

Lab 01

Introduction to Java Servlet và JSP

Mục tiêu

- Cài đặt Netbeans tích hợp sẵn Java EE
- Cài đặt các web server và biết cách start, stop, deploy application trên web server
- Tạo ứng dụng đầu tiên HelloWorld, dùng Servlet và deploy lên Server thực hành với request và response.

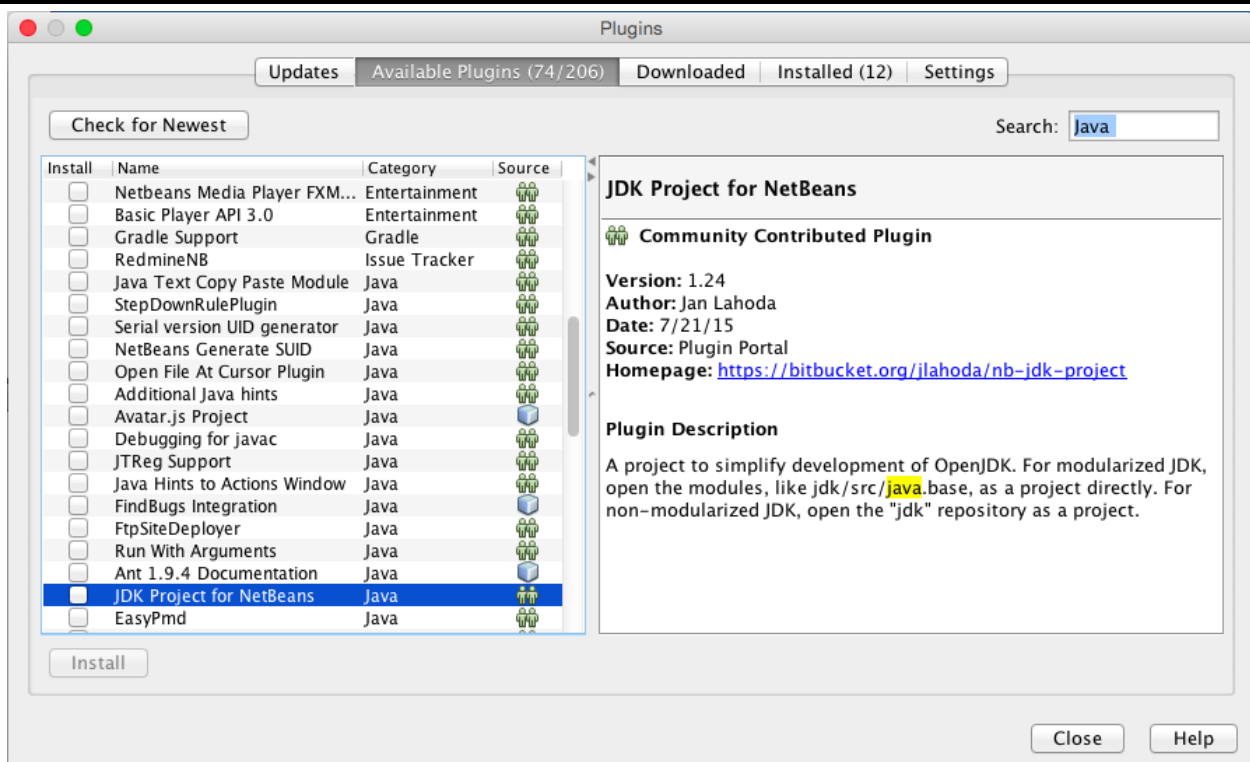
Phần I Bài tập step by step

Bài 1.1

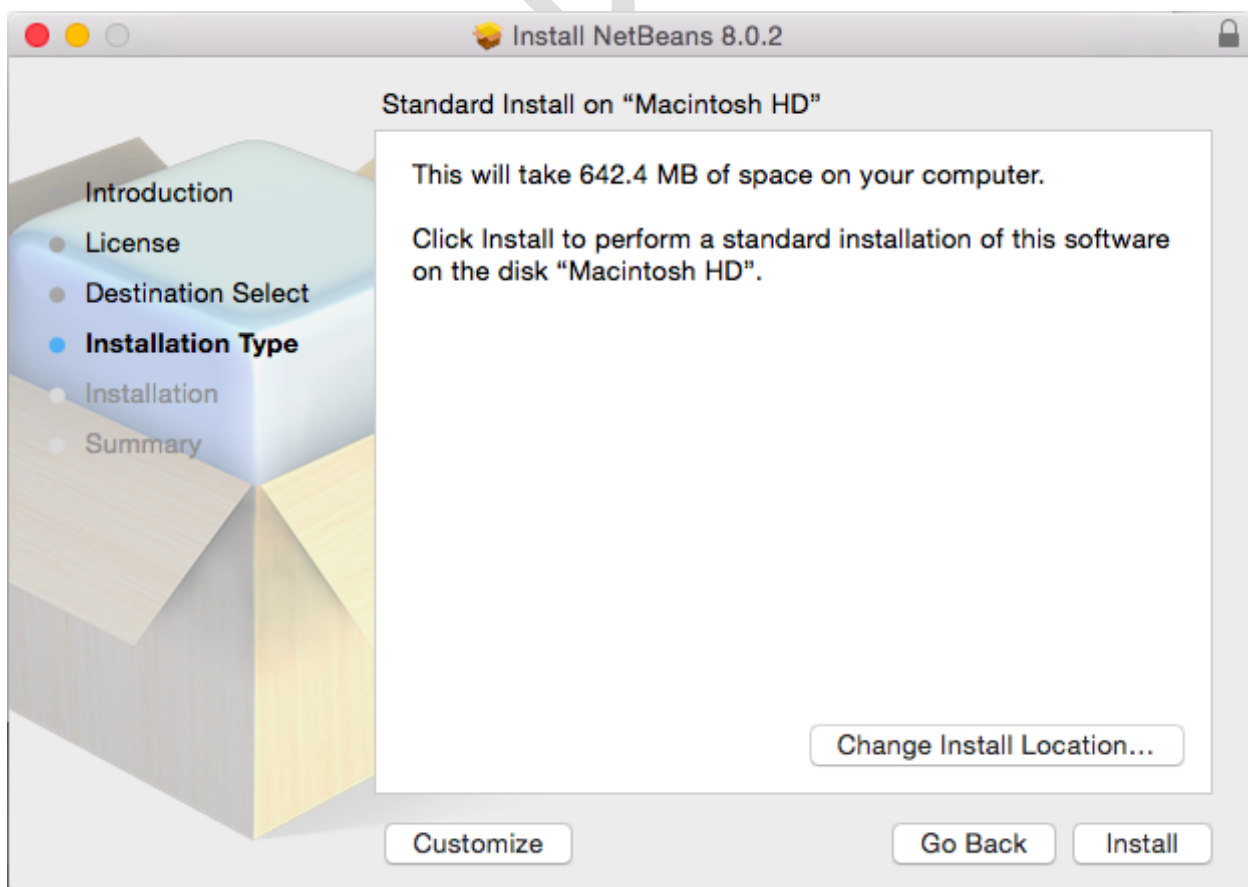
Bước 1: Open <https://netbeans.org/downloads/> to download Netbeans

NetBeans IDE Download Bundles					
Supported technologies *	Java SE	Java EE	C/C++	HTML5 & PHP	All
NetBeans Platform SDK	•	•			•
Java SE	•	•			•
Java FX	•	•			•
Java EE		•			•
Java ME					—
HTML5		•		•	•
Java Card™ 3 Connected					—
C/C++			•		•
Groovy					•
PHP				•	•
Bundled servers					
GlassFish Server Open Source Edition 4.1		•			•
Apache Tomcat 8.0.15		•			•
	Download	Download	Download	Download	Download
	Free, 105 MB	Free, 222 MB	Free, 72 MB	Free, 72 MB	Free, 243 MB

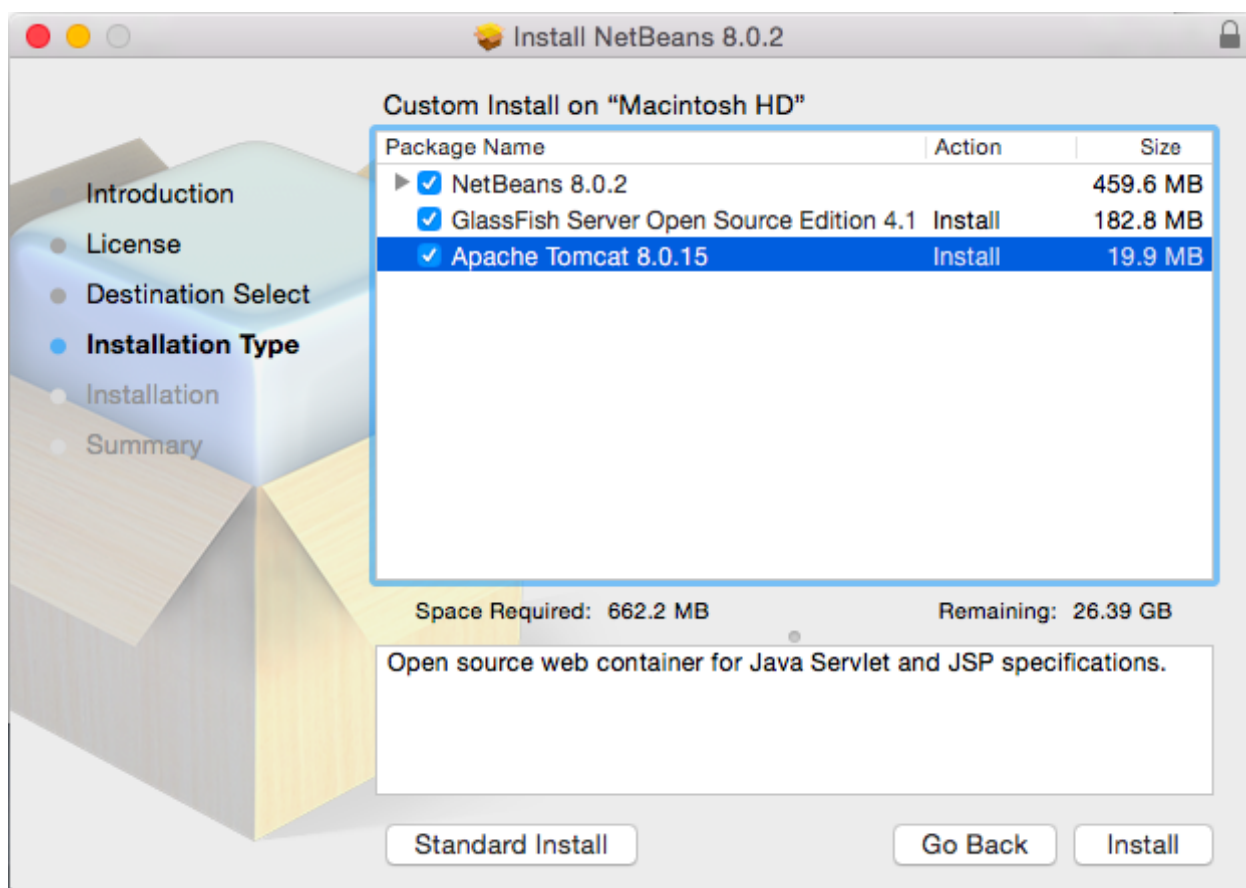
- Chọn Netbeans tích hợp Java EE -> Download
- Nếu cài đặt sẵn trong máy là Netbeans SDK (chưa có Java EE): Chọn Tools->Plugins->Available Plugins-> Tìm và chọn Java WEB and EE -> Install



- Trong quá trình cài đặt cần cài thêm các web servers tích hợp với Netbeans. Netbeans Java EE hỗ trợ tích hợp sẵn GlashFish và Apache Tomcat.



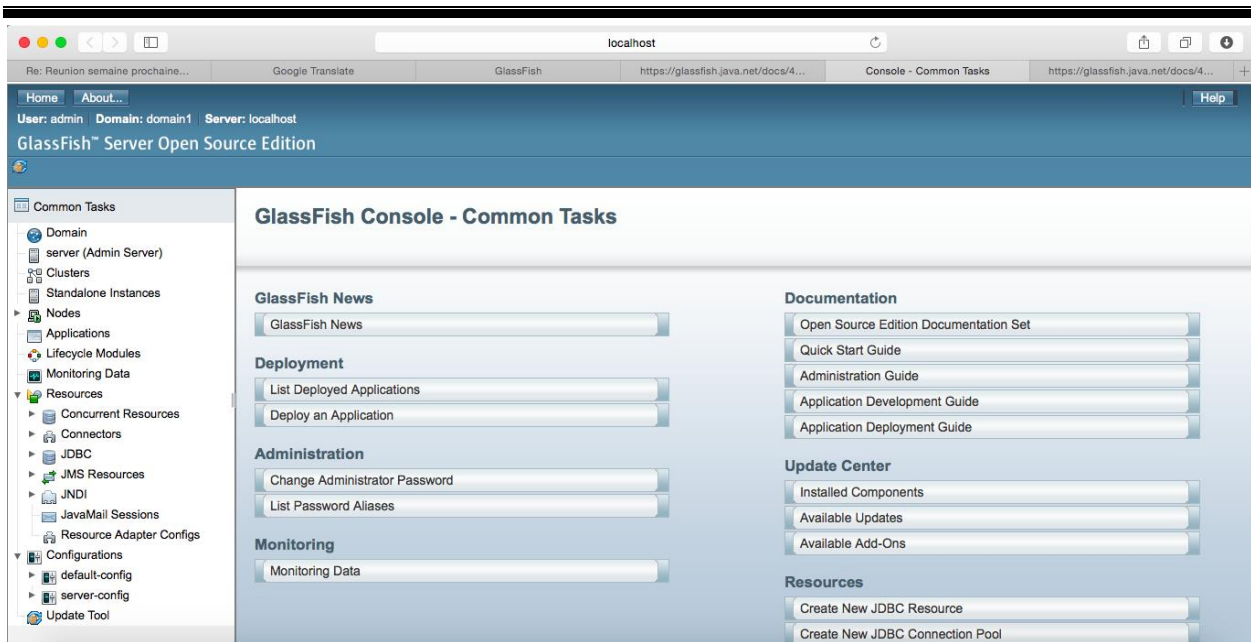
- Chọn Customize -> Chọn GlashFish và Apache Tomcat (hoặc 1 trong 2) -> Install



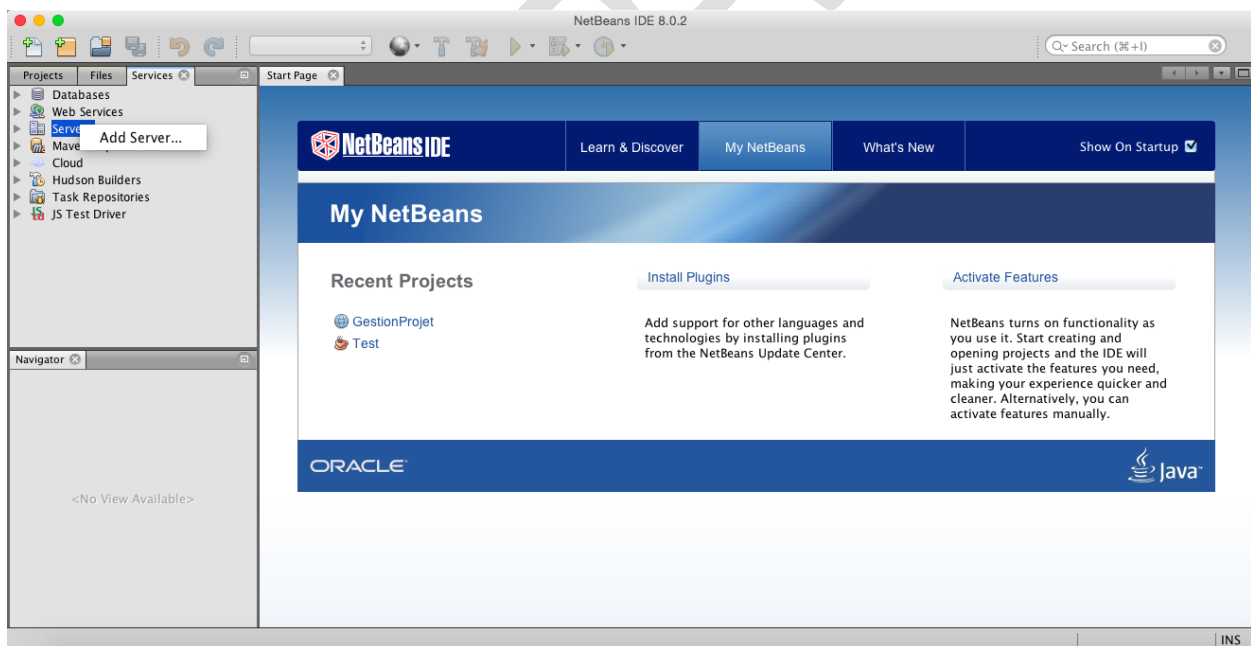
Bước 2: Nếu không có sẵn GlashFish/Tomcat trong Netbeans, cài đặt GlashFish hoặc/và Apache Tomcat như sau:

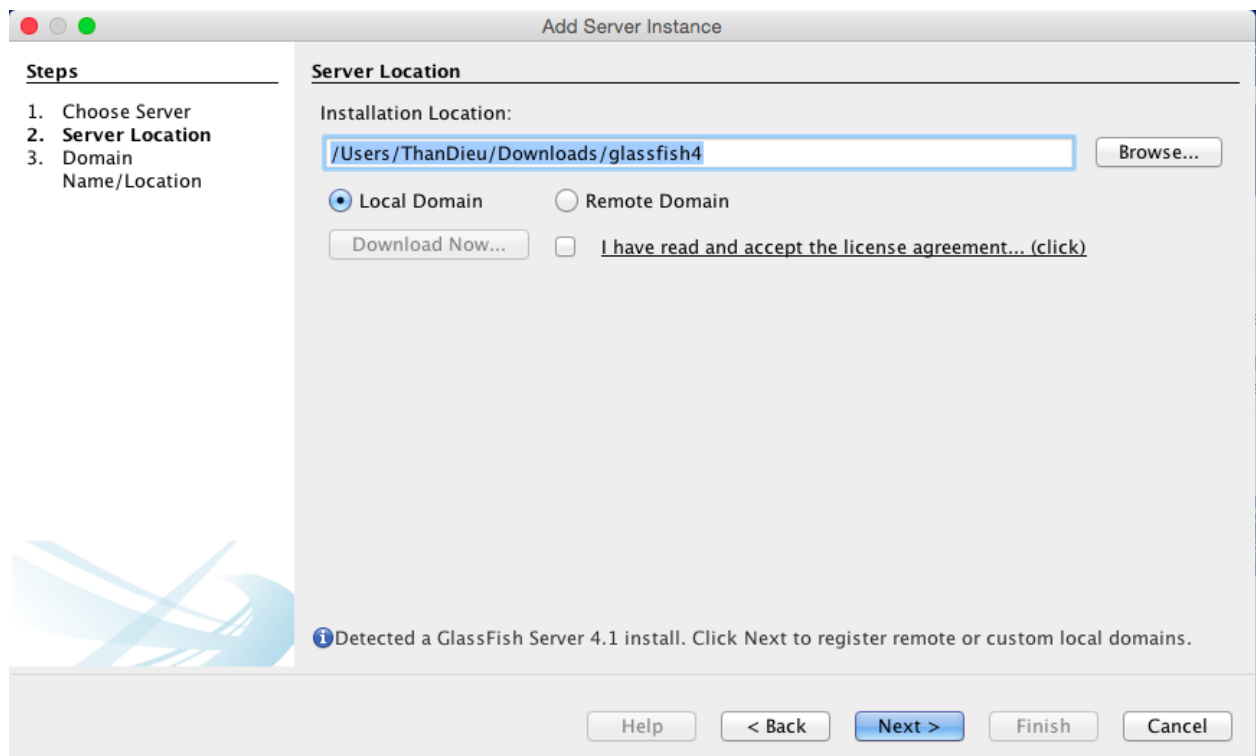
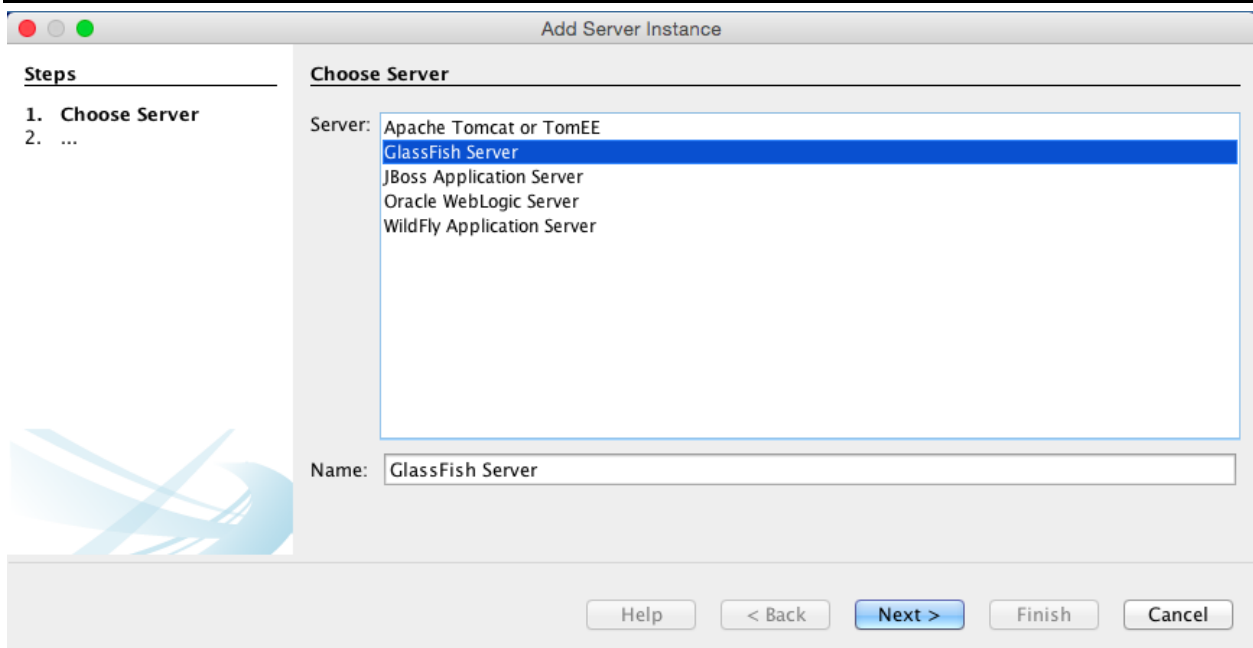
2.1. Cài đặt GlashFish 4.x và cấu hình trong Netbeans

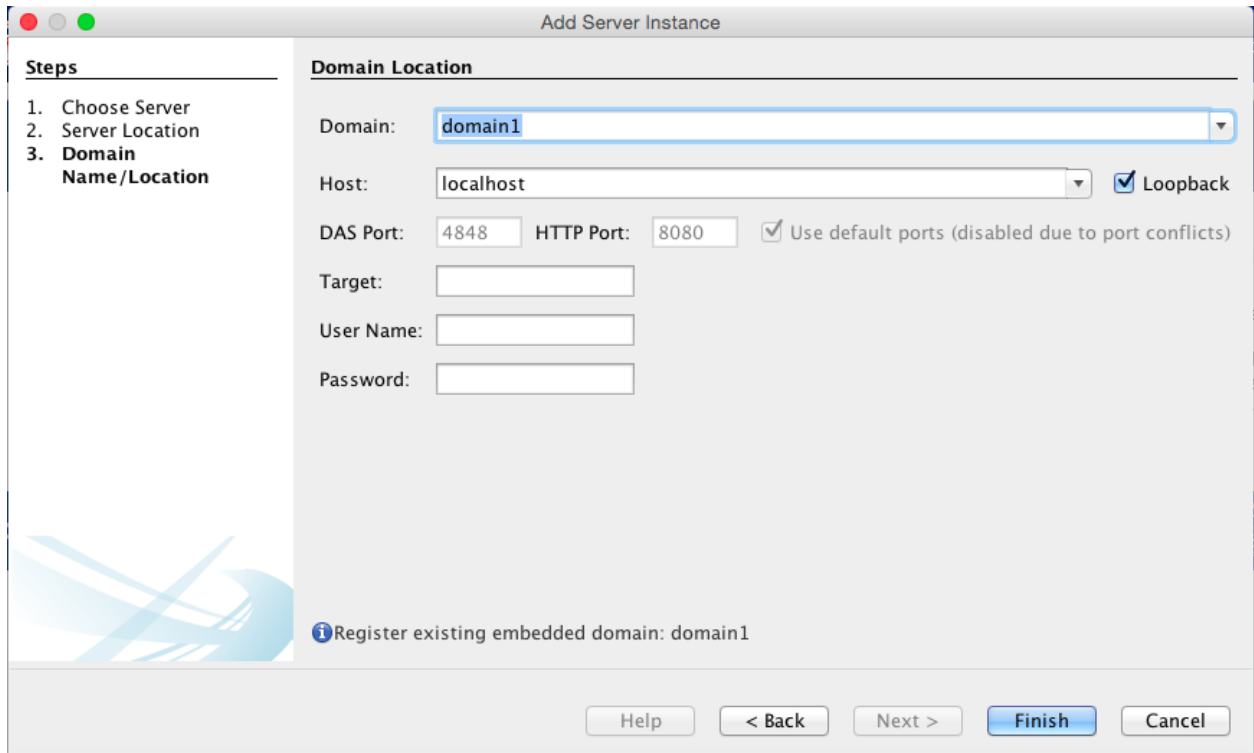
- Mở <https://glassfish.java.net/download.html> -> Download [glashfish-4.1-web.zip](#) trong trường hợp chưa có sẵn Java EE 7 thì download [glashfish-4.1.zip](#)
- Unzip file, open Terminal (Command-Line in Windows), di chuyển tới thư mục đã unzip, ví dụ, unzip vào C:/glashFish4 → `cd C:/glashFish4/bin`
- Sau đó gõ tiếp lệnh: `asadmin start-domain` để start server
- Mở trình duyệt (IE, Chrome,...) gõ: <http://localhost:4848> (4848 là listening port của GlashFish)



- Cấu hình GlashFish trong Netbeans:
 - Mở Netbeans
 - Chọn Services Tab -> Chuột phải Servers -> Add Server

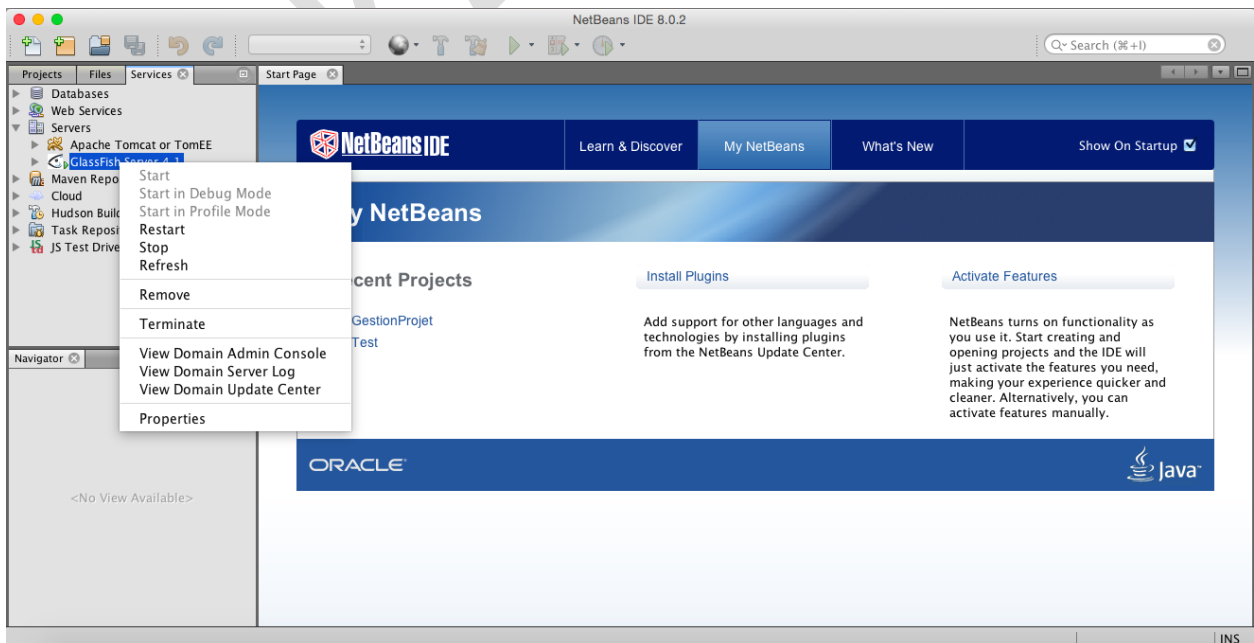




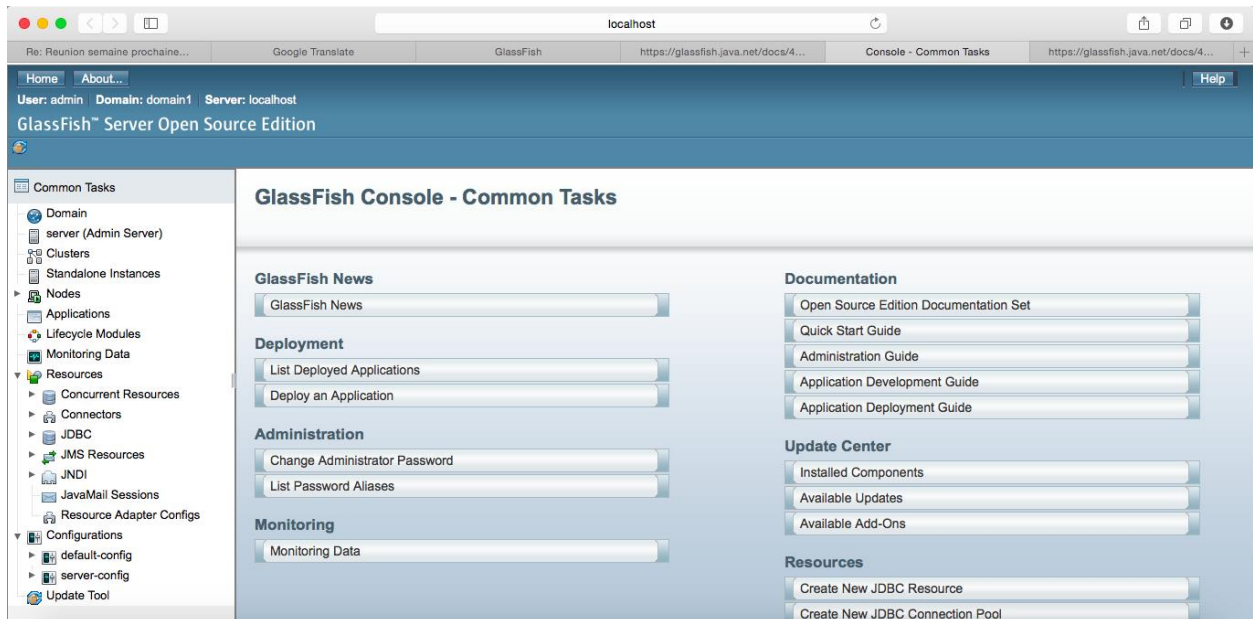


- Start GlashFish trên Netbeans:

Services Tab -> Servers -> Expand Servers -> Chọn GlashFish Server -> Chuột phải chọn Star Server



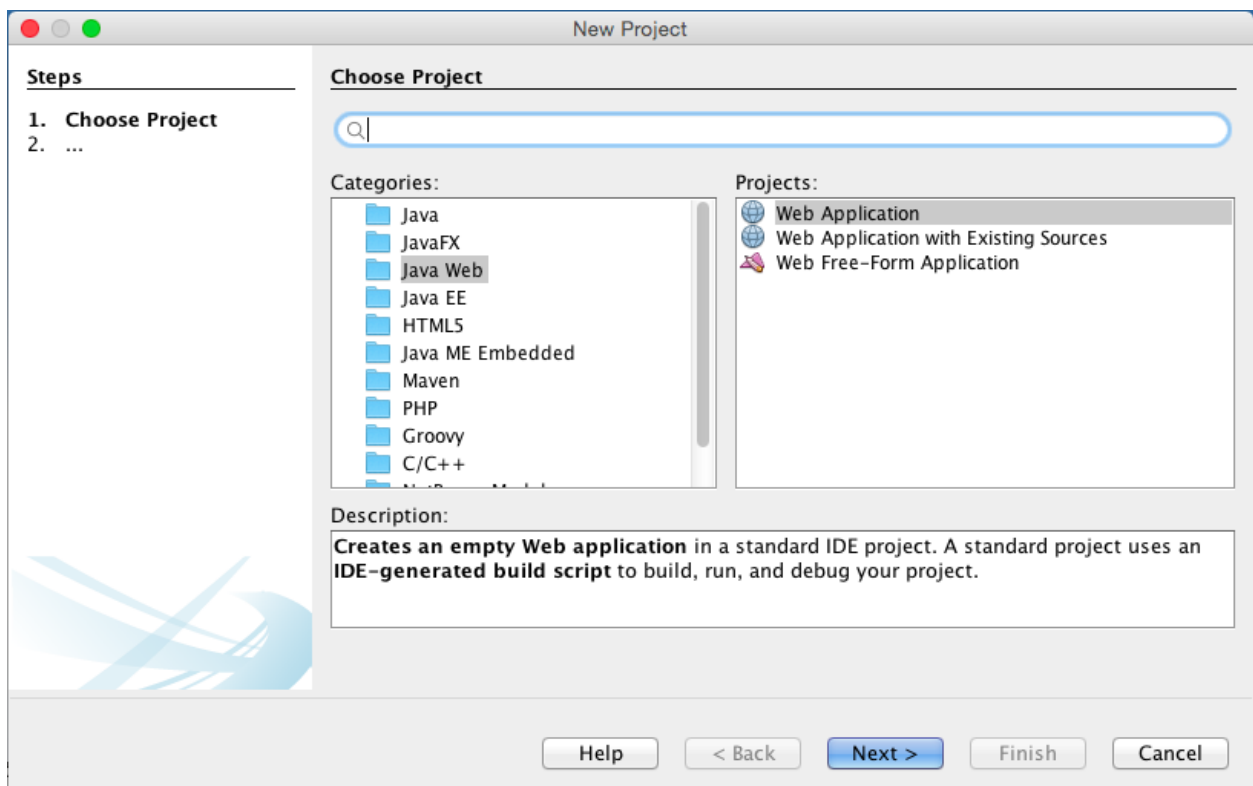
- Chuột phải GlashFish server trong Services tab -> View Domain Admin Console
hoặc mở trình duyệt và gõ: <http://localhost:4848> sẽ được kết quả như sau:



Bây giờ hãy tạo một ứng dụng web và chọn Server là GlassFish.

STEP 2.1.1: Tạo ứng dụng web trên Netbeans

File -> New Project -> Java Web - > Web Application



New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

Đặt tên project là: HelloWorld.

Chọn server GlassFish và Java EE 7

New Web Application

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**
4. Frameworks

Server and Settings

Add to Enterprise Application:

Server:

Java EE Version:

Context Path:

Finish

STEP 2.1.2: Tạo servlet và deploy lên server

Tạo package servlet cho project.

New Servlet -> HelloWorld.

Chọn Add information to deployment descriptor.

The screenshot shows a 'New Servlet' dialog box with the following fields and values:

- Class Name:** HelloWorld
- Project:** HelloWorldGF
- Location:** Source Packages
- Package:** servlet
- Created File:** etBeansProjects/HelloWorldGF/src/java/servlet/HelloWorld.java

The 'Next >' button is highlighted in blue. The 'Steps' panel on the left shows the current step is '2. Name and Location'.

Steps

1. Choose File Type
2. Name and Location
3. **Configure Servlet Deployment**

Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☒ Add information to deployment descriptor (web.xml)

Class Name:

Servlet Name:

URL Pattern(s):

Initialization Parameters:

Name	Value
------	-------

New Edit... Delete

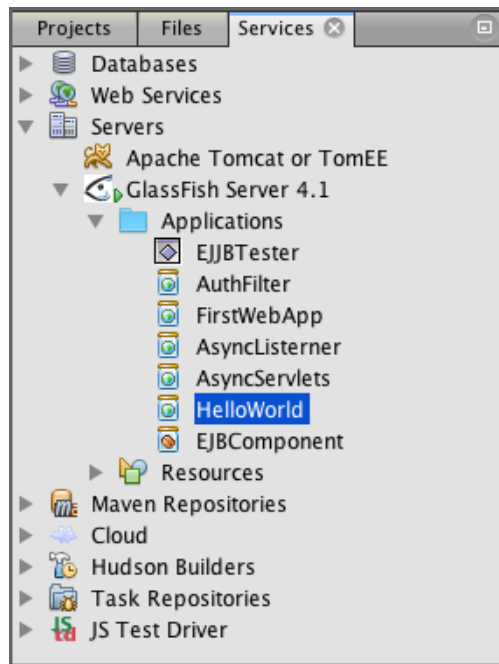
Help < Back Next > Finish Cancel

Mở file web.xml trong thư mục Configuration Files của project, thông tin cấu hình của Servlet đã được thêm vào web.xml

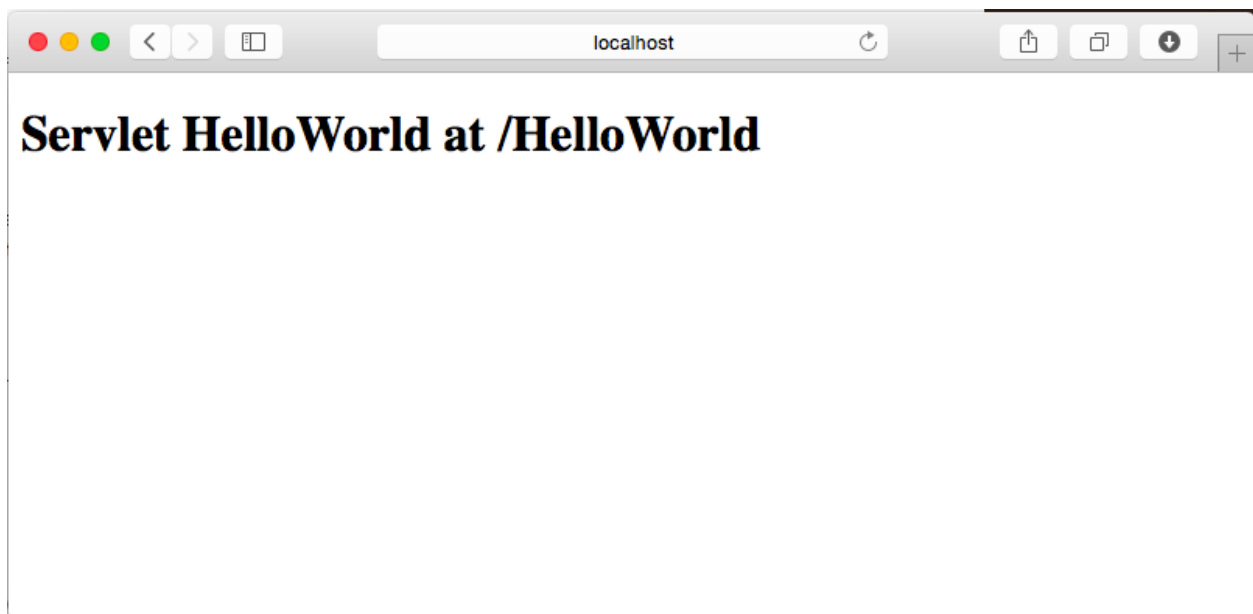
```
<servlet>
  <servlet-name>HelloWorld</servlet-name>
  <servlet-class>servlet.HelloWorld</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>HelloWorld</servlet-name>
  <url-pattern>/HelloWorld</url-pattern>
</servlet-mapping>
```

Trong config này chúng ta thấy tên của Servlet và java class tương ứng. Sau đó là url-pattern tương ứng với tên của servlet.

Trong file HelloWorld.java chúng ta sẽ thấy 1 số method được sinh tự động bởi Netbeans. Build and run project, ứng dụng sẽ được deploy lên GlassFish server và chúng ta sẽ thấy nó trong thư mục Applications của server:



Gõ đường dẫn tới url-pattern của servlet: <http://localhost:8080/HelloWorld/HelloWorld>

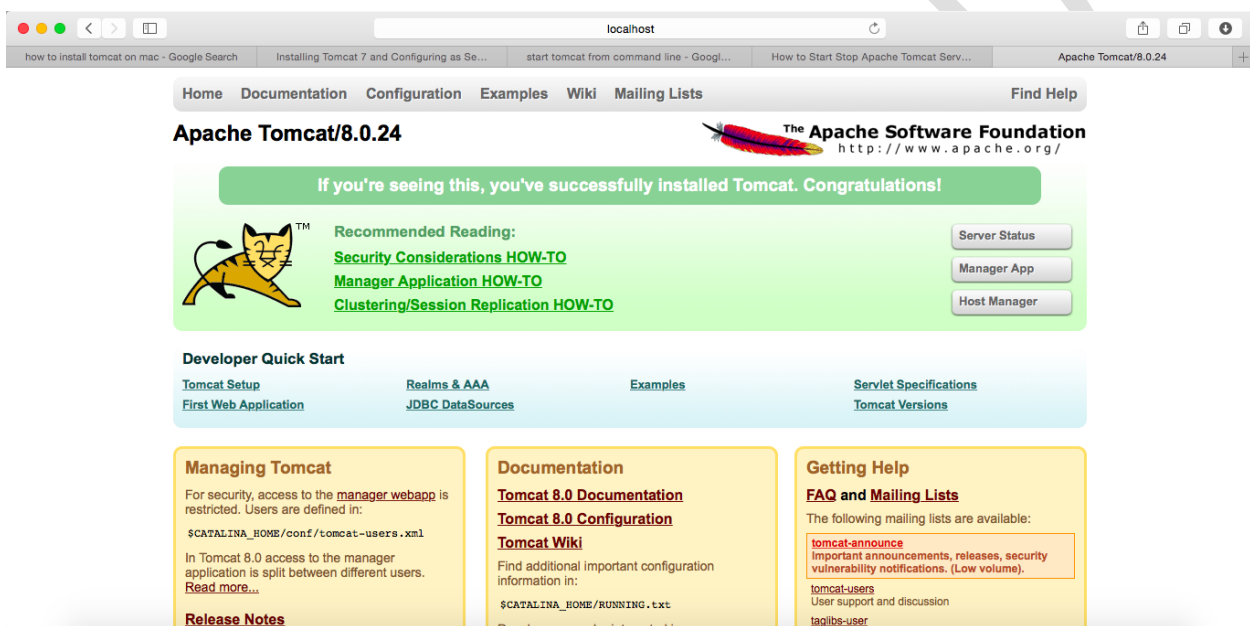


Muốn undeploy ứng dụng trên GlassFish, chúng ta vào Services tab, mở GlassFish server, chọn tên ứng dụng và undeploy.

2.2. Cài đặt Server Apache Tomcat

- Mở <http://tomcat.apache.org/download-80.cgi> để download Tomcat 8.0, chọn Deployer version.
- Un zip file ra thư mục C:/tomcat8

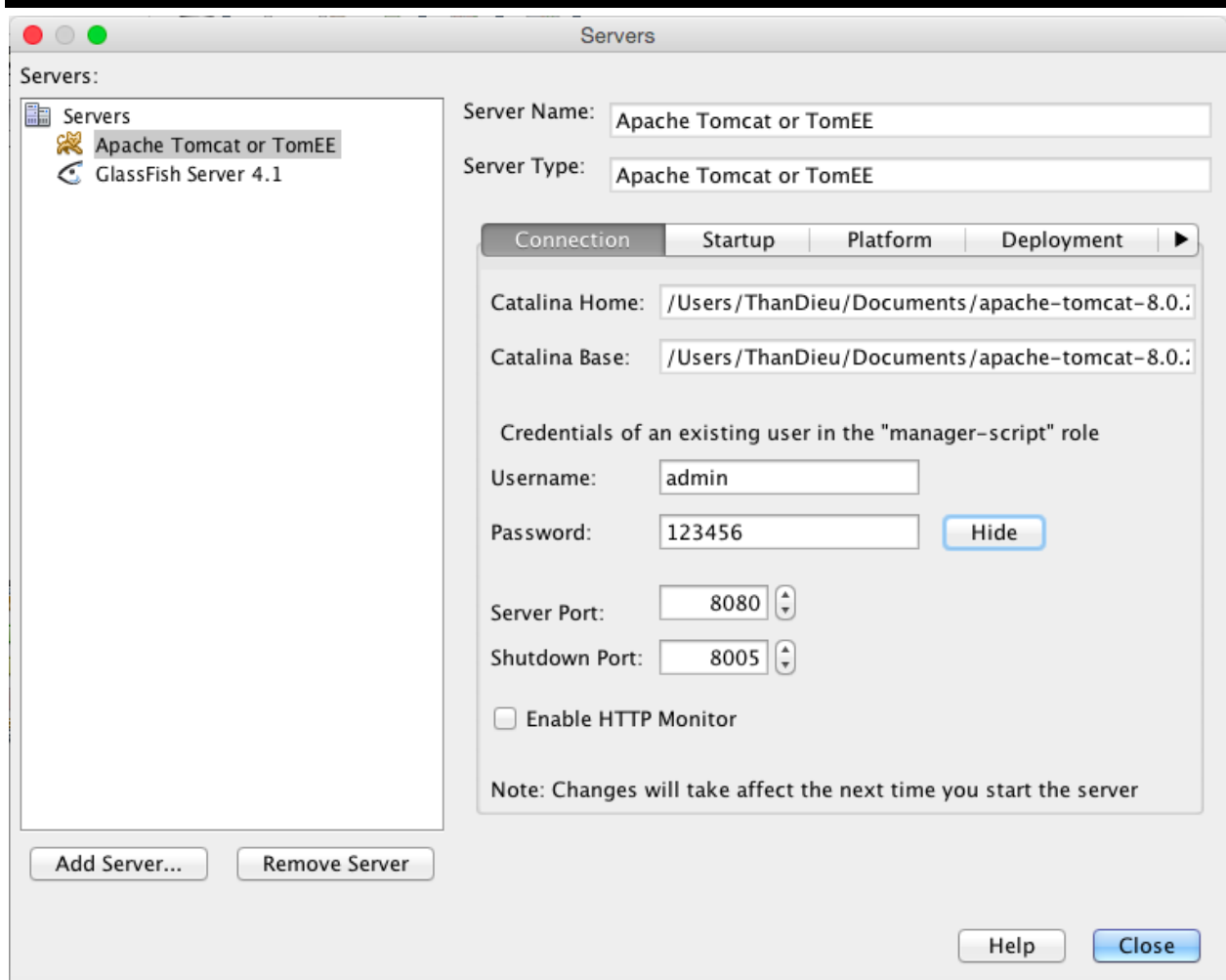
- Start/Stop Tomcat bằng dòng lệnh: Mở Command Line -> cd C:/tomcat8 [Trên Unix/Linux/Mac, cd thư_mục_chứa_Tomcat] -> cd bin
- Trên windows:
 - Start Tomcat: gõ startup.bat
 - Stop Tomcat: gõ shutdown.bat
- Trên Unix/Linux/Mac:
 - Start Tomcat: gõ ./startup.sh
 - Stop Tomcat: gõ ./shutdown.sh
- Test: gõ <http://localhost:8080> trên trình duyệt:



- Cấu hình Tomcat và start, stop từ Netbeans tương tự như cấu hình GlashFish.

Vì GlashFish và Tomcat mặc định cổng HTTP port là 8080. Do đó khi một ứng dụng đang chạy trên GlassFish thì run/start Tomcat sẽ bị xung đột.

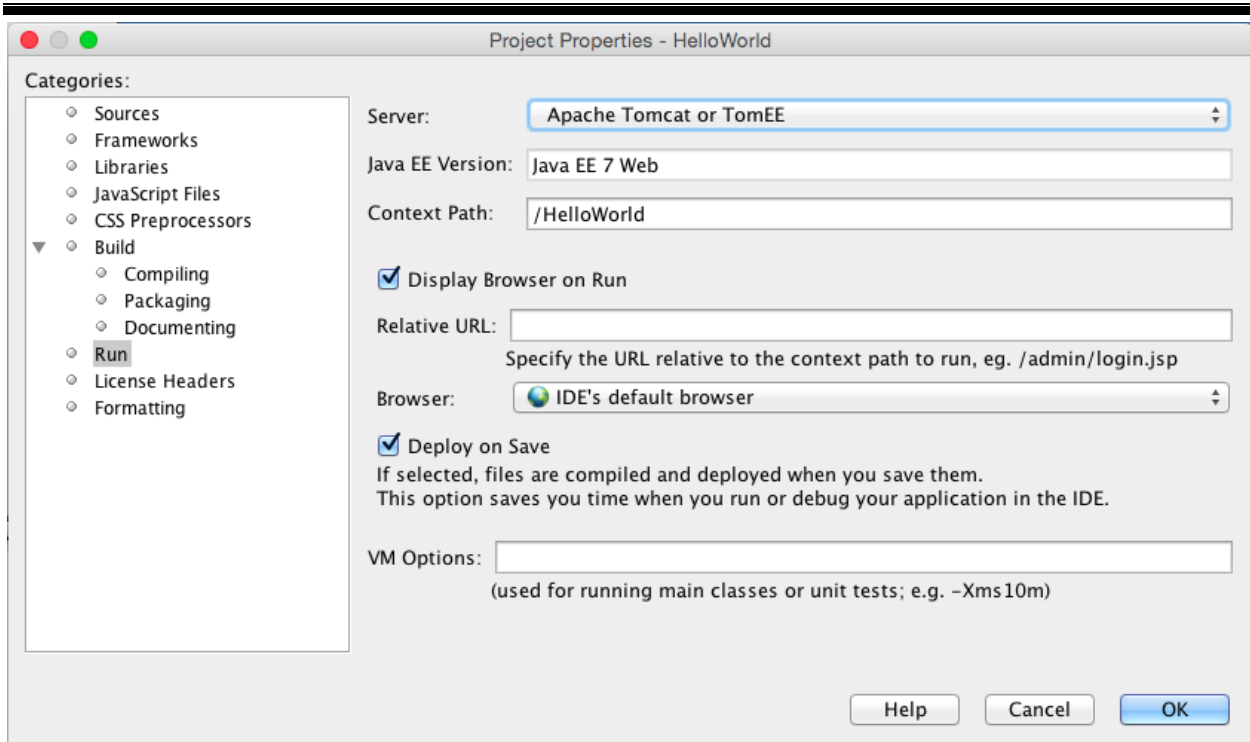
Config lại cổng của Tomcat như sau: Services tab -> Tomcat Server -> Properties



Ví dụ thay Server Port: 8085.

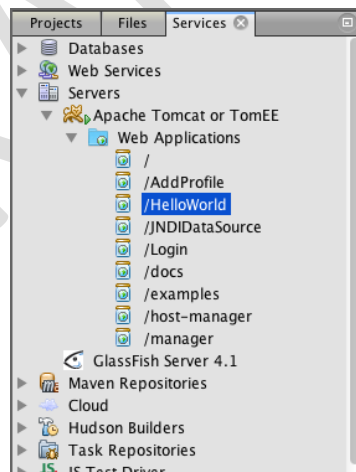
Một ứng dụng web có thể thiết lập lại Server bằng cách chọn project -> Properties -> Run -> Chúng ta đặt lại server cho ứng dụng trong list các server.

Chuột phải project HelloWorld -> Properties -> Run -> Chọn Tomcat Server thay cho GlassFish



Clean and Build -> Deploy

Mở Services tab -> Servers -> Tomcat -> Mở thư mục Web Applications, ứng dụng HelloWorld đã được deploy lên Tomcat Server.



Gõ đường dẫn: <http://localhost:8080/HelloWorld/HelloWorld>

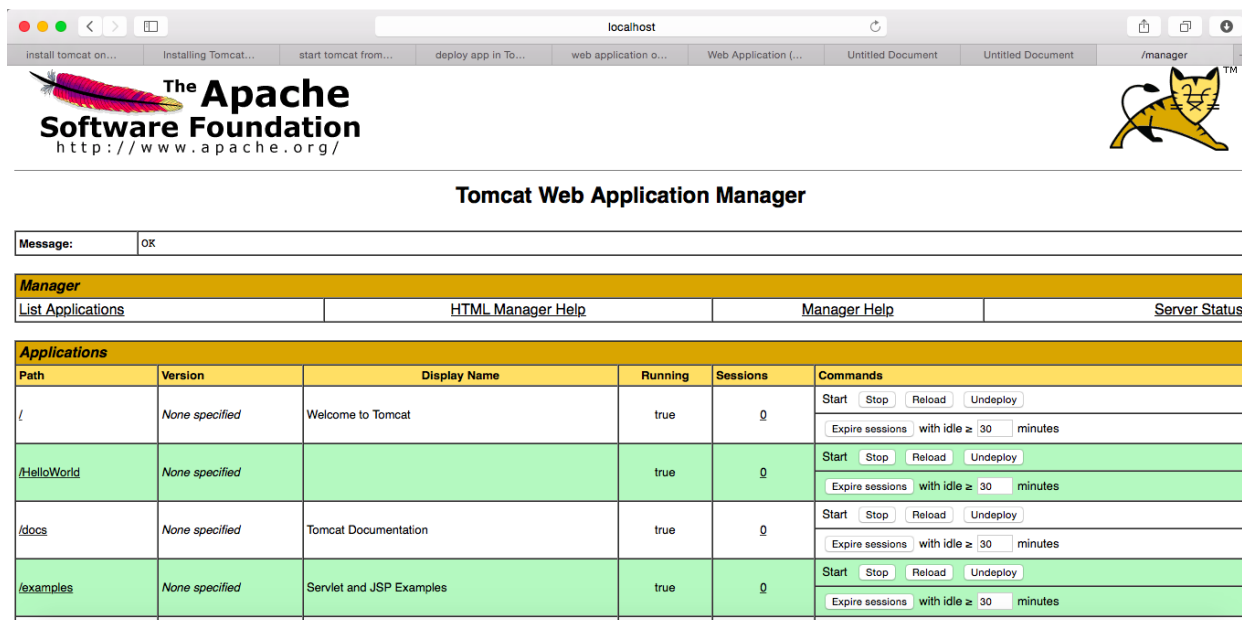
Servlet HelloWorld sẽ được gọi và trả về kết quả như phần trước.

Để undeploy ứng dụng, chọn Services tab -> Servers -> Tomcat Server -> Web Applications -> Chọn ứng dụng muốn undeploy -> Undeploy.

Ngoài ra, chúng ta cũng có thể undeploy hoặc deploy ứng dụng từ Web Console của Tomcat.

Gõ vào trình duyệt: <http://localhost:8080>

Chọn Manage App



Để deploy lại ứng dụng, ở Deploy tab, chúng ta có thể chọn đường dẫn tới file .war của ứng dụng. File .war của ứng dụng nằm trong thư mục build/dist của project.

Bài 1.2: Tạo form nhập và hiển thị Hello + name

Quay trở lại với HelloWorld project đã tạo ra ở bài 1.

Mục tiêu:

- Biết cách thực hành với form trên jsp.
- Biết lấy về tham số từ form và hiển thị lại trên response.

Bài này chúng ta thực hành với GlassFish server. Chọn lại server của ứng dụng HelloWorld là GlassFish.

STEP 1: Sửa lại file index.html để tạo form nhập liệu

```

  <html>
<head>
  <title>TODO supply a title</title>

```

```

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  </head>
  <body>
    <form method="get" action="hello.jsp">
      <input type="text" id="username" name="username" />
      <input type="submit" value="Go" />
    </form>
  </body>
</html>

```

Chúng ta set action của form là một trang jsp khác có tên là hello.jsp.

STEP 2: Tạo trang hello.jsp xử lý dữ liệu nhập được từ form

File -> New JSP -> hello.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>JSP Page</title>

  </head>

  <body>

    <h1>Hello <%= request.getParameter("username") %></h1>

  </body>

</html>

```

Chúng ta dùng: `request.getParameter("username")` để lấy về thông tin từ input text của form. Để hiển thị, sử dụng jsp scriptlet `<%= %>` để lấy value từ dòng code java trên trang jsp.

Thay vì khai báo `action="hello.jsp"` chúng ta có thể khai báo action của form là một servlet.

Sửa lại index.html -> `action="HelloWorld"`

Chú ý: HelloWorld là name của servlet (trong web.xml) đã config ở trên.

Mở HelloWorld.java -> Xoá toàn bộ code tự sinh của Netbeans.

Override method doGet của HttpServlet.

```
package servlet;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HelloWorld extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code.
            */

            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet HelloWorldServlet</title>");
            out.println("</head>");
            out.println("<body>");

            String username = (String) request.getParameter("username");
            out.println("<h1>Hello " + username + "!</h1>");

            out.println("</body>");
            out.println("</html>");

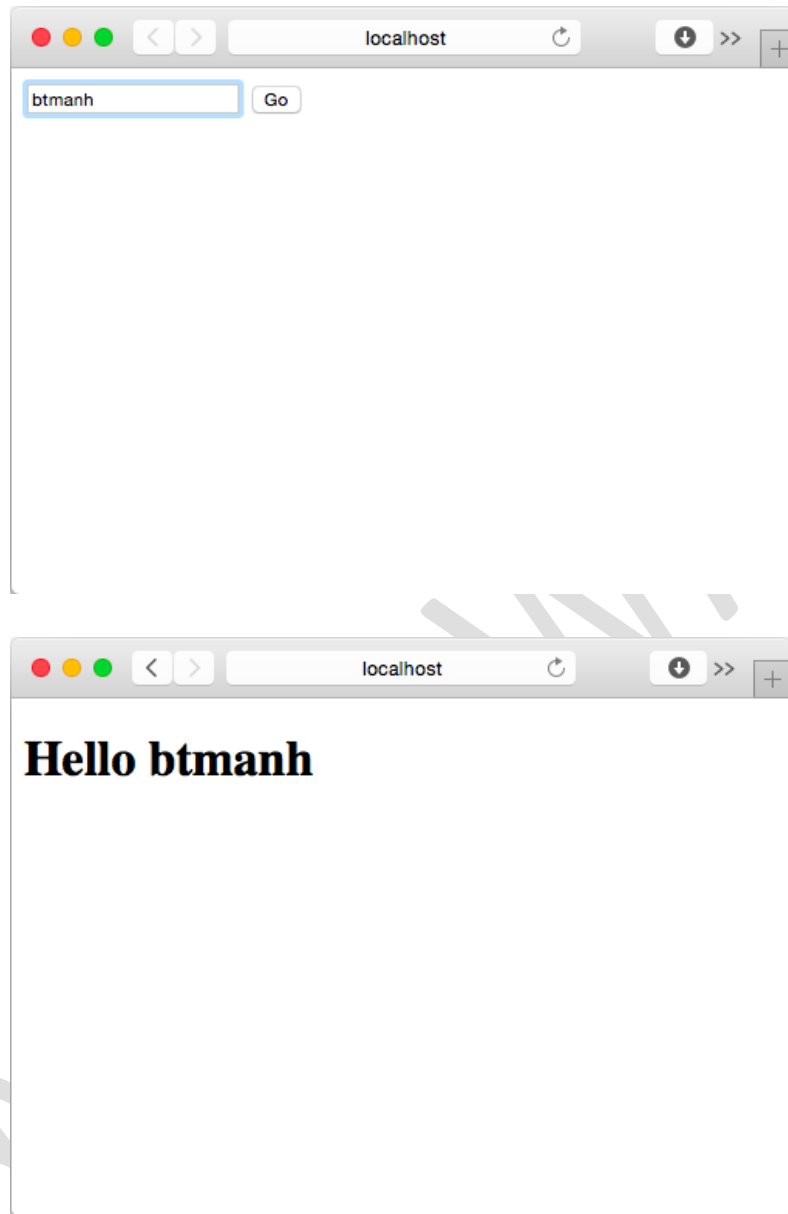
        }
    }
}
```

Tương tự, chúng ta cũng lấy giá trị nhập liệu từ form bằng:

`request.getParameter("username")`.

Để ghi lên response và trả về response, chúng ta dùng `request.getWriter()` trả về object của class `PrintWriter`.

Clean and Build ứng dụng web với hai trường hợp, submit form lên trang hello.jsp và submit form lên servlet. Chúng ta đều có kết quả như sau:



Phần II: Bài tập tự làm

Bài 1.3

Tiếp tục bài trước, hãy tạo 1 form chứa Username, User-email. Sau khi người dùng nhập và ấn OK, hãy hiển thị tên username và tên user-email.

Em hãy giải thích cơ chế hoạt động từ khi người dùng điền thông tin vào form và submit form đến khi trả về kết quả.

HẾT