MINISTRY OF EDUCATION

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

FUNDAMENTAL PROGRAMMING TECHNIQUES

Laboratory Resources

Tudor Cioara
Claudia Daniela Pop

Ionut Anghel
Mitrea Dan

Marcel Antal
Cristina Bianca Pop

2023

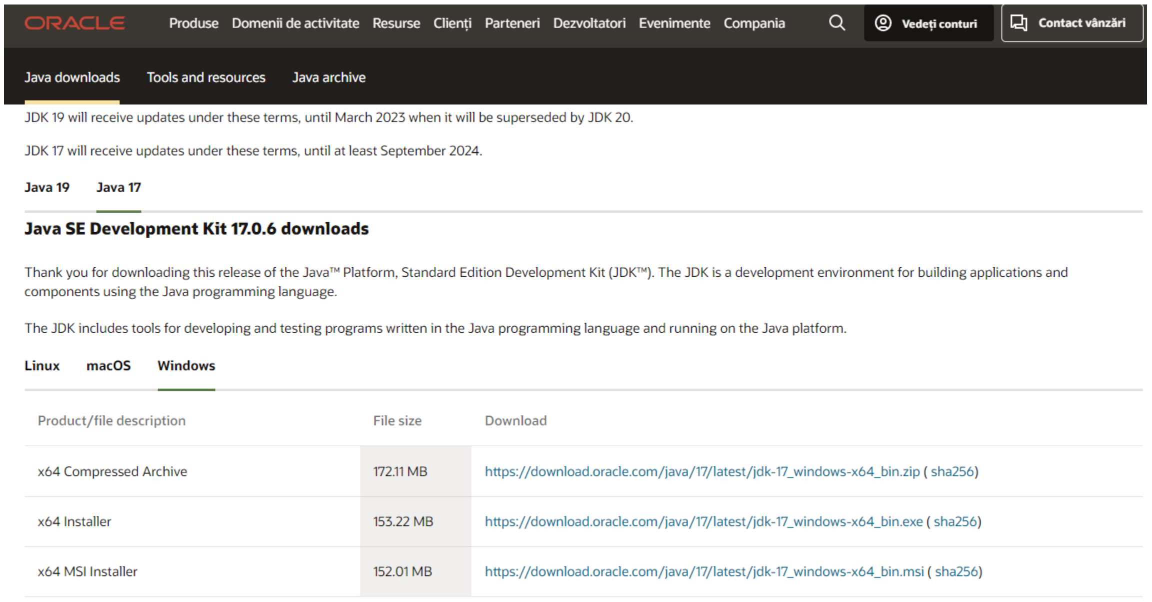
Contents

1 Java

1.1 Java JDK

1) Access the next link:

https://www.oracle.com/ro/java/technologies/downloads/#java17



JDK 19 will receive updates under these terms, until March 2023 when it will be superseded by JDK 20.

JDK 17 will receive updates under these terms, until at least September 2024.

Java SE Development Kit 17.0.6 downloads

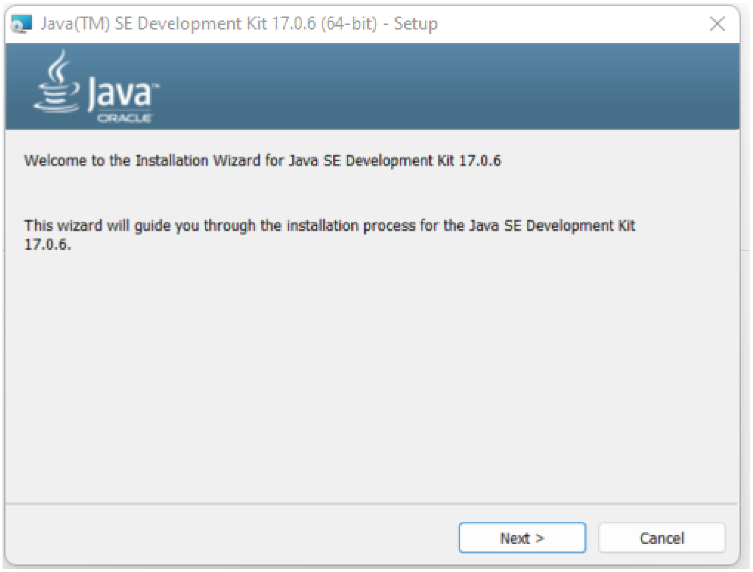
Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Product/file description	File size	Download
x64 Compressed Archive	172.11 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip (sha256)
x64 Installer	153.22 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe (sha256)
x64 MSI Installer	152.01 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi (sha256)

2) Click on the link which corresponds to your version of the Operating System. In the example the version which is used corresponds to Windows x64 and the file is named https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe.

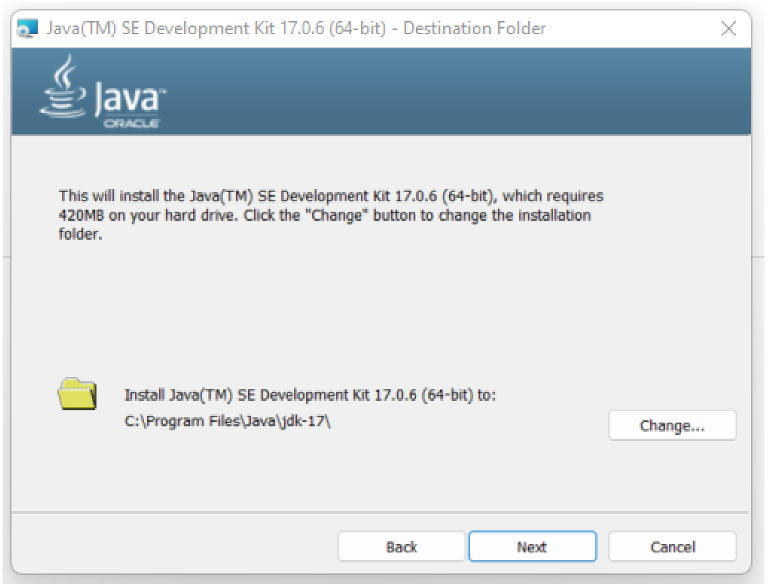
3) Click on *jdk-17_windows-x64_bin.exe*. The window below will appear. Click on the *Next* button.



Welcome to the Installation Wizard for Java SE Development Kit 17.0.6

This wizard will guide you through the installation process for the Java SE Development Kit 17.0.6.

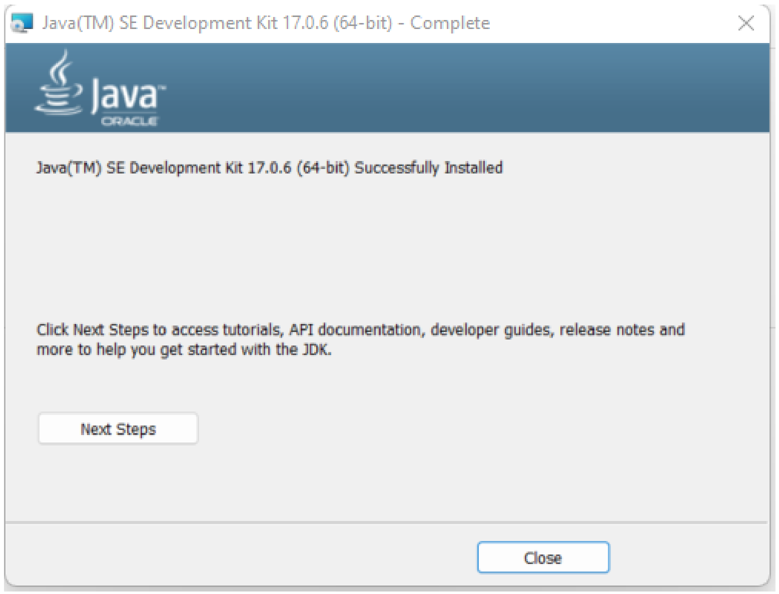
4) You will be asked where you want to install Java. Use the default location and click Next.


Java(TM) SE Development Kit 17.0.6 (64-bit) - Destination Folder
Java
ORACLE
Change...
Back
Next
Cancel

This will install the Java(TM) SE Development Kit 17.0.6 (64-bit), which requires 420MB on your hard drive. Click the "Change" button to change the installation folder.

Install Java(TM) SE Development Kit 17.0.6 (64-bit) to:
C:\Program Files\Java\jdk-17\

5) After the installation is successfully completed the window below will be displayed. Click Close.



Java(TM) SE Development Kit 17.0.6 (64-bit) Successfully Installed

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

1.2 Set the JAVA_HOME variable

Note: the steps may vary according to the Windows version installed on the computer

1) Click *Start*.
2) Right-Click on *This PC*.
3) Select *Properties*.

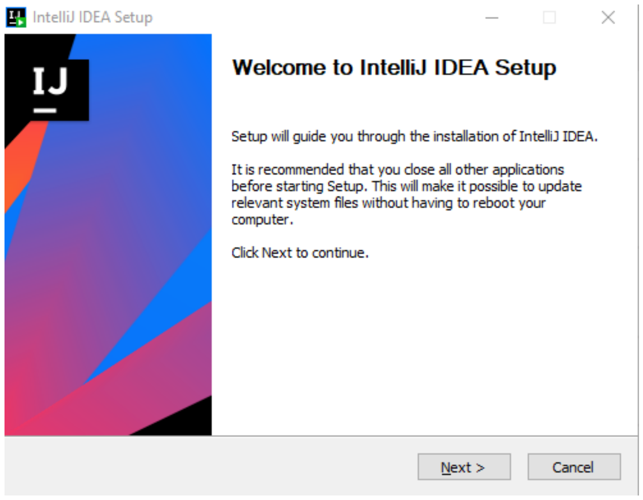
4) Click on *Advanced System Settings*.
5) Click on *Environment Variables*.
6) Under *System Variables* click *New*.
7) In the text field associated with the name of the variable insert JAVA_HOME and in the field associated with the value of the variable insert C:\Program Files\Java\java_version;.
8) Click *OK*.

2. IntelliJ IDEA

We recommend using IntelliJ IDEA as the IDE for developing your applications during the laboratories. As a student, you can benefit of a free educational license for IntelliJ IDEA, by applying here https://www.jetbrains.com/community/education/#students -> Click on *Apply now* -> Fill in the requested information and click on *Apply for free products.* After obtaining the license, download the latest version of Intellij IDEA. Click on the downloaded file. You will be asked whether you allow the program to make changes to this computer -> Click Yes.

Then, follow the steps below:

1) Click next



IntelliJ IDEA Setup

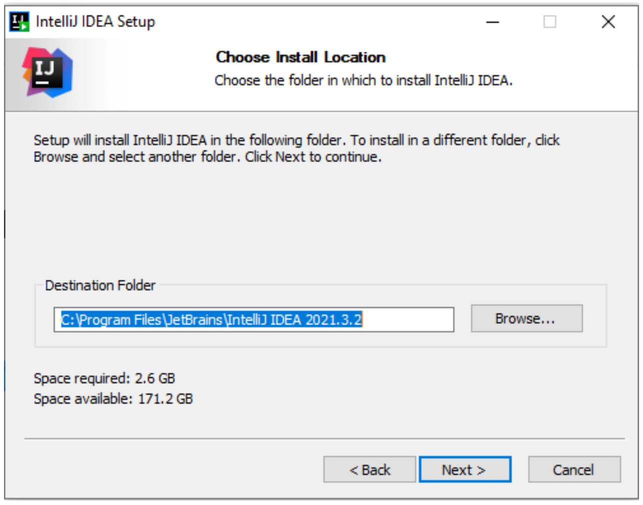
Welcome to IntelliJ IDEA Setup

Setup will guide you through the installation of IntelliJ IDEA.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

2) You will be asked where you want to install Intellij IDEA. Use the default location and click Next.



IntelliJ IDEA Setup

Choose Install Location

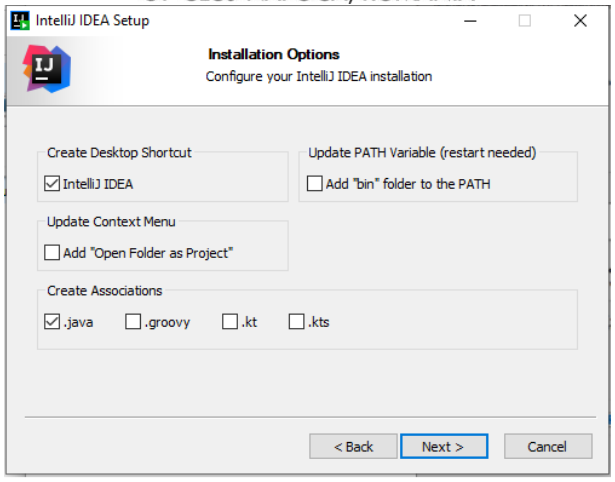
Choose the folder in which to install IntelliJ IDEA.

Setup will install IntelliJ IDEA in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2

Space required: 2.6 GB
Space available: 171.2 GB

3) Configure the installation options (see an example in the image below) and click Next.


IntelliJ IDEA Setup

Installation Options
Configure your IntelliJ IDEA installation

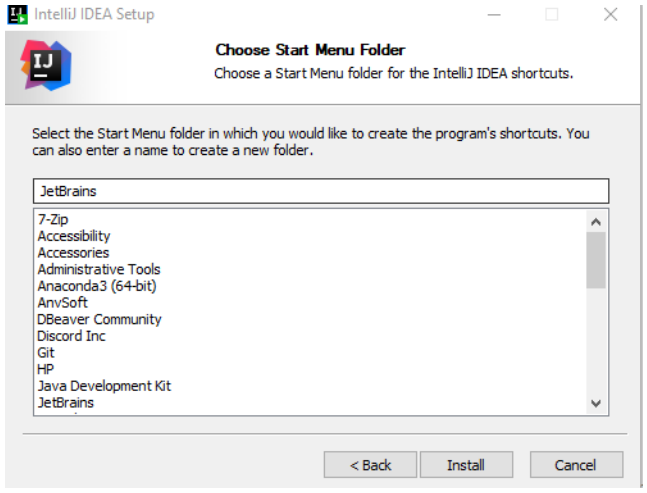
Create Desktop Shortcut
[✓] IntelliJ IDEA

Update PATH Variable (restart needed)
[ ] Add "bin" folder to the PATH

Update Context Menu
[ ] Add "Open Folder as Project"

Create Associations
[✓] .java [ ] .groovy [ ] .kt [ ] .kts

4) Click Install.


IntelliJ IDEA Setup

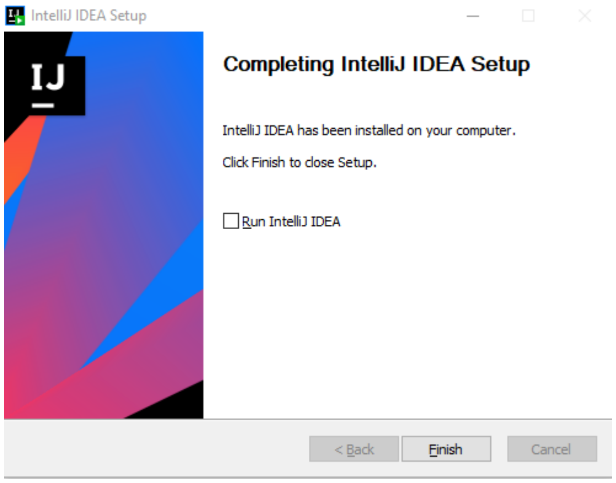
Choose Start Menu Folder
Choose a Start Menu folder for the IntelliJ IDEA shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

JetBrains

7-Zip
Accessibility
Accessories
Administrative Tools
Anaconda3 (64-bit)
AnvSoft
DBeaver Community
Discord Inc
Git
HP
Java Development Kit
JetBrains

5) After the installation completes click on *Finish.*


IntelliJ IDEA Setup

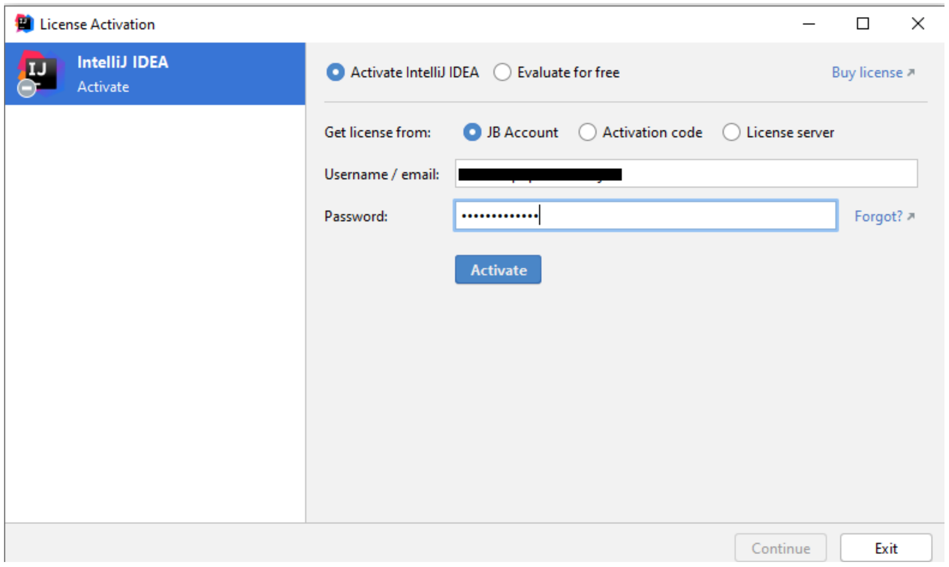
Completing IntelliJ IDEA Setup

IntelliJ IDEA has been installed on your computer.

Click Finish to close Setup.

[ ] Run IntelliJ IDEA

6) Run IntelliJ IDEA. When you first run IntelliJ IDEA after the installation completes you will have to activate it by inserting the username/email and password you used when creating the account for obtaining the students license. Click *Activate*.



License Activation

IntelliJ IDEA
Activate

Activate IntelliJ IDEA [✓] Evaluate for free
Buy license ↗

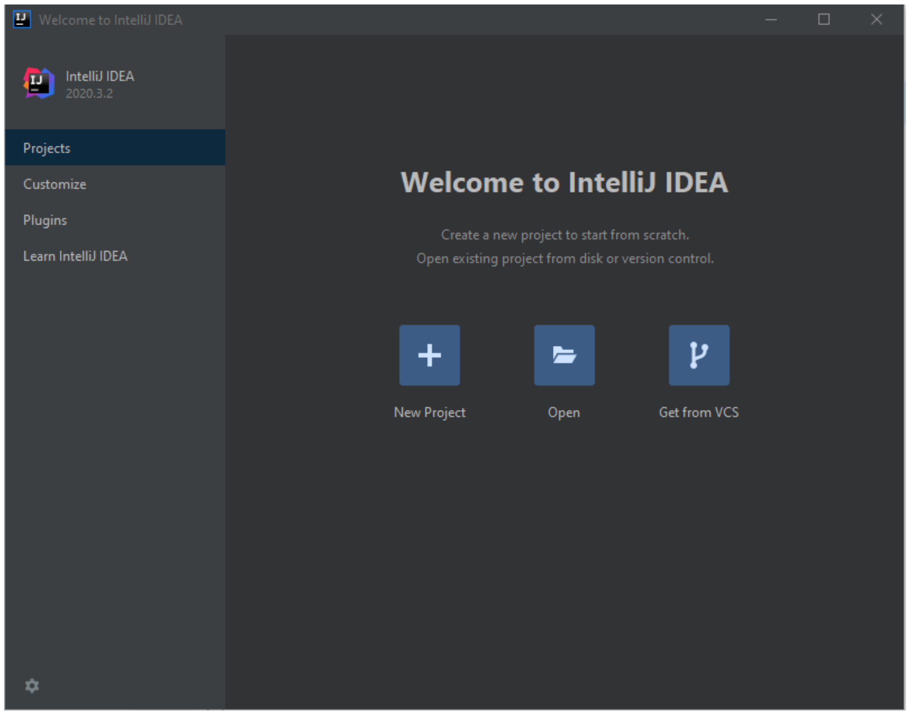
Get license from: JB Account [✓] Activation code License server

Username / email: [redacted]

Password: ••••••••••••••

Forgot? ↗

7) After the activation completes, the IDE will open.



Welcome to IntelliJ IDEA

IntelliJ IDEA
2020.3.2

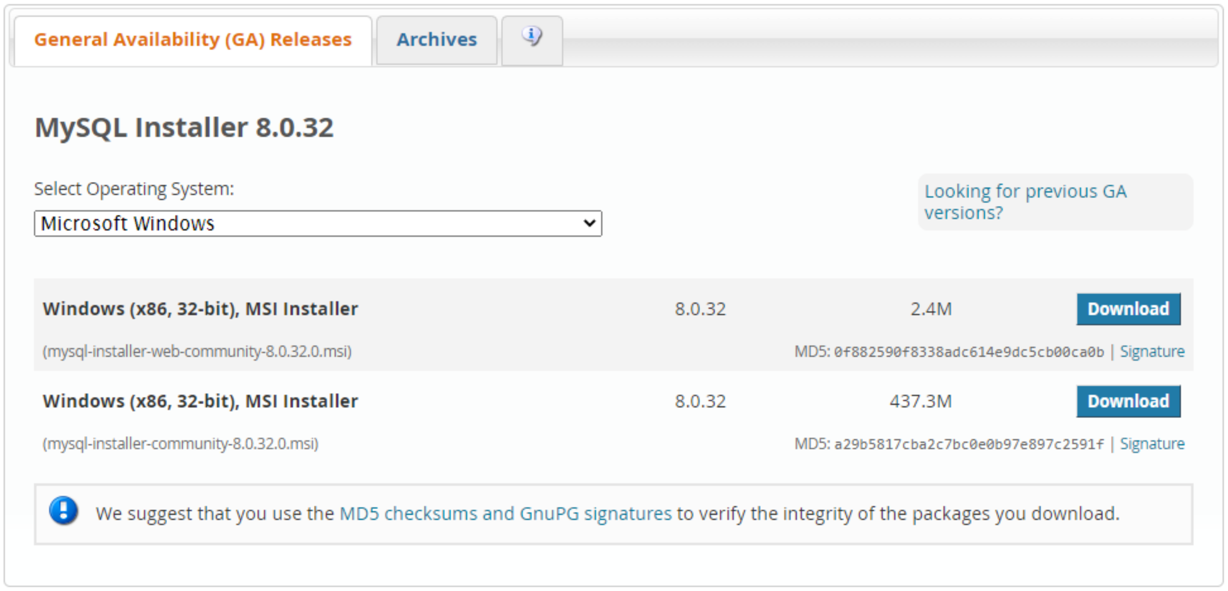
Welcome to IntelliJ IDEA

Create a new project to start from scratch.
Open existing project from disk or version control.

3. Database Server

A database server must be installed on the local machine. A MySQL server can be used to run locally the projects. For installing the MySQL server follow the next steps:

1) Click on the next link: https://dev.mysql.com/downloads/windows/installer/.



MySQL Installer 8.0.32

Select Operating System: Microsoft Windows

Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.32.0.msi)	8.0.32	2.4M	Download MD5: 0f882590f8338adc614e9dc5cb00ca0b | Signature
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.32.0.msi)	8.0.32	437.3M	Download MD5: a29b5817cba2c7bc0e0b97e897c2591f | Signature

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

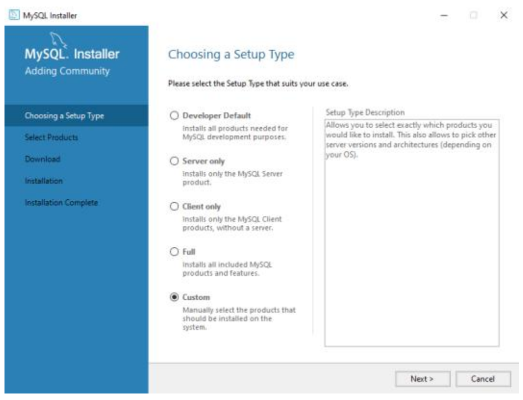
2) Click on the second Download button.

3) Click on *No thanks, just start my download*.

4) Click on the downloaded file.

5) You will be asked whether you allow the program to make changes to this computer -> Click *Yes*.

6) You will be asked to select the Setup Type that suits your use case. Select *Custom* and click *Next*.



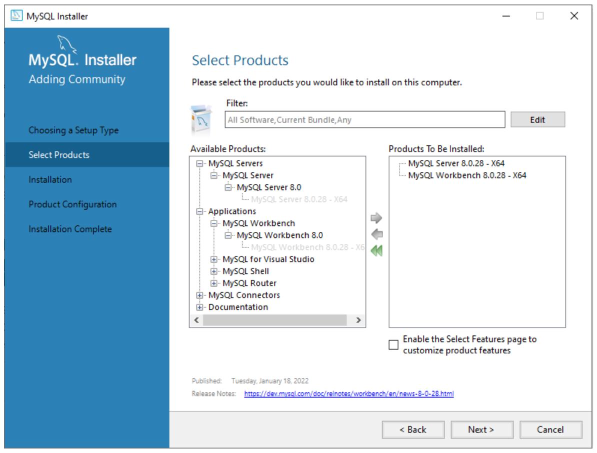
Choosing a Setup Type

Please select the Setup Type that suits your use case.

- ○ Developer Default: Installs all products needed for MySQL development purposes.
- ○ Server only: Installs only the MySQL Server product.
- ○ Client only: Installs only the MySQL Client products, without a server.
- ○ Full: Installs all included MySQL products and features.
- ● Custom: Manually select the products that should be installed on the system.

Setup Type Description: Allows you to select exactly which products you would like to install. This also allows to pick other server versions and architectures (depending on your OS).

7) You will be redirected to “Select Products and Features”. Select “MySQL Server *version*” and “MySQL Workbench *version*” and click *Next*.


MySQL Installer
MySQL. Installer
Adding Community

Choosing a Setup Type
Select Products
Installation
Product Configuration
Installation Complete

Select Products

Please select the products you would like to install on this computer.

Filter: All Software,Current Bundle,Any

Available Products:
- MySQL Servers
  - MySQL Server
    - MySQL Server 8.0
      - MySQL Server 8.0.28 - X64
- Applications
  - MySQL Workbench
    - MySQL Workbench 8.0
      - MySQL Workbench 8.0.28 - X6
  - MySQL for Visual Studio
  - MySQL Shell
  - MySQL Router
- MySQL Connectors
- Documentation

Products To Be Installed:
- MySQL Server 8.0.28 - X64
- MySQL Workbench 8.0.28 - X64

☐ Enable the Select Features page to customize product features

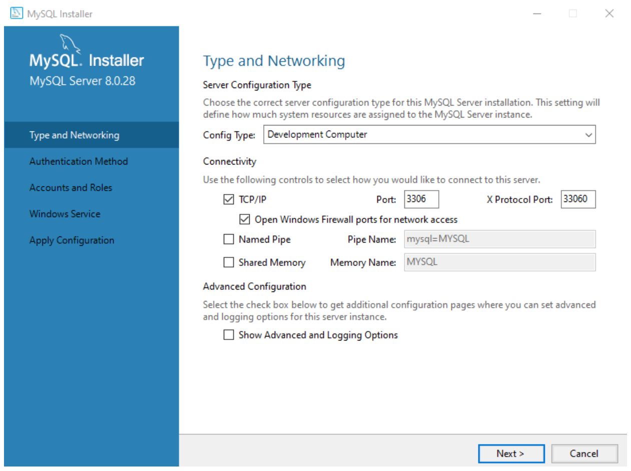
Published: Tuesday, January 18, 2022
Release Notes: https://dev.mysql.com/doc/relnotes/workbench/en/news-8-0-28.html

8) Click Next.

9) Click Execute.

10) Click Next.

11) Click Next and follow the steps for the configuration of the MySQL Server. At this stage pay attention to the port on which the MySQL server is running on (i.e., 3306 is the default port).


MySQL Installer
MySQL. Installer
MySQL Server 8.0.28

Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Apply Configuration

Type and Networking

Server Configuration Type

Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.

Config Type: Development Computer

Connectivity

Use the following controls to select how you would like to connect to this server.

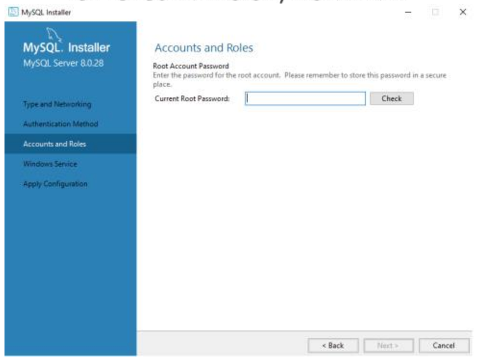
☑ TCP/IP Port: 3306 X Protocol Port: 33060
☑ Open Windows Firewall ports for network access
☐ Named Pipe Pipe Name: mysql=MYSQL
☐ Shared Memory Memory Name: MYSQL

Advanced Configuration

Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.

☐ Show Advanced and Logging Options

- Click *Next*
- Select the Authentication Mode (select *Use Strong Password Encryption for Authentication*) and click *Next*.
- Set the password for the root account – this must be set by you and make sure you remember it as it will be used for further connections to the MySQL server. After inserting the password click on the *Check* button.



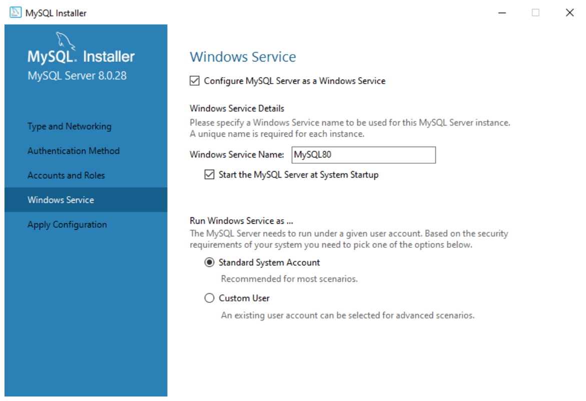
Accounts and Roles

Root Account Password

Enter the password for the root account. Please remember to store this password in a secure place.

Current Root Password:

- Click *Next*



Windows Service

☑ Configure MySQL Server as a Windows Service

Windows Service Details

Please specify a Windows Service name to be used for this MySQL Server instance. A unique name is required for each instance.

Windows Service Name: MySQL80

☑ Start the MySQL Server at System Startup

Run Windows Service as ...

The MySQL Server needs to run under a given user account. Based on the security requirements of your system you need to pick one of the options below.

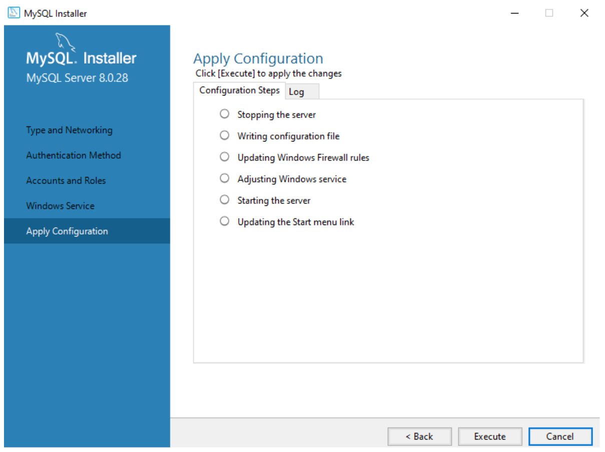
● Standard System Account

Recommended for most scenarios.

○ Custom User

An existing user account can be selected for advanced scenarios.

- Click *Next* and then *Execute*



Apply Configuration

Click [Execute] to apply the changes

○ Stopping the server
○ Writing configuration file
○ Updating Windows Firewall rules
○ Adjusting Windows service
○ Starting the server
○ Updating the Start menu link

12) Click Finish.

Note: in case you uninstall MySQL, follow the steps presented here to completely uninstall it.

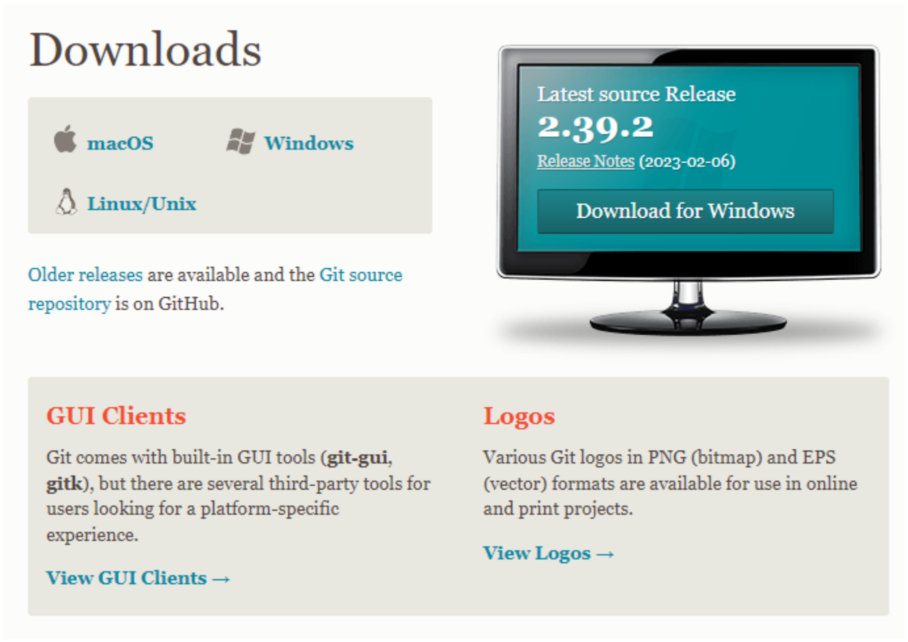
4 Git and Maven Projects

Modern software projects development requires effort from large teams of developers that have to collaborate in order to integrate their work and create the final product. Coordinating and tracking the changes between multiple source files is a difficult task, thus an automatic tool was developed in 2005, initially for the development of the Linux Kernel, and, since then, it has penetrated all levels of software development. The tool is a version control system (VCS), named GIT, which tracks changes of computer files and helps coordinating several people who work on those files.

Furthermore, large applications often encounter problems with the software build settings as well as with the dependency description. To address these problems, another automatic tool was created, MAVEN that defines conventions for the build procedure and uses an XML file to describe the software project, dependencies, external modules, components and plug-ins.

4.1 Git installation

1) Click on https://git-scm.com/downloads.
2) Select your operating system.



Downloads

macOS Windows Linux/Unix

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.39.2
Release Notes (2023-02-06)
Download for Windows

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

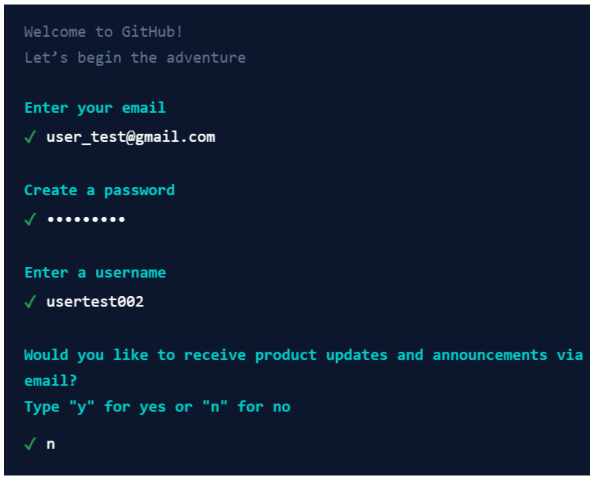
View Logos →

3) If you select Windows, a file called *Git-version-64-bit.exe* should be downloaded. In the case you select another operating system or if your system is on 32 bits then a file with a similar name should be downloaded.
4) Click on this file and follow the default installation guidelines, except for the step where you are asked which terminal emulator you want to use. Select the second option.

4.2 Create Account on GitHub

Click on https://github.com/

1) In the right corner, click on Sign Up. You will be asked to introduce your personal information. Or, if you already have an account, just *Sign In.*



```
Welcome to GitHub!
Let's begin the adventure

Enter your email
✓ user_test@gmail.com

Create a password
✓ •••••••••

Enter a username
✓ usertest002

Would you like to receive product updates and announcements via
email?
Type "y" for yes or "n" for no

✓ n
```

2) You will receive a lunch code on your email which you must introduce in the next window

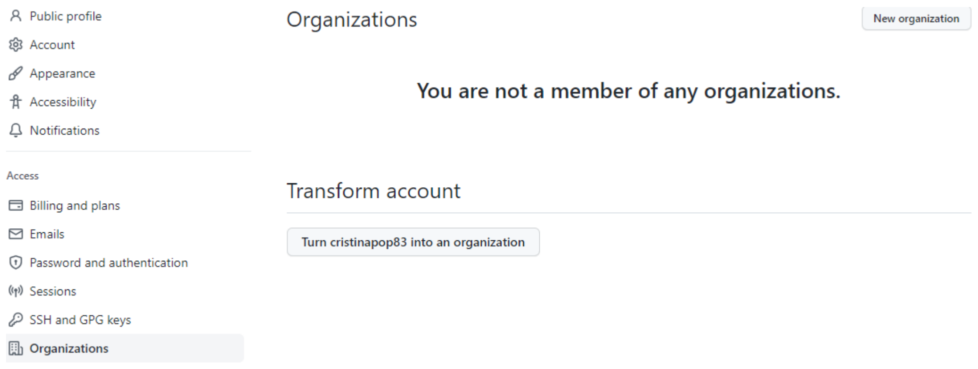
3) You can choose the "Just me" and "Student" options.

4) In the next window, you should check at least the "Collaborative Coding"

5) In the next step, you can choose the "Continue for free" option

6) In the right corner click on your profile photo and go to *Settings*

7) Go to the *Organizations* and click on "New organization"



Organizations

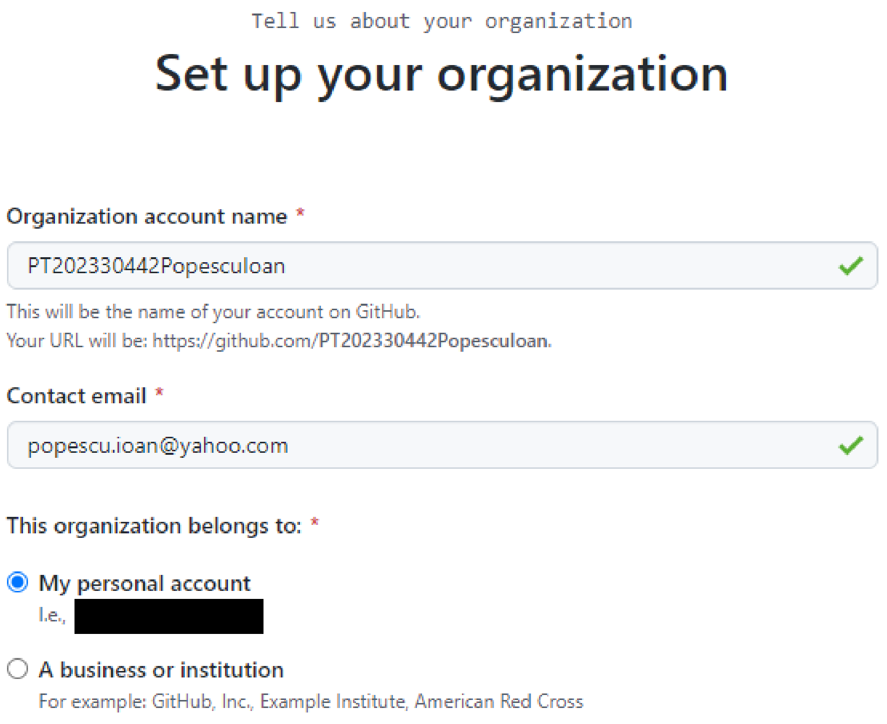
You are not a member of any organizations.

Transform account

8) After choosing "Create a free organization", set the organization account name to PT2023*GroupNumberLastNameFirstName*

(e.g., PT202330442PopescuIoan)

9) Set a Contact email and chose the option My personal account for “This organization belongs to”



Tell us about your organization

Set up your organization

Organization account name *

PT202330442Popesculoan ✓

This will be the name of your account on GitHub.
Your URL will be: https://github.com/PT202330442Popesculoan.

Contact email *

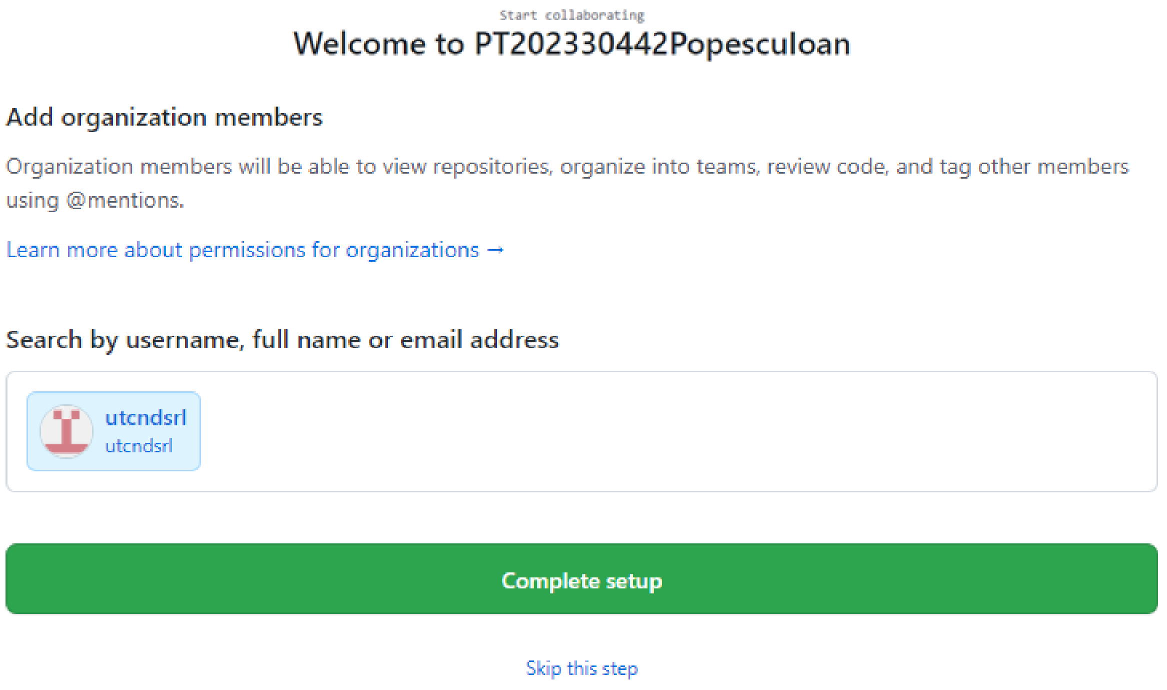
popescu.ioan@yahoo.com ✓

This organization belongs to: *

[✓ My personal account]
I.e., [redacted]

A business or institution
For example: GitHub, Inc., Example Institute, American Red Cross

10) In the next window you must add the *utcndsrl* account as a member by searching the username. Click on Complete Setup.



Start collaborating

Welcome to PT202330442Popesculoan

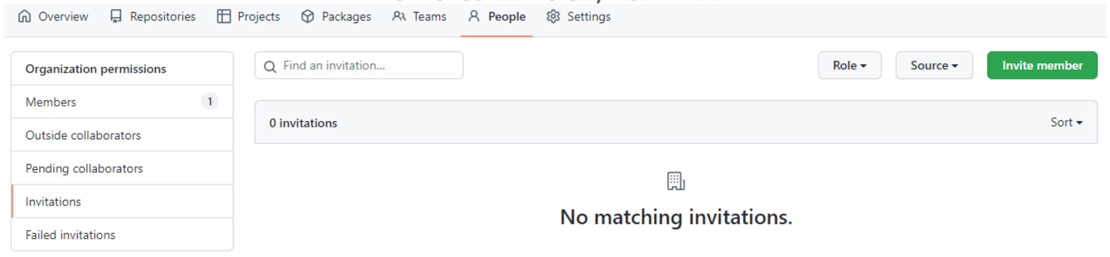
Add organization members

Organization members will be able to view repositories, organize into teams, review code, and tag other members using @mentions.

Learn more about permissions for organizations →

Search by username, full name or email address

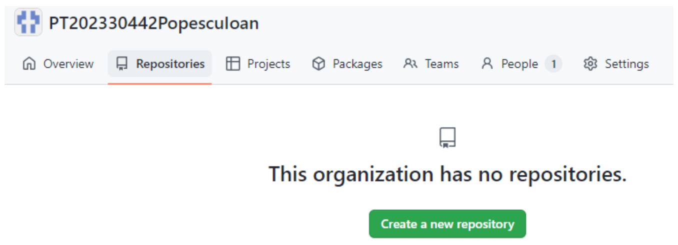
utcndsrl
utcndsrl



0 invitations

No matching invitations.

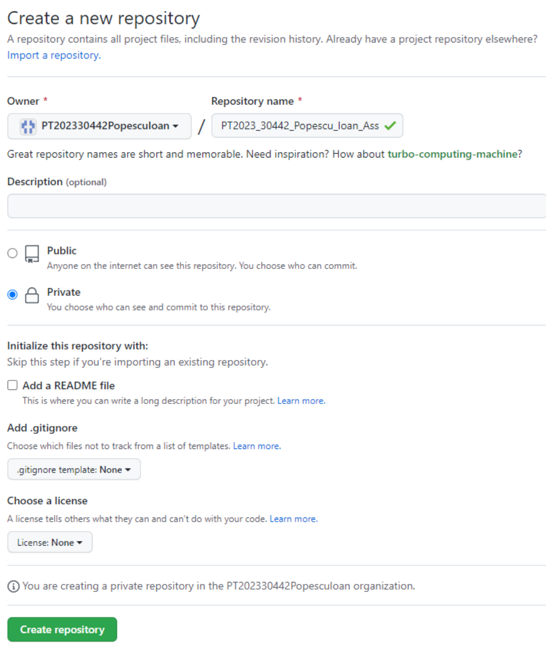
11) Inside the organization, you can create your own PRIVATE repositories for different applications of the PT lab. To create a repository, click on the *Repositories* tab and then on the *Create a new repository* button:



This organization has no repositories.

Insert the name of the repository and remember to keep the same naming conventions for the repositories, considering the following formats:

PT2023_GroupNumber_LastName_FirstName_Assignment_Number



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner *: PT202330442PopescuIoan / Repository name *: PT2023_30442_Popescu_Ioan_Ass ✓

Great repository names are short and memorable. Need inspiration? How about turbo-computing-machine?

Description (optional)

○ Public
Anyone on the internet can see this repository. You choose who can commit.

● Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

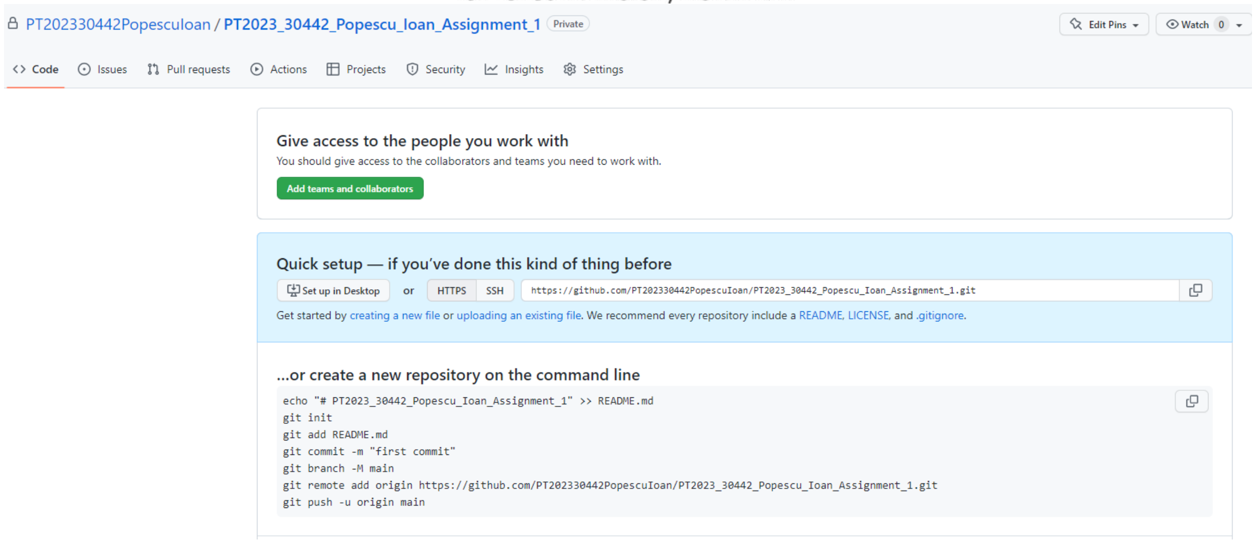
☐ Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. Learn more.
License: None

You are creating a private repository in the PT202330442PopescuIoan organization.

Click on the *Create repository* button. Consequently, you should be redirected to a page similar to the one below:



Give access to the people you work with

You should give access to the collaborators and teams you need to work with.

Quick setup — if you've done this kind of thing before

https://github.com/PT202330442PopescuIoan/PT2023_30442_Popescu_Ioan_Assignment_1.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# PT2023_30442_Popescu_Ioan_Assignment_1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/PT202330442PopescuIoan/PT2023_30442_Popescu_Ioan_Assignment_1.git
git push -u origin main
```

4.3 Basic Instructions

4.3.1 Create a project from scratch

1) Create the folder *PT2023_GroupNumber_LastName_FirstName_Assignment_Number* on *D:\*.
2) Right click on this folder and click *Git Bash Here*.
3) Write the next commands:

   a) git init

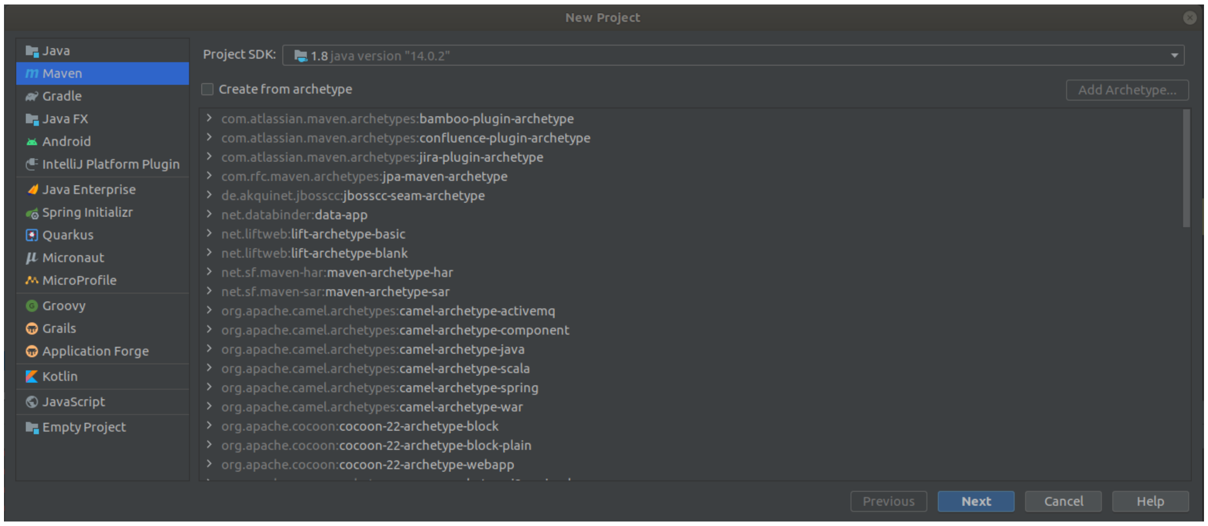
Note: the command will create a local repository with a default main branch. Notice that after executing this command, a hidden .git folder will be created inside the current folder representing the repository where git stores all necessary files. From now on, it will be possible to track all the changes that will be performed to the files from the original folder. The original folder is considered to be the working directory, while the .git folder is referred as the repository that tracks the made changes (for more details check this link).

   b) **git remote add origin** *https://github.com/organization_name/repository_name.git*

For example:
git remote add origin https://github.com/PT202330442PopescuIoan/PT2023_30442_Popescu_Ioan_Assignment_1.git

Note: this command will connect the local repository created using the "git init" command with the remote repository's origin (for more details check this link).

4) Open IntelliJ, select *File -> New -> Project*. Then choose *Maven* from the list from the left, choose the appropriate Java SDK, then click on *Next*.



Project SDK: 1.8 java version "14.0.2"

Create from archetype

```
com.atlassian.maven.archetypes:bamboo-plugin-archetype
com.atlassian.maven.archetypes:confluence-plugin-archetype
com.atlassian.maven.archetypes:jira-plugin-archetype
com.rfc.maven.archetypes:jpa-maven-archetype
de.akquinet.jbosscc:jbosscc-seam-archetype
net.databinder:data-app
net.liftweb:lift-archetype-basic
net.liftweb:lift-archetype-blank
net.sf.maven-har:maven-archetype-har
net.sf.maven-sar:maven-archetype-sar
org.apache.camel.archetypes:camel-archetype-activemq
org.apache.camel.archetypes:camel-archetype-component
org.apache.camel.archetypes:camel-archetype-java
org.apache.camel.archetypes:camel-archetype-scala
org.apache.camel.archetypes:camel-archetype-spring
org.apache.camel.archetypes:camel-archetype-war
org.apache.cocoon:cocoon-22-archetype-block
org.apache.cocoon:cocoon-22-archetype-block-plain
org.apache.cocoon:cocoon-22-archetype-webapp
```

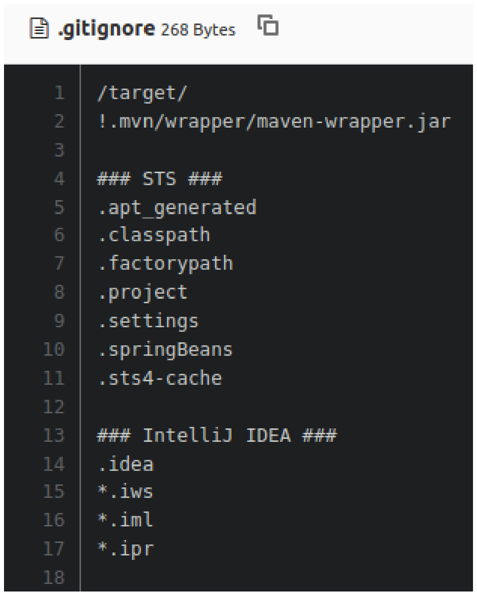
5) Instead of using the default location suggested, use this one:

PT2023_GroupNumber_LastName_FirstName_Assignment_Number

Do not forget to also choose the appropriate name for your new Project, most likely using the format we requested (unless it is a test project, not for one of your assignments).

6) Click *Finish*.

7) Now, before pushing anything to the remote project, you must create a *.gitignore* file in the project, which tells git which files to ignore when committing and pushing to your remote projects. You don't want unnecessary files, such as IDE configuration files, to be pushed, because they are strictly relevant for your local system. Just create a file named "*.gitignore*" and write the following lines:



```
/target/
!.mvn/wrapper/maven-wrapper.jar

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

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

```

8) Right click on the folder

PT2023_GroupNumber_LastName_FirstName_Assignment_Number and click Git Bash Here; then introduce the next commands:

   a) git add .

Note: the command will mark any changes that you have made to your project files (e.g. creating/modifying/deleting files) as staged so that they can be included in the next commit (for more details check this link).

   b) git commit –a –m "initial commit"

Note: the command commits any files you have added with the git add command and commits any files you have changed since then – at this step the changes are saved only locally. An explanation message is given in order to document what has been added/changed (for more details check this link).

   c) git push –u origin main

Note: the command sends the committed changes to your remote repository (for more details check this link).

4.3.2 Update the project

1) Create a new class named *Main* in the same package as the class *App*.
2) Navigate to folder *PT2023_GroupNumber_LastName_FirstName_Assignment_Number*, *right click* and select *Git Bash Here*
3) Insert the next commands:

   a) git add .

   b) git commit –a –m "add new class"

   c) git pull origin main

Note: the command fetches and merges changes on the remote server to your working directory (for more details check this link).

   d) git push –u origin main

4) You can always see the modification that were not committed yet by using:

   a) git status

4.3.3. Create and work with a new branch

The real value of working with git, is the power of *branches*. They allow multiple developers to work simultaneously on the same project, on different features, and then to merge all the new changes in the main branch.

1) **Create a branch production**

The first step towards working with branches, is to create a new branch. When you create a new branch, it will automatically be initialized with the currently existent code. Then, while inside that branch, all changes will be added only on that branch.

To create a new branch:

- pull all the changes from the remote project, to be up to date:
  - *git pull*
- create the branch on your local machine and switch directly to that branch:
  - *git checkout -b <branch-name>*
- push the newly created branch to the remote repository:
  - *git push origin <branch_name>*

An example to create the branch production is the following:

```
#git pull

#git checkout -b production

#git push origin production
```

2) **switch between branches**

When working with multiple branches, it is important to keep track of all the available branches, and to always know on which branch you currently are.

- To bring locally meta-data information about existing branches:
  - *git fetch --all*
- In order to see all existent branches:
  - *git branch -a*
- In order to see on which branch you currently are:
  - *git status*
- *in order to switch from a branch to another, use the above-mentioned command:*
  - *git checkout <branch-name>*
    - *if the branch with that name is already existent, it will just switch to that one, instead of creating a new one*

You can try to create a new branch, make some small change, then switch back to the main branch, and see that change is not present in the main branch.

3) **commit changes to new branch**

When making changes on a new branch, you must always commit and push them to the remote branch, just like working on main.

- First, make sure you are on the right branch:
  - *git status*
- Repeat the same process as if you were working with main. However, pay attention to the names:
  - *git add .*
  - *git commit -m "commit message"*
  - *git push -u origin <branch-name>*
- And now, your remote branch <branch-name> contains all the changes you have pushed.

4) **merge branch with main**

An important step when working with branches, is to always keep the main branch up to date with the latest **working and functional** code from your other branches. Merging two branches, as the name suggests, is the process of merging the code from two branches. If the branches contain changes on different parts of the code, the merge process will work instantly. If both branches contain changes on the same parts of code, git will require you to solve the conflicts: from the two modifications, you must choose the one which you want to remain in the final version.

DO NOT FORGET: do not merge code which is not working properly, or which is not tested, into main. The main branch must always contain the latest functional version of your project.

In order to merge two branches:

- git merge <branch-with-new-changes> <branch-to-be-updated>

For a more in-depth explanation of branches and how they can be manipulated to serve your needs, we suggest checking the following tutorial:

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

4.3.4. Getting git to work with a proxy server

1) In the UTCN laboratories you need to set the proxy server to use GIT bash
2) Open Git Bash
3) Insert the following commands:
   a) git config --global http.proxy http://proxy.utcluj.ro:3128

   b) git config –global --get http.proxy

4) In order to unset the proxy, use the following command:

   a) git config --global --unset http.proxy

4.3.5. Getting MAVEN to work with a proxy server

1) In the UTCN laboratories you need to set the proxy server in order to use MAVEN projects
2) Go to Windows Explorer-> Drive C-> Users -> *Your User* -> .m2
3) Create the folder **conf**
4) Go to conf folder and create the file **settings.xml** with the following content:

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
                      http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <localRepository/>
  <interactiveMode/>
  <usePluginRegistry/>
  <offline/>
  <pluginGroups/>
  <servers/>
  <mirrors/>
  <proxies>
    <proxy>
      <id>myproxy</id>
      <active>true</active>
      <protocol>http</protocol>
      <host>proxy.utcluj.ro</host>
      <port>3128</port>
      <username></username>
      <password></password>
      <nonProxyHosts>localhost,127.0.0.1</nonProxyHosts>
    </proxy>
  </proxies>
  <profiles/>
  <activeProfiles/>
</settings>
```

5) Go back to folder **.m2**
6) Delete the folder **repository**
7) For Eclipse

   a) Open **Eclipse**
   b) Go to **Window->|Preferences->|Maven->|User Settings**
   c) At the **User Settings** tab browse for the **settings.xml** file created at step 4
   d) Click **Apply** and **OK**
   e) Go on the project, right click and go to **Maven->Update Project**

8) For IntelliJ IDEA
   a) Open IntelliJ IDEA
   b) Go to **File->|Settings->|Build, Execution, Deployment->|Build Tools->|Maven**
   c) In the **User Settings file** field browse for the **settings.xml** file created at step 4.
   d) Click **Apply** and **OK**