Django DRF Ecommerce

Phase 1

Inventory RESTful API

V1.0.1

Table of Contents

1. Introduction

1.1. Purpose of Document

This is a Requirements Specification document for a new web-based product inventory RESTful API system.

1.2. Project Scope

The scope of this project is a web-enabled inventory system RESTful API interface that supports client product data requests.

1.3 Responsibilities

The primary responsibilities of the new system:

- The new system will provide a client interface returning detailed and accurate update-to-date product information.

2. Functional Objectives

2.1. High Priority

1. The system shall provide the following data collection API endpoints:
   a. Return all categories
   b. Return a specified product and associated metadata
   c. Return Product(s), including associated product metadata from a specified category
2. The system shall allow employees to add and administrate product inventory
3. The system shall reflect new and changed products and product data changes x minutes of the database being updated by the product owner.

3. Supportability

- The system should be able to accommodate new products and product lines

- The system should support multiple types of product types with varying characteristics. In addition, the system should accommodate physical shippable and downloadable products.

4. **Security**
    - The system will provide password-protected access to product data management and administration.

5. **RESTful API documentation**
    - The system shall provide web-based documentation detailing all API endpoints and endpoint-specific details.

6. **Interfaces**

The system must interface with

- An SQLite database and be compatible with future migration to other database technologies.

7. **Context Diagram**

Appendix A: Functional Requirements

User Requirements

- Return a single product and associated product lines
- Return a list of all products
- Return a list of products by category
- Return a list of products by product attribute
- Return a list of new products
- Sort a list of products by price

Low Priority

- Add a review for a product
- Return a list of products on promotion

Business Requirements

- Add, Update or Delete new products
- Add, Return or Update details related to product stock levels
- Report: Return a list of total products sold per product
- Report: Return a list of newly added products for a given timeframe
- Report: Return a list of low or not-in-stock products
- Sort a list of products by date
- Return the name of the user who entered the product into the database
- When was the product added to the database?

System Requirements

- Return product data and images
- Return a single product and associated sub-product

Appendix B: User Story Analysis

Overview	
Title	Customer Product Browsing Behaviour
Description	Identifying a basic customer behavioural interaction when browsing products
Actors and Interfaces	Customer / Web User
Initial Status and Preconditions	Assumption that customer enters from the root/homepage
Basic Flow	
Step1: Land on the homepage Step2: Select a product category Step3: Browse, select, and view individual products related to the selected category Step4: Select and view individual product-line details	
Alternative Flow(s)	
▪ Customers may prefer searching for the product using keyword search features ▪ Customers may navigate to a product from an internal promotional panel	

User Type	**Activity**	**User Story**
Web User	Browse Products	Step1: Land on the homepage
		Step2: Select a product category
		Step3: Select and view individual products related to the selected category
		Step4: Inspect individual product-line details

Functional Specifications	**Status**
Return all categories	✔

Return all products filtered by category	✔
Return individual product and product-line by (x)	✔

Appendix C: Database Table Specification

Category			
Key	**Field Name**	**Data Type**	**Field Level Physical/Logical Constraints**
PK	id	BigAutoField	
	name	CharField	max_length = 235 unique=True
	slug	SlugField	max_length = 255 unique=True
	is_active	BooleanField	Default=False
FK	parent	TreeForignKey	on_delete=PROTECT null=True Blank=True
Custom Validation			
Methods/Behaviours			
1	Object string representation = name		
2	is_active() model manager queryset method to filter products by is_active Boolean field		
Test Cases / Log			
id	**Test Description/Expectation**		
1	Return error when field name max_length > 235		
2	Return error when field slug max_length > 255		

3	Return unique name error when entering a name which already exists in the category table
4	Return unique name error when entering a slug which already exists in the category table
5	Field is_active on creating a new record = false
6	Deleting a parent category raises ProtectedError
7	On inserting a new record, parent field to remain null
8	is_active() model manager queryset method to return objects where is_active field = True
9	Default object manager to return all products when used with all() method

Product			
Key	**Field Name**	**Data Type**	**Field Level Physical/Logical Constraints**
PK	id	BigAutoField	
	name	CharField	max_length = 235 unique=True
	slug	SlugField	max_length = 255 unique=True
	pid	CharField	max_length = 10 unique=True
	description		
	is_digital		Default=False
FK	category		on_delete=PROTECT null=True Blank=True
	is_active	BooleanField	Default=False
	created_at	DateTimeField	
FK	product_type		
M2M	attribute_value		

Custom Validation	
Methods/Behaviours	
1	Object string representation = name
2	is_active() model manager queryset method to filter products by is_active Boolean field
Test Cases / Log	
id	**Test Description/Expectation**
1	Return error when field name max_length > 235
2	Return error when field slug max_length > 255
3	Return error when field pid max_length > 10
4	Field is_digital on creating a new record = false
5	Deleting a category parent raises ProtectedError
6	is_active() model manager queryset method to return objects where is_active field = True
7	Default object manager to return all products when used with all() method
8	Object string representation return = {name field}

Product Line			
Key	**Field Name**	**Data Type**	**Field Level Physical/Logical Constraints**
PK	id	BigAutoField	
	price	DecimalField	decimal_places=2 max_digits=5
	sku	CharField	max_length=10
	stock_qty	IntegerField	
FK	product		on_delete=PROTECT
	is_active	Boolean	default=False
	order	CustomField	blank=True

	weight	FloatField	
	created_at	DateTimeField	editable=False,
FK	product_type		
M2M	attribute_value		
Custom Validation			
Methods/Behaviours			
1	Object string representation = sku		
2	is_active() model manager queryset method to filter products by is_active Boolean field		
Test Cases / Log			
id	**Test Description/Expectation**		
1	Return error when price field decimal places > 2		
2	Return error when price field max digits > 5		
3	Return error when field sku max_length > 10		
4	Field is_active on creating a new record = False		
5	Deleting a product parent raises ProtectedError		
6	is_active() model manager queryset method to return objects where is_active field = True		
7	Default object manager to return all objects when used with all() method		
8	Object string representation return = {sku field}		
9	Return error when a new record order field number is not unique		

Product Image			
Key	**Field Name**	**Data Type**	**Field Level Physical/Logical Constraints**
PK	id	BigAutoField	
	alternative_text	CharField	max_length=100
	url	ImageField	

	order	CustomField	
FK	productline		
Custom Validation			
Methods/Behaviours			
1	Object string representation = {product_line sku}		
2	is_active() model manager queryset method to filter products by is_active Boolean field		
Test Cases / Log			
id	**Test Description/Expectation**		
1	Return error when field alternative_text max_length > 100		
2	Return error when a new record order field number is not unique		