Lab: Install and Configure Maven

In this lab, we are going to highlight **how to install Maven on windows**.

Maven is a build automation and dependency management tool for Java based applications. To build a project, Maven uses its project object model (POM) and a set of plugins.

For this lab, we will be using:

- JDK 8
- Maven 3.8.3
- Windows 10

Install JDK and Set up *JAVA_HOME*

The installation process of Apache Maven is quite simple, we just need to extract the Maven's zip file and set up maven environment variables.

However, Maven requires JDK to perform some important operations such as code compilation.

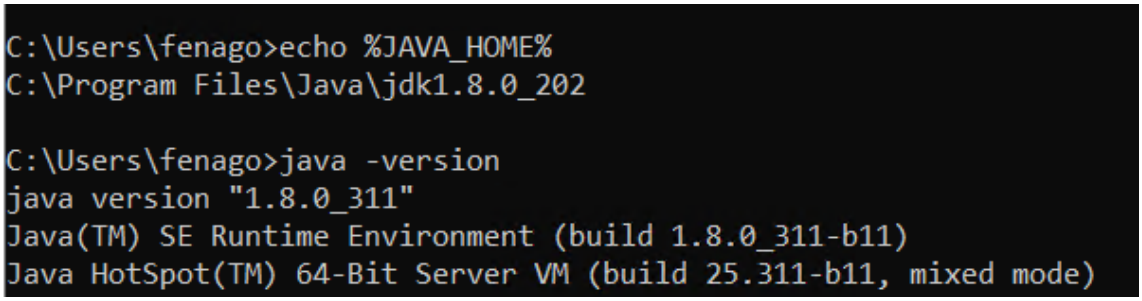
So, before installing Maven on Windows 10, we need to make sure that Java JDK is installed and the *JAVA_HOME* environment variable is configured on our machine.

We can type the following commands to check which JDK version is installed on our machine:

```
C:\Users\fenago>echo %JAVA_HOME%
C:\Program Files\Java\jdk1.8.0_202

C:\Users\fenago>java -version
java version "1.8.0_261"
Java(TM) SE Runtime Environment (build 1.8.0_261-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)

C:\Users\fenago>
```



As we can see, the installed version of JDK is *1.8.0_202*.

Download Apache Maven for Windows

We can download Maven directly from the official website: https://maven.apache.org/download.cgi. Make sure to download the latest stable release which now 3.8.3

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

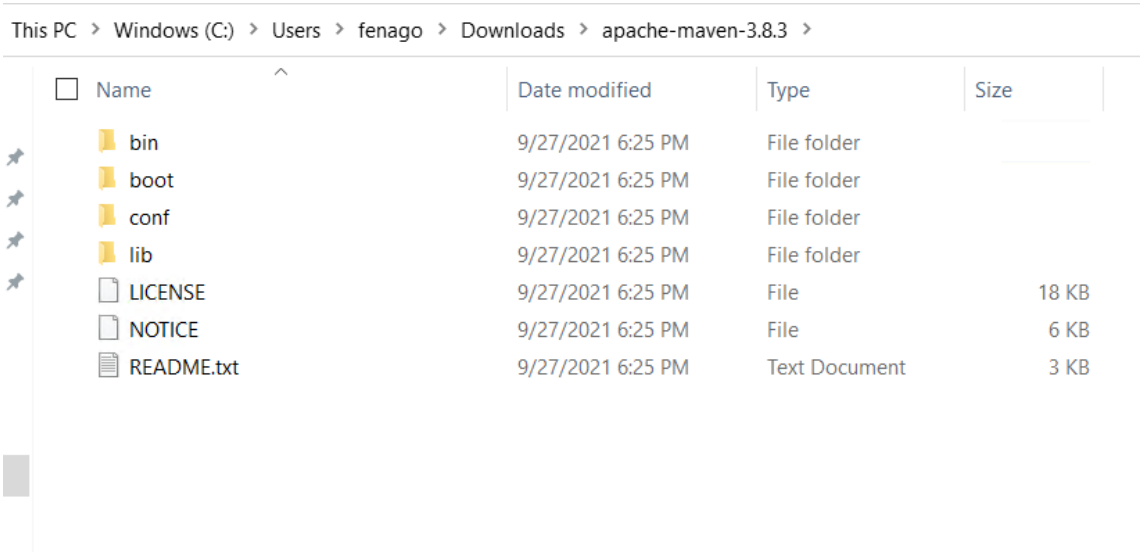
In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.8.1-bin.tar.gz	apache-maven-3.8.1-bin.tar.gz.sha512	apache-maven-3.8.1-bin.tar.gz.asc
Binary zip archive	apache-maven-3.8.1-bin.zip	apache-maven-3.8.1-bin.zip.sha512	apache-maven-3.8.1-bin.zip.asc
Source tar.gz archive	apache-maven-3.8.1-src.tar.gz	apache-maven-3.8.1-src.tar.gz.sha512	apache-maven-3.8.1-src.tar.gz.asc
Source zip archive	apache-maven-3.8.1-src.zip	apache-maven-3.8.1-src.zip.sha512	apache-maven-3.8.1-src.zip.asc

- Release Notes
- Reference Documentation
- Apache Maven Website As Documentation Archive
- All current release sources (plugins, shared libraries,...) available at https://downloads.apache.org/maven/
- latest source code from source repository
- Distributed under the Apache License, version 2.0

Install Apache Maven on Windows 10

Now, let's download *apache-maven-3.8.3-bin.zip* and unzip it to a specific folder of our choice, for example: *C:\Users\fenago\Downloads\apache-maven-3.8.3*

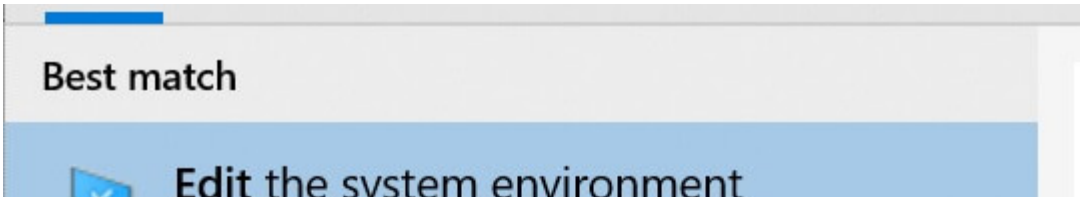


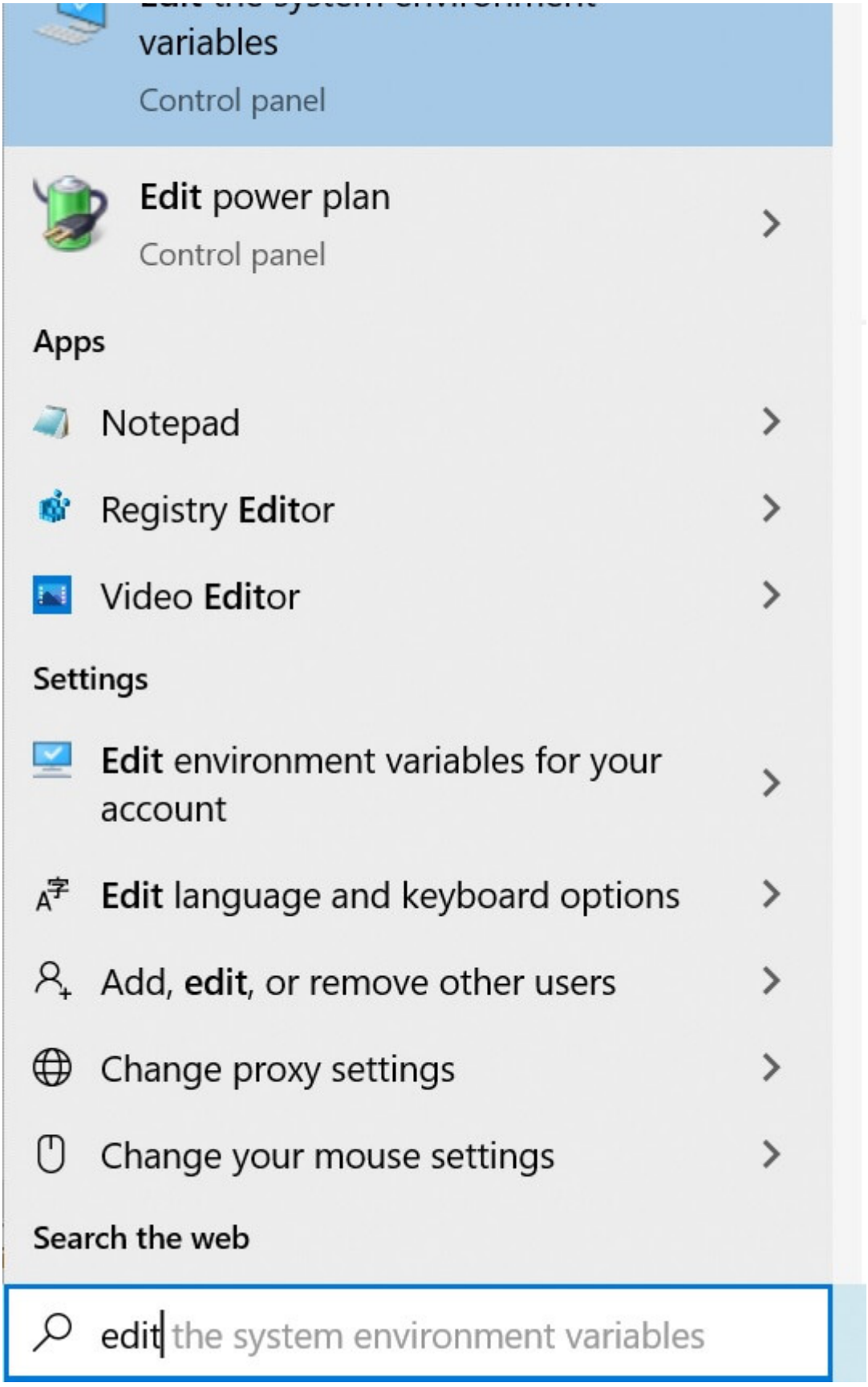
Add *MAVEN_HOME* Environment Variable

The next step is to configure the *MAVEN_HOME* variable on Windows 10.

MAVEN_HOME should point to the folder where we extracted Maven which is *C:\Users\fenago\Downloads\apache-maven-3.8.3* in our case. Follow these steps to add the *MAVEN_HOME* environment variable:

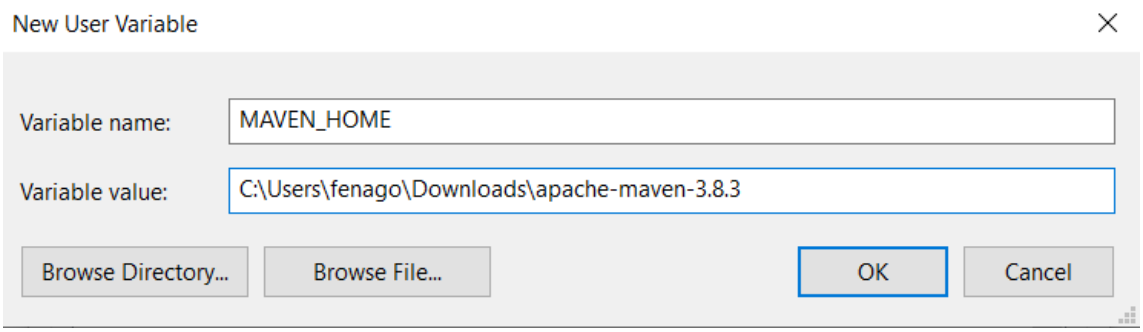
1- Type **"edit"** in the Window search box, then click on the **"Edit the system environment variables"**



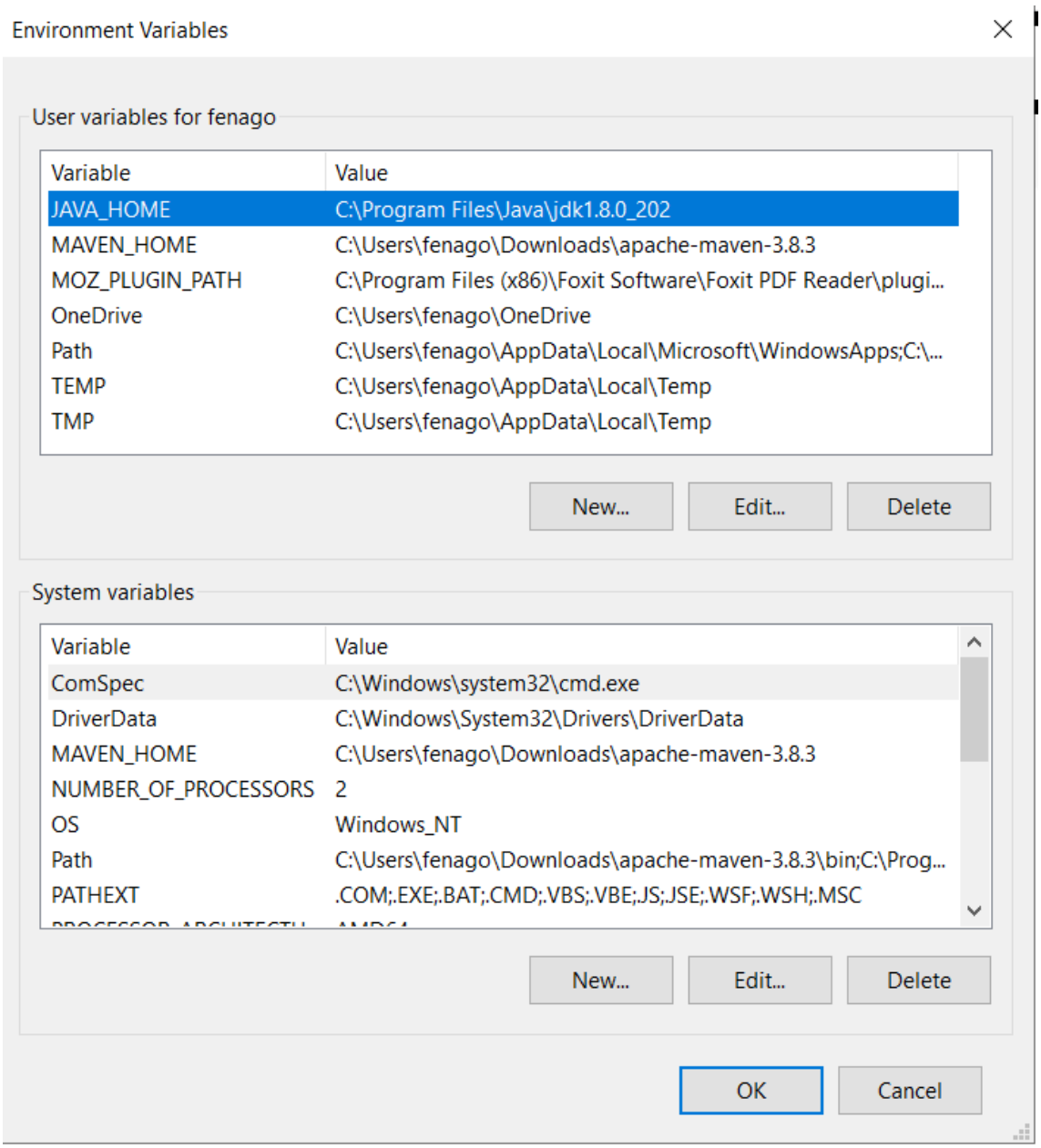

variables
Control panel
Edit power plan
Control panel
Apps
Notepad
Registry Editor
Video Editor
Settings
Edit environment variables for your account
Edit language and keyboard options
Add, edit, or remove other users
Change proxy settings
Change your mouse settings
Search the web
edit the system environment variables

2- Next, select the **"Environment Variables..."** button

3- Click on the **"New..."** button and put *MAVEN_HOME* as variable name and *C:\Users\fenago\Downloads\apache-maven-3.8.3* as variable value.



Note: Make sure to add variable in `System Variables` as well:



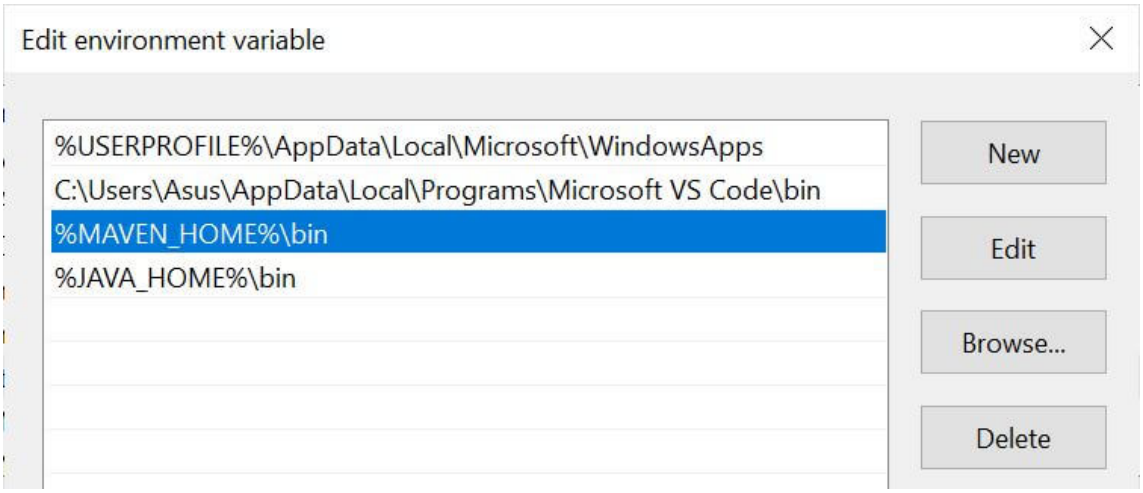
another way to set up *MAVEN_HOME* is to use command lines. To do so, open a new command prompt as administrator and type the following command:

```
        C:\Windows\system32>setx /M MAVEN_HOME "C:\Users\fenago\Downloads\apache-
maven-3.8.3"
        SUCCESS: Specified value was saved.
```

Add *%MAVEN_HOME%\bin* to *PATH*

Now let's add the last missing piece of the puzzle. This step is very important to run the Mvn command everywhere directly from the command prompt.

To do so conveniently, we need to edit the *PATH* variable by appending the Maven bin folder *%MAVEN_HOME%\bin*.



Please bear in mind that we can accomplish the same thing using this command line:

```
        C:\Windows\system32>setx /M PATH "%MAVEN_HOME%\bin;%PATH%"
```

Verify Mvn Installation

Now that we put all the pieces together, let's test if Apache Maven is successfully installed on Windows 10.

To verify the installation, we run:

```
        C:\Users\fenago>mvn --version
        Apache Maven 3.8.3 (05c21c65bdfed0f71a2f2ada8b84da59348c4c5d)
        Maven home: C:\Users\fenago\Downloads\apache-maven-3.8.3\bin\..
        Java version: 1.8.0_261, vendor: Oracle Corporation, runtime: C:\Program
Files\Java\jdk1.8.0_261\jre
        Default locale: en_US, platform encoding: Cp1252
        OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

        C:\Users\fenago>echo %MAVEN_HOME%
        C:\Users\fenago\Downloads\apache-maven-3.8.3

        C:\Users\fenago>
```

As shown above, the command indeed displays the Maven version, Maven bin folder, Java version, and operating system information.

Common Issues

1) *mvn is not recognized as an internal or external command*

```
        C:\Users\fenago>mvn --version
        'mvn' is not recognized as an internal or external command,
```

```
        operable program or batch file.
```

This means that the installation is not done properly, make sure *%MAVEN_HOME%\bin* is prepended to the *PATH* variable the right way.

2) *JAVA_HOME environment variable is not defined correctly*

```
        C:\Users\fenago>mvn --version
        The JAVA_HOME environment variable is not defined correctly
        This environment variable is needed to run this program
        NB: JAVA_HOME should point to a JDK not a JRE
```

This simply means that JDK is not installed and the *JAVA_HOME* variable is not properly configured.

Conclusion

In this lab, we walked you through the steps of how to install Maven on Windows 10.