WebSphere Application Server V8.5 Liberty Profile

Lab 2: First application

Module 1: Develop the code

LEARN

Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Worklight is a trademark registered in the United States to Worklight, an IBM company. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

- Other company products or service names may be trademarks or service marks of others.

- This document may not be reproduced in whole or in part without the prior written permission of IBM.

About IBM®

- See http://www.ibm.com/ibm/us/en/

About this lab

- In this lab, you will write a small web application and deploy it on IBM WebSphere Application Server V8.5 Liberty Profile

Prerequisites

- You must complete Lab 1 (WebSphere Application Server V8.5 Liberty Profile Install) before you can start this lab
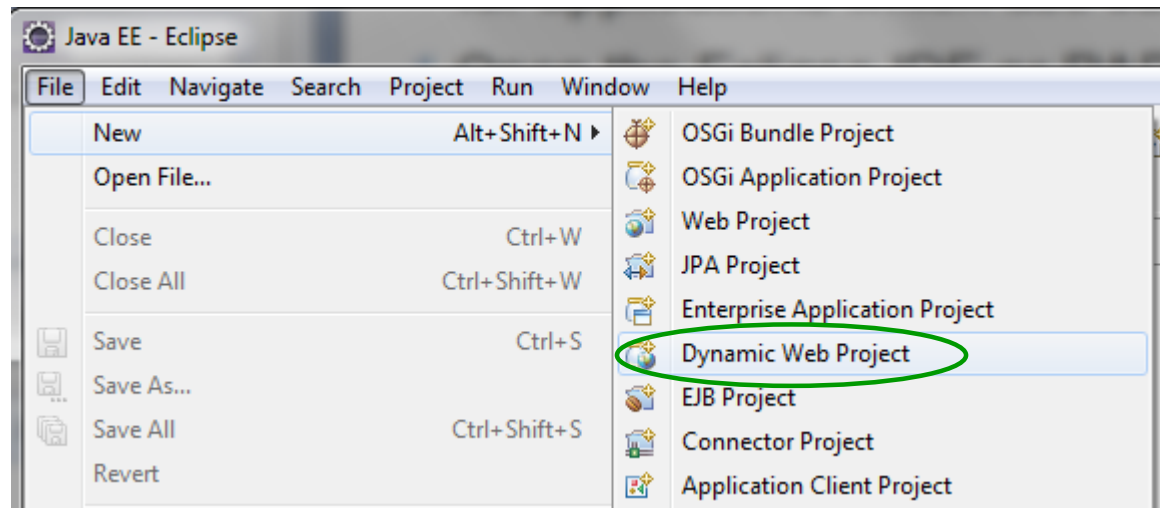
LEARN

About this module

- In this module, you will write a small web application containing a single HTTP servlet

- This servlet displays a "Hello World" message, the date, and the time

- Eclipse and RAD use projects to manage applications. In this module you will create an Eclipse dynamic web project which is pre-configured for servlets that use Java EE resources

LEARN

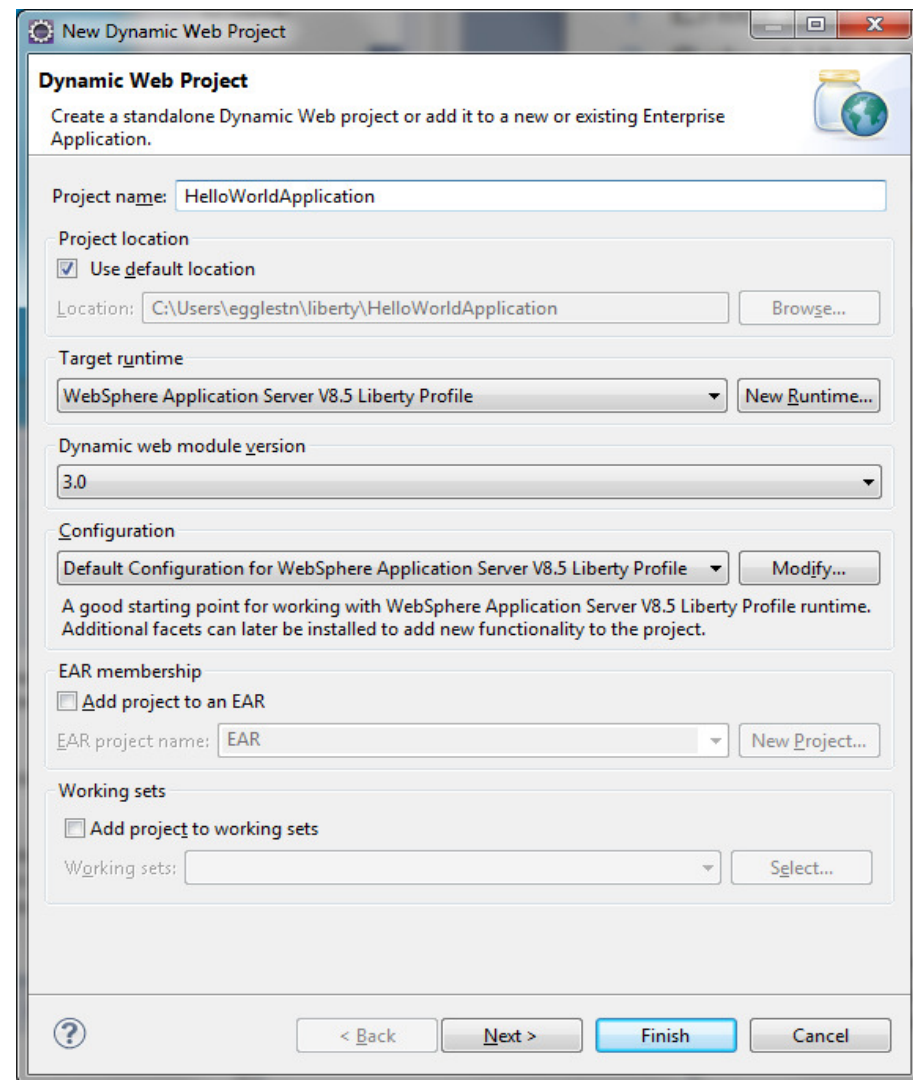# Create a dynamic web project

1 Open the Eclipse IDE or RAD workbench

2 Click File → New → Dynamic Web Project.

LEARN

Define the project

1 Enter the project name, for example HelloWorldApplication

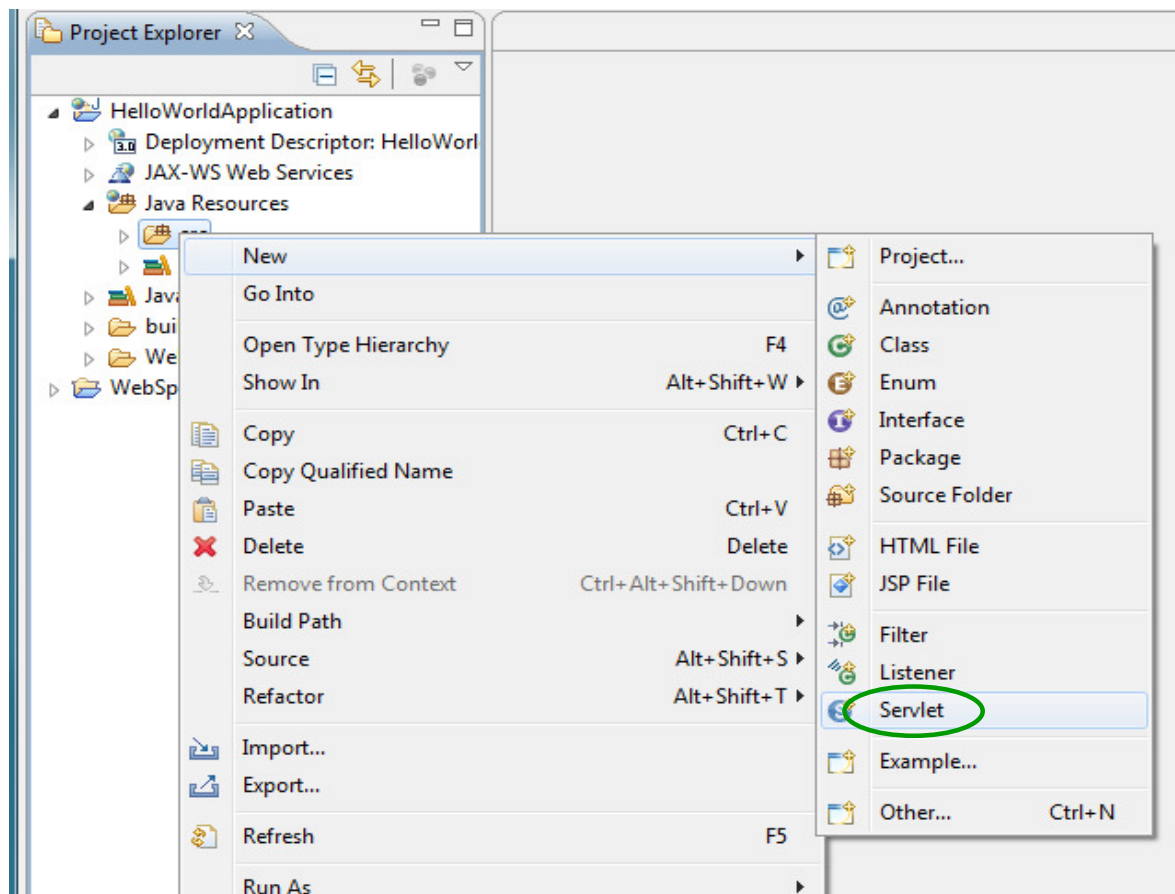2 Select WebSphere Application Server V8.5 Liberty Profile as the Target Runtime*

3 Click Finish

* This should be the Liberty runtime environment that you created in Lab 1, or if you prefer you can create a new runtime

Create a new servlet

1 Expand the new HelloWorldApplication → Java Resources
2 Right-click on src and select New → Servlet

LEARN

Define the new servlet

1 Enter

- Java package of my.application.helloWorld
- Class name of HelloWorldServlet

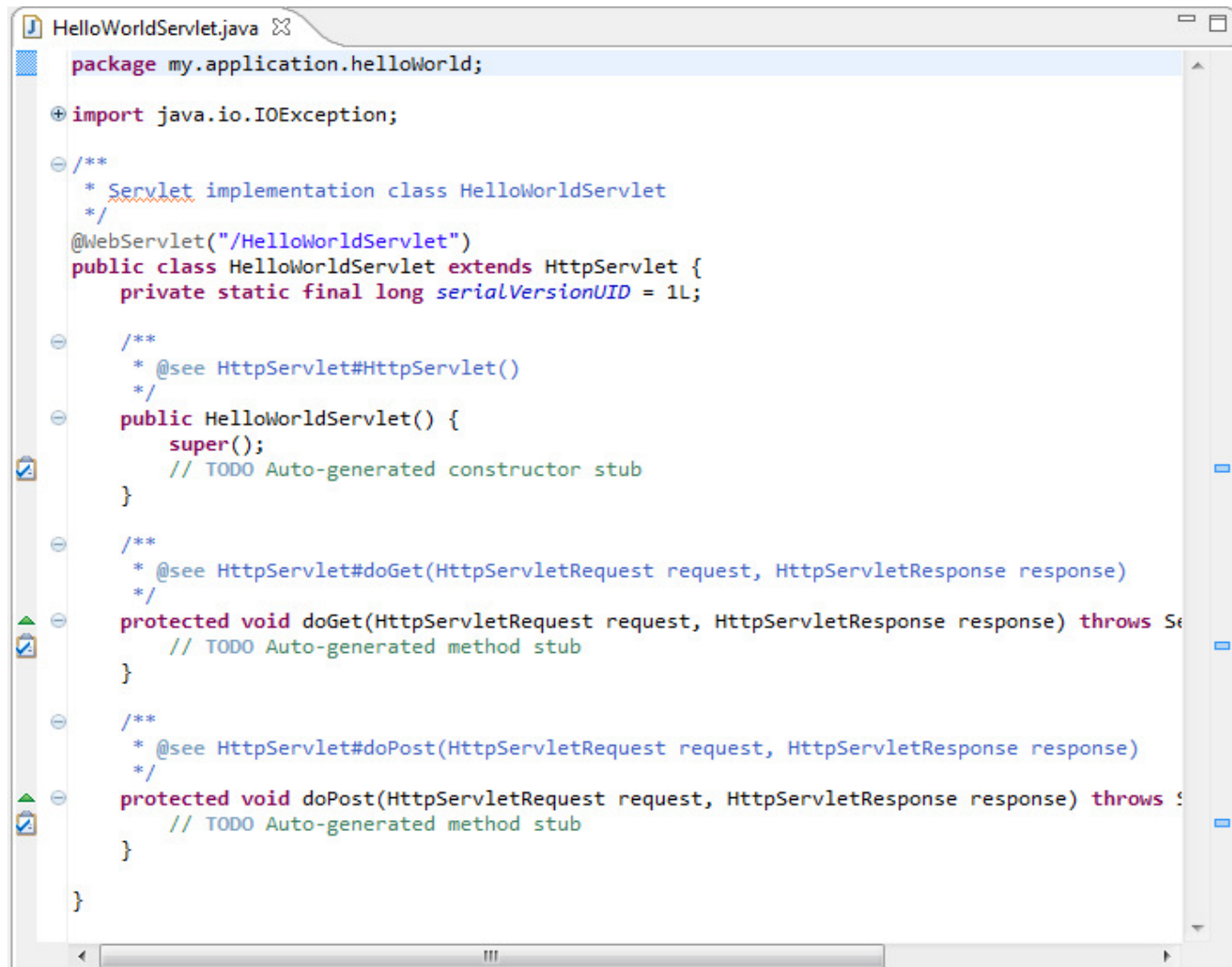2 Click Finish

Review the generated servlet

- The servlet code stubs are automatically generated and the source page opened so that you can review the contents


```java
package my.application.helloWorld;

import java.io.IOException;

/**
 * Servlet implementation class HelloWorldServlet
 */
@WebServlet("/HelloWorldServlet")
public class HelloWorldServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HelloWorldServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Se
        // TODO Auto-generated method stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws S
        // TODO Auto-generated method stub
    }

}
```

LEARN

Write the Hello World code in the servlet

1 In the HelloWorldServlet class, replace the doGet method with the following code:
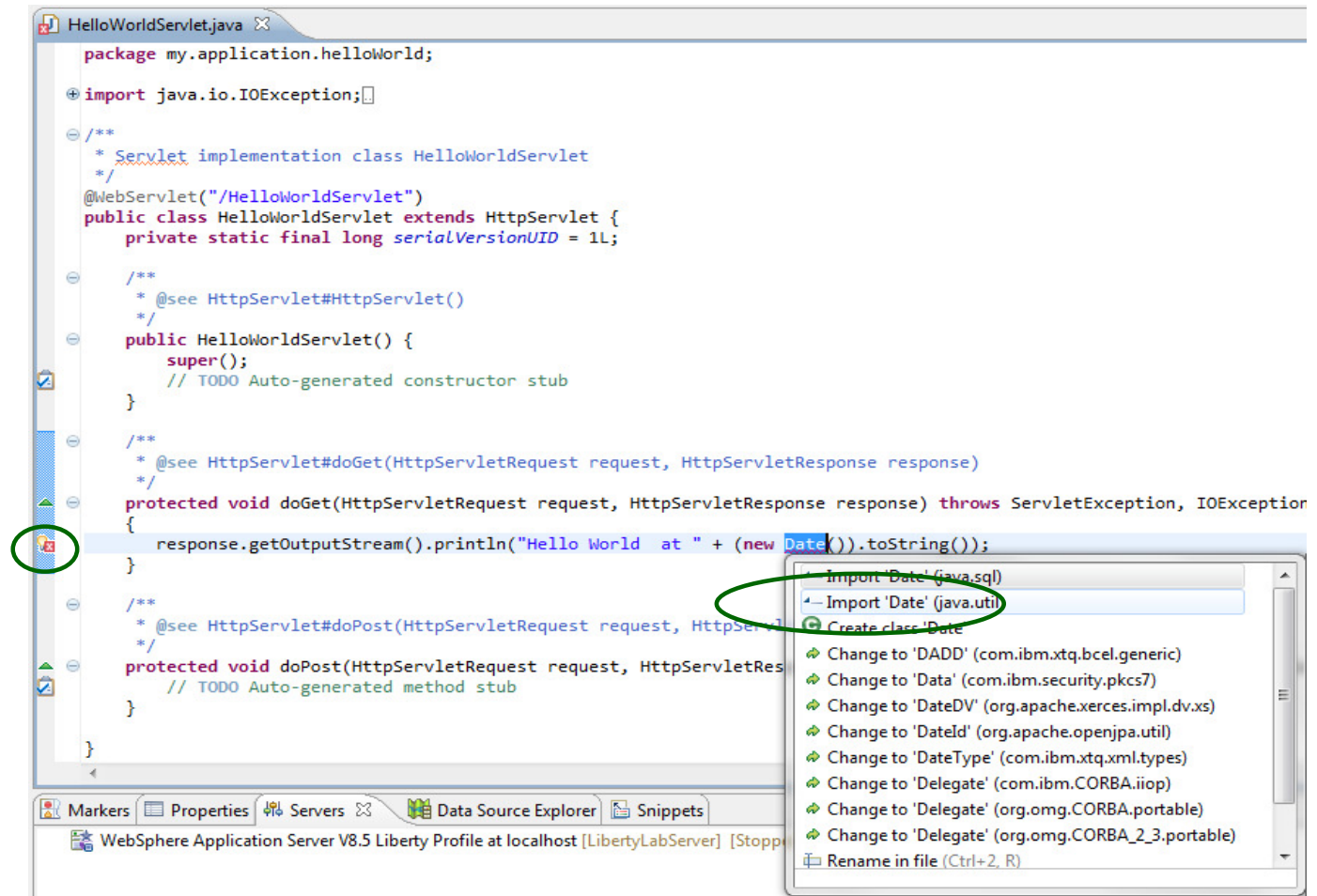
```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
    response.getOutputStream().println("Hello World  at " + (new Date()).toString());
}
```

LEARN

Correct the missing import

- You will see an error because "Date" is not resolved as a type

1 Click the highlighted error (X)
2 From the list, double-click on Import 'Date' (java.util)

Your application is ready

- The error has been corrected
- The application code is now ready to run
- Click File → Save


```java
package my.application.helloWorld;

import java.io.IOException;
import java.util.Date;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class HelloWorldServlet
 */
@WebServlet("/HelloWorldServlet")
public class HelloWorldServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HelloWorldServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.getOutputStream().println("Hello World  at " + (new Date()).toString());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }

}
```

LEARN

Module review

- In this module, you developed code for a Java EE servlet

- Next, continue to module 2 where you will deploy and test the servlet

LEARN

Thank you

LEARN