
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	Assumptions, Dependencies, Risks / Constraints.....	4
3	Business Validations.....	4
4	Rest Endpoints.....	5
5	Template Code Structure.....	6
6	Execution Steps to Follow.....	8

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 delete an existing note
3. Allows to update the existing note (partially and fully)
4. Allows to display specific note by id
5. Allows to display all the notes
6. Allows to search the note based on the id
7. Allows to search the notes based on Author
8. Allows to search the notes based on Status

- While fetching the note by ID, if note id does not exist then operation should throw custom exception.
- While deleting the note by ID, if note id does not exist then operation should throw custom exception.
- While updating the status of note, if note id does not exist then operation should throw custom exception.
- While searching the note by ID, if note id does not exist then operation should throw custom exception.
- While searching the note by status, if note status does not exist then operation should throw custom exception.
- While searching the note by author, if note author does not exist then operation should throw custom exception.

3 BUSINESS VALIDATIONS

1. Note description is max 200 characters.
2. Note status values should be either “Completed” or “Pending”.
3. Note title is max 20 characters.
4. Note author is max 20 characters.

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
SearchNotesByIdView	<code>get_queryset(self)</code>	To search the note based on the id.
SearchNotesByAuthorView	<code>get_queryset(self)</code>	To search the note based on the author.
SearchNotesByStatusView	<code>get_queryset(self)</code>	To search the note based on the status.
NotesCRUDView	<code>get(self,request,pk=None,format=None)</code>	To display specific note and all the notes.
	<code>post(self, request,format=None)</code>	To add a note.
	<code>put(self,request,pk,format=None)</code>	To full update the existing note.
	<code>patch(self,request,pk,format=None)</code>	To partial update the existing note.
	<code>delete(self,request,pk,format=None)</code>	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, put, patch, delete functionalities on NotesModel model.	To be implemented
SearchNotesByIdView	A class for searching note by id functionality.	To be implemented
SearchNotesByAuthorView	A class for searching note by author functionality.	To be implemented
SearchNotesByStatusView	A class for searching note by status functionality.	To be implemented

Resources (Services)

Class	Description	Status
NotesService	A service class for searching note by id, by author and by status functionalities.	To be implemented

Resources (Exceptions)

Class	Description	Status
IdNotAvailable	Custom exception IdNotAvailable is raised in case specified id is not available.	Already implemented.
AuthorNotAvailable	Custom exception AuthorNotAvailable is raised in case specified author is not available.	Already implemented.
StatusNotAvailable	Custom exception StatusNotAvailable is raised in case specified status is not available.	Already implemented.

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 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 'django rest framework' module before running the code. For this use the following command.

```
pip install djangorestframework
```

9. Use the following command to run the server

```
python manage.py runserver
```

10. Mandatory: Before final submission run the following commands to execute testcases

```
python manage.py test notesapp.test.test_functional
```

```
python manage.py test notesapp.test.test_exceptional
```

```
python 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.