



SOAP/JMS を使用したエンタープライズ Bean からの Web サービスの作成

目次

SOAP/JMS を使用したエンタープライズ

Bean からの Web サービスの作成. . . . 1

演習 1.1: JMS のサーバーとサーバー構成の作成 . . . 1

JMS サーバーの作成. 1

JMS を処理するためのサーバーの構成. 2

演習のチェックポイント 3

演習 1.2: Web サービスの作成 4

演習のチェックポイント 7

要約 7

SOAP/JMS を使用したエンタープライズ Bean からの Web サービスの作成

学習目標

このチュートリアルでは、次の内容を学習します。

- JMS サーバーおよびサーバー構成の作成
- TestEJB.wsdl という名前の WSDL 文書の作成
- Web サービスの、WebSphere Application Server へのデプロイ
- Java クライアント・プロキシの生成
- サンプル・アプリケーションの生成および起動

このチュートリアルを完了するには、約 2 時間必要です。チュートリアルの使用中に Web サービスの他のファセットを調べるように決めた場合は、終了するのにこれより長くなる場合があります。

前提条件

このチュートリアルを最初から最後まで完了するには、次の内容を十分に理解している必要があります。

- Web サービスの概念
- JMS サーバーの概念
- 基本的な EJB の概念

準備が完了したら、『演習 1.1: JMS のサーバーとサーバー構成の作成』を開始します。

フィードバック

関連情報



PDF 版の表示

演習 1.1: JMS のサーバーとサーバー構成の作成

この演習では、Web サービスで使用する JMS サーバー構成の作成方法を学習します。

プロジェクト交換ウィザードを使用すると、必要な EAR ファイルをインポートできます。

JMSEAR.ear のインポート

J2EE パースペクティブの JMSService に注目してください。このプロジェクトには、エンタープライズ Bean、リモート・インターフェース、および EJB ホームを含めて、EJB アプリケーションのすべてのリソースが含まれています。JMSServiceRouter プロジェクトも作成されています。これは、Web サービス用のメッセージ・ルーター・メッセージ駆動型 Bean が入ることになる EJB プロジェクトです。

JMS サーバーの作成

以下の手順を実行して、JMS サーバーを作成します。

1. 「ファイル」メニューから、「新規」 → 「その他」 → 「サーバー」 → 「サーバー」 → 「次へ」の順に選択する。
2. サーバーのタイプとして「**WebSphere v6.1 Server**」を選択する。「次へ」をクリックします。
3. このランタイムがワークスペースに作成されていない場合は、サーバーのインストール・ディレクトリを選択するよう、プロンプトが出される。「次へ」をクリックします。
4. デフォルト・サーバー・ポートと名前を受け入れる。「次へ」をクリックします。
5. 使用可能プロジェクトのリストから **JMSEAR** を選択し、「追加」をクリックして、サーバーに追加する。「終了」をクリックします。
6. サーバーが開始するまで待つ。サーバーが開始すると、コンソールに「e-business のためにサーバー server1 がオープンされました」と表示されます。

JMS を処理するためのサーバーの構成

このサーバーの JMS 設定値は、WebSphere Application Server 管理コンソールで設定されなければなりません。このコンソールは、Windows の「スタート」メニューから起動させることもできますし、Web ブラウザーから <http://localhost:9060/ibm/console> にアクセスして起動させることもできます。

1. コンソールを起動したならば、「サーバー」 → 「アプリケーション・サーバー」の順に選択して、作成したサーバーがリストされていることを確認する。
2. 左側のペインで、「サービス統合 (Service Integration)」 → 「バス (Buses)」の順に展開して、「新規」をクリックする。「名前」フィールドに固有名 (例: WS_tutorial_bus) を入力し、「次へ」をクリックしてから「終了」をクリックします。
3. 現行のサーバーを新規に作成された統合バスに関連付けるために、作成したばかりのバスの名前を選択し、「トポロジー」の下にある「バス・メンバー (Bus members)」をクリックする。「追加」をクリックして統合バスを関連付けたいサーバーの選択を行い、次に「次へ」をクリックします。メッセージの永続性の状態として「ファイル・ストア (File store)」を選択し、「次へ」をクリックします。このチュートリアルでのデフォルトのメッセージ・ストア・プロパティを受け入れ、「次へ」をクリックします。独自の Web サービス用に JMS バスを作成する場合は、「ヘルプ」を選択し、「ファイル・ストア設定 (File store settings)」で、最適な設定に関する追加情報を検索します。確認のために「終了」をクリックします。
4. 要求メッセージ用の物理キューを作成する。
 - a. 左側のペインで、「サービス統合 (Service Integration)」 → 「バス (Buses)」の順に展開する。前に作成したバスを選択します。
 - b. 「宛先リソース (Destination resources)」の下にある「宛先」をクリックする。
 - c. 「宛先」ページで、「新規」をクリックする。
 - d. 宛先タイプとして「キュー」を選択し、「次へ」をクリックする。
 - e. ws_tutorial_queueJms といった ID を入力する。「次へ」をクリックします。
 - f. デフォルトのバス・メンバーを受け入れる。「次へ」をクリックします。
 - g. 「終了」をクリックして、変更を確認し、次に変更を保管する。
5. 新規に作成されたキューに対して JMS 設定値を割り当てる。
 - a. 左側のナビゲーション・パネルで、「リソース」 → 「JMS」 → 「JMS プロバイダー」の順に進む。
 - b. 「範囲」ドロップダウン・リストから、範囲としてサーバーを選択し、プロバイダー・リストから「デフォルト・メッセージング・プロバイダー (Default messaging provider)」を選択する。
 - c. 「追加プロパティ (Additional Properties)」の下にある「キュー」をクリックする。

- d. 名前 (たとえば `ws_tutorial_queueJms`) および JNDI 名 (たとえば `jis/ws_tutorial_queue`) を入力する。接続ペインで、前に作成しておいたバス (`WS_tutorial_bus`) およびキュー (`ws_tutorial_queueJms`) を選択します。
 - e. 「**OK**」をクリックして変更を保管する。
6. 入力キュー用にキュー接続ファクトリーを作成する。
 - a. 「リソース」 → 「**JMS**」 → 「**キュー接続ファクトリー (Queue connection factories)**」の順に進む。
 - b. 「範囲」ドロップダウン・リストから、範囲としてサーバーを選択し、「**新規**」をクリックする。
 - c. デフォルト・メッセージング・プロバイダーを選択し、「**OK**」をクリックする。
 - d. 「一般プロパティ」で、名前 (例: `WebServicesInput_QCF`) および JNDI 名 (例: `jis/ws_tutorial_qcf`) を入力する。
 - e. 「接続」ペインで、前に作成しておいたバス (`WS_tutorial_Bus`) をバス名として選択する。
 - f. 「**OK**」をクリックして変更を保管する。
7. 応答キュー用に、キュー接続ファクトリーを作成する。
 - a. 「リソース」 → 「**JMS**」 → 「**キュー接続ファクトリー (Queue connection factories)**」の順に進む。
 - b. 「範囲」ドロップダウン・リストから、範囲としてサーバーを選択し、「**新規**」をクリックする。
 - c. デフォルト・メッセージング・プロバイダーを選択し、「**OK**」をクリックする。
 - d. 「一般プロパティ」で、名前として「`WebServicesReply_QCF`」(このフィールドでは「`WebServicesReply_QCF`」を使用する必要があります) および JNDI 名 (例: `jis/WebServicesReplyQCF`) を入力する。 応答キュー接続ファクトリーにカスタムの名前を使用したい場合は、`JMSServiceRouter` デプロイメント記述子で参照別名を変更する必要があります。この参照は、「Web サービス」ウィザードを実行するときにセットアップされます。つまり、異なる JNDI 名を使用することが決定している場合は、このプロジェクトを調べてデフォルト設定をオーバーライドする必要があります。
 - e. 「接続」ペインで、前に作成しておいたバス (`WS_tutorial_Bus`) をバス名として選択し、「**OK**」をクリックして変更を保管する。
8. 入力キューをバインドし、メッセージ主導 EJB を listen するために、JMS 活動化仕様が必要である。
 - a. 「リソース」 → 「**JMS**」 → 「**アクティベーション・スペック (Activation specifications)**」の順に進む。
 - b. 「範囲」ドロップダウン・リストから、範囲としてサーバーを選択し、「**新規**」をクリックする。
 - c. デフォルト・メッセージング・プロバイダーを選択し、「**OK**」をクリックする。
 - d. 名前 (例: `ws_tutorial_JMSRouter`) を入力し、JNDI 名 (例: `jis/ws_tutorial_JMSRouter`) を入力する。「宛先」ペインで、「**キュー**」を宛先タイプとして選択し、宛先 JNDI 名 (`jis/ws_tutorial_queue`) を入力し、バス名 (`WS_tutorial_Bus`) を選択します。
 - e. 「**OK**」をクリックして変更を保管する。
9. 必要な接続ファクトリー、およびキューまたはトピックを追加した後に、WebSphere Application Server v6.1 をいったん停止してから再始動し、開発ワークスペースに戻る。

演習のチェックポイント

これで、『演習 1.2: Web サービスの作成』を開始する準備が完了しました。

フィードバック

演習 1.2: Web サービスの作成

この演習では、ボトムアップ EJB Web サービスの作成方法を学習します。

始める前に、『演習 1.1: JMS のサーバーとサーバー構成の作成』を終了させておく必要があります。

WSDL 文書は、Web サービスのデプロイ先、およびこのサービスが提供する操作を記述したものです。以下の手順を実行すると、WSDL 文書、デプロイメント記述子ファイル、プロキシー、およびサンプルを作成できます。

1. 「ファイル」 → 「新規」 → 「その他」をクリックする。「Web サービス」を選択して、各種の Web サービス・ウィザードを表示します。「Web サービス」ウィザードを選択します。「次へ」をクリックして、Web サービス・ウィザードを開始してください。
2. 「Web サービス・タイプ」フィールドで、「EJB Web サービスのボトムアップ (Bottom up EJB Web service)」を選択する。
3. 「参照」をクリックしてサービス実装を選択する。ウィンドウに、使用可能なすべてのエンタープライズ Bean のリストが表示されます。TestEJB Bean を選択します。
4. サービスとクライアントを作成しこれら両方をサーバーで開始するために、サービスのスライダーを「開始」位置に移動し、クライアントのスライダーを「テスト」位置に移動する。この操作により、適切なサービスとクライアント・コード、プロジェクト、およびルーター・モジュールがすべて作成され、正しい EAR に関連付けられます。また、デプロイメント・コードが作成され、ターゲット・サーバーに EAR がインストールされてサービスが開始します。Web サービス・エクスプローラーは JMS 上の Web サービスをサポートしないため、サービスはテストされません。
5. 次のサーバー・サイド・オプションが選択されていることを確認する。
 - サーバー: WebSphere v6.1 Server
 - ランタイム: IBM WebSphere JAX-RPC ランタイム環境
 - サービス・プロジェクト: JMSService
 - サービス EAR プロジェクト: JMSEAR

次のクライアント・サイド・オプションが選択されていることを確認します。

- サーバー: WebSphere v6.1 Server
- ランタイム: IBM WebSphere JAX-RPC ランタイム環境
- クライアント・プロジェクト: JMSServiceEJBClient
- クライアント EAR プロジェクト: JMSServiceEJBClientEAR

まだ存在していないプロジェクトは、ウィザードによって作成されます。すべてのオプションを選択した場合は、次のようになります。

Web Service

Web Services

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

Web service type:

Service implementation:

Start service

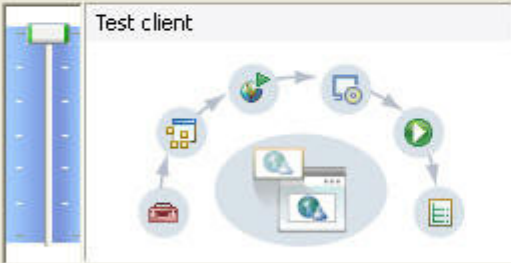


Configuration:

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

Client type:

Test client



Configuration:

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

☐ Publish the Web service

☐ Monitor the Web service

- Web サービス EJB 構成ページで、トランスポート方式として「**JMS**」を選択し、HTTP チェック・ボックスが選択されている場合はそれをクリアにする。JMSEAR の一部としてインポートした JMSServiceRouter ルーター・プロジェクトを選択する必要があります。「次へ」をクリックします。JMS は WS-I に準拠していないので、WS-I 準拠の設定を「無視」にしない場合、非準拠であることを警告するエラー・メッセージが表示されます。「詳細」をクリックすると、警告メッセージの理由が表示されます。この警告は無視しても問題ないので、「無視」をクリックしてください。
- EJB Web サービス・バインディング構成ページで、前の演習で作成したキューおよび接続ファクトリーと一致する次の値を手動で入力する。
 - JMS 宛先として キューが選択されることを確認する。このチュートリアルはトピック用には機能しません。

- 宛先 JNDI 名としての `.jms/ws_tutorial_queue`
- JMS 接続ファクトリー としての `.jms/ws_tutorial_qcf`
- ポート・コンポーネントの名前はターゲット・サービス名であるので、`TestEJB` がターゲット・サービス名として使用される。
- ActivationSpec JNDI 名としての `eis/ws_tutorial_JMSRoute`

完成したページは、次のようになります。

Web Service

EJB Web Service Binding Configuration

EJB Web Service Binding Configuration

JMS Binding

JMS destination:

Destination JNDI name:

JMS connection factory:

Port component name:

MDB deployment mechanism:

ActivationSpec JNDI name:

Listener input port name:

Initial context factory name:

JNDI provider URL:

Delivery Mode:

Request message lifetime:

JMS request message priority:

Connection factory userid:

Connection factory password:

- ウィザードの「Web サービス Java Bean の識別」ページで、ユーザーの Web サービス URI、スコープ、および生成されたファイルの名前を指定できる。また、Web サービスに組み込まれるメソッドと

エンコード・スタイルを選択したり、Web サービス用のセキュリティーを構成したりすることもできます。デフォルト値を受け入れるには、「次へ」をクリックしてください。

重要: Web サービスの URI は、Web サービスにするために選択した成果物から、ウィザードによって自動的に生成されます。デフォルトのベース URI `http://tempuri.org/` は、エンティティーとの固有の関連がない URI を構成する場合に使用します。ホスト名 `tempuri` は、WSDL 仕様から得られるものであり、一時 URI (temporary URI) の略語です。URI をグローバルに固有なものにしたい場合は、デフォルトのベース URI を使用してください。継続的な固定エンティティーのベースとして `http://tempuri.org/` を使用することはお勧めしません。

9. 「Web サービス・プロキシー」ページに、プロキシー・コードの生成場所がリストされる。クライアント・プロキシーが、ユーザーの Web サービスへのリモート・プロシージャ・コール・インターフェースを提供します。生成したプロキシーのセキュリティーを使用可能にしないでください。「次へ」をクリックします。
10. 「Web サービス・クライアント・テスト」ページを使用して、以下のオプションを選択する。
 - テスト機能としてサンプルの Web サービス・サンプル JSP を生成することを選択する。
 - JSP が置かれるフォルダーを選択し、その JSP にすべてのメソッドが組み込まれていることを確認する。
 - 「サーバーでテストを実行 (Run test on server)」を選択して、サーバーが自動的に開始されるようにする。「終了」をクリックします。
11. `http://localhost:9080/JMSClient/sample/TestEJB/TestClient.jsp` で、プロキシー JSP が Web ブラウザーで起動される。メソッドを選択し、そのメソッドに値を入力して「起動」をクリックすると、このサンプル・アプリケーションを使用して Web サービスをテストすることができます。メソッドの結果、すなわちテキスト・フィールドに入力したストリングのエコーが、結果ペインに表示されます。

演習のチェックポイント

『要約』の内容を検討して、チュートリアルを終了してください。

フィードバック

要約

JMS トランSPORTを使用する Web サービスの開発とテストが、これで終了しました。

このチュートリアルでは、「Web サービス」ウィザードを使用して、TestEJB エンタープライズ Bean から、TestEJB.wsdl という名前の Web サービス記述言語 (WSDL) 文書を生成しました。その次に、Web サービスを WebSphere Application Server にデプロイし、サンプル Web サービス JSP を使用してそれをテストしました。

このチュートリアルを完了し、テスト用の Web プロジェクトがなくなったら、各プロジェクトを右クリックし、「削除」を選択してワークスペースから除去してください。

追加リソース

Web サービスの詳細については、オンライン・ヘルプ (「ヘルプ」 → 「ヘルプ目次」) を参照してください。

関連情報



developerWorks Web サービス
Web サービスに関する詳細な技術資料。