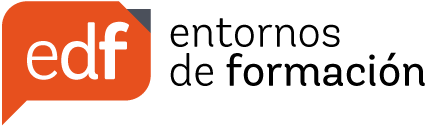
****

Sakai Install guide on Windows 10

28/02/2023 - Sakai 22.x and newer

**Índice**

[**Introduction**](#_zf8n32osg01i) **2**

[**Install dependencies**](#_302wbmx5fv0e) **3**

[Installing GIT](#_tot5ceufphqh) 3

[Download the Sakai source](#_8buu839sszwc) 7

[Installing Java 11](#_1df5c5m79bda) 8

[Installing Maven](#_6ia4jhjcgav2) 11

[Setting the environment variables](#_e7pwa9ul5ljb) 13

[Compiling Sakai](#_vsv7g5ktu3q9) 18

[Installing Tomcat](#_7vtm4h1awttr) 20

[Installing MariaDB](#_d1idd3ao4z8x) 26

[Configuring Sakai](#_8rdkpd5efr3l) 31

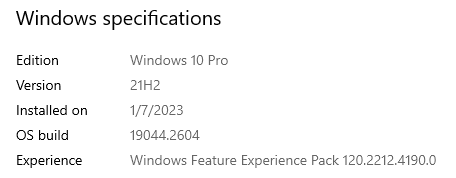
[**Starting up Sakai**](#_z3vireq1lray) **33**

[**Stopping Sakai**](#_fcix18g2j4a7) **35**

# 

# Introduction

This guide pretends to describe the installation process of the Sakai LMS software in a Windows 10 machine, at the moment this guide was written the OS compilation was:



It should work on Windows 11, however, there could be small differences in the Windows UI.

Sakai 22.x and 23.x is supported by Java 11 and Tomcat 9, and also supports MySQL, MariaDB and Oracle as databases, this guide will be constructed around these versions but newer versions would work:

* Java 11
* Apache Tomcat 9.0.72
* Apache Maven 3.8.7
* MariaDB 10.11.2
* GIT 2.39.2

Everything on this guide will be installed in a hard disk location without spaces, let’s use **C:\opt** as example.

If you have any doubt about this guide you can always contact [sakai-dev@apereo.org](mailto:sakai-dev@apereo.org), it was written and contributed by [Miguel Pellicer](mailto:mpellicer@entornosdeformacion.com).

# 

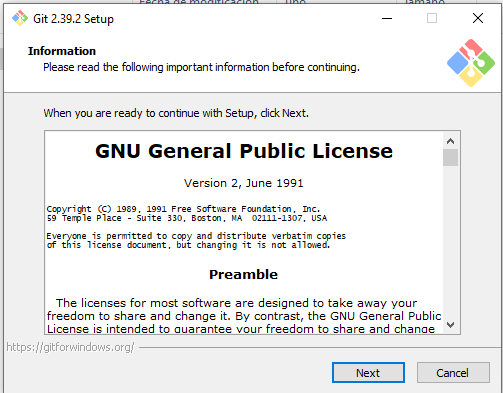
# Install dependencies

## Installing GIT

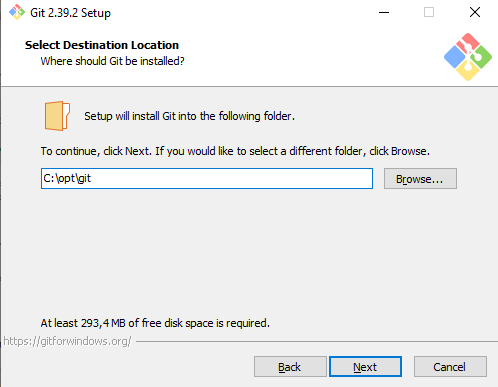
Let’s install some dependencies, first we need a GIT client, we can use the [GitHub client](https://desktop.github.com/) or a simple [GIT client](https://git-scm.com/downloads). We’re going to use a simple GIT client in this guide but feel free to choose your favorite GIT client.



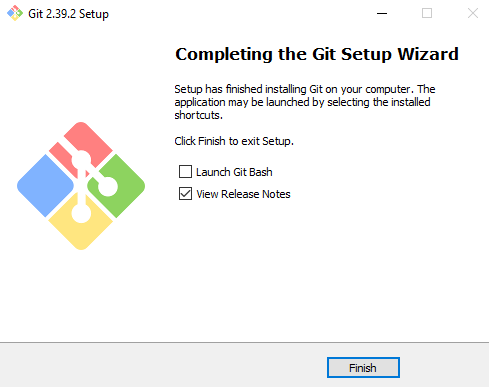
Click download for windows, download the file and double click the installer:



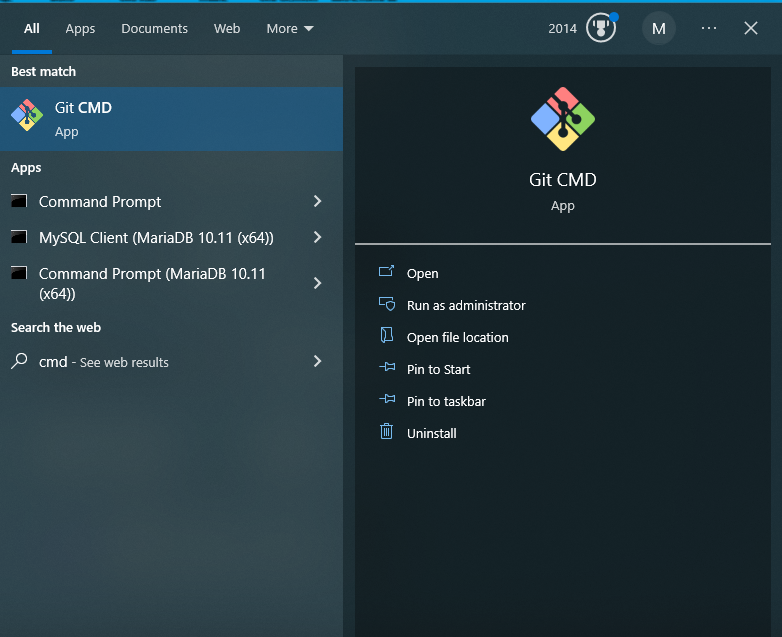
You can install it in the **C:\opt\** folder:



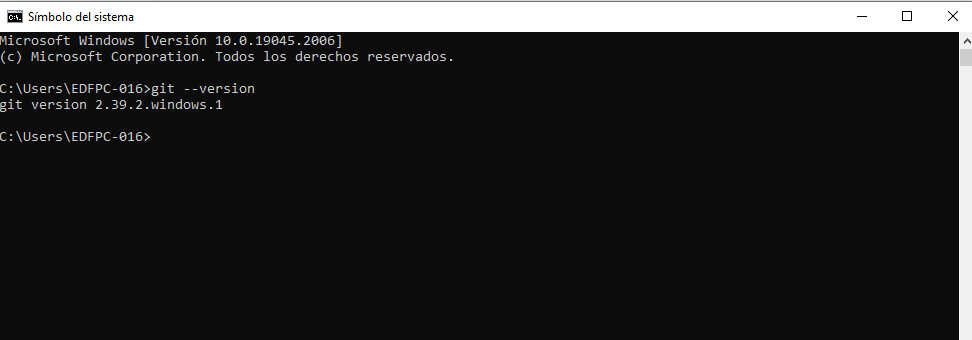
Just choose all the default options, next, next, next, install. After the installation process you’ll see something like this:



Let’s open a console and verify the GIT installation, press the ‘Windows’ button and look for ‘cmd’.



Type “git –version” to ensure that GIT is installed correctly.



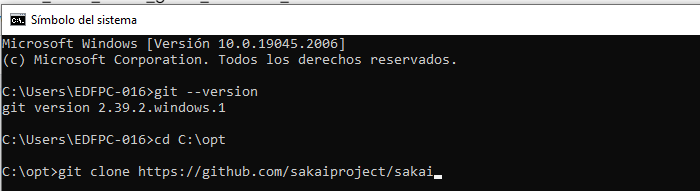
If you see the version now you’re ready to download the Sakai source.

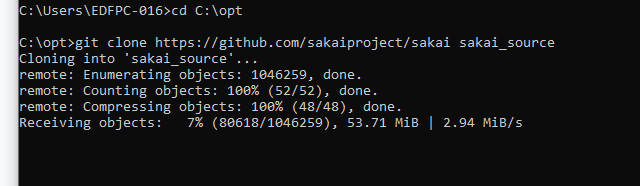
## 

## Download the Sakai source

The Sakai source is in GitHub, here [GitHub - sakaiproject/sakai: Sakai is a freely available, feature-rich technology solution for learning, teaching, research and collaboration. Sakai is an open source software suite developed by a diverse and global adopter community.](https://github.com/sakaiproject/sakai)

Go to the console (CMD) and change the directory to **C:\opt**, then we need to clone the source code with the command ‘git clone <https://github.com/sakaiproject/sakai> sakai\_source’. This command will download the Sakai source code into the C:\opt\sakai\_source directory.





While the source is downloading, we can just install other components, let’s install Java.

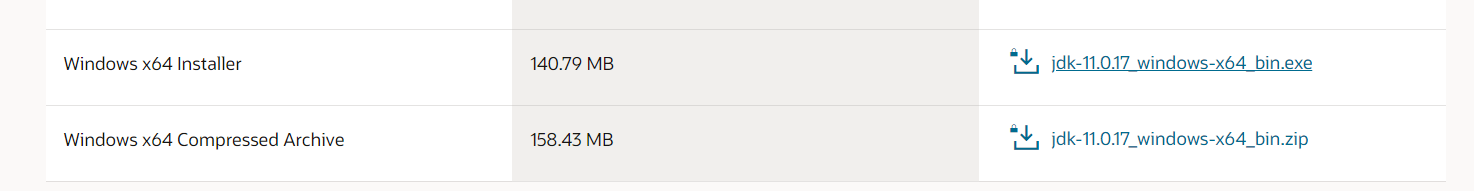
## 

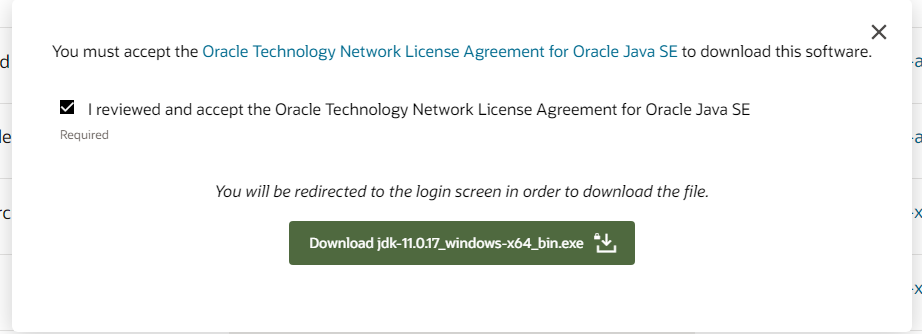
## Installing Java 11

Go to this website and download the latest version of Java 11, in the case of this guide it is going to be **Java SE Development Kit 11.0.17 but newer versions should work.**

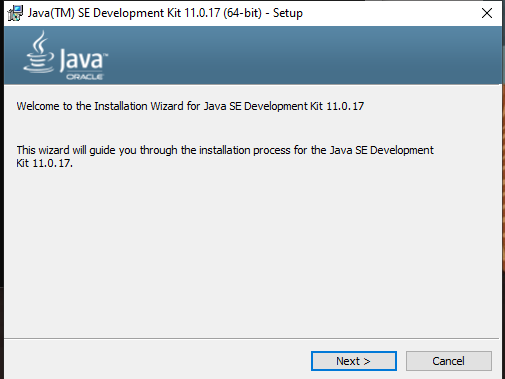
<https://www.oracle.com/es/java/technologies/javase/jdk11-archive-downloads.html>

Choose Windows 64:

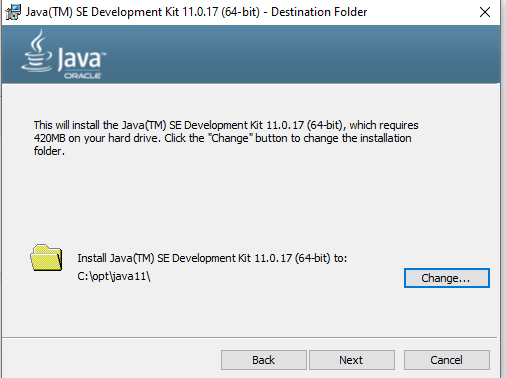




Double click on the installer, you should see something like this:



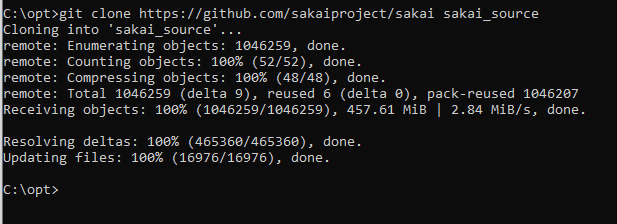
Let’s choose a simple path for the installation, **C:\opt\java11** is ideal.



You should see something like this:



At the moment we installed Java, our source code must be downloaded on C:\opt\sakai\_source.



## 

## Installing Maven

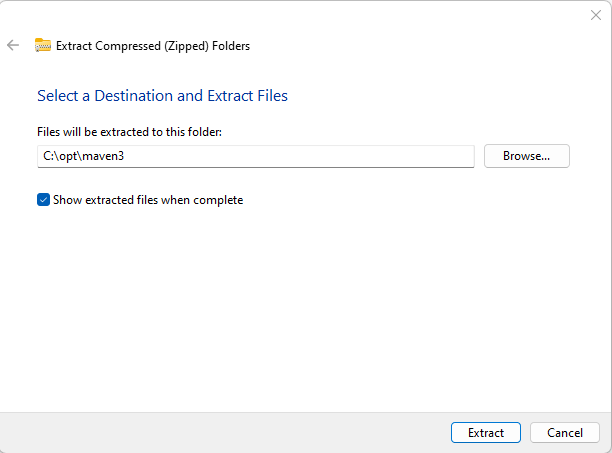
Now we have the source code, and Java installed, we need Maven to compile and deploy Sakai, setting the environment variables first, but let’s focus on Maven.

We need Maven 3.8.7 as a minimum, newer versions like 3.9 do not work at the moment this guide was written, so let’s focus on the 3.8.x series.

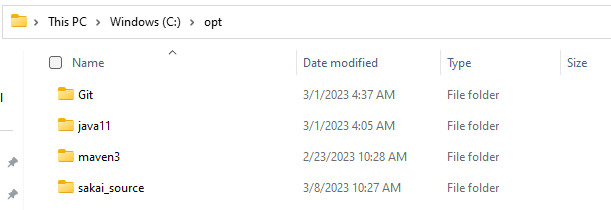
Download Maven from here:

<https://archive.apache.org/dist/maven/maven-3/3.8.7/binaries/apache-maven-3.8.7-bin.zip>

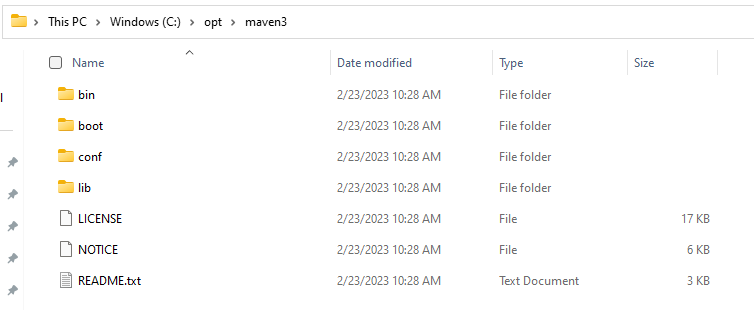
It’s a simple ZIP file, uncompress it in a nice location without spaces, **C:\opt\maven3** is optimal! To uncompress the file just left click and choose uncompress:



Now you should have something like this on your C:\opt folder:



It’s SUPER important that inside the maven3 and java11 folders, you find the BIN folder.



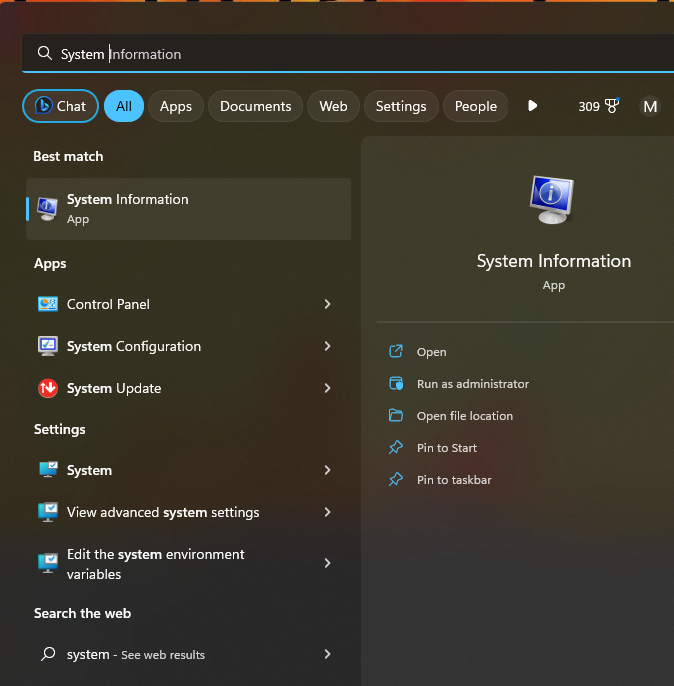
## 

## Setting the environment variables

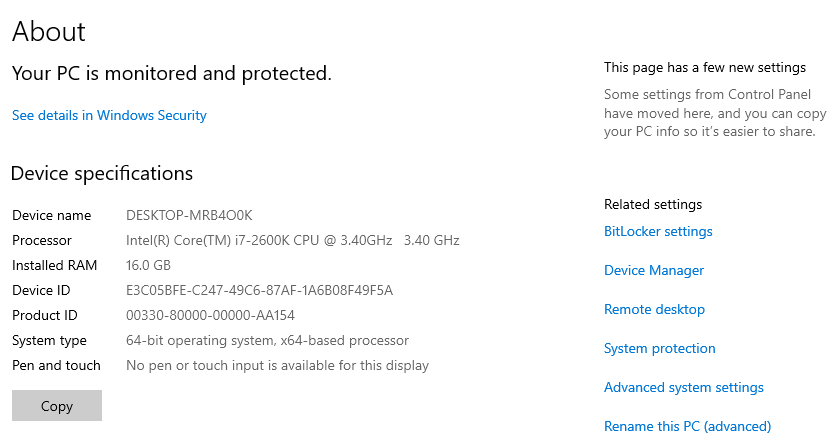
This is the always the hardest part because we need to define the environment variables in Windows, we need at least the following variables:

* JAVA\_HOME
* MAVEN\_HOME
* CATALINA\_HOME
* CATALINA\_BASE
* Modify the PATH variable

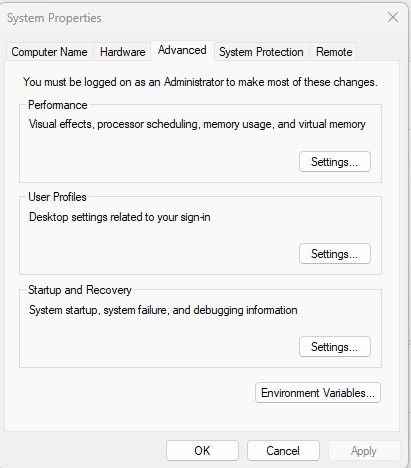
Press the combination of ‘Windows button and pause’, you can also look for System after pressing the windows button:



You will see something like this:



Click Advanced Configuration which is in the left part of the screen, something like this will be showed:



Click the button at the end, it should say something like environment variables.

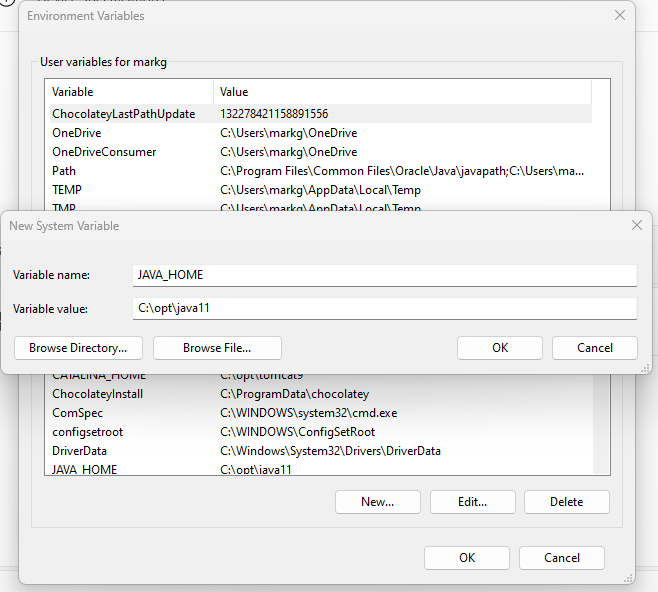
Let’s define the variables, create some variables with these values:

JAVA\_HOME = C:\opt\java11

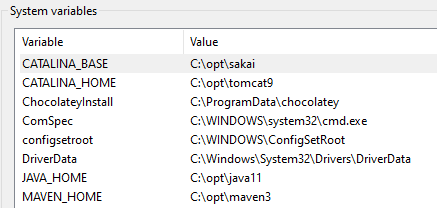
MAVEN\_HOME = C:\opt\maven3

CATALINA\_HOME = C:\opt\tomcat9

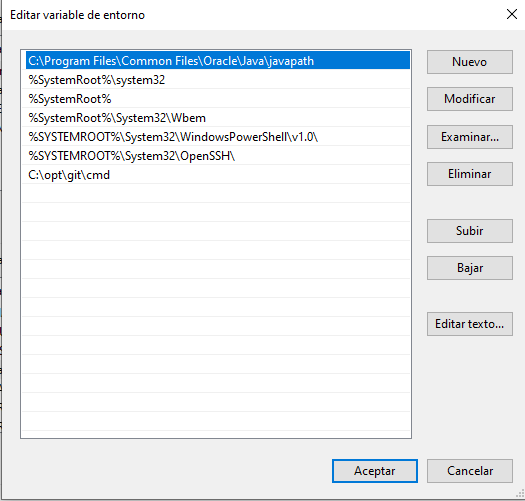
CATALINA\_BASE = C:\opt\sakai

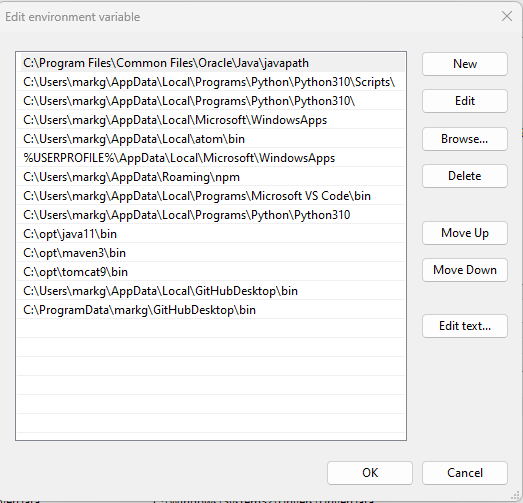


If you have something like this, you’re good:



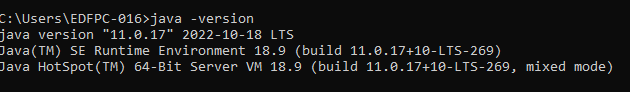
Now it’s the trickiest part, we must add the variables to the PATH variable, search for the PATH variable and edit it, by default should look like this:



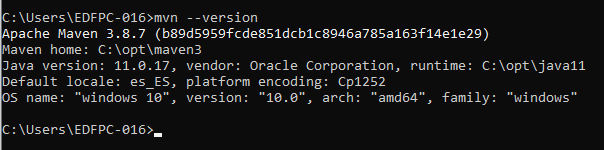
Add **C:\opt\java11\bin, C:\opt\maven3\bin and C:\opt\tomcat9\bin**, this will grant access to the executable files, confirm everything and close all the windows.

Let’s verify the environment, open a new console typing CMD in the search menu.

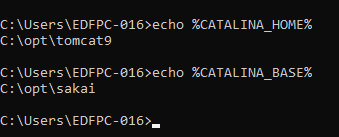
Verify Java ‘java -version’



Verify Maven ‘mvn –version’.



Verify Tomcat variables, ‘echo %CATALINA\_HOME%’ and ‘echo %CATALINA\_BASE%’

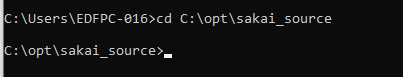


## 

## Compiling Sakai

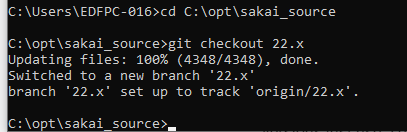
Now we have Maven and Java, we can compile Sakai, let’s compile Sakai 22.x because it’s the latest release at the moment this guide was written.

Go to the Sakai Source folder, C:\opt\sakai\_source using the CMD.



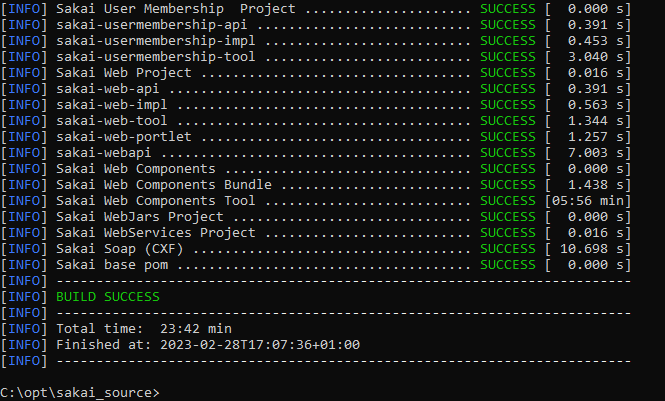
Checkout the Sakai 22.x branch because we’re going to install Sakai 22.x, for other versions just use a different branch.

‘git checkout 22.x’



Now we need the Maven command to cleanup the existing artifacts and compile the new ones, for that we need to execute the command ‘mvn clean install -DskipTests’. We’re going to skip the tests in our local installation, they are performed in every change to the community branch, but it would be good to perform them while doing development tasks.

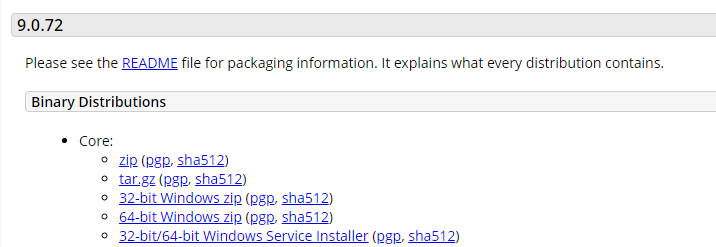
This is going to take a while, the first time Maven must download all the libraries and dependencies, just wait and take a coffee!



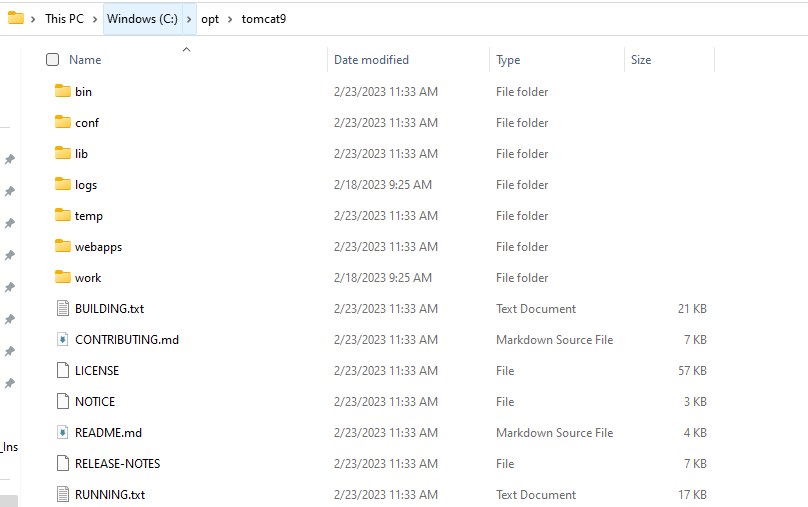
Very nice! If you see that Sakai compiled well, we’re ready to deploy it to an application server, let’s install Tomcat 9!

## Installing Tomcat

Go to this website <https://tomcat.apache.org/download-90.cgi> and click on the Windows 64-bit link.

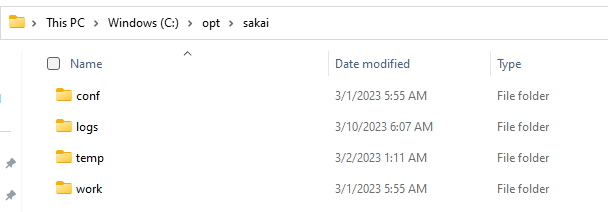


Then uncompress the file into the **C:\opt\tomcat9** location, ensure that the BIN folder is inside **C:\opt\tomcat9**. Your installation should look like this:

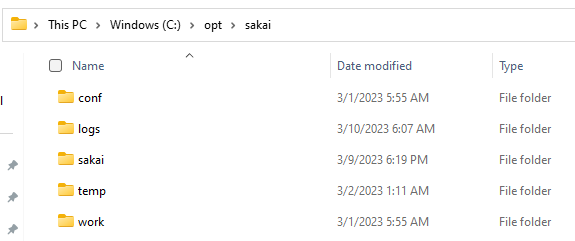


Let’s also create the sakai directory on **C:\opt\sakai**, it should be empty for now. Copy from the Tomcat directory, **C:\opt\tomcat9**, the folders conf, logs, temp and work.

Now **C:\opt\sakai** should look like this:



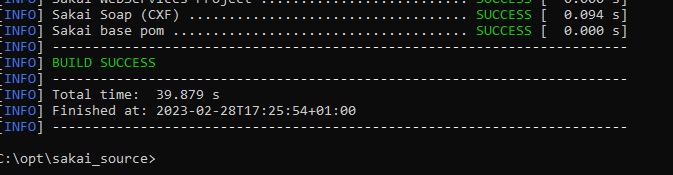
Create a folder called ‘sakai’.



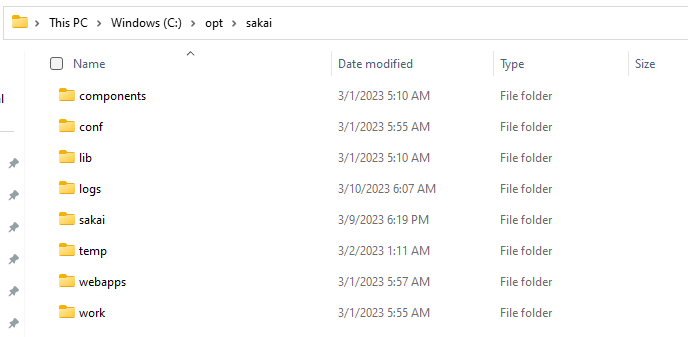
Now we can deploy the Sakai source code, go to the CMD again and type the command ‘mvn sakai:deploy -Dmaven.tomcat.home=C:\opt\sakai -DskipTests’, we’re telling Maven to deploy everything we compiled into **C:\opt\sakai**.



After waiting a little bit, it should state a successful build.

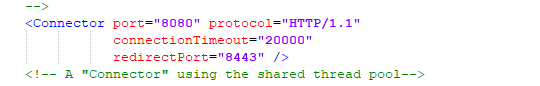


After this, the Sakai installation should be present in **C:\opt\sakai**, ensure you have the components, lib and webapps folder:

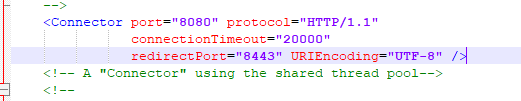


If your **C:\opt\sakai** folder looks like this then you’re good to continue! Let’s configure tomcat.

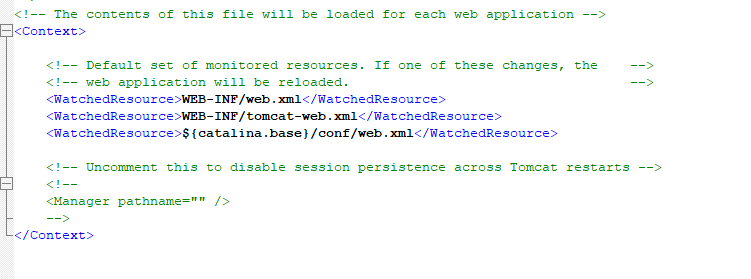
Open the file ‘conf/server.xml’ and edit the ‘Connector’ section, you should find something like this:



We must add the URIEncoding property, ‘URIEncoding=”UTF-8”’ to the connector, it will look like this:



Now open the ‘conf/context.xml’ file, it should look like this:



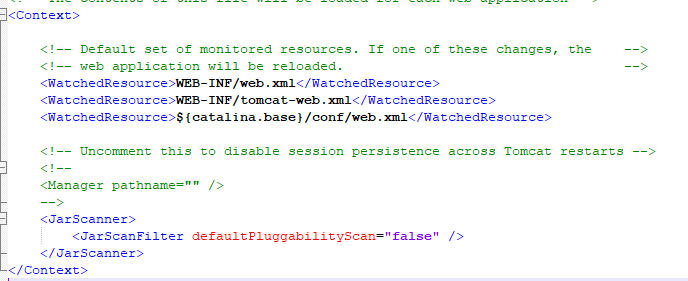
Add these lines inside the context block:

<JarScanner>

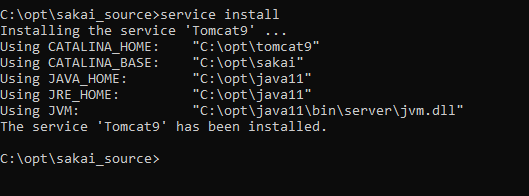
<JarScanFilter defaultPluggabilityScan="false" />

</JarScanner>

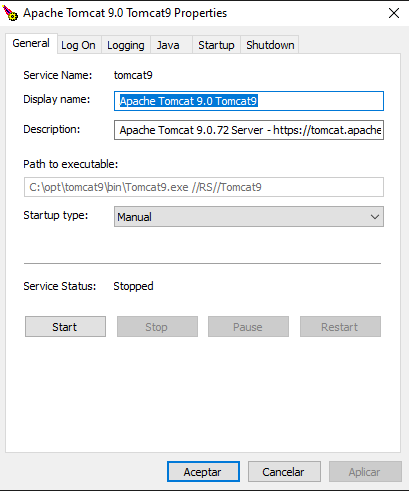
It will look like this:



Now we can install the Tomcat service, go again to your CMD and type ‘service install’, it should install the tomcat service.

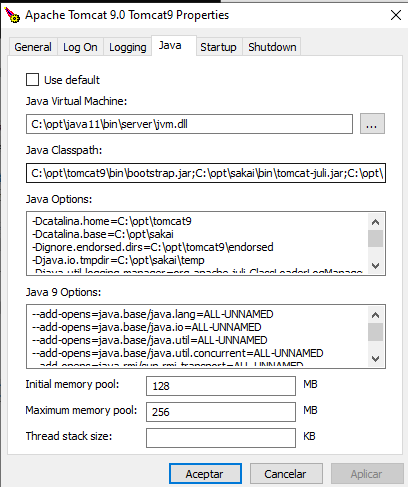


If this does not work go back and fix your environment variables, typing ‘tomcat9w’ will open the tomcat9 service window.



DO NOT click Start, we still need to provide more memory to Sakai, define the language and the most important, connect Sakai to a database!

Let’s define the memory, go to the ‘Java’ tab and see the values:



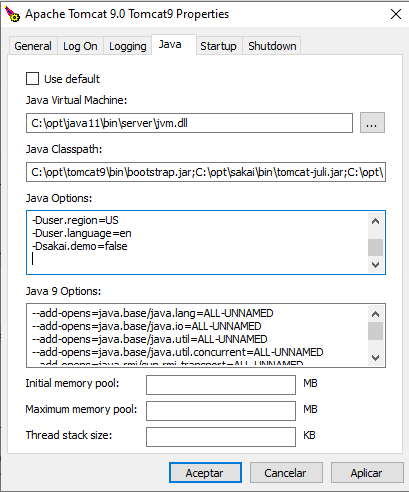
We must remove the ‘initial memory pool’ and the ‘maximum memory pool’ values.



Then, in the ‘Java Options’, define the following values:

| -Xmx4g  -Xms512m  -XX:MaxMetaspaceSize=1g  -Dorg.apache.jasper.compiler.Parser.STRICT\_QUOTE\_ESCAPING=false  -Djava.awt.headless=true  -Dcom.sun.management.jmxremote  -Dhttp.agent=Sakai  -Djava.util.Arrays.useLegacyMergeSort=true  -Dfile.encoding=UTF8  -Duser.region=US  -Duser.language=en  -Dsakai.demo=false |
| --- |

-Duser.region and -Duser.language will define your default language. -Dsakai.demo will populate some demo data, it is recommended for development and demo purposes but not for production purposes.



Apply the changes but DO NOT startup Tomcat, now we need a database.

## Installing MariaDB

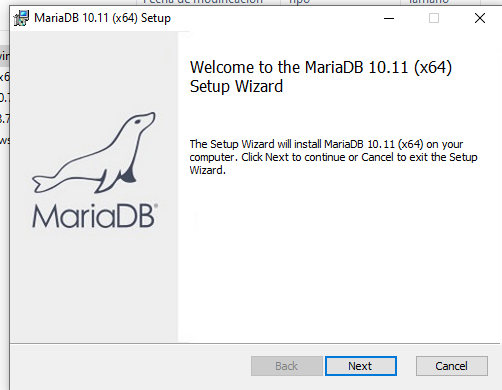
You can install MariaDB, MySQL and Oracle, all of them are supported, however this guide will use MariaDB.

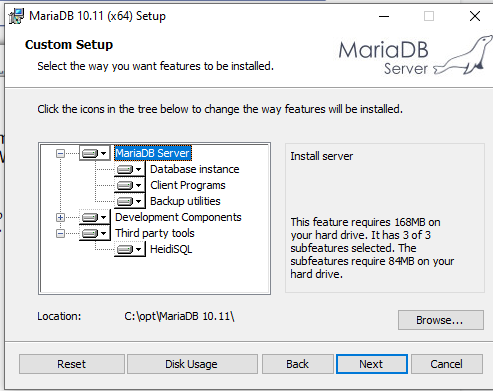
Download and install MariaDB from the link:

<https://mariadb.org/download>

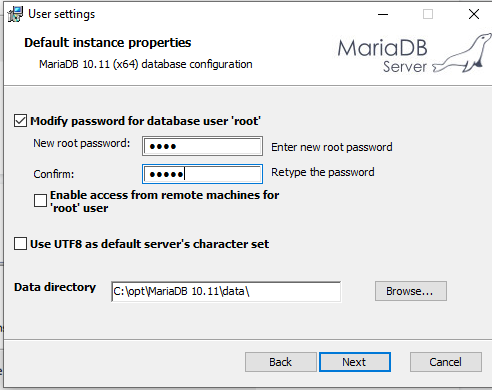
In this tutorial MariaDB 10.11.2 was used, install it in **C:\opt**

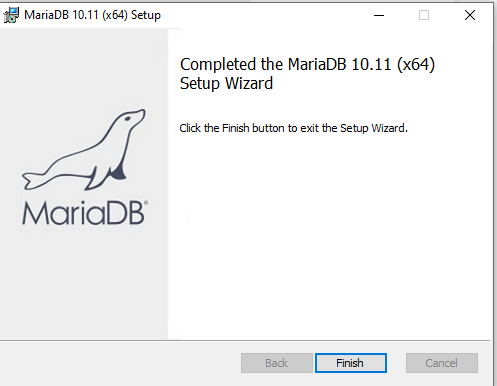




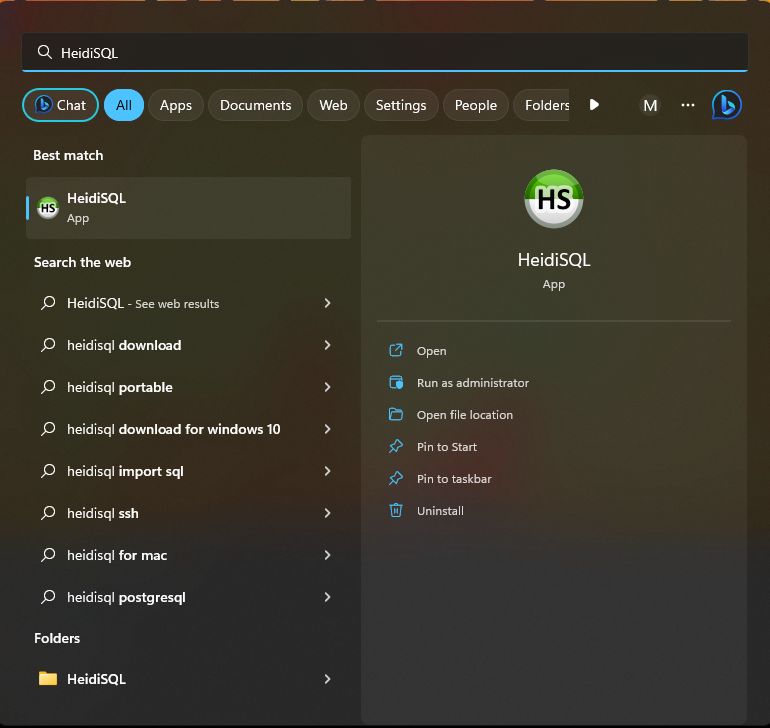


For development purposes, just use a simple user / password, but remember it! Do not use simple passwords for production instances. In this tutorial we used ‘root/admin’.

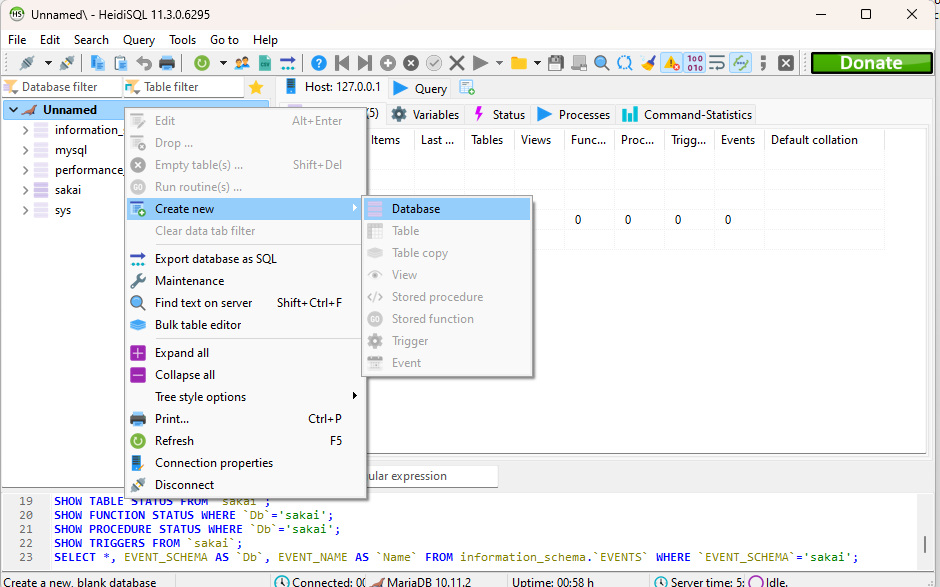


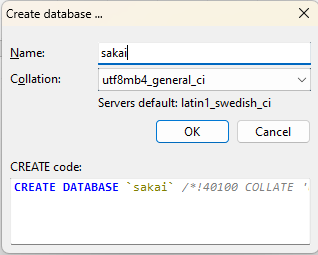


After completing the installation, now you can create a database for Sakai. Open HeidiSQL using the Windows button or your preferred SQL tool.

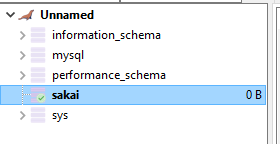


Create a new database ‘sakai’ using Heidi or other software like the MySQL Workbench or TOAD, it should be utf8mb4\_general\_ci



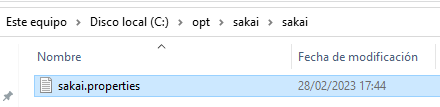


It should be listed and empty!



## Configuring Sakai

Now we should tell Sakai which database is going to use and where Sakai should host the files, let’s create a ‘sakai.properties’ file inside the **C:\opt\sakai\sakai** folder, it should looks like this:



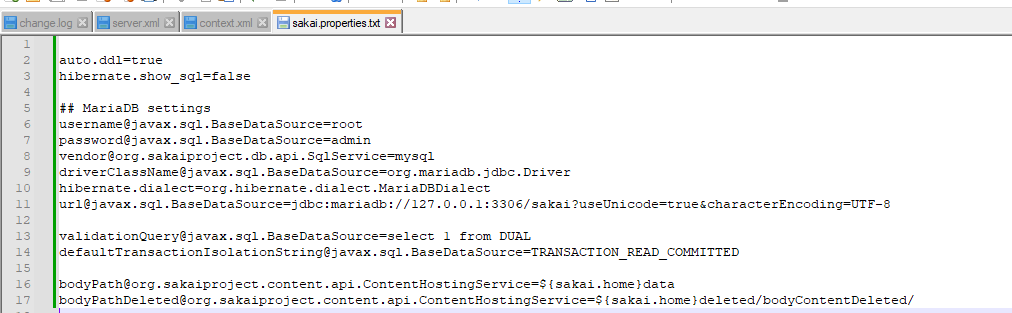
Write the following content, it could be minimum, for advanced configuration check the reference here:

[sakai/default.sakai.properties at master · sakaiproject/sakai · GitHub](https://github.com/sakaiproject/sakai/blob/master/config/configuration/bundles/src/bundle/org/sakaiproject/config/bundle/default.sakai.properties)

| auto.ddl=true  hibernate.show\_sql=false  ## MariaDB settings  username@javax.sql.BaseDataSource=root  password@javax.sql.BaseDataSource=admin  vendor@org.sakaiproject.db.api.SqlService=mysql  driverClassName@javax.sql.BaseDataSource=org.mariadb.jdbc.Driver  hibernate.dialect=org.hibernate.dialect.MariaDBDialect  url@javax.sql.BaseDataSource=jdbc:mariadb://127.0.0.1:3306/sakai?useUnicode=true&characterEncoding=UTF-8  validationQuery@javax.sql.BaseDataSource=select 1 from DUAL  defaultTransactionIsolationString@javax.sql.BaseDataSource=TRANSACTION\_READ\_COMMITTED  bodyPath@org.sakaiproject.content.api.ContentHostingService=${sakai.home}data  bodyPathDeleted@org.sakaiproject.content.api.ContentHostingService=${sakai.home}deleted/bodyContentDeleted/ |
| --- |

It is a super simple configuration, we’re just telling Sakai to use the MariaDB database, and host the files in the installation directory under ‘sakai’ directory.

Your ‘**C:\opt\sakai\sakai.properties**’ should look like this:

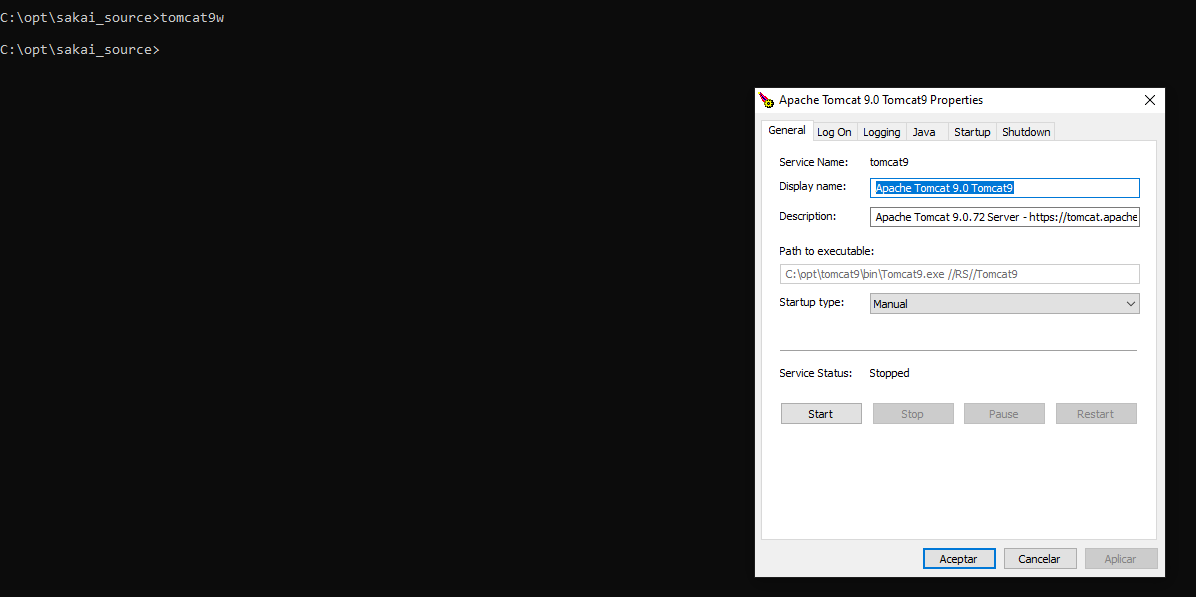


IMPORTANT, ensure it’s ‘sakai.properties’ with the right extension!

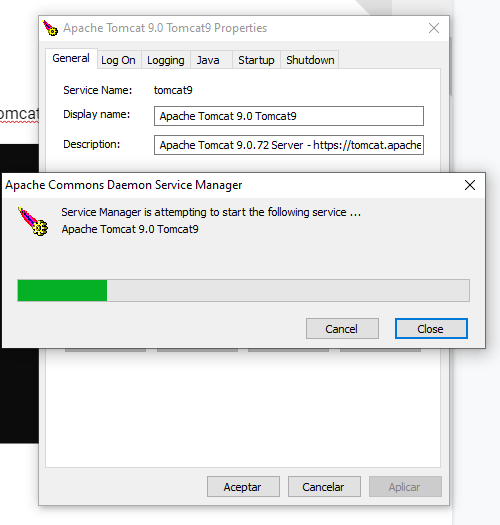
Now we’re ready to start up Sakai!

# Starting up Sakai

Open the tomcat9 service typing ‘tomcat9w’ in the cmd console.

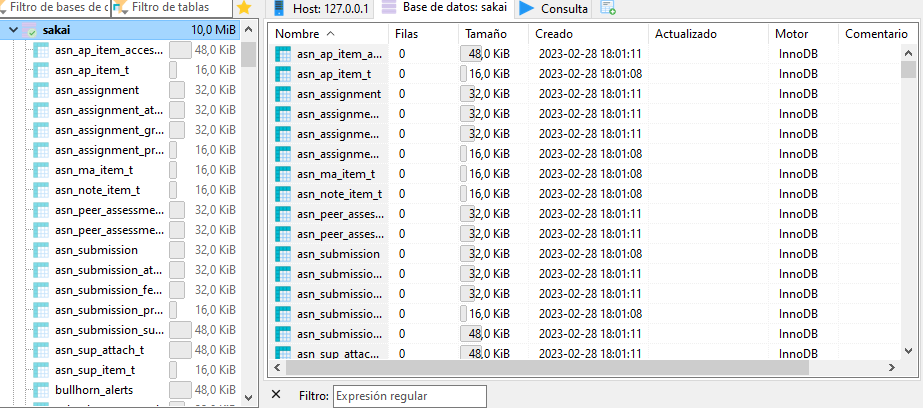


Now you can click ‘Start’, the process will begin. The following page shows the process beginning:



You can follow the startup reading the logs, this is very very recommended, you can find them in **‘C:\opt\sakai\logs\**’. For troubleshooting, always read the logs, they usually give the clue about what is the problem.

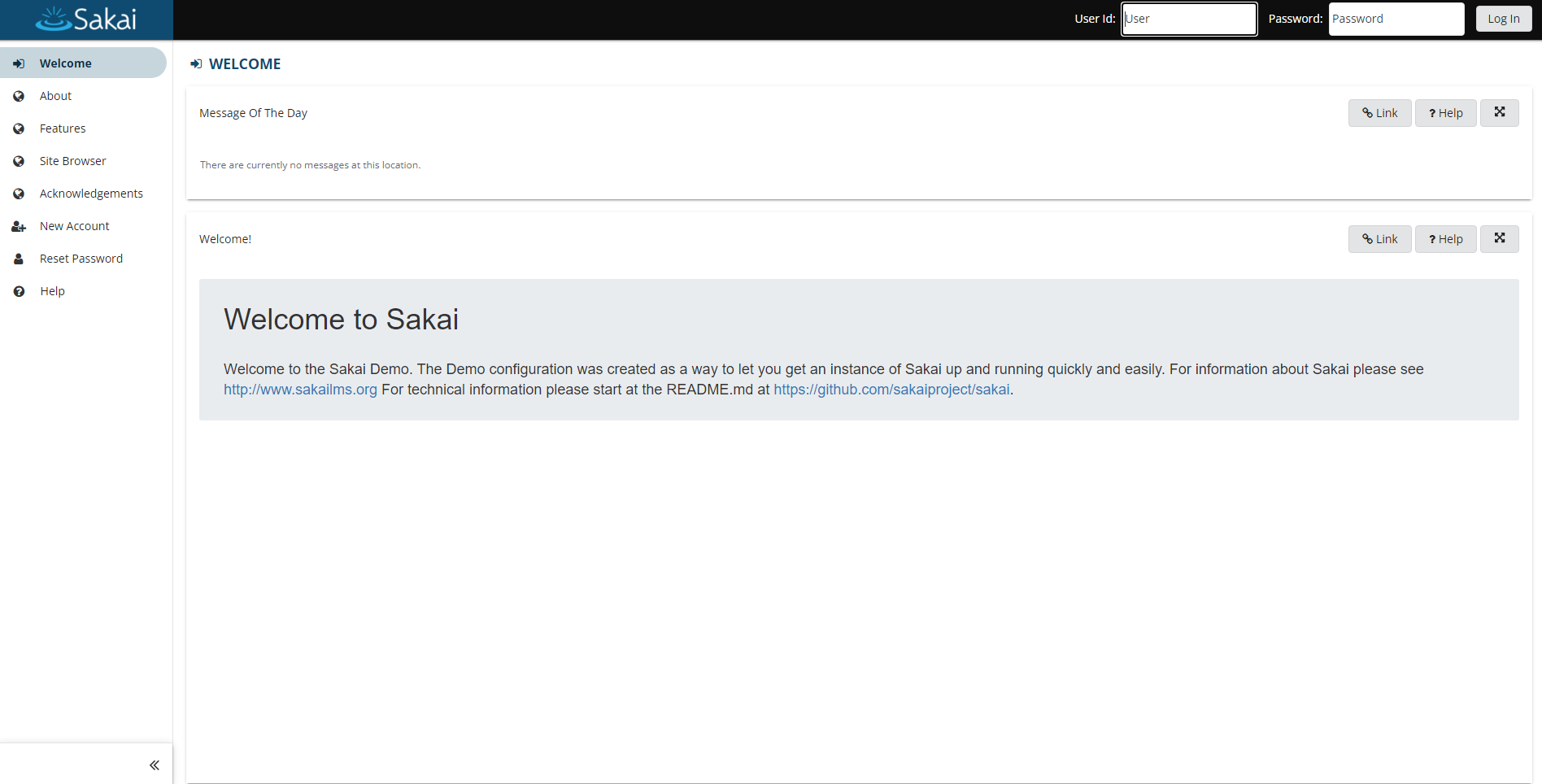
At the same time, you’ll see in Heidi or your SQL tool that is creating the database tables and objects:



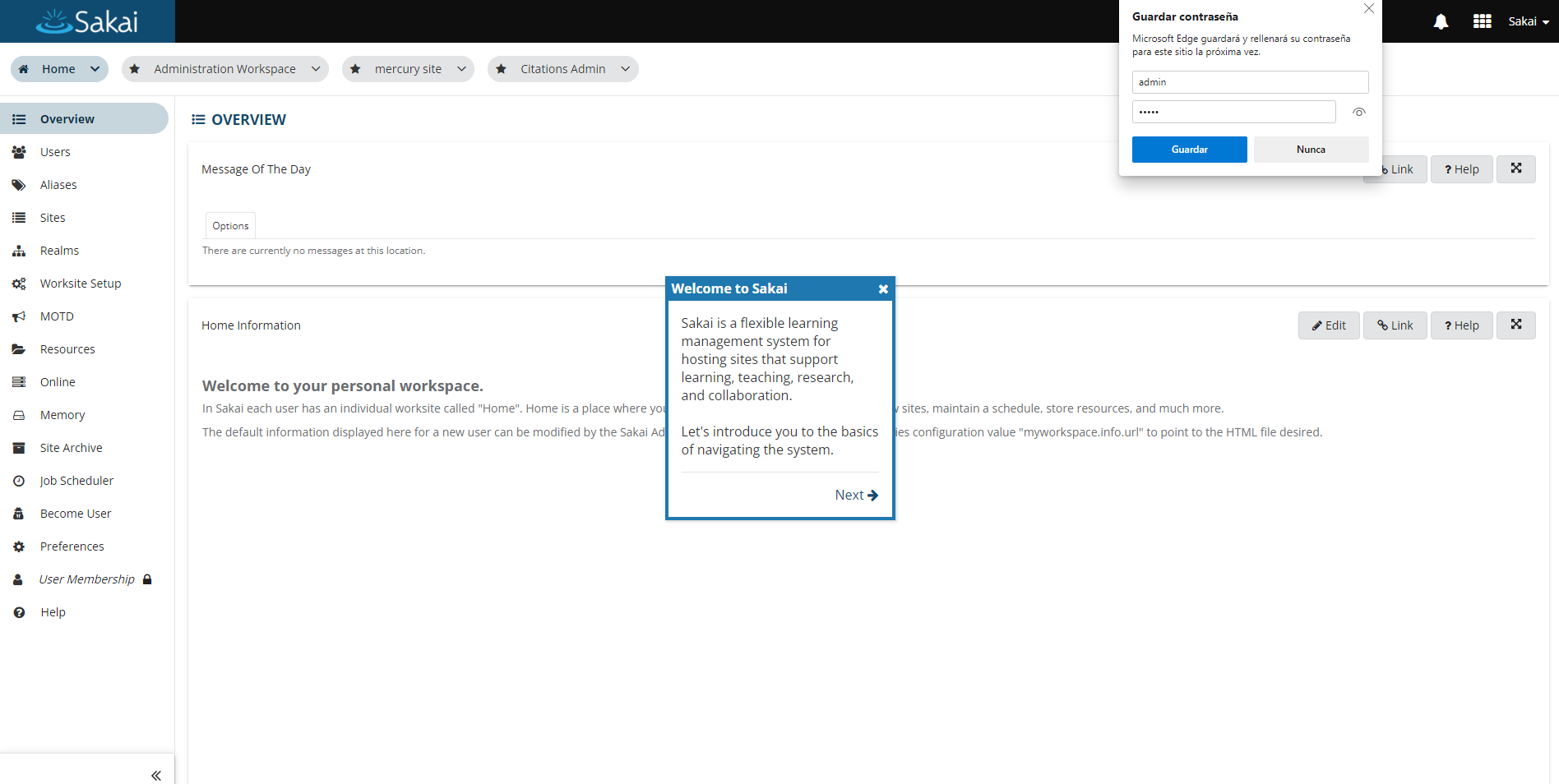
It is going to take some minutes, the logs are pretty clean so no major issues are present, after waiting some minutes voila!

You can now access Sakai through your browser and use it!

<http://localhost:8080/portal>



Using admin / admin you can access your own instance of Sakai!



Now you’re able to play with your own data, start doing some development work just restarting and stopping your instance.

# 

# Stopping Sakai

Just use the Tomcat dialog typing ‘tomcat9w’ in the console and click STOP.

