**Student Scholarship Eligibility Service**

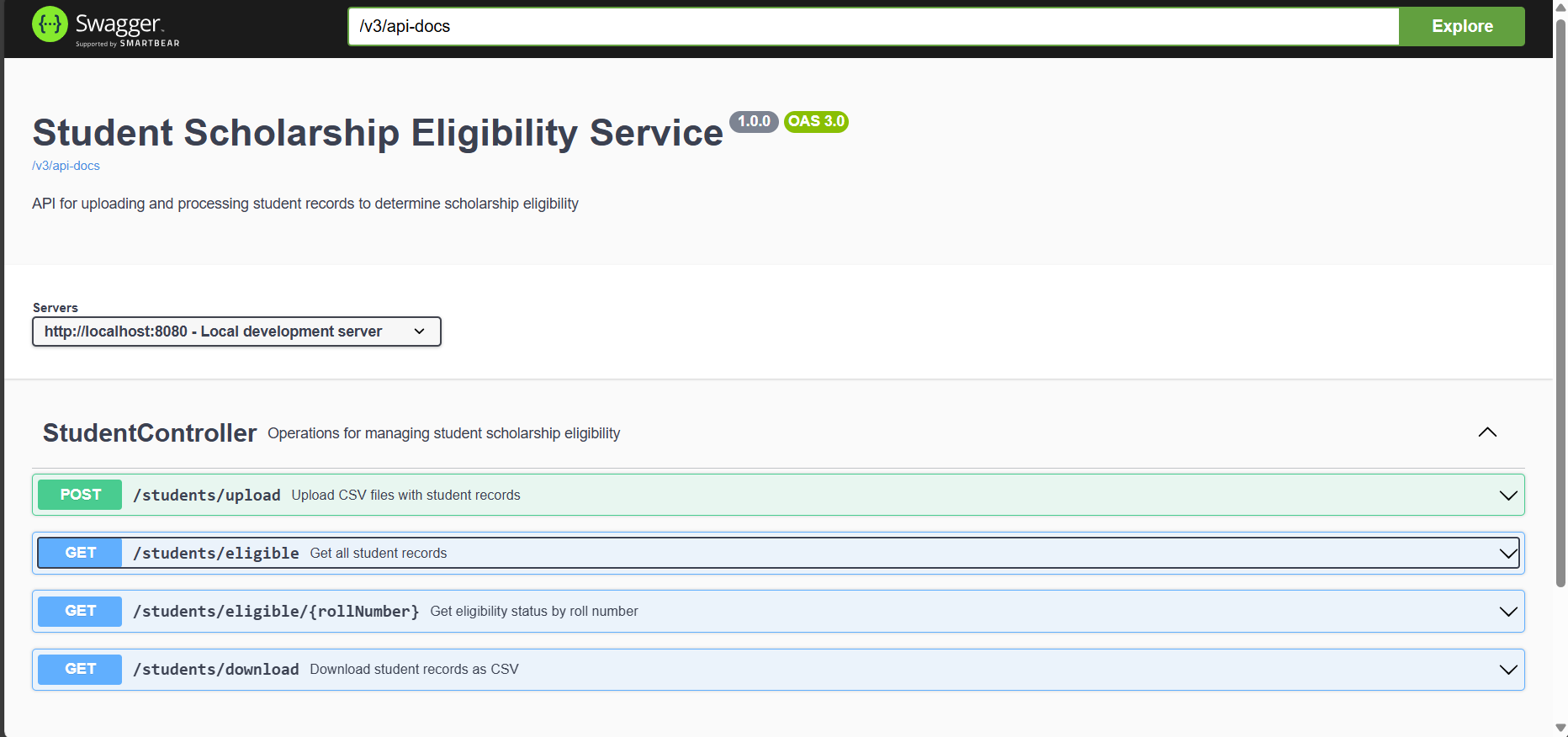
**API Documentation**

**Overview**

The Student Scholarship Eligibility Service is a Spring Boot application designed to upload, process, and manage student records to determine scholarship eligibility based on specified criteria. It supports CSV file uploads, and dynamic eligibility criteria, and provides various endpoints to interact with student data.

**Swagger Integration**

This application integrates Swagger for API documentation and testing. You can access the Swagger UI by visiting the following URL: [Swagger UI](http://localhost:8080/swagger-ui.html).



**Endpoints**

**1. Upload CSV File**

Upload a CSV file containing student records to determine their scholarship eligibility.

* **URL**: /students/upload
* **Method**: POST
* **Content-Type**: multipart/form-data
* **Parameters**:
  + file (MultipartFile) - The CSV file to upload.
  + science (float) - Science criteria for eligibility.
  + maths (float) - Maths criteria for eligibility.
  + english (float) - English criteria for eligibility.
  + computer (float) - Computer criteria for eligibility.

**Example Request:**

* **POST**: <http://localhost:8080/students/upload>

**Screenshot Response:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**2. Get Eligibility Status by Roll Number**

Retrieve the scholarship eligibility status of a student by their roll number.

* **URL**: /students/eligible/{rollNumber}
* **Method**: GET
* **Response**: JSON object of the student.

**Example Request:**

* **GET**: <http://localhost:8080/students/eligible/100101>

**Screenshot Response:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**3. Get All Student Records**

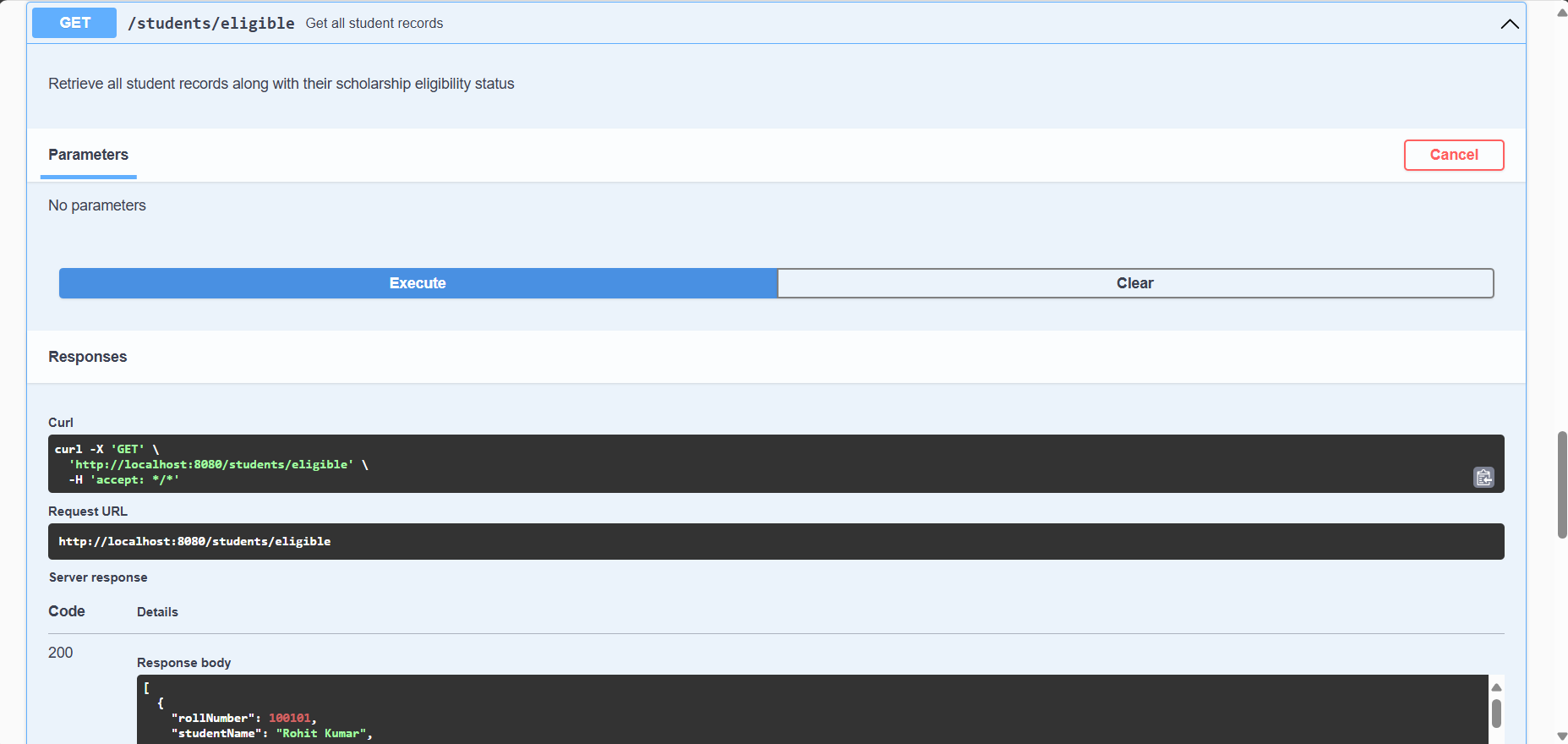
Retrieve all student records along with their scholarship eligibility status.

* **URL**: /students/eligible
* **Method**: GET
* **Response**: JSON array of students.

**Example Request:**

* **GET** <http://localhost:8080/students/eligible>

**Screenshot Response**



**4. Download Student Records as CSV**

Download all student records and their scholarship eligibility status as a CSV file.

* **URL**: /students/download
* **Method**: GET
* **Response**: CSV file.

**Example Request:**

* **GET** <http://localhost:8080/students/download>
* File Name:- students.csv

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**4. Functionality**

1. **Student Entity (com.student.entity.Student):**
   * Defines the structure of a student entity with attributes such as roll number, student name, scores in different subjects, and eligibility status.
2. **Student Repository (com.student.repository.StudentRepository):**
   * Interface extending JpaRepository to perform CRUD operations on the Student entity.
3. **Student Service (com.student.service.StudentService):**
   * Provides various methods to handle business logic related to student data.
   * uploadCSV: Accepts a CSV file containing student records, processes the records to determine eligibility based on provided criteria, and saves the records to the database.
   * saveStudents: Saves a list of student records to the database.
   * getStudentByRollNumber: Retrieves a student record by roll number.
   * getAll: Retrieves all student records from the database.
   * parseCSVFile: Parses the uploaded CSV file, distributes the parsing task across multiple threads for efficiency, and processes each chunk of data to determine eligibility.
   * processChunk: Processes a chunk of CSV records to determine eligibility for each student.
   * checkEligibility: Checks whether a student is eligible for a scholarship based on provided criteria.
   * generateCSVData: Retrieves all student records from the database and generates a CSV file containing the records along with their eligibility status.
4. **Student Controller (com.student.controller.StudentController):**
   * Exposes REST endpoints for interacting with student data.
   * uploadCSV: Handles file upload of CSV containing student records and processes them.
   * getEligibility: Retrieves the eligibility status of a student by roll number.
   * getAll: Retrieves all student records along with their eligibility status.
   * downloadCSV: Provides an endpoint to download all student records along with their eligibility status as a CSV file.
5. **OpenAPI Annotations:**
   * Annotations from io.swagger.v3.oas.annotations package are used to document the API endpoints and provide additional information like summary, description, response codes, etc.
6. **Maven Configuration (pom.xml):**
   * Dependencies are managed using Maven. Key dependencies include Spring Boot starters for web, data JPA, validation, and testing. Additionally, dependencies for Apache Commons CSV, MySQL connector, and Springdoc OpenAPI are included.

