CSC8110 Cloud Computing

Coursework

Abdullah Sheikh

18 December 2015

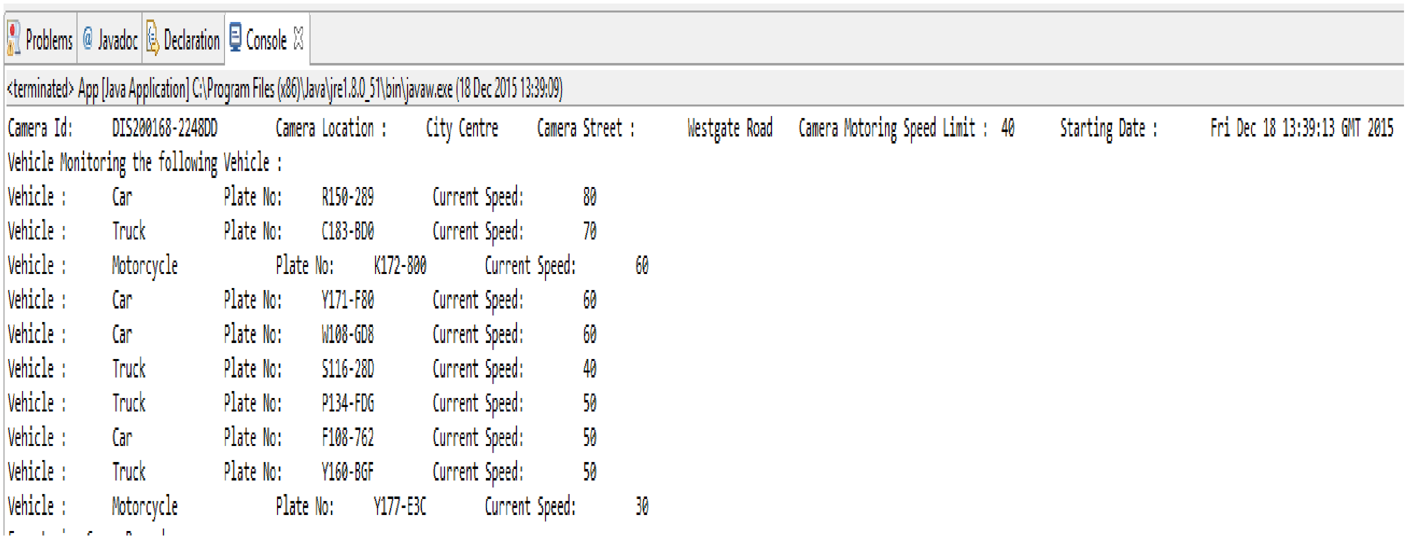
Introduction:

This course work will consider a network of a smart speed camera monitoring any passing vehicles to store all data in the cloud using Microsoft cloud azure.

Project Specification:

The project will design a system that covers the following:

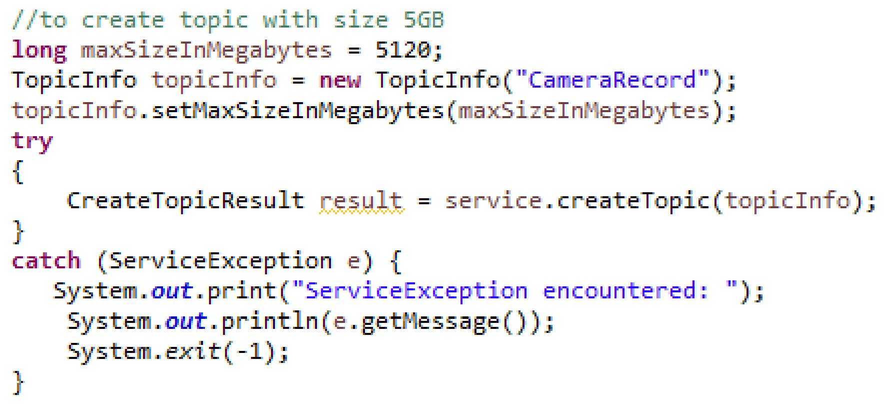
- Part1:
    - Creating a smart speed camera class as object with random values (id, location, street, and speed limit). Note, that the street name related to the location.
    - Creating vehicles as anew object with random value type from a list (Car, Truck, and Motorcycle). Also, it will create a random plate number and random current speed for each vehicles. These information will be send with the camera details that in sight.



```
Problems  Javadoc  Declaration  Console
<terminated> App [Java Application] C:\Program Files (x86)\Java\jre1.8.0_51\bin\javaw.exe (18 Dec 2015 13:39:09)
Camera Id:      DIS200168-2248DD        Camera Location :       City Centre     Camera Street :         Westgate Road   Camera Motoring Speed Limit :  40       Starting Date :         Fri Dec 18 13:39:13 GMT 2015
Vehicle Monitoring the following Vehicle :
Vehicle :       Car             Plate No:       R150-289        Current Speed:          80
Vehicle :       Truck           Plate No:       C183-BD0        Current Speed:          70
Vehicle :       Motorcycle              Plate No:       K172-800        Current Speed:          60
Vehicle :       Car             Plate No:       Y171-F80        Current Speed:          60
Vehicle :       Car             Plate No:       W108-GD8        Current Speed:          60
Vehicle :       Truck           Plate No:       S116-28D        Current Speed:          40
Vehicle :       Truck           Plate No:       P134-FDG        Current Speed:          50
Vehicle :       Car             Plate No:       F108-762        Current Speed:          50
Vehicle :       Truck           Plate No:       Y160-BGF        Current Speed:          50
Vehicle :       Motorcycle              Plate No:       Y177-E3C        Current Speed:          30
```

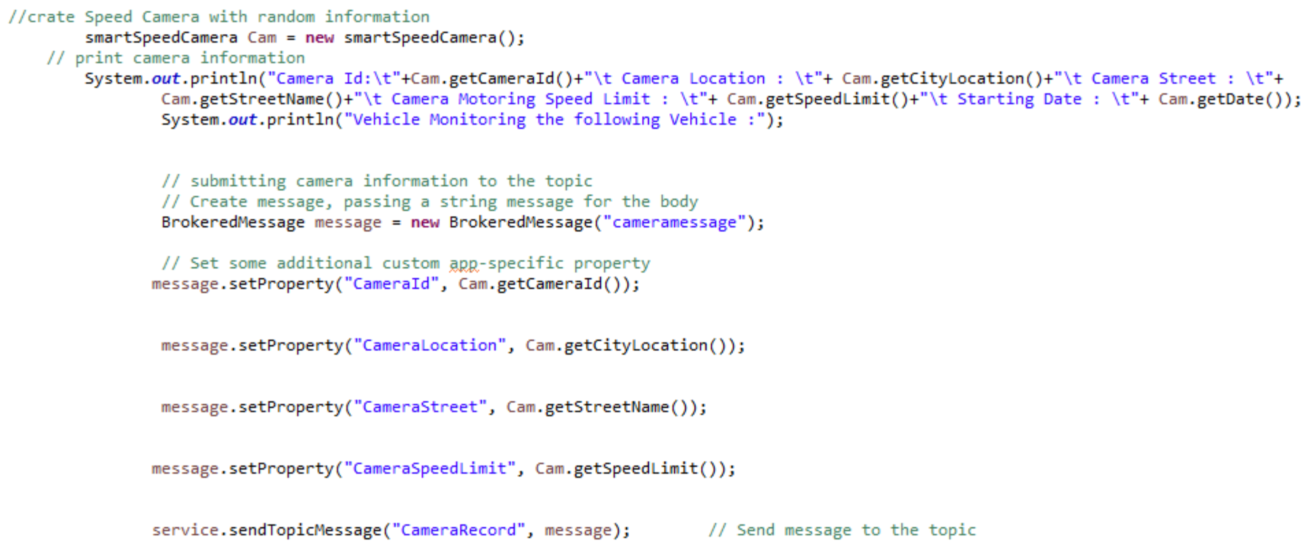
The picture shows the camera that turned on in city centre and all information required and all vehicles that passing from that sight with current speed check.

- Part2:
    - Then the project creates a service bus and topic and subscription in the cloud using Azure connection.
    - The Camera speed information will be send to the topic when it turn on.
    - All vehicles in this sight will be creating randomly regarding to the traffic rate which will be read from the project argument. The following figure shows code of how to create topic with 5GB, that needed to send all camera and vehicle information as messages.



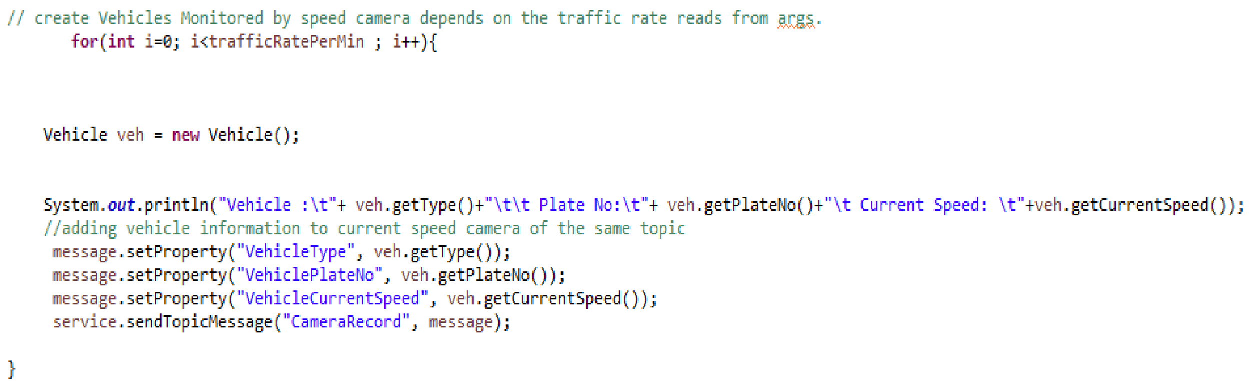
```java
//to create topic with size 5GB
long maxSizeInMegabytes = 5120;
TopicInfo topicInfo = new TopicInfo("CameraRecord");
topicInfo.setMaxSizeInMegabytes(maxSizeInMegabytes);
try
{
    CreateTopicResult result = service.createTopic(topicInfo);
}
catch (ServiceException e) {
   System.out.print("ServiceException encountered: ");
    System.out.println(e.getMessage());
    System.exit(-1);
}
```

Then it will create camera with random values and send it as messages to the topic



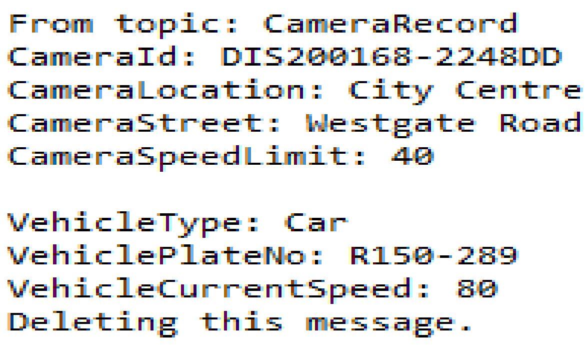
```java
//crate Speed Camera with random information
        smartSpeedCamera Cam = new smartSpeedCamera();
    // print camera information
        System.out.println("Camera Id:\t"+Cam.getCameraId()+"\t Camera Location : \t"+ Cam.getCityLocation()+"\t Camera Street : \t"+
                Cam.getStreetName()+"\t Camera Motoring Speed Limit : \t"+ Cam.getSpeedLimit()+"\t Starting Date : \t"+ Cam.getDate());
                System.out.println("Vehicle Monitoring the following Vehicle :");


                // submitting camera information to the topic
                // Create message, passing a string message for the body
                BrokeredMessage message = new BrokeredMessage("cameramessage");

                // Set some additional custom app-specific property
               message.setProperty("CameraId", Cam.getCameraId());


                message.setProperty("CameraLocation", Cam.getCityLocation());


                message.setProperty("CameraStreet", Cam.getStreetName());


               message.setProperty("CameraSpeedLimit", Cam.getSpeedLimit());


               service.sendTopicMessage("CameraRecord", message);        // Send message to the topic
```

Then creates vehicle depends on the traffic rates that will be read from argument as follows:



```java
// create Vehicles Monitored by speed camera depends on the traffic rate reads from args.
      for(int i=0; i<trafficRatePerMin ; i++){



    Vehicle veh = new Vehicle();


    System.out.println("Vehicle :\t"+ veh.getType()+"\t\t Plate No:\t"+ veh.getPlateNo()+"\t Current Speed: \t"+veh.getCurrentSpeed());
    //adding vehicle information to current speed camera of the same topic
     message.setProperty("VehicleType", veh.getType());
     message.setProperty("VehiclePlateNo", veh.getPlateNo());
     message.setProperty("VehicleCurrentSpeed", veh.getCurrentSpeed());
     service.sendTopicMessage("CameraRecord", message);

}
```

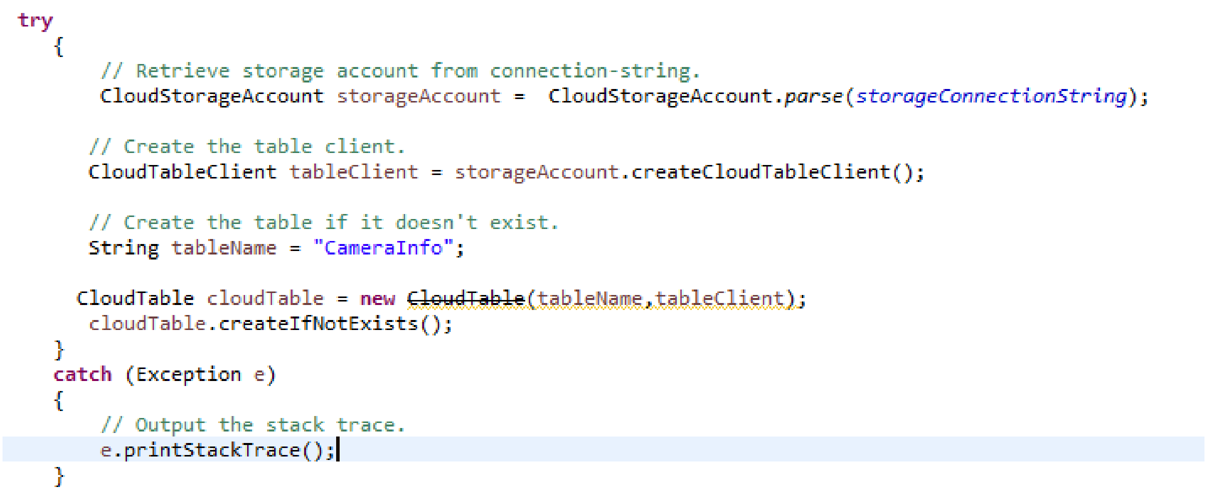
Also, it will be send directly to the topic in messages. For checking the system will retrieve all information in the topic depend on the vehicle with sighted camera as shown in the following figure as example of a message from a topic:



```
From topic: CameraRecord
CameraId: DIS200168-2248DD
CameraLocation: City Centre
CameraStreet: Westgate Road
CameraSpeedLimit: 40

VehicleType: Car
VehiclePlateNo: R150-289
VehicleCurrentSpeed: 80
Deleting this message.
```

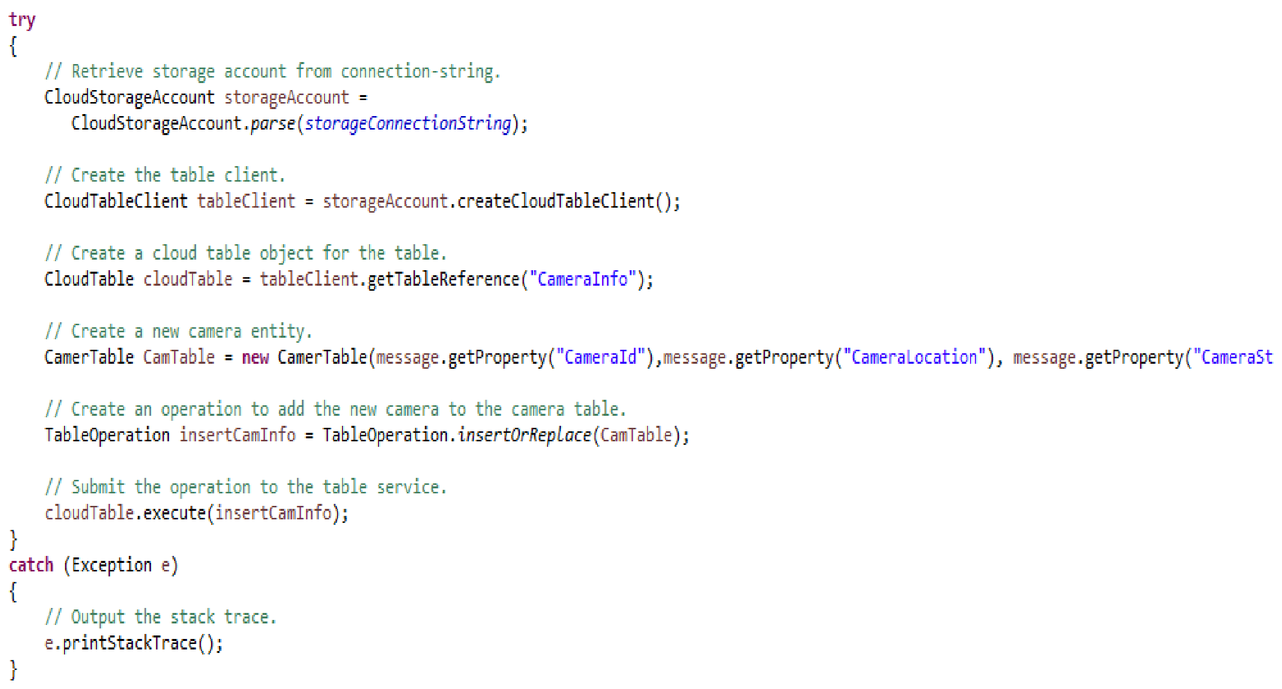
- Part3:
    - After sending messages to the topic, the system will create camera table to retrieve all camera information. Note, storage has been created from azure portal to get all connection configuration to this step.
    - Also, it creates a vehicle table to retrieve all vehicle information from the topic.
    - From these two tables the system will make query to get any information that needed to further process.

  The following figure shows the connection to the storage account and to get table information from the table listed into that account storage:



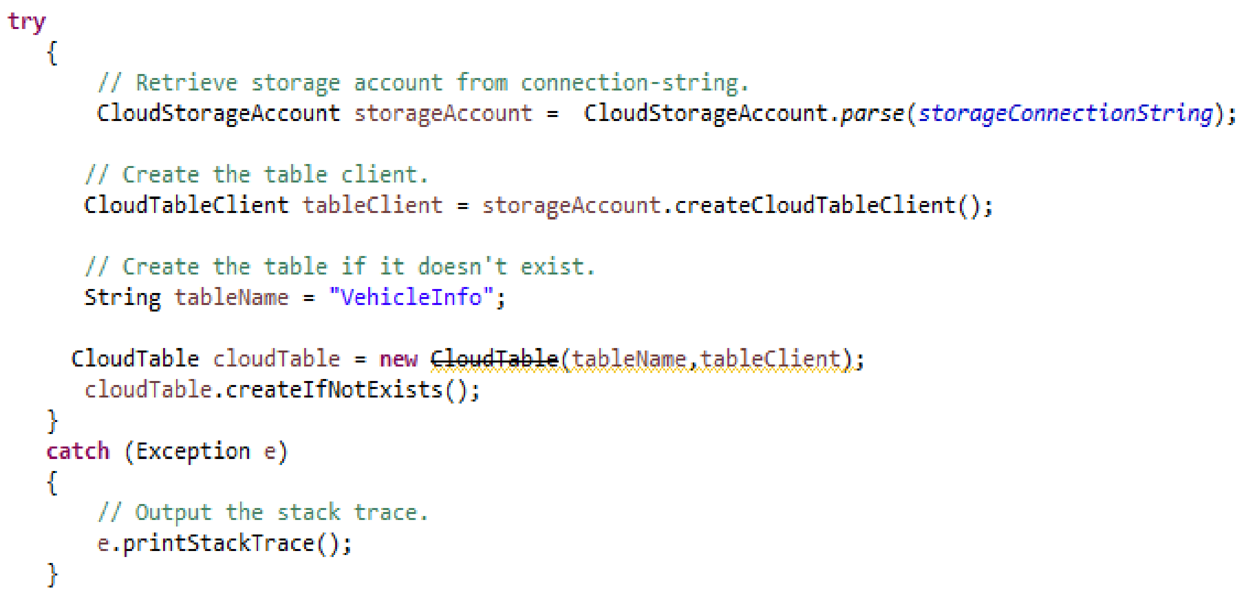
```
try
    {
        // Retrieve storage account from connection-string.
        CloudStorageAccount storageAccount =  CloudStorageAccount.parse(storageConnectionString);

       // Create the table client.
       CloudTableClient tableClient = storageAccount.createCloudTableClient();

       // Create the table if it doesn't exist.
       String tableName = "CameraInfo";

      CloudTable cloudTable = new ~~CloudTable~~(tableName,tableClient);
       cloudTable.createIfNotExists();
    }
    catch (Exception e)
    {
        // Output the stack trace.
        e.printStackTrace();
    }
```

The following figure shows how to send information to be stored as entities in camera table:



```java
try
{
    // Retrieve storage account from connection-string.
    CloudStorageAccount storageAccount =
       CloudStorageAccount.parse(storageConnectionString);

    // Create the table client.
    CloudTableClient tableClient = storageAccount.createCloudTableClient();

    // Create a cloud table object for the table.
    CloudTable cloudTable = tableClient.getTableReference("CameraInfo");

    // Create a new camera entity.
    CamerTable CamTable = new CamerTable(message.getProperty("CameraId"),message.getProperty("CameraLocation"), message.getProperty("CameraSt

    // Create an operation to add the new camera to the camera table.
    TableOperation insertCamInfo = TableOperation.insertOrReplace(CamTable);

    // Submit the operation to the table service.
    cloudTable.execute(insertCamInfo);
}
catch (Exception e)
{
    // Output the stack trace.
    e.printStackTrace();
}
```

Then, the following figure shows how to send information to be stored as entities in the vehicle table:



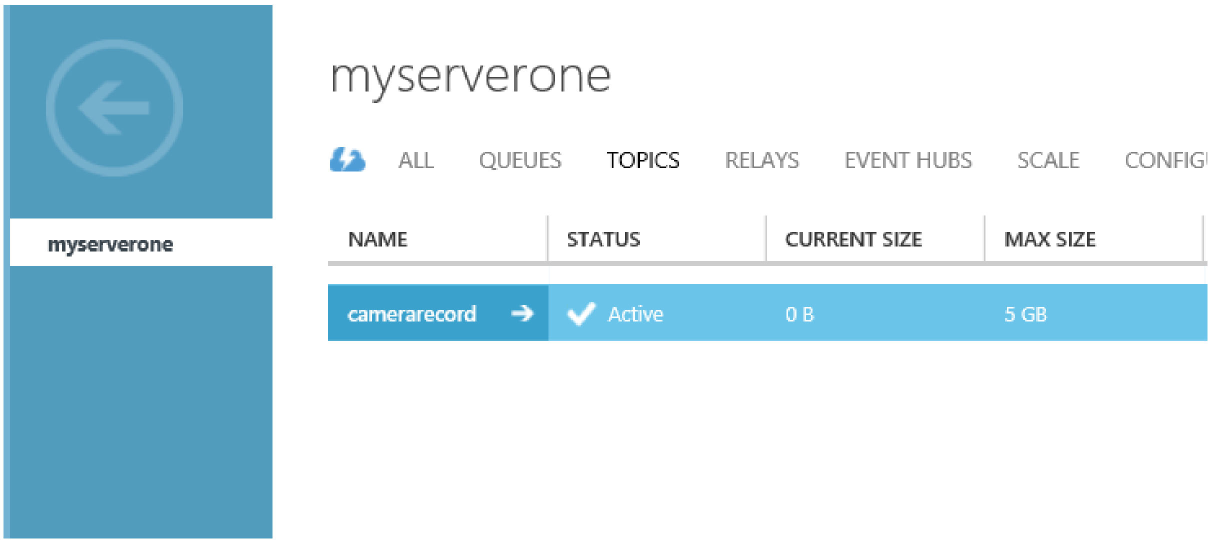
```java
try
  {
     // Retrieve storage account from connection-string.
     CloudStorageAccount storageAccount =  CloudStorageAccount.parse(storageConnectionString);

    // Create the table client.
    CloudTableClient tableClient = storageAccount.createCloudTableClient();

    // Create the table if it doesn't exist.
    String tableName = "VehicleInfo";

   CloudTable cloudTable = new ~~CloudTable~~(tableName,tableClient);
    cloudTable.createIfNotExists();
  }
  catch (Exception e)
  {
     // Output the stack trace.
     e.printStackTrace();
  }
```

- Part4:
  - Police Monitor.
- Part5:
  - Vehicle Check.
- Part6:
  - Relational Databases in Azure

Tools used:

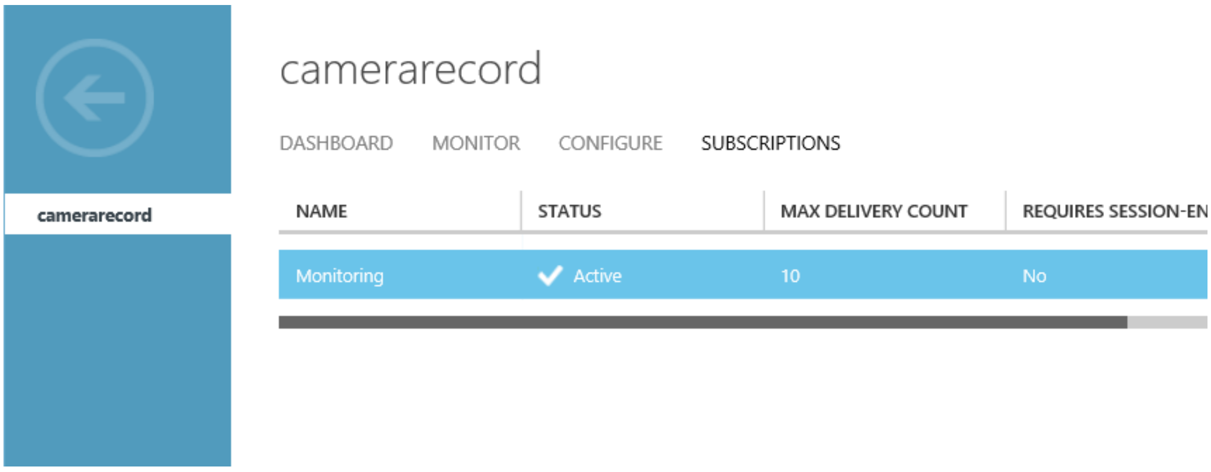
- GitHub: to commit all works regularly.
- Maven.
- Eclipse Java programming
- Windows Azure.

In windows azure, the following figure shows creating topic:



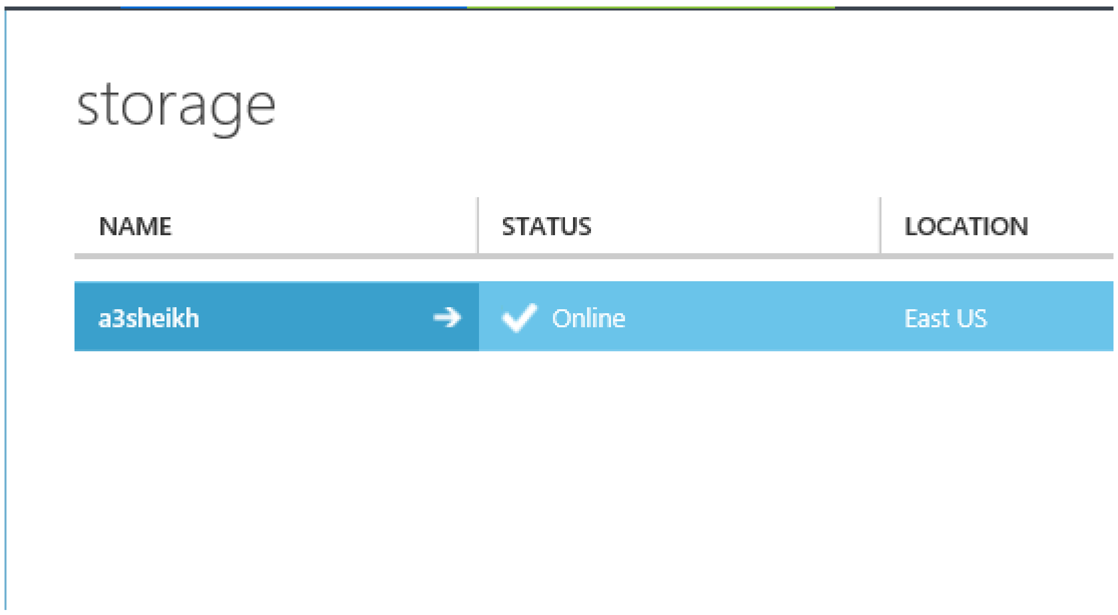
NAME	STATUS	CURRENT SIZE	MAX SIZE
camerarecord	✓ Active	0 B	5 GB

And creating Subscripting which is monitoring vehicles:



NAME	STATUS	MAX DELIVERY COUNT	REQUIRES SESSION-EN
Monitoring	✓ Active	10	No

And the storage account :



storage

NAME	STATUS	LOCATION
a3sheikh →	✓ Online	East US