

## 1. Create (Add a New Student)

### Endpoint:

- **URL:** http://localhost:8080/api/v1/students
- **Method:** POST

### Steps:

1. Open Postman and create a new request.
2. Set the request method to POST.
3. Enter the URL: http://localhost:8080/api/v1/students
4. Go to the Body tab and select raw and JSON from the dropdown.
5. Enter the following JSON in the request body:

```
{  
  "studentId": "S001",  
  "studentName": "John Doe",  
  "studentClass": "Class A",  
  "teacherComment": "Excellent student"  
}
```

6. Click Send to send the request.
7. Review the response to ensure the student was added successfully.

## 2. Read (Get All Students)

### Endpoint:

- **URL:** http://localhost:8080/api/v1/students
- **Method:** GET

### Steps:

1. Open Postman and create a new request.
2. Set the request method to GET.
3. Enter the URL: http://localhost:8080/api/v1/students
4. Click Send to send the request.
5. Review the response to see the list of all students.

### 3. Read (Get a Specific Student by ID)

#### Endpoint:

- **URL:** `http://localhost:8080/api/v1/students/{id}`
- **Method:** GET

#### Steps:

1. Open Postman and create a new request.
2. Set the request method to GET.
3. Enter the URL: `http://localhost:8080/api/v1/students/S001` (replace S001 with the actual student ID).
4. Click Send to send the request.
5. Review the response to see the details of the specific student.

### 4. Update (Update a Student's Details)

#### Endpoint:

- **URL:** `http://localhost:8080/api/v1/students/{id}`
- **Method:** PUT

#### Steps:

1. Open Postman and create a new request.
2. Set the request method to PUT.
3. Enter the URL: `http://localhost:8080/api/v1/students/S001` (replace S001 with the actual student ID).
4. Go to the Body tab and select raw and JSON from the dropdown.
5. Enter the following JSON in the request body:

```
{
  "studentName": "John Doe",
  "studentClass": "Class A",
  "teacherComment": "Updated comment"
}
```

6. Click Send to send the request.
7. Review the response to ensure the student was added successfully.

## 5. Delete (Delete a Student by ID)

### Endpoint:

- **URL:** `http://localhost:8080/api/v1/students/{id}`
- **Method:** DELETE

### Steps:

1. Open Postman and create a new request.
2. Set the request method to DELETE.
3. Enter the URL: `http://localhost:8080/api/v1/students/S001` (replace S001 with the actual student ID).
4. Click Send to send the request.
5. Review the response to ensure the student was deleted successfully.
6. Filter Students by Teacher Comment

## 6. Filter by teacher comment

### Endpoint:

- **URL:** `http://localhost:8080/api/v1/students/filter`
- **Method:** GET
- **Query Parameter:** `comment=excellent`

### Steps:

1. Open Postman and create a new request.
2. Set the request method to GET.
3. Enter the URL: `http://localhost:8080/api/v1/students/filter`
4. Add a query parameter:
5. Key: `comment`
6. Value: `excellent` (or any other comment you want to filter by)
7. Click Send to send the request.
8. Review the response to see the list of students whose teacher comments contain the specified string.
9. Example Request to Filter Students by Teacher Comment:  
`http://localhost:8080/api/v1/students/filter?comment=excellent`

**URL:**

### Example Response 1:

```
{
  "studentId": "S001",
  "studentName": "John Doe",
  "studentClass": "Class A",
  "teacherComment": "Excellent student"
},
```

## Example Response 2:

```
{
  "studentId": "S002",
  "studentName": "Jane Smith",
  "studentClass": "Class B",
  "teacherComment": "Excellent progress"
}
```

## 7. Create a Fraudulent Example

- **Example:** Attempt to Add a Student with Missing Required Fields

### Postman Request:

- **Method:** POST
- **URL:** <http://localhost:8080/api/v1/students>

### Body:

```
{
  "studentId": "S002",
  "studentName": "",
  "studentClass": "Class B",
  "teacherComment": "Needs improvement"
}
```

### Steps:

1. Open Postman and create a new request.
2. Set the request method to POST.
3. Enter the URL: <http://localhost:8080/api/v1/students>.
4. Go to the Body tab, select raw, and choose JSON from the dropdown.
5. Enter the JSON body with the student details, **intentionally** leaving the studentName field empty.

## 8. Create a Fraudulent Example

- **Example:** Attempt to Update a Student with an Invalid ID

### Postman Request:

- **Method:** PUT
- **URL:** <http://localhost:8080/api/v1/students/invalid-id>

### Body:

```
{
  "studentName": "Jane Doe",
  "studentClass": "Class B",
  "teacherComment": "Updated comment"
}
```

**Steps:**

1. Open Postman and create a new request.
2. Set the request method to PUT.
3. Enter the URL: `http://localhost:8080/api/v1/students/invalid-id`.
4. Go to the Body tab, select raw, and choose JSON from the dropdown.
5. Enter the JSON body with the updated student details.
6. Click Send to send the request.
7. Review the response to see if the API handles the error correctly.
8. Expected Response:
9. The API should return a 404 Not Found status code with an error message indicating that the student with the specified ID does not exist.