Create a device management module using Java with Azure IoT Hub, store in Cosmos DB, and deploy on Azure App Service.

1. Introduction

IoT Middleware serves as a bridge between physical IoT devices and cloud applications, handling communication, device management, and data storage. In this project, a Java-based device management module is employed, which interacts with Azure IoT Hub for secure device provisioning and messaging and stores device metadata in Azure Cosmos DB for scalability and low-latency access. The application is hosted on Azure App Service to ensure high availability and seamless integration.

Technologies Utilized:

• Azure IoT Hub: To establish secure connectivity for devices and to manage them.

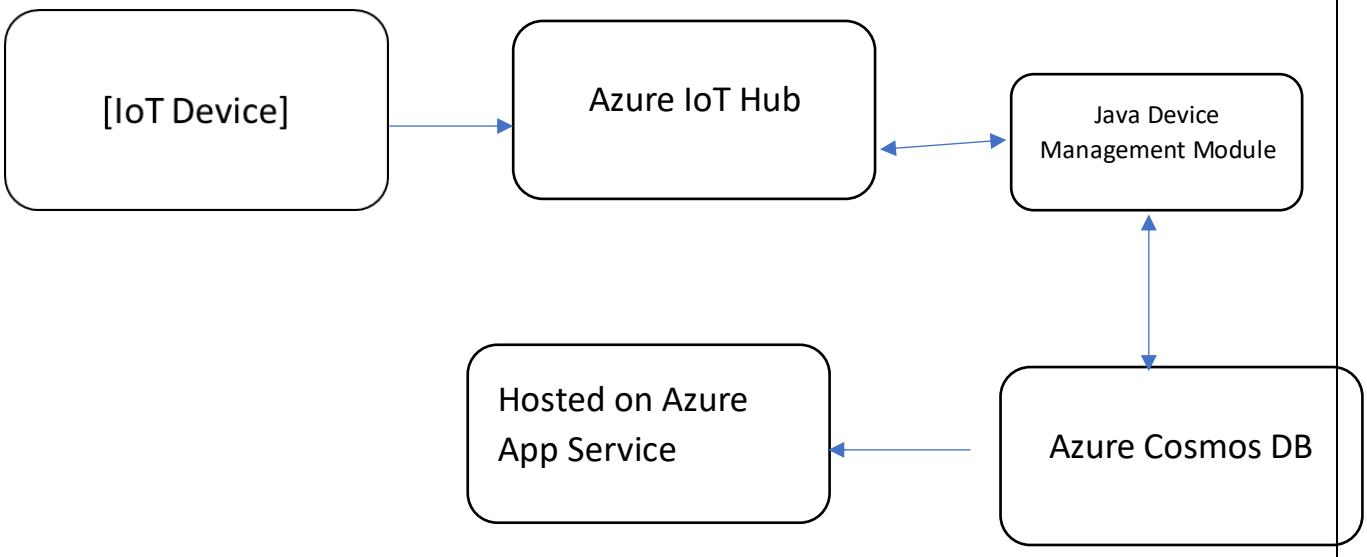
• Java (Spring Boot): To code the backend for device management.

• Azure Cosmos DB: For metadata storage of devices and query them.

• Azure App Service: To host the Java-based web application.

2. Overview of System Architecture

Architecture Diagram:



Component Explanation:

• IoT Device: Registers with Azure IoT Hub and sends messages via it.

• Azure IoT Hub: Central hub to handle devices and routes telemetry.

• Java App (Spring Boot): Handles device registration, updates, deletes, and maintains metadata.

• Cosmos DB: NoSQL database to hold the device information (device ID, status, timestamps, etc.).

• Azure App Service: Cloud platform for hosting and scaling the Java app.

3. Azure IoT Hub Setup Steps

1. Go to Azure Portal → Create new IoT Hub.

2. Register device identity in the hub.

3. Generate connection string and securely store.

Java Integration:

• Use Azure IoT Device SDK for Java.

• Connect using device credentials.

```
Device device = Device.createFromId("device001", null, null);
registryManager.addDevice(device);
```

4. Java Device Management Module

Features:

• Register, update, and delete IoT devices using Azure IoT Hub APIs.

• Store device information (status, config) in Cosmos DB.

Code Snippet Sample:

```
@RestController
public class DeviceController {
   @PostMapping("/register")
public ResponseEntity<?> registerDevice(@RequestBody DeviceDTO device) {
      deviceService.registerToIoTHub(device);
      cosmosTemplate.insert(device, "DeviceCollection");
}
return ResponseEntity.ok("Device registered");
}
```

5. Deploy on Azure App Service

1. Package Java Spring Boot app with Maven or Gradle.

2. Create Azure App Service (Java stack).

3. Deploy app with GitHub Actions or Azure CLI.

Deployment with Azure CLI:

az webapp up --name device-manager-java --runtime "JAVA|11-java11" --resource-group MyResourceGroup

6. CI/CD with GitHub Actions (4 Marks)

Workflow Overview:

- **Trigger**: On push to main
- **Steps**:
    - Build Java app with Maven.
    - Deploy JAR to Azure App Service.

GitHub Actions YAML Snippet:

yaml

CopyEdit

```
name: Java CI/CD


on:
  push:
    branches: [ main ]


jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Set up Java
        uses: actions/setup-java@v2
        with:
          java-version: '11'
      - name: Build with Maven
        run: mvn clean package
      - name: Deploy to Azure Web App
```

uses: azure/webapps-deploy@v2

with:

app-name: 'device-manager-java'

package: 'target/*.jar'

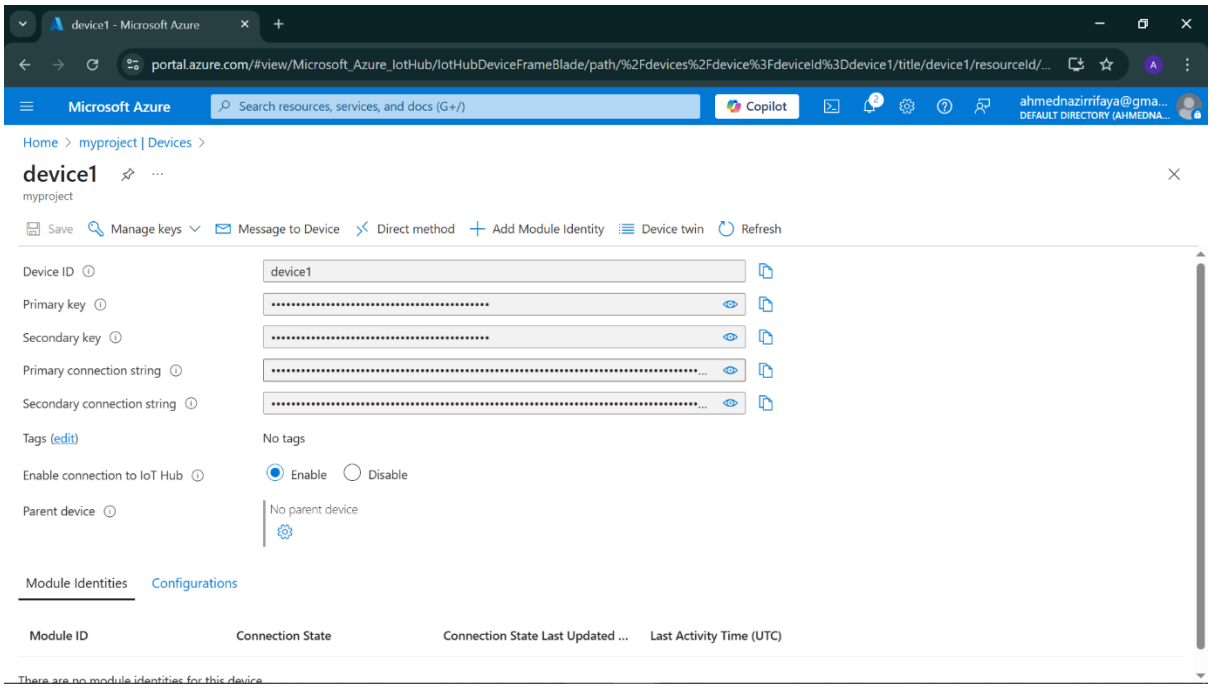
7. Scalability and Monitoring

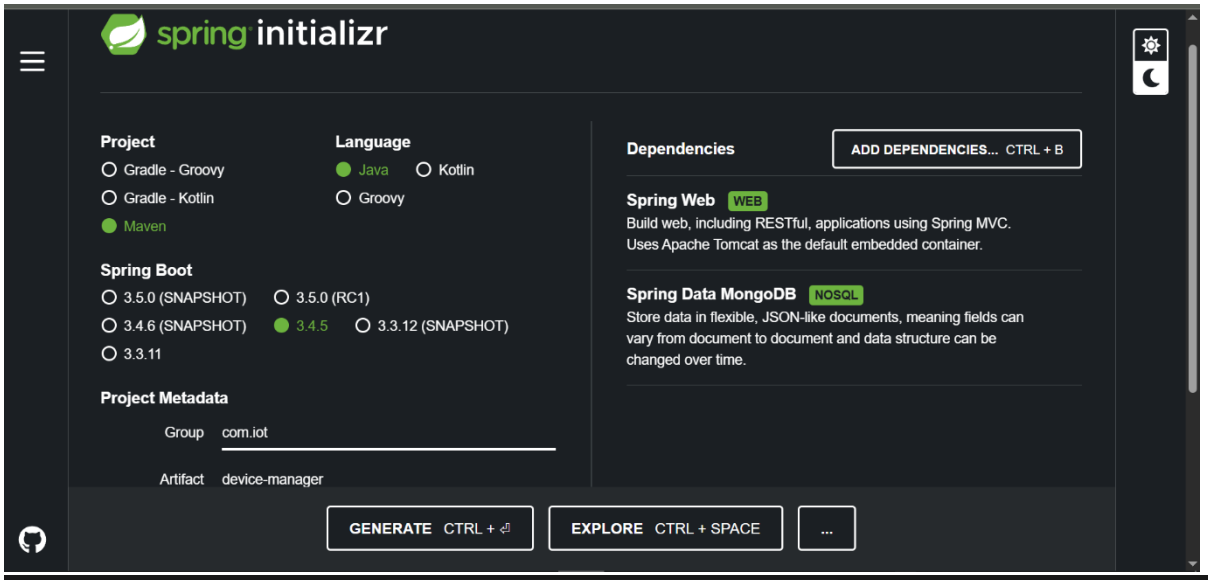
- Azure App Service: Supports autoscaling and zero-downtime deployments.
- Cosmos DB: Automatically scales throughput and handles high volumes of reads/writes.
- Monitoring: Enable Application Insights and Azure Monitor for logs, performance, and alerts.

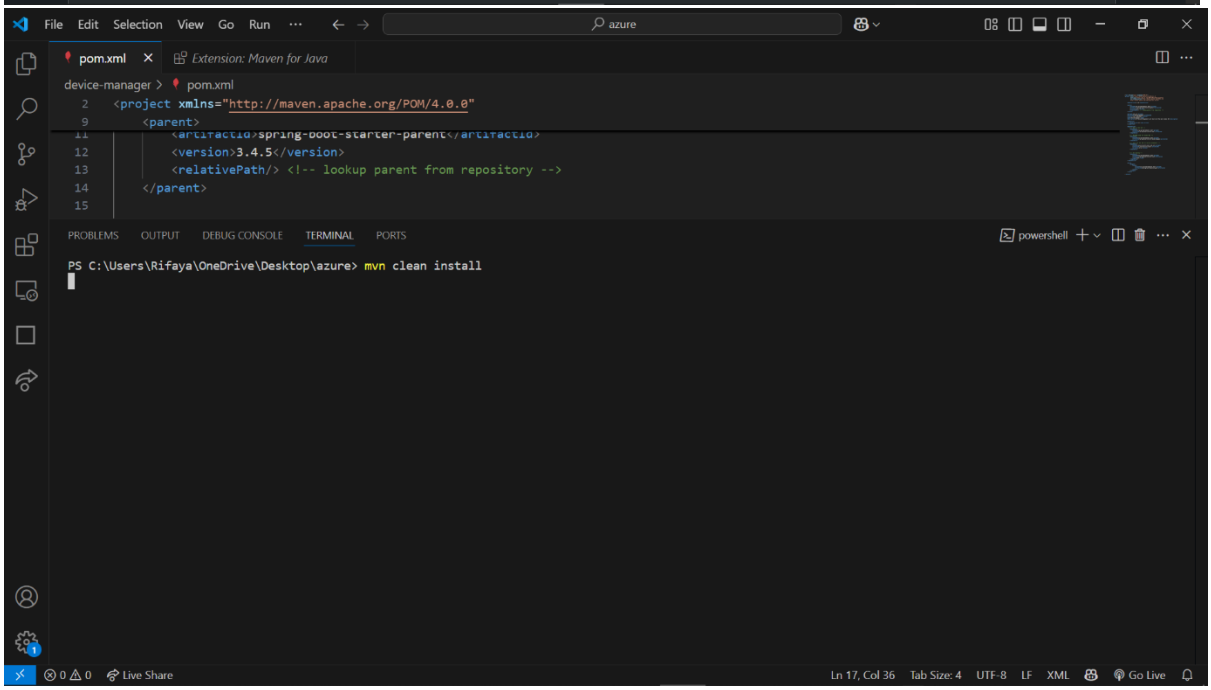
8. Conclusion

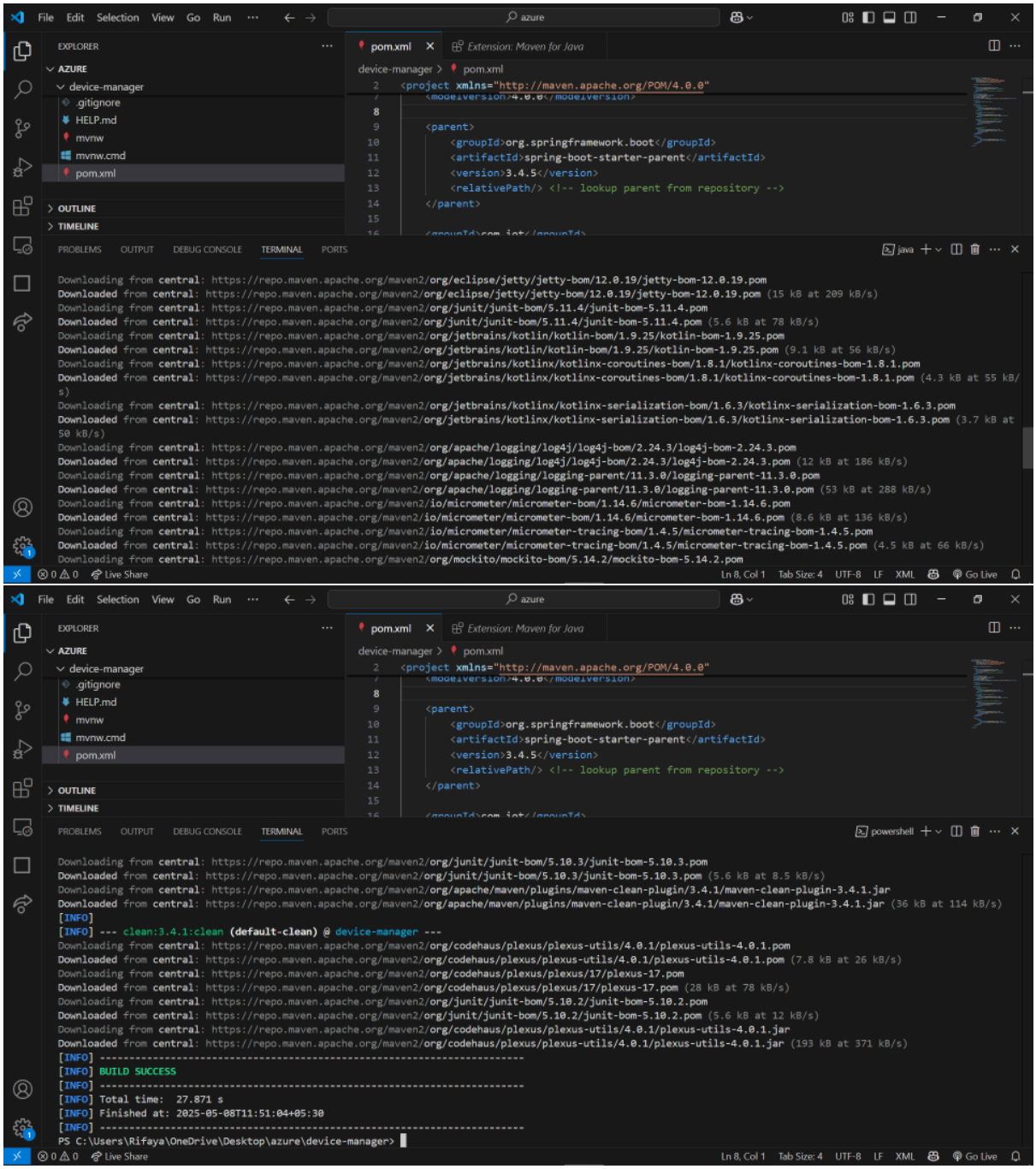
This deployment provides an end-to-end, scalable IoT device management solution built with Java and Azure services. With IoT Hub for secure device communication, Cosmos DB for fault-tolerant storage, and App Service for deployment, the solution provides real-time control and management through end-to-end CI/CD automation.

Screenshots:

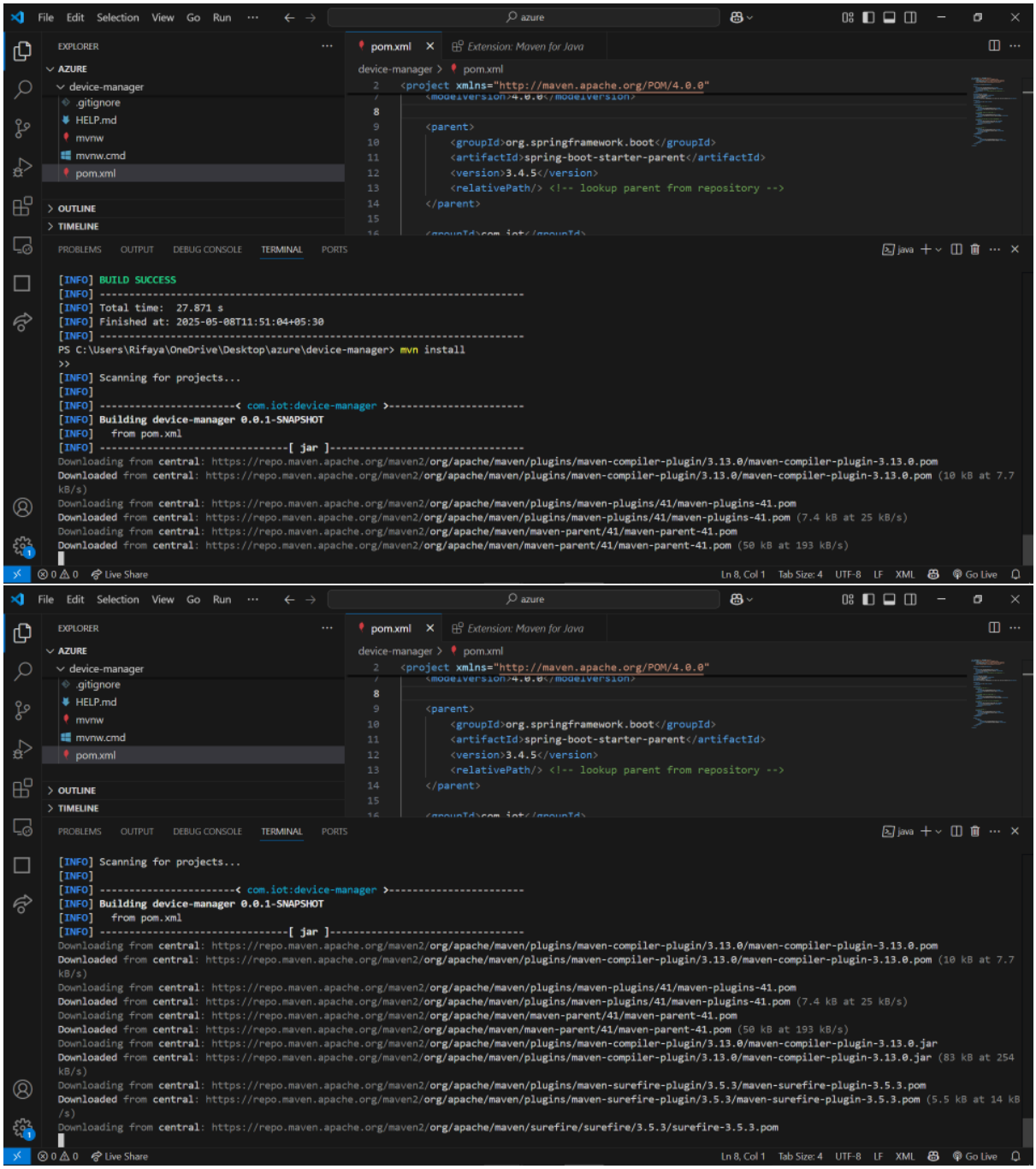



spring initializr
Project
Gradle - Groovy
Gradle - Kotlin
Maven
Language
Java
Kotlin
Groovy
Spring Boot
3.5.0 (SNAPSHOT)
3.5.0 (RC1)
3.4.6 (SNAPSHOT)
3.4.5
3.3.12 (SNAPSHOT)
3.3.11
Project Metadata
Group com.iot
Artifact device-manager
Dependencies
ADD DEPENDENCIES... CTRL + B
Spring Web WEB
Build web, including RESTful, applications using Spring MVC.
Uses Apache Tomcat as the default embedded container.
Spring Data MongoDB NOSQL
Store data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.
GENERATE CTRL +
EXPLORE CTRL + SPACE
...


File Edit Selection View Go Run
azure
pom.xml
Extension: Maven for Java
device-manager > pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
<parent>
<version>3.4.5</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\Users\Rifaya\OneDrive\Desktop\azure> mvn clean install
Live Share
Ln 17, Col 36 Tab Size: 4 UTF-8 LF XML Go Live


File Edit Selection View Go Run
azure
EXPLORER
AZURE
device-manager
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
OUTLINE
TIMELINE
pom.xml
Extension: Maven for Java
device-manager > pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.4.5</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
java
Downloading from central: https://repo.maven.apache.org/maven2/org/eclipse/jetty/jetty-bom/12.0.19/jetty-bom-12.0.19.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/eclipse/jetty/jetty-bom/12.0.19/jetty-bom-12.0.19.pom (15 kB at 209 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/junit-bom/5.11.4/junit-bom-5.11.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/junit-bom/5.11.4/junit-bom-5.11.4.pom (5.6 kB at 78 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/jetbrains/kotlin/kotlin-bom/1.9.25/kotlin-bom-1.9.25.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/jetbrains/kotlin/kotlin-bom/1.9.25/kotlin-bom-1.9.25.pom (9.1 kB at 56 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/jetbrains/kotlinx/kotlinx-coroutines-bom/1.8.1/kotlinx-coroutines-bom-1.8.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/jetbrains/kotlinx/kotlinx-coroutines-bom/1.8.1/kotlinx-coroutines-bom-1.8.1.pom (4.3 kB at 55 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/jetbrains/kotlinx/kotlinx-serialization-bom/1.6.3/kotlinx-serialization-bom-1.6.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/jetbrains/kotlinx/kotlinx-serialization-bom/1.6.3/kotlinx-serialization-bom-1.6.3.pom (3.7 kB at 50 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/logging/log4j/log4j-bom/2.24.3/log4j-bom-2.24.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/logging/log4j/log4j-bom/2.24.3/log4j-bom-2.24.3.pom (12 kB at 186 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/logging/logging-parent/11.3.0/logging-parent-11.3.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/logging/logging-parent/11.3.0/logging-parent-11.3.0.pom (53 kB at 288 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/io/micrometer/micrometer-bom/1.14.6/micrometer-bom-1.14.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/io/micrometer/micrometer-bom/1.14.6/micrometer-bom-1.14.6.pom (8.6 kB at 136 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/io/micrometer/micrometer-tracing-bom/1.4.5/micrometer-tracing-bom-1.4.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/io/micrometer/micrometer-tracing-bom/1.4.5/micrometer-tracing-bom-1.4.5.pom (4.5 kB at 66 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/mockito/mockito-bom/5.14.2/mockito-bom-5.14.2.pom
Live Share
Ln 8, Col 1 Tab Size: 4 UTF-8 LF XML Go Live

File Edit Selection View Go Run
azure
EXPLORER
AZURE
device-manager
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
OUTLINE
TIMELINE
pom.xml
Extension: Maven for Java
device-manager > pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.4.5</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/junit-bom/5.10.3/junit-bom-5.10.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/junit-bom/5.10.3/junit-bom-5.10.3.pom (5.6 kB at 8.5 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.4.1/maven-clean-plugin-3.4.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.4.1/maven-clean-plugin-3.4.1.jar (36 kB at 114 kB/s)
[INFO]
[INFO] --- clean:3.4.1:clean (default-clean) @ device-manager ---
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.pom (7.8 kB at 26 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/17/plexus-17.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/17/plexus-17.pom (28 kB at 78 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/junit-bom/5.10.2/junit-bom-5.10.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/junit-bom/5.10.2/junit-bom-5.10.2.pom (5.6 kB at 12 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.jar (193 kB at 371 kB/s)
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 27.871 s
[INFO] Finished at: 2025-05-08T11:51:04+05:30
[INFO] --
PS C:\Users\Rifaya\OneDrive\Desktop\azure\device-manager>
Live Share
Ln 8, Col 1 Tab Size: 4 UTF-8 LF XML Go Live


File Edit Selection View Go Run
azure
EXPLORER
AZURE
device-manager
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
OUTLINE
TIMELINE
pom.xml
Extension: Maven for Java
device-manager > pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.4.5</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
java
[INFO] BUILD SUCCESS
[INFO] Total time: 27.871 s
[INFO] Finished at: 2025-05-08T11:51:04+05:30
PS C:\Users\Rifaya\OneDrive\Desktop\azure\device-manager> mvn install
[INFO] Scanning for projects...
[INFO] < com.iot:device-manager >
[INFO] Building device-manager 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] [jar]
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler-plugin/3.13.0/maven-compiler-plugin-3.13.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler-plugin/3.13.0/maven-compiler-plugin-3.13.0.pom (10 kB at 7.7 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/41/maven-plugins-41.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/41/maven-plugins-41.pom (7.4 kB at 25 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/41/maven-parent-41.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/41/maven-parent-41.pom (50 kB at 193 kB/s)
Live Share
Ln 8, Col 1 Tab Size: 4 UTF-8 LF XML Go Live

File Edit Selection View Go Run
azure
EXPLORER
AZURE
device-manager
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
OUTLINE
TIMELINE
pom.xml
Extension: Maven for Java
device-manager > pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.4.5</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
java
[INFO] Scanning for projects...
[INFO] < com.iot:device-manager >
[INFO] Building device-manager 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] [jar]
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler-plugin/3.13.0/maven-compiler-plugin-3.13.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler-plugin/3.13.0/maven-compiler-plugin-3.13.0.pom (10 kB at 7.7 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/41/maven-plugins-41.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/41/maven-plugins-41.pom (7.4 kB at 25 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/41/maven-parent-41.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/41/maven-parent-41.pom (50 kB at 193 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler-plugin/3.13.0/maven-compiler-plugin-3.13.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler-plugin/3.13.0/maven-compiler-plugin-3.13.0.jar (83 kB at 254 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-surefire-plugin/3.5.3/maven-surefire-plugin-3.5.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-surefire-plugin/3.5.3/maven-surefire-plugin-3.5.3.pom (5.5 kB at 14 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire/3.5.3/surefire-3.5.3.pom
Live Share
Ln 8, Col 1 Tab Size: 4 UTF-8 LF XML Go Live

Perform batch analytics in Azure Synapse using SQL to summarize historical sensor logs

Azure Synapse Analytics

1. Introduction

- **Define Azure Synapse Analytics**: Azure Synapse is an integrated analytics service that accelerates time to insight across data warehouses and big data systems. It enables you to query data using serverless or provisioned resources at scale.
- **Purpose of Batch Analytics**: To summarize and extract meaningful trends from large volumes of historical IoT sensor logs for business insights.

2. Data Ingestion and Storage

- **Data Source**: Sensor logs stored in Azure Data Lake Gen2 as Parquet/CSV files.
- **Linked Service**: Create linked service in Synapse Studio to connect to Data Lake.
- **External Table Creation**: Use CREATE EXTERNAL TABLE to query data directly.

```
CREATE EXTERNAL TABLE sensor_data (
  device_id STRING,
  temperature FLOAT,
  humidity FLOAT,
  timestamp DATETIME
)
WITH (
  LOCATION = '/sensorlogs/',
  DATA_SOURCE = my_datalake,
  FILE_FORMAT = parquet_format
);
```

3. Batch Analytics with SQL

- **Summary Queries**: Use T-SQL for aggregation of sensor logs.

```
SELECT
```

```
 device_id,
 AVG(temperature) AS avg_temp,
 MAX(humidity) AS max_humidity,
 COUNT(*) AS total_records
FROM sensor_data
WHERE timestamp BETWEEN '2024-01-01' AND '2024-12-31'
GROUP BY device_id;
```

- **Purpose**: Identify performance trends, detect anomalies, and generate periodic reports.

4. Visualization and Reporting

- **Power BI Integration**: Connect Synapse workspace with Power BI to visualize summarized data.
- **Sample Visuals**: Device-wise average temperature trends, max humidity spikes.

5. Optimization and Scheduling

- **Performance Tips**: Use partitioning on timestamp, columnstore indexes.
- **Pipeline**: Create Synapse pipeline with Data Flow or Notebook activity for batch runs.
- **Trigger**: Time-based trigger to run analytics daily or weekly.

6. Monitoring and Logging

- **Monitor**: Use Synapse Monitor hub to track job performance and failures.
- **Log Analytics**: Integrate with Azure Monitor for deeper diagnostics.

7. Conclusion

- **Summary**: Batch analytics in Synapse provides scalable and efficient processing of sensor logs. Integration with Power BI enables real-time insights for business decisions.

GITHUB:

https://github.com/rifaya22/project_1