



PharmaLink II

Problem Statement:

 Classroom

Password Encoding & Security Practices
[PharmaLink II](#) ▾

Problem Submissions Solutions Doubts

 PharmaLink II

Easy • Score 80/80 • Spring Hibernate

Problem statement [Send](#)

We have a third-party application called **StoreFinder** which helps applications like PharmaLink to find medical stores nearby. You need to make the StoreFinder 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

JwtAuthenticationFilter class

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

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

3. Test the application using Postman.

▼ Reference image

Output:

The screenshot shows a REST client interface with a POST request to `http://localhost:8081/store/create`. The request body is a JSON object representing a store. The response status is 200 OK, with a time of 649 ms and a size of 506 B. The response body is a JSON object representing the created store.

```
POST http://localhost:8081/store/create
```

```
{
  "name": "Apollo Medical Store",
  "contact": 12344523,
  "area": "Indore",
  "xcoordinate": "5423",
  "ycoordinate": "104234",
  "medicines": [
    "Aspirin", "Paracetamol", "Crosin"
  ]
}
```

Status: 200 OK Time: 649 ms Size: 506 B

```
{
  "id": 1,
  "name": "Apollo Medical Store",
  "contact": 12344523,
  "area": "Indore",
  "medicines": [
    "Aspirin",
    "Paracetamol",
    "Crosin"
  ],
  "xcoordinate": 5423,
  "ycoordinate": 104234
}
```

The screenshot shows a REST client interface with a GET request to `http://localhost:8081/store/1`. The response status is 200 OK, with a time of 37 ms and a size of 506 B. The response body is a JSON object representing the store with id 1.

```
GET http://localhost:8081/store/1
```

This request does not have a body

Status: 200 OK 37 ms 506 B

```
{
  "id": 1,
  "name": "Apollo Medical Store",
  "contact": 12344523,
  "area": "Indore",
  "medicines": [
    "Aspirin",
    "Paracetamol",
    "Crosin"
  ],
  "xcoordinate": 5423,
  "ycoordinate": 104234
}
```