibung nicht mit der Jobbeschreibung überein, die Sie mit dem Befehl `dspusrprf` ermittelt haben, geben Sie den Namen der Jobbeschreibung ein, den Sie mit diesem Befehl ermittelt haben, und drücken die Eingabetaste, um Ihre Änderungen zu sichern.

---

## Anwendung in Development Studio Client importieren und ausführen

Fünf zip-Dateien sind in Ihren installierten Programmdateien für Development Studio Client vorhanden. Sie befinden sich standardmäßig im Verzeichnis `c:\wdsc\wdscsampl`. Hierbei handelt es sich um:

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

- SV000501.zip

Sie importieren jede Datei separat in die Workbench von Development Studio Client und geben für jede Datei eine entsprechende EAR-Datei (EAR = Enterprise Archive) an. Eine EAR-Datei ist eine Standard-JAR-Datei (JAR = Java Archive) mit der Erweiterung .ear. EAR-Dateien können mehrere Webprojekte enthalten und werden zum Packen und Implementieren Ihrer Webanwendungen in WebSphere Administrative Server (WAS) verwendet.

Die ersten vier Dateien entsprechen den Webprojekten, die fünfte entspricht einem WebFacing-Projekt. Zunächst erstellen Sie die Webprojekte, dann das WebFacing-Projekt. Anschließend importieren Sie den Inhalt der zip-Dateien in die fünf Projekte.

## Zip-Dateien in Ihrem lokalen System extrahieren

Bevor Sie den Inhalt der zip-Dateien des Szenarios in die Workbench von Development Studio Client importieren können, müssen Sie sie in separaten Verzeichnissen extrahieren, damit Sie sie als Dateisysteme und nicht als zip-Dateien importieren können.

**Anmerkung:** Verwenden Sie ein anderes Programm als WinZip, führen Sie die Tasks für das entsprechende Programm aus.

1. Wechseln Sie auf Ihrem lokalen System in c:\wdsc\wdscsampl. Das Verzeichnis kann anders lauten, wenn Sie die Installation auf einem anderen Laufwerk oder in einem anderen Basisordner durchgeführt haben. Ist dies der Fall, wechseln Sie in dieses Verzeichnis und suchen dann \wdscsampl.
2. Klicken Sie die folgende zip-Datei doppelt an: SV000501.zip.
3. Klicken Sie den Knopf "Extrahieren" an und navigieren Sie zu dem Ordner, in dem sich die zip-Dateien befinden (standardmäßig c:\wdsc\wdscsampl).
4. Wählen Sie den Radioknopf **All Files** (Alle Dateien) aus und klicken Sie dann **Extract** (Extrahieren) an.
5. In Ihrem Verzeichnis wdscsampl sehen Sie, dass ein Ordner SV000501 mit allen erforderlichen Dateien erstellt wurde.
6. Sobald alle Dateien in c:\scenario\SV000501 extrahiert sind, löschen Sie die Datei SV000501.zip, damit Sie sie nicht versehentlich auswählen, wenn Sie Ihr Dateisystem importieren.
7. Wiederholen Sie die Schritte in diesem Abschnitt für die anderen vier zip-Dateien (SV000618.zip, SV001585.zip, SV001586.zip und SV000514.zip).

## Webprojekte erstellen

In diesem Abschnitt werden Sie Webprojekte für SV000514, SV000618, SV001585 und SV001586 erstellen. Die Schritte zum Erstellen jedes Projekts werden separat kurz beschrieben. Es sieht vielleicht so aus, als würden sich die Schritte wiederholen, es gibt jedoch einige kleinere, aber wichtige Unterschiede, die Sie beim Erstellen der vier Projekte beachten sollten (z. B. die Zuordnung der einzelnen EAR-Dateien zu den verschiedenen Projekten).

Um die Webprojekte zu erstellen, gehen Sie wie folgt vor:

1. Öffnen Sie IBM WebSphere Development Studio Client für iSeries vom Menü "Start" aus.
2. Öffnen Sie in Development Studio Client die Webperspektive über die Menüleiste, indem Sie **Fenster > Perspektive öffnen > Andere > Web** anklicken. Klicken Sie dann **OK** an.

3. Klicken Sie **Datei > Neu > Dynamisches Webprojekt** an.
  - a. Geben Sie in das Feld **Projektname** den Wert SV000514 ein.
  - b. Wählen Sie das Markierungsfeld **Configure advanced options** (Erweiterte Optionen konfigurieren) aus und klicken Sie **Weiter** an.
  - c. Klicken Sie den Knopf **Neu** neben dem Feld **EAR-Projekt** an.
  - d. Geben Sie in das Feld **Projektname** den Wert SVWholeSaleEAR ein und klicken Sie **Fertig stellen** in beiden Dialogfenstern an.
4. Klicken Sie **Datei > Neu > Dynamisches Webprojekt** an.
  - a. Geben Sie in das Feld **Projektname** den Wert SV001586 ein und klicken Sie **Weiter** an. (Das Markierungsfeld **Configure advanced options** (Erweiterte Optionen konfigurieren) sollte automatisch ausgewählt sein).
  - b. Stellen Sie sicher, dass SWholeSaleEAR im Dropdownfeld **EAR-Projekt** ausgewählt ist.
  - c. Klicken Sie **Fertig stellen** an.
5. Klicken Sie **Datei > Neu > Dynamisches Webprojekt** an.
  - a. Geben Sie in das Feld **Projektname** den Wert SV000618 ein und klicken Sie **Weiter** an.
  - b. Klicken Sie den Knopf **Neu** neben dem Feld **EAR-Projekt** an.
  - c. Geben Sie in das Feld **Neuer Projektname** den Wert SVStoreEAR ein und klicken Sie **Fertig stellen** in beiden Dialogfenstern an.
6. Klicken Sie **Datei > Neu > Dynamisches Webprojekt** an.
  - a. Geben Sie in das Feld **Projektname** den Wert SV001585 ein und klicken Sie **Weiter** an.
  - b. Stellen Sie sicher, dass SVStoreEAR im Dropdownfeld **EAR-Projekt** ausgewählt ist.
  - c. Klicken Sie **Fertig stellen** an.

## Webprojektdateien importieren

Um die Dateisysteme in Ihren Arbeitsbereich zu importieren, verwenden Sie die folgende Prozedur, die die Vorgehensweise beim Import der Dateien in SV000514 kurz beschreibt, als Richtlinie. Die Prozedur ist für die anderen drei Webprojektdateisysteme SV001586, SV001585 und SV000618 identisch. In dem darauf folgenden Abschnitt werden Erstellung und Import des WebFacing-Projekts erläutert.

Gehen Sie wie folgt vor, um die Dateien zu importieren:

1. Klicken Sie "Durchsuchen" an und navigieren Sie.
2. Klicken Sie den Ordner **SV000514** mit der rechten Maustaste an und wählen Sie **Import** aus.
3. Klicken Sie im Assistenten **Import** die Angabe **Dateisystem** an und klicken Sie dann **Weiter** an.
4. Wechseln Sie in das Verzeichnis, in das Sie die Dateien des Szenarios extrahiert haben (standardmäßig c:\wdsc\wdscsampl).
5. Klicken Sie **SV000514** und dann **OK** an.
6. Klicken Sie **Alles auswählen** an, um sicherzustellen, dass alle Komponenten des Dateisystems SV000501 ausgewählt werden.
7. Geben Sie in das Feld **Into Folder** (In Ordner) den Wert **SV000514** ein, wenn dieser noch nicht standardmäßig vorgegeben ist.

8. Wählen Sie das Markierungsfeld **Vorhandene Ressourcen ohne Warnung überschreiben** aus.
9. Klicken Sie **Fertig stellen** an und klicken Sie **Ja** an, wenn Sie Dialoge empfangen, die Sie zum Ändern des Kontextstamms auffordern.
10. Wiederholen Sie die Schritte 2 bis 9 für SV001586, SV001585 und SV000618. Stellen Sie dabei sicher, dass Sie die unterschiedlichen Projektnamen verwenden.

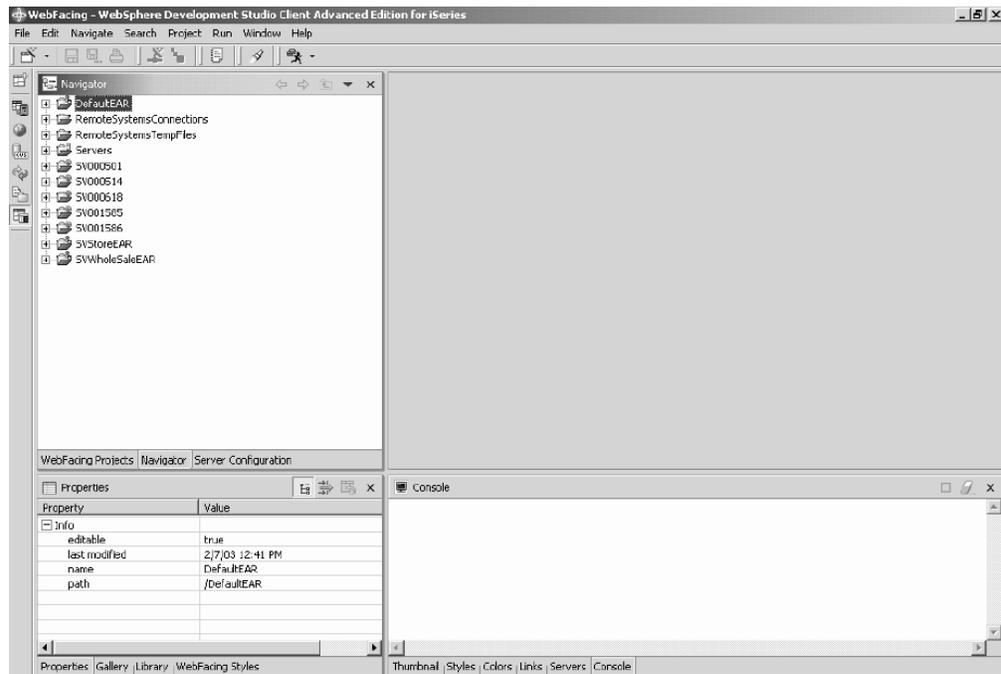
## **WebFacing-Projektdateien importieren und WebFacing-Projekt erstellen**

Projekt SV000501 unterscheidet sich von den vier soeben erstellten Webprojekten, da SV000501 ein WebFacing-Projekt ist. WebFacing-Projekte sind anders, da sie andere Dateien enthalten und eine eindeutige Struktur haben, die für die Bearbeitung von DDS-Dateien von Ihrem iSeries-Server spezifisch ist.

Gehen Sie wie folgt vor, um die Dateien zu importieren und ein WebFacing-Projekt zu erstellen:

1. Wird die Sicht "Projektnavigator" nicht angezeigt, klicken Sie die Registerkarte **Projektnavigator** an.
2. Klicken Sie mit der rechten Maustaste in der Sicht und wählen Sie **Import** aus.
3. Wählen Sie im Importfenster **WebFacing-Projekte** aus und klicken Sie **Weiter** an.
4. Wechseln Sie in das Verzeichnis, in das Sie die Dateien des Szenarios extrahiert haben (standardmäßig c:\wdsc\wdscsampl).
5. Wählen Sie **SV000501** aus und klicken Sie **OK** an. Klicken Sie **Weiter** an.
6. Wählen Sie im Bereich **Gefundene WebFacing-Projekte** des Fensters das Markierungsfeld **SV000501** aus.
7. Geben Sie SVStoreEAR in das Feld **Enterprise Application-Projekt (EAR)** ein.
8. Klicken Sie **Fertig stellen** an.

Nachdem Sie alle Dateien importiert haben, sollte Ihr Arbeitsbereich ähnlich wie in folgender Abbildung aussehen:



Werden Fehler neben Ihren importierten Projekten angezeigt (markiert durch rote X), klicken Sie jedes Projekt mit der rechten Maustaste an und wählen **Projekt erneut erstellen** aus. Abhängig vom Inhalt des Arbeitsbereichs sollten dadurch die meisten Fehler behoben werden, so dass die Anwendung ordnungsgemäß ausgeführt wird. Sind jetzt noch Fehler vorhanden, kann das daran liegen, dass die Migration des Codes auf Release 5.1 von Development Studio Client teilweise nicht vollständig ausgeführt wurde. Sie können mit den Übungen in diesem Kapitel fortfahren, um einen Einblick in das Produkt zu erhalten, oder Sie können mit dem nächsten Kapitel fortfahren, um die Anwendungen zu erstellen.

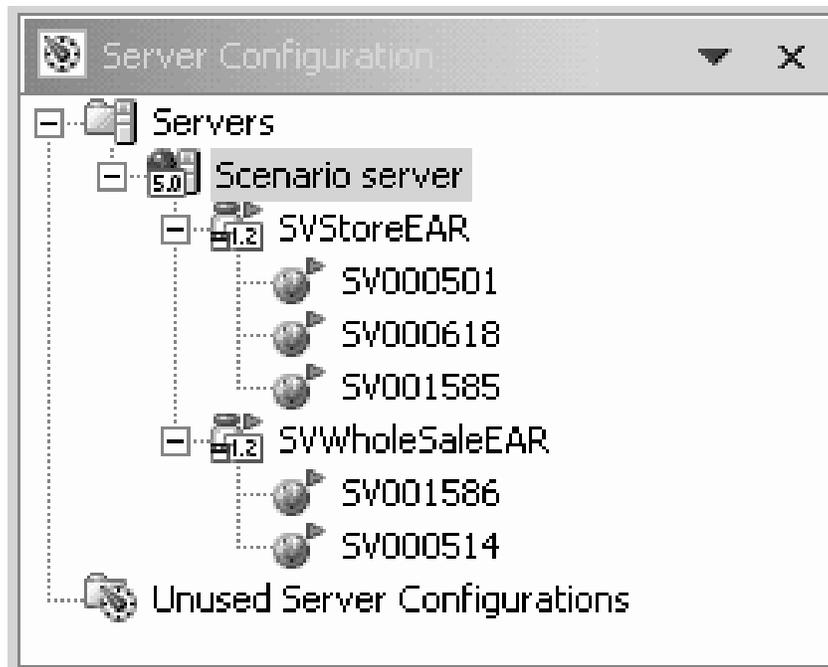
## Serverkonfiguration

Nachdem Sie alle Quellendateien importiert haben, müssen Sie den Server der WebSphere Test Environment so konfigurieren, dass er die Anwendungen SVWholeSaleEAR und SVStoreEAR erkennt.

Um den Server der WebSphere Test Environment zu konfigurieren, gehen Sie wie folgt vor:

1. Wechseln Sie in die Perspektive "Server". Klicken Sie in der Menüleiste der Workbench **Fenster > Perspektive öffnen > Andere > Server** und dann **OK** an.
2. Klicken Sie in der Sicht "Serverkonfiguration" mit der rechten Maustaste **Server** an und wählen Sie **Neu > Server und Serverkonfiguration** aus.
3. Geben Sie in das Feld **Servername** den Wert **Scenario server** ein.
4. Geben Sie in das Feld **Ordner** den Wert **Scenario folder** ein.
5. Erweitern Sie in dem Kästchen **Servertyp** die Angabe **WebSphere Version 5.0** und klicken Sie **Test Environment** an, sofern dieser Wert nicht bereits standardmäßig ausgewählt ist.
6. Klicken Sie **Fertig stellen** an und klicken Sie **Ja** an, wenn Sie Nachrichten empfangen.

7. Um der Serverkonfiguration EAR-Dateien hinzuzufügen, gehen Sie wie folgt vor:
  - a. Erweitern Sie **Server**.
  - b. Klicken Sie mit der rechten Maustaste **Scenario server** an und wählen Sie **Hinzufügen > SVWholeSaleEAR** aus.
  - c. Erweitern Sie **Server**.
  - d. Klicken Sie mit der rechten Maustaste erneut **Scenario server** an und wählen Sie **Add and Remove Projects** (Projekte hinzufügen und entfernen) aus.
  - e. Klicken Sie den Knopf **Add All** (Alles hinzufügen) an, um SVStoreEAR und SVWholeSaleEAR auf die rechte Seite des Fensters zu versetzen.
  - f. Klicken Sie **Fertig stellen** an.
8. Erweitern Sie in der Sicht "Serverkonfiguration" **Scenario server** und erweitern Sie dann **SVStoreEAR** und **SVWholeSaleEAR**, um eine Liste aller Ihrer Anwendungen anzuzeigen:



Gehen Sie wie folgt vor, um sicherzustellen, dass der Server die Projekte berücksichtigt hat:

1. Klicken Sie in der Sicht "Server", die sich rechts von der Sicht "Serverkonfiguration" befindet (klicken Sie die Registerkarte **Server** an, wenn die Sicht "Server" nicht angezeigt wird), mit der rechten Maustaste **Scenario server** an und wählen Sie **Publizieren** aus. Ein Dialogfenster **Publizieren**, das Ihnen den Fortschritt des Publikationsprozesses zeigt, wird aufgerufen.
2. Klicken Sie **OK** an, um Ihre Anwendungen in der WebSphere Application Server Test Environment zu publizieren.
3. Nachdem die Publikation beendet ist, klicken Sie in dem Dialogfenster **OK** an.
4. Klicken Sie in derselben Sicht mit der rechten Maustaste erneut **Scenario server** an und klicken Sie **Start** an. Die Sicht "Konsole" wird geöffnet und zeigt die Aktivitäten des Servers. Der Server ist gestartet, wenn die Zeile "Server *Servername* für e-business geöffnet" unten in der Sicht angezeigt wird. Blättern Sie vor, um alle Nachrichten anzuzeigen.

## WebFacing-Projekt konfigurieren

Bevor Sie die Anwendung ausführen oder mit dem Modul SV000501 arbeiten, müssen Sie den WebFacing-Server für die Anwendung, die ausgeführt werden soll, starten und das Projekt so konfigurieren, dass es den korrekten iSeries-Server verwendet.

Um den WebFacing-Server zu starten, gehen Sie wie folgt vor:

1. Öffnen Sie einen 5250-Green-Screen-Emulator und melden Sie sich mit Ihrer Benutzer-ID und Ihrem Kennwort an.
2. Geben Sie in die Befehlszeile `strtcpsvr *webfacing` ein.
3. Unten in der Anzeige sollte die Nachricht "WEBFACING-Server wird gestartet" angezeigt werden.

Zurück in der Workbench von Development Studio Client müssen Sie jetzt die WebFacing-Merkmale ändern:

1. Wechseln Sie in die Webperspektive. (Sie können zwischen Perspektiven wechseln, indem Sie die Symbole anklicken, die auf der linken Seite der Anzeige angeordnet sind.)
2. Klicken Sie die Registerkarte **Projektnavigator** an, damit Ihre Projektstrukturen angezeigt werden.
3. Erweitern Sie **SV000501** und klicken Sie **Web Deployment Descriptor** doppelt an.
4. Klicken Sie die Registerkarte **Parameter** an.
5. Klicken Sie **WFDefaultHost** an.
6. Löschen Sie im Feld **Wert** den vorhandenen Wert und geben Sie den Namen Ihres iSeries-Servers ein.
7. Klicken Sie in der Menüleiste der Workbench das Symbol für Sichern an oder klicken Sie **Datei > Web Deployment Descriptor sichern** an. Klicken Sie **OK** an, wenn Sie Nachrichten empfangen. Schließen Sie die Datei.

## Serverinformationen definieren

Jetzt müssen Sie die iSeries-Serverinformationen für alle fünf Projekte definieren, um sicherzustellen, dass sie für die Ausführung mit Ihrem iSeries-Server unter Ihrer Benutzer-ID und Ihrem Kennwort konfiguriert sind. Gehen Sie wie folgt vor, um Serverinformationen zu definieren:

1. Stellen Sie sicher, dass Sie sich in der Webperspektive befinden.
2. Klicken Sie in der Sicht "Projektnavigator" mit der rechten Maustaste **SV000501** an und wählen Sie **Laufzeitkonfiguration für iSeries-Webtools angeben** aus.
3. Löschen Sie im Feld **iSeries-Servername** den vorhandenen Wert und geben Sie den Namen des iSeries-Servers ein.
4. Löschen Sie die anderen vorhandenen Werte und geben Sie Ihre Benutzer-ID und Ihr Kennwort ein.
5. Klicken Sie **Fertig stellen** an (und klicken Sie, falls erforderlich, erneut **Fertig stellen** an).
6. Wiederholen Sie die Schritte 2 bis 5 für die anderen vier Projekte: SV000514, SV000618, SV001585 und SV001586.

---

## Anwendung in der Workbench ausführen

An diesem Punkt können Sie die Anwendung in der WebSphere Test Environment in der Workbench ausführen. Wenn Sie davon überzeugt sind, dass die Anwendung korrekt ausgeführt wird, können Sie die EAR-Datei auf den iSeries-Server exportieren und auf dem iSeries-WebSphere Application Server implementieren.

## Anwendung als Kunde in der Workbench ausführen

Als Kunde würden Sie auf der Webseite "Shopping" für das Einzelhandelsunternehmen beginnen, durch die Artikel blättern und dann eine Menge und eine Größe bestellen.

**Anmerkung:** Möglicherweise treten Probleme auf, wenn Sie versuchen, die Anwendung hinter einer Firewall auszuführen, da Ihre Datei web.xml nach der folgenden Datei sucht: `http://java.sun.com/j2ee/dtds/web-app_2_2.dtd`. Um dieses Problem zu beheben, ändern Sie vor dem Ausführen der Anwendung die Anweisung DOCTYPE in allen Ihren web.xml-Dateien wie folgt:

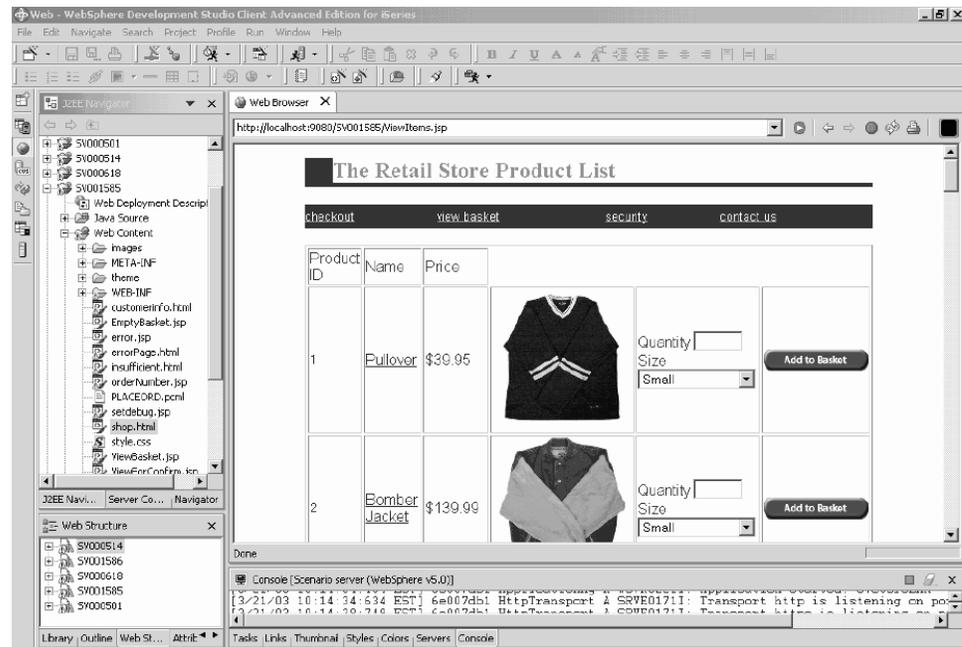
```
!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"x:/wdsc/eclipse/plugins/com.ibm.etools.j2ee/dtds/web-app_2_2.dtd"
```

Dabei ist `x:/wdsc` das Verzeichnis, in dem Sie das Produkt installiert haben. Normalerweise können Sie Ihre web.xml-Dateien unter *Projektname* > **Web Content** > **WEB-INF** finden. Nachdem Sie die Datei geöffnet haben, klicken Sie auf die Registerkarte **Quelle**, um die Datei direkt zu editieren.

Um die Anwendung als Kunde in der Workbench auszuführen, gehen Sie wie folgt vor:

1. Erweitern Sie in der Webperspektive **SV001585** > **Web Content**.
2. Klicken Sie mit der rechten Maustaste **shop.html** an und wählen Sie **Auf Server ausführen** aus. Damit wird die Anwendung im Workbench-Browser gestartet.
3. Klicken Sie **Fertig stellen** an.
4. Der Einstieg in die Anwendung erfolgt durch Anklicken der Abbildung für die T-Shirts.
5. Versuchen Sie, auf den folgenden Seiten Werte einzugeben, indem Sie vorgeben, ein Kunde zu sein, der einen Pullover oder einen Blouson (Bomber Jacket) bestellt, die Größe (size) und Menge (quantity) auswählt und seinem

Warenkorb (basket) Artikel hinzufügt:



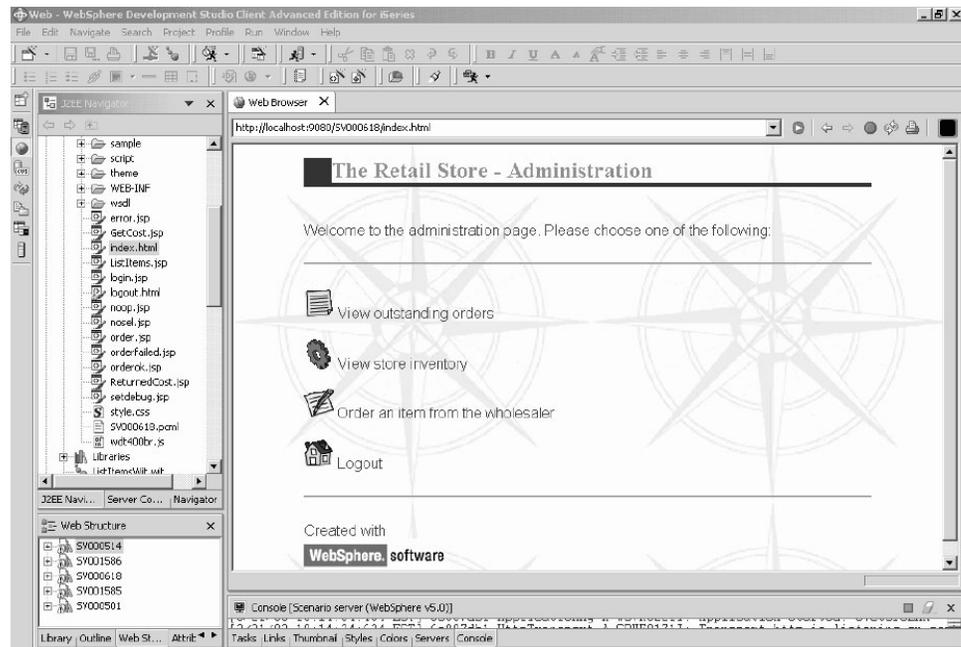
Die Seite, die den Text "your order has been processed" (Ihre Bestellung wurde verarbeitet) anzeigt, ist die letzte Seite des Musters. Sind Sie mit dem Muster zufrieden, können Sie den Browser schließen.

## Anwendung als Administrator in der Workbench ausführen

Als Administrator würden Sie auf der Webseite "Administration" für das Einzelhandelsunternehmen beginnen und Aufträge oder den Warenbestand prüfen. Um die Anwendung als Administrator in der Workbench auszuführen, gehen Sie wie folgt vor:

1. Erweitern Sie in der Webperspektive **SV000618 > Web Content**.
2. Klicken Sie mit der rechten Maustaste **index.html** an und wählen Sie **Auf Server ausführen** aus. Damit wird die Anwendung im Workbench-Browser gestartet.
3. Klicken Sie in Ihrer Rolle als Administrator, der entscheiden muss, welche Waren er für sein Unternehmen kaufen soll, das Symbol links neben **View store**

inventory (Bestand anzeigen) an, um die folgende Seite anzuzeigen:



Klicken Sie versuchsweise die Symbole an, um die Seiten für die Verwaltung anzuzeigen.

## Szenario-Anwendung in WebSphere Application Server implementieren

Nachdem Sie die iSeries-Szenario-Anwendung in der Test Environment der Workbench ausgeführt haben, können Sie die Anwendung jetzt wie bei der realen Ausführung der Anwendung in WebSphere Application Server implementieren. Bevor Sie die Anwendung implementieren, müssen Sie jedoch einige Anpassungen vornehmen, wie z. B. das Sichern der Seite für den Administrator und das Ändern des URL des Web-Service, damit die Anwendung auf die korrekte Position zeigt; dies wird in den nachfolgenden Abschnitten erläutert.

**Anmerkung:** Die Implementierung in WebSphere Application Server ist optional; Sie können auch mit dem nächsten Kapitel fortfahren und die Module ausführen, ohne die Anwendung in WebSphere Application Server zu testen.

## WebSphere Application Server konfigurieren

Weiter oben in diesem Kapitel wurde das Ausführen der Anwendung in der Workbench erläutert (siehe „Anwendung in der Workbench ausführen“ auf Seite 19). Nachdem Sie die Dateien auf dem iSeries-Server implementiert haben, können Sie jetzt die Anwendung auf Ihrer iSeries unter Verwendung von WebSphere Application Server ausführen.

Um die Anwendung in WebSphere Application Server zu implementieren (optional), müssen Sie Folgendes sicherstellen:

- Sie haben ein Exemplar von WebSphere Application Server Version 4.0, Version 5.0 oder WebSphere Application Server Express auf dem iSeries-Server erstellt oder Zugriff auf ein derartiges Exemplar und das Exemplar ist aktiv (nur, wenn die Implementierung in WebSphere Application Server getestet werden soll).

- Sie kennen die Portnummern für die HTTP- und WebSphere Application Server-Exemplare auf dem iSeries-Server.
- Auf Ihrer Workstation ist die WebSphere-Administrationskonsole Version 4.0 oder 5.0 installiert.

**Anmerkung:** Bei WAS Version 5.0 ist die Konsole browserbasiert, sodass Sie keine Konsole auf Ihrer Workstation benötigen. Weitere Informationen zu den verschiedenen Versionen von WAS finden Sie in der Onlinedokumentation.

Ihre webfähigen iSeries-Anwendungen verwenden WebSphere Application Server zur Ausführung der Java-Servlets und JavaServer Pages™ (JSPs), die die Kommunikation zwischen dem Browser des Webbenutzers und den iSeries-Programmen oder -Daten bewerkstelligen. Um Ihre HTML-Seiten und JSP-Dateien von demselben iSeries-System aus zu "bedienen", ist auf diesem System auch ein HTTP-Server erforderlich. Wir empfehlen die Verwendung des IBM HTTP-Servers (powered by Apache). Informationen zur Verwendung dieses Servers finden Sie in der Dokumentation im IBM HTTP Server for iSeries Documentation Center unter [http://publib.boulder.ibm.com/pubs/html/iseries\\_http/v5r1/index.htm](http://publib.boulder.ibm.com/pubs/html/iseries_http/v5r1/index.htm).

WebSphere Application Server führt die JavaServer Pages, JavaBeans und Java-Servlets aus, die für verschiedene Prozesse generiert werden. Die primären Dokumentationsressourcen für IBM WebSphere Application Server für iSeries und die IBM WebSphere-Administrationskonsole für iSeries sind an folgenden Websites verfügbar:

- IBM WebSphere Application Server Version 4.0 Advanced Edition for iSeries unter <http://publib.boulder.ibm.com/was400/40/AE/english/docs/>
- IBM WebSphere Application Server Version 4.0 Advanced Single Server Edition for iSeries unter <http://publib.boulder.ibm.com/was400/40/AEs/english/docs/>

Es wird unbedingt empfohlen, sich mit der Dokumentation zu IBM WebSphere Application Server vertraut zu machen, insbesondere mit den Abschnitten zu J2EE-Modulen (*J2EE modules*), zum Installieren von WebSphere Application Server (*Installing WebSphere Application Server*) und zum Konfigurieren mehrerer Exemplare von WebSphere Administrative Server (*Setting up multiple instances of the WebSphere administrative server*). Als Mindestvoraussetzung ist es erforderlich, die Schritte unter dem Link *Installation* auszuführen.

Verwenden Sie die Site-Übersicht, um Informationen zum Installieren, Konfigurieren und Anfordern der erforderlichen PTFs für die WebSphere-Administrationskonsole zu finden.

## Seite für den Administrator schützen

Da der Zugriff auf die Seite für den Administrator, `index.html`, nur für berechtigte Personen möglich sein sollte, muss die Seite korrekt geschützt werden. Dies kann über das Programm in der Webanwendungslogik oder mit Hilfe der Sicherheitseinrichtung von WebSphere erfolgen. In diesem Szenario wurde die Seite mit Hilfe der Sicherheitseinrichtung von WebSphere geschützt. Beachten Sie, dass dabei WebSphere Application Server V4.0 Advanced Edition verwendet wurde. Informationen zu WebSphere Application Server finden Sie auf den zugehörigen Websites:

- IBM WebSphere Application Server Version 4.0 Advanced Edition for iSeries unter <http://publib.boulder.ibm.com/was400/40/AE/english/docs/>

- IBM WebSphere Application Server Version 4.0 Advanced Single Server Edition for iSeries unter <http://publib.boulder.ibm.com/was400/40/AEs/english/docs/>
- RedBooks-Seite für WebSphere: Suchen Sie nach RedBooks für WebSphere Application Server Version 5.0 und WebSphere Application Server Express Edition Version 5.0 unter <http://publib.boulder.ibm.com/redbooks.nsf/portals/WebSphere>

Wenn Sie eine andere Version von WebSphere Application Server verwenden, lesen Sie die Dokumentation für diese Version, um Informationen zum Schützen von Webressourcen zu erhalten.

Sie können die Sicherheit für Webressourcen, wie Webseiten und Servlets, in Development Studio Client oder im Application Assembly Tool konfigurieren. Für dieses Szenario wird Development Studio Client verwendet.

Um die Sicherheitskonfiguration und die Merkmale für diese Webanwendung zu überprüfen, gehen Sie wie folgt vor:

1. Wechseln Sie in Development Studio Client in die Webperspektive.
2. Erweitern Sie in der Sicht "Navigator" **SV000618 > Web Content > WEB-INF**.
3. Klicken Sie **web.xml** doppelt an, um die web.xml-Sicht zu öffnen.
4. Klicken Sie die Registerkarte **Sicherheit** an.
5. Um index.html, die Administrationsseite, zu schützen, wurde auch eine Sicherheitseinschränkung definiert. Klicken Sie oben in der Sicht **Sicherheitseinschränkungen** an.
6. Klicken Sie das erste Exemplar von **SecurityConstraint** in der Liste an.
7. Klicken Sie auf der rechten Seite **AdminPage** an.
8. Klicken Sie **Editieren** an, um das Dialogfenster **Webressourcengruppen** aufzurufen. Beachten Sie, dass die Methoden GET und POST für index.html bereits im Voraus ausgewählt sind.
9. Klicken Sie **OK** an.

Beachten Sie im Abschnitt **Sicherheitsberechtigungsklassen** die definierte Sicherheitsberechtigungsklasse mit dem Namen "Administrator." Während der Implementierung werden Einzelpersonen dieser Berechtigungsklasse zugeordnet und erhalten damit Zugriff auf die Seite index.html. Beachten Sie, dass im Abschnitt **Berechtigte Berechtigungsklassen** der Berechtigungsklasse "Administrator" der Zugriff auf diese Sicherheitseinschränkung erteilt wurde. Ist diese Sicherheit aktiviert, wird nur Benutzern, die der Berechtigungsklasse "Administrator" zugeordnet sind, der Zugriff auf die Seite index.html erteilt, nachdem sie die korrekten Berechtigungsnachweise, wie Benutzer-ID und Kennwort, angegeben haben. Ist eine Ressource geschützt, versucht WebSphere Application Server zunächst, den Benutzer zu authentifizieren. Die Authentifizierung erfolgt anhand von Zertifikaten oder durch Aufforderung des Benutzers zur Eingabe einer Benutzer-ID und eines Kennworts. Die Aufforderung zur Eingabe dieser Angaben kann über den Basisauthentifizierungsdialog oder ein benutzerdefiniertes Formular erfolgen.

In diesem Szenario wurde eine eigene Anmeldeseite mit dem Namen login.jsp entworfen. Um die zugehörige Authentifizierungsaufforderung zu konfigurieren, wählen Sie die Registerkarte **Seiten** in der Sicht web.xml aus. Beachten Sie, dass im Abschnitt **Anmelden** bereits **Formular** als Authentifizierungsmethode im Voraus ausgewählt ist. Beachten Sie außerdem, dass der Name der Anmeldeseite

login.jsp lautet. Die Fehlerseite wird angezeigt, wenn die Anmeldung fehlschlägt. In diesem Fall zeigt die Anwendung erneut die Seite login.jsp an.

## EAR-Dateien für die iSeries WebSphere Application Server-Implementierung erstellen

Die Erstellung von EAR-Dateien ist für die Implementierung Ihrer Anwendung in iSeries WebSphere Application Server erforderlich. Eine EAR-Datei ist eine Standard-JAR-Datei (JAR = Java Archive) mit der Erweiterung .ear. EAR-Dateien können mehrere Webprojekte enthalten und werden zum Packen und Implementieren Ihrer Webanwendungen in WebSphere Administrative Server (WAS) verwendet.

Um die EAR-Dateien zu erstellen, gehen Sie wie folgt vor:

1. Wechseln Sie in die Webperspektive.
2. Klicken Sie in der Sicht "Navigator" mit der rechten Maustaste **SVStoreEAR** an und wählen Sie **Export** aus.
3. Klicken Sie im Fenster "Export" **EAR-Datei** und dann **Weiter** an.
4. Klicken Sie unter **Wohin sollen die Ressourcen exportiert werden?** die Schaltfläche **Durchsuchen** an und navigieren Sie zu einem Verzeichnis in Ihrem iSeries-Integrated File System, in dem Sie die EAR-Dateien speichern können. (Sie müssen Ihr Netzlaufwerk einem iSeries-IFS zugeordnet haben.)
5. Geben Sie **SVRetailStorEAR.ear** in das Feld **Dateiname** ein und klicken Sie **Öffnen** an.
6. Klicken Sie **Fertig stellen** an.
7. Wiederholen Sie die Schritte 2 bis 6 für **SVWholeSaleEAR**.

## EAR-Dateien in iSeries WebSphere Application Server implementieren

Nachdem Sie die EAR-Dateien erstellt haben, können Sie sie jetzt in WebSphere Application Server implementieren.

1. Öffnen Sie die WebSphere-Administrationskonsole.
2. Klicken Sie mit der rechten Maustaste **Enterprise Applications** an und wählen Sie **Enterprise Application installieren** aus.
3. Wählen Sie den Radioknopf **Anwendung installieren (\*ear)** aus.
4. Klicken Sie die obere Schaltfläche **Durchsuchen** an (die untere Schaltfläche ist nicht verfügbar).
5. Navigieren Sie zu dem IFS-Verzeichnis, in das Sie die EAR-Dateien exportiert haben.
6. Wählen Sie **SVRetailStorEAR.ear** aus.
7. Geben Sie "RETAILSTOR" in das Feld **Anwendungsname** ein.
8. Klicken Sie **Weiter** an und geben Sie Ihre Benutzer-ID für den iSeries-Server für die Berechtigungsklasse für Administratoren ein.
9. Klicken Sie wiederholt **Weiter** an, bis die Seite mit dem Namen **Virtuelle Hosts für Webmodule auswählen** angezeigt wird.
10. Klicken Sie für alle drei Webmodule **Virtuellen Host auswählen** an und wählen Sie Ihren bevorzugten virtuellen Server aus der Dropdownliste aus. (Sind Sie sich über die Auswahl nicht im Klaren, verwenden Sie **Standard** oder **Standard-Host**.)
11. Klicken Sie **Weiter** an.

12. Klicken Sie für alle drei Webmodule **Server auswählen** an und wählen Sie den gewünschten Server aus. (Sind Sie sich über die Auswahl nicht im Klaren, verwenden Sie **Standardserver**.)
13. Klicken Sie **Weiter** an.
14. Klicken Sie **Fertig stellen** an und klicken Sie im Dialogfenster **OK** an.
15. Klicken Sie mit der rechten Maustaste erneut **Enterprise Applications** an und wählen Sie **Enterprise Application installieren** aus.
16. Klicken Sie die untere Schaltfläche **Durchsuchen** an (die obere Schaltfläche ist nicht verfügbar).
17. Navigieren Sie zu dem IFS-Verzeichnis, in das Sie die EAR-Dateien gestellt haben.
18. Wählen Sie **SVWholesaleEAR.ear** aus.
19. Geben Sie "WHOLESALE" in das Feld **Anwendungsname** ein.
20. Klicken Sie wiederholt **Weiter** an, bis **Fertig stellen** aktiviert wird.
21. Klicken Sie **Fertig stellen** an und klicken Sie im Dialogfenster **OK** an.

**Anmerkung:** Es kann einige Minuten dauern, bis eine Bestätigungsnachricht angezeigt wird.

## **Anwendung in WebSphere Application Server ausführen**

Um in der Rolle des Kunden den Eingangspunkt für "Retailstore" auszuführen, geben Sie den folgenden URL in einen Web-Browser ein:

`http://Ihr_iSeries-Hostname:Ihre_HTTP-Portnummer/SV001585/shop.html`

Um in der Rolle des Administrators den Eingangspunkt für "WholeSale" auszuführen, geben Sie den folgenden URL in einen Web-Browser ein:

`http://Ihr_iSeries-Hostname:Ihre_HTTP-Portnummer/SV000618/index.html`

Ist Ihnen die HTTP-Portnummer nicht bekannt, wenden Sie sich an Ihren WebSphere Application Server-Administrator.



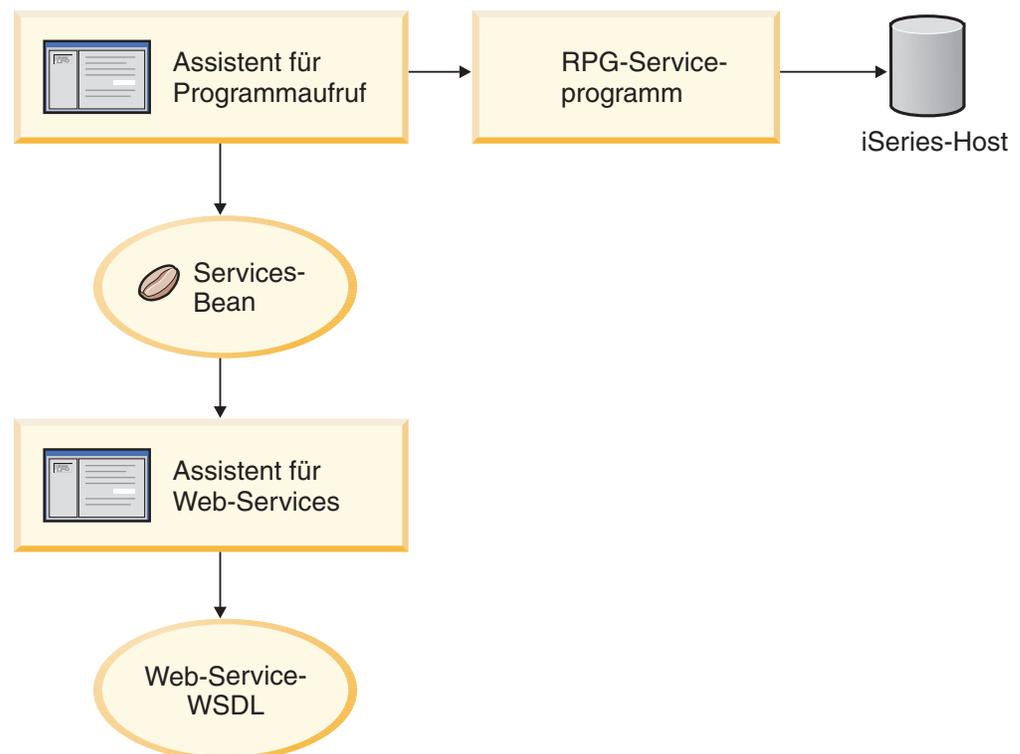
---

## Kapitel 4. Modul 1 mit Schrittfolgen: Erstellen eines Web-Service, um Produktpreise zurückzugeben (SV000514)

---

### Einführung

In diesem Modul erstellen Sie einen Web-Service aus einem RPG-Programm, das auf dem iSeries-Server vorhanden ist, um Produktpreise für Ihren Bestand anzuzeigen. Zunächst erstellen Sie ein RPG-Serviceprogramm mit einer Prozedur, die den Preis für einen Artikel anhand einer Artikelnummer aus der iSeries-Datenbank abrufen kann. Sie verwenden den Assistenten für Programmaufruf der iSeries Java Development Tools, um das RPG-Programm aufzurufen, und erstellen eine Services-Bean. Sie können dann mit Hilfe des Assistenten für Web-Services einen Web-Service erstellen und den Web-Service mit Hilfe des generierten Musters überprüfen.



---

### Vorbereitende Schritte

Sie können die Übungen nur dann vollständig ausführen, wenn die folgenden Voraussetzungen erfüllt sind. Die Voraussetzungen sind ausführlich in Kapitel 3, „Szenario ausführen“, auf Seite 9 erläutert.

- Sie verfügen über TCP/IP-Zugriff auf einen iSeries-Server.
- Sie haben die aktuellen PTFs für den iSeries-Server angelegt. Informationen zu PTFs und Service-Packs finden Sie auf unserer Unterstützungsseite: <http://www.ibm.com/software/awdtools/wdt400/support/>
- Sie haben die iSeries-Server mit dem Befehl STRTCPSVR \*ALL gestartet.

- Sie haben den WebFacing-Server mit dem Befehl STRTCPSVR \*WEBFACING gestartet.
- Sie haben alle Tasks in Kapitel 3, „Szenario ausführen“, auf Seite 9 vollständig ausgeführt (mit Ausnahme der optionalen WebSphere Application Server-Tasks, die zum Testen der Anwendung in der Workbench nicht erforderlich sind).

---

## Neues Webprojekt erstellen

Der erste Schritt beim Erstellen dieses Web-Service ist das Erstellen eines neuen Webprojekts zum Aufnehmen Ihrer Daten.

1. Wechseln Sie von der Workbench-IDE aus in die Webperspektive oder öffnen Sie die Webperspektive, indem Sie **Fenster > Perspektive öffnen > Andere > Web > OK** anklicken.
2. Klicken Sie **Datei > Neu > Dynamisches Webprojekt** an.
3. Geben Sie in das Feld **Projektname** den Wert Project514 ein.
4. Wählen Sie das Markierungsfeld **Configure advanced options** (Erweiterte Optionen konfigurieren) aus und klicken Sie **Weiter** an.
5. Klicken Sie den Knopf **Neu** neben dem Feld **EAR-Projekt** an.
6. Geben Sie in das Feld **Projektname** den Wert Project514EAR ein und klicken Sie **Fertig stellen** in beiden Dialogfenstern an.

Sie sehen jetzt, dass die Projekte Project514 und Project514EAR Ihrem Arbeitsbereich in der Sicht "Projektnavigator" hinzugefügt wurden.

---

## iSeries-Serverinformationen definieren

Nachdem Sie das Webprojekt erstellt haben, müssen Sie definieren, welchen iSeries-Server das Projekt zum Abrufen von Informationen verwendet.

1. Klicken Sie mit der rechten Maustaste Ihr Projekt **Project514** an und wählen Sie **Laufzeitkonfiguration für iSeries-Webtools angeben** aus.
2. Geben Sie den Namen Ihres iSeries-Servers ein, auf dem die zurückgeschriebene Bibliothek WHOLESale gespeichert ist, z. B. PR0D400.
3. Geben Sie Ihre Benutzer-ID und Ihr Kennwort für diesen iSeries-Server ein.
4. Geben Sie "Wholesale" in das Feld **Bibliothek** ein und klicken Sie **Hinzufügen** an.
5. Klicken Sie **Fertig stellen** an.

---

## RPG-Serviceprogramm erstellen

Sie möchten mit Ihrer Anwendung den Preis eines Artikels anhand einer bestimmten Artikelnummer abrufen. Dies erfolgt durch ein RPG-Serviceprogramm, das eine Prozedur mit dem Namen QryProdCost enthält. Die Bibliothek WHOLESale enthält ein RPG-Serviceprogramm mit dem Namen CWWSSRV. Dieses Programm enthält eine Prozedur QryProdCost, die die Artikelnummern als Eingabe akzeptieren kann, die Datei Inventory in der Bibliothek WHOLESale öffnet, den Preis aus der Bestandsdatenbank abrufen und den Preis zurückgibt. Um diese Task ausführen zu können, verfügt die Schnittstelle über zwei Parameter, einen für die Artikelnummer und den anderen für den Preis. Wird die Artikelnummer oder der Preis nicht gefunden, gibt das RPG-Programm eine Nachricht an die Schnittstelle zurück.

Um diesen Web-Service zu erstellen, erstellen Sie zunächst mit dem Assistenten für Programmaufruf eine JavaBean, um die RPG-Prozedur QryProdCost aufzurufen. Dann aktivieren Sie mit Hilfe von Web-Services die RPG-Prozedur als einen Web-Service über die JavaBean.

Um die JavaBean zu erstellen, gehen Sie wie folgt vor:

1. Klicken Sie in der Webperspektive mit der rechten Maustaste **Project514** an und wählen Sie **Neu > Andere** aus.
2. Wählen Sie **iSeries > Java** im linken Teilfenster des Fensters und dann **Programmaufruf-Bean** im rechten Teilfenster aus.
3. Klicken Sie **Weiter** an, um den Assistenten für Programmaufruf aufzurufen.
4. Geben Sie in das Feld **Java Bean-Name** unter dem Titel **Programm hinzufügen** den Wert **Inventory** ein.
5. Geben Sie in das Feld **Programmobjekt** den Wert **CWWSSRV** ein; dies ist der Name des RPG-Serviceprogramms.
6. Geben Sie in das Feld **Bibliothek** den Wert **WHOLESALE** ein.
7. Wählen Sie in der Dropdown-Liste **Programmtyp** den Wert **\*SRVPGM** aus.
8. Geben Sie in das Feld **Eingangspunkt** den Wert **QryProdCost** ein.
9. Klicken Sie **OK** an, um die Programmdefinition hinzuzufügen.

## Parameter erstellen und JavaBean generieren

Nachdem Sie das Programm angegeben haben, können Sie jetzt Parameter hinzufügen. Das RPG-Programm CWWSSRV enthält die folgenden zwei Parameter:

- Parameter für die Artikelnummer - Das Programm sucht anhand dieser Nummer in der Datenbank nach dem Artikel.
- Parameter für den Artikelpreis - Das Programm sucht anhand dieses Werts in der Datenbank nach dem Preis für den Artikel.

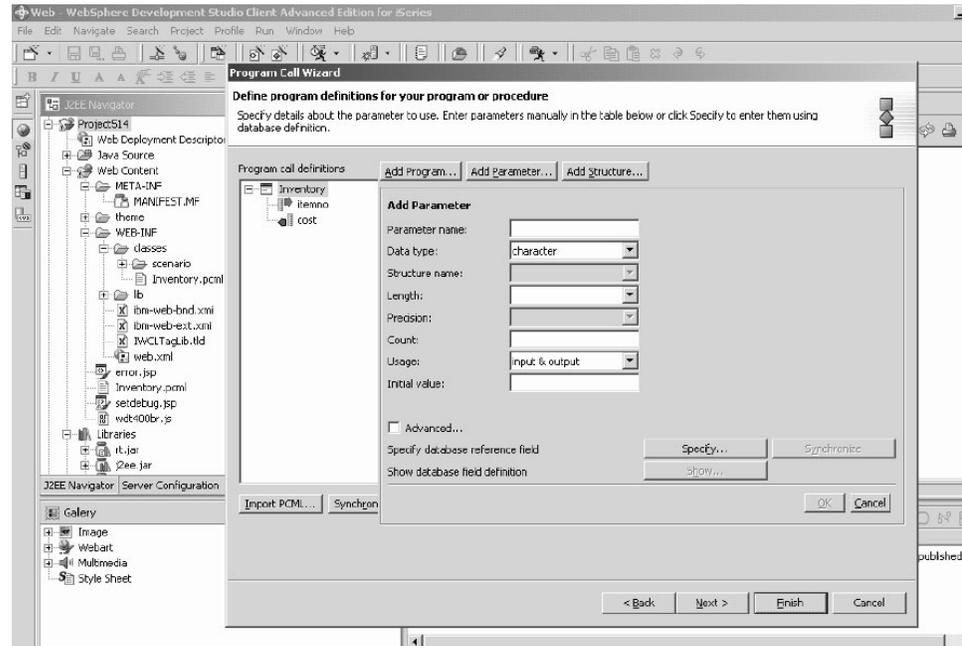
Um den Parameter für die Artikelnummer hinzuzufügen, gehen Sie wie folgt vor:

1. Klicken Sie im linken Fenster des Assistenten für Programmaufruf die Programmaufrufdefinition **Inventory** an, um diese auszuwählen. Mit dieser Aktion werden die Felder auf der rechten Seite der Seite erneut gefüllt.
2. Klicken Sie **Parameter hinzufügen** an.
3. Geben Sie in das Feld **Parametername** den Wert **itemno** (Artikelnummer) ein.
4. Wählen Sie in der Dropdown-Liste **Datentyp** den Wert **Gepackt dezimal** aus.
5. Geben Sie in das Feld **Länge** den Wert **5** ein.
6. Geben Sie in das Feld **Genauigkeit** den Wert **0** ein.
7. Wählen Sie in der Dropdown-Liste **Verwendung** den Wert **Eingabe** aus.
8. Klicken Sie **OK** an, um diesen Parameter hinzuzufügen. Beachten Sie, dass im linken Teilfenster **itemno** unterhalb von **Inventory** angezeigt wird. Sie können jetzt den zweiten Parameter hinzufügen.

Um den Parameter für den Artikelpreis einzugeben, gehen Sie wie folgt vor:

1. Klicken Sie im linken Fenster des Assistenten für Programmaufruf erneut die Programmaufrufdefinition **Inventory** an, um diese auszuwählen. Mit dieser Aktion werden die Felder auf der rechten Seite der Seite erneut gefüllt.
2. Klicken Sie **Parameter hinzufügen** an.
3. Geben Sie in das Feld **Parametername** den Wert **cost** (Preis) ein.
4. Wählen Sie in der Dropdown-Liste **Datentyp** den Wert **Gepackt dezimal** aus.

5. Geben Sie in das Feld **Länge** den Wert 7 ein.
6. Geben Sie in das Feld **Genauigkeit** den Wert 2 ein.
7. Wählen Sie in der Dropdown-Liste **Verwendung** den Wert **Ausgabe** aus.
8. Klicken Sie **OK** an, um diesen Parameter hinzuzufügen. Beachten Sie, dass im linken Teilfenster **cost** unterhalb von **Inventory** angezeigt wird. An diesem Punkt sollte der Assistent wie folgt aussehen. Beachten Sie, dass das Symbol links neben einem Parameter anzeigt, ob der Parameter den Typ "Eingabe", "Eingabe & Ausgabe" oder "Ausgabe" hat:



Um den Paketnamen für diese Parameter im Assistenten für Programmaufruf zu definieren, gehen Sie wie folgt vor:

1. Klicken Sie **Weiter** an.
2. Geben Sie in das Feld **Paket** den Wert **scenario** als Paketname ein. Lassen Sie die Standardwerte in den anderen Feldern unverändert.
3. Löschen den Inhalt des Markierungsfelds **Java-Anwendung**.

**Anmerkung:** Überprüfen Sie die Liste der Dateien unter "Diese Dateien werden von dem Assistenten generiert" und beachten Sie, dass der Name der generierten JavaBean **InventoryServices.java** lautet.

4. Klicken Sie **Fertig stellen** an, um die Dateien zu generieren.

## Web-Service aus der JavaBean erstellen

Der nächste Schritt nach dem Erstellen der JavaBean, die das RPG-Programm aufruft, ist die Konvertierung der JavaBean in einen Web-Service, sodass andere Programme über das Internet auf dasselbe RPG-Programm zugreifen können.

Beim Erstellen des Web-Service generiert ein Assistent für Web-Services die WSDL-Dateien, die an Benutzer verteilt werden, die den Web-Service verwenden müssen. Um den Service zu erstellen, gehen Sie wie folgt vor:

1. Erweitern Sie in der Sicht "Navigator" der Webperspektive **Project514 > Java-Quelle > scenario**.

2. Klicken Sie mit der rechten Maustaste **InventoryServices.java** an und wählen Sie **Neu > Andere** aus.
3. Klicken Sie im Fenster "Neu" **Web-Services** an, klicken Sie auf der rechten Seite des Fensters **Web-Service** und dann **Weiter** an.
4. Wählen Sie die Markierungsfelder **Proxy generieren** und **Generierten Proxy testen** aus.
5. Klicken Sie **Weiter** dreimal an und beobachten Sie die Generierung der erforderlichen Dateien. Wenn das Fenster für die Identität der Web-Service-JavaBean angezeigt wird, klicken Sie **Fertig stellen** an. Werden Fehler angezeigt, müssen Sie unter Umständen **Fertig stellen** noch einmal anklicken. Wenn Sie den Assistenten nicht beenden können, klicken Sie lediglich **Abbrechen** an. Die meisten erforderlichen Dateien werden trotzdem erstellt.

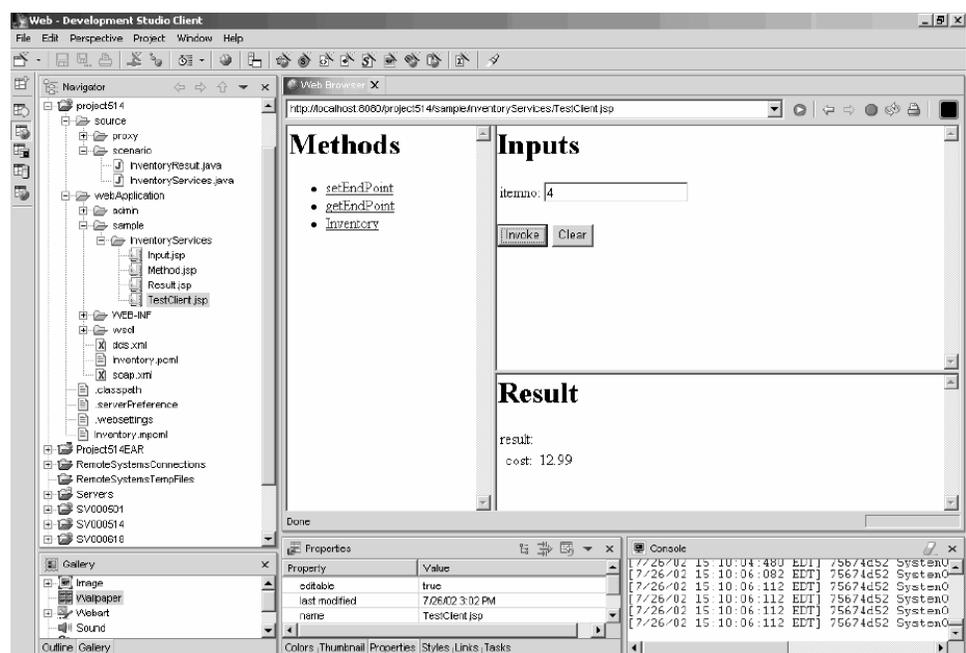
## Muster testen

Als Sie den Web-Service erstellten, war eine der Anweisungen die Anforderung, ein Muster zu generieren. Da diese Option ausgewählt wurde, erstellte der Assistent für Web-Services ein neues Projekt mit Testseiten, mit deren Hilfe Sie den Web-Service testen können. Beachten Sie, dass in der Sicht **Navigator** dieser neue Projektname **Project514Client** lautet. Beachten Sie außerdem, dass das Tool automatisch **TestClient.jsp** auf der rechten Seite der Workbench geöffnet hat. Diese Datei befindet sich unter **Project514Client > Web Content > sample > InventoryServices**.

Um das Muster zu testen, gehen Sie wie folgt vor:

1. Blättern Sie auf der geladenen Seite ganz bis zum Ende des Teilfensters "Methoden". Klicken Sie die Methode **Inventory (java.math.BigDecimal)** an.
2. Geben Sie in das Feld **itemno** den Wert 4 ein und klicken Sie **Invoke** an.
3. Überprüfen Sie, ob die zurückgegebene Ausgabe wie folgt aussieht:

```
result:
cost: 12.99
```

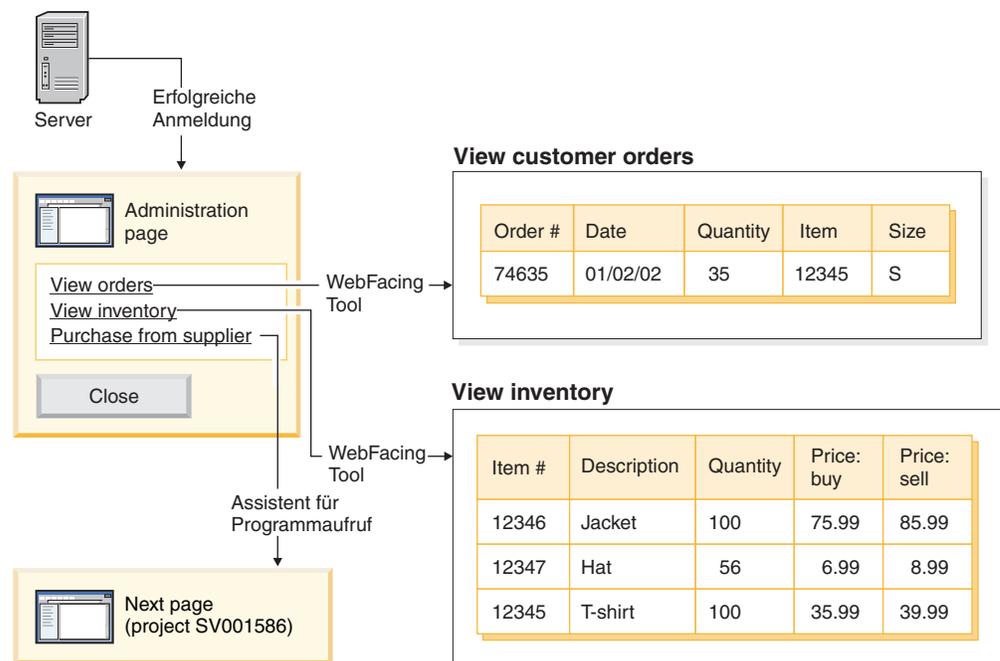


Nachdem Sie das Ergebnis bestätigt haben, ist die Arbeit mit dem Modul abgeschlossen! Sie haben einen Web-Service erstellt, der Produktpreise in der Anwendungsschnittstelle anzeigt.

# Kapitel 5. Modul 2 mit Schrittfolgen: Erstellen einer Schnittstelle, um den Bestand anzuzeigen und Artikel zu bestellen (SV000501)

## Einführung

In diesem Modul erstellen Sie eine Schnittstelle, die die Operationsoptionen eines Administrators zeigt, wenn das IBM WebFacing Tool verwendet wird. Nach der erfolgreichen Anmeldung am iSeries-Server zeigt die Schnittstelle den verfügbaren Bestand an und Sie können vorhandene Bestellungen überprüfen. Als Administrator können Sie dann Artikel von dem Großhandelsunternehmen kaufen.



In diesem Projekt arbeiten Sie mit zwei Programmen und zwei Anzeigedateien, die auf Ihren iSeries-Server zurückgeschrieben wurden. Die Programmnamen lauten ViewInventory und ViewOrder. Die Programme verwenden die folgenden zwei Anzeigedateien: ORDERDSP und QUERY. Diese zwei Dateien enthalten Webeinstellungen, die für die vom WebFacing Tool generierten JSP-Dateien angepasst wurden und für Images und Hyperlinks verwendet werden. Die Webeinstellung für Images erlaubt Ihnen die Verwendung des Inhalts eines Felds zum Generieren des Imagedateinamens und zum Anzeigen des Images in der JSP-Datei. Die Webeinstellung für Hyperlinks erlaubt Ihnen das Aufrufen einer anderen Webanwendung, wenn Sie das Image in der JSP-Datei anklicken. Sie können den Quellcode der Anzeigedatei entweder mit dem CODE-Editor oder mit CODE Designer prüfen, um festzustellen, wie die Webeinstellungen anzugeben sind.

---

## Vorbereitende Schritte

Sie können die Übungen nur dann vollständig ausführen, wenn die folgenden Voraussetzungen erfüllt sind. Die Voraussetzungen sind ausführlich in Kapitel 3, „Szenario ausführen“, auf Seite 9 erläutert.

- Sie verfügen über TCP/IP-Zugriff auf einen iSeries-Server.
- Sie haben die iSeries-Server mit dem Befehl STRTCPSVR \*ALL gestartet.
- Sie haben den WebFacing-Server mit dem Befehl STRTCPSVR \*WEBFACING gestartet.
- Sie haben alle Tasks in Kapitel 3, „Szenario ausführen“, auf Seite 9 vollständig ausgeführt (mit Ausnahme der optionalen WebSphere Application Server-Tasks, die zum Testen der Anwendung in der Workbench nicht erforderlich sind).

---

## WebFacing-Projekt erstellen

Der erste Schritt besteht im Erstellen eines WebFacing-Projekts und im Angeben der relevanten CL-Befehle. Um das WebFacing-Projekt zu erstellen, gehen Sie wie folgt vor:

1. Wechseln Sie in der Workbench in die WebFacing-Perspektive, indem Sie eines der Symbole für die Perspektiven anklicken, die entlang der linken Seite des Arbeitsbereichs angeordnet sind, oder indem Sie **Fenster > Perspektive öffnen > Andere > WebFacing** und dann **OK** anklicken.
  2. Erstellen Sie ein neues WebFacing-Projekt, indem Sie **Datei > Neu > WebFacing-Projekt** anklicken.
  3. Ordnen Sie dem Projekt den Namen Project501 zu.
  4. Klicken Sie den Radioknopf **Vorhanden** neben **Enterprise Application-Projekt** an.
  5. Geben Sie den Wert SVStoreEAR in das Feld **Name des vorhandenen Projekts** ein (Sie müssen die Groß-/Kleinschreibung beachten). Klicken Sie **Weiter** an.
  6. Wählen Sie in der Dropdownliste **J2EE-Ebene** den Wert **1.3** aus. Klicken Sie **Weiter** an.
  7. Im Feld **Verbindung** sollte der Name Ihres iSeries-Servers automatisch angezeigt werden. Ist dies nicht der Fall, klicken Sie "Neu" an und geben den Namen Ihres iSeries-Servers in den Dialog ein. Klicken Sie dann **Fertig stellen** an.
  8. Klicken Sie **DDS-Liste aktualisieren** an und geben Sie Ihr Kennwort und Ihre Benutzer-ID in das Dialogfenster ein, um die Liste zu aktualisieren.
  9. Erweitern Sie in der generierten Liste der Bibliotheken **RETAILSTOR**.
  10. Klicken Sie **QDDSSRC** und dann den Rechtspfeil (>>) an, um die Dateien zu versetzen.
  11. Klicken Sie zweimal **Weiter** an, um die Seite **CL-Befehle angeben** aufzurufen. Jetzt müssen Sie einen Befehl und eine Befehlsbezeichnung für den Administrator hinzufügen:
  12. Geben Sie in das Feld **CL-Befehl** call viewinvent ein.
  13. Löschen Sie in dem Feld **Befehlsbezeichnung** den vorhandenen Wert und geben Sie View inventory ein.
  14. Wählen Sie den Radioknopf **Anmeldung mit angegebenen Werten** aus.
  15. Klicken Sie **Hinzufügen** an und beachten Sie die Hinzufügung in der Liste unten im Fenster.
- Jetzt müssen Sie einen zweiten Befehl und eine zweite Befehlsbezeichnung für den Kunden hinzufügen:

16. Löschen Sie im Feld **CL-Befehl** den vorhandenen Wert und geben Sie call vieworder ein.
17. Löschen Sie im Feld **Befehlsbezeichnung** den vorhandenen Wert und geben Sie View orders ein.
18. Wählen Sie den Radioknopf **Bedienerführung für Anmeldung** aus.
19. Klicken Sie **Hinzufügen** an und beachten Sie die Hinzufügung in der Liste unten im Fenster. Klicken Sie **Weiter** an.
20. Blättern Sie im Fenster **Webdarstellung auswählen** durch die verfügbaren Darstellungen. Da später aus dieser Darstellung Informationen abgerufen werden müssen, wählen Sie für dieses Modul **avenue** aus. Klicken Sie **Weiter** an.
21. Stellen Sie sicher, dass das Markierungsfeld **Nein. Ich möchte jetzt nur das Projekt erstellen** ausgewählt ist, und klicken Sie **Fertig stellen** an.
22. Wird die Nachricht "Serverkonfiguration korrigieren" angezeigt, klicken Sie **OK** an. Diese Nachricht dient nur als Bestätigung, dass das Projekt der Datei SVStoreEAR hinzugefügt werden soll.

Da Sie Project501 als Spiegel des gesamten Projekts SV000501 erstellen, müssen Sie jetzt einige der Imagedateien aus SV000501 in Project501 kopieren, damit Project501 korrekt angezeigt wird. (Falls gewünscht, können Sie jedoch eigene Images für die Schnittstelle hinzufügen.) Um die Images zu kopieren, gehen Sie wie folgt vor:

1. Wechseln Sie in die Webperspektive.
2. Erweitern Sie in der Sicht "Navigator" **SV000501 > Web Content > images**.
3. (Optional) Erweitern Sie **generated**, um alle Images, die Sie in das Projekt importieren, anzuzeigen.
4. Klicken Sie mit der rechten Maustaste **generated** an und wählen Sie **Kopieren** aus.
5. Blättern Sie in der Sicht "Navigator" nach oben, bis wieder **Project501** angezeigt wird.
6. Erweitern Sie **Project501 > Web Content**.
7. Klicken Sie mit der rechten Maustaste den Ordner **images** an und wählen Sie im Kontextmenü **Einfügen** aus.
8. Jetzt ist der Ordner **generated** für Project501 vorhanden. Erweitern Sie **generated**, um die hinzugefügten Images anzuzeigen.

---

## DDS-Quelle konvertieren

Nachdem Sie das Projekt erstellt haben, können Sie jetzt die DDS-Anzeigedateien für Ihre Webseite in JSP-Dateien konvertieren. Beim Konvertieren Ihrer DDS-Anzeigedateien generiert das WebFacing Tool JSPs und XML-Dateien, die den DDS-Code ersetzen und den Webzugriff ermöglichen. Die generierten Dateien enthalten die Daten für das Satzformat oder steuern sein Aussehen, zeigen die Webversion der Anzeige an, fordern zur Eingabe von Daten auf und behandeln Eingabefehler. Der Assistent generiert außerdem eine Homepage für die Anwendung zum Starten der webfähigen Version Ihres Programms.

Zunächst müssen Sie jedoch Ihre Quelle so ändern, dass sie auf den korrekten iSeries-Server verweist. Genauer gesagt müssen Sie eine Änderung in einer Ihrer DDS-Anzeigedateien durchführen, damit der Link für die Anwendung korrekt funktioniert. Sie müssen den Standardnamen, SV000501, durch den Namen Ihres Projekts, Project501, ersetzen:

1. Kehren Sie zur WebFacing-Perspektive zurück und rufen Sie die Sicht "WebFacing-Projekte" auf, wenn diese noch nicht geöffnet ist.
2. Erweitern Sie **Project501 > DDS**.
3. Klicken Sie den zweiten Eintrag,   
<i>Seriesserver>**RETAILSTOR/QDDSSRC(QUERY)**, doppelt an.
4. Blättern Sie vor (ungefähr bis zu Zeile 16), bis Sie folgende Zeile sehen:  
A\*%WB 12 FLD 1 next ('/SV000501/DetailPage.do?PRODNO=&{PRODNO}')
5. Löschen Sie SV000501 und geben Sie Project501 ein.
6. Sichern und schließen Sie die Datei.

**Anmerkung:** Möchten Sie irgendwann einmal Projekt SV000501 erneut ausführen, müssen Sie den Wert Project501 wieder zurück in SV000501 ändern, da diese Datei auf dem iSeries-Server vorhanden ist und beide Projekte auf diese Datei zugreifen.

Um die DDS-Quelle zu konvertieren, gehen Sie wie folgt vor:

1. Klicken Sie in der WebFacing-Perspektive die Registerkarte **WebFacing-Projekte** an, um die Sicht "WebFacing-Projekte" aufzurufen, wenn diese noch nicht geöffnet ist.
2. Erweitern Sie **Project501 > DDS**.
3. Wählen Sie *iSeriesserver* > **RETAILSTOR/QDDSSRC(ORDERDSP)** und *iSeriesserver* > **RETAILSTOR/QDDSSRC(QUERY)** aus, indem Sie die erste Angabe anklicken und dann die Umschalttaste gedrückt halten, während Sie die zweite Angabe anklicken.
4. Klicken Sie mit der rechten Maustaste **Konvertieren** an, um die Konvertierung zu starten. Unter Umständen müssen Sie Ihre Benutzer-ID und Ihr Kennwort eingeben, falls die Verbindung zu Ihrem iSeries-Server irgendwann unterbrochen wurde.

## UTF-8-Unterstützung in der Workbench konfigurieren — Nur für Benutzer von WAS Version 4.0

IBM WebFacing Tool-Anwendungen unterstützen die Anzeige mehrerer Sprachen in einer Anzeige. Da Sprachen unterschiedliche Zeichensätze verwenden, sind Datenströme zwischen dem Browser und dem WebSphere Application Server in UTF-8 codiert. Um die korrekte Funktionsweise von IBM WebFacing Tool sicherzustellen, müssen Sie die UTF-8-Unterstützung in der Merkmaldatei der Anwendung in der Workbench konfigurieren.

**Anmerkung:** Dieser Abschnitt gilt nur für Benutzer von WAS 4.0. Bei WAS Version 5.0 wird diese Task automatisch ausgeführt.

Um die UTF-8-Unterstützung zu konfigurieren, gehen Sie wie folgt vor:

1. Wechseln Sie in die Webperspektive. (Sie können zwischen Perspektiven wechseln, indem Sie die Symbole anklicken, die auf der linken Seite der Anzeige angeordnet sind.)
2. Klicken Sie die Registerkarte **Projektnavigator** an, damit Ihre Projektstrukturen angezeigt werden.
3. Erweitern Sie **Project501** und klicken Sie **Web Deployment Descriptor** doppelt an.
4. Klicken Sie die Registerkarte **Umgebung** an.
5. Klicken Sie die Schaltfläche **Hinzufügen** an.

6. Wechseln Sie in die Perspektive "Server", indem Sie das zugehörige Symbol in der Taskleiste oben links anklicken, oder klicken Sie in der Menüleiste **Perspektive > Öffnen > Andere > Server** und dann **OK** an.
7. Erweitern Sie in der Sicht "Navigator" den Ordner **Servers**.
8. Klicken Sie **defaultInstance.wsi** doppelt an, um die Datei im Standardeditor zu öffnen.
9. Klicken Sie die Registerkarte **Umgebung** an und klicken Sie die Schaltfläche **Hinzufügen** an.
10. Ein editierbarer Wert, "(Neue Variable)", wird angezeigt. Löschen Sie diese Standardzeichenfolge und geben Sie `client.encoding.override` ein.
11. Geben Sie in das Feld **Wert** UTF-8 ein.
12. Klicken Sie das Symbol für Sichern an oder klicken Sie **Datei > Web Deployment Descriptor sichern** an.

## UTF-8-Unterstützung für WebSphere Application Server konfigurieren

(Optional) Wenn Ihre iSeries-Anwendung in WebSphere Application Server implementiert werden soll, müssen Sie die UTF-8-Unterstützung sowohl in WebSphere Application Server als auch in der Workbench konfigurieren.

### UTF-8-Unterstützung in WebSphere Application Server 4.0 Advanced Edition konfigurieren

1. Starten Sie die WebSphere-Administrationskonsole.
2. Erweitern Sie das Symbol **Knoten** und erweitern Sie **Knotenname > Anwendungsserver > Standardserver**.
3. Wählen Sie die Registerkarte **JVM-Einstellungen** aus und klicken Sie die Schaltfläche **Erweiterte JVM-Einstellungen** an, um den Dialog **Erweiterte JVM-Einstellungen** zu öffnen.
4. Geben Sie Folgendes in das Feld **Befehlszeilenparameter** ein:  
`-Dclient.encoding.override=UTF-8`
5. Klicken Sie **OK** an und klicken Sie **Anwenden** auf der Registerkarte **JVM-Einstellungen** an.
6. Damit diese Änderung für Ihre WebSphere-Anwendungen wirksam wird, stoppen Sie den Standardserver und starten Sie ihn dann erneut. Um den Server zu stoppen, klicken Sie mit der rechten Maustaste **Standardserver** an und wählen Sie **Stoppen** aus. Nachdem dieser Prozess beendet ist, klicken Sie mit der rechten Maustaste **Standardserver** an und wählen Sie **Starten** aus.

### UTF-8-Unterstützung in WebSphere Application Server 4.0 Advanced Single Server Edition konfigurieren

1. Starten Sie die WebSphere-Administrationskonsole.
2. Erweitern Sie in der browserbasierten Administrationskonsole das Symbol **Knoten** und erweitern Sie **Knotenname > Anwendungsserver > Standardserver > Prozessdefinition > JVM-Einstellungen**.
3. Blättern Sie zum Abschnitt "Erweiterte Einstellungen" auf der Seite "JVM-Einstellungen" und klicken Sie den Link "Systemmerkmale" an. Die Seite "Systemmerkmale" wird angezeigt.
4. Klicken Sie **Neu** an, um ein neues Systemmerkmal hinzuzufügen.
5. Geben Sie in das Feld **Name** den Wert `client.encoding.override` ein.
6. Geben Sie in das Feld **Wert** UTF-8 ein.

7. Klicken Sie **OK** an. Empfangen Sie eine Nachricht **Konfiguration muss gesichert werden** mit einem Link oben auf der Seite **JVM-Einstellungen**, klicken Sie den Link an, um die Seite **Konfiguration sichern** aufzurufen. Wählen Sie **Sichern** aus und klicken Sie dann **OK** an.
8. Damit diese Änderung für Ihre WebSphere-Anwendungen wirksam wird, stoppen Sie den Anwendungsserver und starten Sie ihn dann erneut. Die Vorgehensweise zum Stoppen und Starten des Anwendungsservers kann abhängig von der Plattform, auf der WebSphere Application Server installiert ist, unterschiedlich sein. Die WebSphere Application Server-Dokumentation für Ihre Plattform enthält Informationen zum Stoppen und Starten des Anwendungsservers.

---

## Style-Sheet erstellen

Möchten Sie zusätzliche Seiten mit einem kaskadierenden Style-Sheet (CSS = Cascading Style Sheet) integrieren, müssen Sie entweder die Darstellung des WebFacing-Projekts oder das kaskadierende Style-Sheet anpassen, damit diese übereinstimmen. Nach der Anpassung des Style-Sheets können Sie mit Hilfe des Assistenten für Webinteraktion unter Verwendung des Style-Sheets eine detaillierte Webseite erstellen, um Informationen zu Artikeln, wie z. B. Preis und Farbe, anzuzeigen. Für das Modul wird das Style-Sheet `DetailPageResults.jsp` aus dem Projekt SV000501 eingegliedert. Die CSS-Datei kann jedoch für die spätere Bezugnahme manuell angepasst werden. (Dies wird kurz im nächsten Abschnitt erläutert.)

### (Optional) Kaskadierendes Style-Sheet manuell anpassen

Wie zuvor erwähnt, können Sie, statt das Style-Sheet `DetailPageResults.jsp` aus dem Projekt SV000501 für Project501 einzugliedern, Ihre eigenen Style-Sheets anpassen:

1. Wechseln Sie in die Sicht "Projektnavigator" und erweitern Sie **Project501 > Web Content > styles > apparea**.
2. Klicken Sie **apparea.css** doppelt an, um die Datei im CSS Designer zu öffnen. Dieses Style-Sheet steuert die Darstellung Ihres Anwendungsbereichs.
3. Ändern und sichern Sie die Datei.
4. Wenn wieder die Sicht "Projektnavigator" angezeigt wird, navigieren Sie zu **Project501 > Web Content > styles > chrome**.
5. Klicken Sie **avenue.css** doppelt an, um die Datei im CSS Designer zu öffnen. Dieses Style-Sheet steuert die Darstellung der gesamten Seite.
6. Ändern und sichern Sie die Datei.

---

## WebFacing-Projekt mit einer Webinteraktion erweitern und verbessern

In diesem Abschnitt werden Sie das WebFacing-Projekt verbessern. Wenn die Liste der Artikel angezeigt wird, möchten Sie die Abbildung eines Artikel anklicken können, um Einzelangaben zu diesem Artikel anzuzeigen. Dazu verwenden Sie den Assistenten für Webinteraktion, um ein RPG-Programm aufzurufen, das die Einzelangaben zu dem Artikel abrufen und auf einer anderen Webseite anzeigt. Sie werden die folgenden Schritte ausführen:

- Definieren der Serverinformationen
- Kopieren des korrekten Style-Sheets
- Erstellen der Interaktion
- Hinzufügen von Programmen und Parametern zu der Interaktion

- Ändern der Parameterverwendung

Zunächst müssen Sie die Serverinformationen definieren:

1. Wechseln Sie in die Webperspektive.
2. Klicken Sie in der Sicht "Navigator" mit der rechten Maustaste **Project501** an und wählen Sie **Laufzeitkonfiguration für iSeries-Webtools angeben** aus.
3. Geben Sie den Namen Ihres iSeries-Servers, Ihre Benutzer-ID und Ihr Kennwort ein.
4. Geben Sie Retailstor in das Feld **Bibliothek** ein und klicken Sie dann **Hinzufügen** an.
5. Klicken Sie **Fertig stellen** an. Klicken Sie, falls erforderlich, erneut **Fertig stellen** an.

Bevor Sie die Webinteraktion erstellen, müssen Sie die korrekten Darstellungen und Style-Sheets kopieren, damit Project501 das korrekte JSP-Dateiformat anzeigt. (Diese Task müssen Sie nicht ausführen, wenn Sie die Übung „(Optional) Kaskadierendes Style-Sheet manuell anpassen“ auf Seite 38 ausgeführt haben.) Zunächst kopieren Sie den Ordner styles und anschließend die DetailPageResults.jsp:

1. Erweitern Sie in der Sicht "Navigator" **SV000501 > Web Content**.
2. Klicken Sie mit der rechten Maustaste **Styles** an und wählen Sie **Kopieren** aus.
3. Blättern Sie in der Sicht "Navigator" zurück, bis Ihr Projekt **Project501** wieder angezeigt wird.
4. Erweitern Sie das Projekt, klicken Sie mit der rechten Maustaste **Web Content** an und wählen Sie im Kontextmenü **Einfügen** aus. Klicken Sie **Ja** an, um die vorhandenen Darstellungen zu überschreiben.
5. Blättern Sie wieder vor zu **SV000501 > Web Content**.
6. Klicken Sie mit der rechten Maustaste **DetailPageResults.jsp** an und wählen Sie **Kopieren** aus.
7. Blättern Sie wieder zurück, bis Ihr Projekt **Project501** wieder angezeigt wird.
8. Erweitern Sie das Projekt, klicken Sie mit der rechten Maustaste **Web Content** an und wählen Sie im Kontextmenü **Einfügen** aus.

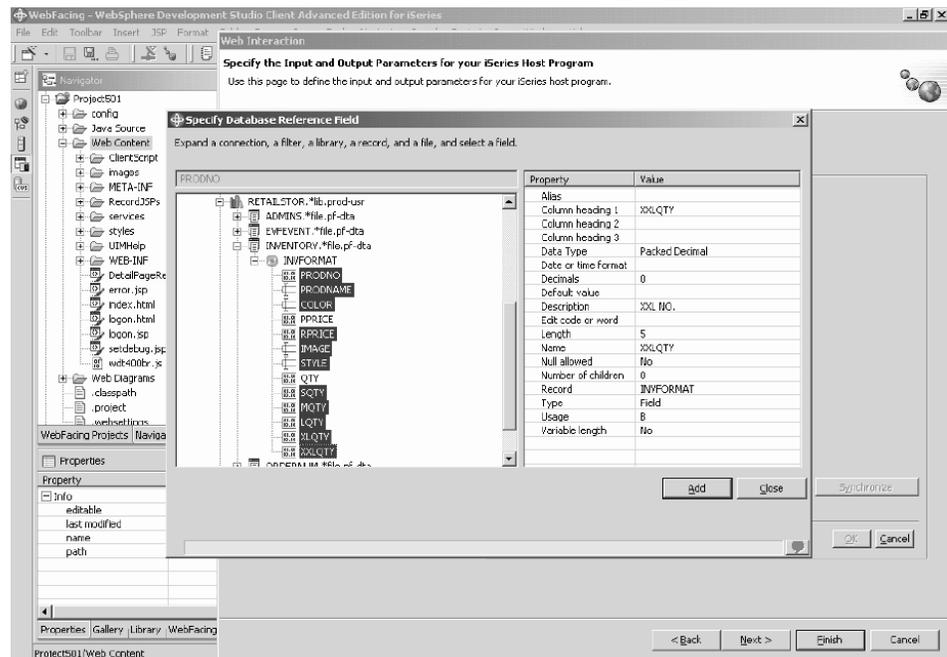
Beachten Sie die neuen Elemente, die unter dem Ordner Web Content für Project501 hinzugefügt wurden.

Sie können jetzt die Webinteraktion erstellen:

1. Klicken Sie **Datei > Neu > Andere** an.
2. Klicken Sie im Dialog **Neu** die Option **Web** auf der linken Seite an und klicken Sie dann **Webinteraktion** auf der rechten Seite an. Klicken Sie **Weiter** an.
3. Geben Sie in das Feld **Webinteraktionsname** den Wert **DetailPage** ein und klicken Sie **Weiter** an.
4. Wählen Sie den Radioknopf **Eingabe-JSP generieren** aus.
5. Wählen Sie den Radioknopf **Ausgabeseiten verwenden** aus (sofern er nicht bereits ausgewählt ist) und klicken Sie **Hinzufügen** an.
6. Erweitern Sie im Dialog **Ausgabe-JSP** den Ordner **Web Content**, wählen Sie **DetailPageResults.jsp**, Ihre vorformatierte Ausgabeseite, aus und klicken Sie **OK** an.
7. Klicken Sie **Weiter** an.

Jetzt können Sie Ihrer Interaktion Programme und Parameter hinzufügen. Sie müssen demselben Programm 11 Parameter hinzufügen. Statt jeden Parameter mit seinen individuellen Werten manuell hinzuzufügen, können Sie auch den folgenden ein wenig schnelleren Weg wählen:

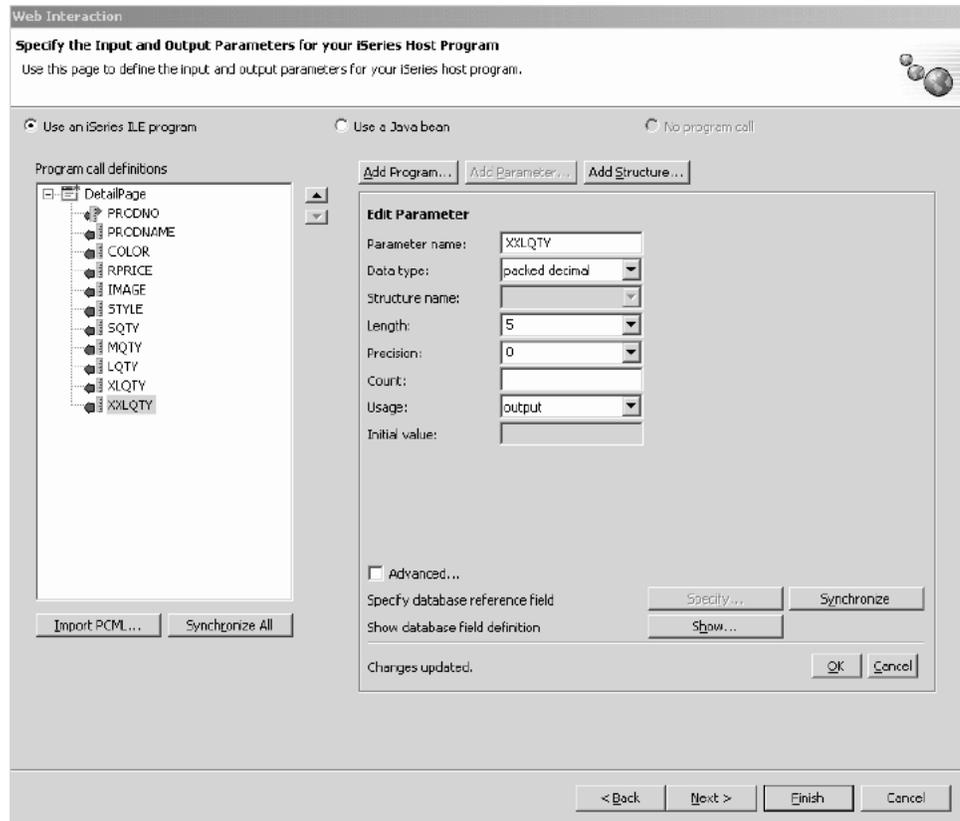
1. Wählen Sie **iSeries-ILE-Programm verwenden** aus, falls dies noch nicht bereits ausgewählt ist.
2. Klicken Sie **Programm hinzufügen** an.
3. Geben Sie im Feld **Programmaliasname** den Wert **DetailPage** ein.
4. Klicken Sie für das Feld **Programmobjekt** die Schaltfläche **Durchsuchen** an.
  - a. Erweitern Sie *iSeriesserver* > **\*LIBL** > **RETAILSTOR**.
  - b. Klicken Sie **DETAILPAGE.\*pgm.rpgle** an (die erste Datei unter **RETAILSTOR**) und klicken Sie **OK** an.
5. Wieder im Assistenten für Webinteraktion klicken Sie **OK** (unten rechts) an. Beachten Sie, dass **DetailPage** auf der linken Seite des Assistenten unter "Programmaufrufdefinitionen" hinzugefügt wurde.
6. Klicken Sie im Abschnitt "Programmaufrufdefinitionen" im linken Teilfenster des Assistenten **DetailPage** an, um sie auszuwählen.
7. Klicken Sie **Parameter hinzufügen** an.
8. Klicken Sie neben **Datenbankreferenzfeld angeben** unten im Assistenten **Angeben** an.
9. Erweitern Sie *iSeriesserver* > **\*LIBL** > **RETAILSTOR** > **INVENTORY.\*file.pf-dta** > **INVFORMAT**, um eine Liste mit 13 Parametern anzuzeigen. Sie müssen die folgenden 11 Parameter hinzufügen, indem Sie jeden einzelnen Parameter anklicken und **Hinzufügen** anklicken, oder indem Sie die Taste **STRG** gedrückt halten und jeden Parameter anklicken und dann **Hinzufügen** anklicken: **PRODNO**, **PRODNAME**, **COLOR**, **RPRICE**, **IMAGE**, **STYLE**, **SQTY**, **MQTY**, **LQTY**, **XLQTY**, **XXLQTY**. Im Grunde wählen Sie alle Felder außer **PPRICE** und **QTY** aus.



10. Klicken Sie **Schließen** an.

Jetzt müssen Sie die Verwendung für 10 der 11 Parameter in "Ausgabe" ändern (alle Parameter bis auf den Parameter PRODNO).

1. Während immer noch der Assistent für Webinteraktion aktiv ist, klicken Sie **PRODNAME** an, um diesen Parameter auszuwählen.
2. Ändern Sie im kombinierten Feld **Verwendung** den ausgewählten Wert in **Ausgabe** und klicken Sie **OK** an. Wiederholen Sie diesen Schritt (und den vorherigen Schritt) für alle Parameter bis auf den ersten (PRODNO). Nachdem Sie alle Parameter angepasst haben, sollte Ihr Arbeitsbereich wie nachfolgend gezeigt aussehen. Beachten Sie, dass das Symbol links neben einem Parameter anzeigt, ob der Parameter den Typ "Eingabe", "Eingabe & Ausgabe" oder "Ausgabe" hat.



3. Klicken Sie **Weiter** an, um die Voranzeige **Eingabeformular** anzuzeigen.
4. Klicken Sie **Fertig stellen** an, um die Webinteraktion zu erstellen.
5. Wählen Sie **Ja für alle** oder **OK** und erneut **Fertig stellen** aus, wenn Sie Nachrichten empfangen.

## Projekt mit einer Webinteraktion verlinken

Nachdem Sie eine Webinteraktions-JSP-Datei erstellt haben, die Eingabe- und Ausgabeparameter verwendet, müssen Sie jetzt die JSP-Datei anpassen, damit sie auch mit der Komponente WebFacing verwendet werden kann. Sie müssen Code eingeben, um einen Link von Ihrer WebFacing-Anwendung zu erstellen, um diese Webinteraktion aufzurufen. Dazu müssen Sie der Datei `webface.js` eine JavaScript-Funktion hinzufügen, sodass Sie das Servlet `DetailPageServlet` mit dem Parameter `PRODNO` in einem neuen Fenster aufrufen können.

Um den Link zu erstellen, gehen Sie wie folgt vor:

1. Erweitern Sie in der Sicht "Navigator" **Project501 > Web Content > ClientScript**.
2. Klicken Sie **webface.js** doppelt an, um die Datei in einem Editor zu öffnen.
3. Blättern Sie an das Ende der Datei und geben Sie die folgenden Zeilen ein:

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

```
4. Klicken Sie das Symbol für Sichern an oder klicken Sie **Datei > webface.js sichern** an.

Soll zu einem späteren Zeitpunkt eine ähnliche Anwendung wie SV000501 erstellt werden, müssen Sie auch eine Webeinstellung in Ihrer DDS-Quelle ändern, um das hinzugefügte Image zu aktivieren, und außerdem den Link zum Schließen des Fensters für die JavaScript-Funktion aktivieren.

Der RPG-Code in dieser Anwendung wird geändert, um die Änderung widerzuspiegeln, in zukünftigen Anwendungen müssten Sie die Änderung jedoch manuell durchführen. Nach dem Ändern der Webeinstellungen müssten Sie außerdem die DDS-Quelle erneut konvertieren.

Sie können den Code überprüfen, bevor das Ergebnis kopiert wird.

Um die DDS-Quelle anzuzeigen, gehen Sie wie folgt vor:

1. Wechseln Sie in die WebFacing-Perspektive.
2. Erweitern Sie in der Sicht "WebFacing-Projekte" **Project501 > DDS**.
3. Klicken Sie mit der rechten Maustaste **<iSeries-Server > RETAILSTOR/QDDSSRC(QUERY)** an und wählen Sie **Öffnen mit > CODE Designer** aus.
4. Nachdem der CODE Designer geöffnet wurde, erweitern Sie **SCREEN1 > ITEMSUB**.
5. Klicken Sie **IMAGESRC** an, um diesen Wert auszuwählen.
6. Klicken Sie die Registerkarte **Quelle** an.
7. Klicken Sie die Registerkarte **Webeinstellungen** unten rechts im Fenster an.

**Anmerkung:** Überprüfen Sie die Merkmale für die Webeinstellungen, wie z. B. die Breite in Pixel und den Dateinamen. In der Zukunft müssen Sie dieselben Änderungen an Ihrer DDS-Quelle durchführen und dann die Quelle erneut konvertieren.

8. Beachten Sie die folgenden Zeilen in der Quelle:

```
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}')
```

Beachten Sie insbesondere, dass Project501 in der letzten Zeile angegeben ist. Wie bereits erwähnt, müssen Sie den Wert Project501 zurück in SV000501 ändern, wenn Sie die Anwendung SV000501 erneut ausführen möchten.

9. Sichern Sie die Datei, indem Sie das Symbol für Sichern anklicken oder indem Sie **Datei > Sichern** in der Menüleiste anklicken.

10. Schließen Sie die Datei und CODE Designer.

Weitere Informationen zur Arbeit mit der DDS-Quelle enthält die Hilfeperspektive der Workbench sowie die Dokumentation zum IBM WebFacing Tool.

---

## Dateien publizieren und Server erneut starten

Dieser Abschnitt erläutert, wie Sie Ihre Dateien publizieren und den Server erneut starten, sodass Ihre Anwendung Project501 alle Änderungen berücksichtigen kann.

Bevor Sie den Server erneut starten, müssen Sie jedoch sicherstellen, dass Project501.war der Datei SVStoreEAR hinzugefügt wurde.

Um Ihre Konfiguration zu überprüfen, gehen Sie wie folgt vor:

1. Erweitern Sie in der Sicht "Navigator" den Ordner **Scenario**.

2. Klicken Sie **Scenario server.wsi** doppelt an.

Unter Umständen wird ein Dialog mit der Frage angezeigt, ob der Editor die Projekteinträge automatisch korrigieren soll. Klicken Sie **Ja** an. Wird kein derartiger Dialog angezeigt, ist keine Aktion erforderlich. Die Datei wird in jedem Fall automatisch geändert, sobald sie geöffnet wird.

3. Sichern und schließen Sie die Datei.

Um den Server erneut zu starten, gehen Sie wie folgt vor:

1. Wechseln Sie in die Perspektive "Server".

2. Erweitern Sie in der Sicht "Serverkonfiguration" unten links **Server** und klicken Sie **Scenario server** doppelt an.

3. Klicken Sie die Registerkarte **Server** unten rechts in der Anzeige an, um den Status Ihres Servers in der Sicht "Server" anzuzeigen.

4. Klicken Sie mit der rechten Maustaste **Scenario server** an und wählen Sie **Publizieren** aus. Klicken Sie **OK** an, wenn die Publikation abgeschlossen ist.

5. Klicken Sie mit der rechten Maustaste Ihren Server in der Sicht "Server" an und wählen Sie **Start** oder **Neustart** aus (je nachdem, welche Option aktiviert ist). Wird eine Bedienerführung zu nicht gesicherten Dateien angezeigt, brechen Sie den Dialog ab, sichern Sie alle offenen Dateien und fahren Sie wieder wie zuvor fort.

6. Prüfen Sie die Informationen im Konsolprotokoll (das automatisch geöffnet wird). Der Server ist gestartet, wenn die folgende Zeile unten im Protokoll angezeigt wird: Server *Servername* für e-business geöffnet.

---

## Schnittstelle testen

Sie haben jetzt die erforderlichen Schritte zum Erstellen einer Schnittstelle zum Anzeigen von Bestellungen und zum Anzeigen des Bestands beendet. Um Ihre Schnittstelle zu testen, gehen Sie wie folgt vor:

1. Wechseln Sie in die Webperspektive.

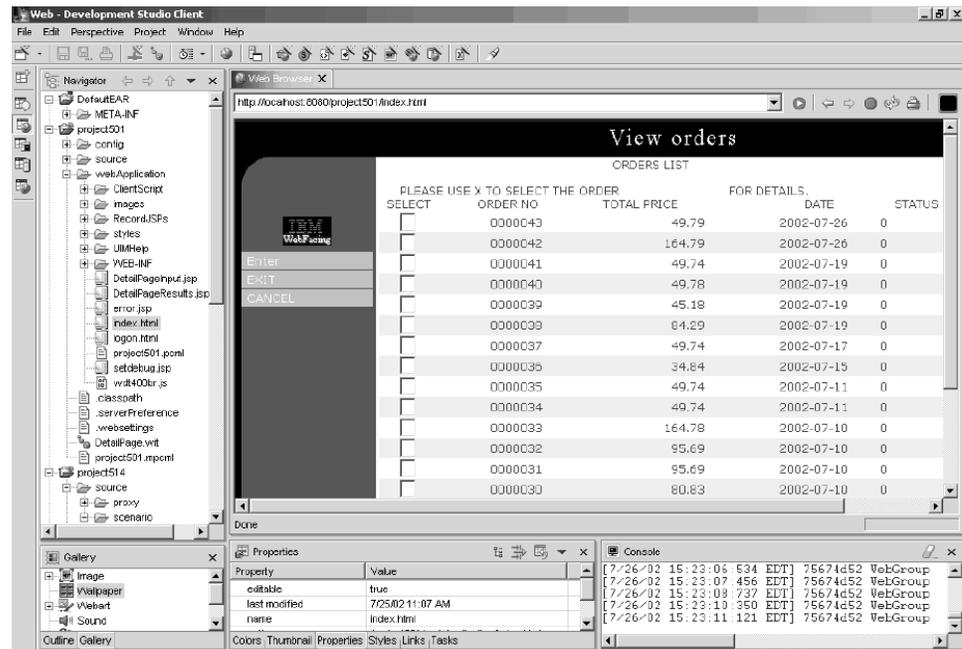
2. Erweitern Sie **Project501 > Web Content**.

3. Klicken Sie mit der rechten Maustaste **index.html** an und wählen Sie **Auf Server ausführen** aus.

4. Ein Dialog **Serverauswahl** wird geöffnet. Überprüfen Sie, ob **Scenario server** standardmäßig ausgewählt ist, und klicken Sie **Fertig stellen** an.

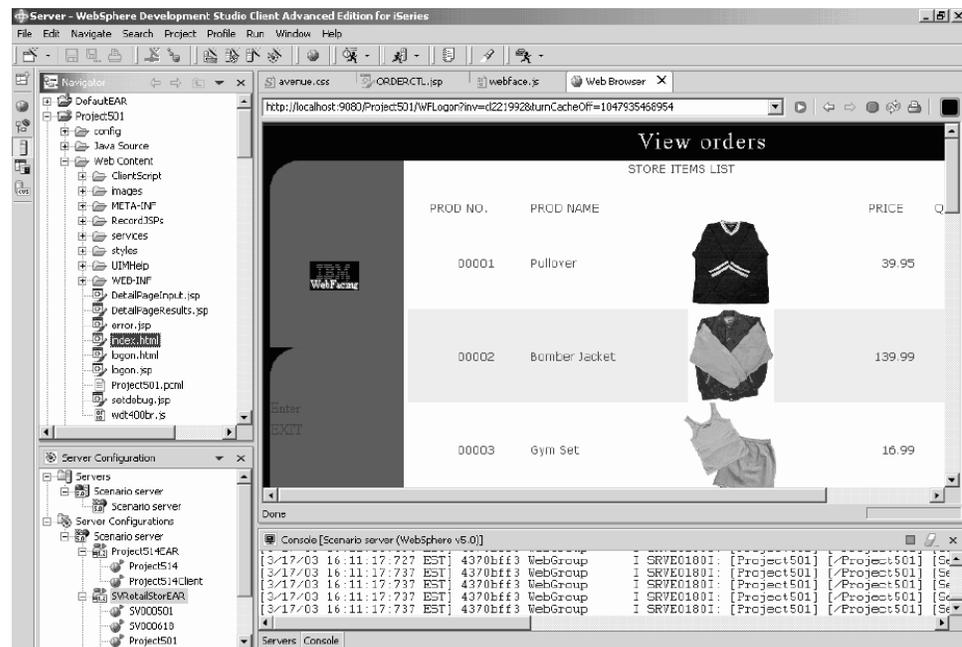
5. Klicken Sie **View orders – Im Hauptbrowserfenster starten** an, um die Administratoranwendung aufzurufen. Nachdem Sie sich mit Ihrer

iSeries-Benutzer-ID und Ihrem iSeries-Kennwort (bei dem iSeries-Server, den Sie beim Entwickeln dieses Projekts verwendet haben) angemeldet haben, wird die folgende Seite angezeigt:



Geben Sie versuchsweise ein X neben einer der Produktnummern ein, um Einzelangaben zu diesem Produkt anzuzeigen.

6. Klicken Sie den Rückwärtspfeil an, um die Seite index.html aufzurufen, und klicken Sie **View inventory – Im Hauptbrowserfenster starten** an, um die Kundenanwendung aufzurufen; daraufhin wird die folgende Seite angezeigt (nachdem Sie sich mit Ihrer Benutzer-ID und Ihrem Kennwort bei Ihrem iSeries-Server angemeldet haben):



Klicken Sie versuchsweise eine Abbildung an, um Einzelangaben zu diesem spezifischen Artikel anzuzeigen.

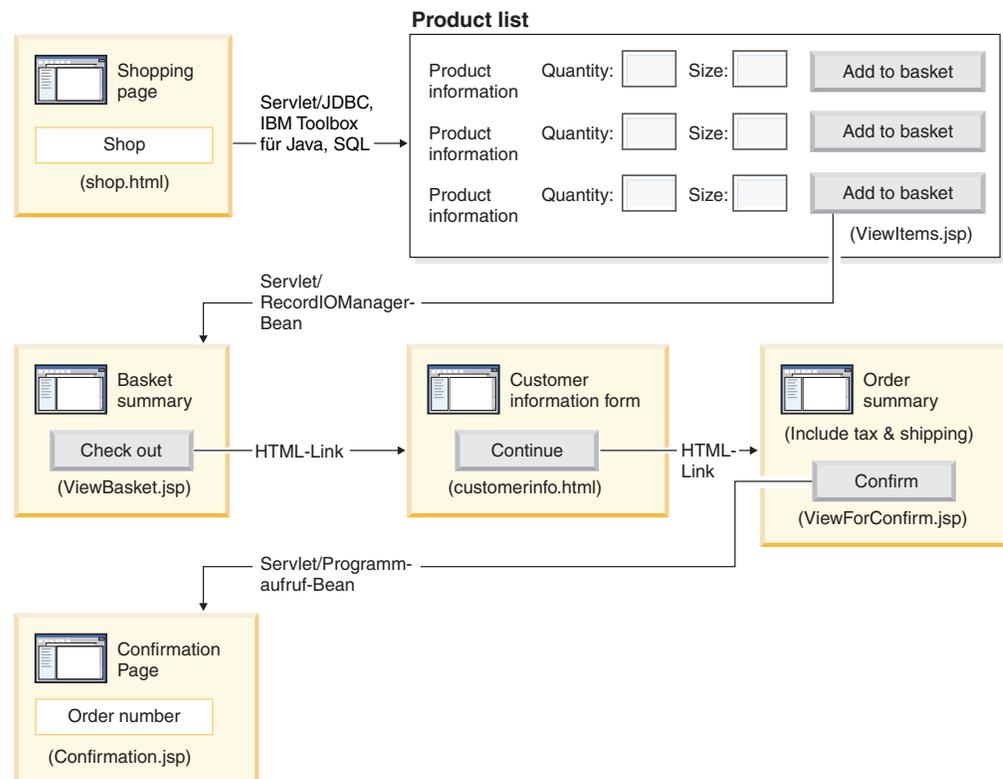
Sie sind jetzt am Ende des Moduls angelangt! Sie haben eine Benutzerschnittstelle zum Anzeigen des Bestands und zum Bestellen von Artikeln erstellt.



# Kapitel 6. Erweitertes Modul 1: HTML-Code, Servlets und JSP-Dateien erstellen, mit denen Kundenaufträge auf einen iSeries-Server gestellt werden (SV001585)

## Einführung

Dieses Projekt ist für Benutzer gedacht, die über gute Kenntnisse in der Programmierung mit Java und über einige Kenntnisse in der iSeries-Datenverwaltung und in RPG verfügen. Das Projekt zeigt Ihnen den Umgang mit IBM Toolbox für iSeries-Datenzugriffsklassen, der Bean RecordIOManager und der Programmaufruf-Bean zum Erstellen von HTML-Code, Servlets und JSPs, mit denen Kundenaufträge auf einen iSeries-Server gestellt werden. Als Benutzer starten Sie auf der Seite "Shopping", um die verfügbaren Produkte anzuzeigen und Ihrem Warenkorb (basket) Artikel hinzuzufügen. Sind Sie mit Ihrer Auswahl der Artikel zufrieden, die in Ihrer Warenkorbübersicht (Basket summary) angezeigt werden, klicken Sie eine Schaltfläche zum Beenden des Einkaufs (Check out) an und werden zu einem Formular mit Kundeninformationen (Customer information form) weitergeleitet. Nachdem Sie dieses Formular ausgefüllt haben, gibt das Projekt eine Übersicht der Aufträge (Order summary) mit einer Schaltfläche zum Bestätigen (Confirm) zurück, über die eine Seite zur Bestätigung (Confirmation Page) aufgerufen wird, die Ihren Auftrag (Order) und Ihre Auftragsnummer (Order number) zeigt.



---

## Zusammenfassung der Hauptschritte

Da es sich hierbei um ein erweitertes Modul handelt, führen die Anweisungen Sie nicht durch jeden Schritt zum Erstellen des Projekts, sondern umreißen die iSeries-spezifischen Entwicklungsschritte zum Erstellen eines derartigen Projekts. Dies sind die Hauptschritte:

1. Schreiben einer HTML-Webseite "Shopping"
2. Schreiben eines Servlets, das (unter Verwendung von JDBC und SQL) eine JavaBean mit verfügbaren Artikeln füllt, die der Kunde kaufen kann
3. Schreiben einer JSP-Datei, um Artikel anzuzeigen, die zum Kauf stehen, und es Kunden ermöglicht, die Menge und die Größe des gewünschten Artikels einzugeben und den Artikel auszuwählen, indem eine Schaltfläche **Add to basket** (Dem Warenkorb hinzufügen) angeklickt wird.
4. Verwenden der Bean RecordIOManager, Schreiben eines Servlets, das durch Anklicken der Schaltfläche **Add to basket** (Dem Warenkorb hinzufügen) aufgerufen wird und die iSeries-Datenbank INVENTORY aktualisiert, indem die für den Artikel erforderliche Menge und Größe subtrahiert werden und diese Auswahl einer JavaBean mit dem Namen "basket" hinzugefügt wird. Das Servlet leitet dann die Antwort an `ViewBasket.jsp` um, wenn die Operation erfolgreich ist. Schlägt die Operation fehl, zeigt das Servlet eine Fehlerseite an.
5. Schreiben eines HTML-Formulars für den Kunden zur Eingabe persönlicher Daten
6. Schreiben einer JSP-Dateiseite für die Bestätigung des Einkaufs, die den Inhalt des Warenkorbs, Steuern sowie Versandkosten und Bearbeitungsgebühren anzeigt. Die Seite muss auch eine Schaltfläche zum Bestätigen für den Kunden enthalten.
7. Schreiben eines Servlets, das aufgerufen wird, wenn der Kunde die Schaltfläche zum Bestätigen anklickt, die eine mit dem iSeries-Assistenten für Programmaufruf erstellte JavaBean verwendet. Eine der Methoden der JavaBean ruft ein RPG-Programm auf, um einen neuen Auftrag in der Datenbank ORDERS auf dem iSeries-Server zu erstellen, der dem Inhalt des Warenkorbs des Kunden entspricht. Das Servlet gibt dann eine Auftragsnummer zurück, stellt die JavaBean in die Sitzung der Webanwendung und lädt eine JSP-Datei zur Auftragsbestätigung, die die Auftragsnummer enthält.

---

## Vorbereitende Schritte

Sie können die Übungen nur dann vollständig ausführen, wenn die folgenden Voraussetzungen erfüllt sind. Die Voraussetzungen sind ausführlich in Kapitel 3, „Szenario ausführen“, auf Seite 9 erläutert.

- Sie verfügen über TCP/IP-Zugriff auf einen iSeries-Server.
- Sie haben die iSeries-Server mit dem Befehl `STRTCPSVR *ALL` gestartet.
- Sie haben den WebFacing-Server mit dem Befehl `STRTCPSVR *WEBFACING` gestartet.
- Sie haben die Bibliotheken `WHOLESALE` und `RETAILSTOR` auf Ihren iSeries-Server zurückgeschrieben.
- Die Bibliotheken `WHOLESALE`, `RETAILSTOR` und `QGPL` befinden sich in Ihrer Bibliotheksliste.
- Sie haben alle Tasks in Kapitel 3, „Szenario ausführen“, auf Seite 9 vollständig ausgeführt (mit Ausnahme der optionalen WebSphere Application Server-Tasks, die zum Testen der Anwendung in der Workbench nicht erforderlich sind).

---

## Webseite, Servlets und JSP-Dateien erstellen

Um die Komponenten von Projekt SV001585 zu erstellen, gehen Sie wie folgt vor:

1. Erstellen Sie ein Webprojekt.
2. Schreiben Sie eine Seite `shop.html` mit Page Designer, die einen Link enthält, mit dem ein Servlet `GetItems` aufgerufen wird.
3. Importieren Sie die Datei `jt400.jar` für die iSeries Toolbox für Java-Klassen in den Ordner *lib* des Webprojekts. Diese JAR-Datei befindet sich in `x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime`; dabei ist *x* das Verzeichnis, in dem Sie Development Studio Client installiert haben.

**Anmerkung:** `GetItems.java` und `ViewItems.jsp` in Projekt SV001585 zeigen die wichtigsten JDBC- und SQL-bezogenen Teile von iSeries Toolbox für Java für das Servlet und die JSP zeigen. In der Sicht "Navigator" der Webperspektive können Sie `GetItems.java` finden, indem Sie **SV001585 > source** erweitern, und `ViewItems.jsp`, indem Sie **SV001585 > webApplication** erweitern.

4. Schreiben Sie ein Servlet `GetItems`, das JDBC und SQL von iSeries Toolbox für Java verwendet, um Artikel (Kleidungsstücke) aus der iSeries-Datenbank INVENTORY abzurufen und:
  - a. die Bean `ResultSet`, die das SQL-Abfrageergebnis enthält, in die Sitzung stellt,
  - b. die Anforderung an `ViewItems.jsp` umleitet.

Code-Muster für `GetItems.jsp`:

```
public void init() {
    .
    .
    .
    // Laden Sie den IBM Toolbox für Java-JDBC-Treiber.
    DriverManager.registerDriver(new com.ibm.as400.access.AS400JDBCdriver());
    // Beachten Sie, dass der Name, die Benutzer-ID und das Kennwort für die
    // as400 mit einem XML-Parser aus der Datei web.xml abgerufen wurden.
    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. Schreiben Sie eine JSP-Datei ViewItems, die die Artikel (Kleidungsstücke) aus der Bean ResultSet (siehe vorheriger Schritt) abrufen, um die Artikel in Tabellenformat anzuzeigen. Die JSP-Datei sollte auch ein Formular für jeden Artikel enthalten, mit dem Sie Größe und Menge auswählen und dann den Artikel Ihrem Warenkorb hinzufügen können. Sie können die JSP mit Hilfe von Page Designer in iSeries Web Development Tools schreiben. Das heißt, Sie können die Seite in der Sicht "Entwurf" entwerfen und den entsprechenden Code in der Sicht "Quelle" hinzufügen. Code-Muster für ViewItems.jsp:

```

<!--Abrufen des Objekts ResultSet aus der Sitzung--><%
    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>Product ID</TD>
    <TD>Name</TD>
    <TD width="551">Price</TD>
    <TD colspan="2"></TD>
</TR>
<%while (rs.next ()) {
<TR>
<!--Erstellen eines Formulars für diese Zeile (oder diesen Artikel)-->
<FORM name="myform" action="/SV001585/AddtoBasket"
onsubmit="return errorChecking(this);">

<!--Abrufen der Daten für jede Spalte aus dieser Zeile des Objekts ResultSet-->
<!--processData ist eine benutzerdefinierte Methode zum Ändern der Daten für die
Anzeige, falls erforderlich-->
    <%
        for (int i = 1; i <= columnCount; ++i){
            String value = rs.getString(i);
            if (rs.wasNull ())
                value = "<null>";
            else{
                if(i==1)
                    prodID=value;
                value = processData(i,value);

            }

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

%>

```

```

<!--Erstellen des Eingabefelds "Quantity" und des Dropdown-Menüs "Size"-->
<!--Beachten Sie, dass die Produkt-ID (prodID) als Name des Felds verwendet wird-->
    <TD width="290">Quantity
        <INPUT size="5" type="text" name='<%=prodID+"Q"%>' ><BR>
        Size <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>
<%
}

```

- Importieren Sie die iSeries Java Development Tools-Datei iseriesut.jar in den Ordner *lib* für Ihr Webprojekt. Diese JAR-Datei befindet sich in `x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime`; dabei ist *x* das Verzeichnis, in dem Sie das Produkt installiert haben. Siehe `AddtoBasket.java` und `ViewBasket.jsp` im Projekt SV001585, die die Implementierung zeigen. In der Sicht "Navigator" der Webperspektive können Sie `AddtoBasket.java` finden, indem Sie **SV001585 > source** erweitern, und `ViewBasket.jsp`, indem Sie **SV001585 > webApplication** erweitern.
- Verwenden Sie die Bean RecordIOManager von iSeries Java Development Tools, um ein Servlet `AddtoBasket` zu schreiben, das über die Schaltfläche **Add to basket** aufgerufen wird, die die iSeries-Datenbank INVENTORY aktualisiert, indem die vom Kunden angeforderte Menge subtrahiert wird und die Artikel einer JavaBean `Basket` hinzugefügt werden. Code-Muster für `AddtoBasket.jsp`:

```

public class AddtoBasket extends HttpServlet {

    //Untergeordnete Klasse von 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);
            //Journal hat denselben Namen wie die Datenbankdatei
            setJournal(file);
            //Journal befindet sich in derselben Bibliothek wie die Datenbankdatei
            setJournalLibrary(lib);
        }
        .
        .
        .
        public synchronized String updatedDBFile(
            String id,

```

```

String size,
String quantity
) {
.
.
.
//Die Datei wird geöffnet.
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);
// Beachten Sie, dass der Satz aktualisiert wird,
// jedoch keine Festschreibung erfolgt, für den
// Fall, dass sich der Kunde dafür entscheidet,
// den Warenkorb zu leeren; in diesem Fall
// wird die Methode rollBack aufgerufen
updateRecord(record);
status = success;
} else {
status = notEnough;
}
} else
status = accessError;
} catch (Exception e) {
e.printStackTrace();
status = accessError;
}

//Die Datei wird geschlossen und Daten werden hinzugefügt.
try {
closeFile();
} catch (Exception e) {
//Im Fall eines Fehlers wird eine Operation ROLLBACK ausgeführt.
try {
rollback();
} catch (Exception e1) {
e1.printStackTrace();
}
}
status = accessError;
}

return status;
}

//Methode init für 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. Schreiben Sie eine JSP-Datei ViewBasket, die den Inhalt des Warenkorbs anzeigt.

9. Schreiben Sie ein Formular `customerinfo.html` für die Eingabe der Informationen zur Zahlung, einschließlich einer Schaltfläche **Continue** (Weiter), die `ViewForConfirm.jsp` aufruft.
10. Schreiben Sie eine `ViewForConfirm.jsp`, die den Inhalt des gesamten Warenkorbs sowie den Endbetrag anzeigt. Sie können `ViewForConfirm.jsp` auf dieselbe Art und Weise entwickeln wie `ViewBasket.jsp`, zusätzlich aber die berechneten Versandkosten und eine Schaltfläche **Confirm** (Bestätigen) angeben, die einen Auftrag in die iSeries-Datenbank ORDERS stellt.
11. Verwenden Sie den iSeries-Assistenten für Programmaufruf von iSeries Java Development Tools, um eine Bean `PLACEORD.java` zu erstellen, die auf das Serviceprogramm `PLACEODR` in der Bibliothek `RETAILSTOR` zugreift. Der Assistent erstellt Beans für die Verwendung durch Java-Anwendungen oder den Assistenten für Web-Services für den Zugriff auf iSeries-ILE-Programme.
  - a. Um den Assistenten zu öffnen, klicken Sie mit der rechten Maustaste `SV001585` in der Sicht "Navigator" an und wählen Sie **Neu > Andere** aus.
  - b. Klicken Sie im Fenster **Neu** die Optionen **iSeries > Java > Programmaufruf-Bean** an.
  - c. Im Assistenten für Programmaufruf geben Sie Informationen wie den iSeries-ILE-Programmnamen, die Bibliothek, den Programmtyp sowie Eingabe- und Ausgabeparameter ein.
  - d. Im letzten Fenster des Assistenten haben Sie die Option, eine Bean für eine Java-Anwendung und/oder einen Web-Service zu erstellen. In diesem Projekt müssen Sie nur eine Bean für eine Java-Anwendung erstellen.

**Anmerkung:** Das Serviceprogramm `PLACEODR` stellt jedes Element einer Strukturfeldgruppe in einen Satz der Datenbank ORDERS und generiert für jede Feldgruppe eine Auftragsnummer als Ausgabe.

12. Schreiben Sie ein Servlet `PlaceOrder`, das über die Schaltfläche **Confirm** (Bestätigen) in `ViewForConfirm.jsp` aufgerufen wird.
  - Das Servlet verwendet die vom iSeries-Assistenten für Programmaufruf generierte Bean, um auf einen iSeries-Server zuzugreifen und Aufträge in die Datenbank ORDERS in der Bibliothek `RETAILSTOR` zu stellen.
  - Aufträge sind die Artikel in dem Warenkorb, die als eine Strukturfeldgruppe an das ILE-Programm gesendet werden.
  - Jede Struktur dieser Feldgruppe ist ein Artikel im Warenkorb.
  - Das RPG-Serviceprogramm `PLACEORD`, das von der Bean aufgerufen wird, gibt die Auftragsnummer als einen Ausgabeparameter zurück und stellt ihn in die Sitzung.

Das folgende Codesegment zeigt, wie das Servlet `PlaceOrder` die Bean `PLACEORD` verwendet:

```

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

    try {
        /* Ein Exemplar der Bean PLACEORD, die von dem iSeries-Assistenten
           für Programmaufruf-Bean erstellt wurde, wird erstellt. */
        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;

    // Die bestellten Artikel werden aus dem Warenkorb abgerufen.
    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();
        // Die Elemente der Strukturfeldgruppe werden definiert.
        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;
        }
        // Die übrigen Feldgruppenelemente werden auf Pseudowerte gesetzt.
        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");
        }

        // Die zwei anderen Eingabeparameter der Bean werden definiert.
        orderBean.setNumOfItems(new BigDecimal(j));
        orderBean.setBalance((BigDecimal)
        request.getSession().getAttribute("balance"));
        try {
            // Das iSeries-Programm wird aufgerufen.
            orderBean.invoke();

            // Die Auftragsnummer aus der Bean PLACEORD wird abgerufen.
            orderNumber = (orderBean.getRetCode()).toString();
            request.getSession().setAttribute("orderNumber",
            orderNumber);
            basket.empty();
            // Dieser Auftrag wird jetzt festgeschrieben.
            recIO.commit();
            response.sendRedirect("orderNumber.jsp");
            return;
        } catch (Exception e) {
            response.sendRedirect("errorPage.html");
            e.printStackTrace();
        }
    }
}

```

```
}  
}  
}
```

13. Schreiben Sie ein Servlet `OrderNumber`, das die Nummer abrufen und sie dem Kunden zusammen mit einer Bestätigungsnachricht anzeigt. Hat der Kunde seinem Warenkorb keine Artikel hinzugefügt, stellen Sie sicher, dass stattdessen eine Fehlerseite zurückgegeben wird.

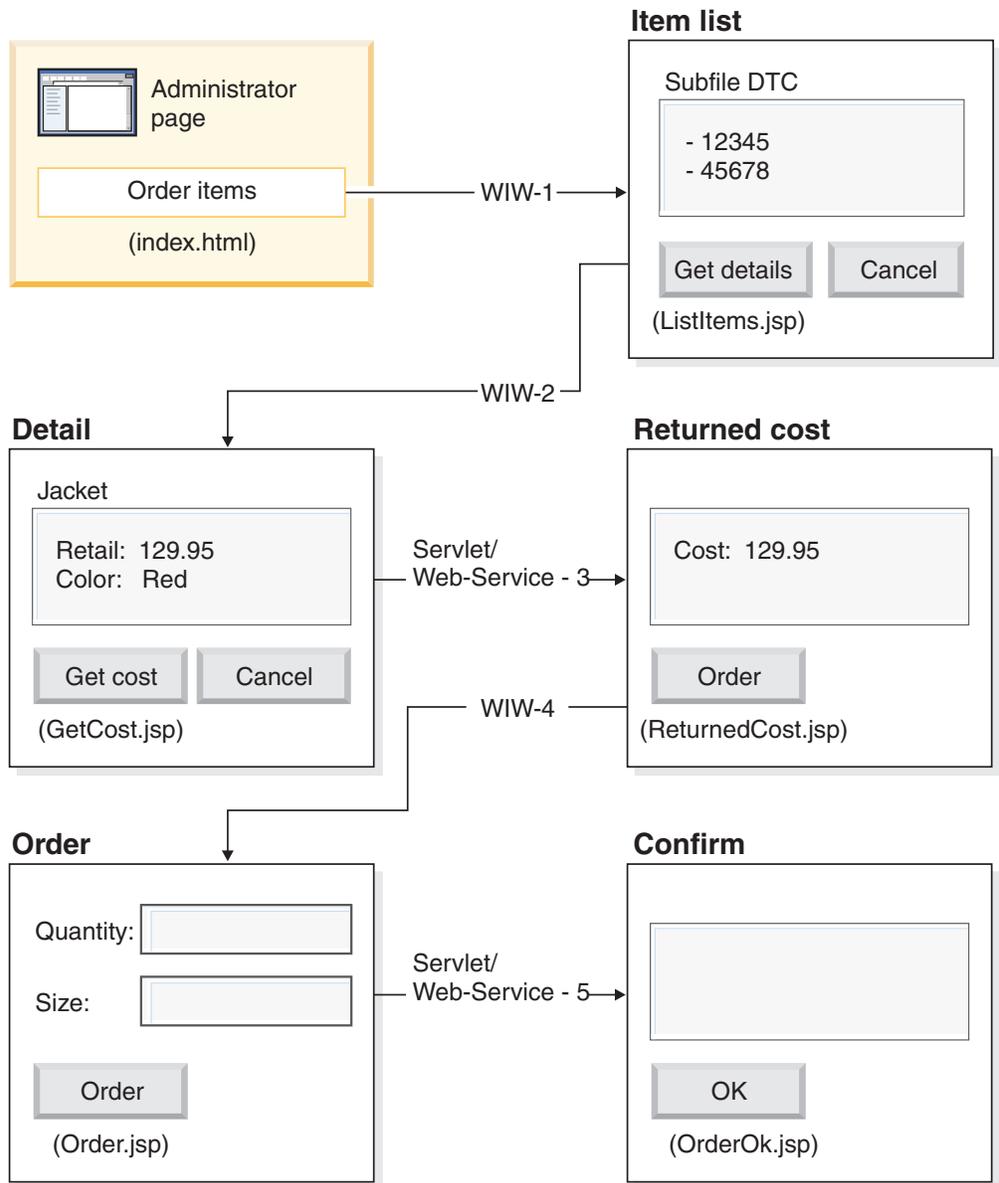
---

## **Kapitel 7. Erweitertes Modul 2: Webprojekt erstellen, das die Web-Services in Projekt SV000514 und Projekt SV001586 verwendet (SV000618)**

---

### **Einführung**

Dieses Projekt zeigt, wie Sie Ihre Kenntnisse in der RPG-Programmierung zum Erstellen von Web-Clients für iSeries Web-Services und RPG-Programme einsetzen können. In diesem Projekt übernehmen Sie die Rolle des Administrators, der eine Reihe von Webseiten durchläuft, um den Bestand zu bestimmen und um zusätzliche Waren für Ihr Einzelhandelsunternehmen bei dem Großhandelsunternehmen zu bestellen. Sie geben die Artikelnummer ein, zeigen die Einzelangaben zu dem Artikel an, bestellen die gewünschte Menge und Größe und akzeptieren die Bestätigung.



(WIW = Web Interaction Wizard = Assistent für Webinteraktion)

Da es sich hierbei um ein erweitertes Modul handelt, führen die Anweisungen Sie nicht durch jeden Schritt zum Erstellen des Projekts, sondern umreißen die Entwicklungsschritte zum Erstellen eines derartigen Projekts. Dieses Projekt verwendet die folgenden Komponenten von Development Studio Client:

- iSeries Web Development Tools zum Erstellen der Webseiten mit Page Designer, bei dem gleichzeitig die Ausgabe aus dem Assistenten für Webinteraktion und verschiedenen Webkomponenten eingegliedert wird
- Remote System Explorer zum Erstellen eines RPG-Serviceprogramms TNLSTITM, das Artikeldaten zurückgibt.
- Der Web-Services-Assistent zum Generieren von Servlet-Proxy-Code zum Suchen von Artikelpreisen und zum Bestellen der Artikel
- iSeries Java Development Tools zum Erstellen der erforderlichen Servlets

- Die WebSphere Test Environment zum Überprüfen der Anwendung vor der Implementierung auf dem iSeries-Server durch WebSphere Application Server

---

## Vorbereitende Schritte

Sie können die Übungen nur dann vollständig ausführen, wenn die folgenden Voraussetzungen erfüllt sind. Die Voraussetzungen sind ausführlich in Kapitel 3, „Szenario ausführen“, auf Seite 9 erläutert.

- Sie verfügen über TCP/IP-Zugriff auf einen iSeries-Server.
- Sie haben die iSeries-Server mit dem Befehl STRTCPSVR \*ALL gestartet.
- Sie haben den WebFacing-Server mit dem Befehl STRTCPSVR \*WEBFACING gestartet.
- Sie haben die Bibliotheken WHOLESale und RETAILSTOR auf Ihren iSeries-Server zurückgeschrieben.
- Die Bibliotheken WHOLESale, RETAILSTOR und QGPL befinden sich in Ihrer Bibliotheksliste.
- Sie haben alle Tasks in Kapitel 3, „Szenario ausführen“, auf Seite 9 vollständig ausgeführt (mit Ausnahme der optionalen WebSphere Application Server-Tasks, die zum Testen der Anwendung in der Workbench nicht erforderlich sind).

---

## Webseiten, Servlets, JSPs und RPG-Code erstellen

Um die Komponenten von SV000618 zu erstellen, gehen Sie wie folgt vor:

1. Erstellen Sie ein Webprojekt, um alle Dateien aufzunehmen, die Sie erstellen werden.
2. Schreiben Sie eine JSP-Datei ListItems, die Einträge in der iSeries-Datenbank INVENTORY auflistet. Das Servlet kann mit Hilfe von Page Designer in iSeries Web Development Tools geschrieben werden. Das heißt, Sie können die Seite in der Sicht "Entwurf" entwerfen und den entsprechenden Code in der Sicht "Quelle" hinzufügen. Sie müssen auch ein Subdatei-Entwurfszeitsteuerelement (Subdatei-DTC) einfügen, um mit dem RPG-Serviceprogramm TNLSTITM zu interagieren und die Subdatei mit Datenbanksätzen zu füllen. Sie können das Serviceprogramm in den DTC-Steuereinstellungen angeben.

Als nächsten Schritt müssen Sie mit Hilfe des Assistenten für Webinteraktion die Eingabeseite erstellen:

- Geben Sie ListItems.jsp als die Ausgabeseite zum Auflisten von Bestandsartikeln an; damit wird sichergestellt, dass der Assistent für Webinteraktion eine Datei ListItems.wit erstellt.
- Stellen Sie sicher, dass Sie im Assistenten für Webinteraktion keine Programmaufrufe angeben, da das Subdatei-DTC automatisch das RPG-Serviceprogramm TNLSTITM aufruft. Der Assistent generiert auch ein Servlet ListItemsWitServlet, das als Link zum Aufrufen der Seite ListItems.jsp dient.
- Um die vom Assistenten für Webinteraktion generierte Datei ListItemsWit.wit zu überprüfen, gehen Sie wie folgt vor:
  1. Erweitern Sie **SV001618** und klicken Sie **ListItems.wit** doppelt an, um den Interaktionsassistenten für die Datei anzuzeigen.
  2. Klicken Sie in dem Assistenten jeweils **Weiter** an, um die für die Interaktion angegebenen Werte zu überprüfen.

Als nächsten Schritt müssen Sie mit Page Designer unter Verwendung der Eingabe aus der Eingabeseite ListItems.jsp eine Ausgabeseite GetCost.jsp schreiben. Klickt

ein Benutzer einen Artikel auf der Seite `ListItems.jsp` an, zeigt eine Seite `GetCost.jsp` Einzelangaben zu dem Artikel an.

Nachdem die Seite `GetCost.jsp` erstellt wurde, erstellen Sie mit Hilfe des Assistenten für Webinteraktion eine Interaktion "WitOrder" zwischen `ListItems.jsp` (diese Seite wählen Sie als die Eingabeseite aus) und `GetCost.jsp` (diese Seite wählen Sie als die Ausgabeseite aus):

- Geben Sie auf der Seite "Programmaufruf" des Assistenten einen Aufruf der Prozedur `GetDetail` und einen Parameter aus dem RPG-Serviceprogramm `TNLSTITM` an.
- In die Prozedur sind Subdatei-DTC-APIs eingegliedert, die bestimmen, welcher Satz ausgewählt wurde. Anhand dieser Informationen ruft die Prozedur den ausgewählten Satz aus der Datenbank `INVENTORY` ab und zeigt Einzelangaben, einschließlich der Abbildung, für den ausgewählten Artikel auf der Seite `GetCost.jsp` an.
- Um die vom Assistenten für Webinteraktion generierte Datei `WitOrderWit.wit` zu überprüfen, gehen Sie wie folgt vor:
  1. Erweitern Sie **SV001618** und klicken Sie **WitOrder.wit** doppelt an, um den Interaktionsassistenten für die Datei anzuzeigen.
  2. Klicken Sie in dem Assistenten jeweils **Weiter** an, um die für die Interaktion angegebenen Werte zu überprüfen.

Beachten Sie, dass der Parameter *flow* als Ablaufcontroller auf der Ausgabeseite angegeben ist. Damit wird über den Wert des Parameters sichergestellt, dass die entsprechenden JSP-Dateien angezeigt werden.

Als nächsten Schritt müssen Sie die WSDL-Dateien (WSDL = Web Services Definition Language) aus Projekt `SV000514` importieren, damit der Administrator den aktuellen Preis eines Artikels vom Großhandelsunternehmen abrufen kann, indem er die Schaltfläche **Get cost** (Preis abrufen) anklickt.

- Die Schaltfläche **Get cost** ruft `QryProdCostServlet.jsp` und den zugehörigen Web-Service aus Projekt `SV000514` auf.
- Verwenden Sie den Assistenten für Web-Services und die importierten WSDL-Dateien, um den Java-Proxy-Code zu generieren, der zum Aufrufen des Web-Service erforderlich ist.
- `QryProdCostServlet.jsp` verwendet die Eingabe der Seite `GetCost.jsp`, ruft mit Hilfe von Java-Proxy-Code den Web-Service in `SV000514` auf, um den Preis des ausgewählten Artikels zu finden, und zeigt den Preis auf einer Seite mit dem Namen `ReturnedCost.jsp` an.
- Um den Code für `QryProdCostServicesProxy.java` und `QryProdCostServlet.java` anzuzeigen, gehen Sie wie folgt vor:
  1. Erweitern Sie **SV001618** > **source** > **proxy** > **soap**.
  2. Klicken Sie **QryProdCostServicesProxy.java** doppelt an.
  3. Klicken Sie für `QryProdCostServlet.java` den Wert **QryProdCostServlet.java** unter **SV001618** > **source** doppelt an und beobachten Sie die Erstellung eines Exemplars des Java-Proxy-Codes.

Als nächsten Schritt müssen Sie mit Hilfe des Assistenten für Webinteraktion `ReturnCost.jsp` als Eingabe und `Order.jsp` als Ausgabe verlinken, sodass der Administrator eine Schaltfläche **Order** anklicken kann, um den ausgewählten Artikel beim Großhandelsunternehmen zu bestellen.

- Bei dieser Interaktion muss kein Programmaufruf verwendet werden, da das Verlinken der beiden Seiten zum Anzeigen der korrekten Informationen ausreicht.
- Um `WitPlaceOrder.wit` anzuzeigen, gehen Sie wie folgt vor:
  1. Erweitern Sie **SV001618**.
  2. Klicken Sie **WitPlaceOrder.wit** doppelt an, um die Interaktion zu öffnen.
  3. Klicken Sie in dem Assistenten jeweils **Weiter** an, um die angegebenen Werte zu überprüfen.

Der nächste Schritt ist die Verwendung des Web-Service in SV001586, sodass der Administrator die Größe und Menge der bestellten Artikel angeben kann.

- Importieren Sie die WSDL-Dateien aus SV001586 in dieses Projekt, generieren Sie Java-Proxy-Code zum Aufrufen des Web-Service und schreiben Sie ein Servlet `OrderSupplyServlet`, das aufgerufen wird, wenn der Benutzer die Schaltfläche **Order** in `Order.jsp` anklickt.
- Das Servlet stellt Informationen aus `Order.jsp` zusammen, ruft den Java-Proxy-Code für den Web-Service auf, der den Web-Service in SV001586 aufruft und den Artikel bestellt.
- Das Servlet zeigt `OrderOK.jsp` an, wenn die Bestellung erfolgreich ausgeführt wird, und eine Fehlerseite, wenn sie fehlschlägt.
- Um zu beobachten, wie ein Exemplar des generierten Web-Service-Proxys erstellt und zur Verwendung des Web-Service in SV001586 aufgerufen wird, gehen Sie wie folgt vor:
  1. Erweitern Sie **SV000618 > source**.
  2. Klicken Sie **OrderSupplyServlet.java** doppelt an und untersuchen Sie den Inhalt.

---

## Vor der Implementierung in WebSphere Application Server

Bevor Sie Ihre Anwendung in WebSphere Application Server implementieren, müssen Sie einen spezifischen URL innerhalb der Java-Klassendateien für SV000514 und SV001586 ändern, damit die Anwendung korrekt funktioniert.

Für SV000514:

- In der Web-Services-Proxyklasse `QryProdCostServicesProxy` ist eine Variable definiert, die den URL des aufzurufenden Web-Service enthält.
- Beim ersten Erstellen des Proxys wird dieser URL zunächst auf `http://localhost:9080/SV000514/servlet/rpcrouter` gesetzt.
- Während die Variable auf diesen Wert gesetzt ist, wird der Web-Service in Projekt SV000514 in der IDE aufgerufen (wie in „Anwendung in der Workbench ausführen“ auf Seite 19 gezeigt).
- Bevor Sie diese Anwendung implementieren, müssen Sie diesen URL-Wert so ändern, dass er auf die Position zeigt, in der die EAR-Datei, `SVWholeSale.ear`, in dem iSeries-IFS-Verzeichnis implementiert wurde.

Für Projekt SV001586:

- In der Web-Services-Proxyklasse `OrderSupplyServicesProxy` ist eine Variable definiert, die den URL des aufzurufenden Web-Service enthält.
- Beim ersten Erstellen des Proxys wird dieser URL zunächst auf `http://localhost:9080/SV001586/servlet/rpcrouter` gesetzt.

- Während die Variable auf diesen Wert gesetzt ist, wird der Web-Service in Projekt SV001586 in der IDE aufgerufen (wie in „Anwendung in der Workbench ausführen“ auf Seite 19 gezeigt).
- Bevor Sie diese Anwendung implementieren, müssen Sie diesen URL-Wert so ändern, dass er auf die Position zeigt, in der die EAR-Datei SVWholeSale.ear in dem iSeries-IFS-Verzeichnis implementiert wurde.

---

## Kapitel 8. Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der Produkte, Programme oder Services können auch andere ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in dieser Dokumentation beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieser Dokumentation ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

*IBM Europe Director of Licensing 92066 Paris La Defense Cedex France*

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekannt gegeben. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

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

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM übernimmt keine Verantwortung für deren Richtigkeit. Fragen zum Leistungsspektrum von Produkten anderer Anbieter sind an den Anbieter dieser Produkte zu richten.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

---

## COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. IBM kann deshalb nicht garantieren, dass die Zuverlässigkeit, Wartungsfreundlichkeit und Funktion dieser Programme gegeben ist. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit den IBM Anwendungsprogrammierschnittstellen konform sind.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

(C) (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. (C) Copyright IBM Corp. 1992, 2002. Alle Rechte vorbehalten.

---

## Informationen zur Programmierschnittstelle

Die Informationen zur Programmierschnittstelle sollen Ihnen bei der Erstellung von Anwendungssoftware mit Hilfe dieses Programms helfen.

Mit allgemeinen Programmierschnittstellen können Sie Anwendungssoftware erstellen, die Services der Tools dieses Programms erhält.

Diese Informationen können jedoch auch Diagnose-, Änderungs- und Optimierungsinformationen enthalten. Diagnose-, Änderungs- und Optimierungsinformationen stehen für die Fehlerbehebung Ihrer Anwendungssoftware zur Verfügung.

**Warnung:** Verwenden Sie diese Diagnose-, Änderungs- und Optimierungsinformationen nicht als Programmierschnittstelle, weil sie Änderungen unterliegen.

---

## Marken und Dienstleistungsmarken

- 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
- 
- 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 ist eine Marke der InstallShield Corporation.

Intel und Pentium sind in gewissen Ländern oder Regionen Marken der Intel Corporation.

Java und alle Java-basierten Marken und Logos sind in gewissen Ländern oder Regionen Marken oder eingetragene Marken der Sun Microsystems, Inc.

Rational ist in gewissen Ländern oder Regionen eine Marke der International Business Machines Corporation und der Rational Software Corporation.

Lotus, Lotus Notes und Domino sind in gewissen Ländern oder Regionen Marken der Lotus Development Corporation.

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT, Win32, Win32s und das Windows-Logo sind in gewissen Ländern oder Regionen Marken der Microsoft Corporation.

Netscape Navigator ist eine Marke der Netscape Communications Corporation.

UNIX ist eine eintragung Marke von The Open Group.

Andere Namen von Unternehmen, Produkten und Dienstleistungen können Marken oder Dienstleistungsmarken anderer Unternehmen sein.







Programmnummer: 5724-A81

Gedruckt in Deutschland