

# Unraveling the Mysteries of J2EE Web Application Communications –An HTTP Primer

Peter Koletzke  
Technical Director &  
Principal Instructor



## Common Problem

What we've got here is ...  
failure to commun'cate.

–Captain, *Cool Hand Luke* (1967)



## Survey

- Jobs
  - Developer?
  - DBA?
  - Sys admin, others?
- Web Application Work
  - J2EE?
  - .NET?
  - PHP, ColdFusion, others?
- Tools
  - JDeveloper
  - Eclipse
  - Others



## Agenda

- HTTP Request/Response
- HTTP Details
- J2EE Specifics
- Web Roundtrip for J2EE

Slides and white paper will  
be available on the Quovera  
and NoCOUG websites



## What is HTTP?

- Hypertext Transfer Protocol
- A communications protocol
  - Runs on Transport Control Protocol/Internet Protocol (TCP/IP)
  - Guideline for formatting messages between systems
- The main protocol used for J2EE web communications
  - Web Servers are technically *HTTP Servers*
- Web browser is HTTP-capable, also runs:
  - File Transfer Protocol (FTP)
  - Lightweight Directory Access Protocol (LDAP)
  - mailto
  - HTTPS (HTTP Secure)



## Why Do We Care?

- Usually, you do not program at the protocol level
- Knowing low level details helps in debugging web applications
- Also helps when programming
  - You know what information is available to the session
  - You know the standard browser services that support your program



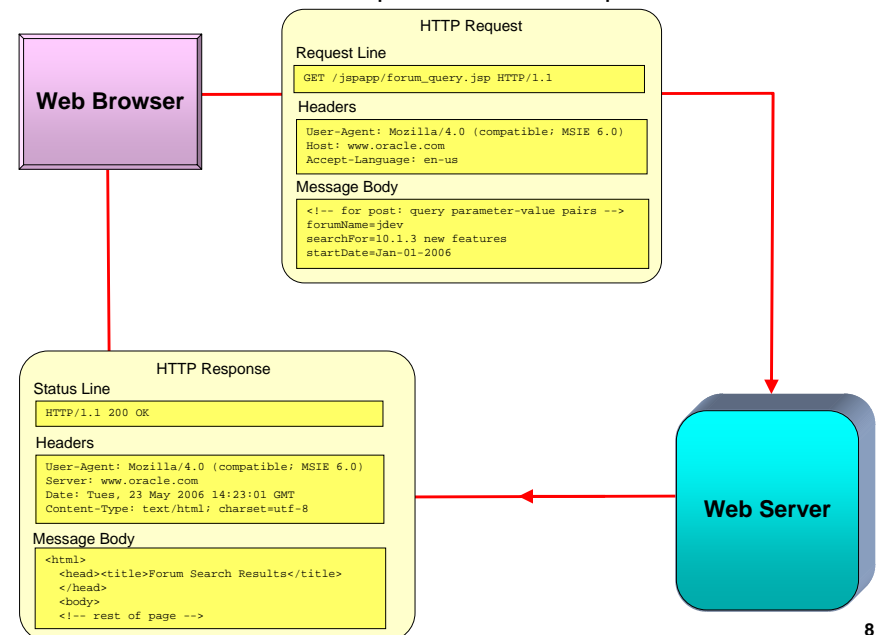
Think of this as like playing scales.

## Web Roundtrip Messages

- Request
  - Ask for resource (HTML page, image file) or action
  - Sent when user clicks a button or link
- Response
  - Return message
  - Can include browser content



## HTTP Request and Response



# Request Contents

- Request line
  - Method
  - Uniform Resource Identifier (URI)
    - Uniquely identifies a resource on the Web
    - A URL is a subset of URI used for HTTP
  - HTTP version (1.1 is current)
- Header fields (headers)
  - Name-value pairs
  - Identify the requestor
  - Indicate how the content will be obtained
  - [www.w3.org/Protocols/rfc2616/rfc2616.html](http://www.w3.org/Protocols/rfc2616/rfc2616.html)
- Message body
  - Used for POST request (more later)



# Sample Request Header

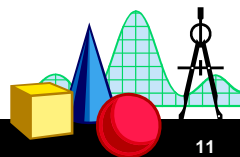
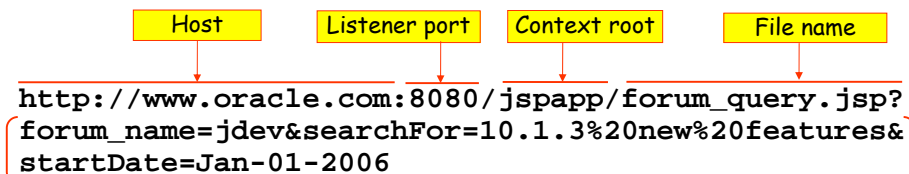
Request Header Name	Request Header Value
Host	www.oracle.com
User-Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.7) Gecko/20060909 Firefox/1.5.0.7
Accept	text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language	en-us,en;q=0.5
Accept-Encoding	gzip,deflate
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive	300
Connection	keep-alive
Cookie	s_cc=true; s_sq=%5B%5B%5D%5D; homepagenum=0

- Browser sends URL request to [www.oracle.com](http://www.oracle.com)
- Displayed using Firefox's add-on Tamper Data
  - <http://addons.mozilla.org/firefox/966/>



# About URLs

- Uniform Resource Locator (URL)
  - Subset of Uniform Resource Identifier notation
  - Example (this would be entered on a single line):



# URL Components

- Host name
  - Uses dot separators between hierarchical components
- Web server listener port
  - Operating system port number – a process
    - Access point associated with a function
  - No port usually means port 80
- Context root
  - Application directory, top-level directory for a web application
  - Also called *virtual directory*
  - Mapped to a physical directory on the server
- File name
  - Followed by ? to separate file name and parameters
- Query parameters
  - Parameter name and value pairs
  - Delimited with & symbol
- Bookmark (optional, not shown), uses # delimiter
  - Place on the page to which the browser navigates
  - Named anchor in HTML (<a name="here">)



## Response Contents

- Status line
  - HTTP version of the response
  - Status code

100s = still processing  
200s = response processed  
300s = redirection problem

400s = error with the request  
500s = server failure

- Reason phrase
  - Readable version of status code (like “OK”)
- Header fields
  - Similar to request
- Message body – the content



## Sample Response Header

Response Header Name	Response Header Value
Status	Moved Temporarily - 302
Set-Cookie	BIGipServerwoc_prod_pool_10g=2198704781.24862.0000; expires=Tue, 17-Oct-2006 22:57:23 GMT; path=/ max-age=0, max-age=0
Cache-Control	max-age=0, max-age=0
Location	http://www.oracle.com/index.html
Content-Type	text/plain
Connection	Keep-Alive
Keep-Alive	timeout=7, max=999
Server	Oracle-Application-Server-10g OracleAS-Web-Cache/10.1.2.2 (H;max-age=300+0;age=15;eid=216172884479614531,0)
Content-Length	0
Date	Tue, 17 Oct 2006 22:52:07 GMT
Vary	Host
Content-Location	/servlet/page/occm/

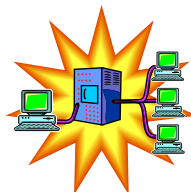
- Response from www.oracle.com
- Also displayed using Tamper Data

Demo



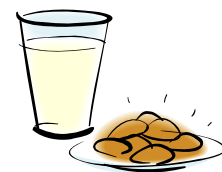
## Agenda

- HTTP Request/Response
- HTTP Details
- J2EE Specifics
- Web Roundtrip for J2EE



## State and Cookies

- HTTP defines a *stateless* connection
  - Server does not know that a request was issued by a client that issued a previous request
  - No good for e-commerce and database transaction applications
- Cookie allows server to store info on client
  - Cookie (name and value) associated with server and sent to that server with next request
  - Stored in file or cached in browser memory
  - Ties one session to another
    - E.g., a shopping cart
- J2EE apps use HttpSession
  - Java API for maintaining state



# Sample Cookies

- From the Firefox add-on *Web Developer Toolkit* utility – View Cookie Information
  - <http://addons.mozilla.org/firefox/60>



## Cookie Information - <http://www.oracle.com/index.html>

<http://www.oracle.com/index.html>

NAME	s_sq
VALUE	%5B%5BB%5D%5D
HOST	oracle.com
PATH	/
SECURE	No
EXPIRES	At End Of Session

NAME	BIgipServerwoc_prod_pool_10g
VALUE	2198704781.24862.0000
HOST	www.oracle.com
PATH	/
SECURE	No
EXPIRES	Wednesday, October 18, 2006 1:33:18 PM

NAME	homepagenum
VALUE	2
HOST	www.oracle.com
PATH	/
SECURE	No
EXPIRES	At End Of Session

quovera

Demo

17

# HTTP Methods

- Commands to the server issued by the client
- The most commonly used:
  - GET
  - POST
  - HEAD
- Others
  - OPTIONS – get info about a resource
  - PUT – user can copy a file to the server
  - DELETE – user can delete a file from the server
  - TRACE – sends the entire request back to the client; good for debugging



quovera

18

# GET

- Retrieves content from the server based on the URL, usually just retrieving a file
- Requests that can be safely repeated without side effects
  - For example, viewing the shopping cart
  - Reissuing a request to view has no side effects
- Supplies parameters via the URL line
  - Query parameters mentioned above
  - Limited amount of data – 2083 characters
  - Query parameters are visible to the user
    - Potential hacking hint to savvy users



quovera

19

# POST and HEAD

- POST sends information to the server
- Best for requests that cannot be repeated safely
  - For example, adding to the shopping cart
    - Resubmitting the request will add to the count
  - Requests cannot be resubmitted
- Parameter values and names are hidden (are not in the URL)
- Used from an HTML form submission
- HEAD - same as POST but response is only headers and resource size (no body)
  - Client can then request resource conditionally



quovera

20

## Other HTTP Features

- Redirection
  - Servers can send a request to another location
  - Status code in the 300s sent to the client
  - One use is to cause the client to show a page from its cache
- HTTPS
  - HTTP Secure
  - Used in an encrypted Secure Socket Layer (SSL) session
  - Only a key shared between client and server allows the messages to be read



## Agenda

- HTTP Request/Response
- HTTP Details
- J2EE Specifics
- Web Roundtrip for J2EE



## J2EE on the Web

- Use this application style as an example
  - Oracle is focusing on J2EE
- Technically “Java Platform”
  - No longer “J2EE,” but “Java EE 5”
- J2EE web client is a web browser
  - Uses HTTP to communicate with web server (HTTP server)
  - Other clients possible now (PDA, cell phone, telnet)
- J2EE has a specific server setup
  - Specific configuration files



## server.xml

- Server configuration file
  - Standard J2EE descriptor in XML format
- In ../j2ee/home/config
  - Web server parses context root from URL
  - Then it looks in server.xml for information about how to handle the request
    - Static HTML?
    - Virtual directory signifies context root?

```
Context root
<application name="jspapp"
  path="../applications/jspapp" />
Physical directory
```





## server.xml Excerpt

```
<?xml version="1.0" ?>
<application-server xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/application-server-10_1.xsd" application-directory=".." applications check-for-updates="adminClientOnly" deployment-directory=".." application-deployments" connector-directory=".." connectors" schema-major-version="10" schema-minor-version="0">
- <shared-library name="global.libraries" version="1.0" library-compatible="true">
  <code-source path=".." applib" />
</shared-library>
+ <shared-library name="global.tag.libraries" version="1.0" library-compatible="true">
+ <shared-library name="global.wsm.libraries" version="1.0" library-compatible="true">
+ <shared-library name="oracle.expression-evaluator" version="10.1.3" library-compatible="true">
+ <shared-library name="adf.oracle.domain" version="10.1.3" library-compatible="true">
+ <shared-library name="adf.generic.domain" version="10.1.3" library-compatible="true">
<rmi-config path=".." rmi.xml" />
<jms-config path=".." jms.xml" />
<javacache-config path=".." /.. /javacache/admin/javacache.xml" />
<j2ee-logging-config path=".." j2ee-logging.xml" />
- <log>
  <file path=".." log/server.log" />
</log>
<java-compiler name="javac" in-process="false" options="-J-Xmx1024m -encoding UTF8"
extdirs="C:\JDev1013\jdk\jre\lib\ext" />
<global-application name="default" path="application.xml" parent="system" start="true" />
<application name="ascontrol" path=".." /home/applications/ascontrol.ear" parent="system"
start="true" />
<application name="bc4j" path=".." /.. /BC4J/redist/bc4j.ear" start="true" />
<application name="tuhraapp" path=".." /applications/tuhraapp.ear" parent="default" start="true" />
<global-web-app-config path="global-web-application.xml" />
<transaction-manager-config path="transaction-manager.xml" />
<web-site default="true" path=".." default-web-site.xml" />
<cluster id="122947750592639" />
</application-server>
```

Demo

25

## web.xml

- Web Module Deployment Descriptor
- Standard J2EE XML descriptor file
  - Contains details about the application
- In ../j2ee/applications/<appname>/WEB-INF
  - Server looks in WEB-INF directory under directory listed in server.xml
  - Example entry:

```
<welcome-file-list>
  <welcome-file>forum_query.jsp</welcome-file>
</welcome-file-list>
```

Startup file



quovera

26

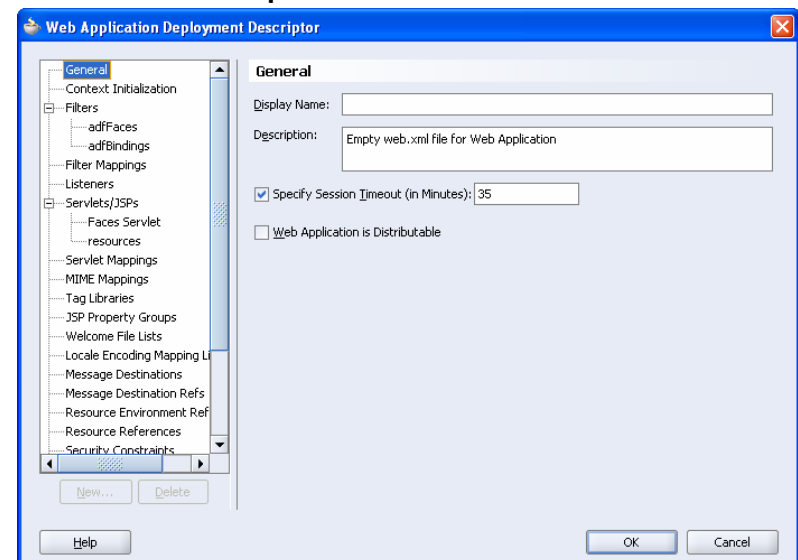
## web.xml Excerpt

```
<?xml version="1.0" encoding="windows-1252" ?>
- <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" version="2.4"
xmlns="http://java.sun.com/xml/ns/j2ee">
<description>Descriptor for the TUHRA application.</description>
- <filter-mapping>
  <filter-name>adfBindings</filter-name>
  <url-pattern>*.jspx</url-pattern>
</filter-mapping>
- <servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
+ <servlet>
- <servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
- <mime-mapping>
  <extension>html</extension>
  <mime-type>text/html</mime-type>
</mime-mapping>
+ <mime-mapping>
- <welcome-file-list>
  <welcome-file>/faces/home.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

Demo

27

## JDeveloper web.xml Editor

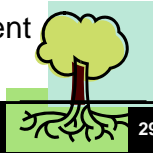


quovera

28

## Context Root

- **Context root** (jspapp) is associated with a program
  - For static HTML file
    - Retrieves file within the physical directory mapped to the context root
  - For J2EE application file (e.g., jsp/forum\_query.jsp)
    - File in the URL is found in the physical directory that maps to the context root/jsp directory
  - J2EE application with no file name
    - E.g., `http://www.oracle.com/jspapp`
    - Find the welcome file and return its content



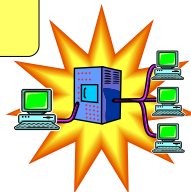
## Information Scope

- Information areas available within the application
- Application
  - Within the web server runtime session, all users access it
- Session
  - One client browser session
- Request
  - One HTTP roundtrip
  - The default for most variables



## Agenda

- HTTP Request/Response
- HTTP Details
- J2EE Specifics
- **Web Roundtrip for J2EE**



## About Domain Name System

- HTTP can only locate a server based on an IP address
  - For example, 141.146.8.66 (`www.oracle.com`)
  - hosts file translates names to IP addresses
    - In `C:\Windows\drivers\system32\etc\`
  - Host name may not be there
- Domain Name System (or Service) (DNS) solves this
  - Host names are registered with with DNS servers
  - They translate name to IP address
- Not part of HTTP but needed for HTTP





# DNS Resolution Example

```

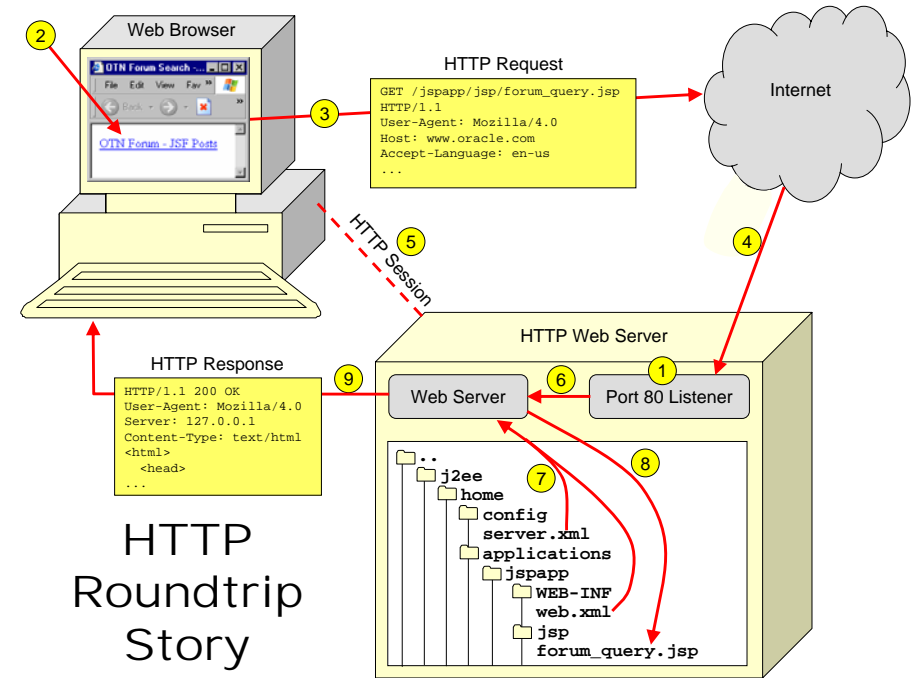
04 Command Prompt
C:\>ping www.oracle.com
Pinging www.oracle.com [141.146.8.66] with 32 bytes of data:
Reply from 141.146.8.66: bytes=32 time=60ms TTL=239
Reply from 141.146.8.66: bytes=32 time=58ms TTL=239
Reply from 141.146.8.66: bytes=32 time=59ms TTL=239
Reply from 141.146.8.66: bytes=32 time=58ms TTL=239

Ping statistics for 141.146.8.66:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 58ms, Maximum = 60ms, Average = 58ms
C:\>
    
```

1. Ping command is sent to the Internet (it is not in the `hosts` file on the local machine)
2. A DNS somewhere finds the domain
  - `www` is not part of the domain
  - DNS returns the IP address
3. Server responds with return message



DNS lookup at <http://www.lookupserver.com>



## HTTP Roundtrip Story

# The HTTP Roundtrip Story

1. HTTP daemon (HTTPD), listens for a request on a specific port (by default, port 80).
2. The user clicks this link on an HTML page:
 

```

http://www.oracle.com/jspapp/forum_query.jsp?
forum_name=jdev&searchFor=JSF%20new%20features&
startDate=Jan-01-2005
            
```

Space character
3. Browser sends HTTP request to the network (the Internet in this example).
4. A DNS server resolves the domain name to an IP address and sends the message to the host.
5. Port 80 listener accepts the request and allows the client to set up a connection session.
6. The listener passes control to the web server.
7. The web server parses the request and routes it to the appropriate servlet.
8. The servlet runs the code associated with the file (if it is a J2EE program such as a servlet or JSP page) or opens the file (if it is an HTML or other type of non-program file).
9. The web server constructs a response and sends it to the browser. The browser renders the content & closes the connection.

# Summary

- HTTP protocol provides communication services for web applications
- HTTP works on top of TCP/IP
- HTTP roundtrip consists of a request and a response
- GET and POST are the two most frequently used methods (along with HEAD)
- URLs and DNS are used by HTTP communications
- The web server parses and processes the request and sends it to the proper program



# Does That Make Sense?

**Boss Paul:** You got your mind right, Luke?

**Luke:** Yah. I got it right. I got it right, Boss.

— *Cool Hand Luke* (1967)



- Please fill out the evals
- Books co-authored with Dr. Paul Dorsey, Avrom Roy-Faderman, & Duncan Mills
- Personal web site:  
[http://ourworld.compuserve.com/homepages/Peter\\_Koletzke](http://ourworld.compuserve.com/homepages/Peter_Koletzke)



<http://www.quovera.com>

- Founded in 1995 as Millennia Vision Corp.
- Profitable for 7+ years without outside funding
- Consultants each have 10+ years industry experience
- Strong High-Tech industry background
- 200+ clients/300+ projects
- JDeveloper Partner
- More technical white papers and presentations on the web site