PERSADO BACKEND ASSIGNMENT – CRUD
Yiannis Papadopoulos

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

 Below are the http verbs that have been implemented:

///////////////////////////BOOK///

- **http://localhost:9191/bookstore/findOrderPublishedBooksByAuthor @GET**

  returns visible books having a publishing company ordered by author lastname

- **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) .

- **http://localhost:9191/bookstore/addBook @POST**
  Stores new book into database.
  I consider ISBN as a primary key with unique value (no extra id for the book table).
  A new value is auto generated everytime that we add a new book.
  Moreover,adding a new book with this api call, does not need author,publisher to be not null ,as the relation between them was not clear.

requires json object of book
eg.
```
{
        "title": "Alice in Wonderland",
        "descr": "trips",
        "author": null,
        "publisher": null,
        "visible": 1,
        "date_creation": "2020-11-11"
  }
```

- **http://localhost:9191/bookstore/addBookWithAuthPubl**
  Stores new book into database, as previous call(addBook),
  but now author,publisher must get inserted as well.

  requires jason obj of book

  eg.

```json
{
    "title": "Alice in Wonderland",
    "descr": "trips",
    "author": {
        "id": 2,
        "firstname": "george",
        "lastname": "konstantaras",
        "email": null,
        "birth_date": "2008-11-10"
    },
    "publisher": {
        "id": 1,
        "name": "ALEX",
        "telephone": "543054654",
        "address": "MAROUSI"
    },
    "visible": 1,
    "date_creation":"2020-22-02"
}
```

- **http://localhost:9191/bookstore/updateBook @POST**
  Searches the given isbn book record and updates book data.

  requires jason obj of book

  eg.
```json
{
    "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": "MAROUSI"
        },
        "visible": 1,
        "date_creation": "2008-11-10"
    }
```

- **http://localhost:9191/bookstore/deleteBook/isbn @POST**

  deletes book record with given **isbn**

/////////////////////////AUTHOR///

- **http://localhost:9191/bookstore/addAuthor @POST**

  adds author record

  requires jason obj of author

  eg.

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

/////////////////////////PUBLISHER///

- **http://localhost:9191/bookstore/addPublisher @POST**

  adds new publisher record

  requires jason obj of publisher

  eg.

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