CPRG352 - Web Application Programming

Intro Lab #1

Topic: Using Tomcat, a Java Web Application Server

Introduction

In this lab you will look at ***Tomcat***, a Java application server program that you will use to run dynamic web applications written using Java code. This is the main focus of this course, as Java Web (and JEE, Java Enterprise Edition, technology) provide a lot of support for writing web-based applications.

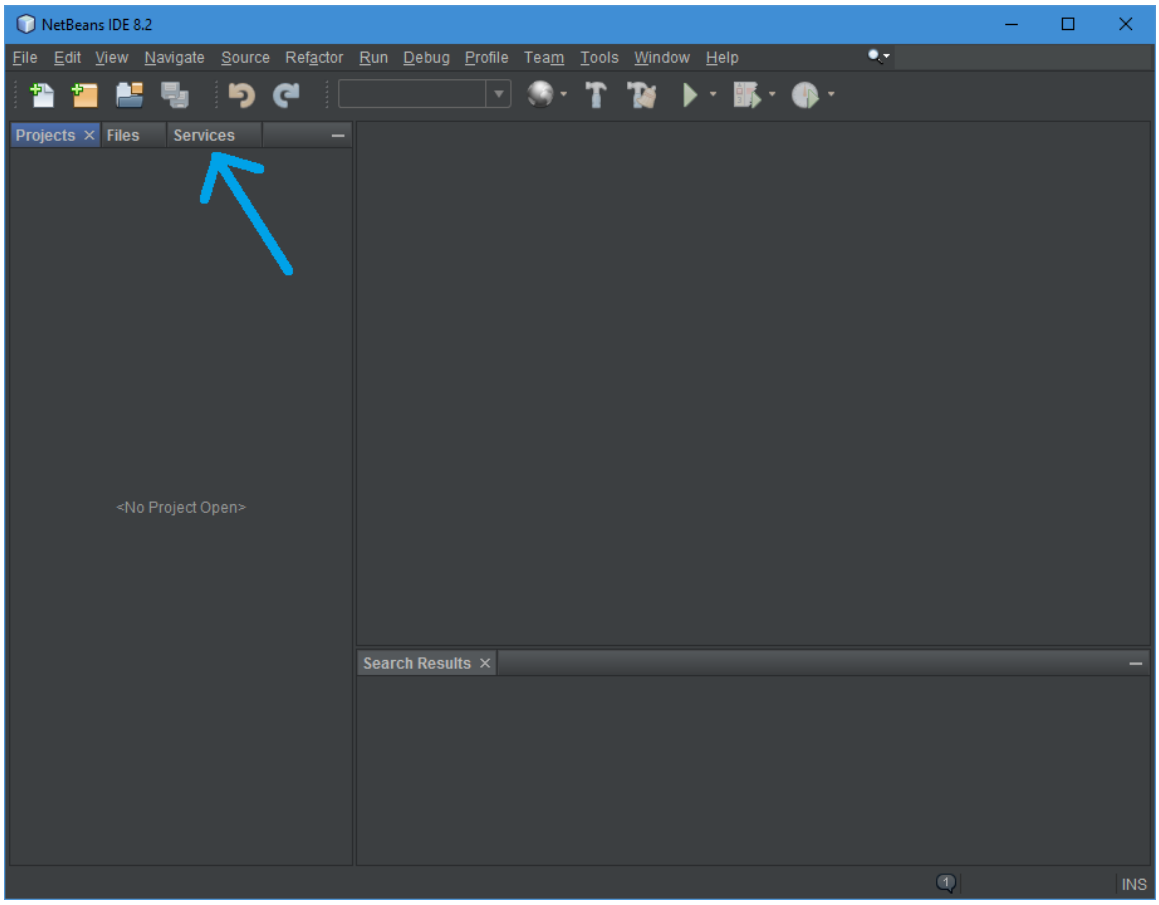
Tomcat can be installed and run as a completely stand-alone program, but in our VM it is tightly integrated with NetBeans, so in this lab you will examine the combination of the two programs and how they work together to allow you to manage Java web applications running on your computer.

Note: Tomcat contains its own web server component, which will be used in this course. It is certainly possible to access Tomcat services through another web server program, e.g. Apache, but that complicates things so we won't do that. The Tomcat web server will be accessed via port ***8080***.

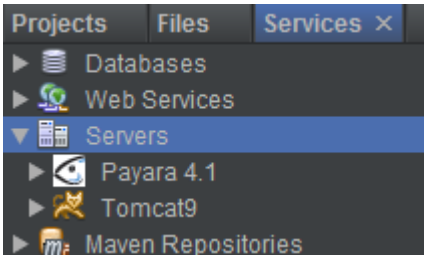
PART 1: TOMCAT

Managing Tomcat from NetBeans

- Run NetBeans
- Click on the ***Services*** tab at the top-left of the NetBeans window



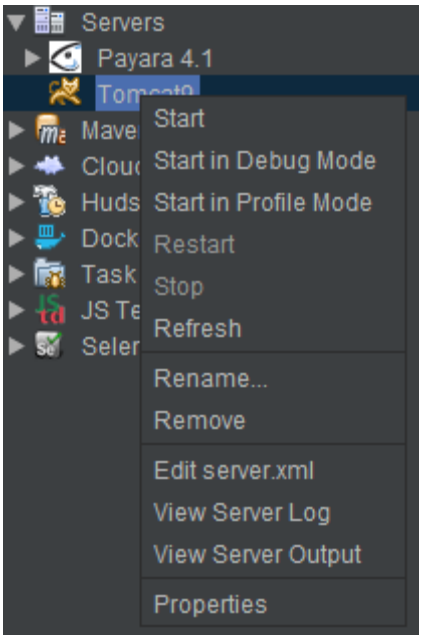
- Expand the ***Servers*** option, and you will see any JEE application servers configured for use in Netbeans :



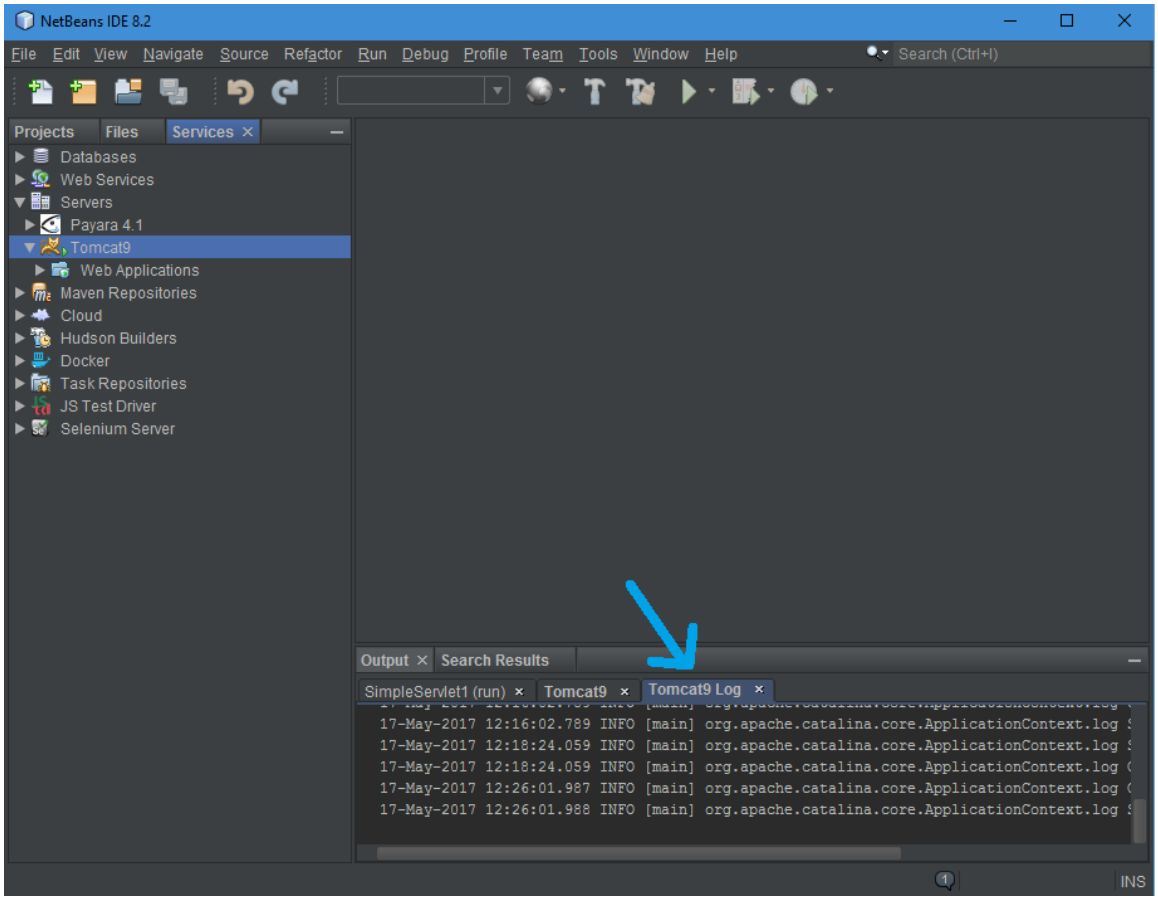
You will probably only see Tomcat listed under ***Servers***, but in theory you could install others also, such as the *Payara* application server shown in the screenshot above.

If you click on the "+" symbol beside the Tomcat9 server it will disappear, indicating that the server is not currently running, and so there is nothing to show for the server at present.

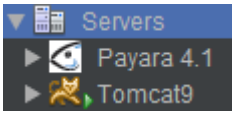
- Right-click on ***Tomcat9***, and select the ***Start*** option to start the server (this will take a little time)



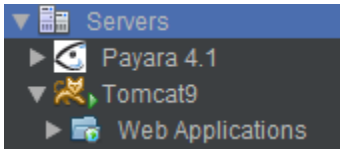
At the bottom-right of the NetBeans you will see an ***Output*** panel. In this panel you can watch the startup messages produced as Tomcat runs, e.g.



When Tomcat has started up you will see the following displayed in the ***Services*** panel at the top-left of NetBeans (note the green "running" arrow beside Tomcat):



- Click on the "▶" beside Tomcat9 to expand it



The option under ***Tomcat*** is ***Web Applications***. This option will show a list of the Java web applications installed on this server (initially there are some standard applications that come with Tomcat, we will install our own applications later).

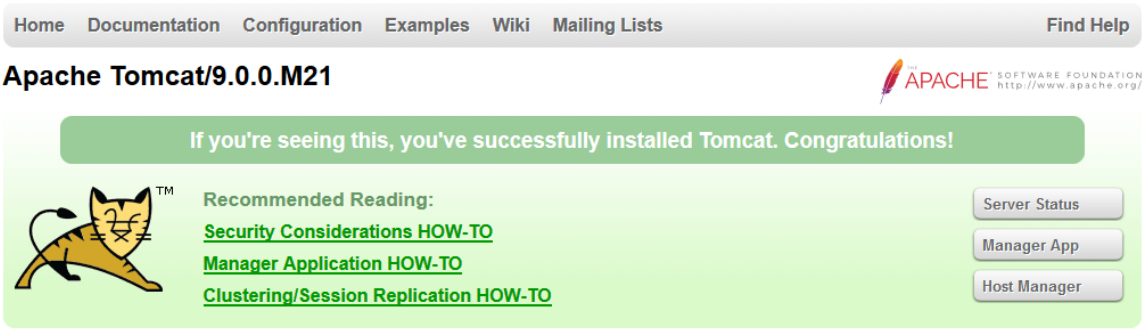
- Download a copy of the file called ***PrintEnv.war*** from the Brightspace web site for this course. You will find in under the **Sample Code** content section

You must ***deploy*** ("install") this program into the Tomcat application server before it can be run by a user.

To deploy the application you need to go to the Tomcat management application (a web service provided by Tomcat on port 8080).

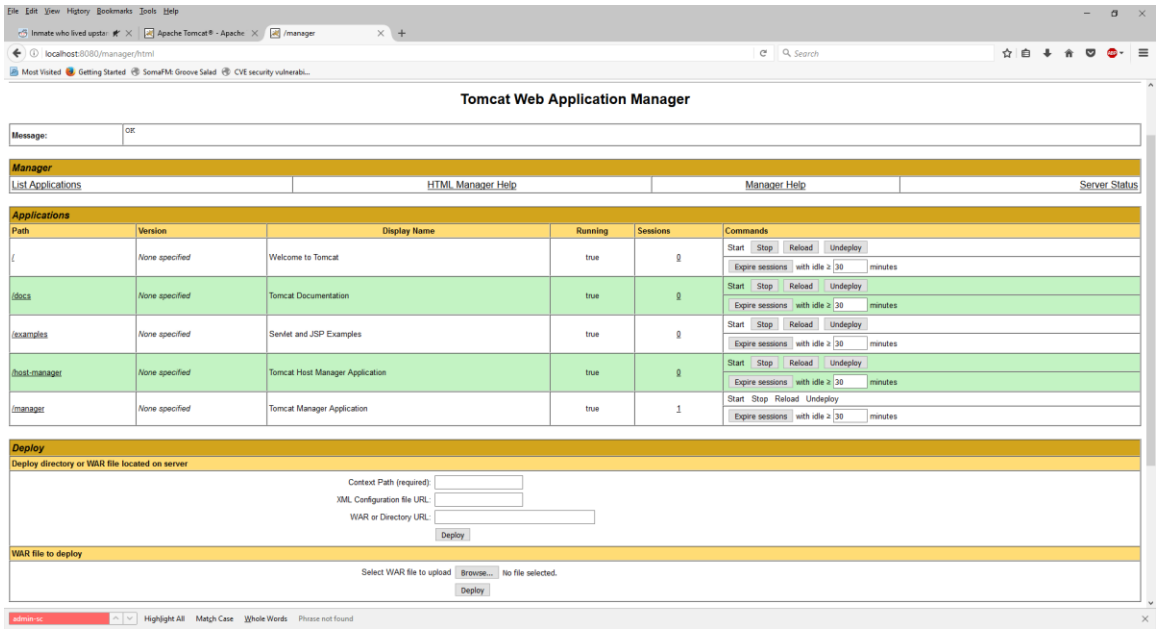
To view this application:

- In your browser navigate to the URL: **http://localhost:8080**. You should see a page starting as shown below:



- Click on the "Manager App" button on the right of the page. You might be prompted for a user name and password at this stage. Use "admin" and "password" for these values

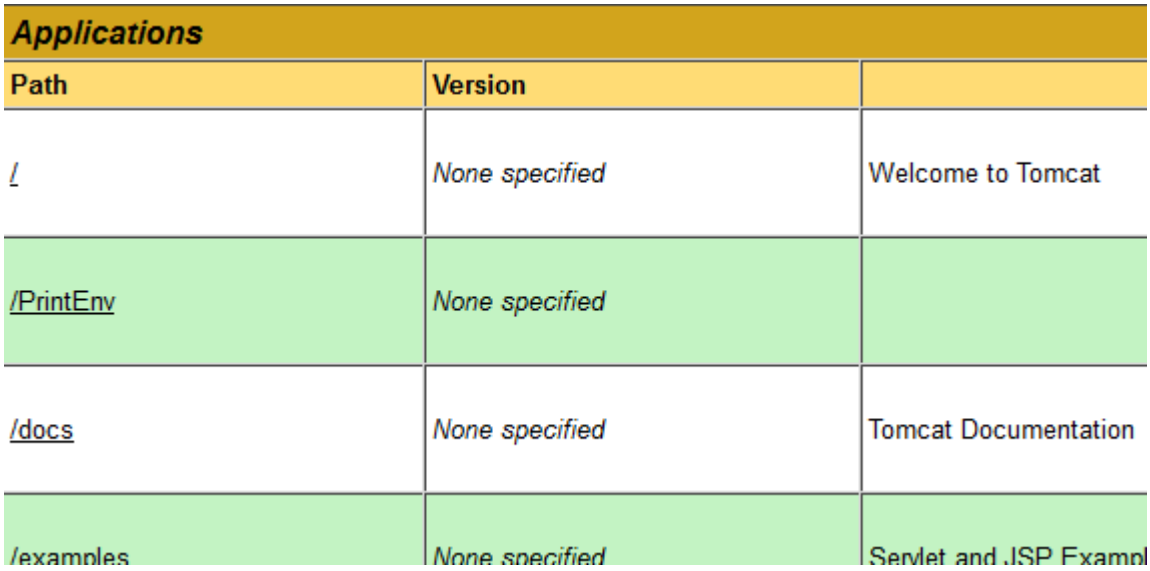
- When you have logged in you should see the page below:



This shows any web applications that are installed on the Tomcat server, and allows you to uninstall existing application, or install new ones, as required.

- In the page section titled "**Deploy**" look for the "**Browse…**" button. This will let you select a ***.war*** (***w***eb ***ar***chive) file that contains the Java web application to install on the server. Click on this button and select the **PrintEnv.war** file given to you with this lab

- Click on the "**Deploy**" button. You should now see the application listed on the server:



- To run the application simply click on its hyperlinked name in the application list (or navigate to ***http://localhost:8080/PrintEnv/*** in your browser***)***. You can also undeploy (uninstall) it using the controls to the right of it in the application list

Either way you should see something like:

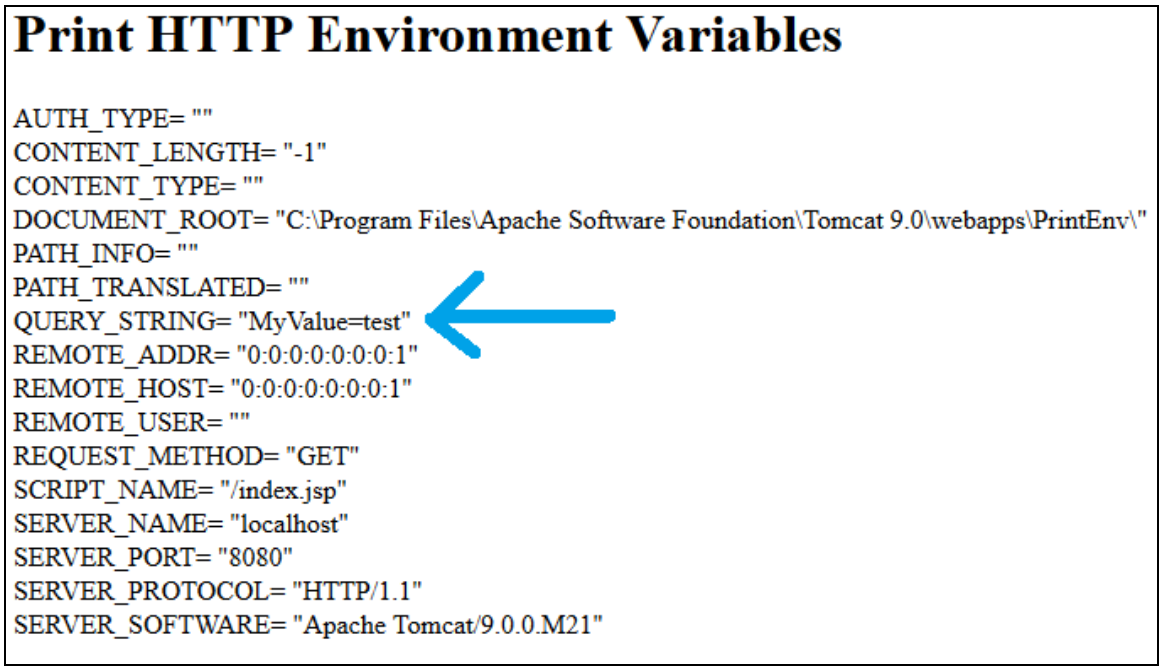
Print HTTP Environment Variables

AUTH_TYPE= ""
CONTENT_LENGTH= "-1"
CONTENT_TYPE= ""
DOCUMENT_ROOT= "C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\PrintEnv\"
PATH_INFO= ""
PATH_TRANSLATED= ""
QUERY_STRING= ""
REMOTE_ADDR= "0:0:0:0:0:0:0:1"
REMOTE_HOST= "0:0:0:0:0:0:0:1"
REMOTE_USER= ""
REQUEST_METHOD= "GET"
SCRIPT_NAME= "/index.jsp"
SERVER_NAME= "localhost"
SERVER_PORT= "8080"
SERVER_PROTOCOL= "HTTP/1.1"
SERVER_SOFTWARE= "Apache Tomcat/9.0.0.M21"

The exact values shown may vary a little on your computer. That's ok.

- Add the following string to the end of the existing URL and press enter to submit the data: ***?MyValue=test***

You should see the value of the ***QUERY_STRING*** variable change to show the submitted data, as before, i.e.



To summarize this section:

1. To run web applications created using Java web technology you must have an application server installed (you have seen Tomcat, but there are many others available)
2. Java web applications cannot run by themselves, they must be installed (deployed) in an application server. NetBeans can do this automatically for you when you run a Java web application
3. The application server manages the web application, and provides many services that the application can use. We will see many of these as we go through the course
4. The Tomcat manager app allows you to manage an application server and what it can do for you
5. In our case the Tomcat application server is integrated with the NetBeans IDE, but it does not have to be, and can run completely standalone if necessary (this is common for production servers)
6. Web applications use HTTP messages to transfer data in exactly the same ways that other types of web applications do, e.g. GET, POST. These messages may include values that are useful to the system, but are not seen on-screen by the user (the environment variables)