**Task:**

Build a web server with Velocity html templates to present a web page with Geo Map using the Google Maps API, showing the user their location with a marker and logs the user’s latitude and longitude on the server.

**Related specification**

JDK 1.8.0\_102

Tomcat 8.0.42

Velocity 1.7

Eclipse Neon.3 Release (4.6.3)

Window 7, 64 bit

**Architect of the application**

This application is implemented using Java servlet + Velocity + JavaScript.

When a user visits the application by URL (e.g. https://192.168.1.103:8443/map), the Java Servlet (src/demo/MyVelocity01.java) responds in doGet function. In doGet function, a template (src/templates/test.vm) was used by VelocityEngine to create a webpage for the user.

When a user clicks button My Position in the webpage, Google Map API was used to show Google Map and user location. This function is realized by JavaScript in the template (src/templates/test.vm). I use my Google API Key in the code, you could replace it with yours.

When a user clicks button Sent Position in the webpage, a POST HTTP request is sent to the Java servlet (src/demo/MyVelocity01.java). The doPost function responds this request and saves the user’s latitude and longitude to log file.

**Instruction of running the application**

You can find the source code in directory: ServletDemo. This is an Eclipse project which can be imported into a new workshop using Import/General/ExistingProjecsIntoWorkspace. Needed .jars (velocity-1.7.jar and velocity-1.7-dep.jar) are already included in the directory webapp/WEB-INF/lib. However, you need to include them into the project by adding them into Project/Properties/JavaBuildPath/Librarys.

I modified the conf/server.xml of Tomcat to run the application. I added the following sentence (you need to change docBase according to the path in your computer):

<Context path="" docBase="E:\LiuBin\Java\ServletDemo\_0101\ServletDemo\webapp" reloadable="true"/>

after the sentence:

<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" prefix="localhost\_access\_log" suffix=".txt" pattern="%h %l %u %t &quot;%r&quot; %s %b" />

After starting Tomcat, the application can be visit by URL: http://localhost:8080/map.

Click the button My Position. Internet browser will ask you whether you allow the application to visit your location. Chose Allow. You can see the Google Map in the internet browser with a mark, which represents your position.

If you click button Send Position, your latitude and longitude will be sent to server. At the same time, a record, which includes your position and current time, will be written down to a log file. The log file provisionally located at the bin directory of Tomcat and names UserLatLon.log.

What’s more, the Zoom In, Zoom Out and Street View buttons of Google Map are at the bottom-right corner and still available.

**Instruction of running the application in other devices**

1. Visiting the application by IP rather than by “localhost”?

Add address="0.0.0.0" to the following sentence in conf/server.xml of Tomcat.

<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />

After modifying, the sentence becomes:

<Connector port="8080" address="0.0.0.0" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />

Now the application can be visited using, say, http://192.168.1.103:8080/map.[[1]](#footnote-1)

2. Visiting the application from other devices?

Because Google stop to support GeoLocation API under HTTP protocol, Tomcat need to be configured to support HTTPS protocol.[[2]](#footnote-2)

First, you need to create a certificate for Tomcat using command:

keytool -genkeypair -alias "tomcat" -keyalg "RSA" -keystore "D:\tomcat.keystore"

D:\tomcat.keystore is the location of certificate. Modify conf/server.xml of Tomcat again.

After the sentence:

<!--

<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true" maxThreads="150" scheme="https" secure="true" clientAuth="false" sslProtocol="TLS"/>

-->

add the following sentence:

<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol" maxThreads="150" SSLEnabled="true" scheme="https" secure="true" clientAuth="false" sslProtocol="TLS" keystoreFile="D:\tomcat.keystore" keystorePass="tomcat"/>

The value of keystoreFile is the location of your certificate file. The value of keystorePass is the password of the Tomcat administrator.

Now, you can visit the application from other devices using the URL: https:// 192.168.1.103:8443/map. However, if the certificate is not bought from an authentic organization, internet browser (e.g. Chrome) may warn that “Your connection is not private”. Please choose advance/proceed to 192.168.1.103 (unsafe) to continue.[[3]](#footnote-3)

On some devices, e.g. iPhone, the location access of the internet browser, e.g. Safari, need to be allowed. Otherwise, the internet browser will report “User denied the request for Geolocation”. The configuration of location access is usually in the setting of the devices.

1. cf. <http://blog.sina.com.cn/s/blog_894ed51901011bhf.html>. Some reference are in Chinese. Sorry for the inconvenient. [↑](#footnote-ref-1)
2. cf. <https://developers.google.com/web/updates/2016/04/geolocation-on-secure-contexts-only#why_are_we_making_this_change> [↑](#footnote-ref-2)
3. cf. <http://www.infocool.net/kb/Tomcat/201703/305194.html> [↑](#footnote-ref-3)