Build Application and Deploy to Tomcat

Scout Team

Version 11.0

Table of Contents

Looking for something else? Visit https://eclipsescout.github.io for all Scout related documentation.

At some point during the application development you will want to install your software on a machine that is intended for productive use. This is the moment where you need to be able to build and package your Scout application in a way that can be deployed to an application server.

As Scout applications just need a servlet container to run, Scout applications can be deployed to almost any Java application server. For the purpose of this tutorial we will use Apache Tomcat.

Verify the Container Security Settings

First you need to decide if the users of your application should communicate via HTTPS with the Scout frontend server. We strongly recommended this setup for any productive environment. This is why even the Scout “Hello World” example is configured to use HTTPS.

As a default Tomcat installation is configured to use HTTP only, we need to first verify if the installtion is properly configured for HTTPS too. In case HTTPS support is already enabled for your Tomcat installation, you may skip this section.

Otherwise, check out the configuration process described in the Tomcat Documentation to enable SSL/TLS.

Create and Install a Self-Signed Certificate

This section describes the creation and usage of a self-signed certificat in a localhost setting.

1. Create a keystore file with a self-signed certificate
2. Uncomment/adapt the HTTPS connector port in Tomcat's server.xml configuration
3. Export the self-signed certificate from the keystore
4. Import the self-signed certificate into the Java certificate store

The first step is to create a self-signed certificate using the keytool provided with the Java runtime. The example command line below will create such a certificate using the alias `tomcat_localhost` and place it into the keystore file `tomcat_localhost.jks`

```
keytool.exe -genkey -keyalg RSA -dname CN=localhost -alias tomcat_localhost -keystore
tomcat_localhost.jks -keypass changeit -storepass changeit
```

The second step is to uncomment the HTTPS connector element in the Tomcat's `server.xml` configuration file. Make sure that parameter `keystoreFile` points to your newly created keystore file (if you are using a windows box, make sure not to use the backslash characters in the path to the keystore). After a restart of Tomcat you should then be able to access Tomcat on https://localhost:8443/manager/html

```
    <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               maxThreads="150" SSLEnabled="true" scheme="https" secure="true">
        <SSLHostConfig>
            <Certificate certificateKeystoreFile="/keystore/tomcat_localhost.jks"
                         type="RSA" />
        </SSLHostConfig>
    </Connector>
```

The third step is to export the newly created self-signed certificate from the `tomcat_localhost.jks` keystore file into the `tomcat_localhost.der` certificate file.

```
keytool.exe -exportcert -alias tomcat_localhost -storepass changeit -keystore
tomcat_localhost.jks -file tomcat_localhost.der
```

In the fourth and last step we add the self-signed certificate to the known certificates of the Java runtime. Make sure that you modify the `cacerts` file of the Java runtime that is used in your Tomcat installation and modify the path to the `cacerts` file accordingly.

```
keytool.exe -import -alias tomcat_localhost -trustcacerts -storepass changeit
-keystore C:\java\jre8\lib\security\cacerts -file tomcat_localhost.der
```

Your Scout application should now properly communicate over HTTPS in your Tomcat installation and after having installed the "Hello World" application to Tomcat it should become available on https://localhost:8443/org.eclipse.scout.apps.helloworld.ui.html/.

In case the Scout frontend server cannot access the Scout backend server your self-signed certificate might be missing in the Java installation. To verify that the certificate has been included in file `cacerts` file use the following command.

```
keytool.exe -list -storepass changeit -keystore C:\java\jre8\lib\security\cacerts |
find "localhost"
```

Once you no longer need the self-signed certificate file in your Java installation make sure to remove the certificate again.

```
keytool.exe -delete -alias tomcat_localhost -storepass changeit -keystore
C:\java\jre8\lib\security\cacerts
```

Update the Scout Application to work with HTTP

If you should prefer to work with HTTP only, you need to modify the security settings of your Scout application. This can be done with the steps described below.

- In file `config.properties` (in folder `helloworld.ui.html.app.war/src/main/resources`):
  - Add the property `scout.auth.cookieSessionValidateSecure=false` to disable the check for an encrypted channel (HTTPS).
  - Change the `scout.backendUrl` property to use HTTP instead of HTTPS and change the port according to your Tomcat setup, typically 8080.
- In file `web.xml` (in folder `helloworld.ui.html.app.war/src/main/webapp/WEB-INF`) delete the `<secure>true</secure>` flag in the `<cookie-config>` element.
- In file `web.xml` (in folder `helloworld.server.app.war/src/main/webapp/WEB-INF`) delete the `<secure>true</secure>` flag in the `<cookie-config>` element.

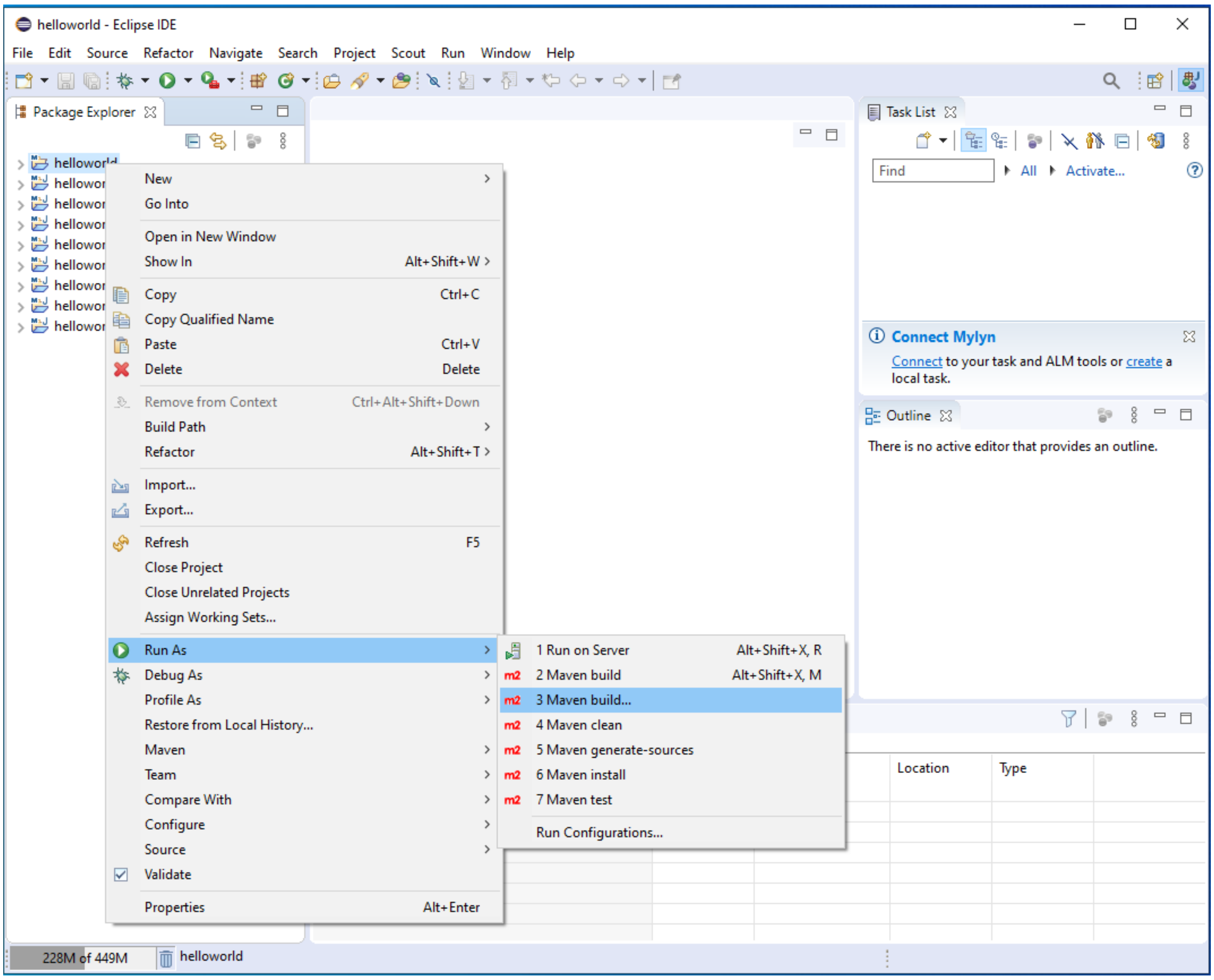
More on this topic can be found in the Scout Architecture Documentation.

Create WAR Files

We are now ready to move the `Hello World` application from our development environment to a productive setup. The simplest option to move our application into the 'wild' is to build it using Maven. This produces two WAR files [1].

The first WAR file contains the Scout backend server with all business logic. The second WAR file contains the Scout frontend server that is responsible for communicating with the web browser part of the Scout application.

To start the build right click on the project `helloworld` and select the context menu **Run As** → **Maven build...** as shown in Figure 1. In the dialog that appears enter `clean verify` into the `Goals` field and press **[Run]**.



Connect to your task and ALM tools or create a local task.

There is no active editor that provides an outline.

Figure 1. Starting the Maven build.

Afterwards the compilation starts, executes all test cases and bundles the result into two WAR files. The output of the build is shown in the Console view within Eclipse. As soon as the build is reporting success you can find the built WAR files:

- The Scout backend WAR file `org.eclipse.scout.apps.helloworld.server.war` in folder `workspace_root/helloworld.server.app.war/target`

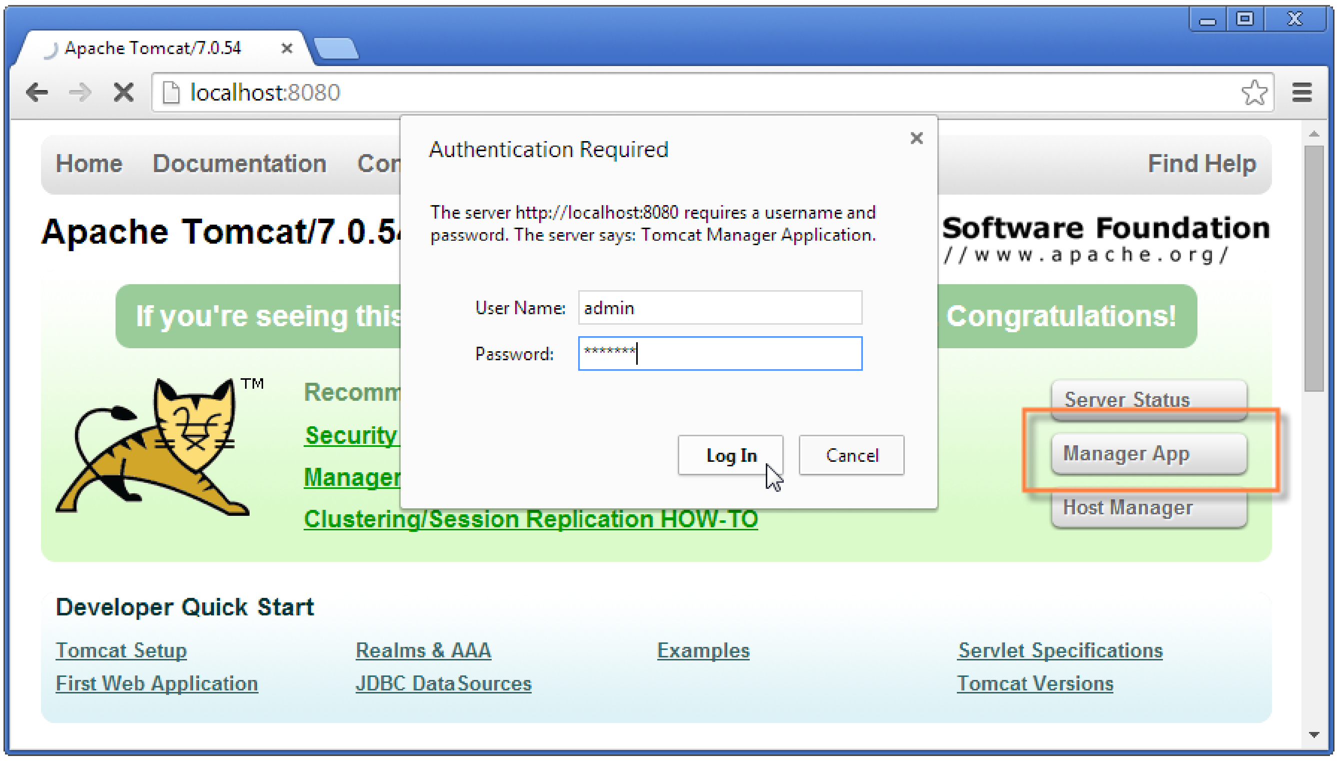
* The Scout frontend WAR file `org.eclipse.scout.apps.helloworld.ui.html.war` in folder `workspace_root/helloworld.ui.html.app.war/target`

To see the new files within Eclipse you may need to refresh the `target` folder below each project using the F5 keystroke.

[1] Web application Archive (WAR): http://en.wikipedia.org/wiki/WAR_file_format_%28Sun%29

Deploy to Tomcat

As the final step of this tutorial, we deploy the two WAR files representing our “Hello World” application to a Tomcat web server. For this, we first need a working Tomcat installation. If you do not yet have such an installation you may want to read and follow the instructions provided in Appendix A. To verify a running Tomcat instance, type http://localhost:8080/ into the address bar of the web browser of your choice. You should then see the page shown in Figure 2.



Authentication Required

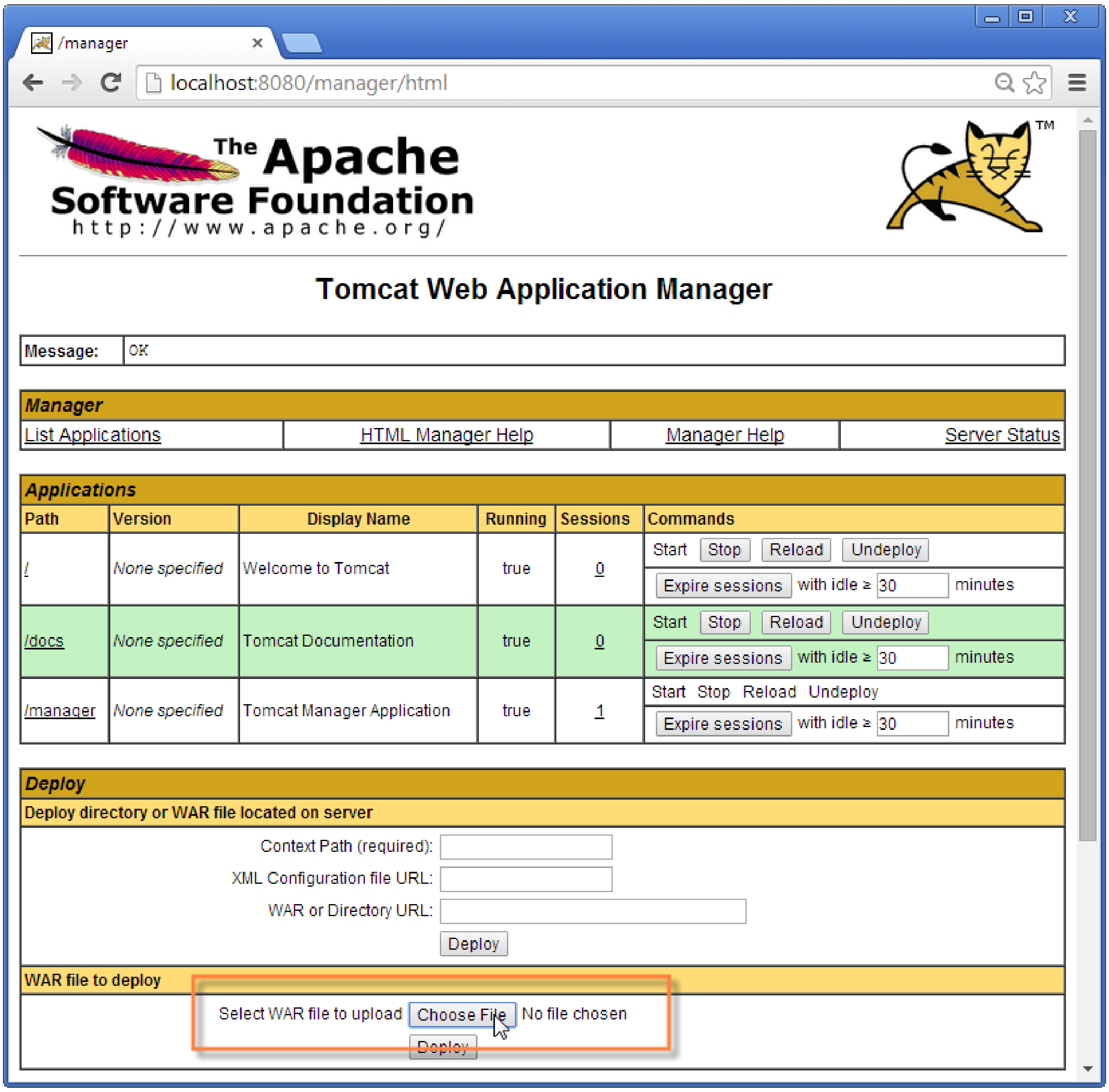
The server http://localhost:8080 requires a username and password. The server says: Tomcat Manager Application.

User Name: admin

Password: *******

Figure 2. The Tomcat shown after a successful installation. After clicking on the “Manager App” button (highlighted in red) the login box is shown in front. A successful login shows the “Tomcat Web Application Manager”.

Once the web browser displays the successful running of your Tomcat instance, switch to its “Manager App” by clicking on the button highlighted in Figure 2. After entering user name and password the browser will display the “Tomcat Web Application Manager” as shown in Figure 3. If you don’t know the correct username or password you may look it up in the file tomcat-users.xml as described in Directories and Files.



Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload Choose File No file chosen

Deploy

Figure 3. The "Tomcat Web Application Manager". The WAR files to be deployed can then be selected using button "Choose File" highlighted in red.

After logging into Tomcat's manager application, you can select the WAR files to be deployed using button "Choose File" according to the right hand side of Figure 3. After picking your just built org.eclipse.scout.apps.helloworld.server.war and closing the file chooser, click on button "Deploy" (located below button "Choose File") to deploy the application to the Tomcat web server. Then we repeat this step with the second WAR file org.eclipse.scout.apps.helloworld.ui.html.war.

This will copy the selected WAR files into Tomcats webapps directory and unpack its contents into subdirectories with the same name. You can now connect to the application using the browser of your choice and enter the following address:

```
http://localhost:8080/org.eclipse.scout.apps.helloworld.ui.html/
```

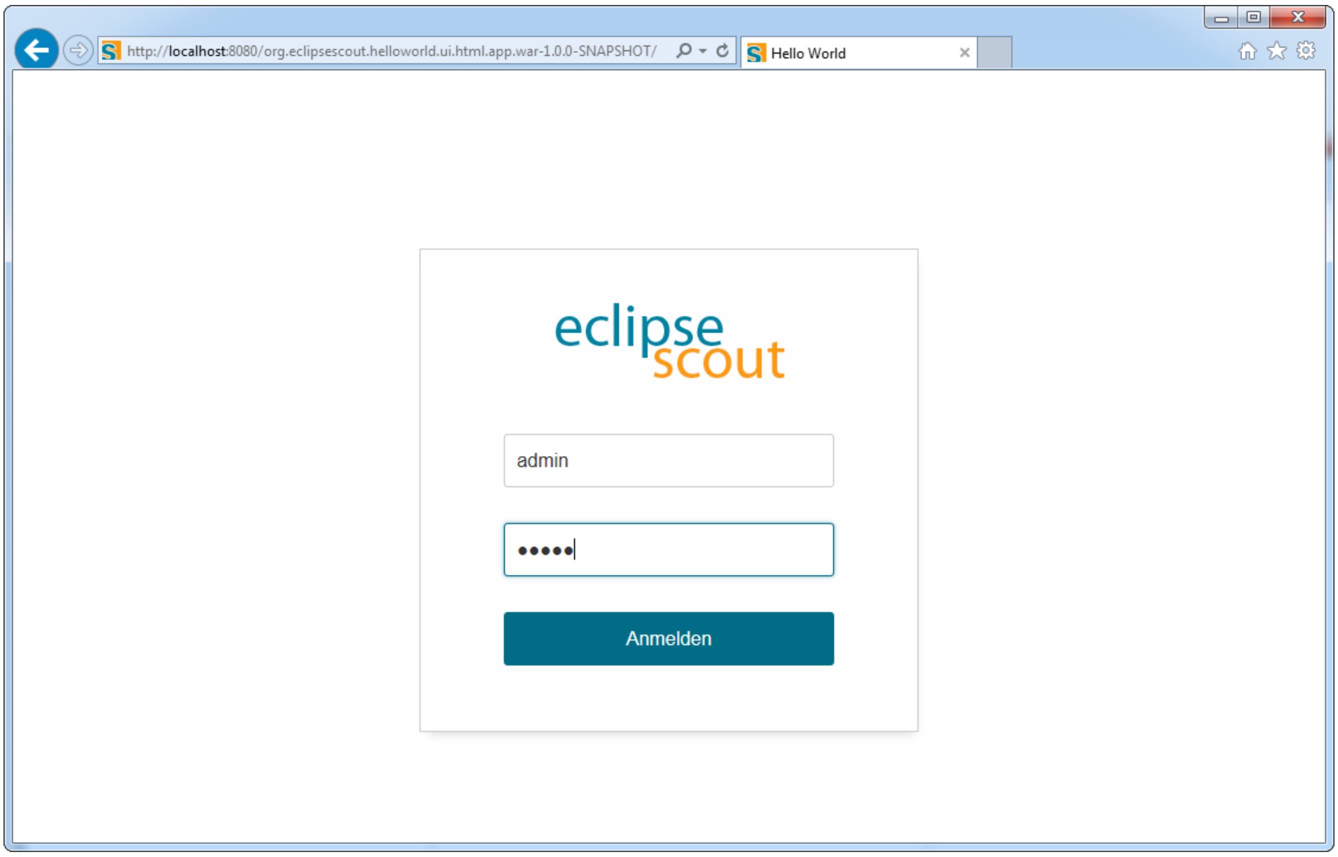

http://localhost:8080/org.eclipsescout.helloworld.ui.html.app.war-1.0.0-SNAPSHOT/
Hello World
admin
Anmelden

Figure 4. The “Hello World” login page.

Then you will see the login page as shown in Figure 4. Two users have been pre defined: “admin” with password “admin” and “scott” with password “tiger”. You can find this configuration in the config.properties file of the application.

Please note: In a productive environment it is recommended to deploy the server and the user interface into two different servlet containers running on dedicated machines. This is because these two tiers have different requirements on resources, load balancing and access protection. Furthermore, it is strongly recommended to use an encrypted connection (e.g. TLS 1.2 [2]) between client browsers and the Scout frontend server AND between the Scout frontend and backend server!

[2] TLS: https://en.wikipedia.org/wiki/Transport_Layer_Security

Appendix A: Apache Tomcat Installation

Apache Tomcat is an open source web server that is a widely used implementation of the Java Servlet Specification. Specifically, Tomcat works very well to run Scout applications. In case you are interested in getting some general context around Tomcat you could start with the Wikipedia article.[3]. Then get introduced to its core component “Tomcat Catalina”.[4] before you switch to the official Tomcat homepage.[5].

This section is not really a step by step download and installation guide. Rather, it points you to the proper places for downloading and installing Tomcat. We recommend to work with Tomcat version 9.0. Start your download from the official download site.[6].

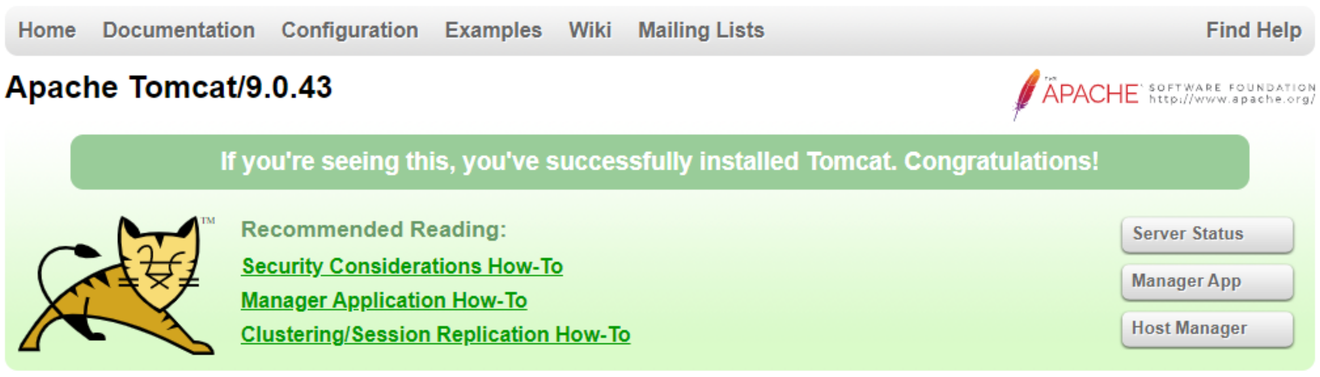

Home
Documentation
Configuration
Examples
Wiki
Mailing Lists
Find Help
Apache Tomcat/9.0.43
APACHE SOFTWARE FOUNDATION http://www.apache.org/
If you're seeing this, you've successfully installed Tomcat. Congratulations!
Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To
Server Status
Manager App
Host Manager

Figure 5. A successful Tomcat installation.

Once you have downloaded and installed Tomcat 9 (see the sections below for plattform specific guidelines) you can start the corresponding service or deamon. To verify that Tomcat is actually running open a web browser of your choice and type http://localhost:8080 into the address bar. You should then see a confirmation of the successful installation according to Figure 5.

Platform Specific Instructions

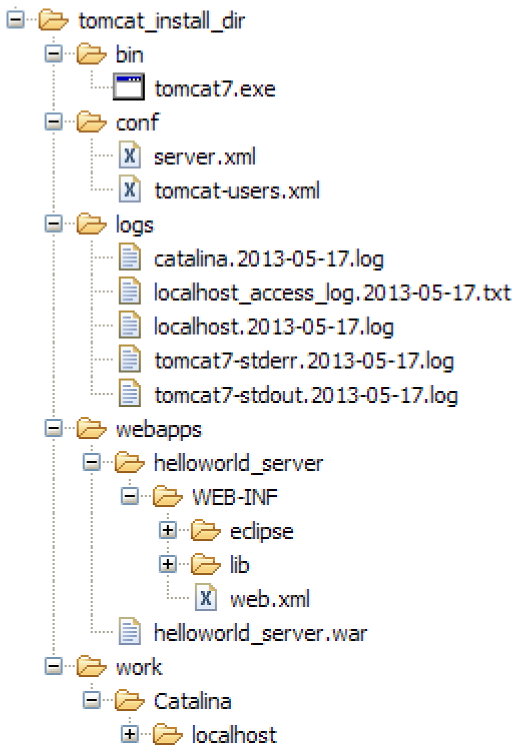
According to the Tomcat setup installation for Windows.[7] download the package “32-bit/64-bit Windows Service Installer” from the Tomcat 9 download site. Then, start the installer and accept the proposed default settings.

For installing Tomcat on OS X systems download the “tar.gz” package from the Tomcat 9 download site. Then, follow the installation guide.[8] provided by Wolf Paulus.

For Linux systems download the “tar.gz” package from the Tomcat 9 download site. Then, follow the description of the Unix setup.[9] to run Tomcat as a deamon. If you use Ubuntu, you may want to follow the tutorial.[10] for downloading and installing Tomcat provided by Lars Vogel.

Directories and Files

Tomcat’s installation directory follows the same organisation on all platforms. Here, we will only introduce the most important aspects of the Tomcat installation for the purpose of this book.



```
tomcat_install_dir
    bin
        tomcat7.exe
    conf
        server.xml
        tomcat-users.xml
    logs
        catalina.2013-05-17.log
        localhost_access_log.2013-05-17.txt
        localhost.2013-05-17.log
        tomcat7-stderr.2013-05-17.log
        tomcat7-stdout.2013-05-17.log
    webapps
        helloworld_server
            WEB-INF
                eclipse
                lib
                web.xml
        helloworld_server.war
    work
        Catalina
            localhost
```

Figure 6. The organisation of a Tomcat installation including specific files of interest. As an example, the "Hello World" server application is contained in subdirectory `webapps`.

Note that some folders and many files of a Tomcat installation are not represented in Figure 6. We just want to provide a basic understanding of the most important parts to operate the web server in the context of this book. In the bin folder, the executable programs are contained, including scripts to start and stop the Tomcat instance.

The conf folder contains a set of XML and property configuration files. The file server.xml represents Tomcat's main configuration file. It is used to configure general web server aspects such as the port number of its connectors for the client server communication. For the default setup, port number 8080 is used for the communication between clients applications and the web server. The file tomcat-users.xml contains a database of users, passwords and associated roles.

Folder logs contains various logfiles of Tomcat itself as well as host and web application log files. TODO [7.0] jbr: need to provide more on what is where (especially application logs and exact setup to generate log entries from scout apps).

The folder needed for deploying web applications into a Tomcat instance is called webapps. It can be used as the target for copying WAR files into the web server. The installation of the WAR file then extracts its content into the corresponding directory structure as shown in Figure 6 in the case of the file helloworld_server.war.

Finally, folder work contains Tomcat's runtime "cache" for the deployed web applications. It is organized according to the hierarchy of the engine (Catalina), the host (localhost), and the web application (helloworld_server).

The Tomcat Manager Application

Tomcat comes with the pre installed "Manager App". This application is useful to manage web applications and perform tasks such as deploying a web application from a WAR file, or starting and stopping installed web applications. A comprehensive documentation for the “Manager App” can be found under the Tomcat homepage.[11]. Here we only show how to start this application from the hompage of a running Tomcat installation.

To access this application you can switch to the “Manager App” with a click on the corresponding button on the right hand side. The button can be found on the right hand side of Figure 5. Before you are allowed to start this application, you need to provide username and password credentials of a user associated with Tomcats’s manager-gui role.

Listing 1. Tomcat Users configuration file.

```
<tomcat-users>
  <!--
  NOTE: By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application. If you wish to use it
  you must define such a user - the username and password are arbitrary.
  -->
  <user name="admin" password="s3cret" roles="manager-gui"/>
</tomcat-users>
```

To get at user names and passwords you can open file tomcat-users.xml located in Tomcat’s conf directory. In this file the active users with their passwords and associated roles are stored. See Listing Listing 1 for an example. From the content of this file, we see that user admin has password s3cret and also posesses the necessary role manager-gui to access the “Manager App”. If file tomcat-users.xml does not contain any user with this role, you can simply add new user with this role to the existing users. Alternatively, you also can add the necessary role to an existing user. Just append a comma to the existing right(s) followed by the string manager-gui. Note that you will need to restart your Tomcat application after adapting the content of file tomcat-users.xml.

With working credentials you can now start the “Manager App” as described the “Hello World” tutorial in *Deploy to Tomcat*.

Do you want to improve this document? Have a look at the sources on GitHub.

[3] Apache Tomcat Wikipedia: http://en.wikipedia.org/wiki/Apache_Tomcat.

[4] Mulesoft’s introduction to Tomcat Catalina: http://www.mulesoft.com/tomcat-catalina.

[5] Apache Tomcat Homepage: http://tomcat.apache.org/

[6] Tomcat 9 Downloads: http://tomcat.apache.org/download-90.cgi

[7] Tomcat Windows setup: http://tomcat.apache.org/tomcat-9.0-doc/setup.html#Windows

[8] Installing Tomcat on OS X: http://wolfpaulus.com/journal/mac/tomcat8

[9] Tomcat Linux setup: http://tomcat.apache.org/tomcat-9.0-doc/setup.html#Unix_daemon

[10] Apache Tomcat Tutorial: http://www.vogella.com/articles/ApacheTomcat/article.html

[11] The Tomcat Manager Application: http://tomcat.apache.org/tomcat-9.0-doc/manager-howto.html.