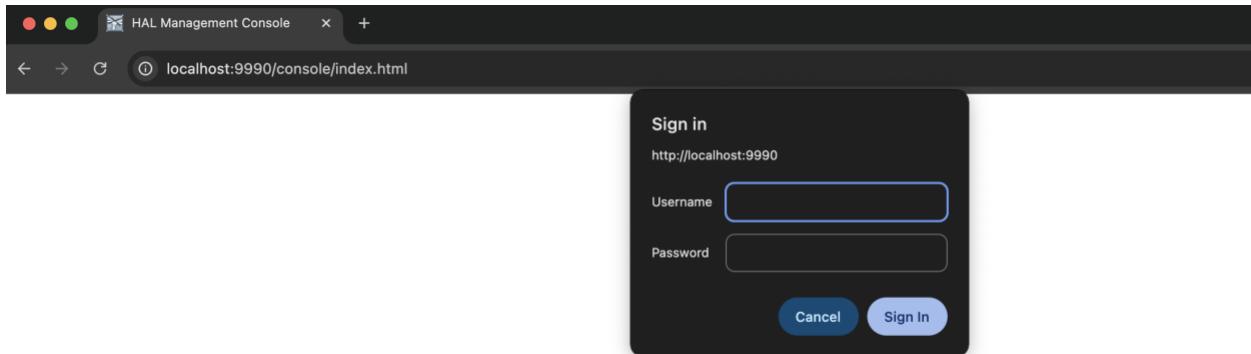


ITE5432 - J2EE Business Components

Lab Activity 2

Step 1:

Enter the following URL in web browser: <http://localhost:9990/>



The default username is **admin**. There is no default password. You can set password for the admin user or can create a new user. To add a user, navigate to **wildfly-35.0.1.Final -> bin** folder.

- Windows user: Execute add-user.bat
- Mac user: Execute ./add-user.sh

Add a user with the following credentials:

- Username: admin123
- Password: admin123

Upon successful login, you will see the following homepage.

The screenshot shows the WildFly Management Console homepage at localhost:9990/console/index.html. The top navigation bar includes links for Homepage, Deployments, Configuration, Runtime, and Access Control, along with a user dropdown for 'admin123'. The main content area is titled 'WildFly Application Server' and contains four sections: 'Deployments' (with 'Deploy an Application' and 'Start' buttons), 'Configuration' (with 'Create a Data source' and 'Start' buttons), 'Runtime' (with 'Monitor the Server' and 'Start' buttons), and 'Access Control' (with 'Assign User Roles' and 'Start' buttons). Each section has a brief description and a numbered list of steps.

Step 2:

We need to add a JMS Queue to send and receive messages. To add a JMS Queue, navigate to **Configuration** → **Subsystems** → **Messaging** → **Server** → **default** → **Destinations**.

Note: If WildFly server is not running with full profile, you may not see the Messaging Tab. Refer to Step 3 to run WildFly server with full profile.

Or you can directly run WildFly server with the following command. Navigate to **wildfly-35.0.1.Final->bin** and execute the following command **./standalone.sh -c standalone-full.xml** through command line.

The screenshot shows the 'Configuration' tab selected in the WildFly Management Console. In the left sidebar, under 'Subsystem (35)', 'Messaging' is expanded, and 'Activemq' is selected. The right panel displays the 'Destinations' configuration for the 'default' server. It includes sections for 'Destinations', 'Connections', 'Clustering', and 'HA Policy'. A detailed description of 'Core Queues' is provided, stating that ActiveMQ Artemis core is JMS-agnostic and maps JMS topics to core queues. The 'Destinations' section lists 'Core Queues', 'JMS Destinations', 'Security Settings', 'Address Settings', and 'Diverts'. A note at the bottom explains that for JMS topics, the address is prepended with 'jms.topic.'.

Click on View

The screenshot shows the WildFly Management Console interface. The URL is `localhost:9990/console/index.html#messaging-server-destination;server=default`. The left sidebar has a tree structure with nodes: Core Queue, JMS Queue, JMS Topic, Security Setting, Address Setting, and Divert. The 'Core Queue' node is selected. The main content area is titled 'Core Queue' and contains the sub-section 'A Queue'. It shows a table with one row: 'No data available in table'. There are buttons for 'Add' and 'Remove'. Below the table are sections for 'Queue Address', 'Filter', and 'Durable' (set to 'true').

Click on JMS Queue

The screenshot shows the WildFly Management Console interface. The URL is `localhost:9990/console/index.html#messaging-server-destination;server=default`. The left sidebar has a tree structure with nodes: Core Queue, JMS Queue, JMS Topic, Security Setting, Address Setting, and Divert. The 'JMS Queue' node is selected. The main content area is titled 'JMS Queue' and contains the sub-section 'Defines a Jakarta Messaging queue.' It shows a table with three entries: DLQ, ExpiryQueue, and TestQueue. There are buttons for 'Add' and 'Remove'. Below the table are sections for 'Selector', 'Entries', and 'Durable' (set to 'true').

To add a JMS Queue, click on **Add** and input the required information as outlined below:

Name: TestQueue

Entries: `java:/jms/queue/TestQueue`

Press **enter** to add entries.

Add JMS Queue

[Help](#)

Name	TestQueue
Durable	<input checked="" type="checkbox"/> ON
Selector	
Entries *	java:/jms/queue/TestQueue

Press ↲ to add new items and ✖ to remove them.

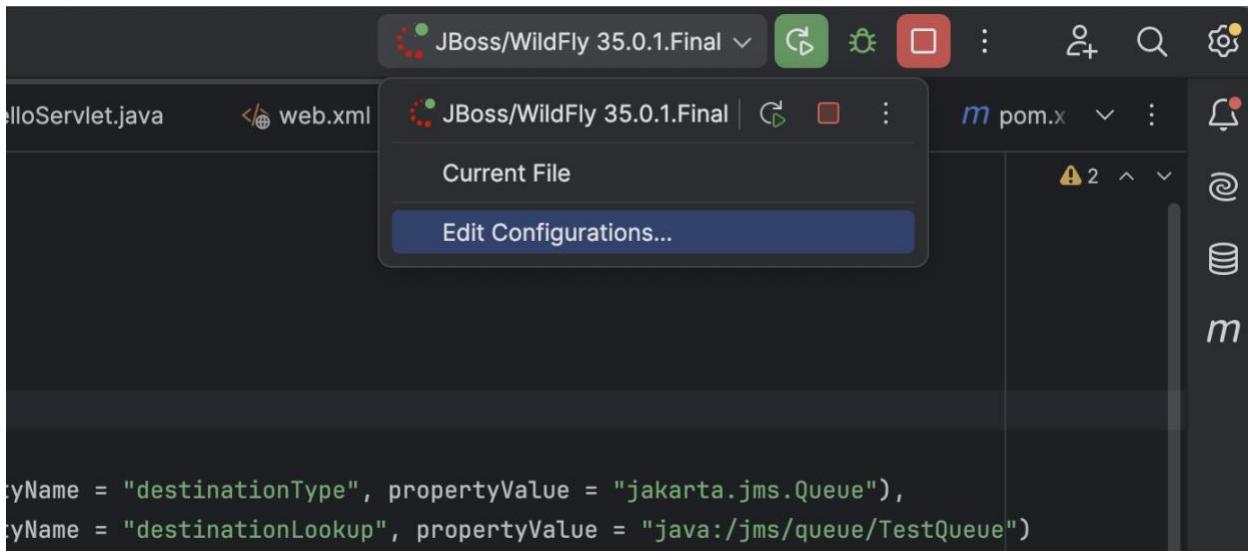
Required fields are marked with *

[Cancel](#) [Add](#)

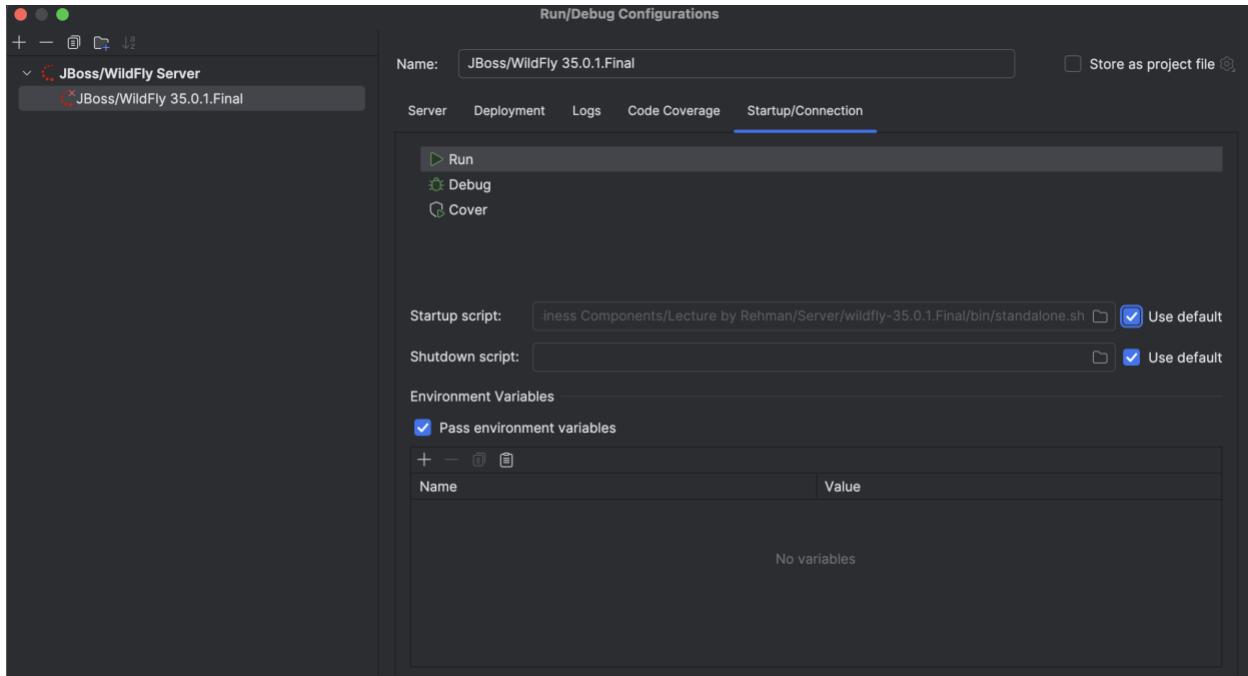
Step 3:

To run the server with full profile, follow these steps:

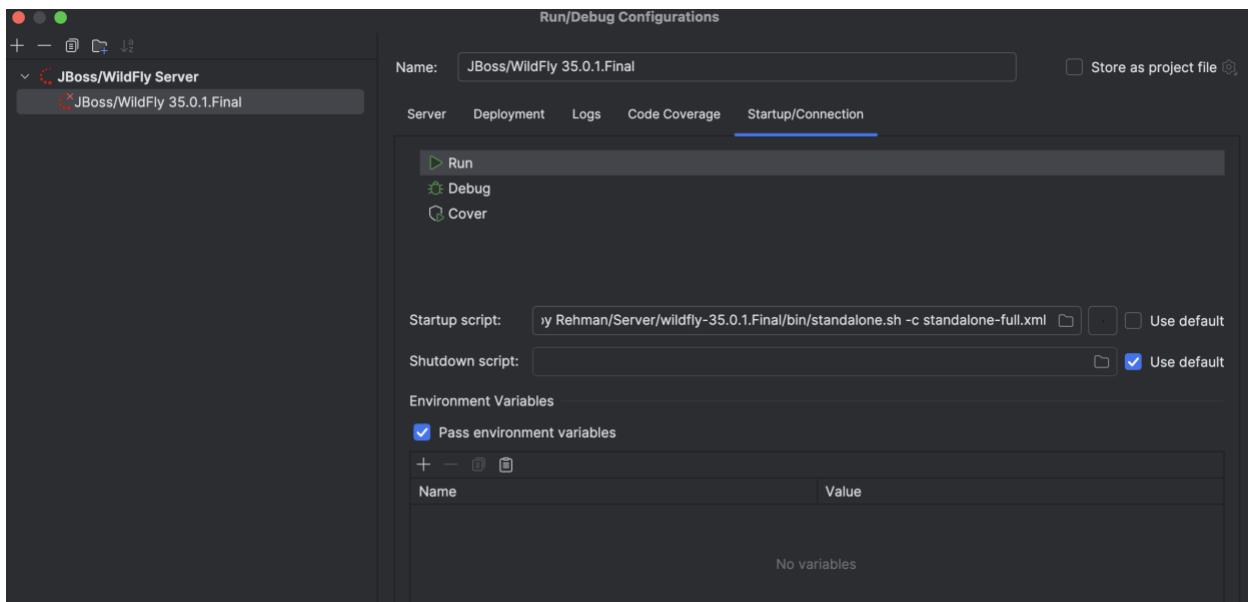
Edit Configuration



Select Startup/Connection tab

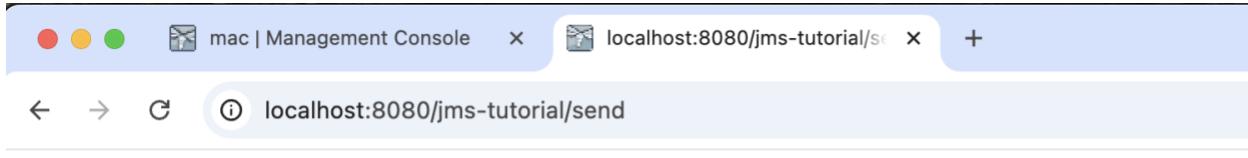


Uncheck startup script and add **-c standalone-full.xml** at the end of the path.

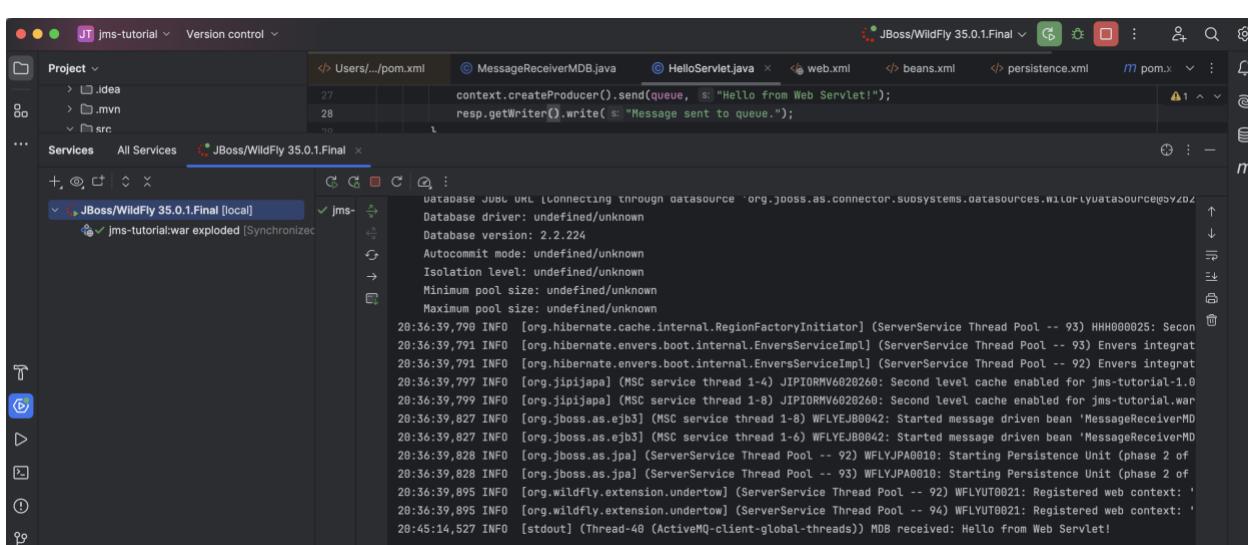


Apply your changes and run the application.

To test the application type in web browser: <http://localhost:8080/jms-tutorial/send>



Producer will send the message and Consumer will receive it. The message sent will appear in the console.



Try it yourself:

- 1) Make a form with an input field.
- 2) Bind that to enter dynamic message.
- 3) Add a label on the same page or add a new page to display the received message.
- 4) Test it with different input.
- 5) Take screenshots of the form, message sent, and message received.
- 6) Upload these screenshots at blackboard in Lab 2.
- 7) Push your code at GitHub.