

1 Practice 13.2. Forms with Symfony

The mark of this practice **can increase the mark of the second evaluation by 0.5 points**, provided that the marks of the practice part and the exam are at least 5.

It's recommended to do this practice after the practice 13.1.

Before start, create a new git branch named `library-u13.2` and switch to it:

```
git branch library-u13.2
git checkout library-u13.2
```

Do all the practice in the `library-u13.2` branch. Once finished, do the submission via GitHub Classroom as detailed at the end of the practice.

1.1 Overview

In this practice we're going to do the forms to view and edit the data of our library app. The required forms are:

- A form for editing, deleting and adding books.
- A form for editing, deleting and adding publishers.
- A form for searching books by title and author.
- A form for searching publishers by name.

Use the styles you want.

Add to each page a navigation bar or links to return to the main pages. All the pages must be accessed by buttons or links.

At the end of the practice, remove or comment the routes and code unused.

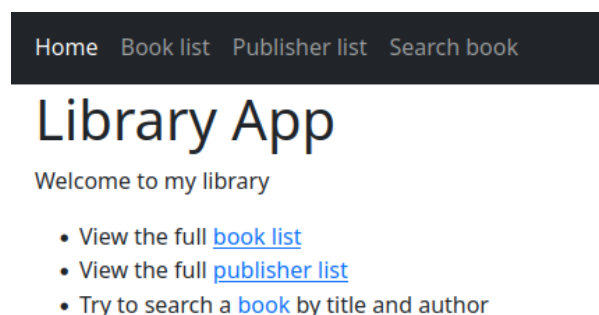


Figure 1: Main page

1.2 Exercise 1. Book form

Create a form for editing, deleting and adding a new book.

The form can be included in the *single book* view and in the *new book* view.

The publisher must be selected by a `select` input or similar. Remember that the publisher can be empty (`null`).

Choose an appropriate range of years for the publication date, i.e. from the year 1000 to the current year.

The validation must be done in the server (disable the client-side validation) and must have the next restrictions:

- ISBN and title required
- Pages, a numeric value equal or greater than zero.

Home Book list Publisher list Search book

Library App

Book:

Isbn

Title

Author

Pages

Pub date

Jul

7

2022

Publisher

Submit

Delete book

Figure 2: Book form

1.3 Exercise 2. Publisher form

Create a form for editing, deleting and adding a new publisher.

The form can be included in the *single publisher* view and in the *new publisher* view.

The validation must be done in the server (disable the client-side validation) and must have the next restrictions:

- Name required
- Email, not required and email format.

Home Book list Publisher list Search book

Library App

Publisher:

Name

Email

Submit

Delete publisher

Figure 3: Publisher form

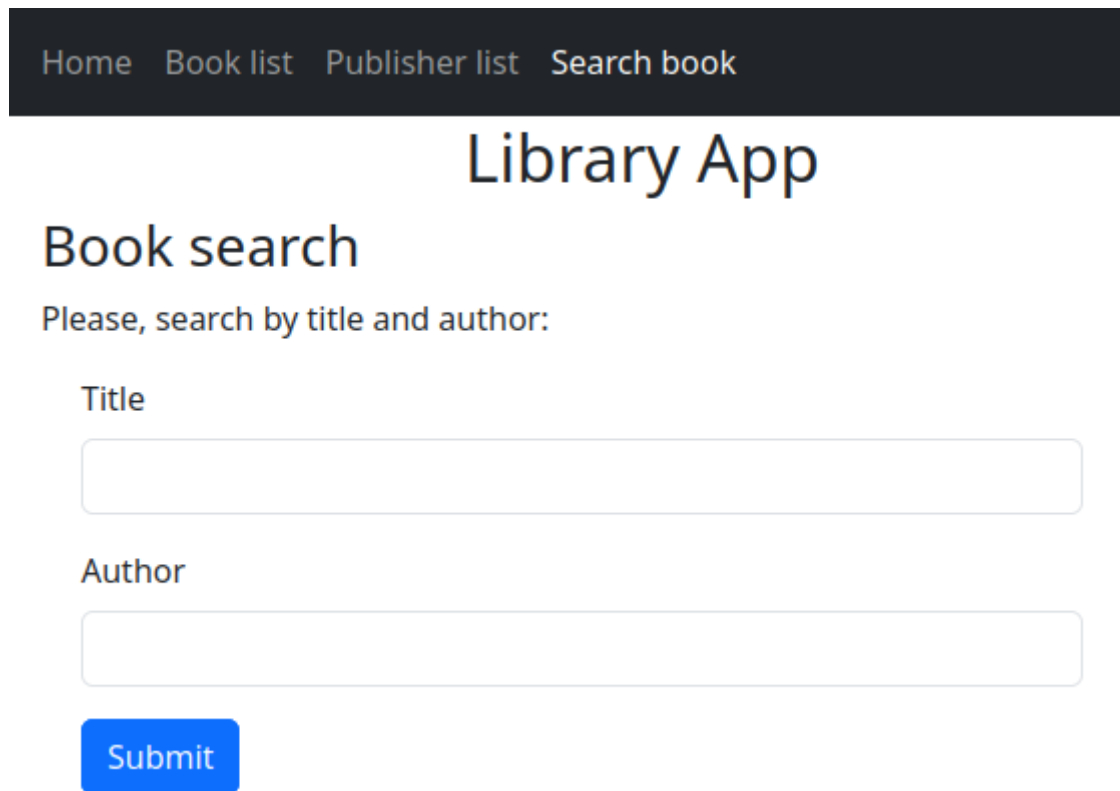
1.4 Exercise 3. Search books form

Create a form for searching a book by title and author.

Make a new view for the form. You can reuse the `/book/search/{title}/{author}` renaming it to `/book/search`.

The fields must accept empty strings.

The result should be shown in the book `list` view, as in the previous practice.



The screenshot shows a web application interface for a library. At the top, there is a dark navigation bar with links: 'Home', 'Book list', 'Publisher list', and 'Search book'. Below the navigation bar, the main heading 'Library App' is displayed in a large, blue, serif font. Underneath, the section 'Book search' is written in a smaller, blue, serif font. A prompt 'Please, search by title and author:' is shown in a blue, serif font. There are two input fields: the first is labeled 'Title' and the second is labeled 'Author', both in a blue, serif font. Below the 'Author' field is a blue 'Submit' button with white text.

Figure 4: Search books form

1.5 Exercise 4. Search publishers form

Create a form for searching a publisher by its name.

Embed the form in the publisher list view instead of creating a new view.

The field must accept empty strings.

The result should be shown in the publisher `list` view, as in the previous practice.

Home Book list Publisher list Search book

Library App

Publisher list

Add new publisher

Search for a publisher:

Submit

	ID	Name	Email
Open publisher	1	Clarion Booksss	info@clarion.com
Open publisher	2	Ecco	ecco_info@ecco.com
Open publisher	3	Scribner	scribner@scr.coma
Open publisher	5	Alfaguara	alfa@guara.com

Figure 5: Publisher list with the publisher search form

1.6 How to submit to GitHub Classroom

Once you finish the task, make a commit with the comment “PRACTICE 13.2 SUBMISSION COMMIT” and push it to GitHub. Make a **pull-request** so that I could know your code is submitted.

1. This task must be submitted to the same repository of the previous one. Remember to make a new branch before starting the practice.
2. Once you finish the task, make a commit with the comment “PRACTICE 13.2 SUBMISSION COMMIT”, merge the branch into the main branch, and push it to GitHub. For example:

```
git commit -m "PRACTICE 13.2 SUBMISSION COMMIT"
git checkout main
git merge library-u13.2
git push
```

3. It's recommended to tag your commit with the tag “Practice_13.2”.
4. Before that, you can do the commits and push you want. If you change your code after your submission commit, make another commit and push with the same text in the message adding the corrections you've done.

If you have any doubt in your task, you can push your code and ask me by email what's your problem. It will make it easier for both the solutions of code issues.