**Student Management API Documentation**

**Overview**

The StudentController is a RESTful web controller that manages Student entities. It provides endpoints to create, retrieve, update, and delete students. The controller interacts with the service layer via StudentService.

**🔗 Base URL**

/api/students

**📄 Endpoints**

**1. Get All Students**

* **URL:** /api/students
* **Method:** GET
* **Description:** Fetches a list of all students.
* **Roles Required:** Admin
* **Response:**
  + 200 OK – Returns a list of all Student objects.
* **Sample Response:**

[

    {

        "id": 1,

        "rollNo": 1,

        "name": "name",

        "dob": "2025-05-11",

        "email": "em@kl.com",

        "address": "address\_5c94de438631",

        "classId": 1,

        "phoneNumber": "1234567890"

    },

    {

        "id": 3,

        "rollNo": 3,

        "name": "name3b",

        "dob": "2025-05-10",

        "email": "em3b@kl.com",

        "address": "address\_2b5c94de438631",

        "classId": 2,

        "phoneNumber": "1234567892"

    }

]

**2. Get Student by ID**

* **URL:** /api/students/{id}
* **Method:** GET
* **Description:** Retrieves a student by their ID.
* **Roles Required:** Admin, Student, Teacher
* **Path Parameters:**
  + id – Long – ID of the student to retrieve.
* **Response:**
  + 200 OK – Student found and returned.
  + 404 Not Found – Student with the given ID does not exist.
* **Sample Response:**

{

    "id": 1,

    "rollNo": 1,

    "name": "name",

    "dob": "2025-05-10",

    "email": "em3@kl.com",

    "address": "address\_25c94de438631",

    "classId": 1,

    "phoneNumber": "1234567892"

}

**3. Create New Student**

* **URL:** /api/students
* **Method:** POST
* **Description:** Creates a new student.
* **Roles Required:** Admin, Teacher
* **Request Body:**

{

  "rollNo": 1,

  "name": "name",

  "dob": "2025-05-10",

  "email": "em3@kl.com",

  "address": "address\_25c94de438631",

  "classId": 1,

  "phoneNumber": "1234567892"

}

* **Response:**
  + 201 Created – Student successfully created.
  + 400 Bad Request – Invalid input.

**4. Update Existing Student**

* **URL:** /api/students/{id}
* **Method:** PUT
* **Description:** Updates the student with the given ID.
* **Roles Required:** Admin, Student, Teacher
* **Path Parameters:**
  + id – Long – ID of the student to update.
* **Request Body:**

{

  "dob": "2025-05-10",

  "email": "em3b@kl.com"

}

* **Response:**
  + 200 OK – Student updated.
  + 404 Not Found – Student with the given ID not found.

**5. Delete Student**

* **URL:** /api/students/{id}
* **Method:** DELETE
* **Description:** Deletes the student with the given ID.
* **Roles Required:** Admin
* **Path Parameters:**
  + id – Long – ID of the student to delete.
* **Response:**
  + 204 No Content – Student deleted.
  + 404 Not Found – Student not found.

**6. Get Students Count**

* **URL:** /api/students/count/school
* **Method:** GET
* **Description:** Gets the count of students in the entire school
* **Roles Required:** Admin
* **Response:**
  + 200 OK – Returns the count of all the Student present in school
* **Sample Response:**

{

  "count": "1290",

}

**7. Get Students by Class**

* **URL:** /api/students/class/{classId}
* **Method:** GET
* **Description:** Fetches a list of all students.
* **Roles Required:** Admin, Teacher
* **Path Parameters:**
  + id – Long – ID of the class to fetch students from.
* **Response:**
  + 200 OK – Returns a list of all Student objects from the class.
  + 404 Not Found – Class not found.
* **Sample Response:**

[

    {

        "id": 1,

        "rollNo": 1,

        "name": "name",

        "dob": "2025-05-11",

        "email": "em@kl.com",

        "address": "address\_5c94de438631",

        "classId": 1,

        "phoneNumber": "1234567890"

    },

    {

        "id": 3,

        "rollNo": 2,

        "name": "name3b",

        "dob": "2025-05-10",

        "email": "em3b@kl.com",

        "address": "address\_2b5c94de438631",

        "classId": 1,

        "phoneNumber": "1234567892"

    }

]