

Create a “Login” REST web service API

INSTRUCTIONS:

1. Create a REST web service API with the following usage:

Usage:

| | |
|----------------------------|---|
| URL: | {endpoint}/login |
| Headers: | Accept = application/json Content-Type = application/json Authorization = Basic Auth <ul style="list-style-type: none">- HTTP requests with the "Authorization" header containing the word "Basic", a space character, and a "username:password" string encoded in Base64.- <i>Example:</i> Basic dXNlcm5hbWU6MTIzNDU2Nzg= |
| Path Variables: | N/A |
| Request Parameters: | N/A |
| Request Method: | GET |
| Request: Body: | N/A |

Sample Response:

| | |
|-----------------------------|---|
| Successful Response: | { "lastName": "Dela Cruz", "firstName": "Juan", "middleName": "A", "birthdate": "01/01/2000" } |
| Failed Response: | { "message": "Login Failed!" } |

2. Create a **User** object with the following fields:

| Data Type | Field Name |
|-----------|------------|
| String | username |
| String | password |
| String | lastName |
| String | firstName |

| | |
|--------|------------|
| String | middleName |
| Date | birthdate |

3. Create a static dummy data for the **User** object. The value of the password should be encrypted using MD5.

Sample dummy data:

| username | password | lastName | firstName | middleName | birthdate |
|-----------|--|-----------|-----------|------------|------------|
| jdelacruz | 25d55ad283a a400af464c76 d713c07ad | Dela Cruz | Juan | A | 01/01/2000 |
| pmanaloto | 25d55ad283a a400af464c76 d713c07ad | Manaloto | Pepito | A | 02/02/2000 |

Note: 25d55ad283aa400af464c76d713c07ad is the MD5 encrypted value of 12345678.

4. Extract username and password in the authorization header.
5. Encrypt the password from the authorization header into MD5 hash.
6. Check if the supplied username/password is equal to any of the dummy data's username/password then return the **User** details (**lastName, firstName, middleName and birthdate**). Else, return **"Login Failed!"**.

Use the following technologies as much as possible:

- Maven
- Spring framework (preferably use **XML configuration**)
- Apache CXF

OPTIONAL TASK:

This is not required but would serve as an advantage.

1. Create a database for the User object; and
2. Retrieve data from the database instead of the dummy data.

Use the following technologies as much as possible:

- Hibernate (preferably use **XML configuration**)
- PostgreSQL

Note:

Once completed, kindly save the whole project/code in zip file and name it with your full name (example: juan.delacruz.zip) then email and attach this as your proof of completion.

If you do the optional task, please export the database schema with data and attach to the email.

Please let me know if you encounter any issues in submitting your work. Enjoy & Goodluck!