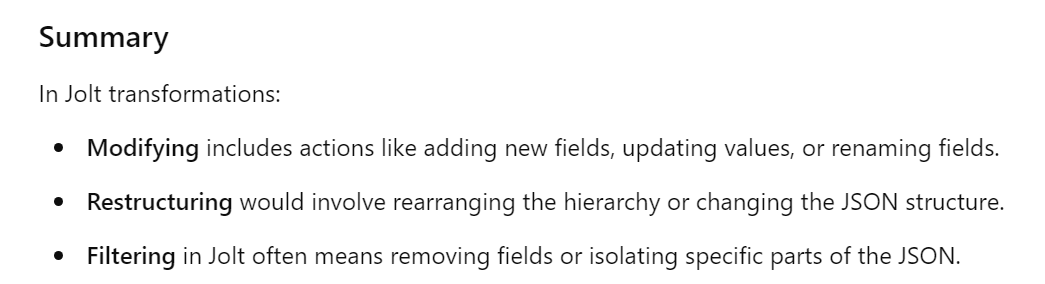
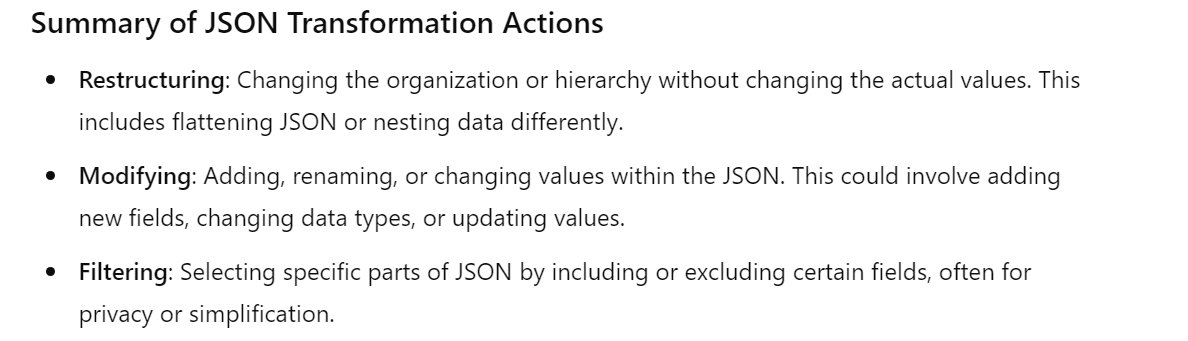
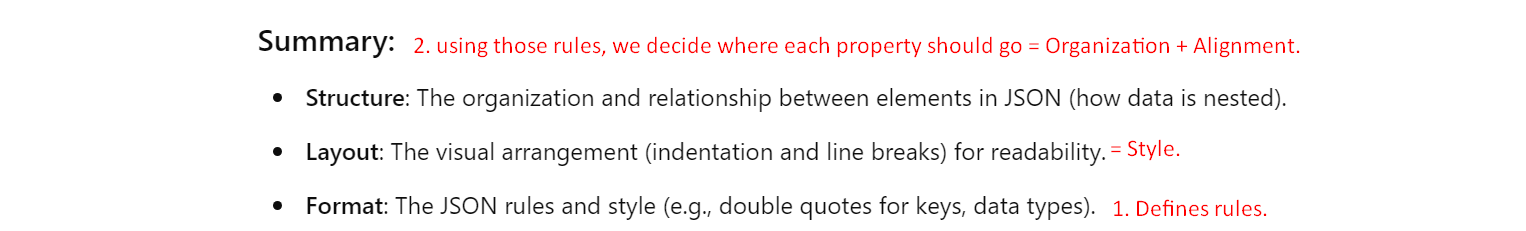
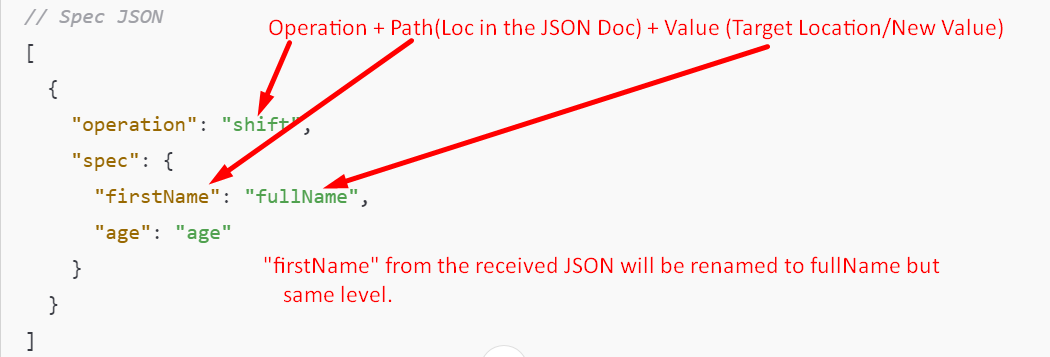
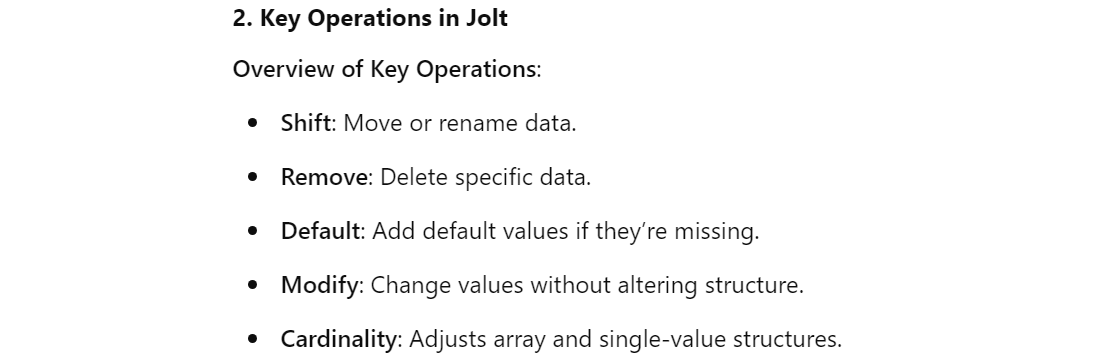
1. JSON transformations refer to the process of restructuring, modifying, or filtering JSON data to meet specific requirements, often used to adapt incoming data to match an expected format for APIs, ETL processes, or database storage.
   1. Transformation: modifying the format, organization, or values in JSON data.
   2. Modifying:
   3. Filtering:
   4. Restructure = Organization + Arrangement.
      1. Example 1: Flattening Nested JSON(**Organization**)  
         Flattened JSON:
      2. Example 2: Grouping JSON Data by a Key: to categorize data by a certain field (e.g., department, category, etc.).  
         Grouped JSON by department:(**Organization**)
      3. Example 3: Sorting JSON Arrays: Sorting JSON data by a field is often necessary for displaying sorted information.(**Arrangement**)  
         Sorted JSON by price (Ascending):
   5. Modifying: Changing key, value, adding computed value.
      1. Changing Values: Changing lowercase to upper case.
      2. Renaming Keys: first\_name to firstName.
      3. Adjusting Data Types: from String to number.
      4. Adding Computed Values:
      5. Adding Missing Fields with Default Values:
   6. Filtering
      1. Removing Unnecessary Fields.
      2. Selecting Specific Nested Data (Same as above)
      3. Filtering Based on Conditions.
2. the structure typically refers to the hierarchical arrangement or organization of data   
   
3. 2. Setting Up for Jolt Transformations
   1. Description:

Jolt is a JSON transformation library for Java that enables you to define transformations using a declarative specification.

1. JOLT Specification = A Document.
   1. JOT Spec = Operations + Paths.
   2. Basic Component.  
      Operation + Path(Loc in the JSON Doc) + Value (Target Location/New Value)
   3. 
2. 

Operations

1. **Shift**:
   1. Moving, **renaming**, **restructuring** JSON Data by specifying the paths in the SPEC.