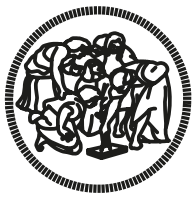

POLITECNICO
MILANO 1863

SafeStreets

# SOFTWARE TO INSTALL

Version 1 – 12/01/2020

---


*Authors:*

Giulio A. Abbo 10538950

Gianmarco Accordi 10587213

Massimiliano Bonetti 10560496

*Professor:*

Elisabetta Di Nitto


*Academic year:*

2019 – 2020

# CONTENTS

# 1 FOR THE CLIENT

To compile the source code of the client we have used Android Studio 3.5.3 with flutter, the complete setup instructions are available here: *https://flutter.dev/docs/get-started/install*

## 1.A ON MAC

### 1.A.1 To install Flutter

Download the Flutter SDK and extract it in a desired location:
*https://storage.googleapis.com/flutter_infra/releases/stable/macos/flutter_macos_v1.12.13+hotfix.5-stable.zip*

Add the flutter tool to your path:

```
export PATH="$PATH:`pwd`/flutter/bin"
```

### 1.A.2 To run the client

You need to install Xcode (we have used the version 11.3 for iOS 13.2.3) (Xcode is available only for Mac): from the App Store of the Mac or from
*https://developer.apple.com/download/more/?=xcode*
(you need to register as a Developer, there is no cost associated) and select Xcode 11.3

Then you should install also CocoaPods by entering in the Terminal:

```
sudo gem install cocoapods
```

### 1.A.3 Finally

Run the following command to see if there are any dependencies you need to install to complete the setup:

```
flutter doctor
```

## 1.B ON WINDOWS

### 1.B.1 To install Flutter

Download the Flutter SDK and extract it in a desired location:
*https://storage.googleapis.com/flutter_infra/releases/stable/windows/flutter_windows_v1.12.13+hotfix.5-stable.zip*

Add the flutter tool to your path:

- From the Start search bar, enter 'env' and select Edit environment variables for your account.
- Under User variables check if there is an entry called Path:
- If the entry exists, append the full path to flutter\bin using ; as a separator from existing values.
- If the entry doesn't exist, create a new user variable named Path with the full path to flutter\bin as its value.

### 1.B.2 To run the client

Download and install Android Studio: *https://developer.android.com/studio*

Install dart and flutter plugins for Android Studio and set up an emulator: *https://flutter.dev/docs/get-started/install/windows#android-setup*

### 1.B.3 Finally

Run the following command to see if there are any dependencies you need to install to complete the setup:

```
flutter doctor
```

## 1.C On Linux

### 1.C.1 To Install Flutter

The complete set of instructions on how to install Flutter are available here: *https://flutter.dev/docs/get-started/install/linux*. The Installation is like the Mac tutorial, in this case after installing it you need to export the environmental path:

```
export PATH="$PATH:`pwd`/flutter/bin"
```

and then you can check the installation with:

```
flutter doctor
```

Then you can configure it in Android Studio, and freely use it to run some builds or some test.

### 1.C.2 To run the client

To access and run the client you need to make the build of the project by using Flutter, so you have to go configure your project inside Android Studio(*https://developer.android.com/studio*), now you have to go in Settings -> Plugin and from here you have to download the Flutter and Dart plugin (as described at this link *https://flutter.dev/docs/get-started/editor*). Then you can launch you application from you android device, or you can setup an emulator by using the following guide:
*https://developer.android.com/studio/run/emulator*

Before launching the app you have to go inside the project folder at this path: /lib and you have to modify the file *handler_backend.dart*, you have to change the following line:

```
final String ip = '192.168.1.14';
```

During our test we have used this ip, that has to be the ip of the machine on which the server is running.

## 1.D For web support

Follow the instructions at *https://flutter.dev/docs/get-started/web* to enable web support. The generate web build is already present on the web server, in the folder: */src/main/webApp*.

# 2 FOR THE APPLICATION SERVER AND THE WEB SERVER

To compile the source code of the application server and the web server, we have used IntelliJ IDEA 2019.3 (Ultimate Edition):

*https://www.jetbrains.com/idea/download*

Important plugins for IntelliJ:

- JavaEE: RESTful Web Services(JAX-RS)
- JavaEE: EJB, JPA, Servlets
- GlassFish
- Database Tools and SQL

Download the JDK of Java 8 and add it to your path:

*https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html*

Download GlassFish 4.0 and put it in a desired location:

*https://javaee.github.io/glassfish/download*

In order to user Glassfish 4.0 you have to use Java 8.

Download the Connector/J 8.0.18:

*https://dev.mysql.com/downloads/connector/j/*

and put it inside GlassFish inside the directory /glassfish4/glassfish/domains/domain1/lib

# 3 FOR THE DATABASE

For the DBMS we have used MySQL Community Server (our version is 8.0.17, but also 8.0.18 should work):

*https://dev.mysql.com/downloads/mysql/*

If you would like to communicate directly to the DBMS in a fast and intuitively way you can use MySQL Workbench (our version is 8.0.17, but also 8.0.18 should work):

*https://dev.mysql.com/downloads/workbench/*