



## Java Plays : Get Eclipse Ready for Development Lab

**Presented by:**

IBM

ECOD

**IBM Cloud**

## **AGENDA**

- ❑ Purpose of This Lab
- ❑ Pre-Requisites
- ❑ Exercise 1: Install Postman
- ❑ Exercise 2: Install WAS Liberty Eclipse Plugin
- ❑ Exercise 3: Install IBM Bluemix Eclipse Plugin
- ❑ Exercise 4: Clone and Import Project from GitHub
- ❑ Exercise 5: Deploy Imported Project to WAS Liberty Server.

## *Purpose of This Lab*

In this lab, you will learn how to install and configure key tools working with Eclipse IDE.

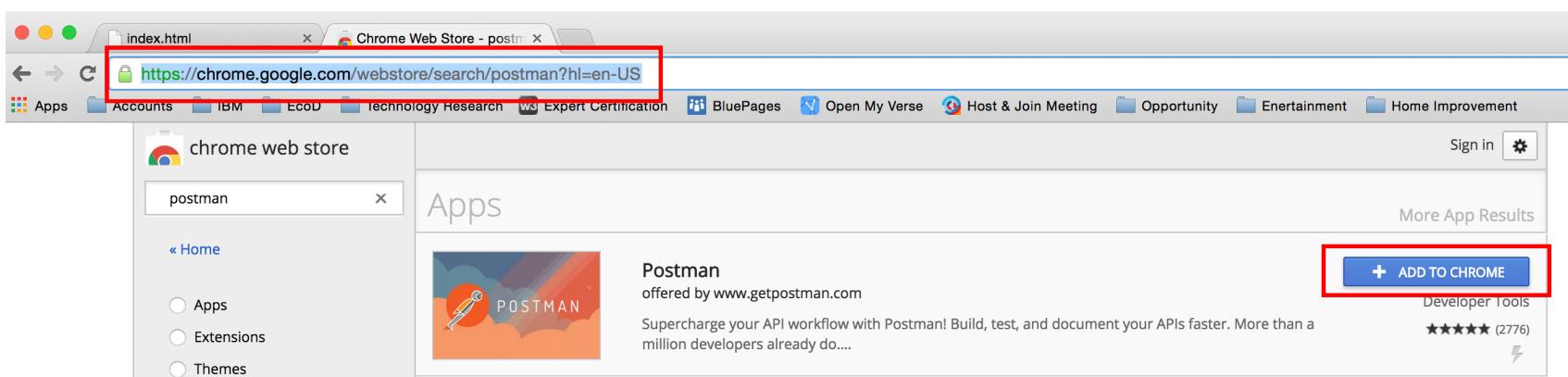
## *Pre-Requisite*

- Install Eclipse Mars  
<https://www.eclipse.org/downloads/>
- Install Java JDK 1.7 +  
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
- Chrome Browser  
[https://www.google.com/chrome/browser/desktop/ !!](https://www.google.com/chrome/browser/desktop/)

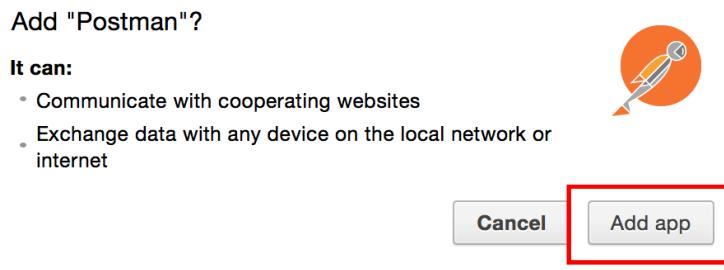
## *Exercise 1: Install Postman*

Postman helps build, test, document APIs faster. In this lab, we will use the tool to test a REST API.

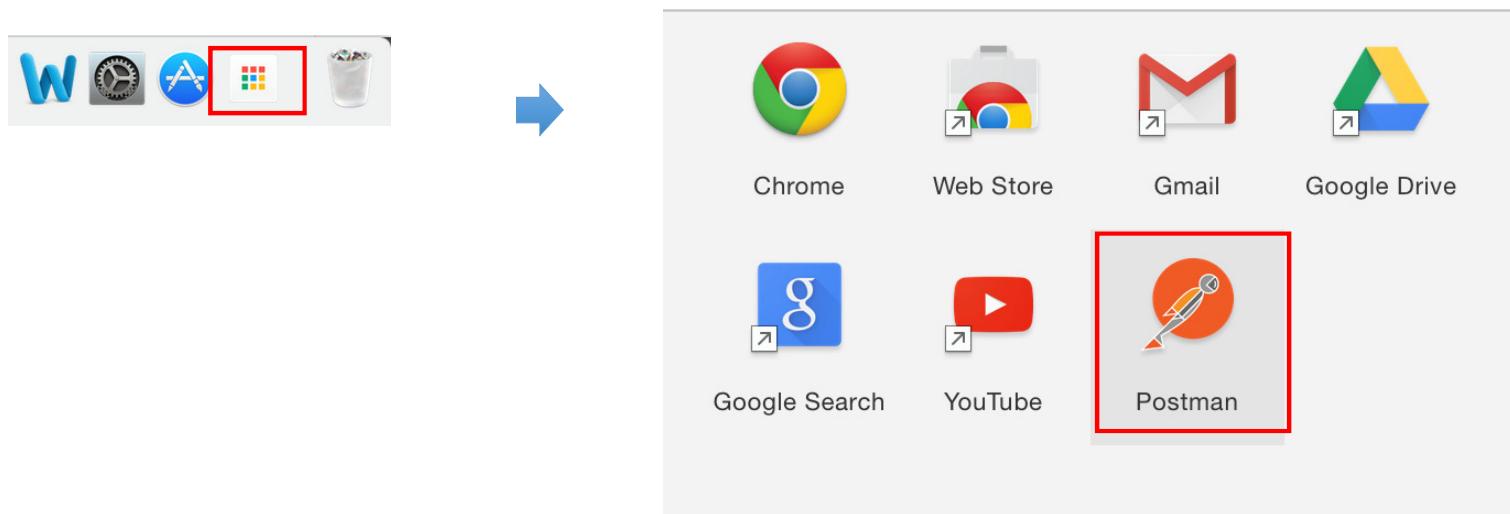
- ❑ Open Chrome Browser.
- ❑ Type in <https://chrome.google.com/webstore/search/postman?hl=en-US> at the address bar.
- ❑ Click Enter.
- ❑ Click + Add to Chrome.



Click Add App.



Launch Chrome App Launcher. >> Click Postman to start.



Java Plays: Get Eclipse Ready for Development Lab

## *Exercise 2: Install WAS Liberty Eclipse Plugin*

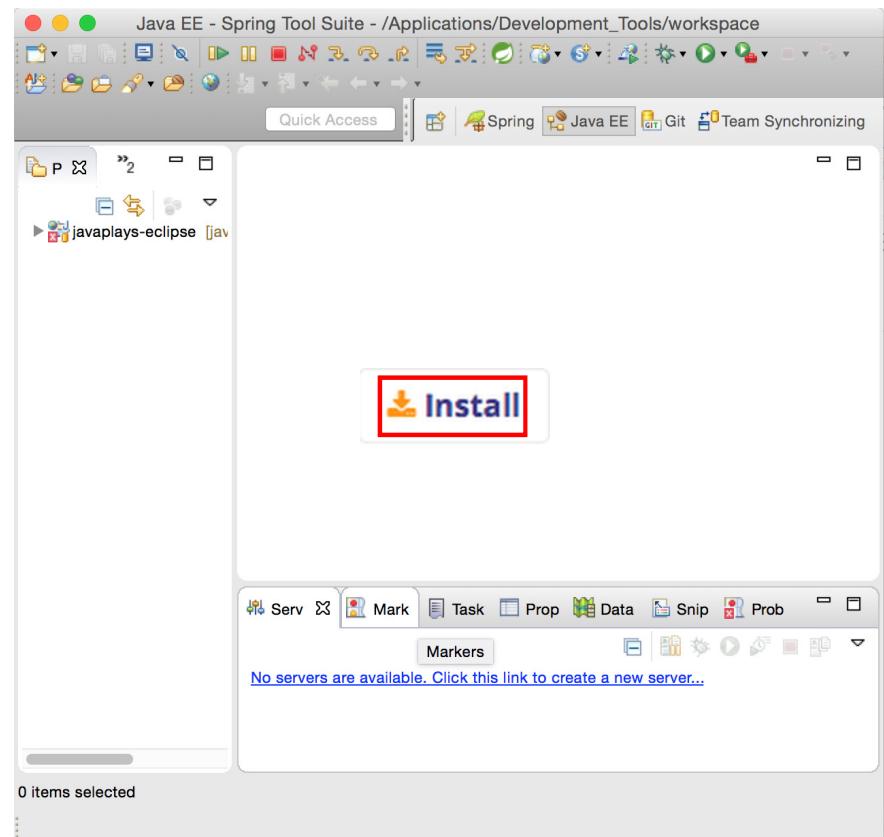
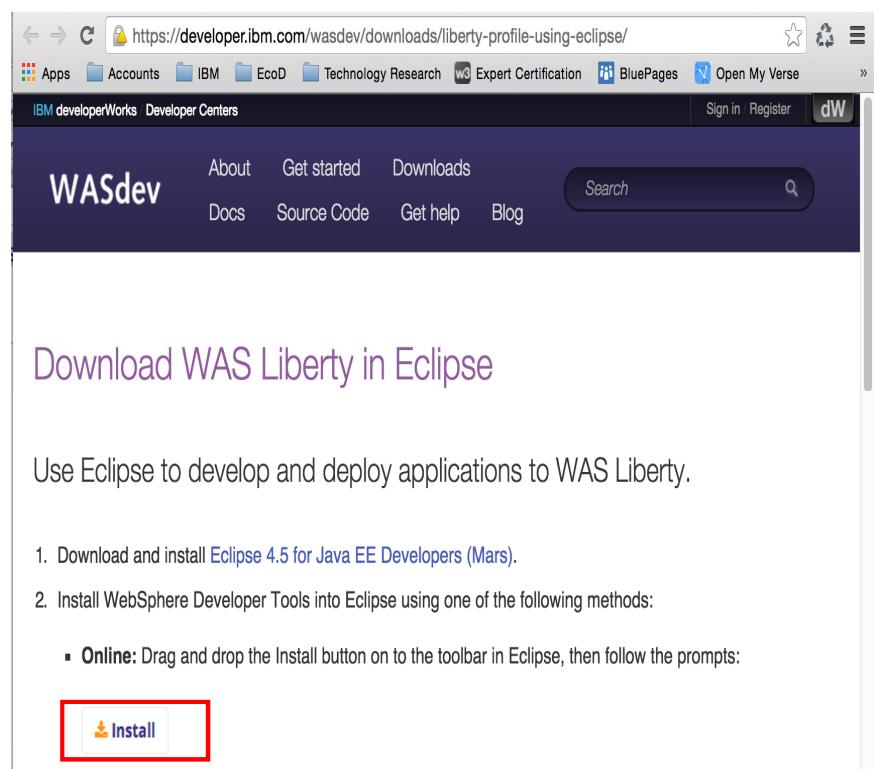
### **Install WAS Liberty Plugin**

Websphere Application Server Liberty profile provides an server runtime environment for an application to run. It not only provides a light weight server and Java runtime environment on Bluemix but also enables Monitoring and Analytics service without additional coding requirement as an example.

There are two ways to install. One is via Eclipse Market search in Eclipse IDE. The other way is drag and drop into Eclipse directly from the download site. We will use the download site option.

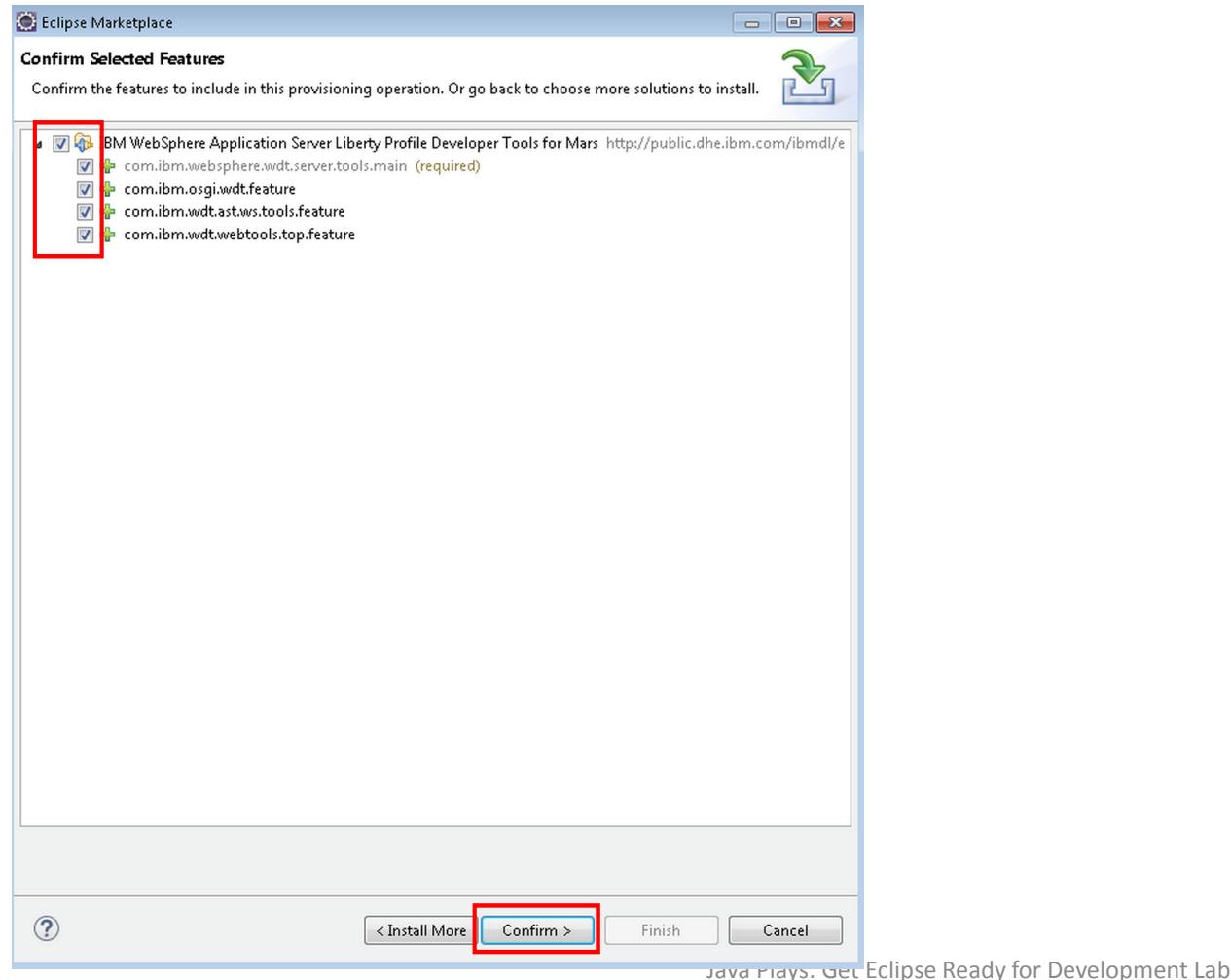
- Open browser and type in <https://developer.ibm.com/wasdev/downloads/liberty-profile-using-eclipse/>
- Click Enter.

- Click and Hold down the Install button. >> Drag and Drop into Eclipse IDE.



Java Plays: Get Eclipse Ready for Development Lab

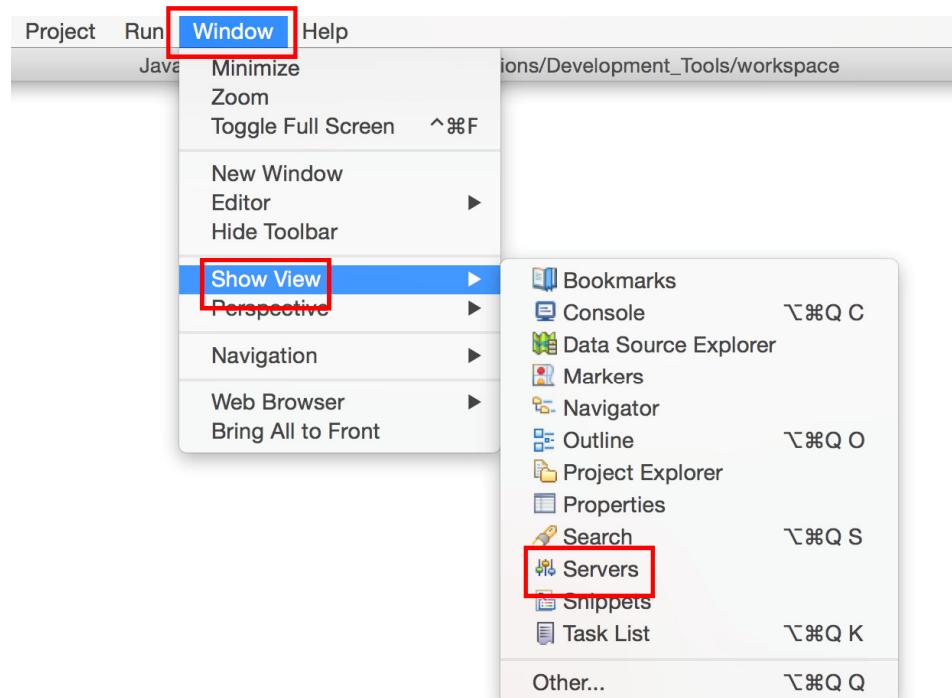
- ❑ Eclipse Market Place panel shows WAS liberty server profile features. Select All. >> Confirm.



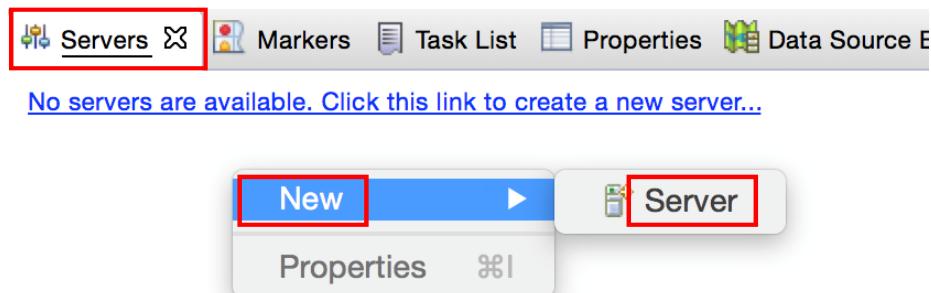
- Accept terms of the license agreements. >> Finish.
- Restart Eclipse when the installation completes successfully.

## Configure WAS Liberty Server

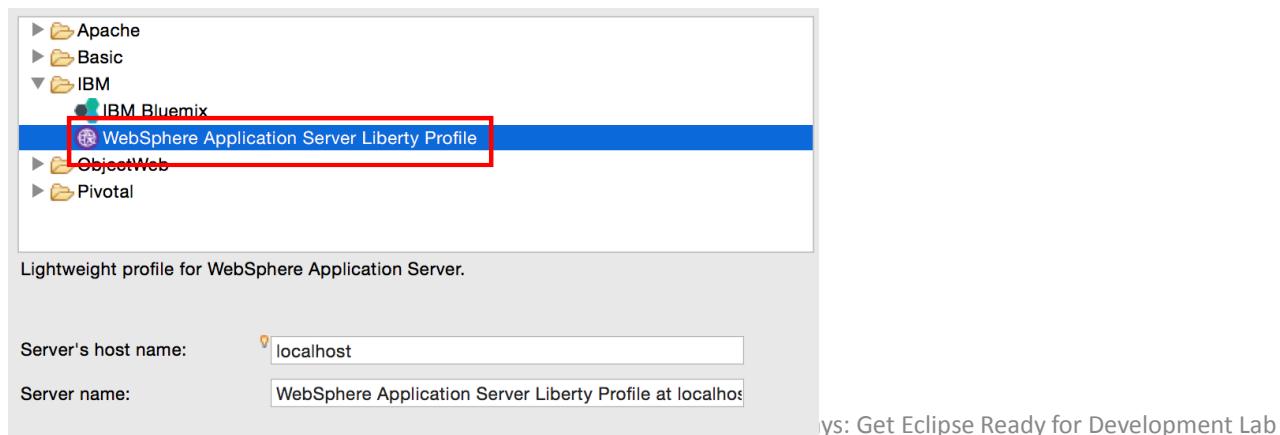
- ❑ Navigate to Eclipse.
- ❑ Go to top navigation menu. >> Click Window. >> Click Show View. >> Select Servers.



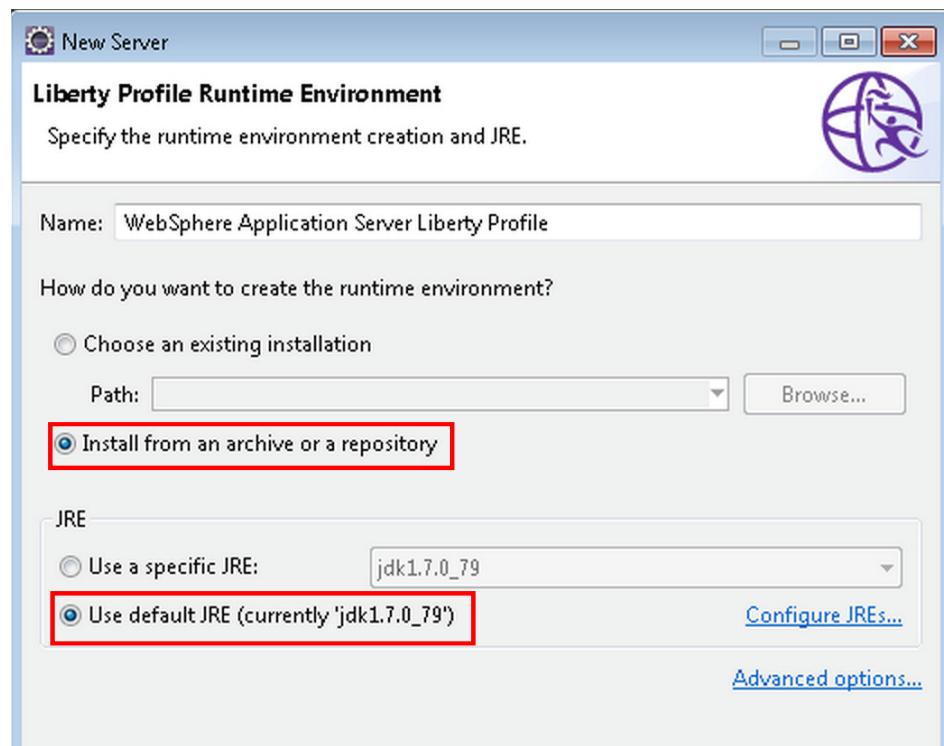
- ❑ Right click in Servers view.
- ❑ Click New. >> Select Server.



- ❑ In New Server panel, Open IBM folder. >> Select Websphere Application Server Liberty Profile.

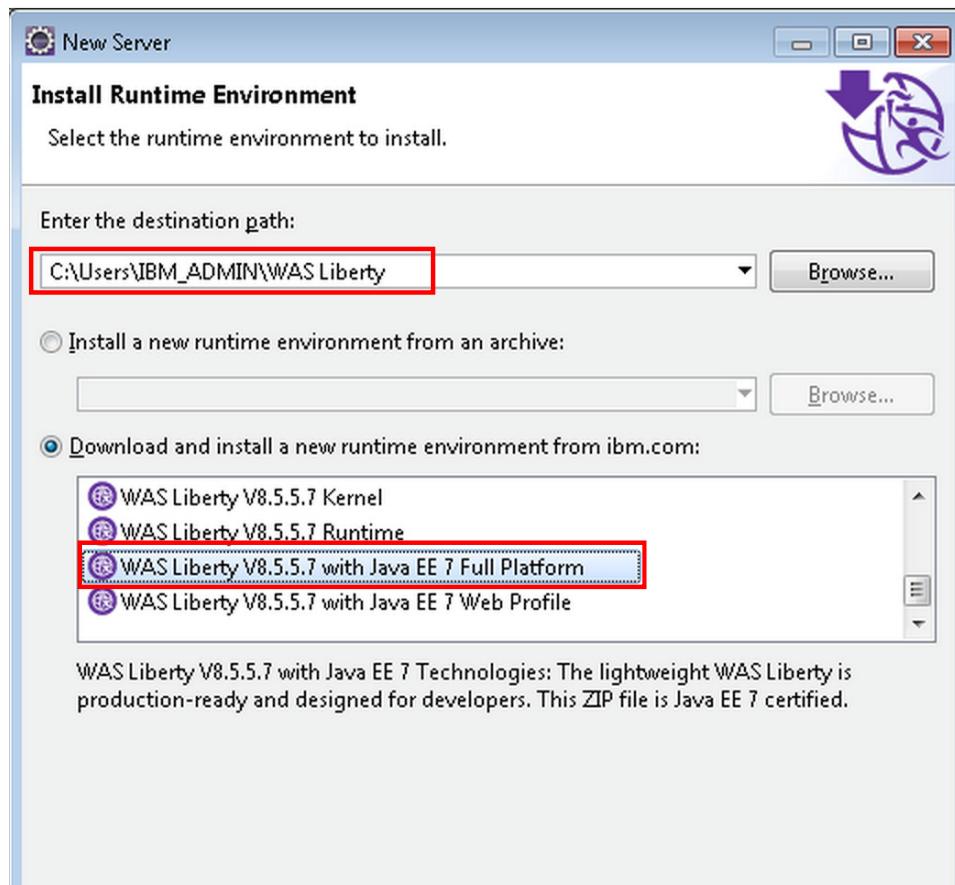


- ❑ Select Install from an archive or a repository.
- ❑ Select Use default JRE.
- ❑ Click Next.



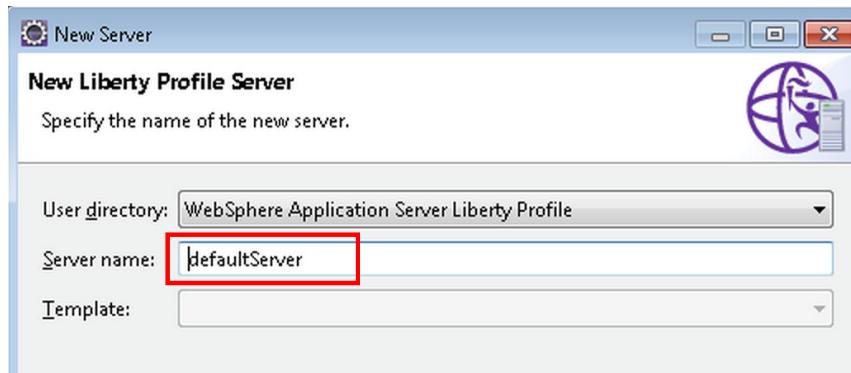
Java Plays: Get Eclipse Ready for Development Lab

- ❑ Enter where the downloaded server runtime will be stored. An example is C:\Users\IBM\_ADMIN\WASLiberty.
- ❑ Select WAS Liberty v8.5.5.7 with Java EE 7 Full Platform. >> Click Next.



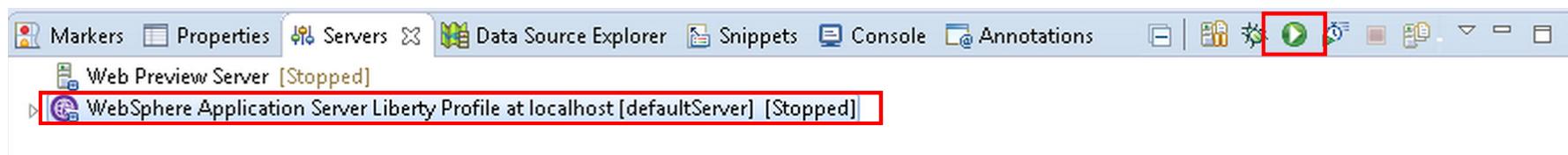
Java Plays: Get Eclipse Ready for Development Lab

- ❑ Click Next.
- ❑ Accept the terms of all the license agreements.
- ❑ Click Next.
- ❑ Leave the server name as defaultServer.



- ❑ Click Finish.

- Click on the server profile created. >> Click Run.



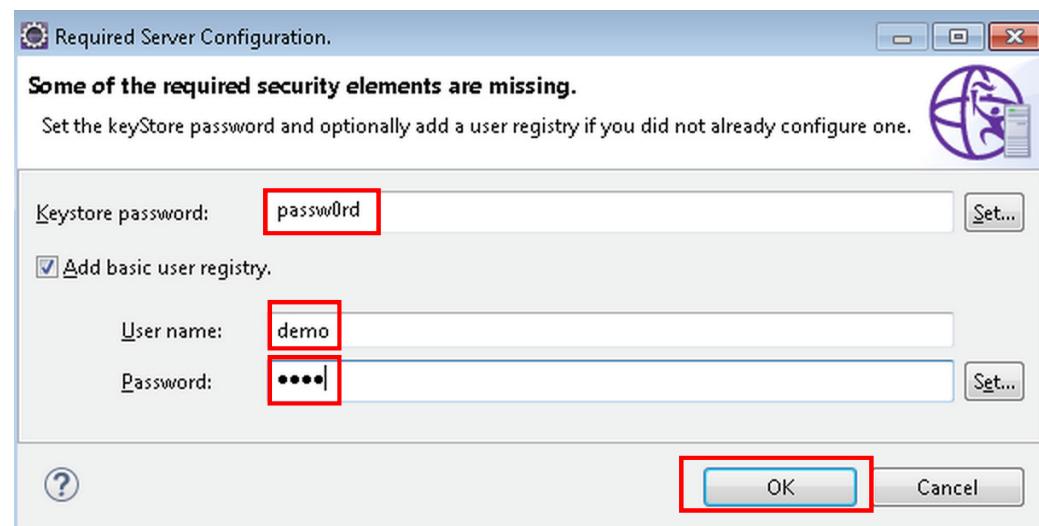
- Type in Keystore password: password

- Select Add basic user registry.

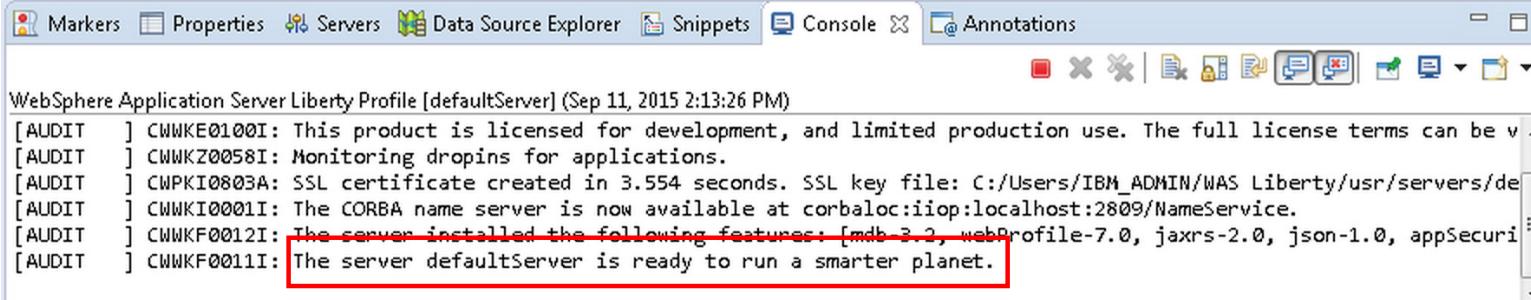
- Type in User name: demo

- Type in Password: demo

- Click OK.



- You have successfully configured WAS Liberty Server when you see the below message.



```
WebSphere Application Server Liberty Profile [defaultServer] (Sep 11, 2015 2:13:26 PM)
[AUDIT ] CWWKE0100I: This product is licensed for development, and limited production use. The full license terms can be v
[AUDIT ] CWWKZ0058I: Monitoring dropins for applications.
[AUDIT ] CWPKI0803A: SSL certificate created in 3.554 seconds. SSL key file: C:/Users/IBM_ADMIN/WAS Liberty/usr/servers/de
[AUDIT ] CWWKI0001I: The CORBA name server is now available at corbaloc:iiop:localhost:2809/NameService.
[AUDIT ] CWWKF0012I: The server installed the following features: [mdb-3.2, webProfile-7.0, jaxrs-2.0, json-1.0, appSecuri
[AUDIT ] CWWKF0011I: The server defaultServer is ready to run a smarter planet.
```

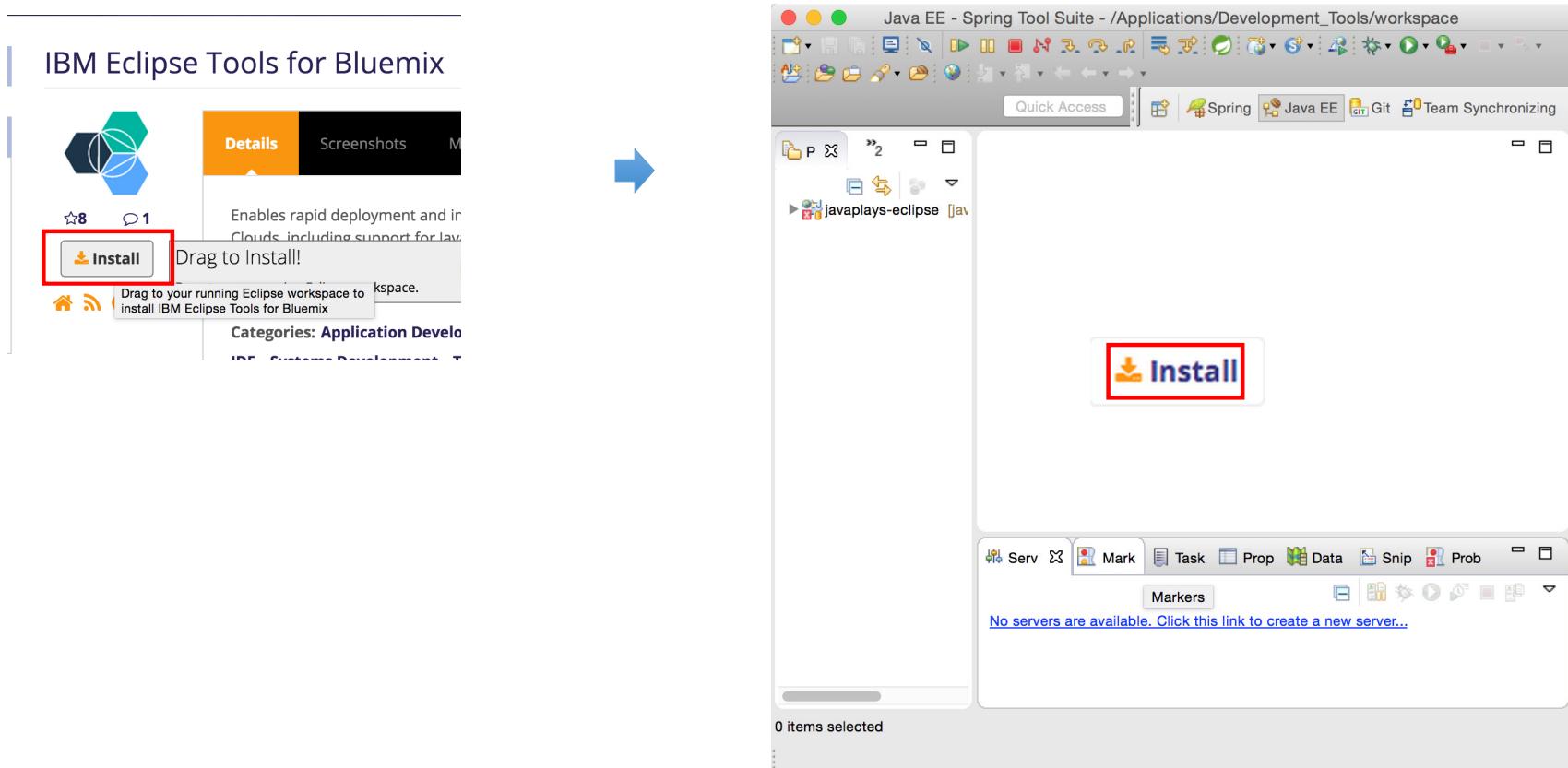
## *Exercise 3: Install IBM Bluemix Eclipse Plugin*

### **Install IBM Bluemix Eclipse Plugin**

Bluemix Eclipse plugin is built on Cloud Foundry APIs to enable easy management and deploy of application(s) to Bluemix via Eclipse.

- Open browser.
- Type in <https://marketplace.eclipse.org/content/ibm-eclipse-tools-bluemix>
- Click Enter.

- ❑ Click and hold down on Install button.
- ❑ Drag and drop into Eclipse.



Java Plays: Get Eclipse Ready for Development Lab

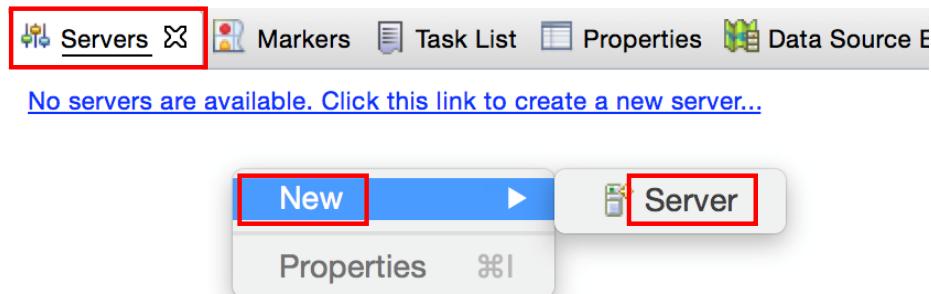
- ❑ Select all the features. >> Click Finish.



- ❑ Restart Eclipse when the installation completes successfully.

## Configure IBM Bluemix Server

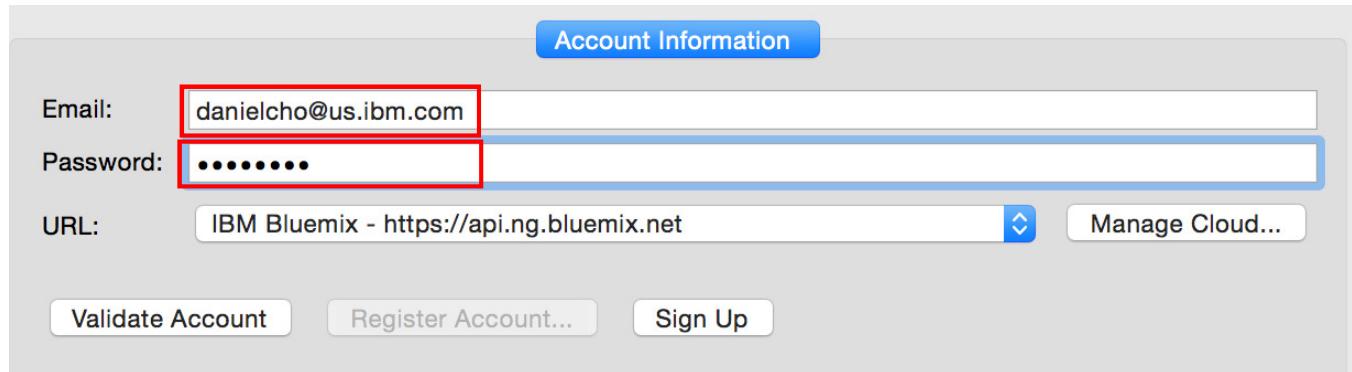
- ❑ Right click in Servers view.
- ❑ Click New. >> Select Server.



- ❑ In New Server panel, Open IBM folder. >> Select IBM Bluemix.

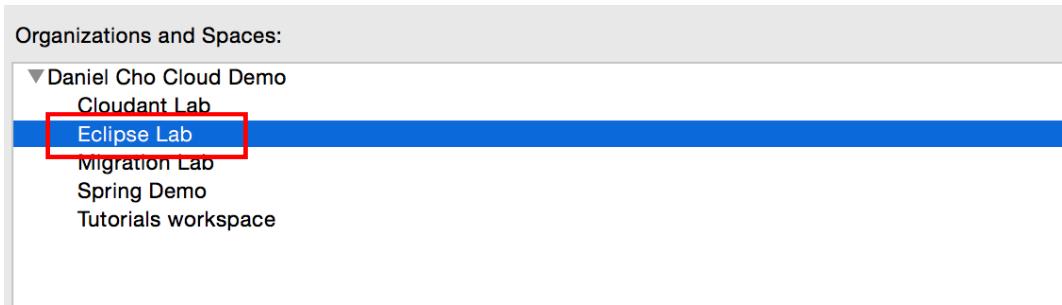


- ❑ Enter your Bluemix User ID next to Email and Password.

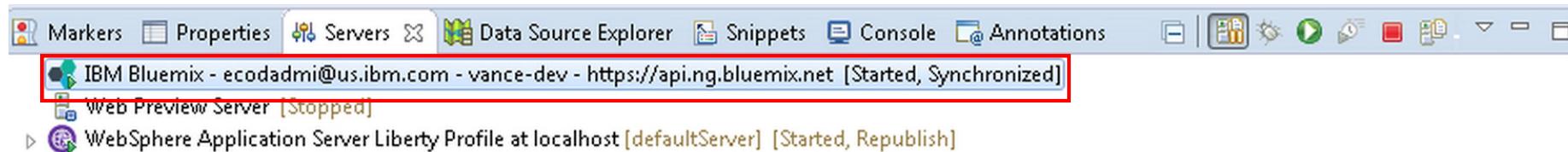


- ❑ Select URL, a target bluemix region where you plan to deploy your application to.
- ❑ Validation Account to ensure your Bluemix account is valid and allow Eclipse to download your organization and spaces defined from Bluemix.
- ❑ Click Next.

- Select a space in your Bluemix organization where you plan to deploy an application to.



- Click Finish.
- You should see IBM Bluemix server started successfully.

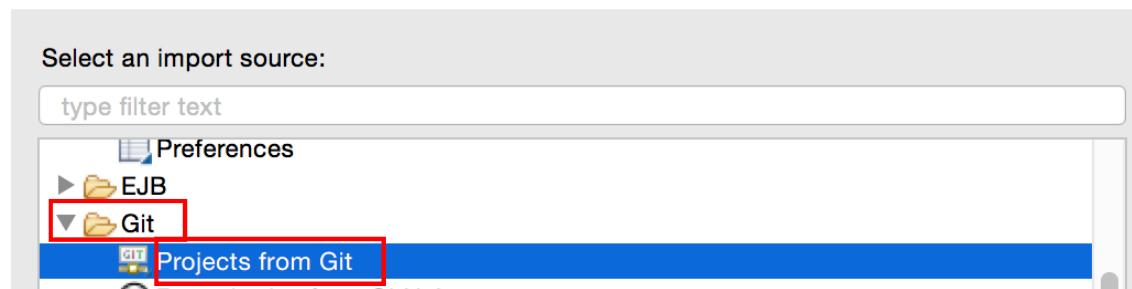


Java Plays: Get Eclipse Ready for Development Lab

## *Exercise 4: Clone and Import Project from GitHub*

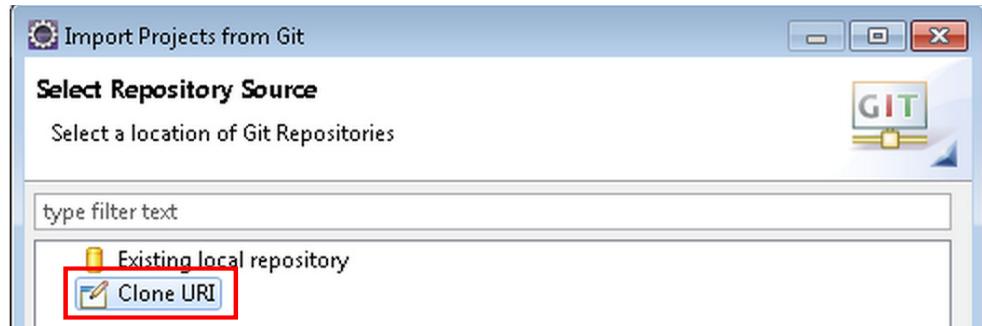
GitHub is a source control management repository where you can manage project source via Eclipse Git client plugin. In this section, we will learn how to check out a code project from GitHub within Eclipse.

- ❑ Navigate to Eclipse.
- ❑ Click File. >> Import. >> Expand Git. >> Select Project from Git.

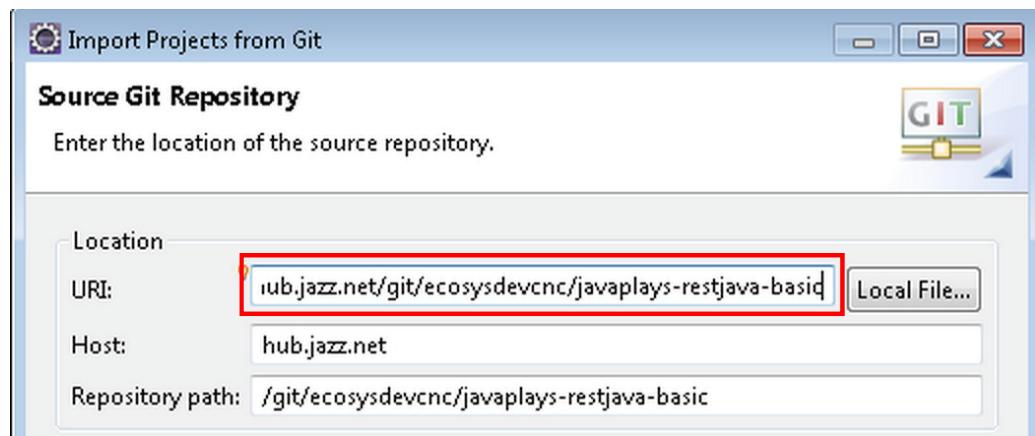


- ❑ Click Next.

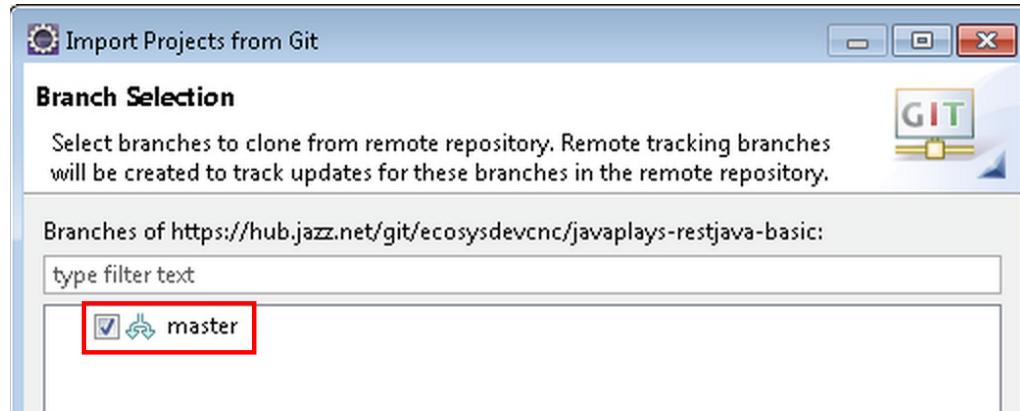
- Select Clone URI. >> Click Next.



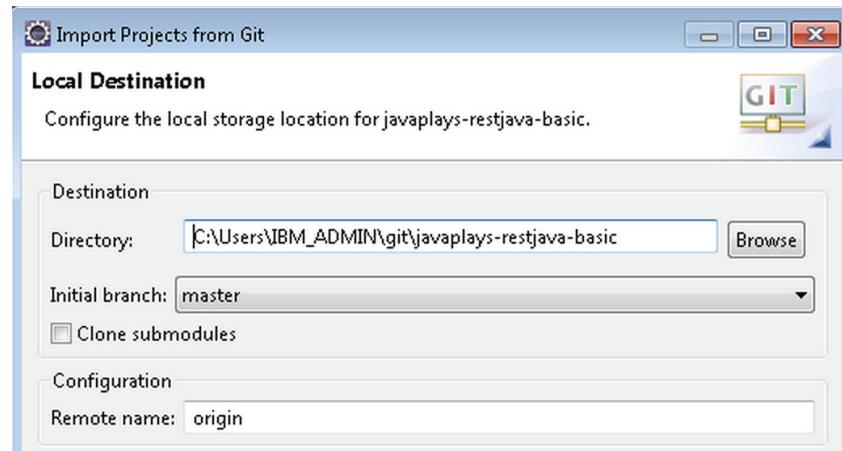
- Enter `https://hub.jazz.net/git/ecosys devcnc/javaplays-restjava-basic` next to URI. >> Click Next.



- Ensure master branch is selected. >> Click Next.

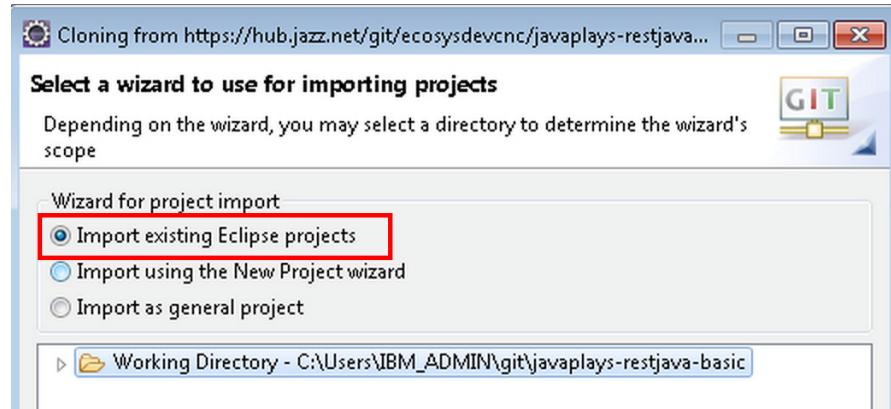


- Accept the default values. >> Click Next.

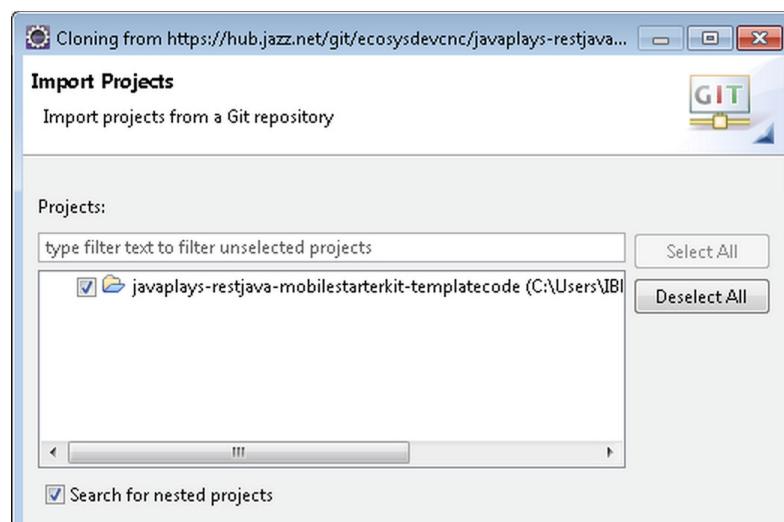


Java Plays: Get Eclipse Ready for Development Lab

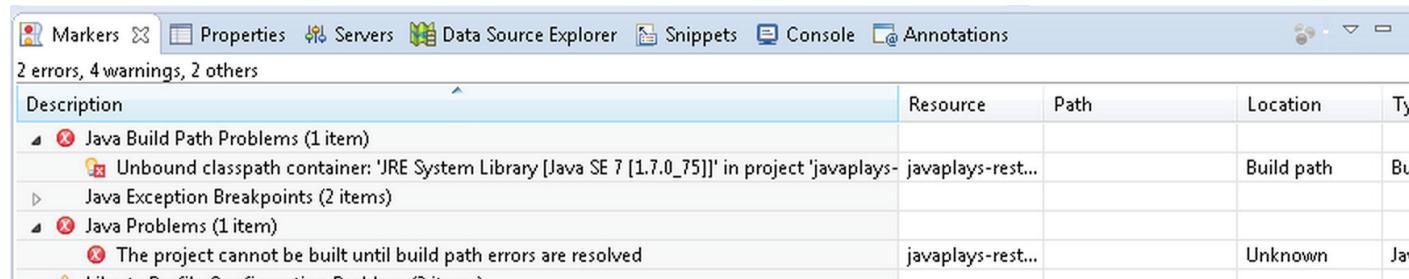
- ❑ Ensure Import Existing Eclipse Project option is selected. >> Click Next.



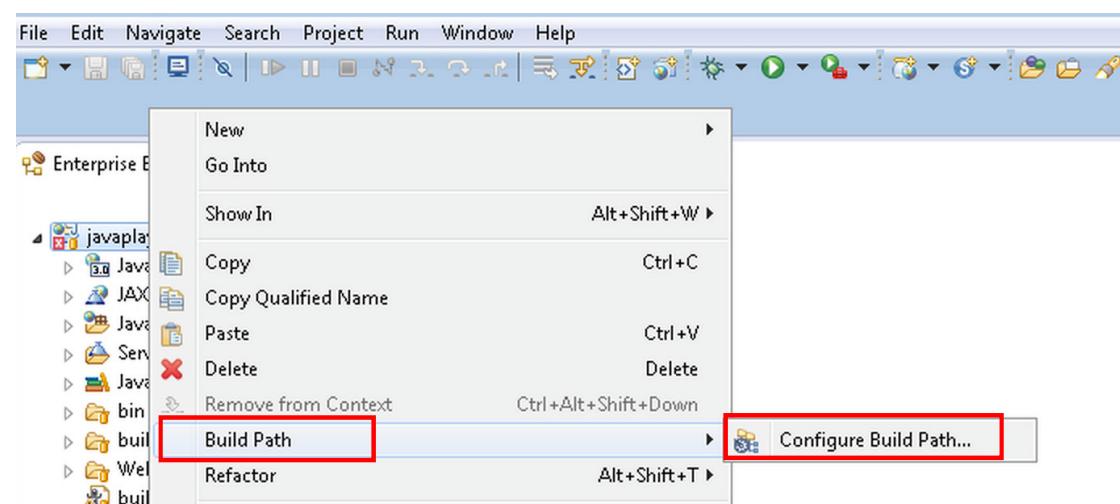
- ❑ Click Finish.



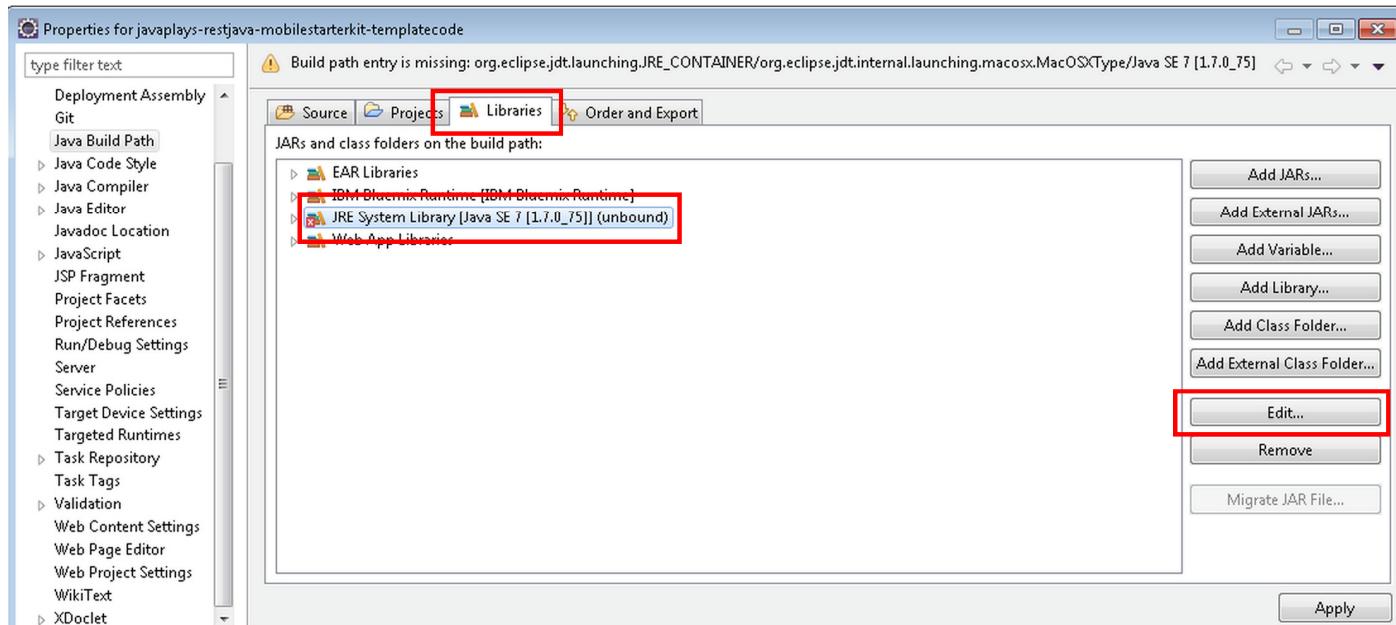
- You may have compile errors after importing a project is completed, which must be fixed.



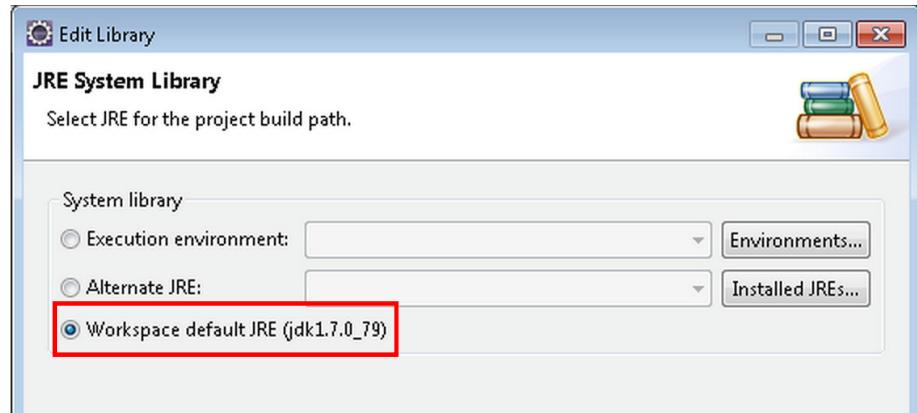
- Right click on the imported project. >> Hover over Build Path. >> Click Configure Build Path.



□ Click Libraries tab. >> Select JRE System Library. >> Click Edit.



- ❑ Select Workspace default JRE. >> Click Finish.

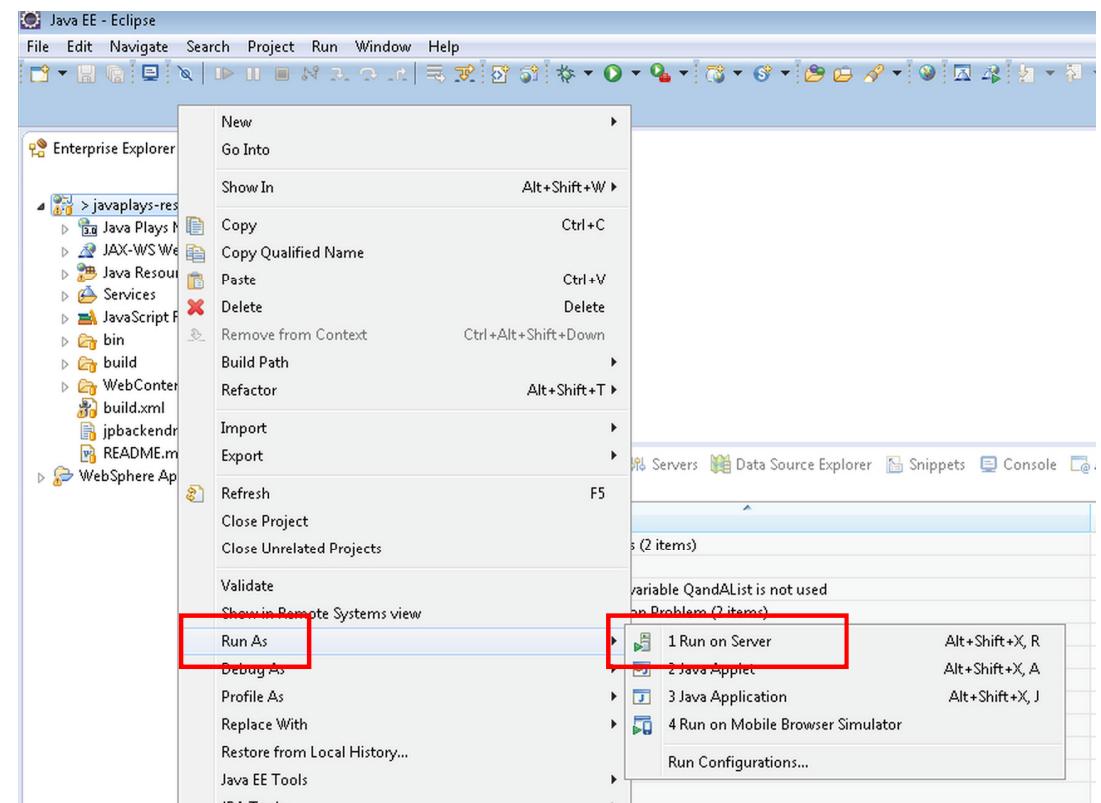


- ❑ Errors associated with incorrect JRE reference should be corrected by now.
- ❑ Click OK to close the active Java Build Path panel.

## *Exercise 5: Deploy the Imported Project to WAS Liberty Server*

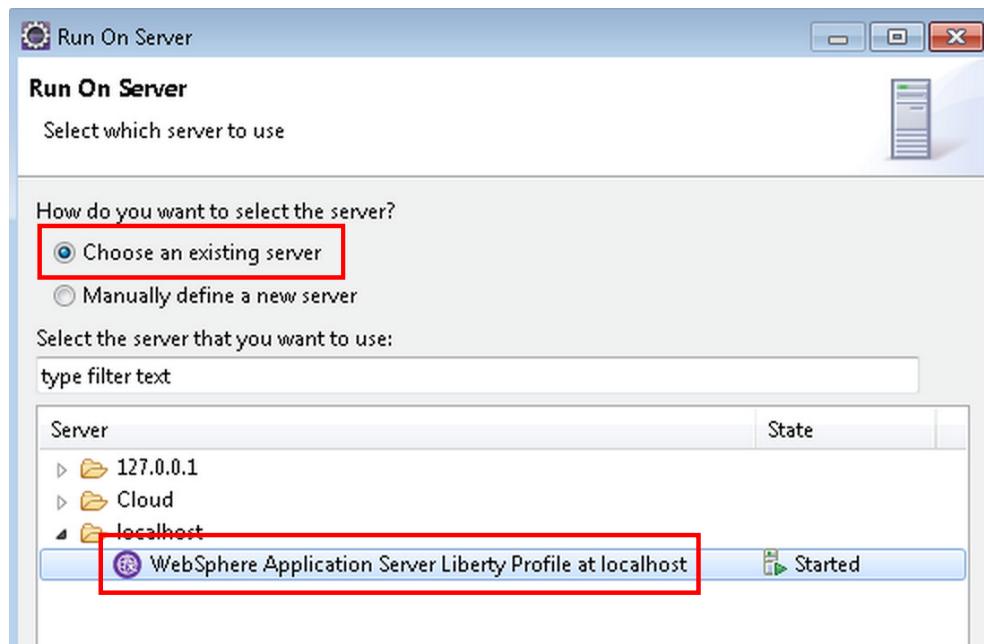
### Deploy Application to Local WAS Liberty Server

- ❑ Right Click on the code project.
- ❑ Click Run As. >> Select Run on Server.

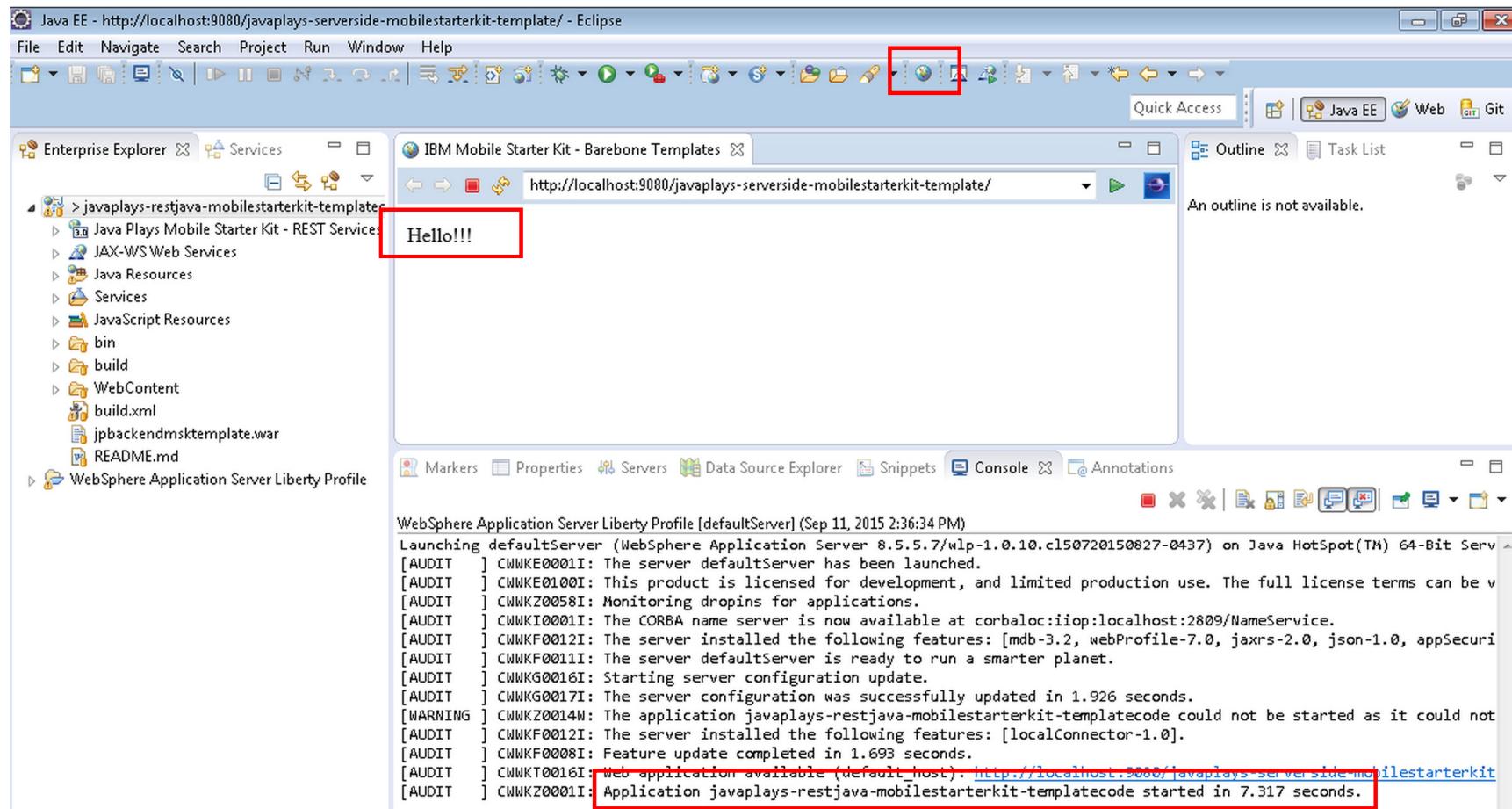


Java Plays: Get Eclipse Ready for Development Lab

- ❑ Select Choose an existing server.
- ❑ Select Websphere Application Server Liberty Profile at localhost.
- ❑ Click Finish.



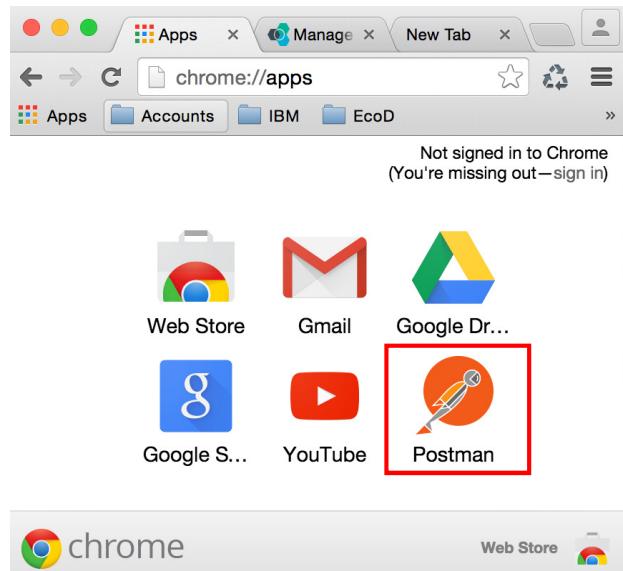
## □ Eclipse embedded web browser should display, Hello !!!



Java Plays: Get Eclipse Ready for Development Lab

## Test Local Application with Postman

- ❑ Open Google App Launcher. >> Click Postman.



- ❑ Change HTTP request mode to POST.
- ❑ Enter URL for a target resource: <http://localhost:9080/javaplays-serverside-mobilestarterkit-template/api/service/login>

The screenshot shows the Postman application interface. At the top, there are tabs for 'Builder' (highlighted in orange) and 'Runner'. Below the tabs are buttons for 'Import' and 'Disabled'. The URL bar contains 'http://localhost:9080...'. The main area shows a 'POST' request to 'http://localhost:9080/javaplays-serverside-mobilestarterkit-template/api/service/login'. Below the request, tabs for 'Authorization', 'Headers (0)', 'Body', 'Pre-request script', and 'Tests' are visible, with 'Authorization' being the active tab. A dropdown menu for 'No Auth' is open. On the right side, there are 'Params' and 'Tests' sections.

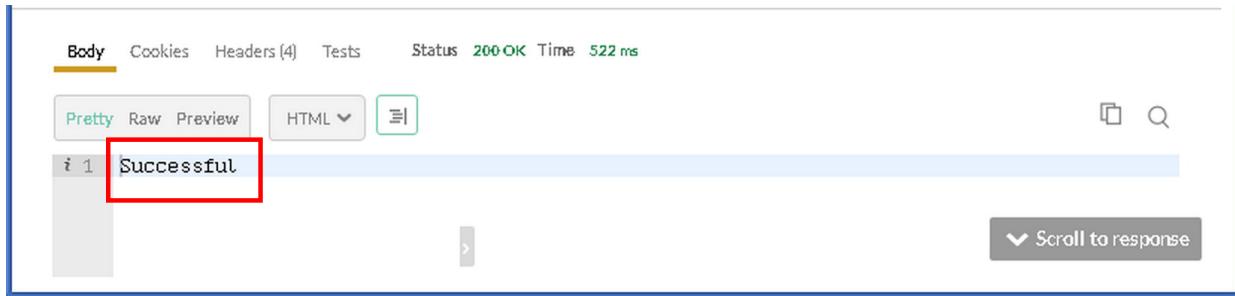
- ❑ Click Body tab.
- ❑ Select raw option. >> Enter user\_id : admin, password : password as the below example. >> Click Send.

The screenshot shows the Postman interface again, with the 'Body' tab selected. The 'raw' option is selected under the body type dropdown. The JSON payload entered is:

```
1 [  
2   "user_id": "admin",  
3   "password": "password"  
4 ]
```

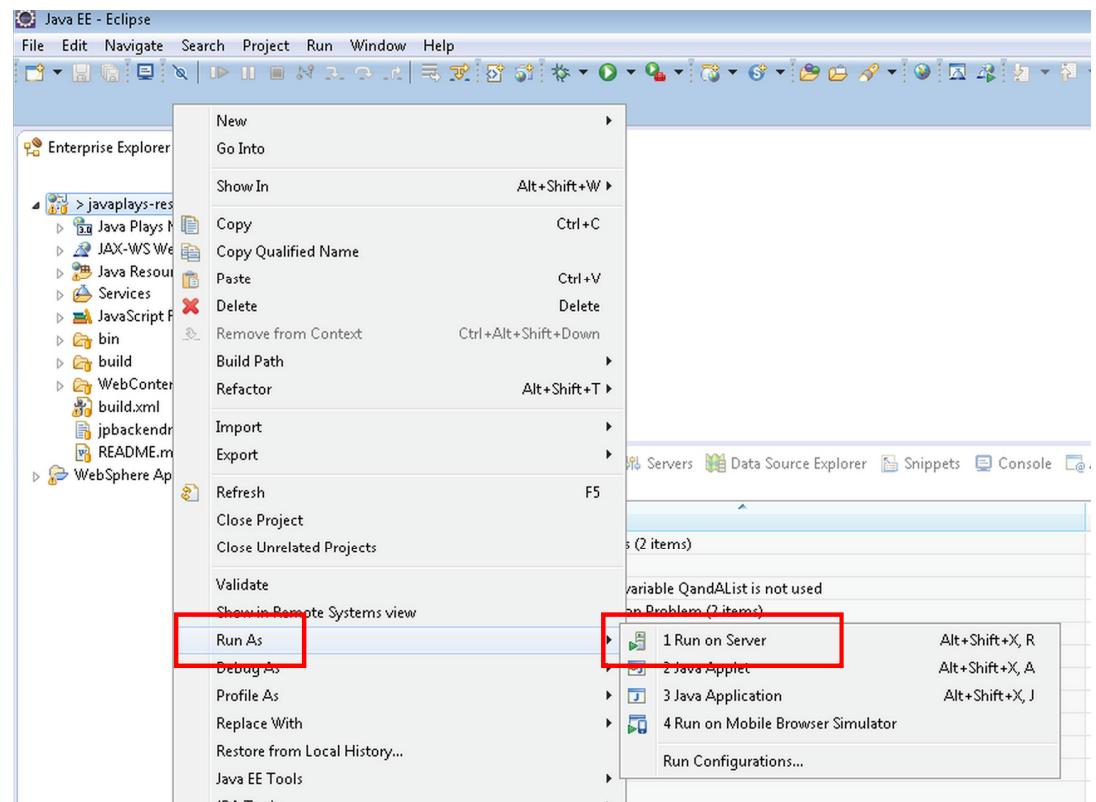
The 'Send' button is highlighted with a red box. The status bar at the bottom says 'Java Plays: Get Eclipse Ready for Development Lab'.

- You should see the Successful message at the bottom message panel.



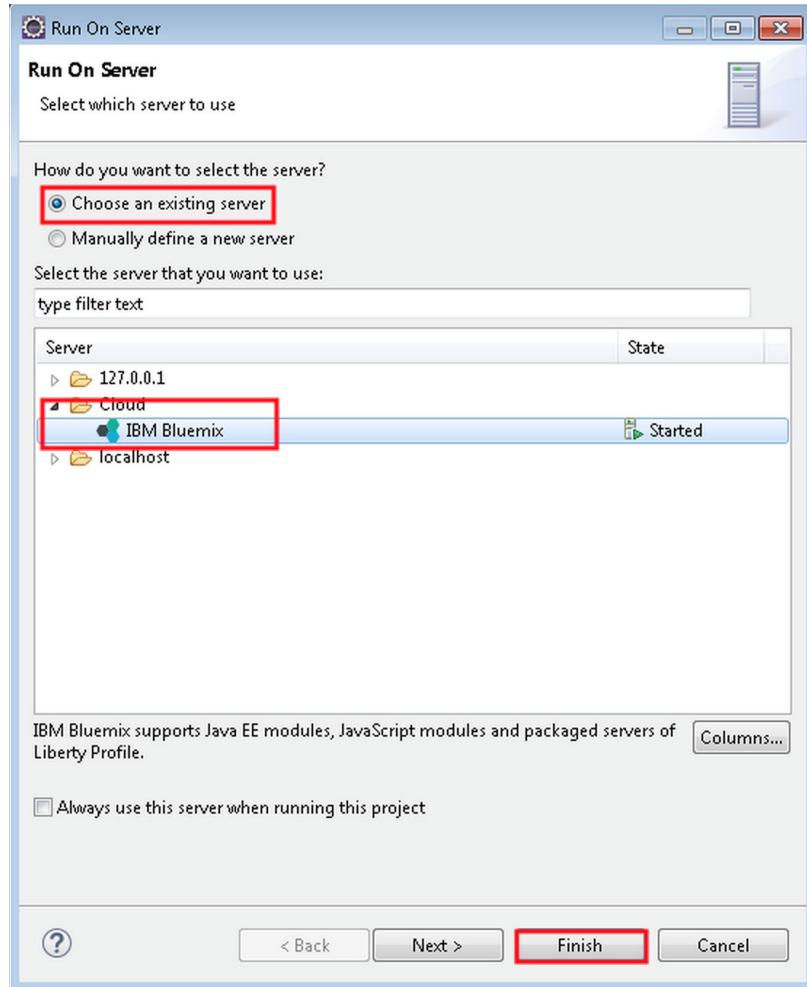
## Deploy Application to Bluemix, a Remote WAS Liberty Server

- ❑ Right Click on the code project.
- ❑ Click Run As. >> Select Run on Server.



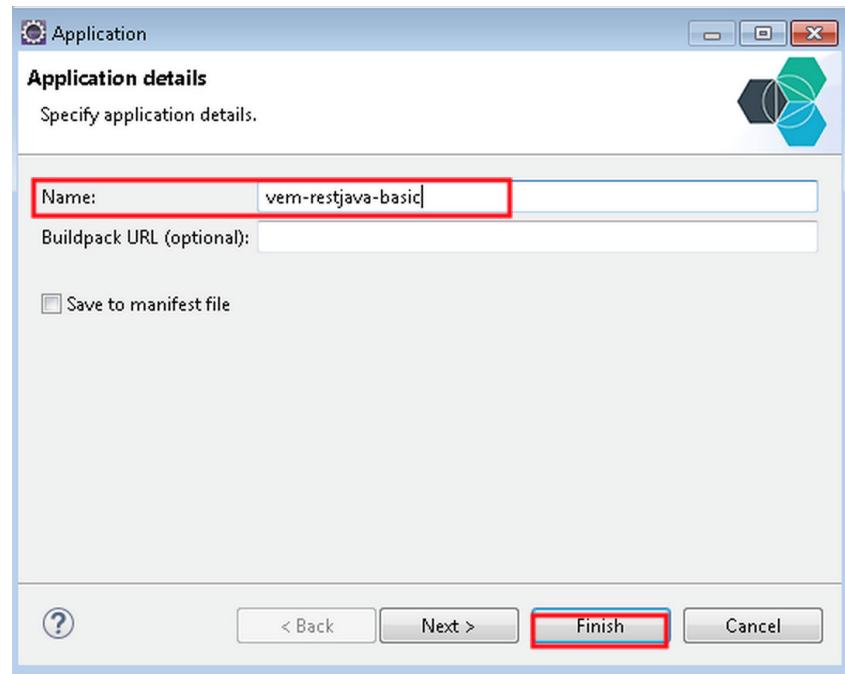
Java Plays: Get Eclipse Ready for Development Lab

- Select Choose from existing server. >> Select IBM Bluemix. >> Click Finish.

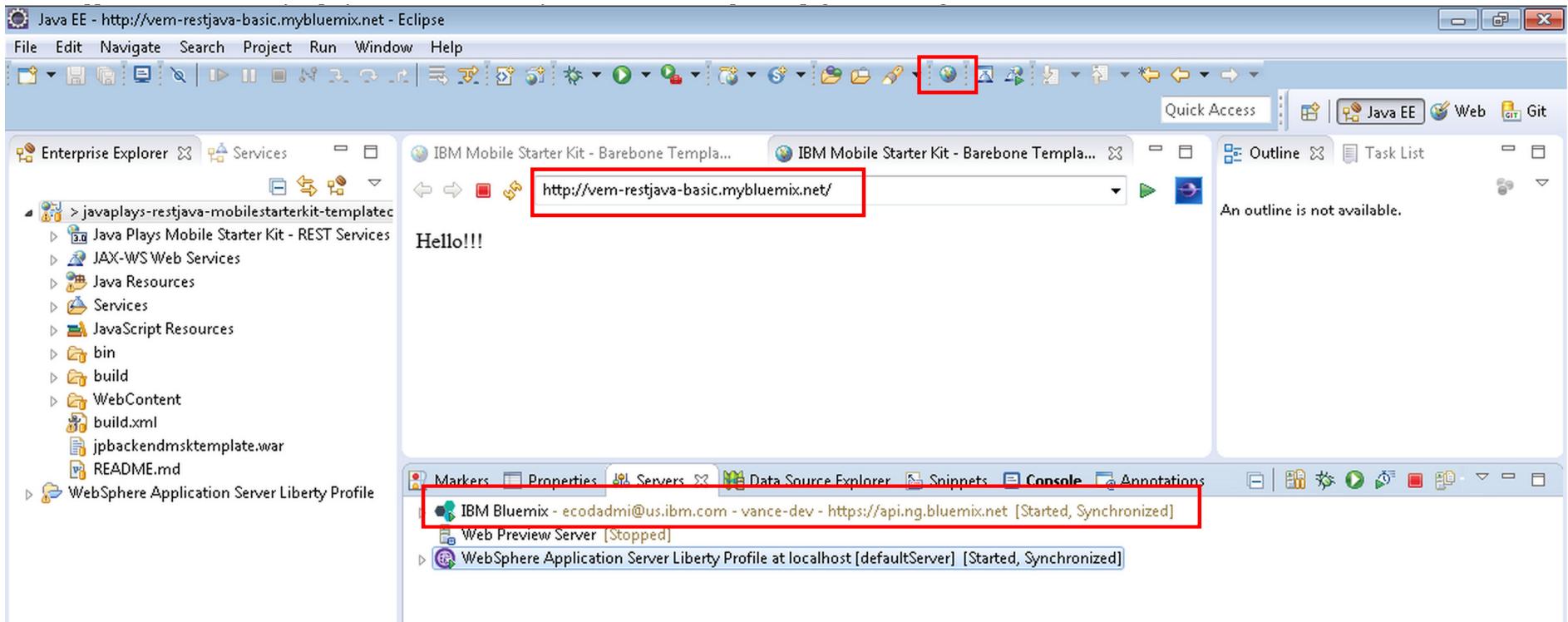


Java Plays: Get Eclipse Ready for Development Lab

- ❑ Give an Unique name to Application as it will be apart of your application's domain name.
- ❑ Click Finish.

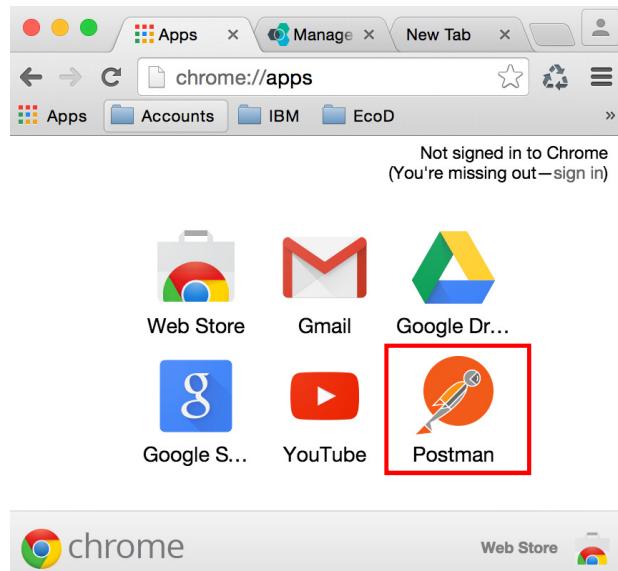


□ You will see Hello !!! when your application is deployed to Bluemix successfully.



## Test Remote Bluemix Application with Postman

- ❑ Open Google App Launcher. >> Click Postman.



- ❑ Change HTTP request mode to POST.
- ❑ Enter URL for a target resource: e.g. `http://<global-unique-name>.mybluemix.net/api/service/login`. <global-unique-name> is what you gave for the deployed application in a previous step.

Java Plays: Get Eclipse Ready for Development Lab

The screenshot shows the Postman application interface. At the top, there are tabs for 'Builder' (selected), 'Runner', and 'Import'. On the right, there are icons for 'Disabled' and a 'Run' button. Below the header, the URL 'http://vem-restjava...' is shown with a dropdown arrow and a '+' icon. The main area displays a POST request to 'http://vem-restjava-basic.mybluemix.net/api/service/login'. The 'Authorization' tab is selected, showing 'No Auth'. Other tabs include 'Headers (0)', 'Body', 'Pre-request script', and 'Tests'. A 'Params' tab is also visible.

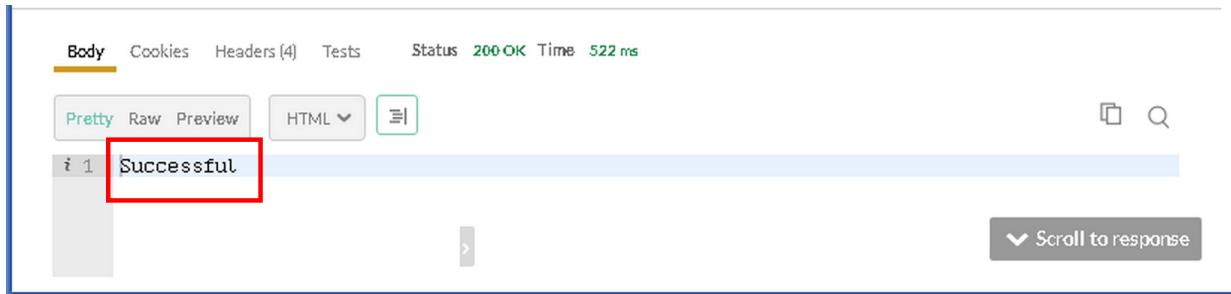
- ❑ Click Body tab.
- ❑ Select raw option. >> Enter user\_id : admin, password : password as the below example. >> Click Send.

The screenshot shows the Postman application interface with a POST request to 'http://vem-restjava-basic.mybluemix.net/api/service/login'. The 'Body' tab is selected, and the 'raw' option is chosen. The request body contains the following JSON:

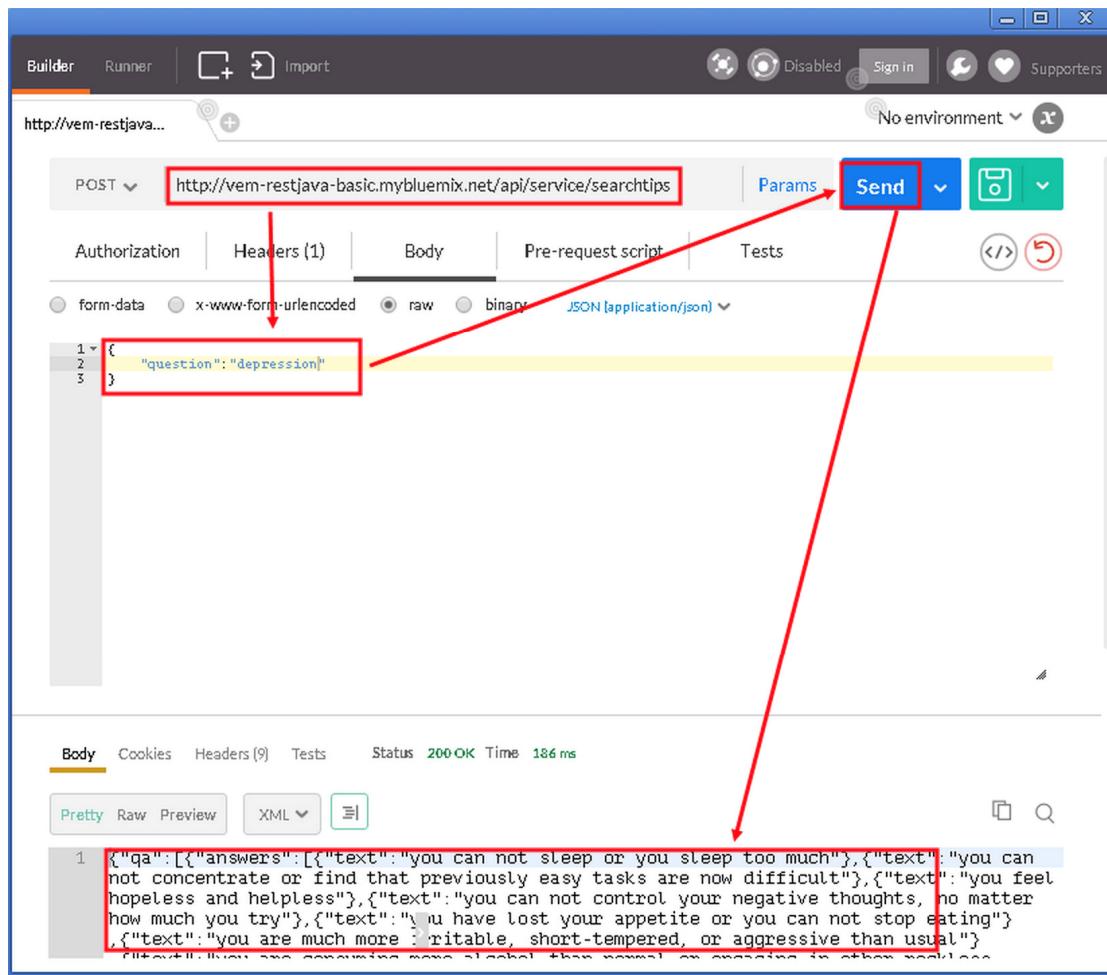
```
1 {
2   "user_id" : "admin",
3   "password" : "password"
4 }
5 }
```

Java Plays: Get Eclipse Ready for Development Lab

- You should see the Successful message at the bottom message panel.



- Try with another URL: <http://<global-unique-name>.mybluemix.net/api/service/searchtips>.
- Click Body tab.
- Select raw option.
- Add, "question" : "depression".
- Click Send.



Java Plays: Get Eclipse Ready for Development Lab

*This ends the Eclipse Tooling Lab1.*

*Thank you !*