Part 1: Works Single View

A Single View application aggregates and reconciles data from multiple sources to create a single view of an entity, in this case, a musical work.

A musical work consists of the musical notes and lyrics (if any) in a musical composition. A musical work may be fixed in any form, such as a piece of sheet music or a sound recording. It is usually represented by metadata like: title, contributors, roles, duration, etc.

Your task is to provide the code to create the Works Single View from the metadata in `works_metadata.csv` file.

Metadata from some sources may be incomplete, the aim of the Works Single View is to have, for each musical work, the most complete metadata without duplicates.

Instructions

- Scripts should be written in Python.
- To store reconciled works you can use a PostgreSQL or MongoDB database.
- Intructions should be provided on how to execute your code.

`works_metadata.csv` columns:

- `title`: Work title

- `contributors`: Work contributors such as composers, lyricists, etc. we skip the role for simplicity. There can be multiple contributors, they are separated by `|`
- `iswc`: International Standard Musical Work Code, it's a musical work identifier
- `source`: Metadata provider
- `id`: Identifier from the metadata provider

Questions

1. Describe briefly the matching and reconciling method chosen.

2. We constantly receive metadata from our providers, how would you automatize the process?

Part 2: Works Single View API

A customer has sent us a report with ISWC codes. He wants us to enrich it with contributors metadata.

Your task is to create a simple API with the following functionalities:

1. Query the Works Single View by ISWC in order to get the metadata related to that work.
2. Export the Single View metadata in CSV format.
3. Import (upload) files with `works_metadata.csv` format and reconcile the metadata into the Works Single View.

Instructions

- The API has to be written in Python.
- You can use frameworks such as `Flask-Restful` or `Django REST`.
- Response has to be in `JSON` format.
- Intructions should be provided on how to execute your code.

Questions

1. Imagine that the Single View has 20 million musical works, do you think your solution would have a similar response time?
2. If not, what would you do to improve it?