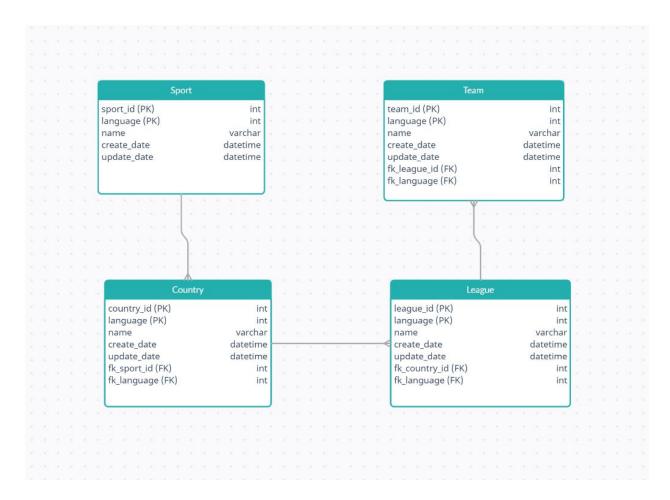
Dictionary Service

Author: Dario Ivanovski

Introduction

Dictionary service is a system which provides saving and reading data for sports, countries, leagues and teams in multiple languages. For starting to load the initial data and save in the database call POST method on localhost:8080/api/manage/import/load. For the configuration and starting the application read the readme.txt file in the project.

The model for the database is given below.



The initial data will take around 3minutes to save because there are around 100k teams for Macedonian and 100k teams for English language. I couldn't find any solution to speed up the process except to add batching with bigger batch size (3000) and using

saveAll() method from the repository instead of saving entities one by one with save() which i realized is slower than the saveAll method. I tried using JMS to send the rows one by one and consume them with multiple consumers and than save them but this it was taking too much time to send all the messages to JMS queue.

API (Management)

For all the endpoints go to http://localhost:8080/swagger-ui.html

Languages Controller

Here you can add new languages or find the ids and names for all the languages available in the database. Initially there are two languages: English and Macedonian but you can just add a new language by calling /api/languages/addLanguage POST with parameter name. The other methods for this controller are:

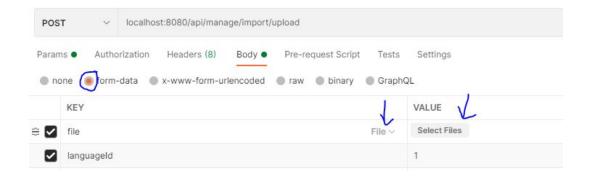
- → /api/languages/addLanguage?name=[name] POST
 Add a new language with name.It returns the generated if for the new language.
- → /api/languages/all GET Findall the languages with their id and name from the database.
- → /api/languages/findId?name=[languageName] GET Find the id of the language with name languageName.
- → /api/languages/findName?id=[languageId] GET Find the name of the language with id languageId

Import Data Controller

→ /api/manage/import/upload?languageId=[id] POST

Upload and parse the csv file.Parameter languageId tells the system in which language the given csv file is.If you want to add data in new language first go to Languages

Controller add the language with name.It will generate the id which you will pass here to the languageId parameter.You can use this method only via Postman.



→ /api/manage/import/load POST

The system loads the initial data from the files for English and Macedonian language.

Sport Management Controller

→ /api/manage/sport/add?languageId=[id] POST

Add a new sport in the db.You should send the data in this format:

```
{
    "sportId": 1,
    "name": "Football"
}
```

→ /api/manage/sport/update?sportId=[sportId]&languageId=[languageId]
Update already existing sport.Send the sport id and language in the query and send the sport name in the request body like this:
{
 "name": "Football Updated"
}

→ /api/manage/sport/delete/{id} Deletes all the sports by given id.

Country Management Controller

→ /api/manage/country/add?languageId=[id] POST

Add a new country in the db.You should send the data in this format:

```
{
  "countryId": 1,
  "name": "Macedonia",
  "fkLanguage": 1,
  "sportId": 1
}
```

→ /api/manage/country/update?countryId=[countryId]&languageId=[languageId]
Update already existing country.Send the country id and language in the query and send
the country name in the request body like this:
{

```
{
   "name": "Country Updated"
}
```

→ /api/manage/country/delete/{id}

Deletes all the countries by given id.

• League Management Controller

→ /api/manage/league/add?languageId=[id] POST

Add a new league in the db.You should send the data in this format:

```
{
  "leagueld": 1,
  "name": "League Name",
  "fkLanguage": 1,
  "countryld": 1
}
```

→ /api/manage/league/update?leagueId=[leagueId]&languageId=[languageId]

Update already existing league.Send the league id and language in the query and send the league name in the request body like this:

```
{
    "name": "League Updated"
}
```

→ /api/manage/league/delete/{id}
Deletes all the leagues by given id.

Team Management Controller

→ /api/manage/team/add?languageId=[id] POST

Add a new team in the db.You should send the data in this format:

```
{
  "teamId": 1,
  "name": "Team Macedonia",
  "fkLanguage": 1,
  "leagueId": 1
}
```

→ /api/manage/team/update?teamId=[teamId]&languageId=[languageId] Update already existing team.Send the team id and language in the query and send the team name in the request body like this: {

```
"name": "League Updated"
}
```

→ /api/manage/team/delete/{id}
Deletes all the teams by given id.

API (Client)

In the client side of the system for listing all the items I am using PagingAndSortingRepository. The maximum page size for all the endpoints is set to 100 to improve the performance and stop the user from sending very large number and crash the system. So if you send some number bigger than 100 it will list only 100 items.

• Sport Client Controller

- → /api/sport/all?pageNumber=[pageNumber]&pageSize=[pageSize] GET

 This will list all the sports pageable.If you want to return the data i specific language send the languageId=[languageId] in the query.All the language ids can be found with Languages Controller.
- → /api/sport/{id}?languageId=[languageId]
 Find the sport with id and language. The parameter languageId is not required, so if you want to find the sport with all languages don't send the parameter.

Country Client Controller

- → /api/country/all?pageNumber=[pageNumber]&pageSize=[pageSize] GET

 This will list all the countries pageable.If you want to return the data i specific language send the languageId=[languageId] in the query.All the language ids can be found with Languages Controller.
- → /api/country/{id}?languageId=[languageId]
 Find the country with id and language. The parameter languageId is not required, so if you want to find the country with all languages don't send the parameter.
- → /api/country/findBySportId/{sportId}
 Finds all the countries in the sport with an id.Also the languageId here is not required.

• League Client Controller

- → /api/league/all?pageNumber=[pageNumber]&pageSize=[pageSize] GET

 This will list all the leagues pageable.If you want to return the data i specific language send the languageId=[languageId] in the query.All the language ids can be found with Languages Controller.
- → /api/league/{id}?languageId=[languageId]

 Find the league with id and language. The parameter languageId is not required, so if you want to find the league with all languages don't send the parameter.
- → /api/league/findByCountryId/{countryId}
 Finds all the leagues in the country with an id.Also the languageId here is not required.

• Team Client Controller

- → /api/team/all?pageNumber=[pageNumber]&pageSize=[pageSize] GET
 This will list all the teams pageable.If you want to return the data i specific language
 send the languageId=[languageId] in the query.All the language ids can be found with
 Languages Controller.
- → /api/team/{id}?languageId=[languageId]
 Find the team with id and language. The parameter languageId is not required, so if you want to find the team with all languages don't send the parameter.
- → /api/team/findByLeagueId/{leagueId}
 Finds all the teams in the league with an id.Also the languageId here is not required.
- → /api/team/plays?languageId=[languageId]
 Find all teams that play sport with id.LanguageId is not required here so you can list the teams with all the languages or for a specific language.