Dictionary Management Onsite Task

Introduction

Our customers need to manage dictionaries (a key/value mapping or sometimes also called a synonym table). Usually, dictionaries are used to clean or normalise values in dataset columns.

Consider the following example of a small dataset representing products:

Original Dataset:

Product	Color	Price
Apple iPhone 6s	Stonegrey	CHF 769
Samsung Galaxy S8	Midnight Black	CHF 569
Huawei P9	Mystic Silver	CHF 272

E-commerce companies like online stores and market places, however, need the colours more standardised, so that they correspond to colour filters and get properly picked up by the search engine. Each company has its own set of applicable colours and requires product data to be in a certain language.

Our customer ABC would like to have the products above look like this:

Desired Dataset:

Product	Color	Price
Apple iPhone 6s	Dark Grey	CHF 769
Samsung Galaxy S8	Black	CHF 569
Huawei P9	Silver	CHF 272

In order to transform the dataset into the desired format, a dictionary is needed:

Domain	Range
Stonegrey	Dark Grey
Midnight Black	Black
Mystic Silver	Silver

The **_Domain_** of a dictionary represents the original value to transform, the **_Range_** of a dictionary represents the desired value.

If we apply this dictionary to the **Color** column of the **Original Dataset** and replace the values by the corresponding **_Range_** value in the dictionary, we get the **Desired Dataset**.

Dictionary Consistency

A dictionary is said to be consistent if none of the following problems occurs:

- **Duplicates**. Duplicate Domain - Range pairs: Two rows in the dictionary map to the same value, simply resulting in duplicate content.

Domain	Range
Stonegrey	Dark Grey
Stonegrey	Dark Grey
Caribbean Sea	Turquoise

- **Forks**. Duplicate Domains with different Ranges: Two rows in the dictionary map to different values, resulting in an ambiguous transformation.

Domain	Range
Stonegrey	Dark Grey
Stonegrey	Anthracite
Midnight Blue	Dark Blue

- **Cycles.** Two or more rows in a dictionary result in cycles, resulting in a never-ending transformation.

Domain	Range
Stonegrey	Dark Grey
Dark Grey	Stonegrey
Midnight Blue	Dark Blue

- **Chains.** A chain structure in the dictionary (a value in Range column also appears in Domain column of another entry), resulting in inconsistent transformation.

Domain	Range
Stonegrey	Dark Grey
Dark Grey	Anthracite
Midnight Blue	Dark Blue

When using dictionaries for data normalisation or other purposes, all these inconsistencies must not occur, otherwise, the result of the transformation is ambiguous or not defined.

Dictionary Management Application

With this introduction and slightly theoretical background in mind, you are now going to create a dictionary management application. The application must satisfy the following requirements:

Functional Requirements

- Creating and deleting dictionaries
- Showing available dictionaries in an overview
- Editing dictionaries (adding, updating and removing rows)
- Validating the entire dictionary regarding consistency (see above)
    - Validations should be shown as some kind of problem markers next to the offending part of the dictionary.
    - Problem markers have different severities, e.g. Duplicates are less severe than a Cycle (in which case you cannot go on processing such a dictionary).
- Persisting dictionaries with validation errors for future editing

Technical Requirements

- Single page web application
- Angular 4, React/Redux, or any Javascript client-side framework you are familiar with
- LESS, SASS or CSS-in-JS library
- Typescript or ES6
- Build tool and dependency management of your choice
- It can be server-less, e.g. all data is only stored locally, e.g. using HTML 5 local history API or equivalent technologies.