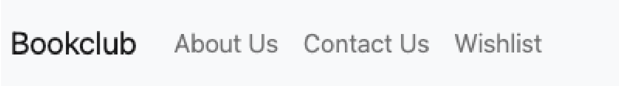
Instructions for Assignment 4.2

Forms and Data Validation

Instructions

1. Update the pom.xml file by adding an entry for groupId “org.springframework.boot” and artifactid “spring-boot.starter-validation.” Make sure you reload the project and/or maven dependencies.
2. Update the header.html page’s links to include a route for “Wishlist.”

   Figure 2.

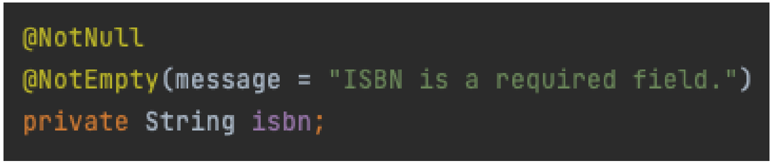
   

3. Add a new class to the model package and name it WishlistItem

   Additional requirements

   a. Give the class two private properties for isbn and title.
   b. Add a @NotNull and @NotEmpty decorator to each property with a message of “<PropertyName is a required field.”>.

      Figure 3.b.

      ```
      @NotNull
      @NotEmpty(message = "ISBN is a required field.")
      private String isbn;
      ```

   c. Give the class two constructors, a default and one with two parameters for isbn and title.
   d. Provide the class with getter/setter methods.
   e. Override the classes toString method and return string formatted as “WishlistItem{isbn=<isbnValue>, title=<titleValue>}

4. Add a new interface to dao package and name it WishlistDao (notice the spelling?) and extend the GenericDao interface with the WishlistItem object.
5. Add a new class to the impl package and name it MemWishlistDao (notice the spelling?) and write methods to implement list() and find(). In the classes constructor populate a private property called wishlist of type List<WishlistItem>. Return this array in the list() method and use this array to return a matching object by isbn. Use the work you completed in the MemBookDao as a baseline for how to write the Java code.
6. Add a new class to the web package and name it WishlistController

Additional requirements

a. Give the class a RequestMapping of "/wishlist"
b. Add three new methods: showWishlist(Model model), wishlistForm(Model model), and addWishlistItem(@Valid WishlistItem wishlistItem, BindingResult bindingResult).

Additional requirements showWishlsit(Model model)

1) Add a decorator of RequestMapping with a method of GET.
2) Create a new instance of the MemWishlistDao, bind the results from the list() call to a variable named wishlist of type List&lt;WishlistItem&gt;, and add a model attribute with a key of wishlist and the variable wishlist. Return an HTML page located at wishlist/list (we will create this page in a later step).

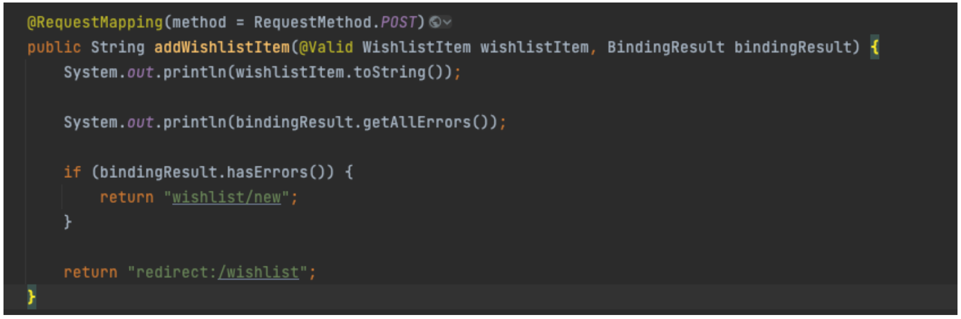
Additional requirements wishlistForm(Model model)

1) Add a decorator of RequestMapping with a method of GET and a path of "/new"
2) Add a model attribute with a key of wishlistItem and a value of a new WishlistItem object. Return an HTML page located at wishlist/new (we will create this page in later step).

Additional requirements addWishlistItem(@Valid WishlistItem wishlistItem, BindingResults bindingResults)

1) Add a decorator of RequestMapping with a method of POST.
2) Add an if statement to check if the bindingResults.hasErrors() and return the HTML page wishlist/new for true evaluations. Otherwise, return "redirect:/wishlsit"

Figure 5.b.2.2.2.



```
@RequestMapping(method = RequestMethod.POST)
public String addWishlistItem(@Valid WishlistItem wishlistItem, BindingResult bindingResult) {
    System.out.println(wishlistItem.toString());

    System.out.println(bindingResult.getAllErrors());

    if (bindingResult.hasErrors()) {
        return "wishlist/new";
    }

    return "redirect:/wishlist";
}
```

Additional comments

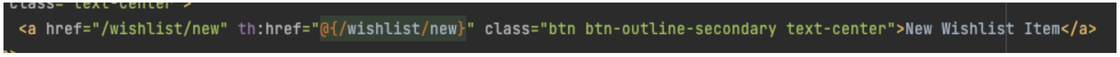
Pay close attention to what's happening here. We are receiving an object called wishlistItem and an object called bindingResults. The wishlistItem is the return data from the Thymeleaf form and the bindingResutls object is the results of the form submission. If there are errors in the form, the bindingResults.hasErrors() call will evaluate to true and we will return the New.html page with the errors that were generated.

7. Add a new folder to the templates directory and name it wishlist.
8. Add two new HTML files to the wishlist folder: list.html and new.html.

Additional requirements list.html

a. Add an anchor tag with a route to /wishlsit/new and a value of “New Wishlist Item.”

Figure 7.a.



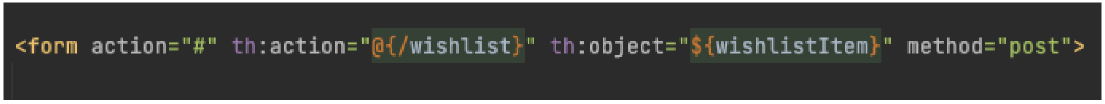
```
class="text-center">
<a href="/wishlist/new" th:href="@{/wishlist/new}" class="btn btn-outline-secondary text-center">New Wishlist Item</a>
```

b. Using a Bootstrap table and the th:each directive, populate an HTML table with the results from the modelAttribute “wishlist.”

Additional requirements new.html

a. Using Bootstrap, create a new HTML form with a signature of th:action=”@{/wishlist}”, a th:object of ${wishlistItem}, and a method of post.

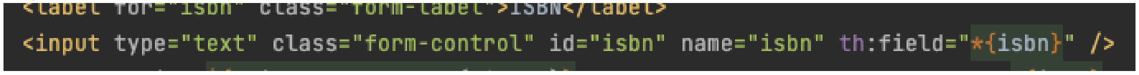
Figure 7.a.a.



```
<form action="#" th:action="@{/wishlist}" th:object="${wishlistItem}" method="post">
```

b. For the forms fields, use the th:field directive to bind the model’s fields.

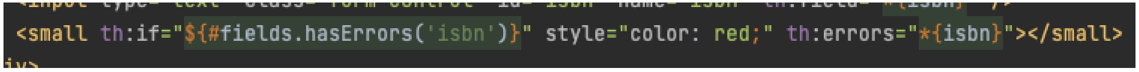
Figure 7.b.b.



```
<label for="isbn" class="form-label">ISBN</label>
<input type="text" class="form-control" id="isbn" name="isbn" th:field="*{isbn}" />
```

c. Under the input fields, use the th:errors directive to bind the model’s bindingResults error messages.

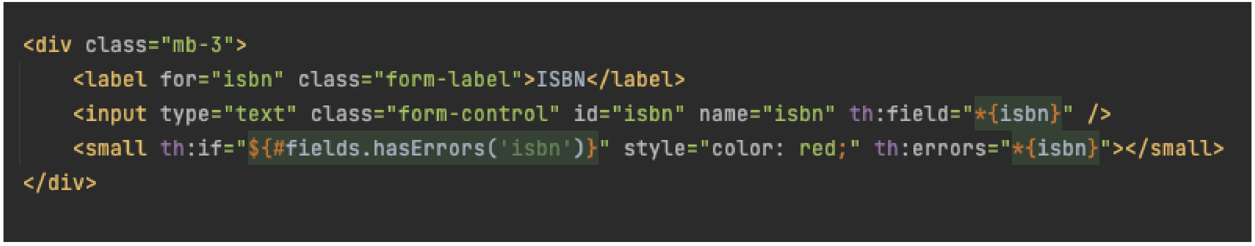
Figure 7.c.



```
<input type="text" class="form-control" id="isbn" name="isbn" th:field="*{isbn}" />
<small th:if="${#fields.hasErrors('isbn')}" style="color: red;" th:errors="*{isbn}"></small>
iv>
```

d. Underneath the form add an HTML anchor tag with a route of /wishlist and a value of “Return to wishlist.”

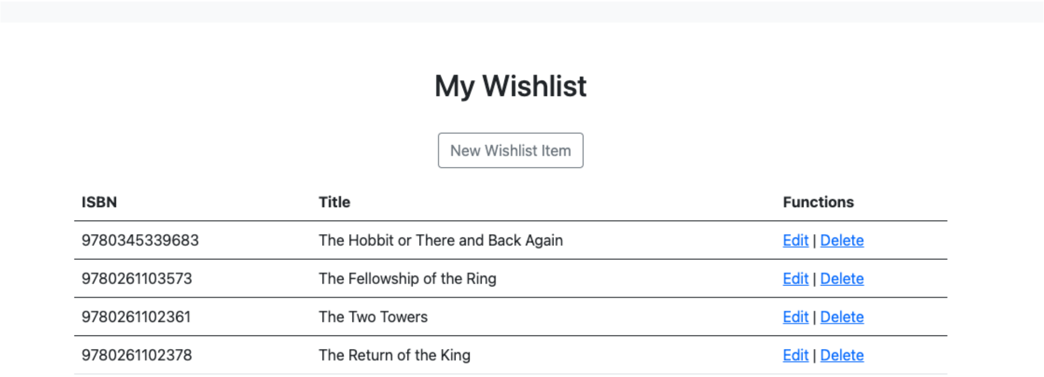
Helpful code HTML code snippet



```
<div class="mb-3">
    <label for="isbn" class="form-label">ISBN</label>
    <input type="text" class="form-control" id="isbn" name="isbn" th:field="*{isbn}" />
    <small th:if="${#fields.hasErrors('isbn')}" style="color: red;" th:errors="*{isbn}"></small>
</div>
```

9. Run and test the application. You should see a list of wish list items when you select the wishlist link from the navbar in the header.html page. Additionally, you should be able to select a "New Wishlist Item" and be taken to the New Wishlist Form, and submitting a form should provide data validation for empty fields with the error messages we created in the model.

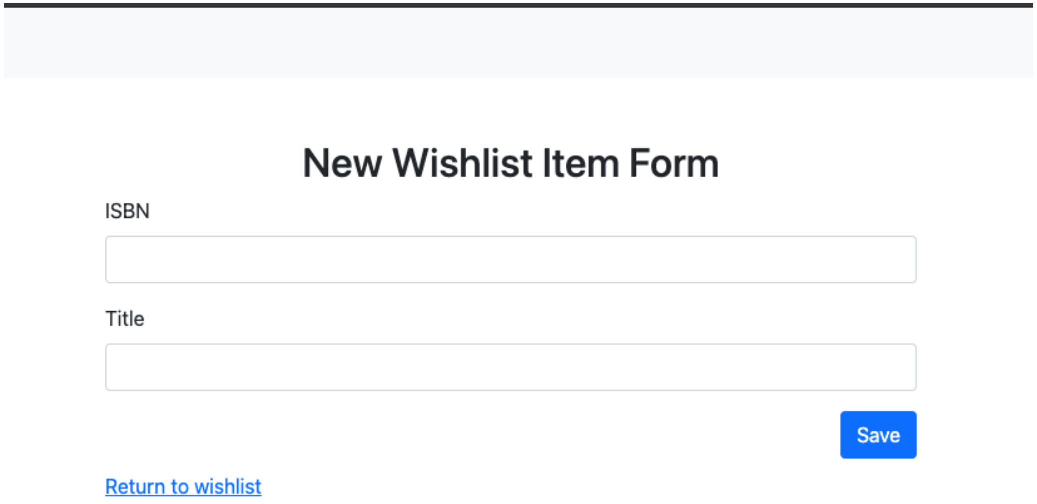
Figure 8



My Wishlist

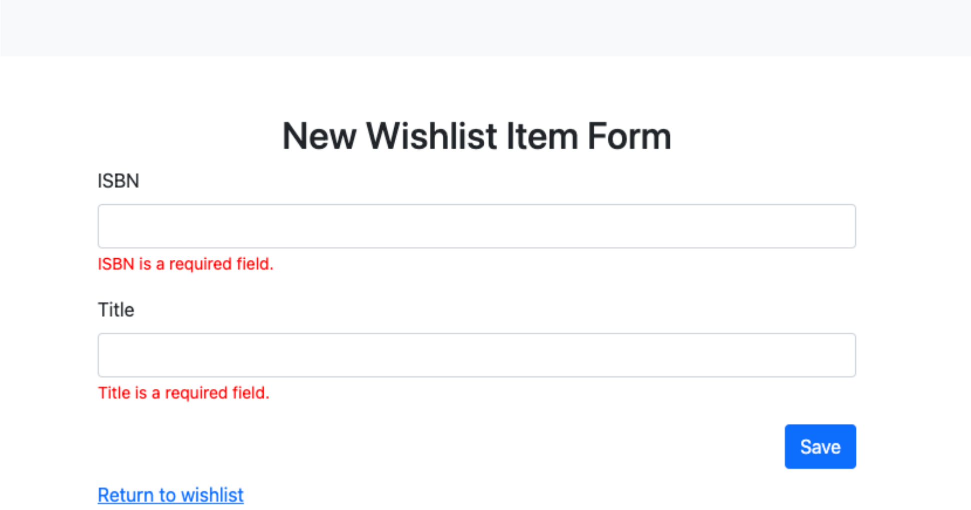
ISBN	Title	Functions
9780345339683	The Hobbit or There and Back Again	Edit | Delete
9780261103573	The Fellowship of the Ring	Edit | Delete
9780261102361	The Two Towers	Edit | Delete
9780261102378	The Return of the King	Edit | Delete

Figure 8.1



New Wishlist Item Form

Figure 8.2



New Wishlist Item Form

ISBN

ISBN is a required field.

Title

Title is a required field.

Special note. For the red font, I set the HTML elements style to “color: red;” If you run into issues, you may visit the courses GitHub repository and compare the code there against what you have. However, please be advised, to get the most out of this course, it benefits you to try to complete the assignments on your own before falling back to the solution.