

Scan Event Exercise

Objective

Your task is to create a worker application that consumes a scan event API and records the last event made against a parcel and any pickup or delivery times. The worker application should be built using .NET, but you can use any supporting libraries or tools that you see fit.

Requirements

1. The application must consume from a scan event API and keep a track of the last event that was fetched so that it can continue where it left off if the service is stopped and started.
2. The application should persist the scan event data in such a way that the following information could later be fetched easily;
 - a. The most recent scan event against a parcel, specifically only values from fields; EventId, ParcelId, Type, CreatedDateTimeUtc, StatusCode, RunId are required.
 - b. DateTimes indicating when a parcel has been;
 - i. Picked up (a scan event of Type 'PICKUP' has occurred).
 - ii. Delivered (a scan event of Type 'DELIVERY' has occurred).
3. The application should be fault tolerant and resilient (e.g. handle new event types, malformed data).
4. The application should contain appropriate logging.

Scan Event API Detail

1. For the purposes of development, you can assume a Scan Event API exists that has a single endpoint available e.g. GET <http://localhost/v1/scans/scanevents>
2. The scan event API endpoint above supports the following URL parameters;
 - a. FromEventId – return scan events with an EventId greater than or equal to this (defaults to 1).
 - b. Limit – the total number of scan events to return (defaults to 100).e.g. <http://localhost/v1/scans/scanevents?FromEventId=83269&Limit=100>
3. Sample Scan Event JSON;

```
{
  "ScanEvents": [
    {
      "EventId": 83269, //Unique Event ID
      "ParcelId": 5002, //The Parcel The Scan Event Applies To
      "Type": "PICKUP", //PICKUP, STATUS, DELIVERY
      "CreatedDateTimeUtc": "2021-05-11T21:11:34.1506147Z",
      "StatusCode": "",
      "Device": {
        "DeviceTransactionId": 83269,
        "DeviceId": 103
      },
      "User": {
        "UserId": "NC1001",
        "CarrierId": "NC", //NC, PH, CP, NW
        "RunId": "100"
      }
    }
  ]
}
```

4. The inline comments in the Sample Scan Event JSON are for explanatory purposes and do not appear in the scan event json returned.
5. EventId is always unique per event.
6. ParcelId represents the parcel that a scan is related to. A parcel can have one or more scans made against it.

Assumptions

1. List any assumptions you have made. If anything in these instructions is not clear, write down an assumption that captures this.

Improvements

1. List any improvements that could be made to the application you have built including what kind of things could be done to productionise this application.
2. Describe what kind of changes or additions could be performed to enable another worker application downstream to also perform actions against the same scan events processed by this application. Describe what the high-level architecture of this overall system could look like.

Deliverable

1. Submit the source code of your application as a zip file (or supply a link to a GIT repository) and list any additional scripts or external dependencies that may need to be installed or set up for the application to run.