**Webtoon Collection API**

This project is a RESTful API that manages a collection of webtoons inspired by “Top 14 Castle Swimmer Characters You Need to Know.” It provides endpoints to fetch, add, and delete webtoon entries. The API is built using Spring Boot and MongoDB, ensuring efficiency, security, and optimized performance.

**Features**

* **GET /webtoons**: Fetch all webtoons with basic details (title, description, and characters).
* **POST /webtoons**: Add a new webtoon entry, including the title, summary, and characters.
* **GET /webtoons/{id}**: Fetch a specific webtoon by its ID, returning detailed information.
* **DELETE /webtoons/{id}**: Remove a webtoon entry by its ID.

**Tech Stack**

* **Backend**: Spring Boot
* **Database**: MongoDB
* **Security**: JWT-based authentication, input validation, and rate limiting

**Installation**

**Prerequisites**

* Java 11 or later
* Maven
* MongoDB

**Steps**

1. Clone the repository:

bash

Copy code :

git clone https://github.com/your-username/webtoon-api.git

1. Navigate to the project directory:

bash

Copy code

cd webtoon-api

1. Install dependencies:

bash

Copy code

mvn clean install

1. Set up your MongoDB database and update the application.properties file with your MongoDB URI:

properties

Copy code

spring.data.mongodb.uri=mongodb://localhost:27017/webtoonDB

1. Run the application:

bash

Copy code

mvn spring-boot:run

**API Endpoints**

**1. Fetch All Webtoons**

* **Endpoint**: GET /webtoons
* **Description**: Returns all webtoons with basic details.
* **Response**:

json

Copy code

[

{

"id": "1",

"title": "Castle Swimmer",

"description": "A story about prophecies and unlikely heroes",

"characters": ["Kappa", "Siren", "Prophecy Fish"]

}

]

**2. Fetch a Webtoon by ID**

* **Endpoint**: GET /webtoons/{id}
* **Description**: Fetches a specific webtoon entry by its ID.
* **Response**:

json

Copy code

{

"id": "1",

"title": "Castle Swimmer",

"description": "A story about prophecies and unlikely heroes",

"characters": ["Kappa", "Siren", "Prophecy Fish"]

}

**3. Add a New Webtoon**

* **Endpoint**: POST /webtoons
* **Description**: Adds a new webtoon entry.
* **Authentication**: JWT token required.
* **Request Body**:

json

Copy code

{

"title": "New Webtoon",

"description": "Summary of the webtoon",

"characters": ["Character 1", "Character 2"]

}

**4. Delete a Webtoon by ID**

* **Endpoint**: DELETE /webtoons/{id}
* **Description**: Removes a webtoon entry by its ID.
* **Authentication**: JWT token required.

**Security Features**

1. **JWT Authentication**: The POST and DELETE endpoints are secured with JWT-based authentication to ensure only authorized users can add or delete webtoons.
2. **Input Validation**: Validates all incoming data to prevent malicious inputs.
3. **Rate Limiting**: Implements rate limiting to protect against excessive API calls.

**Testing**

* The API has been tested using **Postman**.
* Unit tests are written using **JUnit** and **Mockito**.
* To run tests, use:

bash

Copy code

mvn test

**Deployment**

This application can be deployed on any cloud platform supporting Java applications, such as AWS, Azure, or Heroku. Ensure the MongoDB instance is accessible from the deployed environment.

**Future Enhancements**

* Implement user roles and permissions for enhanced security.
* Add support for updating webtoon entries.
* Integrate Swagger for API documentation.

**Contribution**

Feel free to fork the repository and submit pull requests. Contributions are welcome!