

Bug Fixes (Final Project)

Sunday, June 7, 2020 8:50 PM

1. The **NewBook** model is not working

- **Problem**
 - unknown
 - it was not reading authorId field
 - always coming as null
- **Solution** Replaced with a new class **SimpleBook**
- **Code changes**
 - Addition
 - ViewModels/SimpleBook.cs
 - Modification
 - Controllers/BookController.cs Line

```
[HttpPost]
[Authorize(Roles = "Admin")]
public async Task<ActionResult> AddBook([FromBody] SimpleBook vm) //replaced NewBook vm
{
    ...
}
```

2. The FlatFileBookRepository.GetById() was throwing EntityNotFoundException

- **Problem**
 - The entire application was designed to expect null return
 - The exception required a large refactoring
 -
- **Solution**
 - Replaced exception with return null
 - Project: ConceptArchitect.BookManagement.FlatFileRepository
 - File: FlatFileBookRepository.cs Line 66-67
- **Code changes**
 - Addition
 - ViewModels/SimpleBook.cs
 - Modification
 - Controllers/BookController.cs Line

```
public Book GetById(string id)
{
    id = id.ToLower();
    if (store.books.ContainsKey(id))
        return store.books[id];
    else
        return null;
    //throw new EntityNotFoundException(Entity: typeof(Book), Id: id);
}
```

3. SignalR force closing on update

- **Problem**
 - Book Object has cyclic reference to author
 - Looks like signalR has marshalling issue with cyclic references.
- **Solution**
 - Send the **SimpleBook object** instead of Book over to signalR
 - BooksController.cs **AddBook()**
- **Code changes**
 - Addition
 - ViewModels/SimpleBook.cs
 - Modification
 - Controllers/BookController.cs Line

```
[HttpPost]
[Authorize(Roles = "Admin")]
public async Task<ActionResult> AddBook([FromBody] SimpleBook vm)
{
    if (ModelState.IsValid)
    {
        var book = new Book()
        {
            Id = vm.Id,
            Title = vm.Title,
            Price = vm.Price,
            Description = vm.Description,
            CoverPage = vm.CoverPage,
            Tags = vm.Tags,
            Author = authorManager.GetAuthorById(vm.Author)
        };

        bookManager.AddBook(book);
        vm.Id = book.Id;
        //await bookHub.Clients.All.SendAsync("BookAdded", book);
        await bookHub.Clients.All.SendAsync("BookAdded", vm);
        return Created($"api/books/{book.Id}", vm);
    }
    else
    {
        return BadRequest(ModelState);
    }
}
```