Servlets Tutorial

Vania Marangozova-Martin

ids.forge.imag.fr

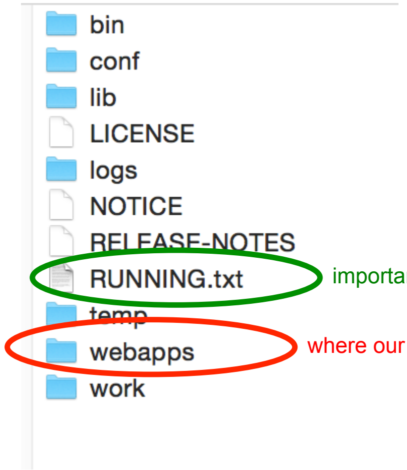
The Webserver Installation

- Install TomCat
  - Download it from ids.forge.imag.fr
    - `apache-tomcat-8.0.30-students.tar.gz`
  - Unzip it in a directory
    - `tar xfz apache-tomcat-8.0.30.tar.gz`
  - Download additional files
    - `student-files.tar.gz`

Defining environment variables

- define the variable `CATALINA_HOME`
  - the root directory where TomCat is installed
  - tcsh: `setenv CATALINA_HOME …/apache-tomcat-8.0.30`
  - bash: `export CATALINA_HOME=..../apache-tomcat-8.0.30`
  - Windows: `set CATALINA_HOME=..../apache-tomcat-8.0.30`
- you can add `$CATALINA_HOME/bin to` you `PATH`
- define the variable JAVA_HOME
  - where your JDK is
- (define the variable JRE_HOME)

In the TomCat Directory

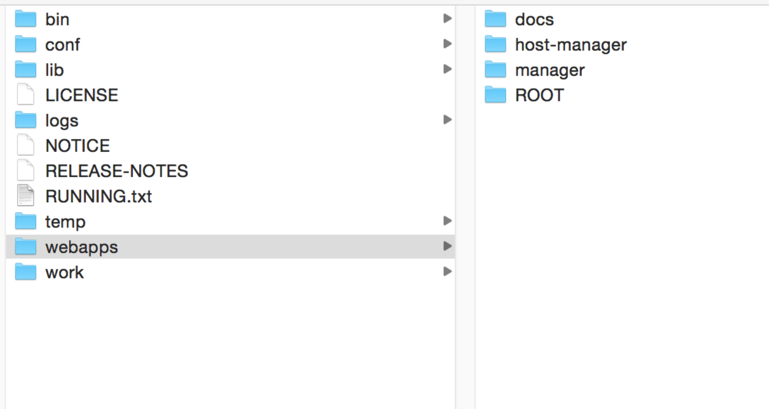


```
bin
conf
lib
LICENSE
logs
NOTICE
RELEASE-NOTES
RUNNING.txt
temp
webapps
work
```

important information on starting TomCat

where our servlets applications will be

The webapps directory

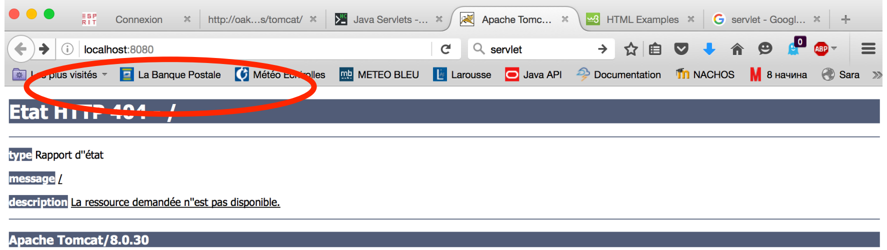


bin
conf
lib
LICENSE
logs
NOTICE
RELEASE-NOTES
RUNNING.txt
temp
webapps
work

docs
host-manager
manager
ROOT

You can run the server…

- Run `$CATALINA_HOME/bin/startup.sh` on a Unix machine (or `$CATALINA_HOME/bin/startup.bat` for Windows).
- For the moment being we have not provided the default content so…



Etat HTTP 404 - /

type Rapport d"état

message /

description La ressource demandée n"est pas disponible.

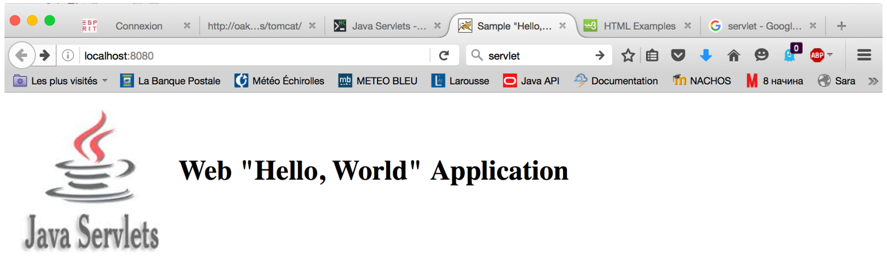
Apache Tomcat/8.0.30

The server port

- The Apache Tomcat Servlet server is started on a particular port specified in the $CATALINA_HOME/conf/server.xml configuration file.
- The default port for accepting HTTP connections is 8080
  - see the corresponding "Connector port" value in the server.xml configuration file

Add some content

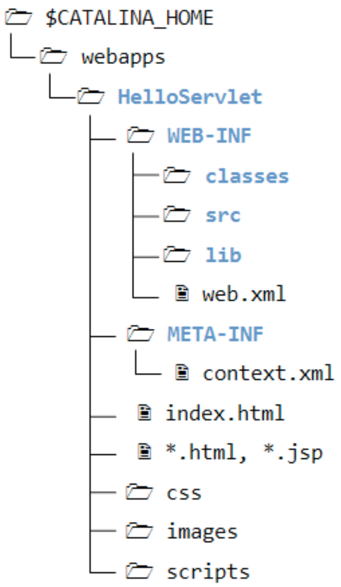
- Copy sampleIndex.html as index.html in webapps/ROOT
- Copy the index.jpg image also
- Reload…



Web "Hello, World" Application

Define a servlet-based application

- The organization of webapps is predefined
- Here is an example



```
$CATALINA_HOME
└─ webapps
   └─ HelloServlet
      ├─ WEB-INF
      │  ├─ classes
      │  ├─ src
      │  ├─ lib
      │  └─ web.xml
      ├─ META-INF
      │  └─ context.xml
      ├─ index.html
      ├─ *.html, *.jsp
      ├─ css
      ├─ images
      └─ scripts
```

Defining our first servlet app

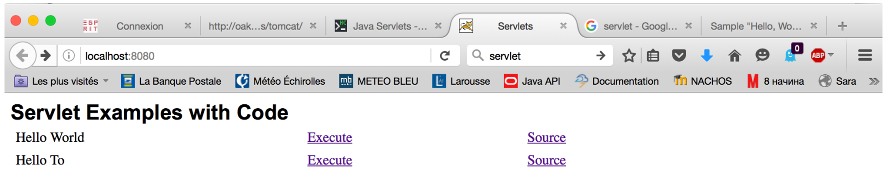
- In webapps/ROOT
  - create WEB-INF
    - create WEB-INF/classes
    - create WEB-INF/src
    - create WEB-INF/lib
- Copy web.xml from student-files
  - put it into WEB-INF
- Copy index.html from student-files
  - put it into webapps/ROOT
- Copy HelloToServlet.java and HelloWorldServlet.java into WEB-INF/src

Our first servlet app…

- Compile the java files and put the result in WEB-INF/classes
  - to compile you will need `apache-tomcat-8.0.30-students/lib/servlet-api.jar` in your `CLASSPATH`

```
javac -cp apache-tomcat-8.0.30-students/lib/servlet-api.jar
apache-tomcat-8.0.30-students/webapps/ROOT/WEB-INF/src/*.java
-d apache-tomcat-8.0.30-students/webapps/ROOT/WEB-INF/
classes/
```

You should have…



Servlet Examples with Code

Hello World	Execute	Source
Hello To	Execute	Source

The index.html

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"
    <title>Servlets</title>
</head>
<body bgcolor="#FFFFFF">
<font size=+2>Servlet Examples with Code</font></font></b>
<table BORDER=0 CELLSPACING=5 WIDTH="85%" >
    <tr VALIGN=TOP>
        <td>Hello World</td>
        <td VALIGN=TOP WIDTH="30%">
            <a href="servlet/HelloWorldServlet">Execute</a></td>
        <td WIDTH="30%">
            <a href="src/HelloWorldServlet.java">Source</a></td>
    </tr>

…
</table>
</body>
</html>
```

The web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app ...>
  <display-name>Servlets</display-name>
  <description> Servlets </description>-

<!--
 Define servlets that are included in the example application
-->
- <servlet>
<servlet-name>HelloWorldServlet</servlet-name>
<servlet-class>HelloWorldServlet</servlet-class>
</servlet>

- <servlet-mapping>
<servlet-name>HelloWorldServlet</servlet-name>
<url-pattern>/servlet/HelloWorldServlet</url-pattern>
</servlet-mapping>


</web-app>
```

The servlets

```
public class HelloWorldServlet extends HttpServlet {

    public void doGet( HttpServletRequest request,
                       HttpServletResponse response)
       throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello World!</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

The servlets

```
public class HelloToServlet extends HttpServlet {

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
       throws IOException, ServletException{
       String firstName, lastName;
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello To</title>");
        out.println("</head>");
        out.println("<body>");
   firstName = request.getParameter("firstname");
   lastName= request.getParameter("lastname");
        if (firstName != null && lastName != null) {
            out.println("<h1>Hello " + firstName +
                    " " + lastName + "!</h1>");
…
```

Question

- Change the servlet mapping so as to access these servlets at

```
http://<host>:<port>/<your first name>/HelloWorldServlet
http://<host>:<port>/<your pet’s name>/HelloToServlet
```

Question

- Type the following URL request on your web browser

```
http://<host>:<port>/servlet/
HelloToServlet?
firstname=John&lastname=Malkovich
```

  - where <host> and <port> are respectively the host name and the port on which the Servlet server is running.

- What does this request do?

Question

- Modify the application in order for a web client to be able to perform a "Hello To John Malkovich" request directly from the http://<host>:<port> main page.
- What are the web application's parts/files that are impacted by this change?
- Take look at `$CATALINA_HOME/conf/web.hml` `<welcome-file-list>`

Session Management

- Copy the `SessionServlet.java` from student-files to webapps/ROOT/WEB-INF/src
- Compile and deploy the servlet
  - compile with javac and the proper CLASSPATH
  - change the web.xml descriptor
- Execute
- Understand the code

Cookie Management

- Copy the `Counter.java` from student-files to webapps/ROOT/WEB-INF/src
- Compile and deploy the servlet
  - compile with javac and the proper CLASSPATH
  - change the web.xml descriptor
- Execute
- Understand the code