Process

1. Server is listening for new clients on port 4321
2. New Client sending request to Server for registration, the request contains following information:
   1. ClientID
   2. ClientIP (will be received automatically from connection data, but it could be changed due to DHCP)
   3. ClientKey (randomly generated and stored in client)
3. Server sending this information to Sqlite3 to table Clients
4. On response from Sqlite3 sending back code:
   1. 1 – Client added
   2. 2 – Client already registered (ID is already exists)
   3. 3 – Client deleted
   4. 4 – Key replaced (?)
5. Every minute, Server checks for list of Clients in format:
   1. ID = Client ID
   2. Key = Client Key
   3. IP = Client IP
6. Server creating HTTP “Keep Alive” request (Get or Post) through RestAPI
7. Server sending requests to Clients
8. Clients sends back responses based on status.txt files.
9. Server put all responses to table Raw Data at Sqlite3 DB.

Client side: basic HTTP server for responding on Server requests

Server side: HTTP server with Flask to show queries from DB

Table’s structure:

DB Name: WaterSRV

Clients:

ClientID (PK)

ClientIP (Not Null) – relevant only in Static IP configuration, required updating /etc/hosts records

ClientKey (Not Null)

ClientName (Not Null)

Raw Data:

recordID (PK)

clientID (FK)

datetime (Not Null)

status (Not Null)

alarm1 (Not Null)

alarm2 (Not Null)

message (Null)