Coding Assessment:

The goal is to implement a RESTful inventory tracking web service. Clients can request information about the entire inventory, or a single item. Resource identifiers are assigned as follows:

- "/inventory" refers to the entire inventory
- "/inventory/Apples" refers to the item named "Apples" in the inventory

The data accepted and generated by the web service is encoded using JSON.

The web service listens for requests on port 8081. So, for example, a **GET** request to the URL

http://localhost:8081/inventory should return the initial inventory as a JSON array, which is

```
[
 {
  "name": "Apples",
  "quantity": 3 ,
 “createdOn”: ”2020-01-01”
 },
 {
  "name": "Oranges",
  "quantity": 7 ,
  “createdOn”: ”2020-02-01”
 },
 {
  "name": "Pomegranates",
  "quantity": 55,
  “createdOn”: ”2020-01-10”
 }
]
```

The starting point code includes a handler for the HTTP **GET** method which allows the client to access

the entire inventory, or a single item. So, for example, the URL

http://localhost:8081/inventory/Apples should return a single item as a JSON object:

```
{
 "name": "Apples",
 "quantity": 3 ,
 “createdOn”: ”2020-01-01”
}
```

Service Methods to implement:

Handle HTTP **PUT** requests as follows:

- If the resource identifier is "/inventory/*itemname*"
    - Update the entry if found.
    - Create new record if not found

Handle HTTP **POST** requests as follows:

- If the resource identifier is "/inventory"
    - The input can be a single entry or multiple.
        - Handle updates and creation of documents based on inventory store.

Handle **DELETE** requests as follows:

- If the resource identifier is "/inventory/*itemname*", the named item is deleted

Note that delete requests have no body.

Business Functionality

1. Provider functionality for the user to query the inventory based on
    - Name
    - Highest quantity (user queries the inventory for the Highest quantity item)
    - Lowest quantity (user queries the inventory for the Highest quantity item)
    - Oldest item
    - Newest item
    - Any additional functionality you might think will be useful to the user.

2. Allow the service to handle multiple requests
    - 2 posts requests
    - 2 deletes
        - Logic to handle edge cases

Testing

Create a way for to show these functionalities in action and well as showcase that all of the business requirements have been fulfilled.