IBM WebSphere Application Server V8.5 lab
Liberty profile administration in Eclipse

Scenario

You are a web application developer interested in learning how to deploy and administer WebSphere Application Server V8.5 Liberty profile servers in an Eclipse IDE that includes the WebSphere Application Server V8.5 Liberty Profile Developer Tools.

Goals

During this lab, you will learn to do the following in an Eclipse IDE:

1. Install the Liberty profile developer tools.
2. Install the Liberty profile and create a new server.
3. Deploy and test a simple servlet application.
4. Deploy and test a servlet application that requires a JDBC connection to a database.
5. Configure a server to use non-default ports for HTTP and HTTPS traffic.
6. Move a portion of a server configuration to an included configuration file.

For more information about the Liberty profile and the Liberty profile developer tools, see the information center topic Using the Liberty profile as an application development environment.

This lab is provided **AS-IS**, with no formal IBM support.

Prerequisites

This lab requires a single host machine with the following software and materials:

- Eclipse IDE for Java EE Developers Version 3.6 (SR2 with the latest updates) or Version 3.7 (SR2 or later). For the most up-to-date system requirements for the Liberty profile developer tools, see the WASdev community site and the information center topic Installing the Liberty profile developer tools and application-serving environment.
- Java 6 installed as the default (system) Java Runtime Environment (JRE). The minimum supported level for the JRE from Oracle is Java™ 6 update 26. For the Java JRE from IBM, the minimum supported level is 6.0 (J9 2.6) SR 1. Java 7 is supported; however, there are several significant restrictions. For more information, see the WebSphere Application Server V8.5 information center topic Liberty profile: Runtime environment known restrictions.
- An up-to-date copy of the Apache Derby database (`derby.jar`).
- The lab materials file, `WASv85Labs_LibEclipse.zip`. To download this file, visit the WebSphere Application Server V8.5 area of the IBM Education Assistant site.

Procedure

A. Getting started

Tasks

1. Extract lab materials file `WASv85Labs_LibEclipse.zip` into a suitable directory. You can extract this file and any other WebSphere Application Server V8.5 lab materials files into the root directory. For example, on Windows, extracting the file into the `C:\` directory will create lab materials directory `C:\WASv85Labs\LibEclipse`, and so on. You can also use the same basic approach on Linux.

B. Install the Liberty profile developer tools

Notes

- The following instructions describe how to install the Liberty profile developer tools from the Eclipse Marketplace. Additional methods are described in the WebSphere Application Server V8.5 information center topic Installing the Liberty profile developer tools and application-serving environment.

Tasks

1. Start Eclipse. When prompted, create a new workspace. If the Eclipse welcome screen is displayed, click **Workbench**.
2. Click **Help** > **Eclipse Marketplace**.
3. Use the Eclipse Marketplace window to search for offerings that include the text `Liberty Profile Developer Tools`. Within the search results, locate the **IBM WebSphere V8.5 Liberty Profile Developer Tools** offering. Next, click the corresponding **Install** button. Then follow the prompts to install the tools and any other required plug-ins.
4. When prompted, restart Eclipse.

C. Install the Liberty profile and create a new server

Notes

- The following instructions describe how to download and install the Liberty profile using the Eclipse New Server wizard. Additional installation methods are described in the WebSphere Application Server V8.5 information center topics Creating a Liberty profile server using developer tools and Installing the Liberty profile.

Tasks

1. Within Eclipse, use the New Server wizard to create a new server of type *IBM* > *WebSphere Application Server V8.5 Liberty Profile*. When creating the server, download the Liberty profile and install it to lab directory `WASv85Labs/LibEclipse/wlp`. After the installation completes, create a new Liberty profile server named **ServletSampleServer**.
    a. Within Eclipse, display the Servers view. Then click the new server wizard link.

b. Within the New Server wizard, do the following:

i. Select the server type **IBM > WebSphere Application Server V8.5 Liberty Profile** and click **Next**.

ii. Click "**Download or install** a new runtime environment".

Within the Install Runtime Environment wizard, select **Download and install a new runtime environment from** and **WebSphere Application Server V8.5 download site**. Next, enter your IBM.com user name and password. (If necessary, use the registration link to request a user name and password.) Then click **Next**.

Indicate that you accept the terms of the license agreement and click **Next**.

Enter or select the installation folder for the Liberty profile, for example,

Windows: `C:\WASv85Labs\LibEclipse\wlp`

Linux: `/WASv85Labs/LibEclipse/wlp`

Then click **Finish** to close the Install Runtime Environment Wizard and install the Liberty Profile.

After the installation is complete, click **Next**.

iii. Click **New** to create a new Liberty Profile server.

Within the New WebSphere Server wizard, set the server name to **ServletSampleServer**. Then click **Finish**.

iv. After the new server has been created, click **Finish**. You can now work with the server in Eclipse.

D. Deploy and test application ServletSample

Notes

- The ServletSample application (`ServletSample.war`) is located in lab directory `WASv85Labs/LibEclipse/apps`
- The application requires Servlet support.
- The application returns a page that displays the current date and time.

Tasks

1. Import the ServletSample application into the workspace.

a. Click **File > Import**.

b. Within the Import wizard, select **Web > WAR file**. Then click **Next**.

c. Specify the full path to the WAR file (`ServletSample.war`) that is located in lab directory `WASv85Labs/LibEclipse/apps.` Then click **Finish**. The status area in the lower right corner of the Eclipse window will provide progress information. After the workspace has been built and the application has been validated, proceed to the next step.

2. Run the ServletSample application on server **ServletSampleServer** (that you created in task set C). Verify that the application correctly displays the current date and time.

a. Within the Project Explorer view, click the **ServletSample** project with your secondary mouse button and select **Run As > Run on Server**.

b. Within the Run On Server wizard, select the new server and click **Finish**.

c. After the Eclipse web browser opens, verify that the page (generated by ServletSample) displays the current date and time.

3. Use the Console view to review the server and application startup messages.

4. Review the server configuration.
   a. Display the Servers view.
   b. Expand the **ServletSampleServer** node to display the Server Configuration node.
   c. Double-click the **Server Configuration** node.
   d. After the Server Configuration view opens, review the final server configuration. Notice that the Liberty profile developer tools added the required feature, application, and application monitoring elements to the configuration.
   e. Click the **Source** tab (at the bottom of the Server Configuration view) to display the XML source of the server configuration.

5. Close all open IDE web browsers, configuration files, and so on.

6. Stop the server.
   a. Display the Servers view.
   b. Click **ServletSampleServer** with your secondary mouse button and select **Stop**.
   c. Review the server shutdown messages in the Console view.

E. Deploy and test the JDBCSample application

Notes

- The application (`JDBCSample.war`) is located in lab directory `WASv85Labs/LibEclipse/apps`
- Each time the application (servlet) is invoked, it creates a database table, adds a record to the table, displays information from the record, and deletes the table.
  - The Derby database (`derby.jar`) is required. If you do not have a copy of the Derby database, see the Prerequisites section for more information.
  - You will create a data source with these attributes:
    - JDNI name: `jdbc/JDBCSampleDS`
    - Database name: `${shared.resource.dir}/derby/data/JDBCSampleDB`
    - JDBC driver: Driver present in `derby.jar`

Tasks

1. Create lab directory `WASv85Labs/LibEclipse/wlp/usr/shared/resources/derby`. Then add the Derby database (`derby.jar`) to the `derby` directory.
2. Import the JDBCSample application into the workspace.
   a. Click **File > Import**.
   b. Within the Import wizard, select **Web > WAR file**. Then click **Next**.

c. Specify the full path to the WAR file (`JDBCSample.war`) that is located in lab directory `WASv85Labs/LibEclipse/apps`. Then click **Finish**. The status area in the lower right corner of the Eclipse window will provide progress information. After the workspace has been built and the application has been validated, proceed to the next step.

3. Deploy the JDBCSample application to a new server named JDBCSampleServer.
   a. Display the Servers view.
   b. Click the existing Liberty profile server (**ServletSampleServer**) with your secondary mouse button and select **New > Server**.
   c. Within the Define a New Server page of the New Server wizard, select the server type **IBM > WebSphere Application Server V8.5 Liberty Profile**. Next, change the (Eclipse-specific) server name to `JDBCSampleServer`. Then click **Next**.
   d. Within the WebSphere Server page of the New Server wizard, do the following:
      i. Click **New** to display the New WebSphere Server window, which allows you to create a new Liberty profile server. Next, enter the name `JDBCSampleServer`. Then click **Finish**.
      ii. Click **Next**.
   e. Within the Add and Remove page of the New Server wizard, add the JDBCSample application to the list of configured applications. Then click **Finish**. You can now work with server JDBCSampleServer in Eclipse.

4. Review the current server configuration.
   a. Within the Project Explorer view, expand the WebSphere Application Server V8.5 Liberty Profile project and the JDBCSampleServer entry. Then double-click the `server.xml` file to display the Server Configuration view.
   b. Using the Design tab of the Server Configuration view, review the current server configuration. Notice that the Liberty profile developer tools added many of the required configuration elements.

5. Add the required data source to the server configuration.

   Using the Design tab of the Server Configuration view, do the following:
   a. Within the Configuration Structure area, select the **Server Configuration** element. Next, click **Add**. Then, within the Add Item window, select the **Data Source** element and click **OK**.
   b. Within the Data Source Details area, do the following:
      i. Set the ID to `JDBCSampleDS`
      ii. Set the JNDI name to `jdbc/JDBCSampleDS`
      iii. For JDBC driver, click **New > Nested**.
   c. Within the JDBC Driver Details area, do the following:
      i. For Shared libraries, click **New > Top Level**. Within the ID window, set the ID for the shared library to `derbyLib` and click **OK**.
   d. Within the Shared Library Details area, do the following:
      i. For Fileset reference, click **New > Nested**.

e. Within the Fileset Service Details area, do the following:

   i. Set Base directory to the directory that contains `derby.jar`, for example,

      Windows: `${shared.resource.dir}\derby`
      Linux: `${shared.resource.dir}/derby`

   ii. Set Includes pattern to `derby.jar`

f. Define the vendor-specific properties of the data source.

   i. Within the Configuration Structure area of the page, select the new data source. Next, click **Add**. Then, within the Add Item window, select **Properties for Derby Embedded** and click **OK**.

   ii. Within the Properties for Derby Embedded Details area of the page, set createDatabase to **create**. Then set databaseName to the following:

      Windows: `${shared.resource.dir}\derby\data\JDBCSampleDB`
      Linux: `${shared.resource.dir}/derby/data/JDBCSampleDB`

g. Press **Ctrl+S** to save the server configuration.

6. Review the final server configuration. Be sure to click the **Source** tab so that you can view the XML source of the configuration. Notice the following:

   The data source and the shared library are top-level configuration elements. This arrangement has several advantages. You can re-use the shared library in cases where your application must directly reference API's from the JDBC driver. In addition, you can re-use the shared library for application-defined data sources. Finally, multiple class loaders will not be created for the same files.

   The base directory of the fileset and the database name include the standard Liberty profile variable `shared.resource.dir`. With this arrangement, the server will continue to function properly when/if the base installation directory of the Liberty profile is changed. For more information, see the WebSphere Application Server V8.5 information center topic Liberty profile: Directory locations and properties.

7. Run the application on the server and verify that it returns a page that displays the population of a city.

   a. Within the Project Explorer view, click the **JDBCSample** project with your secondary mouse button and select **Run As > Run on Server**.

   b. Within the Run On Server wizard, select the server **JDBCSampleServer**. Then click **Finish**.

      The Eclipse web browser should display the population of a city. Each time the application (servlet) is invoked, it creates a database table, adds a record to the table, returns a page that displays information from the record, and deletes the table. For more information, see the source code file for the application, `JDBCSample.java`.

F. Configure JDBCSampleServer to use non-default ports for HTTP and HTTPS traffic

Tasks

1. Configure JDBCSampleServer to use the (non-default) TCP/IP port 9081 for HTTP traffic and port 9444 for HTTPS traffic. Store the port number values in a custom property (variable) in a new server boot properties file, effectively separating the machine-specific settings from the server configuration.
    a. Stop JDBCSampleServer.
        i. Display the Servers view.
        ii. Click **JDBCSampleServer** with your secondary mouse button and select **Stop**.
    b. Store the port number as a custom property (variable) in the server `bootstrap.properties` file.
        i. Within the Project Explorer view, click **WebSphere Application Server V8.5 Liberty Profile > servers > JDBCSampleServer** with your secondary mouse button and select **New > File**. In the New File window, set File name to `bootstrap.properties`. Then click **Finish**.
        ii. After `bootstrap.properties` appears in the Eclipse text editor view, add the following lines (custom properties) to the file:

            ```
            default.http.port=9081
            default.https.port=9444
            ```

        iii. Press **Ctrl+S** to save the changes.
    c. Update the HTTP Endpoint definition in the server configuration.
        i. Display the Design tab of the Server Configuration view for `server.xml`.
        ii. Within the Configuration Structure area, select the **HTTP Endpoint** element.
        iii. Within the HTTP Endpoint Details area, update these attributes:

            Port: `${default.http.port}`

            Secure port: `${default.https.port}`

            Note: When working with the Port and Secure port fields, pressing **Ctrl+Space** will display a selection list of available (defined) variables.
        iv. Press **Ctrl+S** to save the changes to `server.xml`.
    d. Start JDBCSampleServer.
        i. Display the Servers view.
        ii. Click **JDBCSampleServer** with your secondary mouse button and select **Start**.

2. Verify that the HTTP port is now 9081.
    a. Use the Console view to review applicable server messages.
    b. Using the Eclipse web browser, visit this URL to load the JDBCSample application using the new HTTP port: `http://localhost:9081/JDBCSample`

G. Move a portion of the JDBCSampleServer configuration to a shared configuration file

Tasks

1. Move the HTTP Endpoint definition from the primary JDBCSampleServer configuration file to a new, runtime-level shared configuration file named `global.xml`.
    a. Within the Project Explorer view, click **WebSphere Application Server V8.5 Liberty Profile > shared > config** with your secondary mouse button and select **New > Liberty Profile Configuration File**. In the New Liberty Profile Server Configuration File window, set File name to `global.xml`. Then click **Finish**.
    b. Update the new shared configuration file, `global.xml`.
        i. Within the Project Explorer view, expand **Liberty Profile > shared > config**. Then double-click `global.xml`.
        ii. Display the Source tabs of the Server Configuration views for `server.xml` and `global.xml`.
        iii. Copy the entire httpEndpoint element in `server.xml` and paste it into `global.xml`. Be sure to place the httpEndpoint element between the `<server>` and `</server>` tags.
        iv. Press **Ctrl+S** to save the changes to `global.xml`.

            Note: The Source view of the server configuration editor may incorrectly indicate that variables default.http.port and default.https.port are undefined.
    c. Update the server configuration file, `server.xml`.
        i. Display the Design tab of the Server Configuration view for `server.xml`.
        ii. Remove the **HTTP Endpoint** configuration element.
        iii. Include `global.xml` in `server.xml`. First, select the **Server Configuration** element. Next, add an **Include** element (statement). Then, within the Include File Details area, click **Browse > Relative Path**. After that, click **shared.config.dir (Shared Configuration)** and select `global.xml`. Finally, click **OK**.

            Notice that the Location field includes the standard property `${shared.config.dir}` instead of the absolute path to the shared configuration directory.
        iv. Press **Ctrl+S** to save the changes to `server.xml`.

2. Use the Console view to review messages related to the server configuration change. One message will indicate that `global.xml` was included in the server configuration. Another message will indicate that no functional changes were detected, given that the final server configuration is effectively the same.

3. Display and review a merged (consolidated) read-only view of the resulting configuration.
    a. Within the Servers view, expand the **JDBCSampleServer** node.
    b. Click the **Server Configuration** node with your secondary mouse button and select **Open Merged View**. Notice that the Merged View displays server configuration settings present in `server.xml` and settings present in the included configuration file (`global.xml`).

4. Reload the web page to verify that the application still functions properly.