

# INSTALLATION GUIDE

VERSION 1.0

October 24, 2024

## CONTENTS

1	Prerequisites	2
2	Installation	3
2.1	app setup . . . . .	3
2.2	server setup . . . . .	4
2.3	ai setup . . . . .	5

## 1 PREREQUISITES

To use this system you will need the following:

- Flutter SDK version 3.22
- Dart SDK version 3.4.3
- Android SDK API 28
- Python version 3.10
- MongoDB version 6.0.4<sup>1</sup>

I recommend you using IntelliJ IDEA as your working environment because it's the one I used during the developement of the system (and a really good IDE overall!). I cannot give you instructions on how to replicate the installation on other environments.

---

<sup>1</sup> This version is now 2 years old, I haven't tested if this project works on newer versions of MongoDB, but given the very basic use of MongoDB in this project it should work finely.

## 2 INSTALLATION

First, you'll need to clone the repository. After cloning, you'll notice three main directories:

- app
- server
- ai

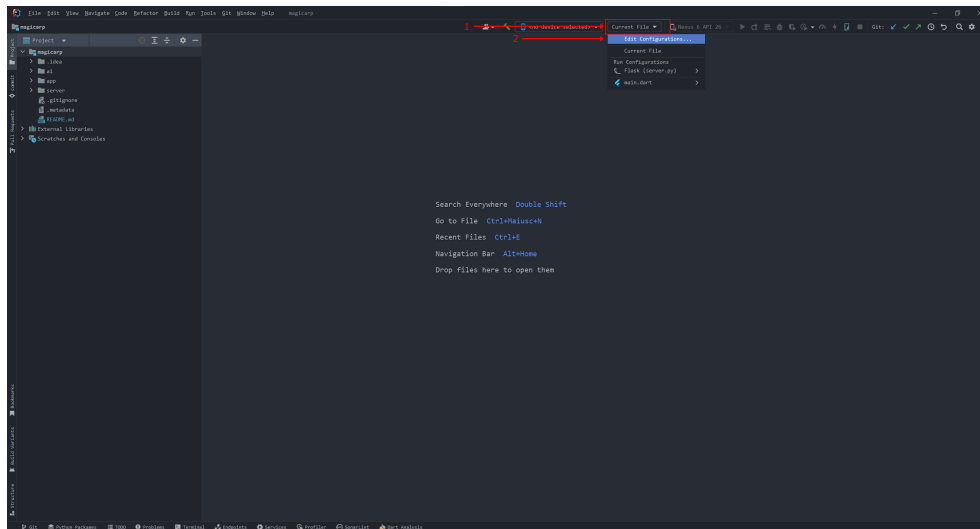
I'll guide you through setting up IntelliJ to get the project running on a freshly cloned repository. Make sure to pay attention to any IntelliJ notifications—it will handle some setup tasks automatically, which can save you from potential issues!

### 2.1 app setup

First, you will need to get all dependencies listed in the `pubspec.yaml`. To do so, open the terminal inside IntelliJ and run the following:

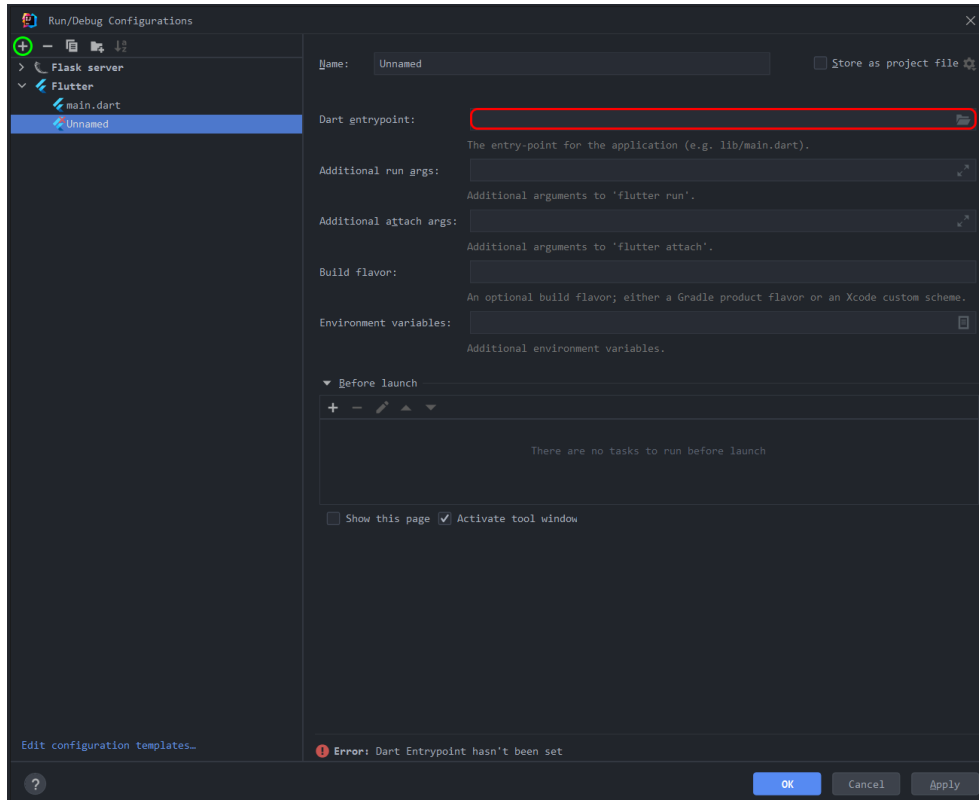
```
cd app
flutter pub get
```

After you got the dependencies, you'll need to create a Flutter configuration. Click on the **Current File** dropdown menu and then on **Edit Configurations...**



A menu will pop on your screen, here you will need to create a new Flutter configuration to run the application. Click on the + icon to create a new configuration (highlighted in green), then select Flutter in the dropdown menu.

You may name the configuration however you want (I suggest you naming it `main.dart`). In the Dart entrypoint field (highlighted in red), you need to select the path to the `main.dart` file, located in the `app/lib` directory.



After creating the configuration you are ready to go! You can launch the app on an emulator, on your device or build an apk. The choice is yours.

## 2.2 server setup

First of all, we need to install the requirements to run the server. Open the terminal inside IntelliJ and run the following:

```
cd server
pip install -r requirements.txt
```

You may want to run a virtual environment for Python, if you wish to do so, then you will need to run the following before installing the requirements:

```
python -m venv venv
```

Then, if you are on Windows, you can activate the virtual environment with

```
venv\Scripts\activate
```

If you are on macOS/Linux, you can activate the virtual environment with

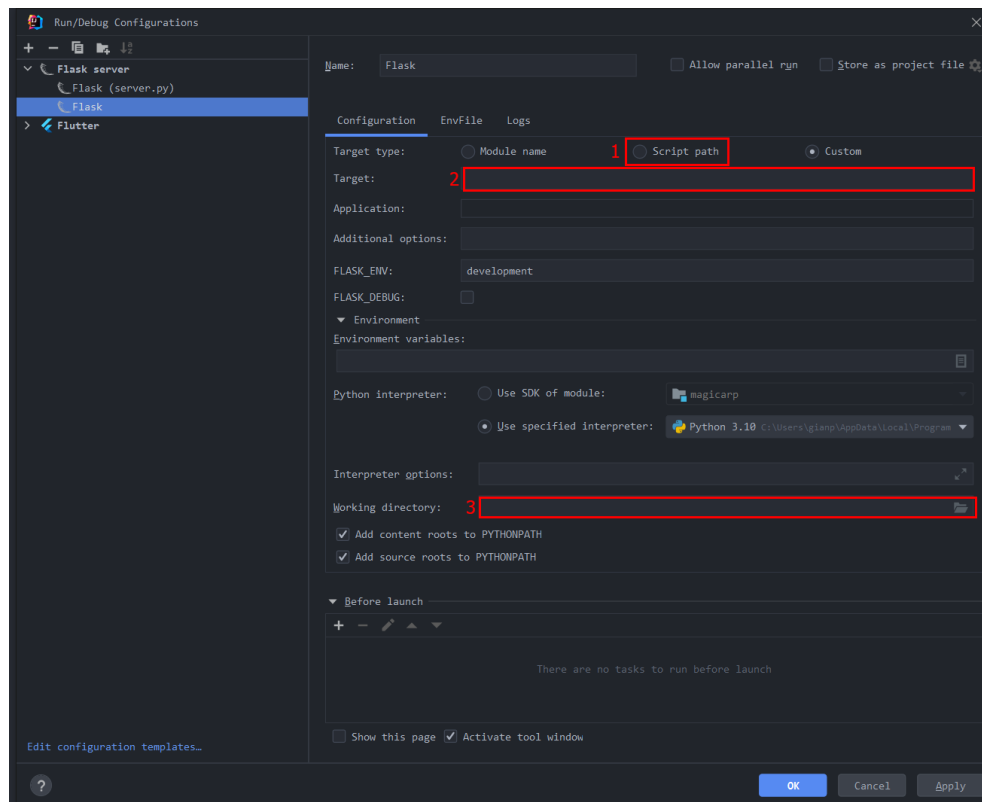
```
source venv/bin/activate
```

You can verify the installed components with

```
pip list
```

In the same way we did for the Flutter configuration, we need to create a Flask configuration. This time, we will need to edit more fields in the configuration:

1. Set the Target type as Script path
2. In the Target field, you need to select the path to the `server.py` file, located in the server directory
3. In the Working directory field, select the server directory path



### 2.3 ai setup

As we did for the server, we need to install the requirements. Open the terminal inside IntelliJ and run the following:

```
cd ai
pip install -r requirements.txt
```

This time, there is no configuration to create, just open the console and run:

```
python main.py
```