

1 Gurobi Installation Guide - Mac

This handout guides you through the process of installing Gurobi on your MacBook. 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 Mac and then explore how to use it in IntelliJ for all your modeling tasks.

1.1 Get Gurobi on your Mac

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”.

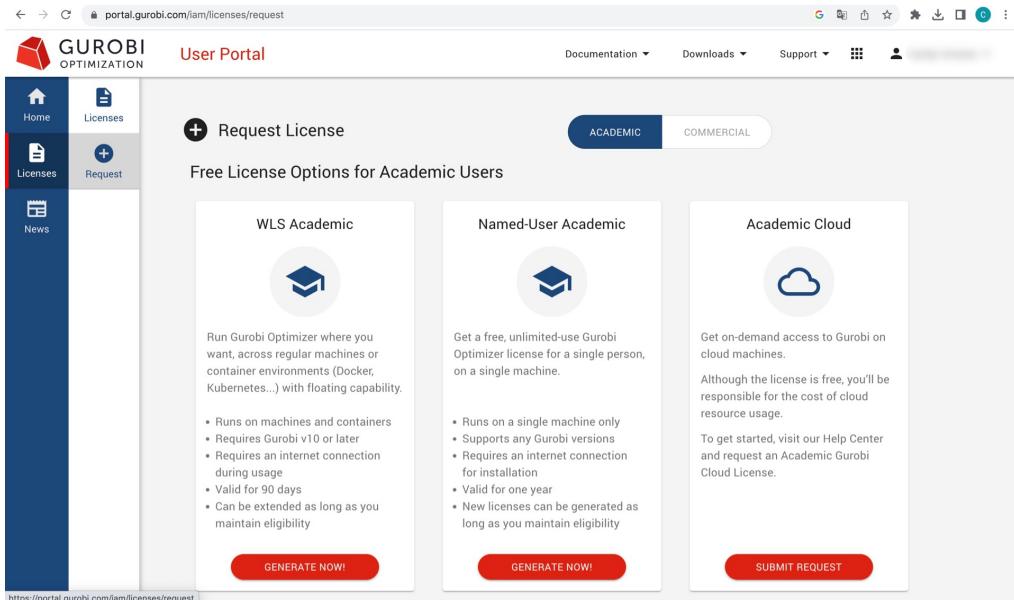


Figure 1: Request Gurobi license

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 | SUMM |
|------------|---------------|---------|-----------------------|-----------------------|----------|---------------|
| [REDACTED] | Free Academic | Trial | Nov 11, 2023 | Nov 11, 2024 | Regular | version MacBr |

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 Mac. For this, you need to go to <https://www.gurobi.com/downloads/gurobi-software/> and select the **macOS Universal2 - Installer** version, which is indicated with blue in Figure 3. Once you click on the .pkg 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 |

Gurobi Remote Services
Gurobi Remote Services is an optional set of Gurobi features that allow a cluster of one or more machines to perform Gurobi computations on behalf of other

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 **Continue** (NL: Ga door).

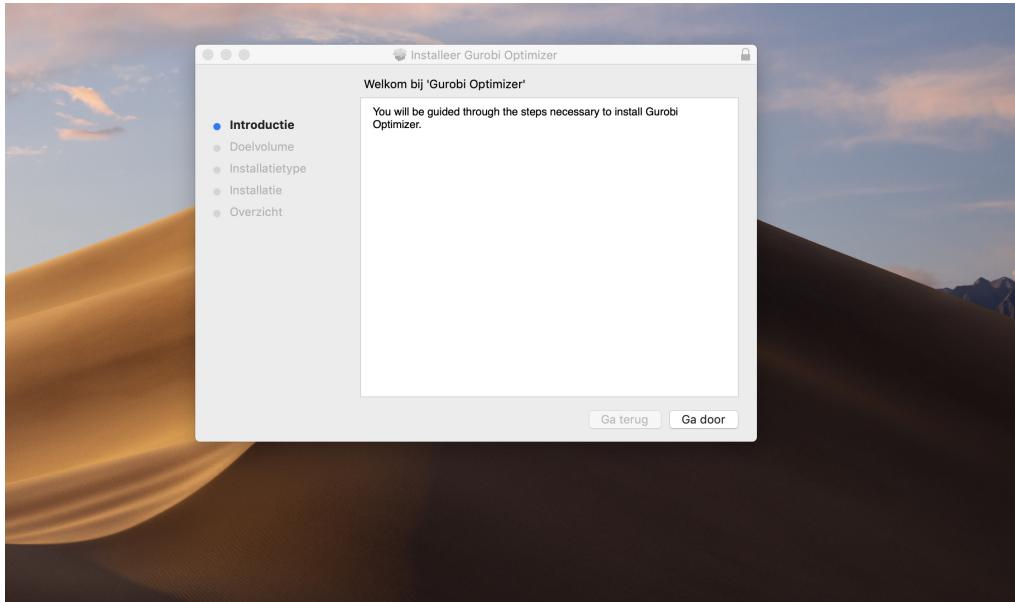


Figure 4: Mac installation prompt

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

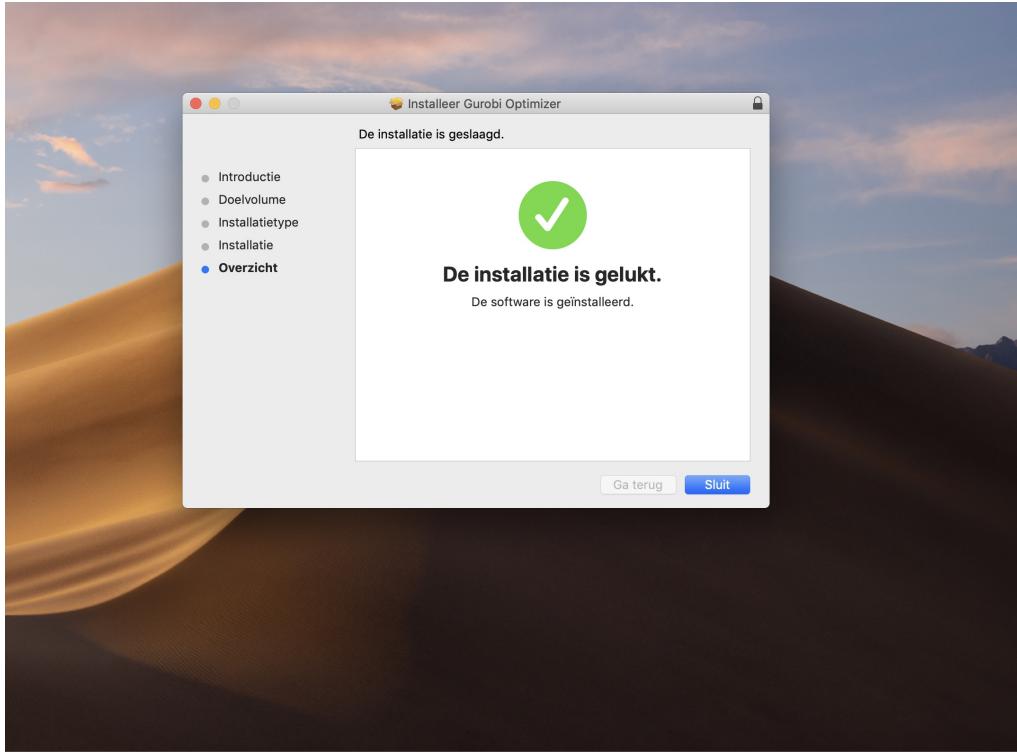


Figure 5: Mac installation finished

By default, Gurobi will be installed in your `/Library/gurobi1100/macos_universal2` directory (NL: `/Bibliotheek/gurobi1100/macos_universal2`). Note note that this is the system's `/Library` directory, not your personal `~/Library` directory.

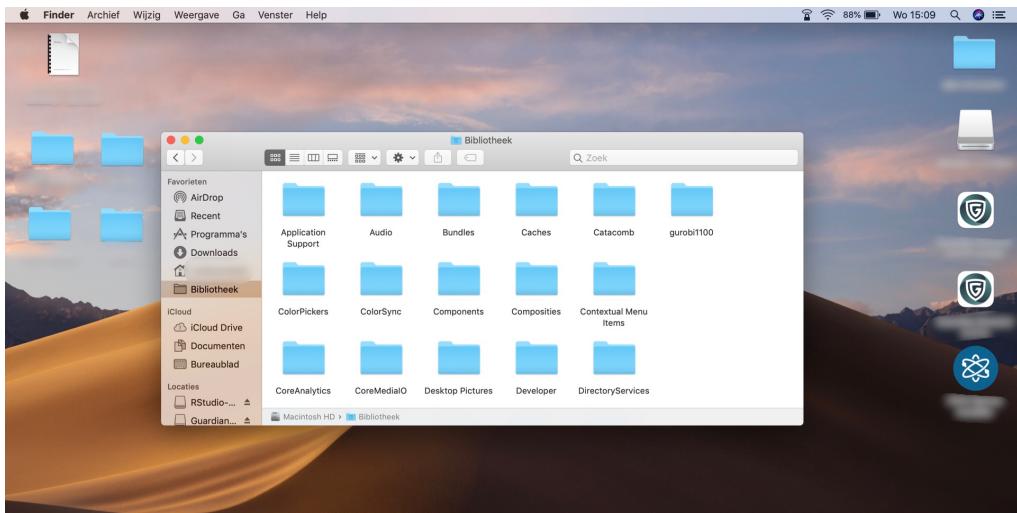


Figure 6: Gurobi installation folder under Mac's Library

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** button that is shown under your Free Academic license type. This will prompt a window explaining how to link the academic license. Since we already installed Gurobi, we can proceed directly to the linking process through the Mac Terminal.

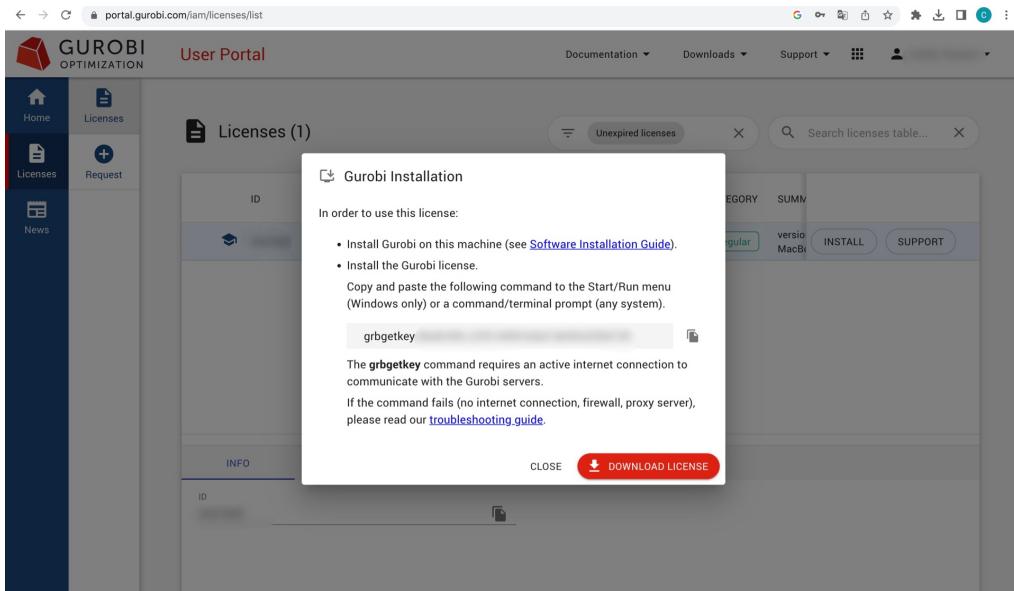


Figure 7: License matching guide

Copy the key that is provided (**grbgetkey YOURKEY**). We will need this in the Terminal. Since you might not be very familiar with the Mac Terminal, Figure 8 shows you what the application symbol looks like. You can always find the terminal by searching through your toolbar or looking in your application overview. Clicking on the terminal opens a new empty terminal window. If the window is already filled with code, open a new window (just like how you would do it for other applications).

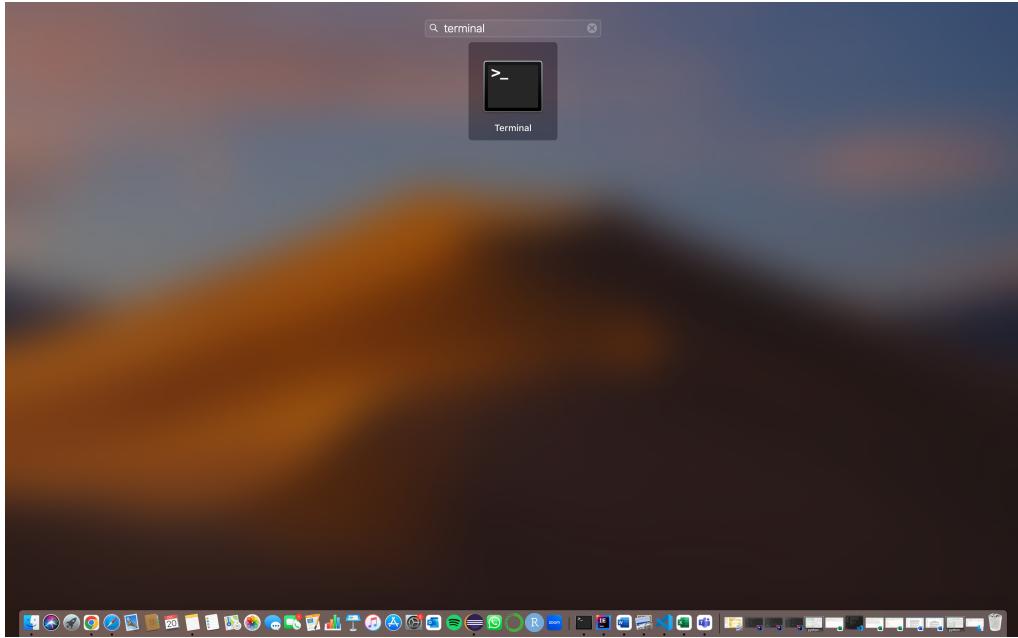


Figure 8: Mac's Terminal application

When you are in the Terminal, you only need to paste your **grbgetkey YOURKEY** in the Terminal window as shown in Figure 13. We recommend saving the license file in the location suggested by the terminal (simply open the Terminal, paste your **grbgetkey YOURKEY**, and 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.

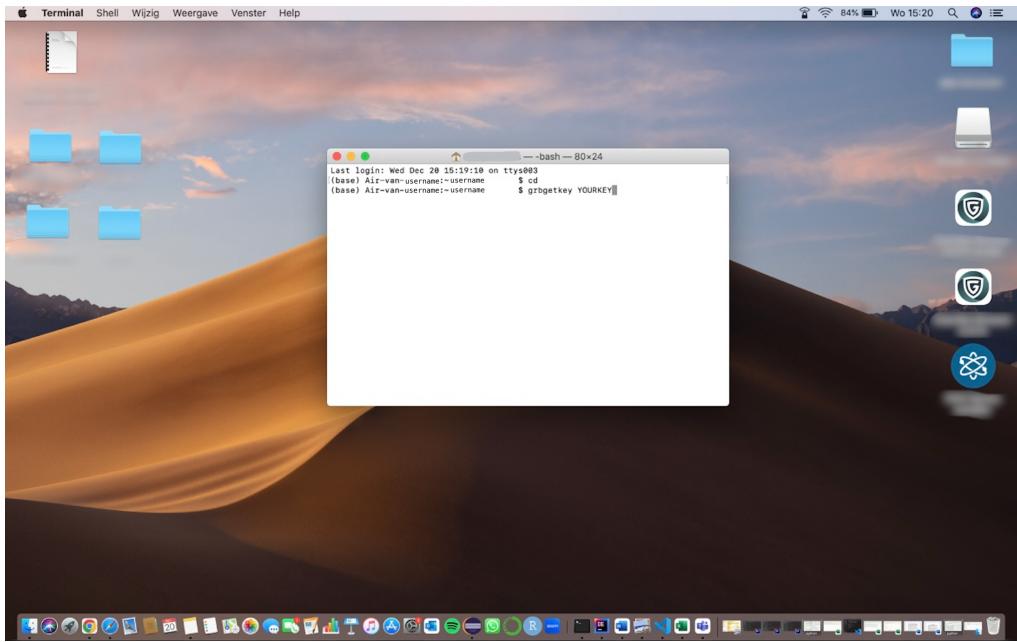


Figure 9: Terminal input to match license

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 **/macos_universal2/examples/java**. Then open IntelliJ and create a new Java project ‘GurobiExample’. Now go back to your Finder 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 10.

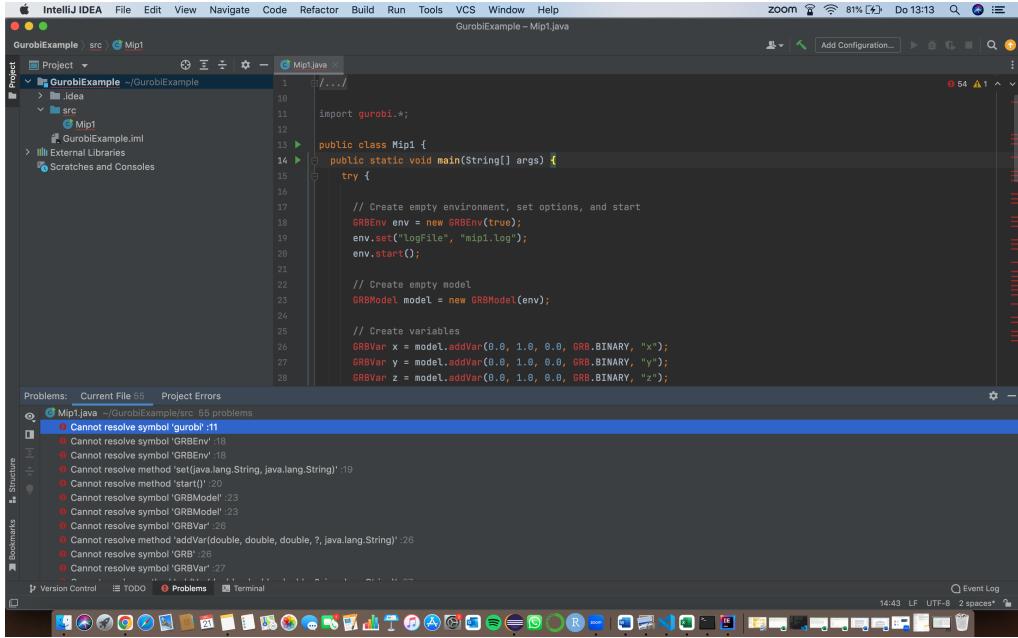


Figure 10: Java class error: Gurobi commands unrecognized

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 11), you should be able to access the Libraries tab (Figure 12).

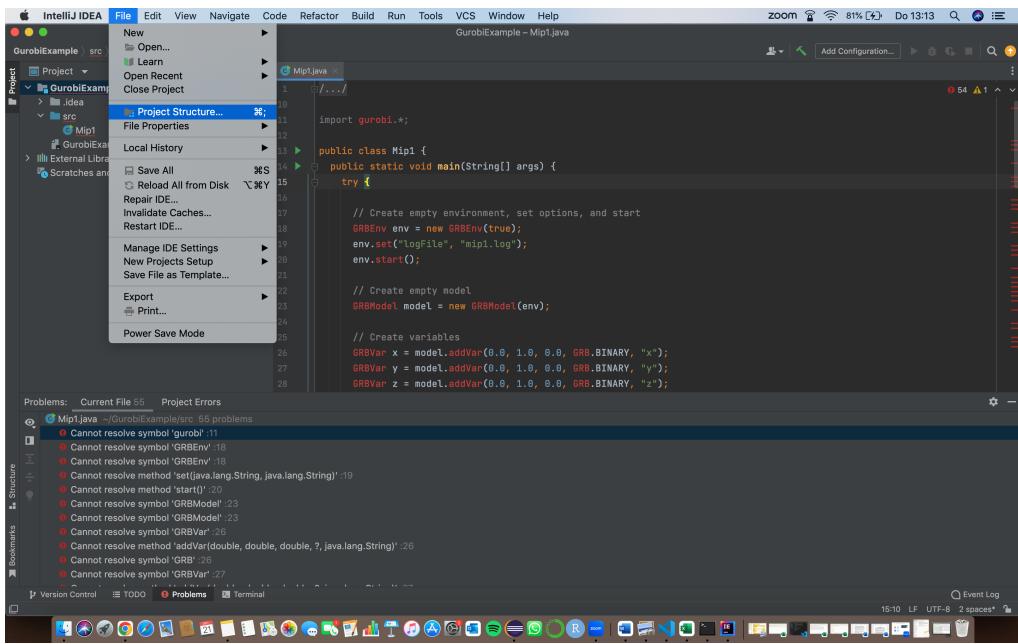


Figure 11: Go to Project Structure to fix it

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/macos_universal2/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.

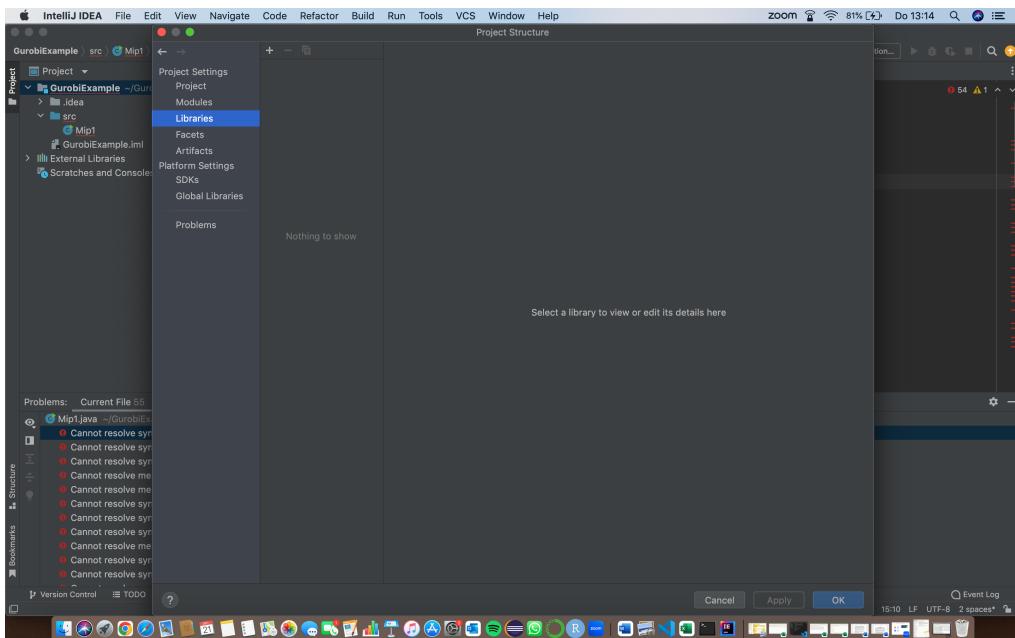


Figure 12: Add the gurobi.jar file in the Libraries tab

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 Mac, 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 an objective function. The code then solves the model and prints the solution.
- Run:** The run configuration is set to "Mip1". The output window shows:
 - 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
- Event Log:** Shows a successful build completed in 11 seconds.

Figure 13: Errors disappeared and Java class works well