MAVEN

Maven Source Installation:

**Step 1:** 1.JAVA -VERSION

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> -> download JDK.RPM (159MB)

save in -> cd /opt/software/java

**Step 2:** Make directory and download tar file

🡪 /opt/software/maven

GOTO CHROME DOWNLOAD http://www.eu.apache.org/ist/maven -3/3.59/binaries

(here download apache -maven3.3.9.tar.gz into your windows)

now send apache -maven3.3.9.tar.gz through winscp(/opt/software/maven)

tar -xzvf apache -maven3.3.9.tar.gzep 2

**Step 3:** Give path for local repository

🡪su maven

🡪 vi ~/.bashrc

export M2\_HOME=/opt/software/maven/apache-maven-3.3.9

export M2=$M2\_HOME/bin

export PATH=$M2:$PATH

🡪source .bashrc

**Step 4:** check version, make directory and give permission

Check version 🡪mvn -version

Giving permission🡪su root

🡪goto cd /opt

🡪mkdir mavenproject

🡪chown -R maven:maven mavenproject

**Step 5:**

mvn archetype:generate (before doing this add proxy)

cd /etc/yum.conf

add line -> proxy=http://web-proxy.sgp.hpecorp.net:8080

and also in [ vi ~/.bashrc ]

**Step 6:**  Sample helloworld in java (Automatically)

**-**> su maven

-> cd /opt/mavenproject

-> mvn archetype:generate -DgroupId=com.hpe.test -DartifactId=helloworld -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

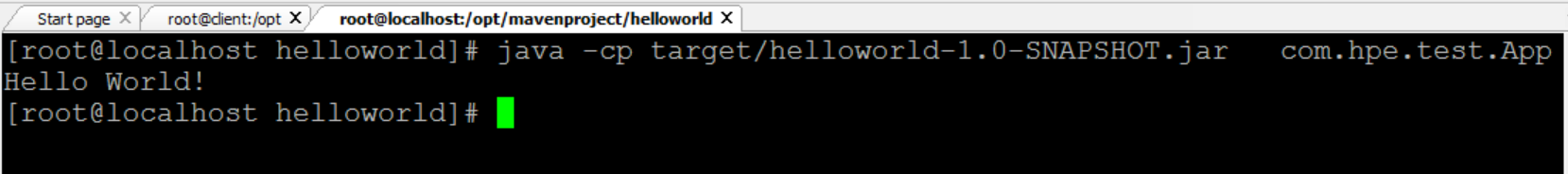
File created -> Helloworld

1. pom.xml 2. src

-> mvn package (created .jar in target)

-> java -cp target/test-1.0-SNAPSHOT.jar com.hpe.test.App

(Hello World!)

****

**Java – Jar Package**

**Flowchart of Java**

SAMPLE

HELLO

Build Package (mvn package)

Pom.xml src target

main . jar

java

Hello.java

**Maven with eclipse**

Create new package -> demo

greeting.java

**package** demo;

**public** **class** greeting {

**public** String greet(){

**return** "hello";

}

}

localtime.java

**package** demo;

**import** org.joda.time.LocalTime;

**public** **class** helloworld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stu

LocalTime currentTime = **new** LocalTime();

System.***out***.println("The current local time is:" + currentTime);

greeting gre = **new** greeting();

System.***out***.println (gre.greet());

}

}

**Add dependency jodatime**

<dependencies>

<dependency>

<groupId>joda-time</groupId>

<artifactId>joda-time</artifactId>

<version>2.2</version>

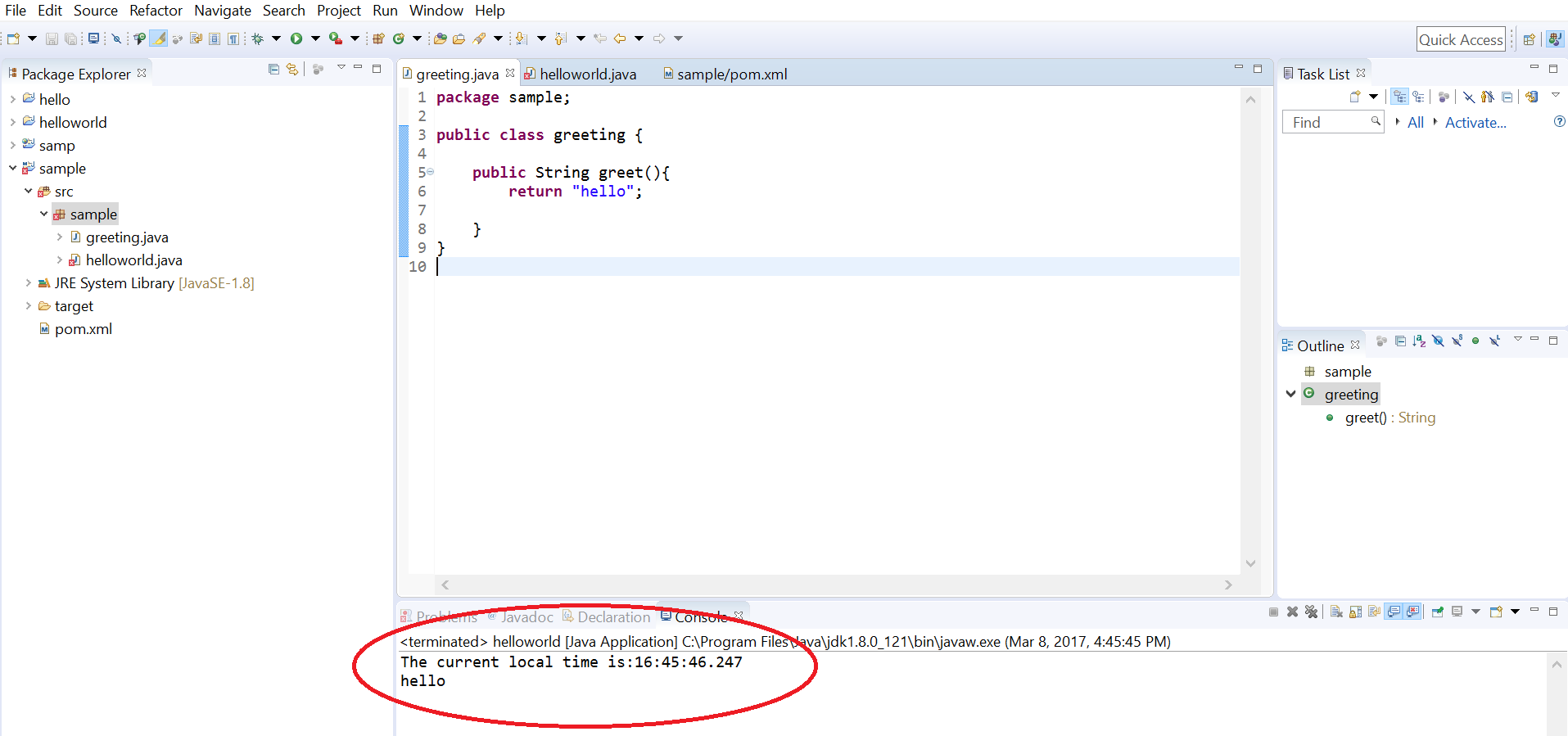
</dependency>

</dependencies>

Project🡪configure🡪convert to maven

Window🡪preference🡪java🡪installed jre🡪add jdk and click checkbox🡪Apply🡪OK

Run🡪maven install



**Create manually package in linux –** follow the flowchart

🡪cd /opt/mavenproject 🡪mkdir demo

🡪cd demo

🡪vi pom.xml

---------------------------------------------------

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.hpe.project-group</groupId>

<artifactId>Helloworld</artifactId>

<version>1.0</version>

<packaging>jar</packaging>

<dependencies>

<dependency>

<groupId>joda-time</groupId>

<artifactId>joda-time</artifactId>

<version>2.2</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<artifactId>maven-assembly-plugin</artifactId>

<configuration>

<archive>

<manifest>

<mainClass>org.joda.time.LocalTime</mainClass>

</manifest>

</archive>

<descriptorRefs>

<descriptorRef>jar-with-dependencies</descriptorRef>

</descriptorRefs>

</configuration>

</plugin>

</plugins>

</build>

</project>

--------------------------------------------------------------------

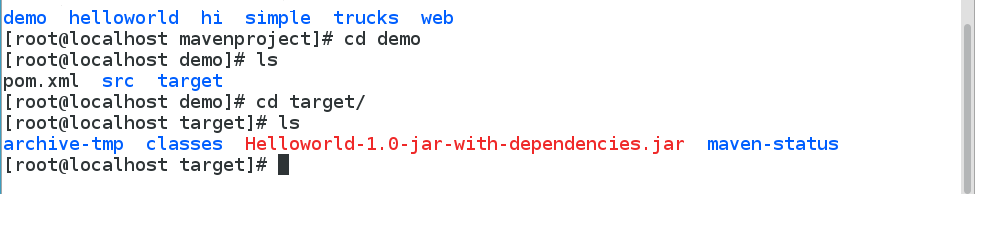
🡪cd /src

Add greeting.java and localtime.java

**Build package**

**🡪**mvn package

**Target** directory 🡪 (.jar) dependencies



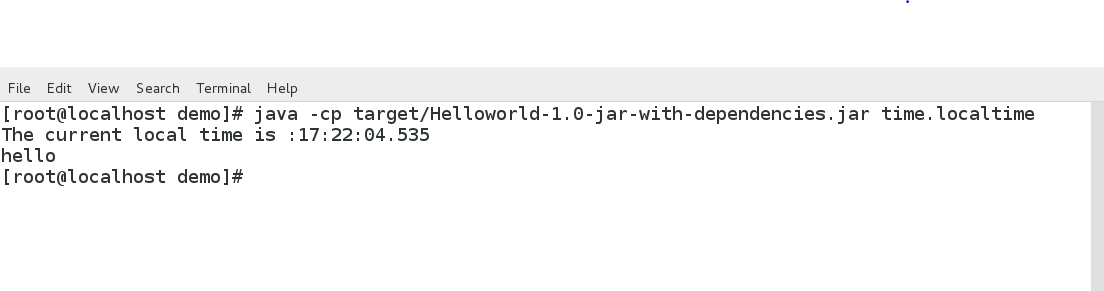
🡪Download the dependencies and shows build success

🡪Created classes and jar file

**Compiling process**

🡪#mvn clean compile assembly:single

🡪java -cp target/Helloworld-1.0-jar-with-dependencies.jar time.localtime (if any issue add custom library)

****

How to include custom library into maven local repository?

------------------------------------------------------------

#mvn install:install-file -Dfile=Helloworld-1.0-jar-with-dependencies.jar -DgroupId=com.google.code

-DartifactId=helloworld -Dversion=1 -Dpackaging=jar

**WebApp – War Package**

**Install Tomcat (**port num – 8080)

**🡪**yum install tomcat

**🡪** yum install tomcat-admin-webapps.noarch tomcat-docs-webapp.noarch tomcat-javadoc.noarch tomcat-systemv.noarch tomcat-webapps.noarch

**Tomcat Authentication**

Create tomcat-users.xml in 🡪 /etc/tomcat/tomcat-users.xml

Add an user with roles manager-gui and manager-script.

🡪vi tomcat-users.xml

-------------------------------------------------------------------------------------

<?xml version='1.0' encoding='utf-8'?>

<tomcat-users>

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<user username="admin" password="password" roles="manager-gui,manager-script" />

</tomcat-users>

**Maven Authentication**

Create settings.xml in .m2 (local repo)

🡪vi settings.xml

<?xml version="1.0" encoding="UTF-8"?>

<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">

<servers>

<server>

<id>mytomcat</id>

<username>admin</username>

<password>admin</password>

</server>

</servers>

</settings>

**To change port number of Tomcat 🡪** /etc/tomcat/server.xml

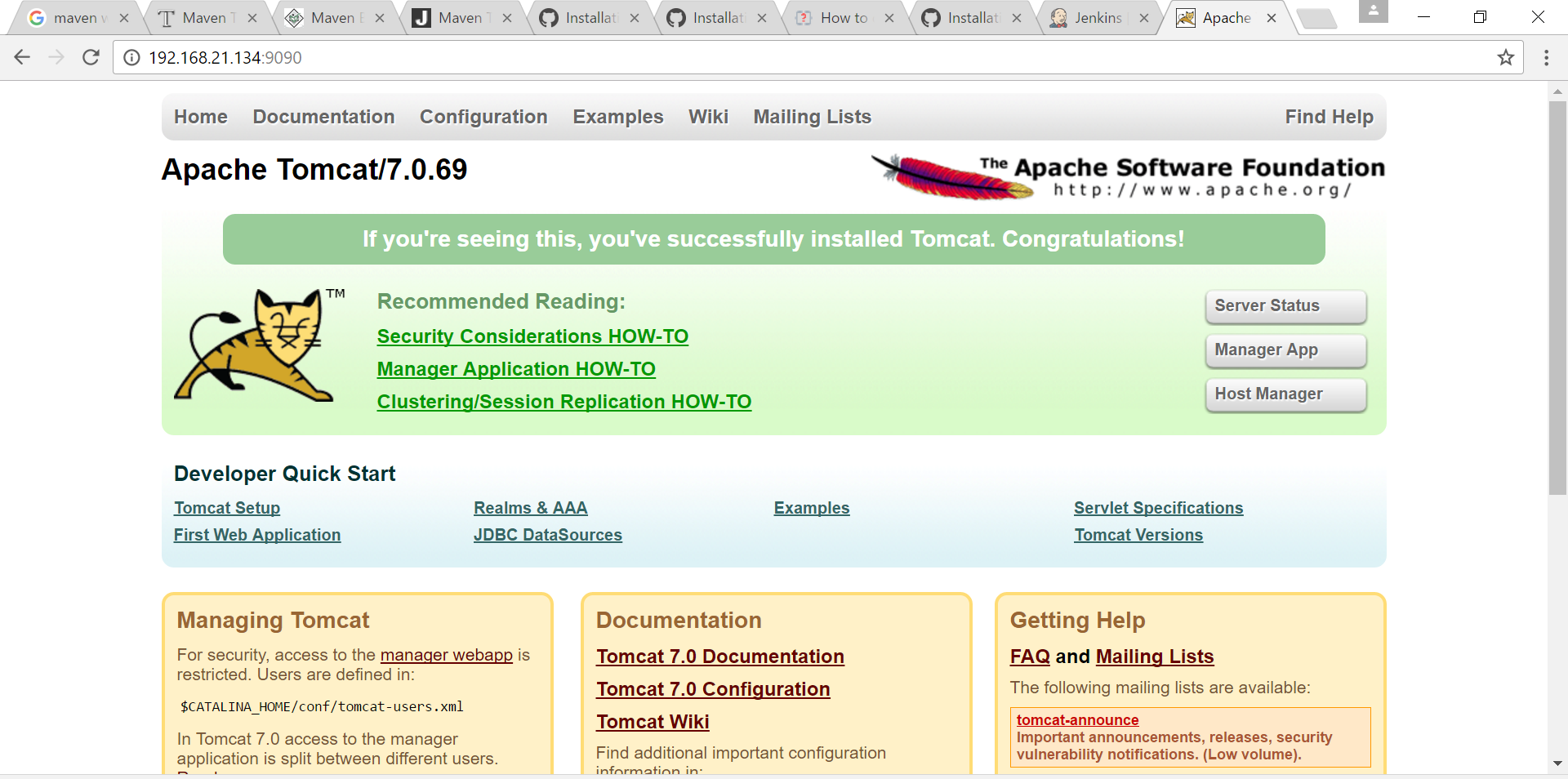
Change 🡪 <connector port: 8080>

Start Service

🡪service tomcat start

🡪service tomcat status

Check Browser 🡪 <http://ipaddress:8080>



**Flowchart of WebApp**

SAMPLE

HELLO

Build package

pom.xmlsrc target

main .war

resources webapp

index.jsp WEB-INF

web.xml

🡪 mvn archetype:generate -DgroupId=com.companyname.automobile -DartifactId=HelloWeb -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false

🡪cd HelloWeb

🡪pom.xml 🡪src /main/webapp/index.jsp

🡪mvn tomcat7:deploy

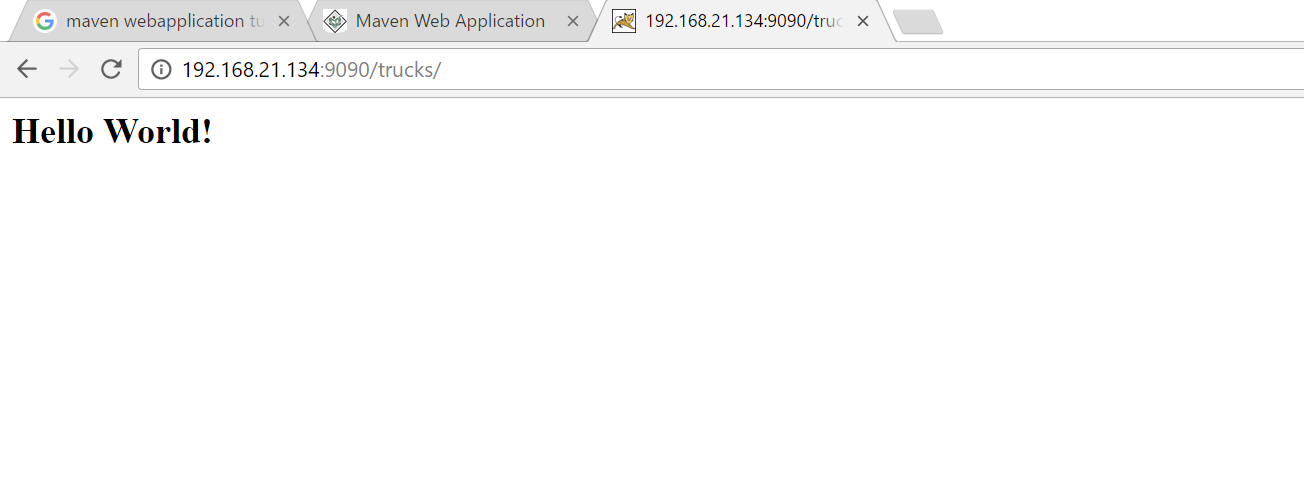
**(**If build failure – do these steps**)**

🡪mvn package

Target🡪.war

Copy **(**.war **)🡪** /var/lib/tomcat/webapps

Check browser: <http://ipaddress/helloweb>



Cache project in 🡪www. github.com/shfq100/cache/

🡪pom.xml 🡪src

Pom.xml

---------------------------------------------------------------

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.hpe.cache</groupId>

<artifactId>cacheImplement4</artifactId>

<version>6.5-SNAPSHOT</version>

<packaging>war</packaging>

<name>cacheImplement</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

<!--<dependency>

<groupId>com.hpe.cache</groupId>

<artifactId>cacheImplement</artifactId>

<version>1.0-SNAPSHOT</version>

<scope>test</scope>

</dependency> -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.4</version>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<source>1.7</source>

<target>1.7</target>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<version>2.2</version>

<configuration>

<url>http://localhost:8080/manager/text</url>

<server>mytomcat</server>

<path>/cacheImplement3</path>

</configuration>

</plugin>

</plugins>

</build>

</project>