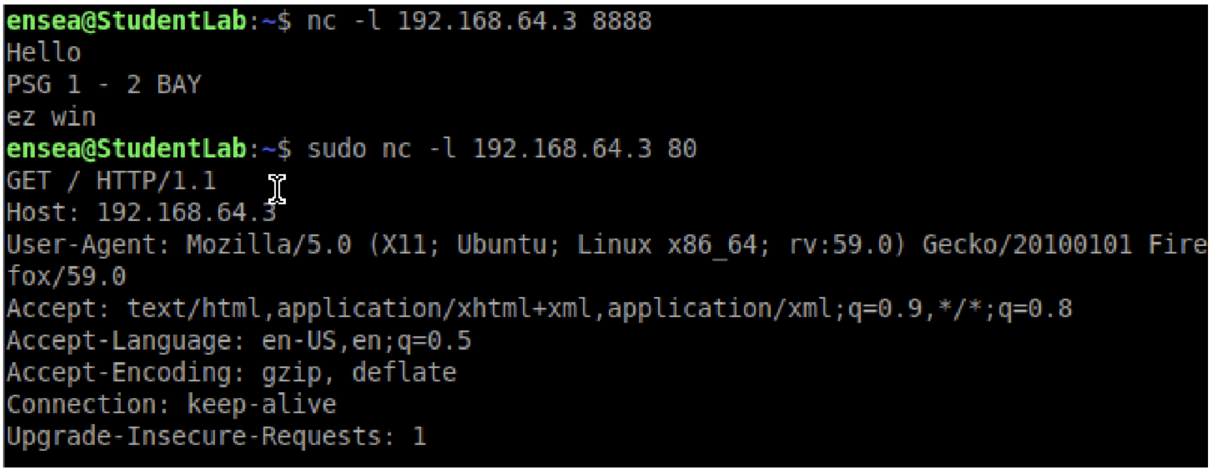
David LE, Mehdi BOULHARTS, 3A RTS
Supervisor: Linda SENIGAGLIESI

TP1 – Java for networks

1 – Client-SeRveR communication with netcat

We used netcat to open a direct, basic connection between two machines. When we typed text on one side, it immediately showed up on the other. This confirmed we established a two-way link (full-duplex) using the TCP protocol. Netcat lets us see raw, direct network communication without any complex protocols on top.

We made netcat listen on the standard HTTP port 80 and connected to it with a web browser. The browser didn't just send plain text; it immediately sent a structured HTTP GET Request. This request included important details like the mandatory Host header and the User-Agent (the browser name) . A web browser always talks in HTTP. It expects a specific HTTP response (like 200 OK) and won't display anything until it gets one. Since netcat didn't send a proper HTTP response, the connection likely just hung or timed out. This shows us how application protocols differ from the simple TCP stream.



```
ensea@StudentLab:~$ nc -l 192.168.64.3 8888
Hello
PSG 1 - 2 BAY
ez win
ensea@StudentLab:~$ sudo nc -l 192.168.64.3 80
GET / HTTP/1.1
Host: 192.168.64.3
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:59.0) Gecko/20100101 Fire
fox/59.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

2 – HTTP Client

2.1 Low-level Approach (Java Sockets)

Our goal was to build a client from scratch using basic Java Sockets, which meant we had to handle the TCP connection and the HTTP protocol manually. The program successfully connected to www.faqs.org:80 and saved the requested file (rfc2068.html) after the server responded with HTTP/1.1 200 OK.
"Low-Level" Meaning: This approach is "low-level" because we had to manually control the connection using Socket and manually send the formatted HTTP request (the GET line and all the headers) using PrintWriter . We were responsible for reading the response data byte-by-byte from the InputStream .

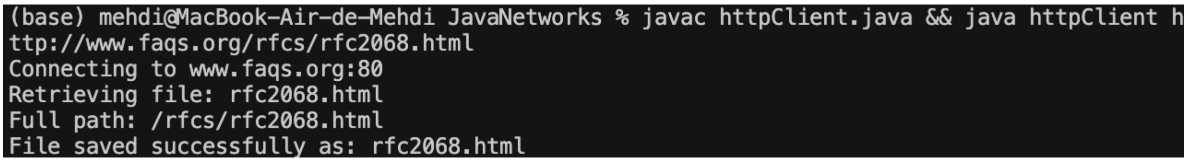
The 200 OK status is crucial. It tells us the request was successful and the resource is in the body, which allowed us to save the content.

Wireshark Analysis (TCP/DNS):

We used Wireshark to observe the network traffic. The capture clearly shows the steps for a complete request. Finally :

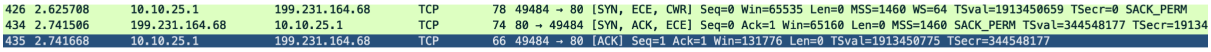
- DNS Request: The client first asks for the IP address of the server (before the TCP connection).
- TCP Connection Opening: The standard three-way handshake occurs: SYN, then SYN, ACK, then ACK.
- TCP Closure: The connection is terminated correctly via the four-way handshake (FIN, ACK from both sides).

<u>java.net* :</u>



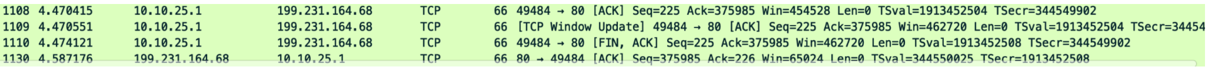
```
(base) mehdi@MacBook-Air-de-Mehdi JavaNetworks % javac httpClient.java && java httpClient h
ttp://www.faqs.org/rfcs/rfc2068.html
Connecting to www.faqs.org:80
Retrieving file: rfc2068.html
Full path: /rfcs/rfc2068.html
File saved successfully as: rfc2068.html
```

Opening



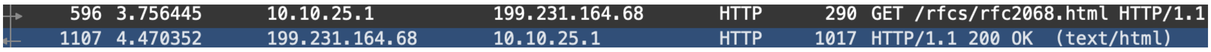
```
426  2.625708    10.10.25.1      199.231.164.68   TCP    78  49484 → 80 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1913450659 TSecr=0 SACK_PERM
434  2.741506    199.231.164.68  10.10.25.1       TCP    74  80 → 49484 [SYN, ACK, ECE] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=344548177 TSecr=19134
435  2.741668    10.10.25.1      199.231.164.68   TCP    66  49484 → 80 [ACK] Seq=1 Ack=1 Win=131776 Len=0 TSval=1913450775 TSecr=344548177
```

Closing



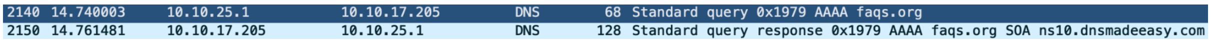
```
1108 4.470415    10.10.25.1      199.231.164.68   TCP    66  49484 → 80 [ACK] Seq=225 Ack=375985 Win=454528 Len=0 TSval=1913452504 TSecr=344549902
1109 4.470551    10.10.25.1      199.231.164.68   TCP    66  [TCP Window Update] 49484 → 80 [ACK] Seq=225 Ack=375985 Win=462720 Len=0 TSval=1913452504 TSecr=34454
1110 4.474121    10.10.25.1      199.231.164.68   TCP    66  49484 → 80 [FIN, ACK] Seq=225 Ack=375985 Win=462720 Len=0 TSval=1913452508 TSecr=344549902
1130 4.587176    199.231.164.68  10.10.25.1       TCP    66  80 → 49484 [ACK] Seq=375985 Ack=226 Win=65024 Len=0 TSval=344550025 TSecr=1913452508
```

HTTP



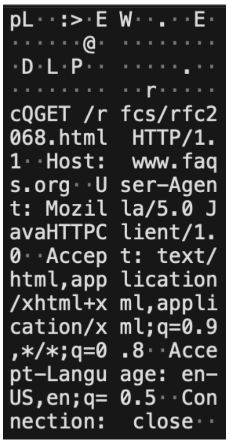
```
 596  3.756445    10.10.25.1       199.231.164.68    HTTP    290  GET /rfcs/rfc2068.html HTTP/1.1
1107  4.470352    199.231.164.68   10.10.25.1        HTTP   1017  HTTP/1.1 200 OK  (text/html)
```

DNS



```
2140  14.740003    10.10.25.1       10.10.17.205    DNS     68  Standard query 0x1979 AAAA faqs.org
2150  14.761481    10.10.17.205     10.10.25.1      DNS    128  Standard query response 0x1979 AAAA faqs.org SOA ns10.dnsmadeeasy.com
```

GET

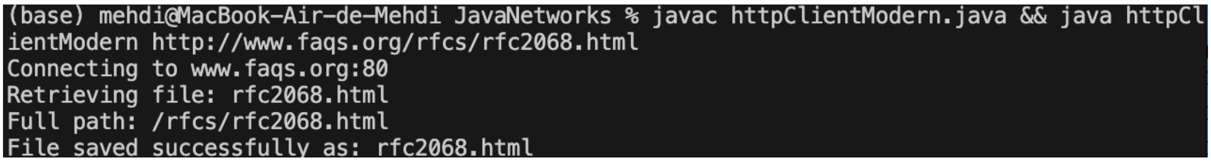


```
pL··:>·E W··.··E·
······@· ········
·D·L·P·· ·····.··
········ ··r·····
cQGET /r fcs/rfc2
068.html  HTTP/1.
1··Host:  www.faq
s.org··U ser-Agen
t: Mozil la/5.0 J
avaHTTPC lient/1.
0··Accep t: text/
html,app lication
/xhtml+x ml,appli
cation/x ml;q=0.9
,*/*;q=0 .8··Acce
pt-Langu age: en-
US,en;q= 0.5··Con
nection:  close··
```

2.2 High-level approach

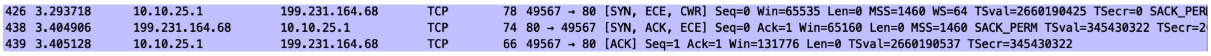
We redid the client using the powerful, built-in Java classes, like those found in java.net. The same file was retrieved successfully, but our code was much simpler. These high-level tools do all the heavy lifting for us. The Java runtime automatically manages the socket, formats the standard headers, handles closing the connection, and even supports modern features like HTTP/2 negotiation. This saves us a lot of manual work compared to using raw sockets.

java.net.HTTP :



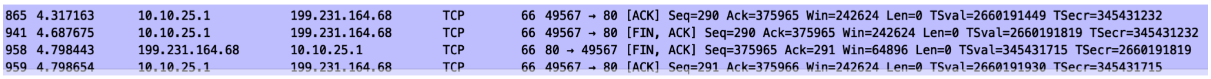
```
(base) mehdi@MacBook-Air-de-Mehdi JavaNetworks % javac httpClientModern.java && java httpCl
ientModern http://www.faqs.org/rfcs/rfc2068.html
Connecting to www.faqs.org:80
Retrieving file: rfc2068.html
Full path: /rfcs/rfc2068.html
File saved successfully as: rfc2068.html
```

Opening



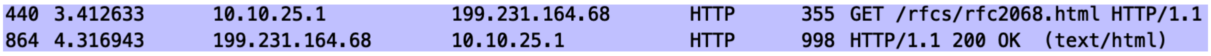
```
426  3.293718     10.10.25.1        199.231.164.68    TCP      78  49567 → 80 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2660190425 TSecr=0 SACK_PER
438  3.404906     199.231.164.68    10.10.25.1        TCP      74  80 → 49567 [SYN, ACK, ECE] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=345430322 TSecr=2
439  3.405128     10.10.25.1        199.231.164.68    TCP      66  49567 → 80 [ACK] Seq=1 Ack=1 Win=131776 Len=0 TSval=2660190537 TSecr=345430322
```

Closing



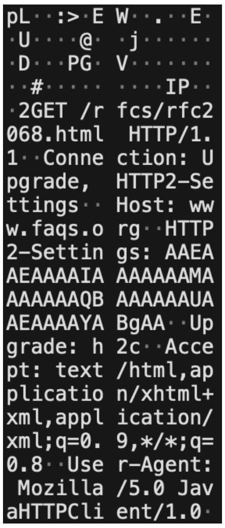
```
865  4.317163     10.10.25.1        199.231.164.68    TCP      66  49567 → 80 [ACK] Seq=290 Ack=375965 Win=242624 Len=0 TSval=2660191449 TSecr=345431232
941  4.687675     10.10.25.1        199.231.164.68    TCP      66  49567 → 80 [FIN, ACK] Seq=290 Ack=375965 Win=242624 Len=0 TSval=2660191819 TSecr=345431232
958  4.798443     199.231.164.68    10.10.25.1        TCP      66  80 → 49567 [FIN, ACK] Seq=375965 Ack=291 Win=64896 Len=0 TSval=345431715 TSecr=2660191819
959  4.798654     10.10.25.1        199.231.164.68    TCP      66  49567 → 80 [ACK] Seq=291 Ack=375966 Win=242624 Len=0 TSval=2660191930 TSecr=345431715
```

HTTP



```
440  3.412633     10.10.25.1        199.231.164.68    HTTP     355  GET /rfcs/rfc2068.html HTTP/1.1
864  4.316943     199.231.164.68    10.10.25.1        HTTP     998  HTTP/1.1 200 OK  (text/html)
```

GET



```
pL··:>·E W··.··E·
·U····@· ·j······
·D···PG· V·······
··#····· ····IP··
·2GET /r fcs/rfc2
068.html  HTTP/1.
1··Conne ction: U
pgrade,  HTTP2-Se
ttings·· Host: ww
w.faqs.o rg··HTTP
2-Settin gs: AAEA
AEAAAAIA AAAAAAMA
AAAAAAQB AAAAAAUA
AEAAAAYA BgAA··Up
grade: h 2c··Acce
pt: text /html,ap
plicatio n/xhtml+
xml,appl ication/
xml;q=0. 9,*/*;q=
0.8··Use r-Agent:
 Mozilla /5.0 Jav
aHTTPCli ent/1.0·
```

3 – HTTP Servers

3.1 Nginx

We installed Nginx using sudo apt install nginx and tested it by navigating to http://localhost . We also checked the main configuration files (like /etc/nginx/nginx.conf) to figure out where Nginx looks for files. Accessing the local address showed us the default Nginx welcome page. The configuration files pointed us to the document root (usually /var/www/html), which is the directory where Nginx fetches the requested files. Nginx is primarily a very fast static web server. Its basic job is to take a request (e.g., GET /image.png) and quickly send the file back from the disk, without running any complex application code. It's excellent for handling many simple connections efficiently.

```
(base) mehdi@MacBook-Air-de-Mehdi JavaNetworks % curl http://localhost:8080/


<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <title>Apache Tomcat/9.0.111</title>
        <link href="favicon.ico" rel="icon" type="image/x-icon" />
        <link href="tomcat.css" rel="stylesheet" type="text/css" />
    </head>

    <body>
        <div id="wrapper">
```

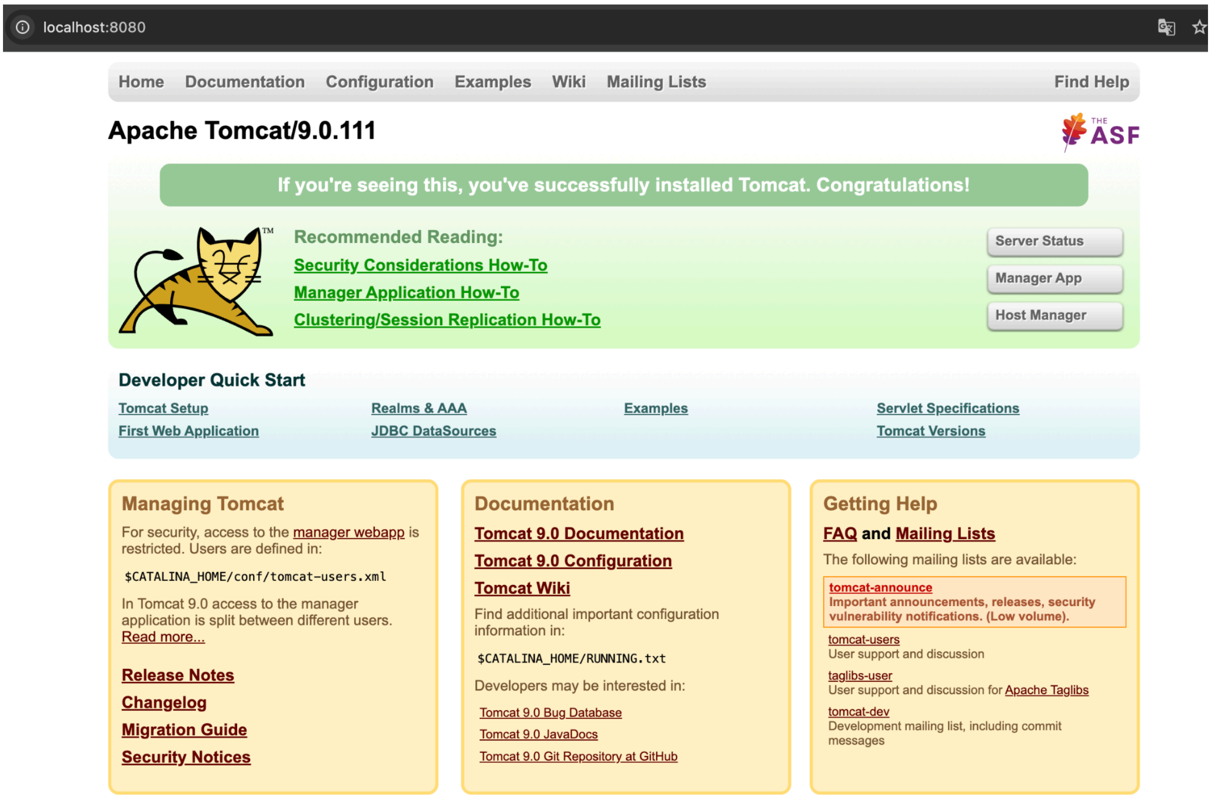
```
      <div id="navigation" class="curved container">
        <span id="nav-home"><a href="https://tomcat.apache.org/">Home</a></span>
        <span id="nav-hosts"><a href="/docs/">Documentation</a></span>
        <span id="nav-config"><a href="/docs/config/">Configuration</a></span>
        <span id="nav-examples"><a href="/examples/">Examples</a></span>
        <span id="nav-wiki"><a href="https://cwiki.apache.org/confluence/display/TOMCAT/">Wiki</a></span>
        <span id="nav-lists"><a href="https://tomcat.apache.org/lists.html">Mailing Lists</a></span>
        <span id="nav-help"><a href="https://tomcat.apache.org/findhelp.html">Find Help</a></span>
        <br class="separator" />
      </div>
      <div id="asf-box">
        <h1>Apache Tomcat/9.0.111</h1>
      </div>
      <div id="upper" class="curved container">
        <div id="congrats" class="curved container">
          <h2>If you're seeing this, you've successfully installed Tomcat. Congratulations!</h2>
        </div>
        <div id="notice">
          <img id="tomcat-logo" src="tomcat.svg" alt="[tomcat logo]" />
          <div id="tasks">
            <h3>Recommended Reading:</h3>
            <h4><a href="/docs/security-howto.html">Security Considerations How-To</a></h4>
            <h4><a href="/docs/manager-howto.html">Manager Application How-To</a></h4>
            <h4><a href="/docs/cluster-howto.html">Clustering/Session Replication How-To</a></h4>
          </div>
        </div>
        <div id="actions">
          <div class="button">
            <a class="container shadow" href="/manager/status"><span>Server Status</span></a>
          </div>
          <div class="button">
            <a class="container shadow" href="/manager/html"><span>Manager App</span></a>
          </div>
          <div class="button">
            <a class="container shadow" href="/host-manager/html"><span>Host Manager</span></a>
          </div>
        </div>
        <br class="separator" />
      </div>
      <div id="middle" class="curved container">
        <h3>Developer Quick Start</h3>
        <div class="col25">
          <div class="container">
            <p><a href="/docs/setup.html">Tomcat Setup</a></p>
            <p><a href="/docs/appdev/">First Web Application</a></p>
          </div>
        </div>
        <div class="col25">
          <div class="container">
            <p><a href="/docs/realm-howto.html">Realms &amp; AAA</a></p>
            <p><a href="/docs/jndi-datasource-examples-howto.html">JDBC
DataSources</a></p>
          </div>
        </div>
        <div class="col25">
          <div class="container">
            <p><a href="/examples/">Examples</a></p>
          </div>
        </div>
        <div class="col25">
          <div class="container">
            <p><a
href="https://cwiki.apache.org/confluence/display/TOMCAT/Specifications">Servlet
Specifications</a></p>
            <p><a
href="https://cwiki.apache.org/confluence/display/TOMCAT/Tomcat+Versions">Tomcat
Versions</a></p>
          </div>
        </div>
        <br class="separator" />
      </div>
      <div id="lower">
        <div id="low-manage" class="">
          <div class="curved container">
            <h3>Managing Tomcat</h3>
            <p>For security, access to the <a href="/manager/html">manager
webapp</a> is restricted.
            Users are defined in:</p>
            <pre>$CATALINA_HOME/conf/tomcat-users.xml</pre>
            <p>In Tomcat 9.0 access to the manager application is split between
              different users. &nbsp; <a href="/docs/manager-howto.html">Read
more...</a></p>
            <br />
            <h4><a href="/docs/RELEASE-NOTES.txt">Release Notes</a></h4>
            <h4><a href="/docs/changelog.html">Changelog</a></h4>
            <h4><a href="https://tomcat.apache.org/migration.html">Migration Guide</a></h4>
            <h4><a href="https://tomcat.apache.org/security.html">Security Notices</a></h4>
          </div>
        </div>
        <div id="low-docs" class="">
          <div class="curved container">
            <h3>Documentation</h3>
            <h4><a href="/docs/">Tomcat 9.0 Documentation</a></h4>
            <h4><a href="/docs/config/">Tomcat 9.0 Configuration</a></h4>
            <h4><a href="https://cwiki.apache.org/confluence/display/TOMCAT/">Tomcat Wiki</a></h4>
            <p>Find additional important configuration information in:</p>
            <pre>$CATALINA_HOME/RUNNING.txt</pre>
            <p>Developers may be interested in:</p>
            <ul>
              <li><a href="https://tomcat.apache.org/bugreport.html">Tomcat 9.0 Bug Database</a></li>
              <li><a href="/docs/api/index.html">Tomcat 9.0 JavaDocs</a></li>
              <li><a href="https://github.com/apache/tomcat/tree/9.0.x">Tomcat 9.0 Git Repository at GitHub</a></li>
            </ul>
          </div>
        </div>
        <div id="low-help" class="">
          <div class="curved container">
            <h3>Getting Help</h3>
            <h4><a href="https://tomcat.apache.org/faq/">FAQ</a> and <a href="https://tomcat.apache.org/lists.html">Mailing Lists</a></h4>
            <p>The following mailing lists are available:</p>
            <ul>
              <li id="list-announce"><strong><a href="https://tomcat.apache.org/lists.html#tomcat-announce">tomcat-announce</a><br />
                Important announcements, releases, security vulnerability notifications. (Low volume).</strong>
              </li>
              <li><a href="https://tomcat.apache.org/lists.html#tomcat-users">tomcat-users</a><br />
                User support and discussion
              </li>
              <li><a href="https://tomcat.apache.org/lists.html#taglibs-user">taglibs-user</a><br />
                User support and discussion for <a href="https://tomcat.apache.org/taglibs/">Apache Taglibs</a>
              </li>
              <li><a
href="https://tomcat.apache.org/lists.html#tomcat-dev">tomcat-dev</a><br />
                Development mailing list, including commit messages
              </li>
            </ul>
          </div>
        </div>
        <br class="separator" />
      </div>
      <div id="footer" class="curved container">
        <div class="col20">
          <div class="container">
            <h4>Other Downloads</h4>
            <ul>
              <li><a href="https://tomcat.apache.org/download-connectors.cgi">Tomcat
Connectors</a></li>
              <li><a href="https://tomcat.apache.org/download-native.cgi">Tomcat
Native</a></li>
              <li><a href="https://tomcat.apache.org/taglibs/">Taglibs</a></li>
              <li><a href="/docs/deployer-howto.html">Deployer</a></li>
            </ul>
          </div>
        </div>
        <div class="col20">
          <div class="container">
            <h4>Other Documentation</h4>
            <ul>
              <li><a href="https://tomcat.apache.org/connectors-doc/">Tomcat
Connectors</a></li>
              <li><a href="https://tomcat.apache.org/connectors-doc/">mod_jk
Documentation</a></li>
              <li><a href="https://tomcat.apache.org/native-doc/">Tomcat
Native</a></li>
              <li><a href="/docs/deployer-howto.html">Deployer</a></li>
            </ul>
          </div>
        </div>
        <div class="col20">
          <div class="container">
            <h4>Get Involved</h4>
            <ul>
              <li><a
href="https://tomcat.apache.org/getinvolved.html">Overview</a></li>
              <li><a href="https://tomcat.apache.org/source.html">Source
Repositories</a></li>
              <li><a href="https://tomcat.apache.org/lists.html">Mailing Lists</a></li>
              <li><a
href="https://cwiki.apache.org/confluence/display/TOMCAT/">Wiki</a></li>
            </ul>
          </div>
        </div>
        <div class="col20">
          <div class="container">
            <h4>Miscellaneous</h4>
            <ul>
              <li><a href="https://tomcat.apache.org/contact.html">Contact</a></li>
              <li><a href="https://tomcat.apache.org/legal.html">Legal</a></li>
              <li><a
href="https://www.apache.org/foundation/sponsorship.html">Sponsorship</a></li>
              <li><a
href="https://www.apache.org/foundation/thanks.html">Thanks</a></li>
            </ul>
          </div>
        </div>
        <div class="col20">
          <div class="container">
            <h4>Apache Software Foundation</h4>
            <ul>
              <li><a href="https://tomcat.apache.org/whoweare.html">Who We
Are</a></li>
              <li><a href="https://tomcat.apache.org/heritage.html">Heritage</a></li>
              <li><a href="https://www.apache.org">Apache Home</a></li>
              <li><a
href="https://tomcat.apache.org/resources.html">Resources</a></li>
            </ul>
          </div>
        </div>
        <br class="separator" />
      </div>
      <p class="copyright">Copyright &copy;1999-2025 Apache Software Foundation.  All
Rights Reserved</p>
    </div>
  </body>

</html>
```

3.2 Tomcat

We installed Tomcat 9 and confirmed its default address, typically on port 8080 . We then deployed a sample application and, crucially, edited a JSP file (JavaServer Page) to insert our own dynamic Java code. When we accessed the modified JSP file via the browser, we didn't see the raw Java source code. Instead, we saw the result of the Java code execution (like the server's current date, time, or server information). Tomcat is not a simple web server; it's a Servlet Container/JSP Engine. Its main job is to execute complex Java code on the server-side to generate dynamic HTML content. This makes it an application server, which handles logic and heavy computing tasks, typically running in the background on port 8080.

```
(base) mehdi@MacBook-Air-de-Mehdi JavaNetworks % curl http://localhost:8080/sample/
<html>
<head>
<title>Sample "Hello, World" Application</title>
</head>
<body bgcolor=white>

<table border="0">
<tr>
<td>
<img src="images/tomcat.gif">
</td>
<td>
<h1>Sample "Hello, World" Application</h1>
<p>This is the home page for a sample application used to illustrate the
source directory organization of a web application utilizing the principles
outlined in the Application Developer's Guide.
</td>
</tr>
</table>

<p>To prove that they work, you can execute either of the following links:
<ul>
<li>To a <a href="hello.jsp">JSP page</a>.
<li>To a <a href="hello">servlet</a>.
</ul>

</body>
</html>
```

```
(base) mehdi@MacBook-Air-de-Mehdi JavaNetworks % curl
http://localhost:8080/sample/test2.jsp

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Hello JSP - Modified</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 40px; }
        .info { background-color: #f0f0f0; padding: 20px; border-radius: 5px; }
        .time { color: #2196F3; }
    </style>
</head>
<body>
    <h1>Hello JSP - Version modifiée</h1>

    <div class="info">

        <h2>Date et heure :</h2>
        <p class="time">
            05/11/2025 23:47:55

        </p>


        <h2>Compteur avec boucle Java :</h2>
        <ul>
            <li>Ligne numéro 1</li>
<li>Ligne numéro 2</li>
<li>Ligne numéro 3</li>
<li>Ligne numéro 4</li>
<li>Ligne numéro 5</li>

        </ul>


        <h2>Informations serveur :</h2>
        <p>
            Nom du serveur : localhost<br>
            Port : 8080<br>
            Protocol : HTTP/1.1
        </p>
    </div>

    <p><a href="/">Retour à la page d'accueil</a></p>
</body>
</html>%
```



Apache Tomcat/9.0.111

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

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation

Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

Tomcat 9.0 Bug Database
Tomcat 9.0 JavaDocs
Tomcat 9.0 Git Repository at GitHub

Getting Help

FAQ and Mailing Lists

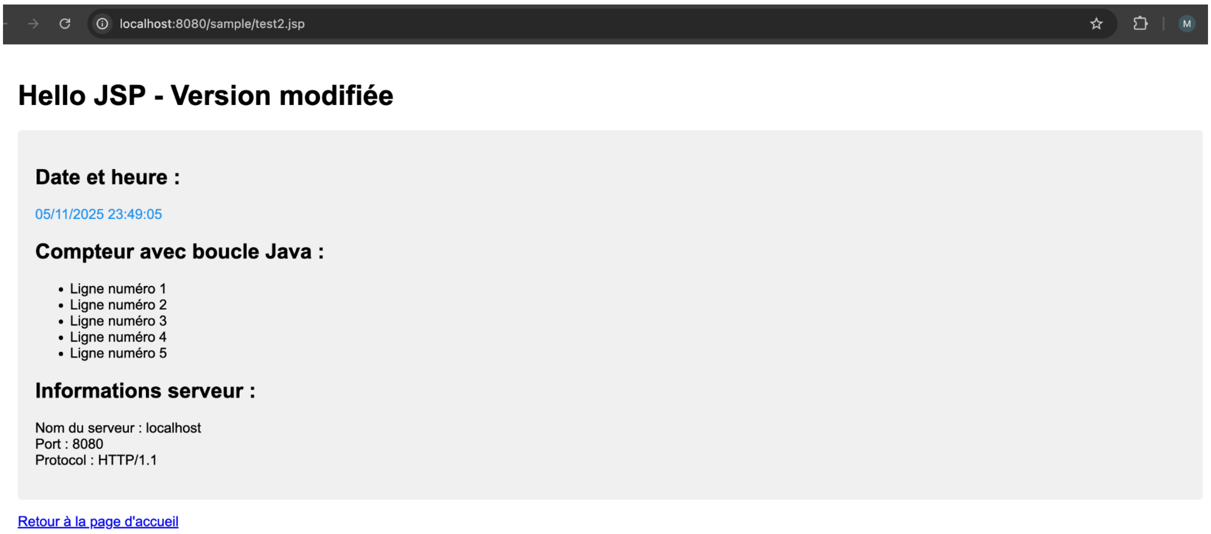
The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users
User support and discussion

taglibs-user
User support and discussion for Apache Taglibs

tomcat-dev
Development mailing list, including commit messages



Hello JSP - Version modifiée

Date et heure :

05/11/2025 23:49:05

Compteur avec boucle Java :

- Ligne numéro 1
- Ligne numéro 2
- Ligne numéro 3
- Ligne numéro 4
- Ligne numéro 5

Informations serveur :

Nom du serveur : localhost
Port : 8080
Protocol : HTTP/1.1

Retour à la page d'accueil

3.3 Roles of Nginx and Tomcat

What is Nginx?

Nginx is an extremely fast web server and reverse proxy.
It sits at the front of the architecture (the frontend). It's excellent for serving static files(images, CSS, HTML) quickly and managing high traffic. It can also manage the secure connection (SSL/TLS).

What is Tomcat?

Tomcat is a Servlet Container and an application server.
It sits at the back (the backend) and is designed to run dynamic Java applications. We confirmed this when we edited the JSP file and saw the Java code run on the server. It handles the logic that requires Java code, databases, and user sessions.

The Link Between Nginx and Tomcat :

The standard practice is to use Nginx as a Reverse Proxy pointing to Tomcat. This setup lets each server do what it's best at: Nginx handles the high-speed, simple traffic (static content, initial connection), and it passes the complex, dynamic requests to Tomcat. This makes the whole system faster and more reliable.