

This application is an online book store, which allows users to buy books with money deposited on their accounts.

You have given a boilerplate of the project with 4 empty models in module `tasks`. Please use Django REST Framework in your solution. If you write your tests with pytest you'll get an extra point.

Your task consists of 3 parts:

1. Implement necessary fields and connections between models in file `tasks/models.py`.
2. Implement an API endpoint `/books/` which lists all available books (id, title and price).
3. Implement an API endpoint `/books/buy/` which accepts an object `{"books": [ids...]}` and allows user to buy available books using previously deposited funds. Successful response should include array with ids of books and total cost.

Specific requirements:

- Everyone can view books, but only authenticated users can buy them.
- Total cost of purchase should be deducted from user's account. If amount exceeds total balance, then endpoint should return an error message.
- You should store information about every purchase in a model `Purchase`.
- Try to implement code responsible for spending user's balance in a reusable fashion.
- User's balance and prices are expressed in a fiat currency, thus they can have fractional part.
- Include at least basic tests of your code. You may need to write a simple mock of deposit functionality.
- Cover as many edge cases as you can find.
- Explain in comments any assumptions and decisions you made.
- Send us a bundled repository with all commits [Git - git-bundle Documentation](#)