MICROSERVICES WITH API GATEWAY

Superset ID : 6393540

Name : Antony Praveen E

E-mail : antonypraveen.2205009@srec.ac.in

Mandatory Question

1) Exercise 1: Creating Microservices for account and loan

Solution:

Account Microservices

//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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <parent>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-parent</artifactId>
                <version>3.5.3</version>
                <relativePath/> <!-- lookup parent from repository -->
        </parent>
        <groupId>com.cognizant</groupId>
        <artifactId>account</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <name>account</name>
        <description>Demo project for Spring Boot</description>
        <url/>
        <licenses>
                <license/>
        </licenses>
        <developers>
```

```
        <developer/>
    </developers>
    <scm>
        <connection/>
        <developerConnection/>
        <tag/>
        <url/>
    </scm>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <build>
```

```
            <plugins>
                <plugin>
                    <groupId>org.springframework.boot</groupId>
                    <artifactId>spring-boot-maven-plugin</artifactId>
                </plugin>
            </plugins>
        </build>
</project>
```

//mvnw.cmd

```
<# : batch portion
@REM ----------------------------------------------------------------------------
@REM Licensed to the Apache Software Foundation (ASF) under one
@REM or more contributor license agreements.  See the NOTICE file
@REM distributed with this work for additional information
@REM regarding copyright ownership.  The ASF licenses this file
@REM to you under the Apache License, Version 2.0 (the
@REM "License"); you may not use this file except in compliance
@REM with the License.  You may obtain a copy of the License at
@REM
@REM    http://www.apache.org/licenses/LICENSE-2.0
@REM
@REM Unless required by applicable law or agreed to in writing,
@REM software distributed under the License is distributed on an
@REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
@REM KIND, either express or implied.  See the License for the
@REM specific language governing permissions and limitations
@REM under the License.
@REM ----------------------------------------------------------------------------
@REM ----------------------------------------------------------------------------
@REM Apache Maven Wrapper startup batch script, version 3.3.2
```

```
@REM
@REM Optional ENV vars
@REM   MVNW_REPOURL - repo url base for downloading maven distribution
@REM  MVNW_USERNAME/MVNW_PASSWORD - user and password for downloading maven
@REM  MVNW_VERBOSE - true: enable verbose log; others: silence the output
@REM ----------------------------------------------------------------------
@IF "%_MVNW_ARG0_NAME%"=="" (SET __MVNW_ARG0_NAME__=%~nx0)
@SET _MVNW_CMD_=
@SET _MVNW_ERROR_=
@SET __MVNW_PSMODULEP_SAVE=%PSModulePath%
@SET PSModulePath=
@FOR /F "usebackq tokens=1* delims==" %%A IN (powershell -noprofile "& {$scriptDir='%~dp0'; $script='%__MVNW_ARG0_NAME__%'; icm -ScriptBlock ([Scriptblock]::Create((Get-Content -Raw '%~f0'))) -NoNewScope}") DO @(
  IF "%%A"=="MVN_CMD" (set _MVNW_CMD_=%%B) ELSE IF "%%B"=="" (echo %%A) ELSE (echo %%A=%%B)
)
@SET PSModulePath=%__MVNW_PSMODULEP_SAVE%
@SET __MVNW_PSMODULEP_SAVE=
@SET __MVNW_ARG0_NAME__=
@SET MVNW_USERNAME=
@SET MVNW_PASSWORD=
@IF NOT "%_MVNW_CMD%"=="" (%MVNW_CMD_% %*)
@echo Cannot start maven from wrapper >&2 && exit /b 1
@GOTO :EOF
: end batch / begin powershell #>
$ErrorActionPreference = "Stop"
if ($env:MVNW_VERBOSE -eq "true") {
  $VerbosePreference = "Continue"
}
# calculate distributionUrl, requires .mvn/wrapper/maven-wrapper.properties
```

```
$distributionUrl = (Get-Content -Raw "$scriptDir/.mvn/wrapper/maven-wrapper.properties" | ConvertFrom-StringData).distributionUrl

if (!$distributionUrl) {

  Write-Error "cannot read distributionUrl property in $scriptDir/.mvn/wrapper/maven-wrapper.properties"

}

switch -wildcard -casesensitive ( $($distributionUrl -replace '^.*/','') ) {

  "maven-mvnd-*" {

    $USE_MVND = $true

    $distributionUrl = $distributionUrl -replace '-bin\.[^.]*$',"-windows-amd64.zip"

    $MVN_CMD = "mvnd.cmd"

    break

  }

  default {

    $USE_MVND = $false

    $MVN_CMD = $script -replace '^mvnw','mvn'

    break

  }

}

# apply MVNW_REPOURL and calculate MAVEN_HOME

# maven home pattern: ~/.m2/wrapper/dists/{apache-maven-<version>,maven-mvnd-<version>-<platform>}/<hash>

if ($env:MVNW_REPOURL) {

  $MVNW_REPO_PATTERN = if ($USE_MVND) { "/org/apache/maven/" } else { "/maven/mvnd/" }

  $distributionUrl = "$env:MVNW_REPOURL$MVNW_REPO_PATTERN$($distributionUrl -replace '^.*'+$MVNW_REPO_PATTERN,")"

}

$distributionUrlName = $distributionUrl -replace '^.*/',"

$distributionUrlNameMain = $distributionUrlName -replace '\.[^.]*$'," -replace '-bin$',"

$MAVEN_HOME_PARENT = "$HOME/.m2/wrapper/dists/$distributionUrlNameMain"
```

```
if ($env:MAVEN_USER_HOME) {
  $MAVEN_HOME_PARENT =
"$env:MAVEN_USER_HOME/wrapper/dists/$distributionUrlNameMain"
}
$MAVEN_HOME_NAME =
([System.Security.Cryptography.MD5]::Create().ComputeHash([byte[]][char[]]$distribution
Url) | ForEach-Object {$_.ToString("x2")}) -join ''
$MAVEN_HOME = "$MAVEN_HOME_PARENT/$MAVEN_HOME_NAME"
if (Test-Path -Path "$MAVEN_HOME" -PathType Container) {
  Write-Verbose "found existing MAVEN_HOME at $MAVEN_HOME"
  Write-Output "MVN_CMD=$MAVEN_HOME/bin/$MVN_CMD"
  exit $?
}
if (! $distributionUrlNameMain -or ($distributionUrlName -eq $distributionUrlNameMain))
{
  Write-Error "distributionUrl is not valid, must end with *-bin.zip, but found
$distributionUrl"
}
# prepare tmp dir
$TMP_DOWNLOAD_DIR_HOLDER = New-TemporaryFile
$TMP_DOWNLOAD_DIR = New-Item -Itemtype Directory -Path
"$TMP_DOWNLOAD_DIR_HOLDER.dir"
$TMP_DOWNLOAD_DIR_HOLDER.Delete() | Out-Null
trap {
  if ($TMP_DOWNLOAD_DIR.Exists) {
    try { Remove-Item $TMP_DOWNLOAD_DIR -Recurse -Force | Out-Null }
    catch { Write-Warning "Cannot remove $TMP_DOWNLOAD_DIR" }
  }
}
New-Item -Itemtype Directory -Path "$MAVEN_HOME_PARENT" -Force | Out-Null
# Download and Install Apache Maven
Write-Verbose "Couldn't find MAVEN_HOME, downloading and installing it ..."
```

```
Write-Verbose "Downloading from: $distributionUrl"

Write-Verbose "Downloading to: $TMP_DOWNLOAD_DIR/$distributionUrlName"

$webclient = New-Object System.Net.WebClient

if ($env:MVNW_USERNAME -and $env:MVNW_PASSWORD) {

  $webclient.Credentials = New-Object
System.Net.NetworkCredential($env:MVNW_USERNAME, $env:MVNW_PASSWORD)

}

[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12

$webclient.DownloadFile($distributionUrl,
"$TMP_DOWNLOAD_DIR/$distributionUrlName") | Out-Null


# If specified, validate the SHA-256 sum of the Maven distribution zip file

$distributionSha256Sum = (Get-Content -Raw "$scriptDir/.mvn/wrapper/maven-
wrapper.properties" | ConvertFrom-StringData).distributionSha256Sum

if ($distributionSha256Sum) {

  if ($USE_MVND) {

    Write-Error "Checksum validation is not supported for maven-mvnd. `nPlease disable
validation by removing 'distributionSha256Sum' from your maven-wrapper.properties."

  }

  Import-Module $PSHOME\Modules\Microsoft.PowerShell.Utility -Function Get-FileHash

  if ((Get-FileHash "$TMP_DOWNLOAD_DIR/$distributionUrlName" -Algorithm
SHA256).Hash.ToLower() -ne $distributionSha256Sum) {

    Write-Error "Error: Failed to validate Maven distribution SHA-256, your Maven
distribution might be compromised. If you updated your Maven version, you need to update
the specified distributionSha256Sum property."

  }

}

# unzip and move

Expand-Archive "$TMP_DOWNLOAD_DIR/$distributionUrlName" -DestinationPath
"$TMP_DOWNLOAD_DIR" | Out-Null

Rename-Item -Path "$TMP_DOWNLOAD_DIR/$distributionUrlNameMain" -NewName
$MAVEN_HOME_NAME | Out-Null

try {
```

```
  Move-Item -Path "$TMP_DOWNLOAD_DIR/$MAVEN_HOME_NAME" -Destination
$MAVEN_HOME_PARENT | Out-Null
} catch {
  if (! (Test-Path -Path "$MAVEN_HOME" -PathType Container)) {
    Write-Error "fail to move MAVEN_HOME"
  }
} finally {
  try { Remove-Item $TMP_DOWNLOAD_DIR -Recurse -Force | Out-Null }
  catch { Write-Warning "Cannot remove $TMP_DOWNLOAD_DIR" }
}
Write-Output "MVN_CMD=$MAVEN_HOME/bin/$MVN_CMD"
```

//mvnw

```
#!/bin/sh
# ----------------------------------------------------------------------------
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied.  See the License for the
# specific language governing permissions and limitations
# under the License.
# ----------------------------------------------------------------------------
```

```
# ----------------------------------------------------------------------------
# Apache Maven Wrapper startup batch script, version 3.3.2
#
# Optional ENV vars
# -----------
#   JAVA_HOME - location of a JDK home dir, required when download maven via java source
#   MVNW_REPOURL - repo url base for downloading maven distribution
#   MVNW_USERNAME/MVNW_PASSWORD - user and password for downloading maven
#   MVNW_VERBOSE - true: enable verbose log; debug: trace the mvnw script; others: silence the output
# ----------------------------------------------------------------------------
set -euf
[ "${MVNW_VERBOSE-}" != debug ] || set -x
# OS specific support.
native_path() { printf %s\\n "$1"; }
case "$(uname)" in
CYGWIN* | MINGW*)
  [ -z "${JAVA_HOME-}" ] || JAVA_HOME="$(cygpath --unix "$JAVA_HOME")"
  native_path() { cygpath --path --windows "$1"; }
  ;;
esac
# set JAVACMD and JAVACCMD
set_java_home() {
  # For Cygwin and MinGW, ensure paths are in Unix format before anything is touched
  if [ -n "${JAVA_HOME-}" ]; then
    if [ -x "$JAVA_HOME/jre/sh/java" ]; then
      # IBM's JDK on AIX uses strange locations for the executables
      JAVACMD="$JAVA_HOME/jre/sh/java"
```

```
      JAVACCMD="$JAVA_HOME/jre/sh/javac"
    else
      JAVACMD="$JAVA_HOME/bin/java"
      JAVACCMD="$JAVA_HOME/bin/javac"
      if [ ! -x "$JAVACMD" ] || [ ! -x "$JAVACCMD" ]; then
        echo "The JAVA_HOME environment variable is not defined correctly, so mvnw cannot run." >&2
        echo "JAVA_HOME is set to \"$JAVA_HOME\", but \"\$JAVA_HOME/bin/java\" or \"\$JAVA_HOME/bin/javac\" does not exist." >&2
        return 1
      fi
  fi
  else
    JAVACMD="$(
      'set' +e
      'unset' -f command 2>/dev/null
      'command' -v java
    )" || :
    JAVACCMD="$(
      'set' +e
      'unset' -f command 2>/dev/null
      'command' -v javac
    )" || :
    if [ ! -x "${JAVACMD-}" ] || [ ! -x "${JAVACCMD-}" ]; then
      echo "The java/javac command does not exist in PATH nor is JAVA_HOME set, so mvnw cannot run." >&2
      return 1
    fi
  fi
}
# hash string like Java String::hashCode
```

```
hash_string() {
  str="${1:-}" h=0
  while [ -n "$str" ]; do
    char="${str%"${str#?}"}"
    h=$(((h * 31 + $(LC_CTYPE=C printf %d "'$char")) % 4294967296))
    str="${str#?}"
  done
  printf %x\\n $h
}
verbose() { :; }
[ "${MVNW_VERBOSE-}" != true ] || verbose() { printf %s\\n "${1-}"; }
die() {
  printf %s\\n "$1" >&2
  exit 1
}
trim() {
  # MWRAPPER-139:
  #   Trims trailing and leading whitespace, carriage returns, tabs, and linefeeds.
  #   Needed for removing poorly interpreted newline sequences when running in more
  #   exotic environments such as mingw bash on Windows.
  printf "%s" "${1}" | tr -d '[:space:]'
}
# parse distributionUrl and optional distributionSha256Sum, requires .mvn/wrapper/maven-wrapper.properties
while IFS="=" read -r key value; do
  case "${key-}" in
  distributionUrl) distributionUrl=$(trim "${value-}") ;;
  distributionSha256Sum) distributionSha256Sum=$(trim "${value-}") ;;
  esac
done <"${0%/*}/.mvn/wrapper/maven-wrapper.properties"
```

```
[ -n "${distributionUrl-}" ] || die "cannot read distributionUrl property in ${0%/*}/.mvn/wrapper/maven-wrapper.properties"

case "${distributionUrl##*/}" in

maven-mvnd-bin.)

  MVN_CMD=mvnd.sh _MVNW_REPO_PATTERN=/maven/mvnd/

  case  "${PROCESSOR_ARCHITECTURE-}${PROCESSOR_ARCHITEW6432-}:$(uname -a)" in

  AMD64:CYGWIN | AMD64:MINGW) distributionPlatform=windows-amd64 ;;

  :Darwin*x86_64) distributionPlatform=darwin-amd64 ;;

  :Darwin*arm64) distributionPlatform=darwin-aarch64 ;;

  :Linux*x86_64*) distributionPlatform=linux-amd64 ;;

  *)

    echo "Cannot detect native platform for mvnd on $(uname)-$(uname -m), use pure java version" >&2

    distributionPlatform=linux-amd64

    ;;

  esac

  distributionUrl="${distributionUrl%-bin.*}-$distributionPlatform.zip"

  ;;

maven-mvnd-*) MVN_CMD=mvnd.sh _MVNW_REPO_PATTERN=/maven/mvnd/ ;;

) MVN_CMD="mvn${0##/mvnw}" _MVNW_REPO_PATTERN=/org/apache/maven/ ;;

esac

# apply MVNW_REPOURL and calculate MAVEN_HOME

# maven home pattern: ~/.m2/wrapper/dists/{apache-maven-<version>,maven-mvnd-<version>-<platform>}/<hash>

[ -z "${MVNW_REPOURL-}" ] || distributionUrl="$MVNW_REPOURL$_MVNW_REPO_PATTERN${distributionUrl#*"$_MVNW_REPO_PATTERN"}"

distributionUrlName="${distributionUrl##*/}"

distributionUrlNameMain="${distributionUrlName%.*}"

distributionUrlNameMain="${distributionUrlNameMain%-bin}"

MAVEN_USER_HOME="${MAVEN_USER_HOME:-${HOME}/.m2}"
```

```
MAVEN_HOME="${MAVEN_USER_HOME}/wrapper/dists/${distributionUrlNameMain-}/$(hash_string "$distributionUrl")"

exec_maven() {
  unset MVNW_VERBOSE MVNW_USERNAME MVNW_PASSWORD MVNW_REPOURL || :
  exec "$MAVEN_HOME/bin/$MVN_CMD" "$@" || die "cannot exec $MAVEN_HOME/bin/$MVN_CMD"
}

if [ -d "$MAVEN_HOME" ]; then
  verbose "found existing MAVEN_HOME at $MAVEN_HOME"
  exec_maven "$@"
fi

case "${distributionUrl-}" in
?-bin.zip | *?maven-mvnd-?-?*.zip) ;;
) die "distributionUrl is not valid, must match *-bin.zip or maven-mvnd-.zip, but found '${distributionUrl-}'" ;;
esac

# prepare tmp dir
if TMP_DOWNLOAD_DIR="$(mktemp -d)" && [ -d "$TMP_DOWNLOAD_DIR" ]; then
  clean() { rm -rf -- "$TMP_DOWNLOAD_DIR"; }
  trap clean HUP INT TERM EXIT
else
  die "cannot create temp dir"
fi

mkdir -p -- "${MAVEN_HOME%/*}"

# Download and Install Apache Maven
verbose "Couldn't find MAVEN_HOME, downloading and installing it ..."
verbose "Downloading from: $distributionUrl"
verbose "Downloading to: $TMP_DOWNLOAD_DIR/$distributionUrlName"

# select .zip or .tar.gz
if ! command -v unzip >/dev/null; then
  distributionUrl="${distributionUrl%.zip}.tar.gz"
```

```
  distributionUrlName="${distributionUrl##*/}"
fi

# verbose opt
__MVNW_QUIET_WGET=--quiet __MVNW_QUIET_CURL=--silent
__MVNW_QUIET_UNZIP=-q __MVNW_QUIET_TAR="
[ "${MVNW_VERBOSE-}" != true ] || __MVNW_QUIET_WGET="
__MVNW_QUIET_CURL=" __MVNW_QUIET_UNZIP=" __MVNW_QUIET_TAR=v

# normalize http auth
case "${MVNW_PASSWORD:+has-password}" in
") MVNW_USERNAME=" MVNW_PASSWORD=" ;;
has-password) [ -n "${MVNW_USERNAME-}" ] || MVNW_USERNAME="
MVNW_PASSWORD=" ;;
esac
if [ -z "${MVNW_USERNAME-}" ] && command -v wget >/dev/null; then
  verbose "Found wget ... using wget"
  wget ${_MVNW_QUIET_WGET:+"$_MVNW_QUIET_WGET"} "$distributionUrl" -O
"$TMP_DOWNLOAD_DIR/$distributionUrlName" || die "wget: Failed to fetch
$distributionUrl"
elif [ -z "${MVNW_USERNAME-}" ] && command -v curl >/dev/null; then
  verbose "Found curl ... using curl"
  curl ${_MVNW_QUIET_CURL:+"$_MVNW_QUIET_CURL"} -f -L -o
"$TMP_DOWNLOAD_DIR/$distributionUrlName" "$distributionUrl" || die "curl: Failed to
fetch $distributionUrl"
elif set_java_home; then
  verbose "Falling back to use Java to download"
  javaSource="$TMP_DOWNLOAD_DIR/Downloader.java"
  targetZip="$TMP_DOWNLOAD_DIR/$distributionUrlName"
  cat >"$javaSource" <<-END
        public class Downloader extends java.net.Authenticator
        {
          protected java.net.PasswordAuthentication getPasswordAuthentication()
          {
```

```
            return new java.net.PasswordAuthentication( System.getenv(
"MVNW_USERNAME" ), System.getenv( "MVNW_PASSWORD" ).toCharArray() );
          }
          public static void main( String[] args ) throws Exception
          {
            setDefault( new Downloader() );
            java.nio.file.Files.copy( java.net.URI.create( args[0] ).toURL().openStream(),
java.nio.file.Paths.get( args[1] ).toAbsolutePath().normalize() );
          }
        }
        END
  # For Cygwin/MinGW, switch paths to Windows format before running javac and java
  verbose " - Compiling Downloader.java ..."
  "$(native_path "$JAVACCMD")" "$(native_path "$javaSource")" || die "Failed to compile
Downloader.java"
  verbose " - Running Downloader.java ..."
  "$(native_path "$JAVACMD")" -cp "$(native_path "$TMP_DOWNLOAD_DIR")"
Downloader "$distributionUrl" "$(native_path "$targetZip")"
fi
# If specified, validate the SHA-256 sum of the Maven distribution zip file
if [ -n "${distributionSha256Sum-}" ]; then
distributionSha256Result=false
  if [ "$MVN_CMD" = mvnd.sh ]; then
    echo "Checksum validation is not supported for maven-mvnd." >&2
    echo "Please disable validation by removing 'distributionSha256Sum' from your maven-
wrapper.properties." >&2
    exit 1
  elif command -v sha256sum >/dev/null; then
    if echo "$distributionSha256Sum  $TMP_DOWNLOAD_DIR/$distributionUrlName" |
sha256sum -c >/dev/null 2>&1; then
      distributionSha256Result=true
    fi
```

```
  elif command -v shasum >/dev/null; then
    if echo "$distributionSha256Sum  $TMP_DOWNLOAD_DIR/$distributionUrlName" |
shasum -a 256 -c >/dev/null 2>&1; then
      distributionSha256Result=true
    fi
  else
    echo "Checksum validation was requested but neither 'sha256sum' or 'shasum' are
available." >&2
    echo "Please install either command, or disable validation by removing
'distributionSha256Sum' from your maven-wrapper.properties." >&2
    exit 1
  fi
  if [ $distributionSha256Result = false ]; then
    echo "Error: Failed to validate Maven distribution SHA-256, your Maven distribution
might be compromised." >&2
    echo "If you updated your Maven version, you need to update the specified
distributionSha256Sum property." >&2
    exit 1
  fi
fi

# unzip and move

if command -v unzip >/dev/null; then
  unzip ${_MVNW_QUIET_UNZIP:+"$_MVNW_QUIET_UNZIP"}
"$TMP_DOWNLOAD_DIR/$distributionUrlName" -d "$TMP_DOWNLOAD_DIR" || die
"failed to unzip"

else

  tar xzf${_MVNW_QUIET_TAR:+"$_MVNW_QUIET_TAR"}
"$TMP_DOWNLOAD_DIR/$distributionUrlName" -C "$TMP_DOWNLOAD_DIR" || die
"failed to untar"

fi

printf %s\\n "$distributionUrl"
>"$TMP_DOWNLOAD_DIR/$distributionUrlNameMain/mvnw.url"

mv -- "$TMP_DOWNLOAD_DIR/$distributionUrlNameMain" "$MAVEN_HOME" || [ -d
"$MAVEN_HOME" ] || die "fail to move MAVEN_HOME"
```

```
clean || :
exec_maven "$@"
//.gitignore
HELP.md
target/
.mvn/wrapper/maven-wrapper.jar
!/src/main//target/
!/src/test//target/

### STS ###
.apt_generated
.classpath
.factorypath
.project
.settings
.springBeans
.sts4-cache

### IntelliJ IDEA ###
.idea
*.iws
*.iml
*.ipr

### NetBeans ###
/nbproject/private/
/nbbuild/
/dist/
/nbdist/
/.nb-gradle/
```

build/

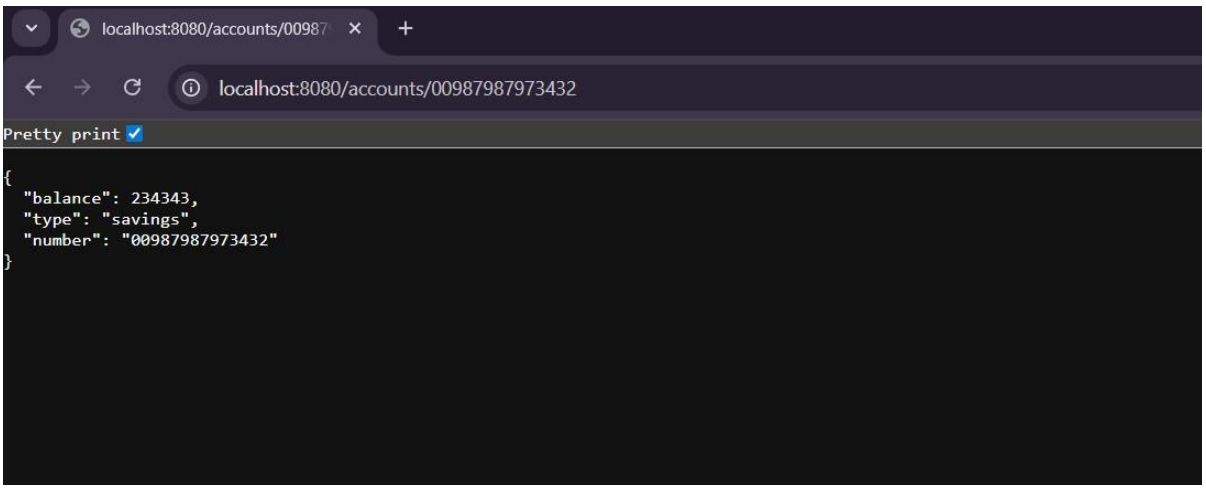
!/src/main//build/

!/src/test//build/

VS Code

.vscode/

OUTPUT:



Loan Microservice:

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

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.5.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.cognizant</groupId>

```
<artifactId>loan</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>loan</name>
<description>Demo project for Spring Boot</description>
<url/>
<licenses>
	<license/>
</licenses>
<developers>
	<developer/>
</developers>
<scm>
	<connection/>
	<developerConnection/>
	<tag/>
	<url/>
</scm>
<properties>
	<java.version>17</java.version>
</properties>
<dependencies>
	<dependency>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-web</artifactId>
	</dependency>
	<dependency>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-devtools</artifactId>
		<scope>runtime</scope>
		<optional>true</optional>
```

```
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-test</artifactId>
                <scope>test</scope>
            </dependency>
        </dependencies>
        <build>
            <plugins>
                <plugin>
                    <groupId>org.springframework.boot</groupId>
                    <artifactId>spring-boot-maven-plugin</artifactId>
                </plugin>
            </plugins>
        </build>
</project>
```

//mvnw.cmd

```
<# : batch portion
@REM ----------------------------------------------------------------------------
@REM Licensed to the Apache Software Foundation (ASF) under one
@REM or more contributor license agreements.  See the NOTICE file
@REM distributed with this work for additional information
@REM regarding copyright ownership.  The ASF licenses this file
@REM to you under the Apache License, Version 2.0 (the
@REM "License"); you may not use this file except in compliance
@REM with the License.  You may obtain a copy of the License at
@REM
@REM    http://www.apache.org/licenses/LICENSE-2.0
@REM
@REM Unless required by applicable law or agreed to in writing,
```

```
@REM software distributed under the License is distributed on an
@REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
@REM KIND, either express or implied.  See the License for the
@REM specific language governing permissions and limitations
@REM under the License.
@REM ----------------------------------------------------------------------------
@REM ----------------------------------------------------------------------------
@REM Apache Maven Wrapper startup batch script, version 3.3.2
@REM
@REM Optional ENV vars
@REM   MVNW_REPOURL - repo url base for downloading maven distribution
@REM  MVNW_USERNAME/MVNW_PASSWORD - user and password for downloading maven
@REM  MVNW_VERBOSE - true: enable verbose log; others: silence the output
@REM ----------------------------------------------------------------------------
@IF "%_MVNW_ARG0_NAME%"=="" (SET __MVNW_ARG0_NAME__=%~nx0)
@SET _MVNW_CMD_=
@SET _MVNW_ERROR_=
@SET __MVNW_PSMODULEP_SAVE=%PSModulePath%
@SET PSModulePath=
@FOR /F "usebackq tokens=1* delims==" %%A IN (powershell -noprofile "& {$scriptDir='%~dp0'; $script='%__MVNW_ARG0_NAME__%'; icm -ScriptBlock ([Scriptblock]::Create((Get-Content -Raw '%~f0'))) -NoNewScope}") DO @(
  IF "%%A"=="MVN_CMD" (set _MVNW_CMD_=%%B) ELSE IF "%%B"=="" (echo %%A) ELSE (echo %%A=%%B)
)
@SET PSModulePath=%__MVNW_PSMODULEP_SAVE%
@SET __MVNW_PSMODULEP_SAVE=
@SET _MVNW_ARG0_NAME_=
@SET MVNW_USERNAME=
@SET MVNW_PASSWORD=
@IF NOT "%_MVNW_CMD%"=="" (%MVNW_CMD_% %*)
```

```
@echo Cannot start maven from wrapper >&2 && exit /b 1
@GOTO :EOF
: end batch / begin powershell #>
$ErrorActionPreference = "Stop"
if ($env:MVNW_VERBOSE -eq "true") {
  $VerbosePreference = "Continue"
}
# calculate distributionUrl, requires .mvn/wrapper/maven-wrapper.properties
$distributionUrl = (Get-Content -Raw "$scriptDir/.mvn/wrapper/maven-wrapper.properties" |
ConvertFrom-StringData).distributionUrl
if (!$distributionUrl) {
  Write-Error "cannot read distributionUrl property in $scriptDir/.mvn/wrapper/maven-
wrapper.properties"
}
switch -wildcard -casesensitive ( $($distributionUrl -replace '^.*/','') ) {
  "maven-mvnd-*" {
    $USE_MVND = $true
    $distributionUrl = $distributionUrl -replace '-bin\.[^.]*$',"-windows-amd64.zip"
    $MVN_CMD = "mvnd.cmd"
    break
  }
  default {
    $USE_MVND = $false
    $MVN_CMD = $script -replace '^mvnw','mvn'
    break
  }
}
# apply MVNW_REPOURL and calculate MAVEN_HOME
# maven home pattern: ~/.m2/wrapper/dists/{apache-maven-<version>,maven-mvnd-
<version>-<platform>}/<hash>
if ($env:MVNW_REPOURL) {
```

```
  $MVNW_REPO_PATTERN = if ($USE_MVND) { "/org/apache/maven/" } else {
"/maven/mvnd/" }

  $distributionUrl =
"$env:MVNW_REPOURL$MVNW_REPO_PATTERN$($distributionUrl -replace
'^.*'+$MVNW_REPO_PATTERN,")"

}

$distributionUrlName = $distributionUrl -replace '^.*/',"

$distributionUrlNameMain = $distributionUrlName -replace '\.[^.]*$'," -replace '-bin$',"

$MAVEN_HOME_PARENT = "$HOME/.m2/wrapper/dists/$distributionUrlNameMain"

if ($env:MAVEN_USER_HOME) {

  $MAVEN_HOME_PARENT =
"$env:MAVEN_USER_HOME/wrapper/dists/$distributionUrlNameMain"

}

$MAVEN_HOME_NAME =
([System.Security.Cryptography.MD5]::Create().ComputeHash([byte[]][char[]]$distribution
Url) | ForEach-Object {$_.ToString("x2")}) -join ''

$MAVEN_HOME = "$MAVEN_HOME_PARENT/$MAVEN_HOME_NAME"

if (Test-Path -Path "$MAVEN_HOME" -PathType Container) {

  Write-Verbose "found existing MAVEN_HOME at $MAVEN_HOME"

  Write-Output "MVN_CMD=$MAVEN_HOME/bin/$MVN_CMD"

  exit $?

}

if (! $distributionUrlNameMain -or ($distributionUrlName -eq $distributionUrlNameMain))
{

  Write-Error "distributionUrl is not valid, must end with *-bin.zip, but found
$distributionUrl"

}

# prepare tmp dir

$TMP_DOWNLOAD_DIR_HOLDER = New-TemporaryFile

$TMP_DOWNLOAD_DIR = New-Item -Itemtype Directory -Path
"$TMP_DOWNLOAD_DIR_HOLDER.dir"

$TMP_DOWNLOAD_DIR_HOLDER.Delete() | Out-Null

trap {

  if ($TMP_DOWNLOAD_DIR.Exists) {
```

```
    try { Remove-Item $TMP_DOWNLOAD_DIR -Recurse -Force | Out-Null }
    catch { Write-Warning "Cannot remove $TMP_DOWNLOAD_DIR" }
  }
}
New-Item -Itemtype Directory -Path "$MAVEN_HOME_PARENT" -Force | Out-Null
# Download and Install Apache Maven
Write-Verbose "Couldn't find MAVEN_HOME, downloading and installing it ..."
Write-Verbose "Downloading from: $distributionUrl"
Write-Verbose "Downloading to: $TMP_DOWNLOAD_DIR/$distributionUrlName"
$webclient = New-Object System.Net.WebClient
if ($env:MVNW_USERNAME -and $env:MVNW_PASSWORD) {
  $webclient.Credentials = New-Object
System.Net.NetworkCredential($env:MVNW_USERNAME, $env:MVNW_PASSWORD)
}
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
$webclient.DownloadFile($distributionUrl,
"$TMP_DOWNLOAD_DIR/$distributionUrlName") | Out-Null
# If specified, validate the SHA-256 sum of the Maven distribution zip file
$distributionSha256Sum = (Get-Content -Raw "$scriptDir/.mvn/wrapper/maven-
wrapper.properties" | ConvertFrom-StringData).distributionSha256Sum
if ($distributionSha256Sum) {
  if ($USE_MVND) {
    Write-Error "Checksum validation is not supported for maven-mvnd. `nPlease disable
validation by removing 'distributionSha256Sum' from your maven-wrapper.properties."
  }
  Import-Module $PSHOME\Modules\Microsoft.PowerShell.Utility -Function Get-FileHash
  if ((Get-FileHash "$TMP_DOWNLOAD_DIR/$distributionUrlName" -Algorithm
SHA256).Hash.ToLower() -ne $distributionSha256Sum) {
    Write-Error "Error: Failed to validate Maven distribution SHA-256, your Maven
distribution might be compromised. If you updated your Maven version, you need to update
the specified distributionSha256Sum property."
  }
}
```

```
# unzip and move
Expand-Archive "$TMP_DOWNLOAD_DIR/$distributionUrlName" -DestinationPath
"$TMP_DOWNLOAD_DIR" | Out-Null
Rename-Item -Path "$TMP_DOWNLOAD_DIR/$distributionUrlNameMain" -NewName
$MAVEN_HOME_NAME | Out-Null
try {
  Move-Item -Path "$TMP_DOWNLOAD_DIR/$MAVEN_HOME_NAME" -Destination
$MAVEN_HOME_PARENT | Out-Null
} catch {
  if (! (Test-Path -Path "$MAVEN_HOME" -PathType Container)) {
    Write-Error "fail to move MAVEN_HOME"
  }
} finally {
  try { Remove-Item $TMP_DOWNLOAD_DIR -Recurse -Force | Out-Null }
  catch { Write-Warning "Cannot remove $TMP_DOWNLOAD_DIR" }
}
Write-Output "MVN_CMD=$MAVEN_HOME/bin/$MVN_CMD"
```

//mvnw

```
#!/bin/sh
# ----------------------------------------------------------------------
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
```

```
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied.  See the License for the
# specific language governing permissions and limitations
# under the License.
# ----------------------------------------------------------------------------
# ----------------------------------------------------------------------------
# Apache Maven Wrapper startup batch script, version 3.3.2
#
# Optional ENV vars
# -----------
#   JAVA_HOME - location of a JDK home dir, required when download maven via java source
#   MVNW_REPOURL - repo url base for downloading maven distribution
#   MVNW_USERNAME/MVNW_PASSWORD - user and password for downloading maven
#   MVNW_VERBOSE - true: enable verbose log; debug: trace the mvnw script; others: silence the output
# ----------------------------------------------------------------------------
set -euf
[ "${MVNW_VERBOSE-}" != debug ] || set -x
# OS specific support.
native_path() { printf %s\\n "$1"; }
case "$(uname)" in
CYGWIN* | MINGW*)
  [ -z "${JAVA_HOME-}" ] || JAVA_HOME="$(cygpath --unix "$JAVA_HOME")"
  native_path() { cygpath --path --windows "$1"; }
  ;;
esac
# set JAVACMD and JAVACCMD
set_java_home() {
```

```
  # For Cygwin and MinGW, ensure paths are in Unix format before anything is touched
  if [ -n "${JAVA_HOME-}" ]; then
    if [ -x "$JAVA_HOME/jre/sh/java" ]; then
      # IBM's JDK on AIX uses strange locations for the executables
      JAVACMD="$JAVA_HOME/jre/sh/java"
      JAVACCMD="$JAVA_HOME/jre/sh/javac"
    else
      JAVACMD="$JAVA_HOME/bin/java"
      JAVACCMD="$JAVA_HOME/bin/javac"
      if [ ! -x "$JAVACMD" ] || [ ! -x "$JAVACCMD" ]; then
        echo "The JAVA_HOME environment variable is not defined correctly, so mvnw cannot run." >&2
        echo "JAVA_HOME is set to \"$JAVA_HOME\", but \"\$JAVA_HOME/bin/java\" or \"\$JAVA_HOME/bin/javac\" does not exist." >&2
        return 1
      fi
  fi
  else
    JAVACMD="$(
      'set' +e
      'unset' -f command 2>/dev/null
      'command' -v java
    )" || :
    JAVACCMD="$(
      'set' +e
      'unset' -f command 2>/dev/null
      'command' -v javac
    )" || :
    if [ ! -x "${JAVACMD-}" ] || [ ! -x "${JAVACCMD-}" ]; then
      echo "The java/javac command does not exist in PATH nor is JAVA_HOME set, so mvnw cannot run." >&2
```

```
      return 1
    fi
  fi
}
# hash string like Java String::hashCode
hash_string() {
  str="${1:-}" h=0
  while [ -n "$str" ]; do
    char="${str%"${str#?}"}"
    h=$(((h * 31 + $(LC_CTYPE=C printf %d "'$char")) % 4294967296))
    str="${str#?}"
  done
  printf %x\\n $h
}
verbose() { :; }
[ "${MVNW_VERBOSE-}" != true ] || verbose() { printf %s\\n "${1-}"; }
die() {
  printf %s\\n "$1" >&2
  exit 1
}
trim() {
  # MWRAPPER-139:
  #   Trims trailing and leading whitespace, carriage returns, tabs, and linefeeds.
  #   Needed for removing poorly interpreted newline sequences when running in more
  #   exotic environments such as mingw bash on Windows.
  printf "%s" "${1}" | tr -d '[:space:]'
}
# parse distributionUrl and optional distributionSha256Sum, requires .mvn/wrapper/maven-wrapper.properties
while IFS="=" read -r key value; do
  case "${key-}" in
```

```
  distributionUrl) distributionUrl=$(trim "${value-}") ;;
  distributionSha256Sum) distributionSha256Sum=$(trim "${value-}") ;;
  esac
done <"${0%/*}/.mvn/wrapper/maven-wrapper.properties"
[ -n "${distributionUrl-}" ] || die "cannot read distributionUrl property in
${0%/*}/.mvn/wrapper/maven-wrapper.properties"
case "${distributionUrl##*/}" in
maven-mvnd-bin.)
  MVN_CMD=mvnd.sh _MVNW_REPO_PATTERN=/maven/mvnd/
  case  "${PROCESSOR_ARCHITECTURE-}${PROCESSOR_ARCHITEW6432-}:$(uname
-a)" in
  AMD64:CYGWIN | AMD64:MINGW) distributionPlatform=windows-amd64 ;;
  :Darwin*x86_64) distributionPlatform=darwin-amd64 ;;
  :Darwin*arm64) distributionPlatform=darwin-aarch64 ;;
  :Linux*x86_64*) distributionPlatform=linux-amd64 ;;
  *)
    echo "Cannot detect native platform for mvnd on $(uname)-$(uname -m), use pure java
version" >&2
    distributionPlatform=linux-amd64
    ;;
  esac
  distributionUrl="${distributionUrl%-bin.*}-$distributionPlatform.zip"
  ;;
maven-mvnd-*) MVN_CMD=mvnd.sh _MVNW_REPO_PATTERN=/maven/mvnd/ ;;
) MVN_CMD="mvn${0##/mvnw}" _MVNW_REPO_PATTERN=/org/apache/maven/ ;;
esac
# apply MVNW_REPOURL and calculate MAVEN_HOME
# maven home pattern: ~/.m2/wrapper/dists/{apache-maven-<version>,maven-mvnd-
<version>-<platform>}/<hash>
[ -z "${MVNW_REPOURL-}" ] ||
distributionUrl="$MVNW_REPOURL$_MVNW_REPO_PATTERN${distributionUrl#*"$_
MVNW_REPO_PATTERN"}"
```

```
distributionUrlName="${distributionUrl##*/}"
distributionUrlNameMain="${distributionUrlName%.*}"
distributionUrlNameMain="${distributionUrlNameMain%-bin}"
MAVEN_USER_HOME="${MAVEN_USER_HOME:-${HOME}/.m2}"
MAVEN_HOME="${MAVEN_USER_HOME}/wrapper/dists/${distributionUrlNameMain-}/$(hash_string "$distributionUrl")"
exec_maven() {
  unset MVNW_VERBOSE MVNW_USERNAME MVNW_PASSWORD MVNW_REPOURL || :
  exec "$MAVEN_HOME/bin/$MVN_CMD" "$@" || die "cannot exec $MAVEN_HOME/bin/$MVN_CMD"
}
if [ -d "$MAVEN_HOME" ]; then
  verbose "found existing MAVEN_HOME at $MAVEN_HOME"
  exec_maven "$@"
fi
case "${distributionUrl-}" in
?-bin.zip | *?maven-mvnd-?-?*.zip) ;;
) die "distributionUrl is not valid, must match *-bin.zip or maven-mvnd-.zip, but found '${distributionUrl-}'" ;;
esac
# prepare tmp dir
if TMP_DOWNLOAD_DIR="$(mktemp -d)" && [ -d "$TMP_DOWNLOAD_DIR" ]; then
  clean() { rm -rf -- "$TMP_DOWNLOAD_DIR"; }
  trap clean HUP INT TERM EXIT
else
  die "cannot create temp dir"
fi
mkdir -p -- "${MAVEN_HOME%/*}"
# Download and Install Apache Maven
verbose "Couldn't find MAVEN_HOME, downloading and installing it ..."
verbose "Downloading from: $distributionUrl"
```

```
verbose "Downloading to: $TMP_DOWNLOAD_DIR/$distributionUrlName"

# select .zip or .tar.gz

if ! command -v unzip >/dev/null; then

  distributionUrl="${distributionUrl%.zip}.tar.gz"

  distributionUrlName="${distributionUrl##*/}"

fi

# verbose opt

__MVNW_QUIET_WGET=--quiet __MVNW_QUIET_CURL=--silent
__MVNW_QUIET_UNZIP=-q __MVNW_QUIET_TAR="

[ "${MVNW_VERBOSE-}" != true ] || __MVNW_QUIET_WGET="
__MVNW_QUIET_CURL=" __MVNW_QUIET_UNZIP=" __MVNW_QUIET_TAR=v

# normalize http auth

case "${MVNW_PASSWORD:+has-password}" in

") MVNW_USERNAME=" MVNW_PASSWORD=" ;;

has-password) [ -n "${MVNW_USERNAME-}" ] || MVNW_USERNAME="
MVNW_PASSWORD=" ;;

esac

if [ -z "${MVNW_USERNAME-}" ] && command -v wget >/dev/null; then

  verbose "Found wget ... using wget"

  wget ${_MVNW_QUIET_WGET:+"$_MVNW_QUIET_WGET"} "$distributionUrl" -O
"$TMP_DOWNLOAD_DIR/$distributionUrlName" || die "wget: Failed to fetch
$distributionUrl"

elif [ -z "${MVNW_USERNAME-}" ] && command -v curl >/dev/null; then

  verbose "Found curl ... using curl"

  curl ${_MVNW_QUIET_CURL:+"$_MVNW_QUIET_CURL"} -f -L -o
"$TMP_DOWNLOAD_DIR/$distributionUrlName" "$distributionUrl" || die "curl: Failed to
fetch $distributionUrl"

elif set_java_home; then

  verbose "Falling back to use Java to download"

  javaSource="$TMP_DOWNLOAD_DIR/Downloader.java"

  targetZip="$TMP_DOWNLOAD_DIR/$distributionUrlName"

  cat >"$javaSource" <<-END

        public class Downloader extends java.net.Authenticator
```

```
        {
          protected java.net.PasswordAuthentication getPasswordAuthentication()
          {
            return new java.net.PasswordAuthentication( System.getenv(
"MVNW_USERNAME" ), System.getenv( "MVNW_PASSWORD" ).toCharArray() );
          }
          public static void main( String[] args ) throws Exception
          {
            setDefault( new Downloader() );
            java.nio.file.Files.copy( java.net.URI.create( args[0] ).toURL().openStream(),
java.nio.file.Paths.get( args[1] ).toAbsolutePath().normalize() );
          }
        }
        END
  # For Cygwin/MinGW, switch paths to Windows format before running javac and java
  verbose " - Compiling Downloader.java ..."
  "$(native_path "$JAVACCMD")" "$(native_path "$javaSource")" || die "Failed to compile
Downloader.java"
  verbose " - Running Downloader.java ..."
  "$(native_path "$JAVACMD")" -cp "$(native_path "$TMP_DOWNLOAD_DIR")"
Downloader "$distributionUrl" "$(native_path "$targetZip")"
fi
# If specified, validate the SHA-256 sum of the Maven distribution zip file
if [ -n "${distributionSha256Sum-}" ]; then
distributionSha256Result=false
  if [ "$MVN_CMD" = mvnd.sh ]; then
    echo "Checksum validation is not supported for maven-mvnd." >&2
    echo "Please disable validation by removing 'distributionSha256Sum' from your maven-
wrapper.properties." >&2
    exit 1
  elif command -v sha256sum >/dev/null; then
```

```
    if echo "$distributionSha256Sum  $TMP_DOWNLOAD_DIR/$distributionUrlName" |
sha256sum -c >/dev/null 2>&1; then
      distributionSha256Result=true
    fi
  elif command -v shasum >/dev/null; then
    if echo "$distributionSha256Sum  $TMP_DOWNLOAD_DIR/$distributionUrlName" |
shasum -a 256 -c >/dev/null 2>&1; then
      distributionSha256Result=true
    fi
  else
    echo "Checksum validation was requested but neither 'sha256sum' or 'shasum' are
available." >&2
    echo "Please install either command, or disable validation by removing
'distributionSha256Sum' from your maven-wrapper.properties." >&2
    exit 1
  fi
  if [ $distributionSha256Result = false ]; then
    echo "Error: Failed to validate Maven distribution SHA-256, your Maven distribution
might be compromised." >&2
    echo "If you updated your Maven version, you need to update the specified
distributionSha256Sum property." >&2
    exit 1
  fi
fi

# unzip and move

if command -v unzip >/dev/null; then
  unzip ${_MVNW_QUIET_UNZIP:+"$_MVNW_QUIET_UNZIP"}
"$TMP_DOWNLOAD_DIR/$distributionUrlName" -d "$TMP_DOWNLOAD_DIR" || die
"failed to unzip"

else

  tar xzf${_MVNW_QUIET_TAR:+"$_MVNW_QUIET_TAR"}
"$TMP_DOWNLOAD_DIR/$distributionUrlName" -C "$TMP_DOWNLOAD_DIR" || die
"failed to untar"

fi
```

```
printf %s\\n "$distributionUrl"
>"$TMP_DOWNLOAD_DIR/$distributionUrlNameMain/mvnw.url"

mv -- "$TMP_DOWNLOAD_DIR/$distributionUrlNameMain" "$MAVEN_HOME" || [ -d
"$MAVEN_HOME" ] || die "fail to move MAVEN_HOME"

clean || :

exec_maven "$@"
```

//.gitignore

```
HELP.md

target/

.mvn/wrapper/maven-wrapper.jar

!/src/main//target/

!/src/test//target/


### STS ###

.apt_generated

.classpath

.factorypath

.project

.settings

.springBeans

.sts4-cache


### IntelliJ IDEA ###

.idea

*.iws

*.iml

*.ipr


### NetBeans ###

/nbproject/private/

/nbbuild/
```

/dist/

/nbdist/

/.nb-gradle/

build/

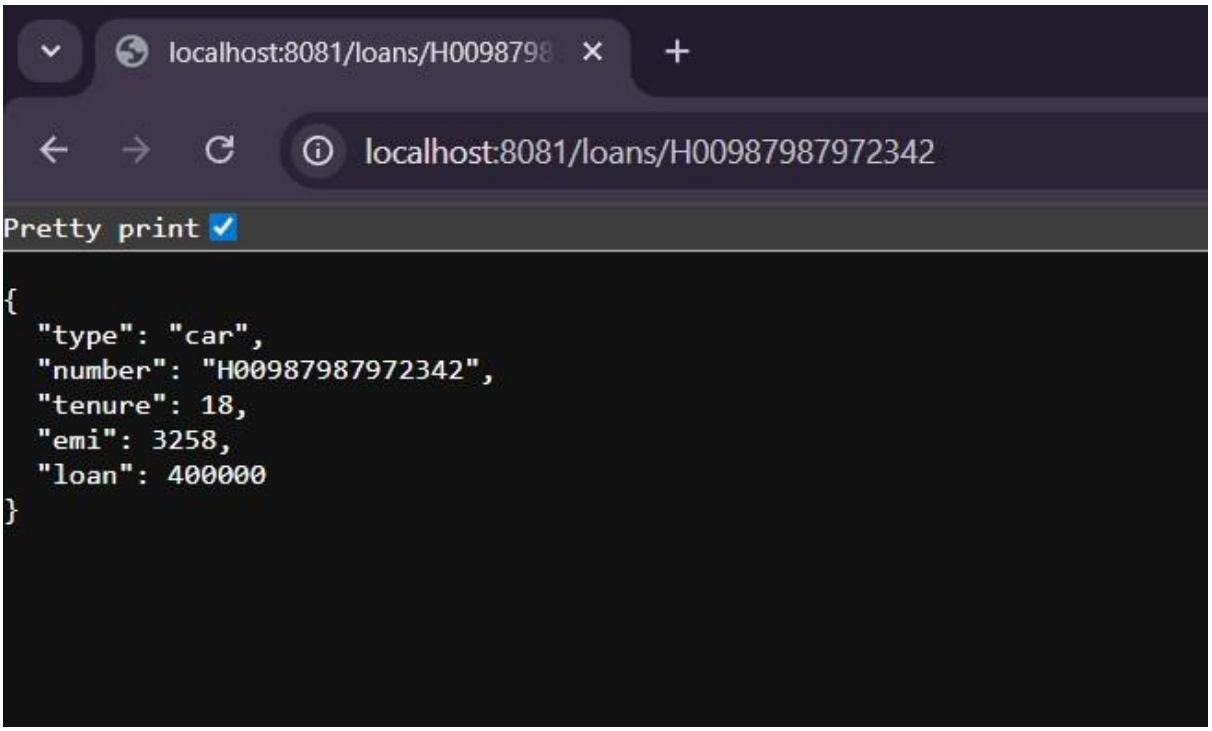
!/src/main//build/

!/src/test//build/

VS Code

.vscode/

OUTPUT:



2) Exercise 2: Create Eureka Discovery Server and register microservices

Solution:

Eureka Discovery Server

//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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.2.3</version>
        <relativePath/>
    </parent>
    <groupId>com.cognizant</groupId>
    <artifactId>eureka-discovery-server</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>eureka-discovery-server</name>
    <description>Eureka Discovery Server</description>
    <properties>
        <java.version>17</java.version>
        <spring-cloud.version>2023.0.1</spring-cloud.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
```

```
            <scope>runtime</scope>
        </dependency>
    </dependencies>
    <dependencyManagement>
        <dependencies>
            <dependency>
                <groupId>org.springframework.cloud</groupId>
                <artifactId>spring-cloud-dependencies</artifactId>
                <version>${spring-cloud.version}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

//mvnw.cmd

```
<# : batch portion
@REM ----------------------------------------------------------------
@REM Licensed to the Apache Software Foundation (ASF) under one
@REM or more contributor license agreements.  See the NOTICE file
@REM distributed with this work for additional information
@REM regarding copyright ownership.  The ASF licenses this file
```

```
@REM to you under the Apache License, Version 2.0 (the
@REM "License"); you may not use this file except in compliance
@REM with the License.  You may obtain a copy of the License at
@REM
@REM    http://www.apache.org/licenses/LICENSE-2.0
@REM
@REM Unless required by applicable law or agreed to in writing,
@REM software distributed under the License is distributed on an
@REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
@REM KIND, either express or implied.  See the License for the
@REM specific language governing permissions and limitations
@REM under the License.
@REM ----------------------------------------------------------
@REM ----------------------------------------------------------
@REM Apache Maven Wrapper startup batch script, version 3.3.2
@REM
@REM Optional ENV vars
@REM   MVNW_REPOURL - repo url base for downloading maven distribution
@REM   MVNW_USERNAME/MVNW_PASSWORD - user and password for
downloading maven
@REM   MVNW_VERBOSE - true: enable verbose log; others: silence the output
@REM ----------------------------------------------------------
@IF "%_MVNW_ARG0_NAME%"=="" (SET __MVNW_ARG0_NAME_=%~nx0)
@SET _MVNW_CMD_=
@SET _MVNW_ERROR_=
@SET __MVNW_PSMODULEP_SAVE=%PSModulePath%
@SET PSModulePath=
@FOR /F "usebackq tokens=1* delims==" %%A IN (powershell -noprofile "&
{$scriptDir='%~dp0'; $script='%__MVNW_ARG0_NAME__%'; icm -ScriptBlock
([Scriptblock]::Create((Get-Content -Raw '%~f0'))) -NoNewScope}") DO @(
  IF "%%A"=="MVN_CMD" (set _MVNW_CMD_=%%B) ELSE IF "%%B"=="" (echo
%%A) ELSE (echo %%A=%%B)
```

```
)
@SET PSModulePath=%_MVNW_PSMODULEP_SAVE%
@SET _MVNW_PSMODULEP_SAVE=
@SET _MVNW_ARG0_NAME_=
@SET MVNW_USERNAME=
@SET MVNW_PASSWORD=
@IF NOT "%_MVNW_CMD%"=="" (%MVNW_CMD_% %*)
@echo Cannot start maven from wrapper >&2 && exit /b 1
@GOTO :EOF
: end batch / begin powershell #>
$ErrorActionPreference = "Stop"
if ($env:MVNW_VERBOSE -eq "true") {
  $VerbosePreference = "Continue"
}
# calculate distributionUrl, requires .mvn/wrapper/maven-wrapper.properties
$distributionUrl = (Get-Content -Raw "$scriptDir/.mvn/wrapper/maven-wrapper.properties" |
ConvertFrom-StringData).distributionUrl
if (!$distributionUrl) {
  Write-Error "cannot read distributionUrl property in $scriptDir/.mvn/wrapper/maven-
wrapper.properties"
}
switch -wildcard -casesensitive ( $($distributionUrl -replace '^.*/','') ) {
  "maven-mvnd-*" {
    $USE_MVND = $true
    $distributionUrl = $distributionUrl -replace '-bin\.[^.]*$',"-windows-amd64.zip"
    $MVN_CMD = "mvnd.cmd"
    break
  }
  default {
    $USE_MVND = $false
    $MVN_CMD = $script -replace '^mvnw','mvn'
```

```
    break
  }
}
# apply MVNW_REPOURL and calculate MAVEN_HOME
# maven home pattern: ~/.m2/wrapper/dists/{apache-maven-<version>,maven-mvnd-<version>-<platform>}/<hash>
if ($env:MVNW_REPOURL) {
  $MVNW_REPO_PATTERN = if ($USE_MVND) { "/org/apache/maven/" } else { "/maven/mvnd/" }
  $distributionUrl = "$env:MVNW_REPOURL$MVNW_REPO_PATTERN$($distributionUrl -replace '^.*'+$MVNW_REPO_PATTERN,")"
}
$distributionUrlName = $distributionUrl -replace '^.*/',"
$distributionUrlNameMain = $distributionUrlName -replace '\.[^.]*$'," -replace '-bin$',"
$MAVEN_HOME_PARENT = "$HOME/.m2/wrapper/dists/$distributionUrlNameMain"
if ($env:MAVEN_USER_HOME) {
  $MAVEN_HOME_PARENT = "$env:MAVEN_USER_HOME/wrapper/dists/$distributionUrlNameMain"
}
$MAVEN_HOME_NAME = ([System.Security.Cryptography.MD5]::Create().ComputeHash([byte[]][char[]]$distributionUrl) | ForEach-Object {$_.ToString("x2")}) -join ''
$MAVEN_HOME = "$MAVEN_HOME_PARENT/$MAVEN_HOME_NAME"
if (Test-Path -Path "$MAVEN_HOME" -PathType Container) {
  Write-Verbose "found existing MAVEN_HOME at $MAVEN_HOME"
  Write-Output "MVN_CMD=$MAVEN_HOME/bin/$MVN_CMD"
  exit $?
}
if (! $distributionUrlNameMain -or ($distributionUrlName -eq $distributionUrlNameMain)) {
  Write-Error "distributionUrl is not valid, must end with *-bin.zip, but found $distributionUrl"
}
```

```
# prepare tmp dir
$TMP_DOWNLOAD_DIR_HOLDER = New-TemporaryFile
$TMP_DOWNLOAD_DIR = New-Item -Itemtype Directory -Path
"$TMP_DOWNLOAD_DIR_HOLDER.dir"
$TMP_DOWNLOAD_DIR_HOLDER.Delete() | Out-Null
trap {
  if ($TMP_DOWNLOAD_DIR.Exists) {
    try { Remove-Item $TMP_DOWNLOAD_DIR -Recurse -Force | Out-Null }
    catch { Write-Warning "Cannot remove $TMP_DOWNLOAD_DIR" }
  }
}
New-Item -Itemtype Directory -Path "$MAVEN_HOME_PARENT" -Force | Out-Null
# Download and Install Apache Maven
Write-Verbose "Couldn't find MAVEN_HOME, downloading and installing it ..."
Write-Verbose "Downloading from: $distributionUrl"
Write-Verbose "Downloading to: $TMP_DOWNLOAD_DIR/$distributionUrlName"
$webclient = New-Object System.Net.WebClient
if ($env:MVNW_USERNAME -and $env:MVNW_PASSWORD) {
  $webclient.Credentials = New-Object
System.Net.NetworkCredential($env:MVNW_USERNAME, $env:MVNW_PASSWORD)
}
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
$webclient.DownloadFile($distributionUrl,
"$TMP_DOWNLOAD_DIR/$distributionUrlName") | Out-Null
# If specified, validate the SHA-256 sum of the Maven distribution zip file
$distributionSha256Sum = (Get-Content -Raw "$scriptDir/.mvn/wrapper/maven-
wrapper.properties" | ConvertFrom-StringData).distributionSha256Sum
if ($distributionSha256Sum) {
  if ($USE_MVND) {
    Write-Error "Checksum validation is not supported for maven-mvnd. `nPlease disable
validation by removing 'distributionSha256Sum' from your maven-wrapper.properties."
  }
```

```
  Import-Module $PSHOME\Modules\Microsoft.PowerShell.Utility -Function Get-FileHash
  if ((Get-FileHash "$TMP_DOWNLOAD_DIR/$distributionUrlName" -Algorithm SHA256).Hash.ToLower() -ne $distributionSha256Sum) {
    Write-Error "Error: Failed to validate Maven distribution SHA-256, your Maven distribution might be compromised. If you updated your Maven version, you need to update the specified distributionSha256Sum property."
  }
}
# unzip and move
Expand-Archive "$TMP_DOWNLOAD_DIR/$distributionUrlName" -DestinationPath "$TMP_DOWNLOAD_DIR" | Out-Null
Rename-Item -Path "$TMP_DOWNLOAD_DIR/$distributionUrlNameMain" -NewName $MAVEN_HOME_NAME | Out-Null
try {
  Move-Item -Path "$TMP_DOWNLOAD_DIR/$MAVEN_HOME_NAME" -Destination $MAVEN_HOME_PARENT | Out-Null
} catch {
  if (! (Test-Path -Path "$MAVEN_HOME" -PathType Container)) {
    Write-Error "fail to move MAVEN_HOME"
  }
} finally {
  try { Remove-Item $TMP_DOWNLOAD_DIR -Recurse -Force | Out-Null }
  catch { Write-Warning "Cannot remove $TMP_DOWNLOAD_DIR" }
}
Write-Output "MVN_CMD=$MAVEN_HOME/bin/$MVN_CMD"
```

//mvnw

```
#!/bin/sh
# ----------------------------------------------------------------------
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
```

```
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied.  See the License for the
# specific language governing permissions and limitations
# under the License.
# ----------------------------------------------------------
# ----------------------------------------------------------
# Apache Maven Wrapper startup batch script, version 3.3.2
#
# Optional ENV vars
# -----------
#   JAVA_HOME - location of a JDK home dir, required when download maven via java source
#   MVNW_REPOURL - repo url base for downloading maven distribution
#   MVNW_USERNAME/MVNW_PASSWORD - user and password for downloading maven
#   MVNW_VERBOSE - true: enable verbose log; debug: trace the mvnw script; others: silence the output
# ----------------------------------------------------------
set -euf
[ "${MVNW_VERBOSE-}" != debug ] || set -x
# OS specific support.
native_path() { printf %s\\n "$1"; }
case "$(uname)" in
```

```
CYGWIN* | MINGW*)
  [ -z "${JAVA_HOME-}" ] || JAVA_HOME="$(cygpath --unix "$JAVA_HOME")"
  native_path() { cygpath --path --windows "$1"; }
  ;;
esac
# set JAVACMD and JAVACCMD
set_java_home() {
  # For Cygwin and MinGW, ensure paths are in Unix format before anything is touched
  if [ -n "${JAVA_HOME-}" ]; then
    if [ -x "$JAVA_HOME/jre/sh/java" ]; then
      # IBM's JDK on AIX uses strange locations for the executables
      JAVACMD="$JAVA_HOME/jre/sh/java"
      JAVACCMD="$JAVA_HOME/jre/sh/javac"
    else
      JAVACMD="$JAVA_HOME/bin/java"
      JAVACCMD="$JAVA_HOME/bin/javac"
      if [ ! -x "$JAVACMD" ] || [ ! -x "$JAVACCMD" ]; then
        echo "The JAVA_HOME environment variable is not defined correctly, so mvnw cannot
run." >&2
        echo "JAVA_HOME is set to \"$JAVA_HOME\", but \"\$JAVA_HOME/bin/java\" or
\"\$JAVA_HOME/bin/javac\" does not exist." >&2
        return 1
      fi
  fi
  else
    JAVACMD="$(
      'set' +e
      'unset' -f command 2>/dev/null
      'command' -v java
    )" || :
    JAVACCMD="$(
```

```
    'set' +e
    'unset' -f command 2>/dev/null
    'command' -v javac
  )" || :
  if [ ! -x "${JAVACMD-}" ] || [ ! -x "${JAVACCMD-}" ]; then
    echo "The java/javac command does not exist in PATH nor is JAVA_HOME set, so mvnw cannot run." >&2
    return 1
  fi
 fi
}
# hash string like Java String::hashCode
hash_string() {
 str="${1:-}" h=0
 while [ -n "$str" ]; do
  char="${str%"${str#?}"}"
  h=$(((h * 31 + $(LC_CTYPE=C printf %d "'$char")) % 4294967296))
  str="${str#?}"
 done
 printf %x\\n $h
}
verbose() { :; }
[ "${MVNW_VERBOSE-}" != true ] || verbose() { printf %s\\n "${1-}"; }
die() {
 printf %s\\n "$1" >&2
 exit 1
}
trim() {
 # MWRAPPER-139:
 #   Trims trailing and leading whitespace, carriage returns, tabs, and linefeeds.
 #   Needed for removing poorly interpreted newline sequences when running in more
```

```
  #   exotic environments such as mingw bash on Windows.
  printf "%s" "${1}" | tr -d '[:space:]'
}
# parse distributionUrl and optional distributionSha256Sum, requires .mvn/wrapper/maven-wrapper.properties
while IFS="=" read -r key value; do
  case "${key-}" in
  distributionUrl) distributionUrl=$(trim "${value-}") ;;
  distributionSha256Sum) distributionSha256Sum=$(trim "${value-}") ;;
  esac
done <"${0%/*}/.mvn/wrapper/maven-wrapper.properties"
[ -n "${distributionUrl-}" ] || die "cannot read distributionUrl property in ${0%/*}/.mvn/wrapper/maven-wrapper.properties"
case "${distributionUrl##*/}" in
maven-mvnd-bin.)
  MVN_CMD=mvnd.sh _MVNW_REPO_PATTERN=/maven/mvnd/
  case  "${PROCESSOR_ARCHITECTURE-}${PROCESSOR_ARCHITEW6432-}:$(uname -a)" in
  AMD64:CYGWIN | AMD64:MINGW) distributionPlatform=windows-amd64 ;;
  :Darwin*x86_64) distributionPlatform=darwin-amd64 ;;
  :Darwin*arm64) distributionPlatform=darwin-aarch64 ;;
  :Linux*x86_64*) distributionPlatform=linux-amd64 ;;
  *)
    echo "Cannot detect native platform for mvnd on $(uname)-$(uname -m), use pure java version" >&2
    distributionPlatform=linux-amd64
    ;;
  esac
  distributionUrl="${distributionUrl%-bin.*}-$distributionPlatform.zip"
  ;;
maven-mvnd-*) MVN_CMD=mvnd.sh _MVNW_REPO_PATTERN=/maven/mvnd/ ;;
) MVN_CMD="mvn${0##/mvnw}" _MVNW_REPO_PATTERN=/org/apache/maven/ ;;
```

```
esac

# apply MVNW_REPOURL and calculate MAVEN_HOME

# maven home pattern: ~/.m2/wrapper/dists/{apache-maven-<version>,maven-mvnd-<version>-<platform>}/<hash>

[ -z "${MVNW_REPOURL-}" ] || distributionUrl="$MVNW_REPOURL$_MVNW_REPO_PATTERN${distributionUrl#*"$_MVNW_REPO_PATTERN"}"

distributionUrlName="${distributionUrl##*/}"

distributionUrlNameMain="${distributionUrlName%.*}"

distributionUrlNameMain="${distributionUrlNameMain%-bin}"

MAVEN_USER_HOME="${MAVEN_USER_HOME:-${HOME}/.m2}"

MAVEN_HOME="${MAVEN_USER_HOME}/wrapper/dists/${distributionUrlNameMain-}/$(hash_string "$distributionUrl")"

exec_maven() {

  unset MVNW_VERBOSE MVNW_USERNAME MVNW_PASSWORD MVNW_REPOURL || :

  exec "$MAVEN_HOME/bin/$MVN_CMD" "$@" || die "cannot exec $MAVEN_HOME/bin/$MVN_CMD"

}

if [ -d "$MAVEN_HOME" ]; then

  verbose "found existing MAVEN_HOME at $MAVEN_HOME"

  exec_maven "$@"

fi

case "${distributionUrl-}" in

?-bin.zip | *?maven-mvnd-?-?*.zip) ;;

) die "distributionUrl is not valid, must match *-bin.zip or maven-mvnd-.zip, but found '${distributionUrl-}'" ;;

esac

# prepare tmp dir

if TMP_DOWNLOAD_DIR="$(mktemp -d)" && [ -d "$TMP_DOWNLOAD_DIR" ]; then

  clean() { rm -rf -- "$TMP_DOWNLOAD_DIR"; }

  trap clean HUP INT TERM EXIT

else
```

```
  die "cannot create temp dir"
fi
mkdir -p -- "${MAVEN_HOME%/*}"
# Download and Install Apache Maven
verbose "Couldn't find MAVEN_HOME, downloading and installing it ..."
verbose "Downloading from: $distributionUrl"
verbose "Downloading to: $TMP_DOWNLOAD_DIR/$distributionUrlName"
# select .zip or .tar.gz
if ! command -v unzip >/dev/null; then
  distributionUrl="${distributionUrl%.zip}.tar.gz"
  distributionUrlName="${distributionUrl##*/}"
fi
# verbose opt
__MVNW_QUIET_WGET=--quiet __MVNW_QUIET_CURL=--silent
__MVNW_QUIET_UNZIP=-q __MVNW_QUIET_TAR="
[ "${MVNW_VERBOSE-}" != true ] || __MVNW_QUIET_WGET="
__MVNW_QUIET_CURL=" __MVNW_QUIET_UNZIP=" __MVNW_QUIET_TAR=v
# normalize http auth
case "${MVNW_PASSWORD:+has-password}" in
") MVNW_USERNAME=" MVNW_PASSWORD=" ;;
has-password) [ -n "${MVNW_USERNAME-}" ] || MVNW_USERNAME="
MVNW_PASSWORD=" ;;
esac
if [ -z "${MVNW_USERNAME-}" ] && command -v wget >/dev/null; then
  verbose "Found wget ... using wget"
  wget ${_MVNW_QUIET_WGET:+"$_MVNW_QUIET_WGET"} "$distributionUrl" -O
"$TMP_DOWNLOAD_DIR/$distributionUrlName" || die "wget: Failed to fetch
$distributionUrl"
elif [ -z "${MVNW_USERNAME-}" ] && command -v curl >/dev/null; then
  verbose "Found curl ... using curl"
  curl ${_MVNW_QUIET_CURL:+"$_MVNW_QUIET_CURL"} -f -L -o
"$TMP_DOWNLOAD_DIR/$distributionUrlName" "$distributionUrl" || die "curl: Failed to
fetch $distributionUrl"
```

```
elif set_java_home; then
  verbose "Falling back to use Java to download"
  javaSource="$TMP_DOWNLOAD_DIR/Downloader.java"
  targetZip="$TMP_DOWNLOAD_DIR/$distributionUrlName"
  cat >"$javaSource" <<-END
          public class Downloader extends java.net.Authenticator
          {
            protected java.net.PasswordAuthentication getPasswordAuthentication()
            {
              return new java.net.PasswordAuthentication( System.getenv(
"MVNW_USERNAME" ), System.getenv( "MVNW_PASSWORD" ).toCharArray() );
            }
            public static void main( String[] args ) throws Exception
            {
              setDefault( new Downloader() );
              java.nio.file.Files.copy( java.net.URI.create( args[0] ).toURL().openStream(),
java.nio.file.Paths.get( args[1] ).toAbsolutePath().normalize() );
            }
          }
          END
  # For Cygwin/MinGW, switch paths to Windows format before running javac and java
  verbose " - Compiling Downloader.java ..."
  "$(native_path "$JAVACCMD")" "$(native_path "$javaSource")" || die "Failed to compile
Downloader.java"
  verbose " - Running Downloader.java ..."
  "$(native_path "$JAVACMD")" -cp "$(native_path "$TMP_DOWNLOAD_DIR")"
Downloader "$distributionUrl" "$(native_path "$targetZip")"
fi
# If specified, validate the SHA-256 sum of the Maven distribution zip file
if [ -n "${distributionSha256Sum-}" ]; then
distributionSha256Result=false
  if [ "$MVN_CMD" = mvnd.sh ]; then
```

```
    echo "Checksum validation is not supported for maven-mvnd." >&2
    echo "Please disable validation by removing 'distributionSha256Sum' from your maven-wrapper.properties." >&2
    exit 1
  elif command -v sha256sum >/dev/null; then
    if echo "$distributionSha256Sum  $TMP_DOWNLOAD_DIR/$distributionUrlName" | sha256sum -c >/dev/null 2>&1; then
      distributionSha256Result=true
    fi
  elif command -v shasum >/dev/null; then
    if echo "$distributionSha256Sum  $TMP_DOWNLOAD_DIR/$distributionUrlName" | shasum -a 256 -c >/dev/null 2>&1; then
      distributionSha256Result=true
    fi
  else
    echo "Checksum validation was requested but neither 'sha256sum' or 'shasum' are available." >&2
    echo "Please install either command, or disable validation by removing 'distributionSha256Sum' from your maven-wrapper.properties." >&2
    exit 1
  fi
  if [ $distributionSha256Result = false ]; then
    echo "Error: Failed to validate Maven distribution SHA-256, your Maven distribution might be compromised." >&2
    echo "If you updated your Maven version, you need to update the specified distributionSha256Sum property." >&2
    exit 1
  fi
fi

# unzip and move
if command -v unzip >/dev/null; then
  unzip ${_MVNW_QUIET_UNZIP:+"$_MVNW_QUIET_UNZIP"} "$TMP_DOWNLOAD_DIR/$distributionUrlName" -d "$TMP_DOWNLOAD_DIR" || die "failed to unzip"
```

```
else
  tar xzf${_MVNW_QUIET_TAR:+"$_MVNW_QUIET_TAR"} "$TMP_DOWNLOAD_DIR/$distributionUrlName" -C "$TMP_DOWNLOAD_DIR" || die "failed to untar"
fi
printf %s\\n "$distributionUrl" >"$TMP_DOWNLOAD_DIR/$distributionUrlNameMain/mvnw.url"
mv -- "$TMP_DOWNLOAD_DIR/$distributionUrlNameMain" "$MAVEN_HOME" || [ -d "$MAVEN_HOME" ] || die "fail to move MAVEN_HOME"
clean || :
exec_maven "$@"
```

//.gitignore

```
target/
.mvn/wrapper/maven-wrapper.jar
!/src/main//target/
!/src/test//target/

### STS ###
.apt_generated
.classpath
.factorypath
.project
.settings
.springBeans
.sts4-cache

### IntelliJ IDEA ###
.idea
*.iws
*.iml
*.ipr
```

NetBeans

/nbproject/private/

/nbbuild/

/dist/

/nbdist/

/.nb-gradle/

build/

!/src/main//build/

!/src/test//build/

VS Code

.vscode/

Register Microservices

//pom.xml

```
<dependencies>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
</dependencies>
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-dependencies</artifactId>
            <version>2023.0.1</version>
```

```
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
```

//mvnw.cmd

```
<# : batch portion
@REM ----------------------------------------------------------------------------
@REM Licensed to the Apache Software Foundation (ASF) under one
@REM or more contributor license agreements.  See the NOTICE file
@REM distributed with this work for additional information
@REM regarding copyright ownership.  The ASF licenses this file
@REM to you under the Apache License, Version 2.0 (the
@REM "License"); you may not use this file except in compliance
@REM with the License.  You may obtain a copy of the License at
@REM
@REM    http://www.apache.org/licenses/LICENSE-2.0
@REM
@REM Unless required by applicable law or agreed to in writing,
@REM software distributed under the License is distributed on an
@REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
@REM KIND, either express or implied.  See the License for the
@REM specific language governing permissions and limitations
@REM under the License.
@REM ----------------------------------------------------------------------------
@REM ----------------------------------------------------------------------------
@REM Apache Maven Wrapper startup batch script, version 3.3.2
@REM
@REM Optional ENV vars
@REM   MVNW_REPOURL - repo url base for downloading maven distribution
```

```
@REM  MVNW_USERNAME/MVNW_PASSWORD - user and password for
downloading maven

@REM  MVNW_VERBOSE - true: enable verbose log; others: silence the output

@REM ----------------------------------------------------

@IF "%_MVNW_ARG0_NAME%"=="" (SET __MVNW_ARG0_NAME_=%~nx0)

@SET _MVNW_CMD_=

@SET _MVNW_ERROR_=

@SET __MVNW_PSMODULEP_SAVE=%PSModulePath%

@SET PSModulePath=

@FOR /F "usebackq tokens=1* delims==" %%A IN (powershell -noprofile "&
{$scriptDir='%~dp0'; $script='%__MVNW_ARG0_NAME__%'; icm -ScriptBlock
([Scriptblock]::Create((Get-Content -Raw '%~f0'))) -NoNewScope}") DO @(

  IF "%%A"=="MVN_CMD" (set _MVNW_CMD_=%%B) ELSE IF "%%B"=="" (echo
%%A) ELSE (echo %%A=%%B)

)

@SET PSModulePath=%__MVNW_PSMODULEP_SAVE%

@SET __MVNW_PSMODULEP_SAVE=

@SET _MVNW_ARG0_NAME_=

@SET MVNW_USERNAME=

@SET MVNW_PASSWORD=

@IF NOT "%_MVNW_CMD%"=="" (%MVNW_CMD_% %*)

@echo Cannot start maven from wrapper >&2 && exit /b 1

@GOTO :EOF

: end batch / begin powershell #>

$ErrorActionPreference = "Stop"

if ($env:MVNW_VERBOSE -eq "true") {

  $VerbosePreference = "Continue"

}

# calculate distributionUrl, requires .mvn/wrapper/maven-wrapper.properties

$distributionUrl = (Get-Content -Raw "$scriptDir/.mvn/wrapper/maven-wrapper.properties" |
ConvertFrom-StringData).distributionUrl

if (!$distributionUrl) {
```

```
  Write-Error "cannot read distributionUrl property in $scriptDir/.mvn/wrapper/maven-wrapper.properties"
}
switch -wildcard -casesensitive ( $($distributionUrl -replace '^.*/','') ) {
  "maven-mvnd-*" {
    $USE_MVND = $true
    $distributionUrl = $distributionUrl -replace '-bin\.[^.]*$',"-windows-amd64.zip"
    $MVN_CMD = "mvnd.cmd"
    break
  }
  default {
    $USE_MVND = $false
    $MVN_CMD = $script -replace '^mvnw','mvn'
    break
  }
}
# apply MVNW_REPOURL and calculate MAVEN_HOME
# maven home pattern: ~/.m2/wrapper/dists/{apache-maven-<version>,maven-mvnd-<version>-<platform>}/<hash>
if ($env:MVNW_REPOURL) {
  $MVNW_REPO_PATTERN = if ($USE_MVND) { "/org/apache/maven/" } else { "/maven/mvnd/" }
  $distributionUrl = "$env:MVNW_REPOURL$MVNW_REPO_PATTERN$($distributionUrl -replace '^.*'+$MVNW_REPO_PATTERN,")"
}
$distributionUrlName = $distributionUrl -replace '^.*/',"
$distributionUrlNameMain = $distributionUrlName -replace '\.[^.]*$'," -replace '-bin$',"
$MAVEN_HOME_PARENT = "$HOME/.m2/wrapper/dists/$distributionUrlNameMain"
if ($env:MAVEN_USER_HOME) {
  $MAVEN_HOME_PARENT = "$env:MAVEN_USER_HOME/wrapper/dists/$distributionUrlNameMain"
```

```
}
$MAVEN_HOME_NAME =
([System.Security.Cryptography.MD5]::Create().ComputeHash([byte[]][char[]]$distribution
Url) | ForEach-Object {$_.ToString("x2")}) -join ''
$MAVEN_HOME = "$MAVEN_HOME_PARENT/$MAVEN_HOME_NAME"
if (Test-Path -Path "$MAVEN_HOME" -PathType Container) {
  Write-Verbose "found existing MAVEN_HOME at $MAVEN_HOME"
  Write-Output "MVN_CMD=$MAVEN_HOME/bin/$MVN_CMD"
  exit $?
}
if (! $distributionUrlNameMain -or ($distributionUrlName -eq $distributionUrlNameMain))
{
  Write-Error "distributionUrl is not valid, must end with *-bin.zip, but found
$distributionUrl"
}
# prepare tmp dir
$TMP_DOWNLOAD_DIR_HOLDER = New-TemporaryFile
$TMP_DOWNLOAD_DIR = New-Item -Itemtype Directory -Path
"$TMP_DOWNLOAD_DIR_HOLDER.dir"
$TMP_DOWNLOAD_DIR_HOLDER.Delete() | Out-Null
trap {
  if ($TMP_DOWNLOAD_DIR.Exists) {
    try { Remove-Item $TMP_DOWNLOAD_DIR -Recurse -Force | Out-Null }
    catch { Write-Warning "Cannot remove $TMP_DOWNLOAD_DIR" }
  }
}
New-Item -Itemtype Directory -Path "$MAVEN_HOME_PARENT" -Force | Out-Null
# Download and Install Apache Maven
Write-Verbose "Couldn't find MAVEN_HOME, downloading and installing it ..."
Write-Verbose "Downloading from: $distributionUrl"
Write-Verbose "Downloading to: $TMP_DOWNLOAD_DIR/$distributionUrlName"
$webclient = New-Object System.Net.WebClient
```

```
if ($env:MVNW_USERNAME -and $env:MVNW_PASSWORD) {
  $webclient.Credentials = New-Object System.Net.NetworkCredential($env:MVNW_USERNAME, $env:MVNW_PASSWORD)
}
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
$webclient.DownloadFile($distributionUrl, "$TMP_DOWNLOAD_DIR/$distributionUrlName") | Out-Null
# If specified, validate the SHA-256 sum of the Maven distribution zip file
$distributionSha256Sum = (Get-Content -Raw "$scriptDir/.mvn/wrapper/maven-wrapper.properties" | ConvertFrom-StringData).distributionSha256Sum
if ($distributionSha256Sum) {
  if ($USE_MVND) {
    Write-Error "Checksum validation is not supported for maven-mvnd. `nPlease disable validation by removing 'distributionSha256Sum' from your maven-wrapper.properties."
  }
  Import-Module $PSHOME\Modules\Microsoft.PowerShell.Utility -Function Get-FileHash
  if ((Get-FileHash "$TMP_DOWNLOAD_DIR/$distributionUrlName" -Algorithm SHA256).Hash.ToLower() -ne $distributionSha256Sum) {
    Write-Error "Error: Failed to validate Maven distribution SHA-256, your Maven distribution might be compromised. If you updated your Maven version, you need to update the specified distributionSha256Sum property."
  }
}
# unzip and move
Expand-Archive "$TMP_DOWNLOAD_DIR/$distributionUrlName" -DestinationPath "$TMP_DOWNLOAD_DIR" | Out-Null
Rename-Item -Path "$TMP_DOWNLOAD_DIR/$distributionUrlNameMain" -NewName $MAVEN_HOME_NAME | Out-Null
try {
  Move-Item -Path "$TMP_DOWNLOAD_DIR/$MAVEN_HOME_NAME" -Destination $MAVEN_HOME_PARENT | Out-Null
} catch {
  if (! (Test-Path -Path "$MAVEN_HOME" -PathType Container)) {
    Write-Error "fail to move MAVEN_HOME"
```

```
  }
} finally {
  try { Remove-Item $TMP_DOWNLOAD_DIR -Recurse -Force | Out-Null }
  catch { Write-Warning "Cannot remove $TMP_DOWNLOAD_DIR" }
}
Write-Output "MVN_CMD=$MAVEN_HOME/bin/$MVN_CMD"
```

//mvnw

```
#!/bin/sh
# ----------------------------------------------------------------------------
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied.  See the License for the
# specific language governing permissions and limitations
# under the License.
# ----------------------------------------------------------------------------
# ----------------------------------------------------------------------------
# Apache Maven Wrapper startup batch script, version 3.3.2
#
# Optional ENV vars
```

```
# ----------
#   JAVA_HOME - location of a JDK home dir, required when download maven via java source
#   MVNW_REPOURL - repo url base for downloading maven distribution
#   MVNW_USERNAME/MVNW_PASSWORD - user and password for downloading maven
#   MVNW_VERBOSE - true: enable verbose log; debug: trace the mvnw script; others: silence the output
# ----------
set -euf
[ "${MVNW_VERBOSE-}" != debug ] || set -x
# OS specific support.
native_path() { printf %s\\n "$1"; }
case "$(uname)" in
CYGWIN* | MINGW*)
  [ -z "${JAVA_HOME-}" ] || JAVA_HOME="$(cygpath --unix "$JAVA_HOME")"
  native_path() { cygpath --path --windows "$1"; }
  ;;
esac
# set JAVACMD and JAVACCMD
set_java_home() {
  # For Cygwin and MinGW, ensure paths are in Unix format before anything is touched
  if [ -n "${JAVA_HOME-}" ]; then
    if [ -x "$JAVA_HOME/jre/sh/java" ]; then
      # IBM's JDK on AIX uses strange locations for the executables
      JAVACMD="$JAVA_HOME/jre/sh/java"
      JAVACCMD="$JAVA_HOME/jre/sh/javac"
    else
      JAVACMD="$JAVA_HOME/bin/java"
      JAVACCMD="$JAVA_HOME/bin/javac"
      if [ ! -x "$JAVACMD" ] || [ ! -x "$JAVACCMD" ]; then
```

```
      echo "The JAVA_HOME environment variable is not defined correctly, so mvnw cannot run." >&2
      echo "JAVA_HOME is set to \"$JAVA_HOME\", but \"\$JAVA_HOME/bin/java\" or \"\$JAVA_HOME/bin/javac\" does not exist." >&2
      return 1
    fi
  fi
  else
    JAVACMD="$(
      'set' +e
      'unset' -f command 2>/dev/null
      'command' -v java
    )" || :
    JAVACCMD="$(
      'set' +e
      'unset' -f command 2>/dev/null
      'command' -v javac
    )" || :
    if [ ! -x "${JAVACMD-}" ] || [ ! -x "${JAVACCMD-}" ]; then
      echo "The java/javac command does not exist in PATH nor is JAVA_HOME set, so mvnw cannot run." >&2
      return 1
    fi
  fi
}
# hash string like Java String::hashCode
hash_string() {
  str="${1:-}" h=0
  while [ -n "$str" ]; do
    char="${str%"${str#?}"}"
    h=$(((h * 31 + $(LC_CTYPE=C printf %d "'$char")) % 4294967296))
```

```
    str="${str#?}"
  done
  printf %x\\n $h
}
verbose() { :; }
[ "${MVNW_VERBOSE-}" != true ] || verbose() { printf %s\\n "${1-}"; }
die() {
  printf %s\\n "$1" >&2
  exit 1
}
trim() {
  # MWRAPPER-139:
  #   Trims trailing and leading whitespace, carriage returns, tabs, and linefeeds.
  #   Needed for removing poorly interpreted newline sequences when running in more
  #   exotic environments such as mingw bash on Windows.
  printf "%s" "${1}" | tr -d '[:space:]'
}
# parse distributionUrl and optional distributionSha256Sum, requires .mvn/wrapper/maven-wrapper.properties
while IFS="=" read -r key value; do
  case "${key-}" in
  distributionUrl) distributionUrl=$(trim "${value-}") ;;
  distributionSha256Sum) distributionSha256Sum=$(trim "${value-}") ;;
  esac
done <"${0%/*}/.mvn/wrapper/maven-wrapper.properties"
[ -n "${distributionUrl-}" ] || die "cannot read distributionUrl property in ${0%/*}/.mvn/wrapper/maven-wrapper.properties"
case "${distributionUrl##*/}" in
maven-mvnd-bin.)
  MVN_CMD=mvnd.sh _MVNW_REPO_PATTERN=/maven/mvnd/
```

```
  case  "${PROCESSOR_ARCHITECTURE-}${PROCESSOR_ARCHITEW6432-}:$(uname -a)" in
  AMD64:CYGWIN | AMD64:MINGW) distributionPlatform=windows-amd64 ;;
  :Darwin*x86_64) distributionPlatform=darwin-amd64 ;;
  :Darwin*arm64) distributionPlatform=darwin-aarch64 ;;
  :Linux*x86_64*) distributionPlatform=linux-amd64 ;;
  *)
    echo "Cannot detect native platform for mvnd on $(uname)-$(uname -m), use pure java version" >&2
    distributionPlatform=linux-amd64
    ;;
  esac
  distributionUrl="${distributionUrl%-bin.*}-$distributionPlatform.zip"
  ;;
maven-mvnd-*) MVN_CMD=mvnd.sh _MVNW_REPO_PATTERN=/maven/mvnd/ ;;
) MVN_CMD="mvn${0##/mvnw}" _MVNW_REPO_PATTERN=/org/apache/maven/ ;;
esac
# apply MVNW_REPOURL and calculate MAVEN_HOME
# maven home pattern: ~/.m2/wrapper/dists/{apache-maven-<version>,maven-mvnd-<version>-<platform>}/<hash>
[ -z "${MVNW_REPOURL-}" ] || distributionUrl="$MVNW_REPOURL$_MVNW_REPO_PATTERN${distributionUrl#*"$_MVNW_REPO_PATTERN"}"
distributionUrlName="${distributionUrl##*/}"
distributionUrlNameMain="${distributionUrlName%.*}"
distributionUrlNameMain="${distributionUrlNameMain%-bin}"
MAVEN_USER_HOME="${MAVEN_USER_HOME:-${HOME}/.m2}"
MAVEN_HOME="${MAVEN_USER_HOME}/wrapper/dists/${distributionUrlNameMain-}/$(hash_string "$distributionUrl")"
exec_maven() {
  unset MVNW_VERBOSE MVNW_USERNAME MVNW_PASSWORD MVNW_REPOURL || :
```

```
  exec "$MAVEN_HOME/bin/$MVN_CMD" "$@" || die "cannot exec
$MAVEN_HOME/bin/$MVN_CMD"
}
if [ -d "$MAVEN_HOME" ]; then
  verbose "found existing MAVEN_HOME at $MAVEN_HOME"
  exec_maven "$@"
fi
case "${distributionUrl-}" in
?-bin.zip | *?maven-mvnd-?-?*.zip) ;;
) die "distributionUrl is not valid, must match *-bin.zip or maven-mvnd-.zip, but found
'${distributionUrl-}'" ;;
esac
# prepare tmp dir
if TMP_DOWNLOAD_DIR="$(mktemp -d)" && [ -d "$TMP_DOWNLOAD_DIR" ]; then
  clean() { rm -rf -- "$TMP_DOWNLOAD_DIR"; }
  trap clean HUP INT TERM EXIT
else
  die "cannot create temp dir"
fi
mkdir -p -- "${MAVEN_HOME%/*}"
# Download and Install Apache Maven
verbose "Couldn't find MAVEN_HOME, downloading and installing it ..."
verbose "Downloading from: $distributionUrl"
verbose "Downloading to: $TMP_DOWNLOAD_DIR/$distributionUrlName"
# select .zip or .tar.gz
if ! command -v unzip >/dev/null; then
  distributionUrl="${distributionUrl%.zip}.tar.gz"
  distributionUrlName="${distributionUrl##*/}"
fi
# verbose opt
```

```
__MVNW_QUIET_WGET=--quiet __MVNW_QUIET_CURL=--silent
__MVNW_QUIET_UNZIP=-q __MVNW_QUIET_TAR="

[ "${MVNW_VERBOSE-}" != true ] || __MVNW_QUIET_WGET="
__MVNW_QUIET_CURL=" __MVNW_QUIET_UNZIP=" __MVNW_QUIET_TAR=v

# normalize http auth

case "${MVNW_PASSWORD:+has-password}" in

") MVNW_USERNAME=" MVNW_PASSWORD=" ;;

has-password) [ -n "${MVNW_USERNAME-}" ] || MVNW_USERNAME="
MVNW_PASSWORD=" ;;

esac

if [ -z "${MVNW_USERNAME-}" ] && command -v wget >/dev/null; then

  verbose "Found wget ... using wget"

  wget ${_MVNW_QUIET_WGET:+"$_MVNW_QUIET_WGET"} "$distributionUrl" -O
"$TMP_DOWNLOAD_DIR/$distributionUrlName" || die "wget: Failed to fetch
$distributionUrl"

elif [ -z "${MVNW_USERNAME-}" ] && command -v curl >/dev/null; then

  verbose "Found curl ... using curl"

  curl ${_MVNW_QUIET_CURL:+"$_MVNW_QUIET_CURL"} -f -L -o
"$TMP_DOWNLOAD_DIR/$distributionUrlName" "$distributionUrl" || die "curl: Failed to
fetch $distributionUrl"

elif set_java_home; then

  verbose "Falling back to use Java to download"

  javaSource="$TMP_DOWNLOAD_DIR/Downloader.java"

  targetZip="$TMP_DOWNLOAD_DIR/$distributionUrlName"

  cat >"$javaSource" <<-END

        public class Downloader extends java.net.Authenticator

        {

          protected java.net.PasswordAuthentication getPasswordAuthentication()

          {

            return new java.net.PasswordAuthentication( System.getenv(
"MVNW_USERNAME" ), System.getenv( "MVNW_PASSWORD" ).toCharArray() );

          }

          public static void main( String[] args ) throws Exception
```

```
            {
              setDefault( new Downloader() );
              java.nio.file.Files.copy( java.net.URI.create( args[0] ).toURL().openStream(), java.nio.file.Paths.get( args[1] ).toAbsolutePath().normalize() );
            }
          }
          END
  # For Cygwin/MinGW, switch paths to Windows format before running javac and java
  verbose " - Compiling Downloader.java ..."
  "$(native_path "$JAVACCMD")" "$(native_path "$javaSource")" || die "Failed to compile Downloader.java"
  verbose " - Running Downloader.java ..."
  "$(native_path "$JAVACMD")" -cp "$(native_path "$TMP_DOWNLOAD_DIR")" Downloader "$distributionUrl" "$(native_path "$targetZip")"
fi
# If specified, validate the SHA-256 sum of the Maven distribution zip file
if [ -n "${distributionSha256Sum-}" ]; then
distributionSha256Result=false
  if [ "$MVN_CMD" = mvnd.sh ]; then
    echo "Checksum validation is not supported for maven-mvnd." >&2
    echo "Please disable validation by removing 'distributionSha256Sum' from your maven-wrapper.properties." >&2
    exit 1
  elif command -v sha256sum >/dev/null; then
    if echo "$distributionSha256Sum  $TMP_DOWNLOAD_DIR/$distributionUrlName" | sha256sum -c >/dev/null 2>&1; then
      distributionSha256Result=true
    fi
  elif command -v shasum >/dev/null; then
    if echo "$distributionSha256Sum  $TMP_DOWNLOAD_DIR/$distributionUrlName" | shasum -a 256 -c >/dev/null 2>&1; then
      distributionSha256Result=true
```

```
    fi
  else
    echo "Checksum validation was requested but neither 'sha256sum' or 'shasum' are available." >&2
    echo "Please install either command, or disable validation by removing 'distributionSha256Sum' from your maven-wrapper.properties." >&2
    exit 1
  fi
  if [ $distributionSha256Result = false ]; then
    echo "Error: Failed to validate Maven distribution SHA-256, your Maven distribution might be compromised." >&2
    echo "If you updated your Maven version, you need to update the specified distributionSha256Sum property." >&2
    exit 1
  fi
fi

# unzip and move
if command -v unzip >/dev/null; then
  unzip ${_MVNW_QUIET_UNZIP:+"$_MVNW_QUIET_UNZIP"} "$TMP_DOWNLOAD_DIR/$distributionUrlName" -d "$TMP_DOWNLOAD_DIR" || die "failed to unzip"
else
  tar xzf${_MVNW_QUIET_TAR:+"$_MVNW_QUIET_TAR"} "$TMP_DOWNLOAD_DIR/$distributionUrlName" -C "$TMP_DOWNLOAD_DIR" || die "failed to untar"
fi

printf %s\\n "$distributionUrl" >"$TMP_DOWNLOAD_DIR/$distributionUrlNameMain/mvnw.url"
mv -- "$TMP_DOWNLOAD_DIR/$distributionUrlNameMain" "$MAVEN_HOME" || [ -d "$MAVEN_HOME" ] || die "fail to move MAVEN_HOME"

clean || :

exec_maven "$@"
```

//.gitignore

```
target/
```

```
.mvn/wrapper/maven-wrapper.jar
!/src/main//target/
!/src/test//target/

### STS ###
.apt_generated
.classpath
.factorypath
.project
.settings
.springBeans
.sts4-cache

### IntelliJ IDEA ###
.idea
*.iws
*.iml
*.ipr

### NetBeans ###
/nbproject/private/
/nbbuild/
/dist/
/nbdist/
/.nb-gradle/
build/
!/src/main//build/
!/src/test//build/
```

VS Code

.vscode/

//AccountServiceApplication.java LoanServiceApplication.java

```
@SpringBootApplication
@EnableEurekaClient
public class AccountServiceApplication {
   public static void main(String[] args) {
      SpringApplication.run(AccountServiceApplication.class,   args);
   }
}
```

// LoanServiceApplication.java

```
@SpringBootApplication
@EnableEurekaClient
public class LoanServiceApplication {
   public static void main(String[] args) {
      SpringApplication.run(LoanServiceApplication.class, args);
   }
}
```

//application.properties for Account Service

```
spring.application.name=ACCOUNT-SERVICE
server.port=8081
eureka.client.service-url.defaultZone=http://localhost:8761/eureka
eureka.instance.hostname=localhost
```

//application.properties for Loan Service

```
spring.application.name=LOAN-SERVICE
server.port=8082

eureka.client.service-url.defaultZone=http://localhost:8761/eureka
eureka.instance.hostname=localhost
```

//AccountServiceController.java

```
@RestController
@RequestMapping("/account")
public class AccountController {
    @GetMapping("/status")
    public String status() {
        return "Account Service is UP!";
    }
}
```

//LoanServiceController.java

```
@RestController
@RequestMapping("/loan")
public class LoanController {
    @GetMapping("/status")
    public String status() {
        return "Loan Service is UP!";
    }
}
```