Advanced Exercise on Computer and Information Science B

Java Servlet/Java Server Pages

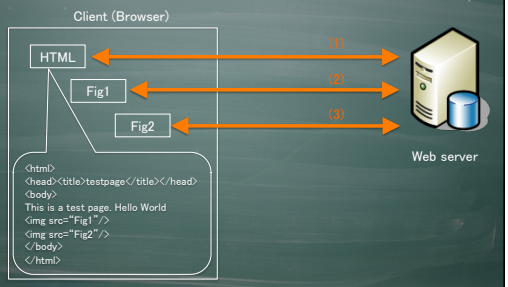
Servlet/JSP

- Servlet?
  - A framework to develop an web application in Java
  - Use class
  - latest – Servlet 4.0 (2014/9)
- JSP?
  - Embed Java logic into HTML
  - Is compiled to Servlet when accessed at the first time
  - Latest – JSP 2.2 (2014/9)

Features of HTTP

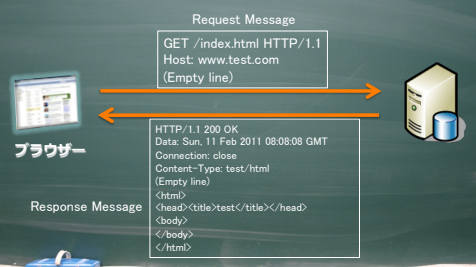
- Features
  - A few instructions, Simple!
  - Does not need complicated negotiations. Low overhead
  - Stateless
- HTTP protocol
  - Request from a client
  - Return a response from server
  - Based on Text (not Binary format)

Interaction between browser and server



HTTP Basis

- Simple Reques/Response



Request

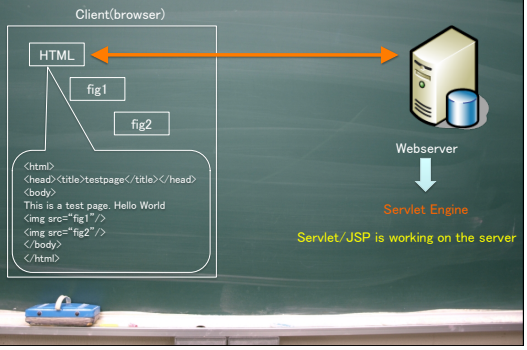
Method request URI HTTP version

e.x) GET /index.html HTTP/1.1

Major methods

Method	Function
GET	Obtain resource specified by URL
POST	Transfer data to the specified URL. The data is included in the request body
HEAD	Obtain only header specified by URL
PUT	Replace data of specified URL with sending data
DELETE	Delete resource specified by URL

Environment required Servlet/JSP



Servlet Engine

- Tomcat – Apache Software Foundation
  - Defact standard
- Resin – CAUCHO
- WebShere – IBM
- WebLogic – Oracle
- Jetty – Eclipse
  - Recommend but . . .

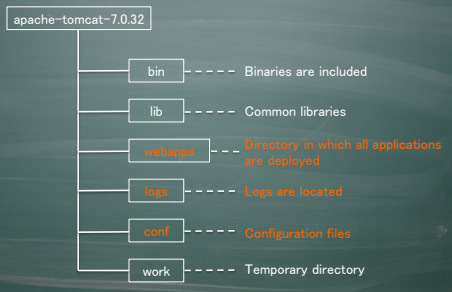
Today's Objective

- Run Tomcat on Linux
  - Set Environment variables
    - CATALINA_HOME, JAVA_HOME…
  - Startup/shutdown operations
- Run HelloWorld servlet
  - Compile and deploy a servlet (class)
  - Understand the directory structure of J2EE
  - Configure servlet environment

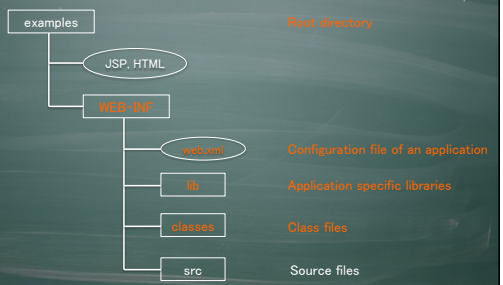
Download

- Download Tomcat
  - http://tomcat.apache.org/

Directory structure



Deployment of an application– J2EE

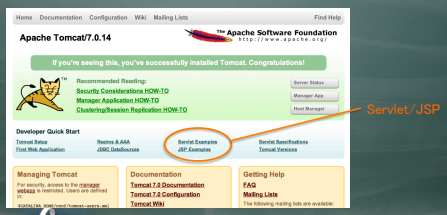


Configurations to run Tomcat

- CATALINA_HOME
  - A directory created by decompression
    - apache-tomcat-7.0.32
- JAVA_HOME
  - JDK base directory
    - /usr/java/jdk1.6
- Assigned by environment variables in init files
  - .cshrc, .bashrc

Startup/Shutdown/Confirm of Tomcat

- Startup – ${CATALINA_HOME}/bin/startup.sh
- Shutdown – ${CATALINA_HOME}/bin/shutdown.sh
- Confirm – http://localhost:8080/



When you fine errors

1. See ${CATALINA_HOME}/logs/catalina.out
2. ${CATALINA_HOME}/bin/shutdown.sh
3. Fix problems
4. ${CATALINA_HOME}/bin/startup.sh
5. http://localhost:8080/ for confirmation
6. Repeat from 1 to 5

Run HelloWorld Servlet

- Instruction
  1. Create source code
  2. Compile
  3. Deploy it as an application
  4. Register it as a servlet
  5. Confirm

1. Create source code

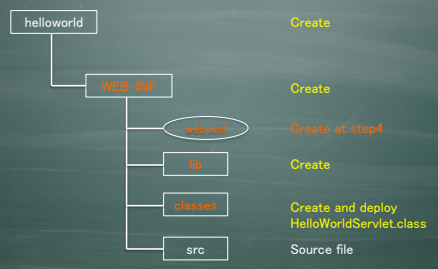
```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class 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("<body>");
                out.println("Hello World");
                out.println("</body>");
                out.println("</head>");
                out.println("</html>");
        }
}
```

2. Compile

1. Add ${CATALINA_HOME}/lib/servlet-api.jar to the CLASSPATH
2. % javac HelloWorldServlet.java
   - HelloWorldServlet.class is created

3. Deploy it as an application



4. Register as a servlet – web.xml

1. Copy Examples/WEB-INF/web.xml
2. Remove unnecessary parts
   - Mandatory tags
     - servlet
     - servlet-mapping
3. Deploy it on WEB-INF

An example of web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                      http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0"
  metadata-complete="true">

  <!-- Define servlets that are included in the example application -->

  <servlet>
    <servlet-name>HelloWorld</servlet-name>
    <servlet-class>HelloWorldServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>HelloWorld</servlet-name>
    <url-pattern>/HelloWorld</url-pattern>
  </servlet-mapping>
</web-app>
```

Definision of servlet name and class
Deployed classes
Including package

Mapping of Servlet and URL

Confirm

- http://localhost:8080/helloworld/HelloWorld
  Folder name / Servlet name
- If errors are found
  - Confirm from step1 to 4
  - Obviously you made mistakes!

Java Servlet

- Main package
  - javax.servlet
  - javax.servlet.http
  - Refer to javadoc in detail
- How to create a servlet
  - Extend HttpServlet
  - Override required methods
    - doGet/doPost

Methods of HttpServlet

HTTPのメソッド	メソッド
GET	proteced void doGet
POST	protected void doPost
PUT	protected void doPut
DELETE	protected void doDelete
HEAD	protected void doHead
TRACE	protected void doTrace
OPTIONS	protected void doOptions

Arguments are HttpServletRequest req, HttpServletResponse res

Basic process in a method – doGet

```
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("<body>");
        out.println("Hello World");
        out.println("</body>");
        out.println("</head>");
        out.println("</html>");
    }
}
```

Override

Specify Content-Type

HTML

HttpServletRequest – include request information

- Obtain information sent by browser
- Major methods
  - String getParameter(String name)
    - Obtain a value specified by name
  - String[] getParameterValues(String name)
    - Obtain values specified by name
      - Several values are sent by the same name
  - String getHeader(String name)
    - Obtain a header value specified by name
  - See javadoc in detail

HttpServletResponse – output response

- Output response to a client (browser)
- Major methods
  - void setContentType(String type)
    - Specify content type
  - PrintWriter getWriter() throws IOException
    - Obtain output stream
  - void setRedirect(String location)
    - Redirects a client to the location specified
  - void setStatus(int sc)/void setHeader(String name, String value)
    - Set Status code/header value

Session Management – HttpSession

- Start Session
  - HttpSession ses = request.getSession(true/false)
    - true : return the session. Create if it does not start yet
    - false : return the session. Return null if it does not start yet
- Set/Get values
  - void setAttribute(String key, Object value)
  - Object getAttribute(String key)
- Remove a value
  - void removeAttribute(String key)

Implement a registration management system that consists of 3 screens

- RegistServlet
  - Receive name and address. Then move to the confirm when the registration button is clicked
- ConfirmServlet
  - Display the name and address. Then move to the complete
- CompleteServlet
  - Display completion message. Need not to store the information