

System Requirements

Specification Index

For

Managing Books in a Library System

(Topic:- Django Views)

Version 1.0

Scenario:

You have joined a Library Management System as a Django developer. The library stores thousands of books, and users frequently request lists of available books. The system needs an efficient way to fetch and display all books stored in the database.

Currently, the library maintains a database of books with attributes such as:

title → The name of the book (CharField, max length: 200).

author → The author of the book (CharField, max length: 100).

published_date → The date the book was published (DateField).

isbn → A unique identifier for the book (CharField, max length: 13, unique=True).

available_copies → The number of copies available in the library (IntegerField, default=1).

Problem Statement

Your task is to implement a Function-Based View (FBV) that returns a list of all books stored in the database.

Your Task

- Implement a Django Function-Based View (FBV) named `book_list_view` to fetch all books.
- Return JSON-formatted book details when the API is called.
- Ensure efficient querying and proper response structure.
- Handle errors gracefully, ensuring that the API doesn't break in case of an empty database.

Execution Steps to Follow:

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to

Application menu(Three horizontal lines at left top)->Terminal->NewTerminal.

3. The editor Auto Saves the code.
4. If you want to exit (logout) and to continue the coding later anytime(using Save & Exit option on Assessment LandingPage) then you need to use CTRL+Shift+B command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while

logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.

6. To test any Restful application, the last option on the left panel of IDE, you can find

ThunderClient, which is the lightweight equivalent of POSTMAN.

7. To test any UI based application the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.

8. Install 'djangoestframework' module before running the code. For this use the following command.
`pip install djangoestframework`
9. Use the following command to run the server
`python3 manage.py runserver`
10. Mandatory: Before final submission run the following commands to execute testcases
`python3 manage.py test library.test.test_functional`
`python3 manage.py test library.test.test_exceptional`
`python3 manage.py test library.test.test_boundary`
11. To test rest end points
Click on 'Thunder Client' or use Ctrl+Shift+R->Click on 'New Request' (at left side of IDE)
12. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.
13. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.