API specification documentation
Section 2: Rest Api

Project Setup Local:-
I am using django rest framework for development. You need to install django in your mechine.

1. python3 -m pip3 install Django
2. python3 -m django --version
3. goto project directory
4. pip3 install requirement.txt
5. python3 manage.py makemigration
6. python3 manage.py migrate
7. python3 manage.py runserver
8. use POSTMAN for using API's or goto url directly.
9. (optional) for admin run :- python3 manage.py createsuperuser

List of apis :-

1. http://127.0.0.1:8000/admin/ for admin login (use username = admin and password = Admin@123#)
2. http://127.0.0.1:8000/myapi/books for fetching all books
3. http://127.0.0.1:8000/myapi/books/details?uuid=6065910 where uuid is book uuid for searchng book in db.
4. http://127.0.0.1:8000/myapi/authors for fetching all authors.

List of codes :-

1. 1000 (successfull)
2. 1001 (something went wrong)

List of short forms :-

1. bo_id = book_id
2. ti = title
3. desc = description
4. au_id = author_id

Request Response Cycle

1. GET, POST (type of request allowed)

   Request
   GET http://127.0.0.1:8000/myapi/books

   Response (will be list of books)

```
{
    "code": 1000,
    "message": [
        {
            "bo_id": 2450966, (book_id)
            "ti": "My new python book for everyone", (title)
```

```
            "desc": "None", (description)
            "au_id": 2 (author_id)
        },
        {
            "bo_id": 7794720,
            "ti": "Python for student",
            "desc": "python book",
            "au_id": 1
        },
        {
            "bo_id": 6065910,
            "ti": "Book of Eli",
            "desc": "this boot refers to book of god",
            "au_id": 4
        }
    ],
    "status": true,
    "rid": "4e839ecb1f5911ebb023c14960b931a7"
}
```


Request
POST http://127.0.0.1:8000/myapi/books


```
{
    "bo_id": 7794720,
    "ti": "Python for student",
    "desc": "python book",
    "au_id": 1
}
```


Response


```
{
    "code": 1000,
```

```
    "message": "Book Created",
    "status": "true”,
    "rid": 6f54b7531f5d11ebb023c14960b931a7
}
```


2. GET, PUT, DELETE

   Request
   GET http://127.0.0.1:8000/myapi/books/details?uuid=6065910 (uses key and value where key is uuid and value id book uuid)

   Response


```
{
    "code": 1000,
   "message": {
        "bo_id": 6065910,
        "ti": "Book of Eli",
        "desc": "this boot refers to book of god",
        "au_id": 4,
        "author_uuid": 7585119,
        "first_name": "janardhan",
        "last_name": "singh",
        "email": "adj@daf.aef"
   },
   "status": true,
   "rid": "24d157421f4d11ebb023c14960b931a7"
}
```


   Request
   PUT http://127.0.0.1:8000/myapi/books/details?uuid=2450966


```
{
    "code": 1000,
   "message": {
        "bo_id": 6065910,
```

```
            "ti": "Book of Eli",
            "desc": "My Book",
            "au_id": 4,
        },
        "status": true,
        "rid": "24d157421f4d11ebb023c14960b931a7"
}
```


Response


```
{
        "code": 1000,
        "message": "book with id '6065910' updated successfully",
        "status": true,
        "rid": "ca86678c1f5b11ebb023c14960b931a7"
}
```


Request
DELETE http://127.0.0.1:8000/myapi/books/details?uuid=6065910

Response


```
{
        "code": 1000,
        "message": "book with id '6065910' deleted successfully",
        "status": true,
        "rid": "047b8b791f5c11ebb023c14960b931a7"
}
```


3. GET, POST

Request
GET http://127.0.0.1:8000/myapi/authors (will give the list of authors)

Response


```
{
        "code": 1000,
```

```
    "message": [
        {
            "id": 1,
            "au_id": 3570681,
            "f_n": "janardhan",
            "l_n": "singh",
            "email": "abc@sadf.adf",
            "nt": "none"
        },
        {
            "id": 2,
            "au_id": 2392070,
            "f_n": "kumar",
            "l_n": "rahul",
            "email": "abc@sadf.ca",
            "nt": "Nul"
        },
        {
            "id": 4,
            "au_id": 7585119,
            "f_n": "janardhan",
            "l_n": "singh",
            "email": "adj@daf.aef",
            "nt": null
        },
    "status": true,
    "rid": "6f54b7531f5d11ebb023c14960b931a7"
}
```


Request
POST http://127.0.0.1:8000/myapi/authors

```
{
      "f_n": "janardhan",
      "l_n": "singh",
      "email": "jds@daf.aef",
 }
```


Response


```
{
      "code": "1000",
      "message": "author created with id [3840210]",
      "status": "true",
      "rid": "6f54b7531f5d11ebb023c14960b931a7"
 }
```


Database Schema

1. for creating author table
   CREATE TABLE "app_author" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "author_uuid" integer NOT NULL, "first_name" varchar(45) NOT NULL, "last_name" varchar(45) NOT NULL, "email" varchar(255) NOT NULL UNIQUE, "notes" text NULL, "is_active" bool NOT NULL, "created_at" datetime NOT NULL, "created_by" varchar(255) NOT NULL, "updated_at" datetime NOT NULL, "updated_by" varchar(255) NOT NULL)

2. for creating book table
   CREATE TABLE "app_book" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "book_uuid" integer NOT NULL, "title" varchar(255) NOT NULL, "description" text NOT NULL, "is_active" bool NOT NULL, "created_at" datetime NOT NULL, "created_by" varchar(255) NOT NULL, "updated_at" datetime NOT NULL, "updated_by" varchar(255) NOT NULL, "author_id" integer NOT NULL REFERENCES "app_author" ("id") DEFERRABLE INITIALLY DEFERRED)