# Eitan Medical

# Installation guide

1. Pre-Installation:

Before downloading and building dockers, you must install tools on your machine (windows or Linux).

* 1. (For Windows) Docker Desktop: <https://docs.docker.com/desktop/install/windows-install/>
  2. (Linux) Docker Compose <https://docs.docker.com/compose/install/linux/>
  3. (Running from source) .Net Core 6
  4. (Running from source) Visual studio 2022.

## Run From Dockers:

* 1. Clone from git: ***git clone git@github.com:orontamir/EitanMedical.git***
  2. Insert into: ***EitanMedical folder (cd EitanMedical)***
  3. Open Console or Power shell and run the command: ***dos2unix init.sh***
  4. Run ***Docker-compose -f docker-compose.yml up -d***on CMD or Power shell.
  5. Open browser: [***http://localhost/swagger/index.html***](http://localhost/swagger/index.html)
  6. Open Database IDE (Data grip , SQL lite i.e.)
  7. Connect to data base by connection string: server=localhost;Port=3306;user=root;password=qwerty;database=EitanMedicalDb
  8. Open EitanMedicalDb

## Run From VS2022 with existing mongo:

* 1. Inset to folder: EitanMedical
  2. Open EitanMedical.sln on VS 2022.
  3. Run EitanMedical and IoTService projects
  4. Open browser: [***https://localhost:7000/swagger/index.html***](https://localhost:7000/swagger/index.html)
  5. Open Database IDE (Data grip , SQL lite i.e.)
  6. Connect to data base by connection string: server=localhost;Port=3306;user=root;password=qwerty;database=EitanMedicalDb
  7. Open EitanMedicalDb

1. Sending Data to the server:
   1. After two dockers are running
   2. Create two folders:
      1. C:\Log
      2. C:\Patients
   3. In C:\Log will save log files with information and errors.
   4. In the Patients folder, we put our JSON file with the data we want to send to the server
2. Example of JSON file:

{

"patients": [

{

"id": "3",

"name": "oron",

"age": 34,

"gender": "female"

},

{

"id": "4",

"name": "tamir",

"age": 45,

"gender": "male"

}

],

"heartRateReadings": [

{

"patientId": "3",

"timestamp": "2024-03-01T08:00:00Z",

"heartRate": 85

},

{

"patientId": "4",

"timestamp": "2024-03-01T10:30:00Z",

"heartRate": 101

}

]

}

1. Show Data From the server:
   1. Open swagger by web browser: <http://localhost/swagger/index.html>
   2. High Heart Rate Events
      1. A screenshot of a computer

         Description automatically generated
   3. A screenshot of a computer

      Description automatically generatedHeart Rate Analytics
   4. Patient Request Tracking
      1. A screenshot of a computer

         Description automatically generated