Entity Management System

Author: Amit Karande

This document mentions about problem statement given, assumptions considered for its scope, design used to develop entity management case study. The entity management system (EMS) is developed for product in catalog scenario. The product has different attributes and sub-entities.

**Preamble**

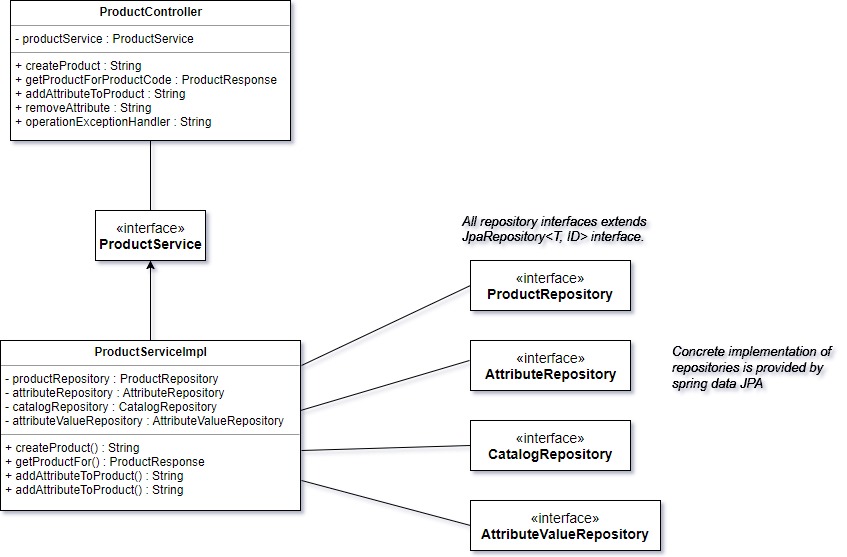
The problem statement is open ended and the domain is very vast hence attempt is made to demonstrate how system works in scenario of product entity in catalog from e-commerce domain. Each product can have multiple attributes which can be of different type e.g. multi values, select, free text etc. Also the product entity is showed to have reviews as sub-entity. Following section explains the design (both back end & DB) & RESTful APIs developed.

Due to time constraint, attempt is made to develop core APIs for attribute, product, catalog & review management and altering their values & associations.

**Technologies used**

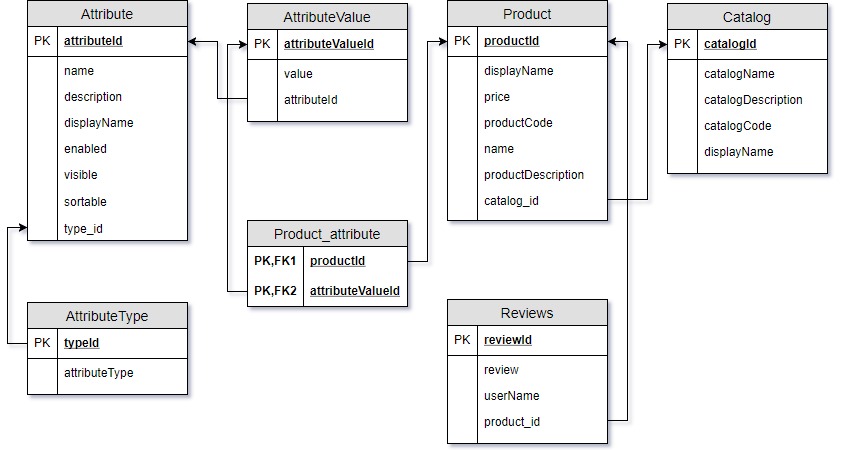
The application is developed using spring boot with **spring-data-JPA**, **spring-data-REST** primarily. It maintain the data using **in memory H2 database**. The application uses **Java 8**.

The implementation follows controller, service, repository pattern (DAO-DTO). The controllers handle responsibility as per Entity type e.g. AttributeController manages attribute entity responsibility, similar is the case for Product, Catalog and Review. Following is a sample class diagram for Product Entity management. Others are very much similar.



**Entity Model and DB design**

This is one of the major challenge in implementing this case study. The entity modeling and DB design follow combination of traditional relational DB design and **Entity-Attribute-Value (EAR)** model. The application is designed to evolve the attributes separately from entity/entities to which they apply to. Entity simply associates itself to attribute thereby having **loose coupling** between entity and attribute.



**APIs**

Following are the APIs implemented for case study. The table describes the API usage, API, sample payload and response.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| API For | API | Method | Sample payload | Response |
| Create Attribute | /api/attribute | POST | {  "name" : "audio",  "displayName" : "Audio",  "description" : "Audio output of TV set",  "visible" : 1,  "sortable" : 1,  "enabled" : 1,  "attributeType" : "integer",  "attributeValues" : ["16"]  } | created attribute:audio |
| Get Attribute details | api/attribute/{attributeName} | GET | api/attribute/audio | {  "name": "audio",  "displayName": "Audio",  "description": "Audio output of TV set",  "sortable": 1,  "visible": 1,  "enabled": 1,  "attributeType": "integer",  "attributeValues": ["16"]  } |
| Update Attribute, values | /api/attribute | PUT | {  "name" : "audio",  "displayName" : "Audio",  "description" : "Audio output of TV set",  "visible" : 1,  "sortable" : 1,  "enabled" : 0,  "attributeType" : "integer",  "attributeValues" : ["17"]  } | Updated attribute |
| Create Catalog | /api/catalog | POST | {  "catalogName" : "TV",  "catalogCode" : "FLATTV1125",  "displayName" : "Flat TVs",  "description" : "Flat television sets"  } | Created catalog TV |
| Create Product | /api/product | POST | {  "productName" : "Samsung flat tv",  "displayName" : "Flat 54' tv by Samsung",  "productCode" : "SAMSUNG54FLAT",  "productDescription" : "Flat 54 inch UHD TV by Samsung",  "catalogCode" : "FLATTV1125",  "price" : 51345.35,  "attributes" : [{"attribute" : "tvsize", "attributeValue" : "54"}, {"attribute" : "weight", "attributeValue" : "14"}]  } | Created product Samsung flat TV |
| Get product | /api/product/{productCode} | GET | /api/product/SAMSUNGFLAT54UHD | {  "productName": "Samsung Flat TV",  "displayName": "Flat 54 tv by Samsung",  "productCode": "SAMSUNGFLAT54UHD",  "productDescription": "Flat 54 inch UHD TV by Samsung",  "catalogCode": "FLATTV1122",  "price": 51343.55,  "attributes": [  {  "attribute": "tvweight",  "attributeValue": "15"  },  {  "attribute": "keywords",  "attributeValue": "SMART, HDTV, BIGTV"  },  {  "attribute": "tvsize",  "attributeValue": "38"  }  ],  "reviews": [  {  "review": "Awesome product! go for it",  "userName": "anonymous"  }  ]  } |
| Add attribute to product | /api/product/{productCode}/{attribute} | PUT | /api/product/SAMSUNGFLAT54UHD/keywords | Attribute added |
| Get all attributes | /api/attribute/all | GET | /api/attribute/all | [  {  "name": "tvsize",  "displayName": "TV Size",  "description": "size of television set",  "sortable": 1,  "visible": 1,  "enabled": 1,  "attributeType": "select",  "attributeValues": ["38","62","54"]  },  {  "name": "tvweight",  "displayName": "TV Weight",  "description": "Weight of television set in kg",  "sortable": 1,  "visible": 1,  "enabled": 1,  "attributeType": "integer",  "attributeValues": ["15" ]  },  {  "name": "keywords",  "displayName": "Keywords",  "description": "Keywords for TV",  "sortable": 1,  "visible": 1,  "enabled": 1,  "attributeType": "multiple",  "attributeValues": ["SMART","HDTV","BIGTV"]  }  ] |
| Remove product attribute | /api/product/ | PUT | {  "productCode" : "SAMSUNGFLAT54UHD",  "attributeList" : ["keywords"]  } | Attribute Removed |
| Submit review | /api/review/{productCode} | POST | {  "review" : "Awesome product! go for it"  } | Review submitted successfully |

References:

<https://en.wikipedia.org/wiki/Entity%E2%80%93attribute%E2%80%93value_model>