# Library Web Service

## Part 1

### Classes

```
Author (idA,
                firstName,
                lastName,
                yearOfBirth,
                nationality,
           +authorImage)

Book (idBook,
          title,
          publicationYear,
          language,
          nbPages,
          idCategory,
          idAuthor,
         +rating
         +price
         +bookImage)

Category (idC,
                  categoryName,
                  parentCategory)

User (idUser,
         name,
         username,
         password,
         privilege,
         userImage)
```

### Endpoints

- List books `/books`
    - Paginated results
        - `/books?page=2`
    - The *page size* can also be customized, the default page size is set to 10
        - `/books?page=2&size=9`
    - Results can be filtered by publication *year*, *language*, and *category*
        - `/books?year=*&lang=*`
    - Fetching books of a selected category can be *recursive* or *non-recursive*
- Books of a selected *author* `/authors?name=*`
- Search using a keyword:
    - The keyword is looked up in the names of authors, books, and categories

        - `/search?q=Math`
    - The search results are paginated and can be filtered by type [Book/Author/Category]
        - `/search?q=Math&page=2&type=Book`
    - The desired page number is sent in the request.
    - The *page size* can be customized in the request, the default page size is set to 10.
        - `/search?q=Math&page=2&size=9&type=Book`
- Login `/login`
- Rate a book `/books/{BOOK}/rate`

```
/books
/books?page=2
/books?page=2&size=9
/books?year=2019&lang=fr
/books?category=Fiction&recursive=true
/books/{BOOK}
/books/{BOOK}/rate

/authors?name=Jack Donovan
/authors/{AUTHOR}

/search?q=Math
/search?q=Math&page=2&type=Book
/search?q=Math&page=2&size=9&type=Book
```

---

## Part 2

- Use Authenticated requests ( `JWT` and Basic Access Auth) in `Header` can:
    - `POST` *Add a Book*
    - `POST` Add an Author
    - `POST` Add a Category
    - `DELETE` Delete a Book
    - `DELETE` *Delete an Author's books*
    - `DELETE` Delete a Category
    - `PUT` Modify a Book
    - `PUT` Modify an Author
    - `PUT` Modify a Category
- Authentication token expire after `10` minutes

---

## Part 3

- Python/Web/Java? client that exploits the API
- Documentation

## Tasks

- [ ] Build the API
    - [ ] REST
    - [ ] GraphQL
- [ ] Make the *dataset* (Authors `min:50`, Books `min:150`, Category `min:10`)
- [ ] Client App
    - [ ] Laravel
    - [ ] Java
        - [ ] Desktop
        - [ ] Phone `Kotlin`
- [ ] Secure against *OWASP 10* and *API 10*
- [ ] Documentation