# Student Records CRUD API

**Student Records CRUD API**

Student Records CRUD API is the api to do all the basic (create ,update ,read ,delete) in the database.

Here the database is student database and the entities have the following fields:

* name
* age
* department
* gpa
* registrationDate
* createdAt
* modifiedAt

The following are the api requests for the CRUD operations:

**POST**

Create new student record:

http://localhost:3000/student/CREATE

The create request is done as below

**Example:**

**Request:**

curl --location 'http://localhost:3000/student/CREATE' \

--data '{

"name":"adarsh",

"department":"cse",

"gpa":9,

"registration\_date":"2021-10-10",

"age":20,

"attendance":80

}'

**Response:**

{

"status": "201 Created",

"stud": {

"name": "adarsh",

"department": "cse",

"age": 20,

"gpa": 9,

"attendance": 80,

"createdAt": "2023-03-16T15:34:31.863Z",

"modifiedAt": "2023-03-16T15:34:31.863Z"

}

}

**GET**

Read all the students

The below request is used to fetch all the student details

**Example:**

**Request:**

curl --location 'http://localhost:3000/student/READ'

**Response:**

{

"status": "200 OK",

"type": {

"contentType": "application/json"

},

"allStudents": [

{

"id": "12",

"name": "adarsh",

"department": "cse",

"age": 20,

"gpa": 9,

"registrationDate": null,

"attendance": 80,

"createdAt": "2023-03-16T15:45:34.000Z",

"modifiedAt": "2023-03-16T15:45:34.000Z"

},

{

"id": "13",

"name": "uma",

"department": "cse",

"age": 19,

"gpa": 9.5,

"registrationDate": null,

"attendance": 90,

"createdAt": "2023-03-16T15:45:59.000Z",

"modifiedAt": "2023-03-16T15:45:59.000Z"

},

{

"id": "14",

"name": "anirudh",

"department": "cse",

"age": 19,

"gpa": 9.8,

"registrationDate": null,

"attendance": 100,

"createdAt": "2023-03-16T15:46:13.000Z",

"modifiedAt": "2023-03-16T15:46:13.000Z"

}

]

}

**GET**

Read student detail by id:

http://localhost:3000/student/READ/

**Example:**

**Request:**

curl --location 'http://localhost:3000/student/READ/14'

**response:**

{

"status": "200 found",

"type": {

"contentType": "application/json"

},

"students": {

"id": "14",

"name": "anirudh",

"department": "cse",

"age": 19,

"gpa": 9.8,

"registrationDate": null,

"attendance": 100,

"createdAt": "2023-03-16T15:46:13.000Z",

"modifiedAt": "2023-03-16T15:46:13.000Z"

}

}

**GET**

Read students sorted based on registration date

The above request is used to fetch the sorted array of students based on the registration date

The below request is used to retrieve details based on id

**Example:**

**Response:**

curl --location 'http://localhost:3000/student/registration\_date'

**Response:**

{

"status": "200 found",

"type": {

"contentType": "application/json"

},

"sortedStudents": [

{

"id": "14",

"name": "anirudh",

"department": "cse",

"age": 19,

"gpa": 9.8,

"registrationDate": null,

"attendance": 100,

"createdAt": "2023-03-16T15:46:13.000Z",

"modifiedAt": "2023-03-16T15:46:13.000Z"

},

{

"id": "13",

"name": "uma",

"department": "cse",

"age": 19,

"gpa": 9.5,

"registrationDate": null,

"attendance": 90,

"createdAt": "2023-03-16T15:45:59.000Z",

"modifiedAt": "2023-03-16T15:45:59.000Z"

},

{

"id": "12",

"name": "adarsh",

"department": "cse",

"age": 20,

"gpa": 9,

"registrationDate": null,

"attendance": 80,

"createdAt": "2023-03-16T15:45:34.000Z",

"modifiedAt": "2023-03-16T15:45:34.000Z"

}

]

}

**GET**

Read student’s based on weighted score

The above request is used to fetch the descending sorted array of students based on the weighted average

weighted average = 0.7\*gpa + 0.3\*attendance

**Example:**

**Request:**

curl --location 'http://localhost:3000/student/weighted\_score'

**Response:**

{

"status": "200 found",

"type": {

"contentType": "application/json"

},

"sortedStudents": [

{

"id": "14",

"name": "anirudh",

"department": "cse",

"age": 19,

"gpa": 9.8,

"registrationDate": null,

"attendance": 100,

"createdAt": "2023-03-16T15:46:13.000Z",

"modifiedAt": "2023-03-16T15:46:13.000Z"

},

{

"id": "13",

"name": "uma",

"department": "cse",

"age": 19,

"gpa": 9.5,

"registrationDate": null,

"attendance": 90,

"createdAt": "2023-03-16T15:45:59.000Z",

"modifiedAt": "2023-03-16T15:45:59.000Z"

},

{

"id": "12",

"name": "adarsh",

"department": "cse",

"age": 20,

"gpa": 9,

"registrationDate": null,

"attendance": 80,

"createdAt": "2023-03-16T15:45:34.000Z",

"modifiedAt": "2023-03-16T15:45:34.000Z"

}

]

}

**PATCH**

Update student records

http://localhost:3000/student/UPDATE/

The below request is used to update the records based on the id

**Example:**

**Request:**

curl --location --request PATCH 'http://localhost:3000/student/UPDATE/13' \

--data '{

"gpa":9.3

}'

**Response:**

{

"status": "200 OK",

"type": {

"contentType": "application/json"

},

"stud": {

"id": "13",

"name": "uma",

"department": "cse",

"age": 19,

"gpa": 9.3,

"registrationDate": null,

"attendance": 90,

"createdAt": "2023-03-16T15:45:59.000Z",

"modifiedAt": "2023-03-16T15:45:59.000Z"

}

}

**DELETE**

Delete student records

http://localhost:3000/student/DELETE/

the above request is used to delete student records based on id

**Example:**

**Request:**

curl --location --request DELETE 'http://localhost:3000/student/DELETE/12'

**Response**

{

"status": "204 no content",

"deleteResponse": 1

}

**ERROR HANDLING:**

**POST:**

Create errors

This is the empty body error for create

**Example:**

**Request:**

curl --location 'http://localhost:3000/student/CREATE'

--data '{}'

**Response:**

400 Bad request

**GET**

Read student id not found error

The below response is returned when id is not found

**Example:**

**Response:**

curl --location 'http://localhost:3000/student/READ/8'

**Request:**

{

"statusCode": 400,

"message": "Record not found"

}

**PATCH**

Update record not found

http://localhost:3000/student/UPDATE/

The below is the response when the update id not found

**Example:**

**Request:**

curl --location --request PATCH 'http://localhost:3000/student/UPDATE/8'

**Response:**

{

"statusCode": 403,

"message": "no such row found"

}

**DELETE**

Delete record not found

http://localhost:3000/student/DELETE/

The below is returned when the id is not found for delete request

**Example:**

**Request:**

curl --location --request DELETE 'http://localhost:3000/student/DELETE/9'

**Response:**

404 not found