

# 1 Gurobi Installation Guide - Windows

This handout guides you through the process of installing Gurobi on your Windows laptop. Make sure to follow the steps in this guide carefully, and you will be ready to solve all kinds of optimization problems with Gurobi. We will start by installing it on Windows and then explore how to use it in IntelliJ for all your modeling tasks.

## 1.1 Get Gurobi on your Windows

To access all tools in Gurobi, you need to apply for an academic license. To do this, visit <https://portal.gurobi.com/iam/register/> and create an account with your university email address. You will get an email from Gurobi to verify your information and then your account is activated. Once your account is set up, you can navigate to a section where you can request your license (Figure 1). To do this, request the **Named-User Academic** license by clicking on ‘Generate Now’. It is essential to use the university’s network for this process since Gurobi needs to verify that you are a student. You can achieve this by either using eduVPN or by installing it while being connected to the campus network. In the pop-up that appears when the license is generated, you can click “Go to licenses”.

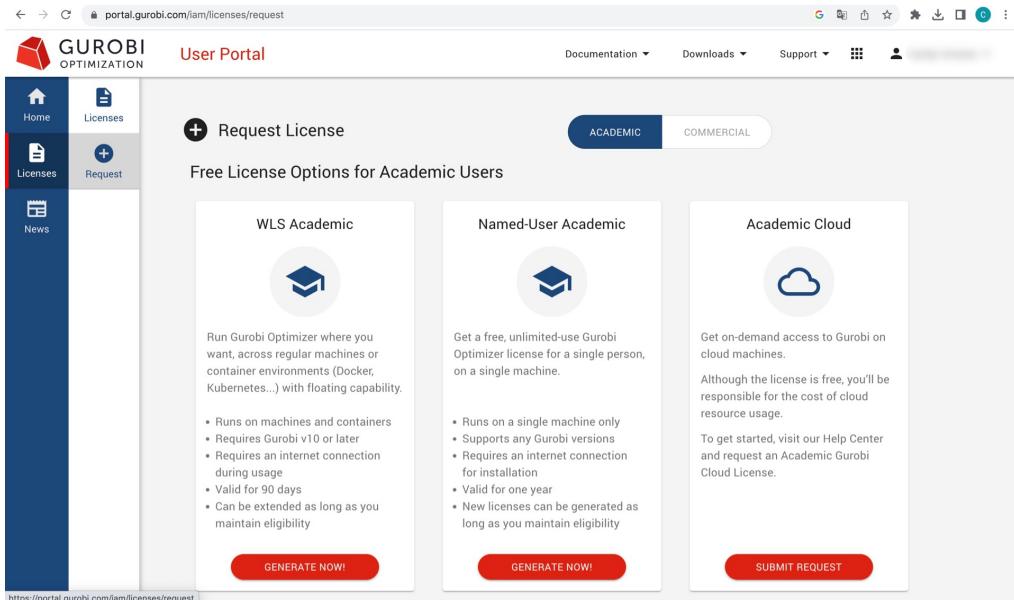


Figure 1: Request Gurobi license

Once you successfully generated your license, it must appear in your account under the tab

'Licenses', as shown in Figure 2.

ID	TYPE	PURPOSE	CREATED AT (Gmt+1)	EXPIRATION (Gmt+1)	CATEGORY	SUMMARY
redacted	Free Academic	Trial	Jan 4, 2024	Jan 4, 2025	Regular	version 11, not installed

Figure 2: Access page for the license

To be able to match your academic license with Gurobi, we must of course install the solver on our Windows computer. For this, you need to go to <https://www.gurobi.com/downloads/gurobi-software/> and select the **x64 Windows - Installer** version, which is indicated with blue in Figure 3. Once you click on the .msi file and accept to download it, it will appear in your Downloads folder.

v11.0.0	Installer	md5 Checksum
x64 Windows	Gurobi-11.0.0-win64.msi	Gurobi-11.0.0-win64.msi.md5
x64 Linux	gurobi11.0.0_linux64.tar.gz	gurobi11.0.0_linux64.tar.gz.md5
macOS Universal2	gurobi11.0.0_macos_universal2.pkg	gurobi11.0.0_macos_universal2.pkg.md5
x64 AIX	gurobi11.0.0_power64.tar.gz	gurobi11.0.0_power64.tar.gz.md5
arm64 Linux	gurobi11.0.0_armlinux64.tar.gz	gurobi11.0.0_armlinux64.tar.gz.md5

Figure 3: Download Gurobi

Just like any other installation, you can now click on this file and follow the general installation procedure. You should get a pop-up window that looks like Figure 4 and you can just press

**Next.** By default, the installer will place the Gurobi files in C:\gurobi1100\win64 directory. We advice you to not change this, as changing it might make it more difficult to setup Gurobi in IntelliJ later. You can simply follow the instructions of the Setup Wizard.



Figure 4: Windows installation prompt

When the installation is completed (Figure 5), Gurobi should be installed on your Windows computer and we are ready to get started.

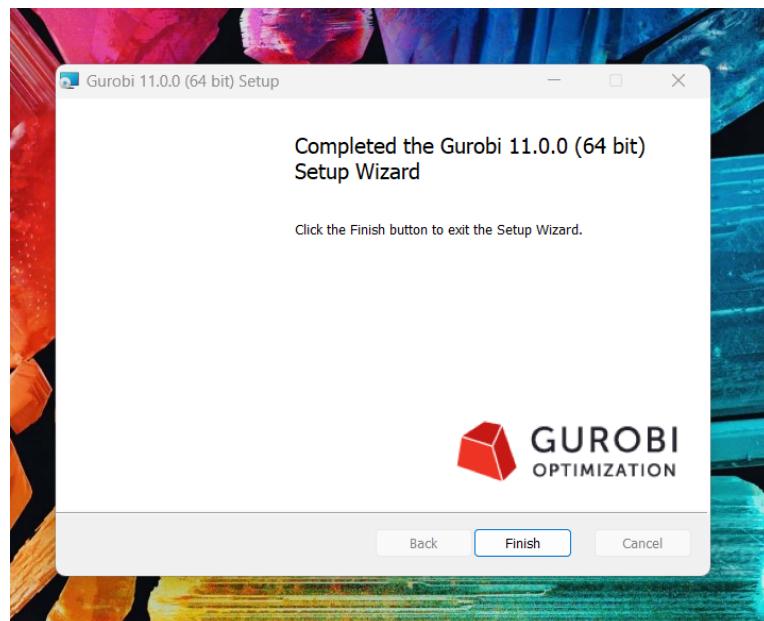


Figure 5: Windows installation finished

Now we are ready to link the academic license to the Gurobi installation. For this, go back to the Licenses page in your Gurobi account, as in Figure 2. Make sure you are connected to the campus network and have a working internet connection. Now, click on the **Install** (screen with downward arrow) button that is shown under your Free Academic license type. This will prompt a window explaining how to link the academic license (Figure 6). Since we already installed Gurobi, we can proceed directly to the linking process through the Start/Run menu (or Terminal). First, copy the key that is provided (`grbgetkey YOURKEY`). We will need this in the Start/Run menu (or Terminal).

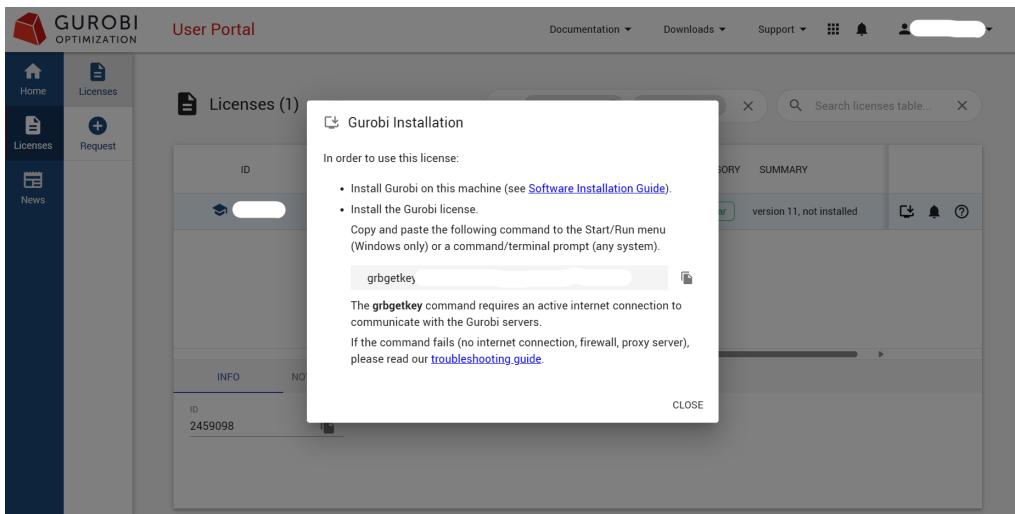


Figure 6: License matching guide

You should now paste the given command to the Start/Run menu, which looks something like Figure 7. To open this Start/Run menu, simply hold down the Windows key and press R on your keyboard. When you copy and paste the command and press OK, the menu should automatically open your Terminal and run the command. Your screen should now look something like Figure 8. We recommend saving the license file in the location suggested by the terminal (simply press Enter). If you prefer storing it elsewhere, you must create a new environment variable called `GRB_LICENSE_FILE` and set it to the right location. However, please do this only if you are familiar with the process.



Figure 7: Start/Run Menu

A screenshot of a Windows Command Prompt window titled 'Command Prompt'. The command entered is 'C:\Program Files\Gurobi>grbgetkey [REDACTED]'. The output shows Gurobi version 11.0.0, build v11.0.0rc2, running on a win64 platform (Windows 10.0). It contacts the license server, retrieves a license file for ID [REDACTED], and saves it to C:\Users\[REDACTED]\gurobi.lic. The prompt then asks for the directory where the license file should be stored, with the default being C:\Users\[REDACTED]. The final message shows the license file was written to C:\Users\[REDACTED]\gurobi.lic. The command prompt ends with 'C:\Program Files\Gurobi>'.

Figure 8: Window's Terminal application

If you carefully followed all the above steps, you are now ready to start using Gurobi!

## 1.2 How to get started with Gurobi in IntelliJ?

Now that we have successfully installed Gurobi, it is time to solve our first optimization problem. In the Gurobi installation folder, you can find example codes for various programming languages to get familiar with the language and the use of Gurobi. We will look at one of these examples (**Mip1.java**) to show how Gurobi can be used in IntelliJ. For this, you should navigate to the **gurobi1100** installation folder and go to **\win64\examples\java**. Then open IntelliJ and create a new Java project ‘GurobiExample’. Now go back to your Gurobi installation folder and paste the **Mip1.java** file under the **src** directory of your ‘GurobiExample’ project. When you go back to IntelliJ, the **Mip1.java** should be visible now, as in Figure 9.

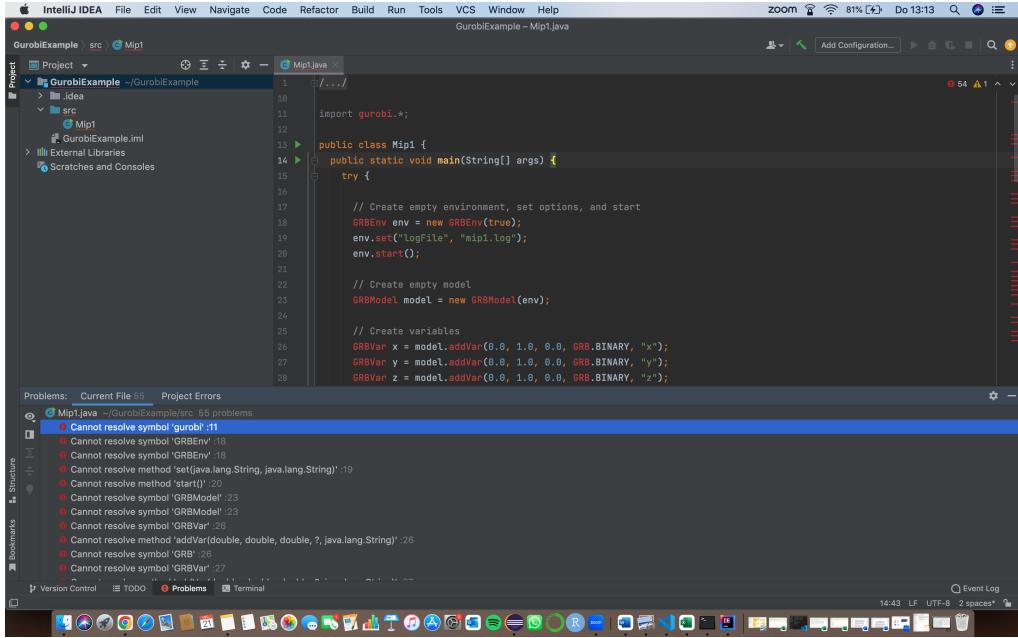


Figure 9: Java class error: Gurobi commands unrecognized

As you may have noticed, IntelliJ marks many errors (the red exclamation marks). This indicates that IntelliJ does not recognize the Gurobi commands yet. Therefore, we need to connect the Gurobi library with IntelliJ, a crucial step in this process. We need to go to **File-ProjectStructure-Libraries** to be able to fix this. When you navigated to the project structure tab (Figure 10), you should be able to access the Libraries tab (Figure 11).

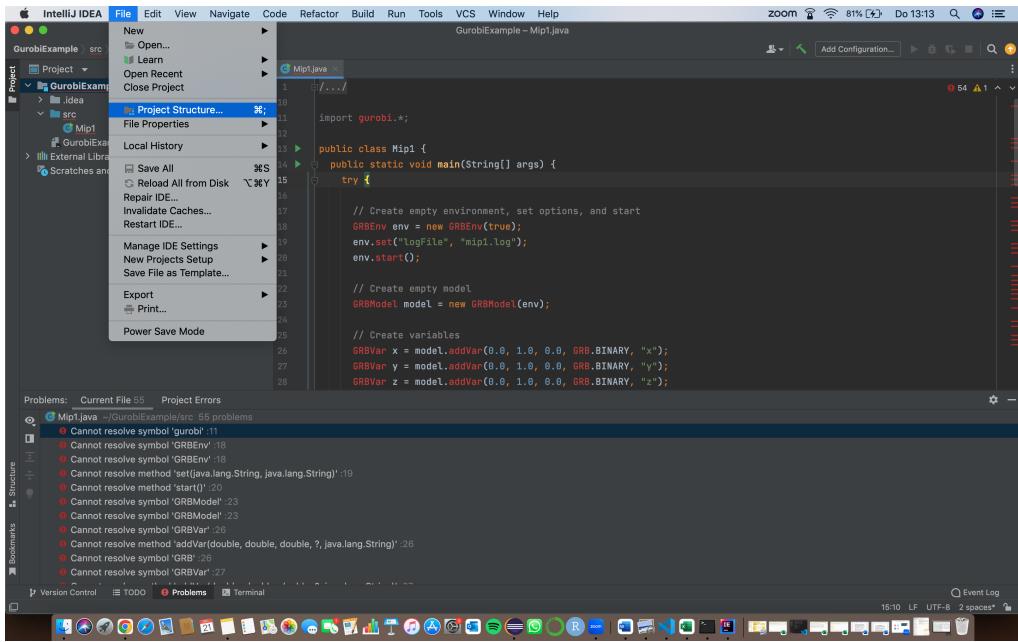


Figure 10: Go to Project Structure to fix it

In the Libraries tab, we must specify the Gurobi .jar file required for IntelliJ to recognize the commands. To this purpose, press +, choose Java, and navigate to \gurobi1100\windows64\lib\gurobi.jar and add it to IntelliJ. When you save your changes, go back to your Java Class: the errors disappeared! Note that for **every** project you want to create with Gurobi, you must link the library and ensure that the `import com.gurobi.gurobi.*;` command is present in every class.

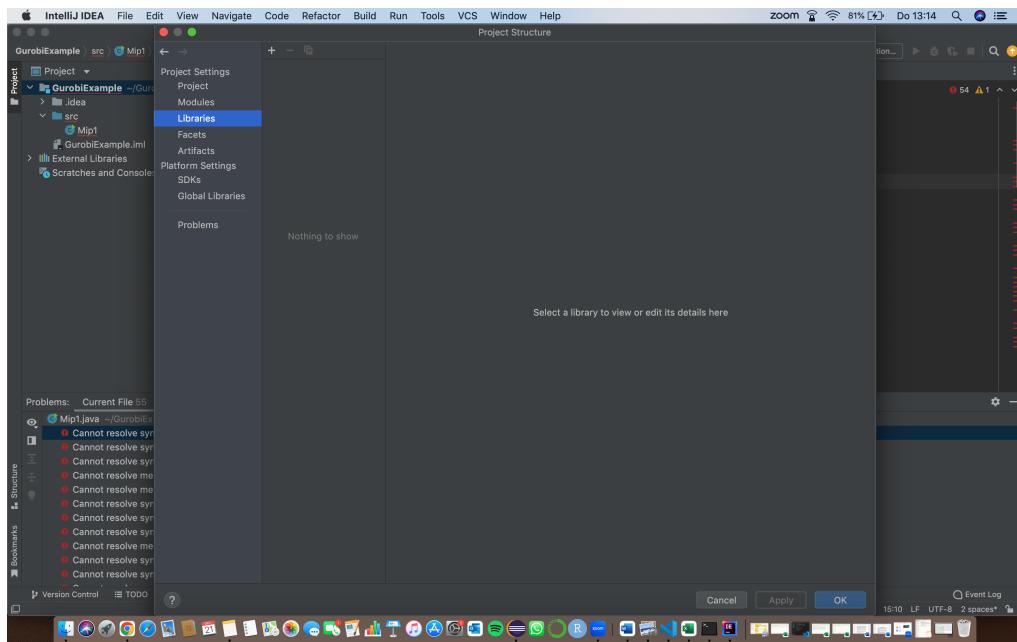


Figure 11: Add the gurobi.jar file in the Libraries tab

Well done! The final step is to run our program. This is a straightforward MIP with an objective value of 3.0 (try to see why). Later on, you will learn more about modeling these problems, but for now, Gurobi has been successfully installed on your Windows, and you now know how to link it with IntelliJ!

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** GurobiExample
- File:** Mip1.java
- Code:** The code implements a simple Mixed Integer Programming (MIP) model using the Gurobi Java API. It creates an environment, adds variables (x, y, z) with binary types and bounds [0, 1], and sets a tolerance of 1.00e-04. The run output shows an optimal solution found with objective value 3.00000000000e+00.
- Run:** Mip1
- Output:** Solution count 2: 3.0  
Optimal solution found (tolerance 1.00e-04)  
Best objective 3.00000000000e+00, best bound 3.00000000000e+00, gap 0.0000%  
x 1.0  
y 0.0  
z 1.0  
Obj: 3.0  
Process finished with exit code 0
- Bottom Bar:** Shows various system icons and the status bar indicating the build completed successfully in 11 sec, 85 ms (moments ago).

Figure 12: Errors disappeared and Java class works well