
System Requirements Specification Index

For

**Django Rest-API
Notes-App**

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1 Project Abstract.....3

2 Rest Endpoints..... 3

3 Template Code Structure..... 4

4 Execution Steps to Follow..... 4

NOTES APPLICATION

System Requirements Specification

1 PROJECT ABSTRACT

NoteApp is Django Rest API application with SQLite Database, where it allows any unregistered users (visitors) to manage the notes like create, view, modify, delete and search activities.

Visitors can perform the following actions:

1. Allows to add a note.
2. Allows to display specific note by id.
3. Allows to display all the notes.
4. Allows to update the existing note (partial and full update).
5. Allows to delete an existing note.

2 REST ENDPOINTS

Rest End-points to be exposed in the controller along with method details for the same to be created.

Class Name	Method Name	Purpose Of Method
NotesCRUDView	get(self,request,pk=None,format=None)	To display specific note and to display all the notes.
	post(self, request,format=None)	To add a new note.
	patch(self,request,pk,format=None)	To update the existing note (partial and full update).
	delete(self,request,pk,format=None)	To delete an existing note.

Resources (Models)

Class	Description	Status
NotesModel	<ul style="list-style-type: none">o A model class for Notes.o It will map to the Notes Model table.	Already implemented.

Resources (Serializers)

Class	Description	Status
NotesSerializer	A serializer for NotesModel	Already implemented

Resources (Views)

Class	Description	Status
NotesCRUDView	A class for the get, post, patch, delete functionalities on NotesModelmodel.	To be implemented

4 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 -> New Terminal.
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 Landing Page) 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 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 'djangorestframework' module before running the code. For this use the following command.
`pip install djangorestframework`
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 notesapp.test.test_functional`
`python3 manage.py test notesapp.test.test_exceptional`
`python3 manage.py test notesapp.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.