Smart Home Transformation with IBM Cloud Functions for IoT Data Processing

Phase – 3 Document Submission

Team Members:

- 110121205029 – Mohamed Shaaheen. A
- 110121205014 – Hasheer Ahamed. M
- 10121205020 – Mohamed Aarkif. S
- 110121205026 – Mohamed Rilwan. J
- 110121205034 – Mohammed Faiz. J
- 110121205050 – Sheejan. J

Project: Serverless IOT Data Processing

Phase – 3: Development Part 1

Step 1: Select Smart Devices

a. Device Selection:

- Carefully choose the smart devices that align with your project goals. Consider factors like sensor types, communication protocols, and compatibility with the IBM Cloud IoT platform.

b. Device Compatibility Check:

- Ensure that the chosen devices support communication protocols like MQTT or HTTP, which are commonly used in IoT solutions.

Step 2: Set Up Device Integration

a. Register Devices on IBM Cloud IoT Platform:

- Log in to the IBM Cloud Console and navigate to the IBM Cloud IoT platform service.

- Create individual device profiles for each smart device. This typically involves providing a unique device ID, device type, and security credentials (such as API keys or certificates).

b. Obtain Device Credentials:

- For each registered device, obtain the necessary credentials (e.g., API key, authentication token) that will be used to establish a secure connection between the device and the IBM Cloud IoT platform.

c. Implement Device Code:

- On the smart devices, implement the necessary code or firmware that enables them to communicate with the IBM Cloud IoT platform. This code should include logic for securely sending data.

  Example (for MQTT protocol in Python using Paho MQTT library):

  ```
  # Connect to IBM Cloud IoT Platform

  client = mqtt.Client(client_id=device_id)

  client.username_pw_set(username, password)

  client.connect(broker, port, keepalive)

  # Publish data

  client.publish(topic, payload)
  ```

Step 3: Establish Data Collection

a. Define Data Attributes:

- Identify the specific data attributes you want to collect from each smart device. For example, if you're working with a temperature sensor, you may collect temperature readings.

b. Set Up Data Collection Pipelines:

- Within your IBM Cloud Functions project, create the necessary components to handle incoming data. This may involve setting up triggers that listen for data from the IoT platform.

c. Implement Data Ingestion Functions:

- Write serverless functions that will handle the incoming data. These functions should be designed to parse and process the data received from the devices.

  Example (for IBM Cloud Functions in Node.js):

  ```
  function main(params) {
  // Process incoming data
  // ...
  return { result: "Data processed successfully" } ;
  }
  ```

Step 4: Test Data Collection and Device Integration

a. Simulate Data (Optional):

- If physical devices aren't available, consider using simulators or mock data generators to simulate device-generated data for testing.

b. Verify Data Integration:

- Send test data from your smart devices to the IBM Cloud IoT platform. Monitor the platform to ensure that the data is being successfully collected and processed by your serverless functions.

c. Implement Error Handling:

- Consider scenarios where data transmission might fail or be corrupted. Implement error handling mechanisms to address potential issues during data collection and device integration.