MyCart.com stores requires the following services to be implemented to support its ECOM business. The application uses mongo DB to store the product details.

The item details are stored in the item collection in the following format



The category collection contains the category information.



Technical Specification/Guidelines

* Use Spring boot with Camel as the technical stack
* All the configurable values like error messages, number of records to be processed, time limit etc. needs to be externalized from the code and stored in property files
* Use camel component wherever possible, minimize the use of java beans when there is in built components available in camel.

Examples:

* Use marshal for converting JSON/XML
* Use unmarshal for converting from JSON to POJO
* Use Camel Split component for iterating over list of objects – avoid using java for loop
* Use camel components for parallel processing instead of direct use of Java Thread API
* Conditions s
* hould be checked and routed accordingly from Camel instead of using java statement.

Sample

If(!productExist){

Then create product

}

Else {

Update product

}

Needs to be replaced with camel component that redirects to the create or update conditions based on “productExist” value

* Use camel rest instead of Spring boot Rest Controller classes.
* All the IO operations from mongo, AMQ, SFTP needs to be strictly developed using Camel components
* Exception Handling- Every route must explicitly define exception handling. A global, onException(Throwable), must be specified in each route
* Route Style- Imbedded, inner Java classes are prohibited in Camel routes. Processors must be used when practical rather than beans. Use of built in Camel functionality must be used when possible rather than implementing code in a processor or bean (i.e. Marshalling/Unmarshalling JSON to/from POJO, transforming XML)
* Wherever unmarshalling/marshalling using Camel components is not possible, use Jackson API for JSON parsing

**Requirement 1**

I need a set of web services to meet the following requirements.

* Requirement 1:

I need a GET webservice that accepts an item\_id and returns the item details as JSON file in the below format. (Basic Rest)



* Requirement 2: I need a GET service that accepts a category ID and returns all the items under that category in the below format. This service needs to accept a filter “includeSpecial”, based on the value the service needs to decide whether to include the special products in the output.

Sample URL: http://<servername>/mycart/items/{category\_id}?includeSpecial=true

The items in the response needs to be grouped based on the category like



* I need a POST service that accepts a new item in JSON format and insert into the database only if the item doesn’t exist. If the category is invalid, then return error response.

The base price and selling price should be greater than zero

Request:



Response:



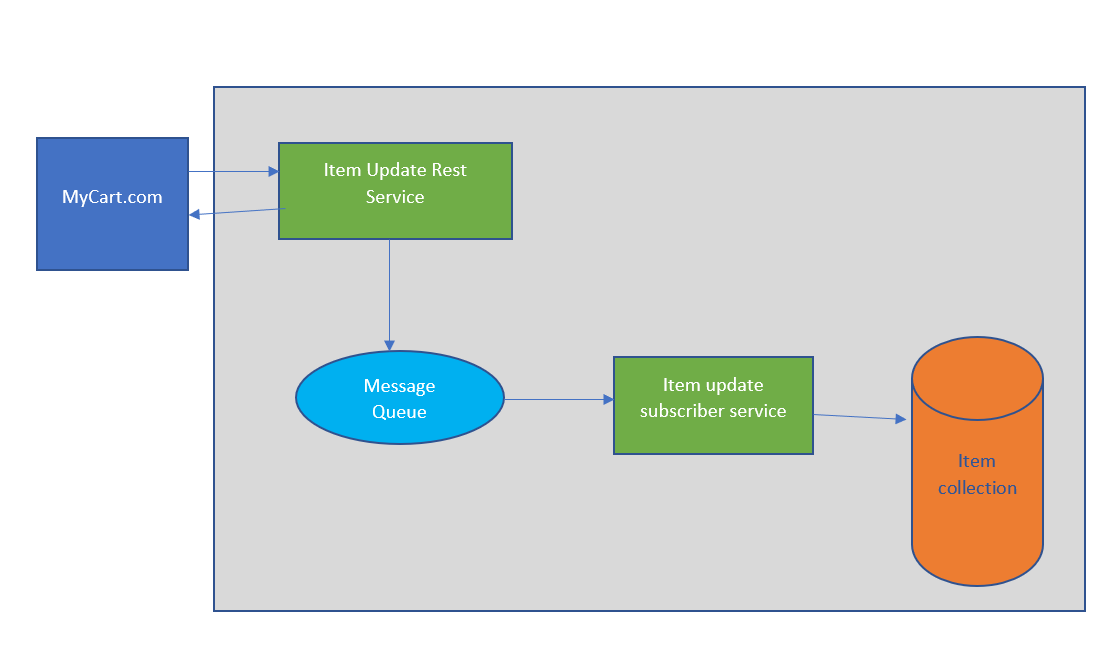
* I need a rest service that accepts the item inventory list in the below format and update each item document in the collection.



availableStock=availableStock-soldOut-damaged, where soldOut and damaged needs to be fetched from the inventory payload.

Requirement 2

My inventory update service is facing performance issues as the number of items in the payload increases. I need a change in architecture such that the records are processed asynchronously. The post services accept the JSON and makes it available for asynchronous service. I need a camel service that reads the JSON and process the records one by one and update the item collection.



Requirement 3

My items details need to be accessed by other supporting applications that reads the files from an SFTP folder. There are three third party applications that needs the item details for various business processes, these applications are provided as SAAS and need to be developed by MyCart.com. MyCart.com just needs to send the files to a SFTP location so that the applications can consume.

Item Trend Analyzer: This application requires the output in the following XML format



Item Review Aggregator: This application accepts the review details for each product in an XML and use it for making some business decisions.



StoreFront App: This is an application used by MyCart.com that is functioning from offline stores. The item details from the item and category collection is placed in an SFTP folder so that the same may be consumed by the StoreFrontApp.



We need to have a scheduler-based event that reads the item records from the collection and sends it to the above three applications in the prescribed format. The processed time and date need to be stored in a collection named “ControlRef” so that each execution picks the items whose lastupdateTs is greater that the lastProcessTs in the controlRef.

To improve the performance of the application the architect suggests processing all three records for an item parallelly and push it to SFTP.

Requirement 4

MyCart.com is developing a new application “MyCart store” for its offline processing of orders. It wants to leverage on the existing item get service to fetch the item details from the item collection.

Develop a camel service for the application “MyCart store” that reads the item details by invoking the rest service developed earlier. Use Camel rest components to invoke the rest service.

Requirement 5

After deploying the service developed as part of the requirement 3, I am facing performance issues due to the slowness in SFTP. SFTP is not able to accept the files in the same rate as it is generated by the service. So, we need to have a controlling mechanism that restricts the number of records moving into SFTP from the service. The processed records need to be regulated, like a specific number of records in a time interval, for example 100 records in a minute.

Requirement 6

There have been multiple instances during which the mongo db was temporarily down but was working when tried multiple times. To make the applications verbose and not to fail on temporary outages, I want the services to try more than once before failing the process. The number of retries needs to be configured in the property file. Also, to avoid rapid back-to-back to tries which may break the system, I want the service to return with increased time gap between each retries.

For example, try after 1 min, 3 min, 6 min etc.