

1. What are the various features you would like your project to offer?
2. What are the API endpoints that you would need to set up for each feature? List them along with the respective HTTP verb, endpoint URL, and any special details (query parameters, request bodies, headers).
3. Provide a description of the database tables required for your application, including column names, data types, constraints, and foreign keys. Include your database name. You can optionally include an ER diagram.

Students Management

1. Use existing Django project to create API endpoints to display list of student's records
2. Using Get, Post, Put/Patch, and Delete to retrieve information about students
3. A database which contains 3 tables to manage the data for a software school

Students

Courses

Registrations

API Reference Table of endpoint paths, methods, parameters

API for students

Index

Show all the students in the database

GET <http://localhost:5000/students>

Show

Show the student with specific id

GET <http://localhost:5000/students/id>

Create

Add a new student to the database using JSON format

POST <http://localhost:5000/students>

E.g.

```
{  
  "id": 13,  
  "first_name": "Abel",  
  "last_name": "Castillo",  
  "email": "abel.castillo@escuela.com",  
  "course_id": 4  
}
```

Delete

Delete the student with the specific id

GET <http://localhost:5000/students/id>

API for courses

Index

Show all the courses in the database

GET <http://localhost:5000/courses>

Create

Add a new course to the database using JSON format

POST <http://localhost:5000/courses>

```
{  
  "id": 5,  
  "name": "R",  
  "units": 4  
}
```

Delete

Delete the course with the specific id

GET <http://localhost:5000/courses/id>