API Challenge

Problem Definition

Our application manages user subscriptions to newsletters about new book releases. Each book belongs to a set of categories and a user can subscribe to any number of those.

Implement an application that provides an HTTP API for manipulating book and subscription data with the following endpoints (**with json bodies**):

Endpoint 1

Should handle category submissions, a category has a unique code, a title and optionally a parent category (e.g. science -> physics).

`POST /categories:`

```
{ “code”: “string”, “title”: “string”, “superCategoryCode”: “string” or null
}
```

Endpoint 2

Should handle book submissions, a book has a title and a list of category codes of categories to which it belongs to.

`POST /books`

```
{ “title”: “string”, “categoryCodes”: [“code1”, “code2”, ...]
}
```

Endpoint 3

Should handle subscriber submissions, a subscriber has an email and a list of category codes of categories for which he/she is interested in.

`POST /subscribers`

```
{ “email”: “string”, “categoryCodes”: [“code1”, “code2”, ...]
}
```

Endpoint 4

Should return a list of “newsletters”, each newsletter relates to a subscriber, a newsletter has a recipient address (the subscriber's email) and a list of notifications, each notification contains the name of a book and a list of categoryPaths for it, a categoryPath is a list that shows the relation of the books' category and the category for which a subscriber is interested.

A subscriber should get notifications for books that belong to the same category or to a child category of a category in which the subscriber is interested.

To make it a bit easier assume that a book can not belong to a parent and child category at the same time, for example if science is the parent category of physics then a book can not belong to both categories, only to one of them, same holds for a subscription. Also assume that the release date of a book is irrelevant and that all books are eligible for a newsletter.

`GET /newsletters?email=[email]`

```
[
{
“recipient”: “string”,
“notifications”: [
{
"book": "string",
"categoryPaths": [
["code1", "code2", "code3"],
["code1", "code2", "code3"]
]
}
]
}
]
```

Example:

Given the following category hierarchy:

science

engineering

software

functional_programming

object_oriented_programming

If a subscriber is interested in engineering and a book e.g. “Programming in Scala” belongs to the category **functional_programming**, then one of the *categoryPaths* for that book in that subscriber's email notification is:

```
["engineering", "software", "functional_programming"]
```

if the same book also belongs to the category **object_oriented_programming** then that same book has also a categoryPath:

```
["engineering", "software", "object_oriented_programming"]
```

and the /newsletters response body should be:

```
[
{
“recipient”: “subscriber@email.com”,
“notifications”: [
{
"book": "Programming in Scala",
"categoryPaths": [
["engineering", "software", "functional_programming"],
["engineering", "software", "object_oriented_programming"]
]
}
]
}
]
```

Project Setup

1. Configure your enviroment setup according to Java Environment Setup
2. Clone the repository:

   ```
   git clone https://[user]@bitbucket.endava.com/scm/bd/java-rampup.git
   ```

3. Create a new branch for your challenge:

   ```
   git checkout -b [endavauser]-javaapichallenge
   ```

4. Import the **api-challenge** gradle project as is described in IntelliJ Import Project.
5. Push your changes when you finish:

   ```
   git push origin -b [endavauser]-javaapichallenge
   ```