

PERSADO BACKEND ASSIGNMENT – CRUD

<https://github.com/yannispapadopoulos/CRUD-API>

Yiannis Papadopoulos

The application was developed using SpringBoot,Java,Microsoft SQL Server and the IDE used was IntelliJ. All the requirements have been fulfilled.

Database Description:

There three main entities.

-**Book**(isbn –primary key,title,descry,visible,author_id,publisher_id,date_creation)

-**Author**(id,firstname,lastname,email,birthdate)

-**Publisher**(id,name,telephone,address)

Book's author_id is foreign key to **Authors** id

and **Book's** publisher_id is foreign key to **Publisher** id

Endpoints:

The functions add,update return the book,author,publisher entities respectively instead of void,
for the sake of easier testing.

- 1) <http://localhost:9191/bookstore/findOrderPublishedBooksByAuthor> @GET

Returns visible books having a publishing company ordered by author lastname.
Books of same author must be ordered in descending direction by using a numeric position attribute.

- 2) <http://localhost:9191/bookstore/singleBookInfo> @GET

returns book displaying specific attributes(title,description,isbn,creation date,author full name, author email address,author birthdate,publisher name,publisher address) .

- 3) <http://localhost:9191/bookstore/addAuthor> @POST

adds author record.Sequence gives new id ,so we don't need to insert id.

requires json obj of author

eg.

```
{
    "id": 2,
    "firstname": "george",
    "lastname": "konstantaras",
    "email": "George@yahoo.com",
    "birth_date": "1994-03-02"
}
```

4) <http://localhost:9191/bookstore/addPublisher> @POST

adds new publisher record. Sequence gives new id ,so we don't need to insert id.

requires json obj of publisher

eg.

```
{
    "id": 1,
    "name": "ALEX",
    "telephone": "543054654",
    "address": "MAROUSI"
}
```

5) <http://localhost:9191/bookstore/addBookStrict> @POST

Stores new book into database, but author,publisher cannot be null.

There was no clarification in the description of the assignment about the procedure of 'add book' , in relation to author /publisher

(for example,when a new book gets inserted ,I was not sure if I had to create author,publisher ,in case they don't exist.At this point, that publisher and author should be mandatory.the user gives their id ,so that book entity connects with the publish and author respective entity ,through foreign key.

requires json obj of book

eg.

```
{
    "title": "Alice in Wonderland",
    "descr": "trips",
    "authorId": 1,
    "publisherId": 2,

    "visible": 1,
    "date_creation": "2020-22-02"
}
```

However, there is a second optional call :

<http://localhost:9191/bookstore/addBookOptional> @POST

Stores new book into database.

I have set ISBN as a primary key with unique value (no extra id for the book table).

isbn and title fields are mandatory in the Jason object .

Adding a new book with this api call, can store a book without author and publisher (author/publisher = null as below **eg1**) or we have to give a whole author and publisher object respectively ,in order to add new values to them(**eg2**)

eg1:

```
{
  "isbn": 1234,
  "title": "Alice in Wonderland",
  "descr": "trips",
  "author": null,
  "publisher": null,
  "visible": 1,
  "date_creation": "2020-11-11"
}
```

eg2:

```
{
  "isbn": 1234,
  "title": "Alice in Wonderland",
  "descr": "trips",
  "author": null,
  "author": {
    "id": 2,
    "firstname": "george",
    "lastname": "k",
    "email": null,
    "birth_date": "2008-11-10"
  },
  "publisher": {
    "id": 1,
    "name": "ALEX",
    "telephone": "543054654",
    "address": "MAROUSHI"
  },
  "publisher": null,
  "visible": 1,
  "date_creation": "2020-11-11"
}
```

6) <http://localhost:9191/bookstore/updateBook> @POST

Searches the given book record and updates book data.

requires json obj of book

eg.

```
{
  "isbn": 1,

  "title": "sacred geometry",
  "descr": "Fibonacci in the nature",
  "author": {
    "id": 2,
    "firstname": "george",
    "lastname": "k",
    "email": null,
    "birth_date": "2008-11-10"
  },
  "publisher": {
    "id": 1,
    "name": "ALEX",
    "telephone": "543054654",
    "address": "MAROUSTI"
  },
  "visible": 1,
  "date_creation": "2008-11-10"
}
```

7) <http://localhost:9191/bookstore/deleteBook/isbn> @POST

deletes book record with given isbn

Finally ,there is an exception package for some basic validation functions and a DTO unit with objects used for book insertion and endpoints 1-2.