Trade Master/Car Quest II

Problem Statement:

Problem | Submissions | Solutions | Doubts

✅ **CarQuest II**

Easy • Score **80/80**   Spring Hibernate

Problem statement

We have a third-party application called **Review** which helps applications like CarQuest to save or fetch user reviews. You need to make the Review application secure by doing the following tasks.

▼ Tasks:-

1. Implement **JWT** authentication by creating a new package named **jwt** and creating the following classes:

JwtAuthenticationHelper class

a. Add the following attributes:

- String secret (initialise with a string)
- long JWT_TOKEN_VALIDITY (initialise it to make token validity of 1 hour)

b. Create the following methods:

- String getUsernameFromToken(String token): This method returns the user name from the token.
- Claim getClaimsFromToken(String token): This method returns a claim from the token.
- Boolean isTokenExpired(String token): This method returns true if the token is expired.
- String generateToken(UserDetails userDetails): This method generates jwt token.

▼ Reference image

Problem | Submissions | Solutions | Doubts

▼ Reference image


```
@Component
public class JwtAuthenticationHelper {

    public String getUsernameFromToken(String token){return null;}

    public Claims getClaimsFromToken(String token){return null;}

    public Boolean isTokenExpired(String token){return null;}

    public String generateToken(UserDetails userDetails){return null;}

}
```
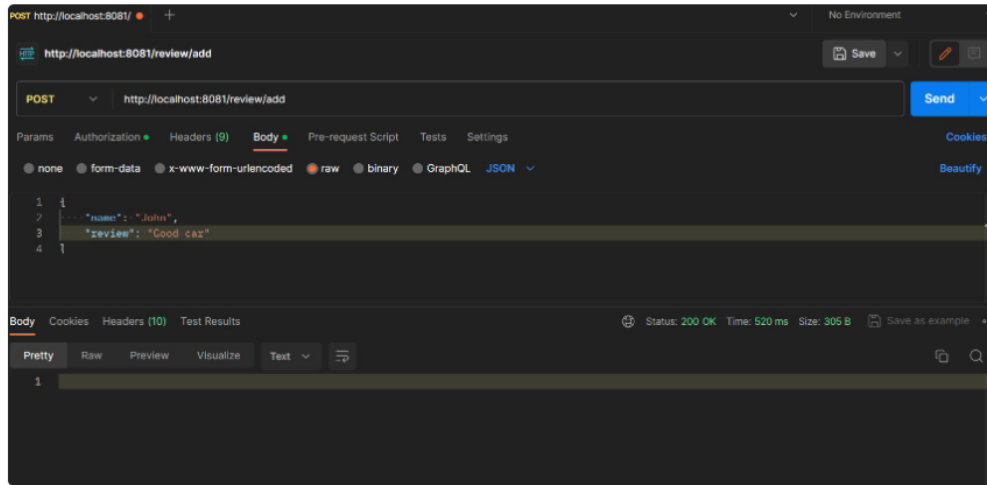
JwtAuthenticationFilter class

- Autowire the JwtAuthenticationHelper and UserDetailService.
- Make the class extend OncePerRequestFilter and override the doFilterInternal method.

2. Make changes in the **ReviewSecurityConfig** to apply **JwtAuthenticationFilter** for every incoming request.

2. Make changes in the **ReviewSecurityConfig** to apply **JwtAuthenticationFilter** for every incoming request.

3. Test the application using Postman.

▼ Reference image

▶ Reference image