1. Options to develop Rest Api :
2. Jersey
3. Spring Rest

Approach: We are Implementing Spring Rest using spring boot because it provides in-built support for developing rest end points. It will also help in DI and security. Spring internally takes care of Marshalling/Unmarshalling and code maintainability is also easy.

1. Options for Authentication :
2. JWT
3. Oauth2
4. Basic Oauth

Approach: We are implementing JWT because it defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed.

In authentication, when the user successfully logs in using their credentials, a JSON Web Token will be returned and must be saved locally (typically in local storage, but cookies can be also used), instead of the traditional approach of creating a session in the server and returning a cookie.

This is a stateless authentication mechanism as the user state is never saved in server memory. The server’s protected routes will check for a valid JWT in the Authorization header, and if it’s present, the user will be allowed to access protected resources. As JWTs are self-contained, all the necessary information is there, reducing the need to query the database multiple times.

**MyUserDetails**

Implements UserDetailsService in order to define our own custom loadUserbyUsername function. The UserDetailsService interface is used to retrieve user-related data. It has one method named loadUserByUsername() which finds a user entity based on the username and can be overridden to customize the process of finding the user.

It is used by the DaoAuthenticationProvider to load details about the user during authentication.

The payload contains all the required information about the user, avoiding the need to query the database more than once.

Spring 4 Security Framework provides the following Advantages:

1. Open Source Security Framework
2. Flexible, Easy to Develop and Unit Test the applications
3. Declarative Security Programming
4. Easy of Extendability
5. Easy of Maintenance
6. Takes full advantage of Spring DI([Dependency Injection](https://www.journaldev.com/2394/java-dependency-injection-design-pattern-example-tutorial)) and AOP.
7. We can develop Loosely-Coupled Applications.
8. Data Marshalling/Unmarshalling ways:
9. JSON
10. XML

Approach: We are using JSON Data because is it light weighted, very easy to read, supports array and it is a way of representing objects.

1. Technology Version : Java 1.8 and spring boot 2.x

**Architectural Workflow:**

Generating JWT:



Validating JWT:

