

# Installation

Due to version differences in software components (whether versions are older or newer), issues may occur.

## Checking the Java Version

Since the Java version specified in the `pom.xml` file is 17,

- ¥ The system environment variable `JAVA_HOME` must also point to a Java installation directory of the same version.

```
java -version
```

- ¥ If running from IntelliJ IDEA Community Edition, follow the menu path below to select a Java version of 17:

*menü yolu*

```
IntelliJ IDEA
  File
    Project Structure
      Project Settings
        Project
          SDK
```

## Downloading the Files

- ¥ To clone the files from the repository, `git` must be installed. If it is installed, run the following command in the console from an empty folder.

```
git clone https://github.com/turkerozturk/SweetCherrySync.git
```

- ¥ Or, the files can be downloaded as an archive from <https://github.com/turkerozturk/SweetCherrySync/archive/refs/heads/main.zip> using a web browser, and the files extracted from the archive.

## Building from Source Code

Open a console window in the `SweetCherrySync` folder. Run the following command.

*windows*

```
mvnw.cmd clean package
```

linux

```
./mvnw clean package
```

!

This command will perform a clean build using the Maven version specified in the project's `.mvn/wrapper/maven-wrapper.properties` file (if not found on the system, it will download it to the `KULLANICININ EV D!Z!N!/.m2/wrapper/dists` directory and then perform a clean build, removing any previous build folder).

||

On Linux, to allow the `mvnw` command to run, you must once run `chmod +x mvnw` to mark the file as executable.

As a result, the built project will be available as a JAR file in the `target` directory within the project directory.

The JAR file will be created with the name `SweetCherrySync-1.0.0.jar`.

## Checking the Port Number Before Running

!

You can proceed to the next section. If your application does not run, return to this section, check the port, adjust the application settings if necessary, and try again.

### If Another Application Is Using the Port Number

You can find an unused port number by trial and error or with a console command. To check whether the default ports used by the application are free, run:

windows cmd

```
netstat -ano -p tcp | find "8081"
```

If the command output shows:

TCP	0.0.0.0: {httpsportnumber}	0.0.0.0:0	LISTENING	PID
numarasõ				

and

TCP	0.0.0.0: 8081	0.0.0.0:0	LISTENING	PID numarasõ
-----	---------------	-----------	-----------	--------------

then the port is in use, and to run our application you need to either find and terminate the application using that port, or change the port number of our application.

!

For Linux, the equivalent commands are: `sudo netstat -ano -p tcp` and `ps -ef |`

```
grep PID numarasõ.
```

## Finding the Process by PID

If you want to find which application is using a specific PID, use the following command:

```
tasklist | find "PID numarasõ"
```

This command will display the name of the process.

If it produces no output, the port is free, and our application should run without issues using that port.

## If You Want to Use Different Port Numbers

You need to specify an unused port number in the `application.yml` file.

¥ If the project is to be run from source code, the file is located at `PROJEKLAS...Rt/src/main/resources/application.yml`.

¥ If the project is to be run as a JAR file, create an `application.yml` file in the same directory as the JAR file. The indentation from the line beginnings must be the same.

*the section that should be in the application.yml file*

```
server:  
  port: 8081 "
```

" HTTP port number

## Running the Project

### Running from Source Code

To run the uncompiled project, open a console window in the directory containing the `mvnw` file and run the following command:

*windows cmd konsolunda*

```
mvnw.cmd spring-boot:run
```

*linux terminalinde veya windows powershell konsolunda*

```
./mvnw spring-boot:run
```

## Running the Compiled File

*windows cmd*

```
java -jar `SweetCherrySync-1.0.0.jar`
```

■

When running the compiled JAR file, make sure to run it from the same directory where the file is located. Otherwise, after the application starts, you may encounter issues with the locations of required files and directories during use.

## Logging in with a Web Browser

Access the application via <https://localhost:8081>.

The web page only displays statistical information; it is not required for synchronization. The web server feature can be disabled.

*the section that should be in the application.yml file (enable/disable embedded Tomcat)*

```
spring:
  main:
    web-application-type: servlet "
```

" The default value is `servlet`; to disable it, set it to `none`.

---

## Resources

¥ [Cloning a repository](#)

¥ [A Quick Guide to Maven Wrapper](#)

¥ [Determine Which Program Uses or Blocks a Port](#)

---

Author: [TŸrker ...ztŸrk](#)