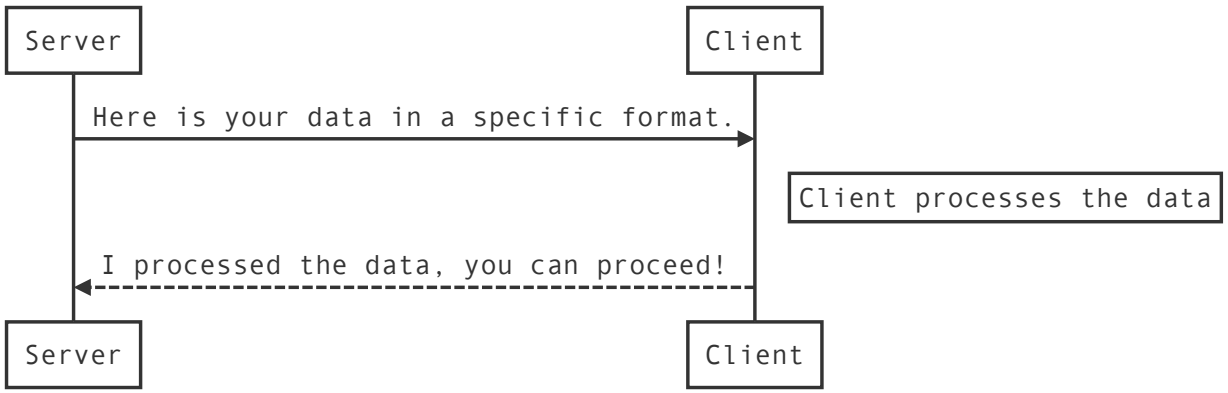
Server/Client Application

We would like you to develop several Python scripts that together will form an application based on server/client architecture. The application will simply be used for transferring data from a server to multiple clients over a network via TCP/IP.

> Some parts of the description is intentionally left open for you to be able to use your creativity.

Basically, the application will work as shown in the following diagram.



Server Specifications

Database Structure

The server will persist data in a MySQL database. The database will have the following tables:

Table 1 - The `clients` table will store information about all clients that the server can communicate with.

ID	NAME	HOST	PORT
1	Client #1	127.0.0.1	5001
2	Client #2	127.0.0.1	5002

Table 2 - The `personnel` table will store information about all personnel that the server can send to clients.

ID	NAME	SURNAME	SSN
1	Isaac	Newton	12345678900
2	Albert	Einstein	98765432100

> SSN stands for Social Security Number.

Table 3 - The `messages` table will store all messages that are waiting to be sent to clients.

ID	CLIENT_ID	PAYLOAD
1	1	{message}
2	2	{message}

> PAYLOAD column will store the message in a specific format determined by you.

Task Description

The server will run until it is stopped by the user and perform the following tasks:

1. Send a specific personnel to a specific client.
2. Send a specific personnel to all clients.
3. Send all personnel to all clients.
4. Delete a specific personnel from a specific client.
5. Delete a specific personnel from all clients.
6. Delete all personnel from all clients.

Client Specifications

Database Structure

The client will persist data in a SQLite database. The database will have the following table:

Table 1 - The `personnel` table will store information about all personnel that are sent by the server.

ID	NAME	SURNAME	SSN
1	Isaac	Newton	12345678900
2	Albert	Einstein	98765432100

> SSN stands for Social Security Number.

Task Description

The client will run until it is stopped by the user and perform the following tasks:

1. Save a personnel after receiving a `save` message from the server.
2. Delete a personnel after receiving a `delete` message from the server.