DESIGN DOC

Overview: In this task, you'll create a Spring Boot application that integrates with the Rick and Morty API to fetch and store character data in a MySQL database. The application will allow management of character data through various CRUD operations and an approval workflow. This includes adding new characters, fetching characters from both the external API and your database, and updating the approval status of characters.

The primary objective is to integrate the Rick and Morty API into a Spring Boot application. This involves consuming the Character APIs, storing the data in a MySQL database, and creating custom APIs to manage the characters within the database. Additionally, an approval process for characters will be implemented using another set of APIs**.**

Goals:

- **API Integration:** Learn how to connect and use external APIs in a Spring Boot application.
- **Database Operations with JPA:** Master CRUD operations with MySQL using JPA and Hibernate. Understand the basics of CRUD operations and how to implement them using repository interfaces.
- **Building RESTful APIs:** Understand how to create and manage RESTful APIs in Spring Boot. Learn how to create controllers and define endpoints for various API operations.
- **Error handling:** Learn how to implement error handling in Spring Boot applications, understand how to provide meaningful error responses to API consumers.
- **JSON Processing:** Handle JSON data
- **Application Configuration:** Configure Spring Boot application
- Learn how to break down complex tasks into manageable parts and prioritize work effectively.

Rick and Morty API Documentation
Refer to the [Rick and Morty API documentation]
(https://rickandmortyapi.com/documentation/#character) for detailed information.

For this project, you will be using only the Character APIs.

Character schema for database.

Key	Type	Description
id	int	The id of the character.
name	string	The name of the character.

status	string	The status of the character ('Alive', 'Dead' or 'unknown').
species	string	The species of the character.
type	string	The type or subspecies of the character.
gender	string	The gender of the character ('Female', 'Male', 'Genderless' or 'unknown').
origin	object	Name and link to the character's origin location.
approval_status	enum	Status of the characters approval after creation
location	object	Name and link to the character's last known location endpoint.
image	string (url)	Link to the character's image. All images are 300x300px and most are medium shots or portraits since they are intended to be used as avatars.
episode	array (urls)	List of episodes in which this character appeared.
url	string (url)	Link to the character's own URL endpoint.
created	string	Time at which the character was created in the database.

APIs
Get all characters: `GET https://rickandmortyapi.com/api/character`
Get a single character: `GET https://rickandmortyapi.com/api/character/2`
Get multiple characters: `GET https://rickandmortyapi.com/api/character/1,183`
Filter characters: `GET https://rickandmortyapi.com/api/character/?name=rick&status=alive`

Part 1

1. **Consume the APIs:**
- Fetch characters from the Rick and Morty APIs and save them to your database.

2. **Create an API to add a character:**
- Upon character creation, the `approval_status` should be set to `NOT_APPROVED`.

3. **Create an API to fetch a single character by ID:**
- Retrieve a character from your database using its ID.

4. **Create an API to fetch all characters:**
- Retrieve all characters from your database.

5. **Create an API to delete a character by ID:**
- Remove a character from your database using its ID.

API: Refer to the JSONBlop API documentation (https://jsonblob.com/api)
Also for more focus: see the attached postman documentation.
Part 2

1. **Create another controller:**
- Develop a new controller for the additional functionalities.

2. **Create a POST request to the JSONBlob to send a character for approval:**
- Submit a character from your database for approval.

3. **Create an API to get the sent character from the JSONBlob by ID:**
- Retrieve the submitted character using its blobID.

4. **Create an API to update a character's approval status:**
- Change the character's `approval_status` to `APPROVED` and update the database.

Notes

- Ensure proper error handling and validation in all APIs.
- Use appropriate HTTP status codes for responses.
- USE MYSQL for the database.
- Implement logging and monitoring for better debugging and maintenance.

If you need further clarification or additional information, please feel free to ask.

-- look at Dtos, use a dto to get the characters and map to your character entity, then save the characters from the entity to your database.

Resources

1. Spring Boot Documentation: [Spring Boot Reference Guide](https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/)
2. Spring Data JPA Documentation: [Spring Data JPA Reference](https://docs.spring.io/spring-data/jpa/docs/current/reference/html/)
3. MySQL Documentation: [MySQL Reference Manual](https://dev.mysql.com/doc/refman/8.0/en/)

4. Rick and Morty API Documentation: [Rick and Morty API Docs](https://rickandmortyapi.com/documentation/#character)
5. JSON Processing in Java: [Jackson Library Documentation](https://github.com/FasterXML/jackson)
6. Spring Boot Tutorials: [Baeldung Spring Boot Tutorials](https://www.baeldung.com/spring-boot)