Keith Erkert

kerkert@asu.edu

Updated 11/24/2020

Abstract

This guide will walk the user through how to setup the backend server software (the API server and helper application) to use as an alternative to our servers.

Back-End Database and API server Setup Guide

Table of Contents

[Introductory 2](#_Toc103523358)

[Hardware/Software Requirements 2](#_Toc103523359)

[Environment Setup 3](#_Toc103523360)

[Running the Backend Applications 11](#_Toc103523361)

# Introductory

As an alternative to using the provided servers from the creators of the IoT monitoring application, users can choose to host their own backend applications for either testing purposes or for their own privacy. This guide will walk users through how to setup the server on a fresh install of Ubuntu 20.04. The backend applications will work on different distros of Linux and will also work on Windows 10/11, there is no guide on how to set the backend applications on those options.

The backend applications are built using .Net 6 framework in the C# language. IotbackendAPI is the web API server that will host the endpoints for the Raspberry Pi to send data, and for the phone apps to retrieve data from. It also handles authentication for users for both the Raspberry Pi and phone apps. The backend helper app runs in a loop and will cache external IP information (such as geolocational data and dns information) so the phone application does not have to query this from 3rd party APIs every time the user inspects an external flow. The application will refresh cached content once it is a week old. Both the web API and helper applications utilize a MySQL database which will be setup in this guide. Mariadb will the engine used to host the MySQL database locally.

# Hardware/Software Requirements

Recommended Server Specifications

* Linux or Windows 10 OS

*\*This guide will use Ubuntu 20.04 LTS, a modern Linux Distro is recommended as the server uses .NET 6*

*See* [*https://docs.microsoft.com/en-us/dotnet/core/install/linux*](https://docs.microsoft.com/en-us/dotnet/core/install/linux) *for supported Distros*

* 4-8GB of RAM

*\*This is heavily dependent on OS and expected workload on this server. If it is being used for personal use only and only running the backend software, the RAM usage should be minimal (around 4GB if using Linux)*

* 1-5GB storage (not including OS size)

*\*The backend applications and libraries should not use more than 1GB of storage. User data size is dependent on how active the devices are on the pi network, how long the data is saved for, and how many users (if not for personal use).*

# Environment Setup

Install .NET libraries

1. Open a terminal
2. Add Microsoft’s packages to the package manager



1. Install .NET6 SDK



1. Install .NET6 Runtime



Install and Setup MariaDB

1. Update package manager and install MariaDB



1. Run setup for MySQL



1. Enter a password (remember this password)

Text

Description automatically generated

1. Skip unix\_socket authentication and change root password (answer n for both)

Text

Description automatically generated

1. Remove anonymous users, disallow remote access for root, delete the test database, and reload the privileges. (answer y for all 4)

Text

Description automatically generated

1. Open mysql under root user and the password used by ubuntu (not the password during setup)
2. Setup a user for the backend application to connect to the database on

*\*Replace \*\*username\*\* and \*\*password\*\* with whatever you like, these will be used in the backend application.*



1. Create non-dynamic tables the backend app uses. First make a new database called main



Text

Description automatically generated

1. We need to let the database engine know we are using that database



1. Create the main tables by copying and pasting the following code snippets (to paste in a terminal, use **ctrl+shift+v**)





1. Verify the tables were made



Text

Description automatically generated

1. As and extra layer of security, our webAPI verifies the connecting client has a valid client token (to reduce web crawlers from getting more information about our http endpoints), add the following client tokens to the program\_auth tables



1. Open the port the web API listens on to allow the Raspberry Pi and Phone Application to communicate with the web API. This will also enable the firewall



*This only opens the port for devices on your LAN/WLAN, not the outside internet*

1. To use the phone application outside your home network, forward port 6000 (TCP) in your router for the servers IP. You should also reserve the IP for the server MAC address, so it will not change if the router resets from a power outage. Please refer to online guides from your routers manufacturer on how to do this.

This will open the webAPI to the internet, other devices from outside you network will be able to connect to the webAPI. There is an authentication layer to the webAPI to prevent anonymous users from accessing your data, but this is a potential security vulnerability, use at your own risk!

# Running the Backend Applications

1. Download the WebAPI and backend helper application from GitHub.
   1. Web API – <https://github.com/ASU-IoT-ResearchProject/IotBackendAPI/releases/tag/v22w45d7a>
   2. Backend Helper – <https://github.com/ASU-IoT-ResearchProject/IoTBackendHelper/releases/tag/v22w45d7a>
2. Unzip each package into its own folder (if its zipped)
3. Open each folder in its own terminal
4. To start the web API run this command



1. To start the backend helper app run this command

