Problem Statement: Analyzing Sales Data with PySpark and Building API with FastAPI

You are given a large dataset containing sales information from multiple stores. Your task is to use PySpark to process this data and extract meaningful insights. After processing the data, you need to build a RESTful API using FastAPI to provide access to the analyzed results.

Task 1: Data Processing with PySpark

1. Load the sales data into a PySpark DataFrame.
2. Perform necessary data cleaning and preprocessing (e.g., handling missing values, data type conversions, etc.).
3. Analyze the sales data to answer the following questions:
    - Total sales revenue per store.
    - Total number of products sold per store.
    - Average sales price per product category.
    - Top selling product in each store.

Task 2: Building API with FastAPI

1. Create endpoints to retrieve the following information:
    - Total sales revenue per store.
    - Total number of products sold per store.
    - Average sales price per product category.
    - Top selling product in each store.
2. Design the API to accept parameters such as store ID, product category, etc., for filtering the results.
3. Ensure that the API responses are in JSON format.

Requirements:

- The PySpark code should be efficient and capable of handling large datasets.
- The API endpoints should be well-documented and follow RESTful principles.
- Implement error handling in both PySpark code and FastAPI endpoints.
- Optimize the API for performance.
- Write clear and concise documentation for both the PySpark code and the API endpoints.

Evaluation:

Your solution will be evaluated based on the following criteria:

1. Accuracy and efficiency of the PySpark data processing code.
2. Robustness and performance of the FastAPI implementation.
3. Clarity and organization of the code.
4. Documentation quality and completeness.

Submission:

Submit your solution as a GitHub repository containing the following:

1. PySpark code for data processing.
2. FastAPI code for building the API.
3. Documentation explaining how to run the code and interact with the API.
4. Any additional notes or explanations deemed necessary.

Note:

Ensure that you adhere to best practices for both PySpark and FastAPI development. This includes but is not limited to using appropriate data structures, leveraging parallel processing where applicable, handling errors gracefully, and following PEP 8 style guidelines.

Data Example:
StoreID,ProductID,ProductName,Category,SalesPrice,Quantity,SaleDate 1,101,Product A,Electronics,500,10,2024-03-01 1,102,Product B,Electronics,700,5,2024-03-02 1,103,Product C,Home Appliances,300,8,2024-03-03 2,104,Product D,Clothing,50,20,2024-03-01 2,105,Product E,Clothing,80,15,2024-03-02 2,106,Product F,Electronics,1000,3,2024-03-03 3,107,Product G,Home Appliances,400,12,2024-03-01 3,108,Product H,Electronics,600,7,2024-03-02 3,109,Product I,Electronics,900,4,2024-03-03