

# PharmaLink III

## Problem Statement:

← Classroom

Password Encoding & Security Practices  
PharmaLink III

Problem Submissions Solutions Doubts

PharmaLink III

Easy • Score 80/80 Spring Hibernate

**Problem statement**

In continuation of the **PharmaLink** application, you must implement the restTemplate communication between **PharmaLink** and **StoreFinder** Services.

▼ **Tasks:-**

1. Create a new package, "**communicator**", and create a new class named "**StoreFinderCommunicator**" inside it.
2. Inject restTemplate inside the StoreFinderCommunicator class.
3. Create the following APIs in the **UserController** of PharmaLink Service:
  - a. GET: "/user/getNearestStores/{userId}/{distance}/{token}": This API communicates to the getNearestMedicalStores(userId, distance, token) API in the StoreFinder Service using RestTemplate to get the medical stores within a radius of distance given. The three variables userId, distance, and token are PathVariables.
  - b. GET: "/user/getStoresWithMedicine/{medicine}/{token}": This API communicates to the getMedicalStoresHavingMedicine(medicine, token) API in the StoreFinder Service using RestTemplate to get medical stores having a particular medicine. The two variables, medicine and token, are PathVariables.

▼ Reference image

```
@GetMapping("")
@ResponseStatus(HttpStatus.OK)
public List<MedicalStoreDto> getNearestStores(@PathVariable Long userId, @PathVariable Long distance, @PathVariable String token){
    return null;
}

@GetMapping("")
@ResponseStatus(HttpStatus.OK)
public List<MedicalStoreDto> getStoresWithMedicine(@PathVariable String medicine, @PathVariable String token){
    return null;
}
```

4. Create two rest template calls to the StoreFinder Service for the APIs created in the step above through the StoreFinderCommunicator:
  - a. public List getNearestMedicalStores(Long userId, Long distance, String jwtToken)
  - b. public List getMedicalStoresWithMedicine(String medicine, String jwtToken)

▼ Reference image

```
public List<MedicalStoreDto> getNearestMedicalStores(Long userId, Long distance, String jwtToken) {
    return null;
}

public List<MedicalStoreDto> getMedicalStoresWithMedicine(String medicine, String jwtToken) {
    return null;
}
```

5. Implement the below methods that are defined for sending requests to the communicator layer for which you will need to auto-wire the **StoreFinderCommunicator** object in the **UserService**.
  - getNearestMedicalStores(Long userId, Long distance, String token)
  - getMedicalStoresWithMedicine(String medicine, String token)

5. Implement the below methods that are defined for sending requests to the communicator layer for which you will need to auto-wire the **StoreFinderCommunicator** object in the **UserService**.

- `getNearestMedicalStores(Long userId, Long distance, String token)`
- `getMedicalStoresWithMedicine(String medicine, String token)`

## Output:

The screenshot shows a Postman interface for a POST request to `localhost:8080/auth/login`. The request body is a JSON object:

```
{  "username": "John",  "password": "John@123"}
```

The response status is 200 OK. The response body is a JSON object containing a JWT token:

```
{  "jwtToken": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJKb2h1IiwiaWF0IjoxNzAyOTc3NDY5LCJleHAiOjE3MDI5ODUwNj19.qX0y3v_NuWajH5K00JG2tuK4S55BgPL1901zKJf5"}
```

The screenshot shows a Postman interface for a GET request to `localhost:8080/user/getStoresWithMedicine/Crosin Advanced`. The request includes a Bearer Token in the Authorization header. The response status is 200 OK. The response body is a JSON object containing store details and a list of medicines:

```
{  "id": 12,  "name": "Balaji Medical Store",  "contact": 12344523,  "area": "Ujjain",  "medicines": [    "Aspirin",    "Crosin Advanced",    "Crosin"  ],  "ycoordinate": 32,  "xcoordinate": 10}
```

HomeWorkspacesAPI Network

Search Postman

InviteSettingsPostman Upgrade

GETGet nearest stores rest1

+

No Environment

Save

Send

PharmaLinkGet nearest stores restTemplate

GETlocalhost:8080/user/getNearestStores/9/10000000/eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJkb2huliwWF0lloxNzAyOTc3NiYzLCJleHAiOiJlMDI5ODEyNjN9ypAY-D4xqxGRPmHpr...

AuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

TypeBearer Token

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables

The authorization header will be automatically generated when you send the request. Learn more about Bearer Token authorization

TokeneyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJkb2huliwWF0lloxNzAyOTc3NiYzLCJleHAiOiJlMDI5ODEyNjN9ypAY-D4xqxGRPmHpr...

BodyCookiesHeaders (12)Test Results

Status: 200 OKTime: 78 msSize: 709 BSave as example

PrettyRawPreviewVisualizeJSON

```
3    "id": 10,
4    "name": "Apollo Medical Store",
5    "contact": 12344523,
6    "area": "Indore",
7    "medicines": [
8      "Aspirin",
9      "Paracetamol",
10     "Crosin"
11  ],
12  "ycoordinate": 104234,
13  "xcoordinate": 5423
14 },
15 {
16   "id": 11,
17   "name": "Balaji Medical Store",
18   "contact": 12344523,
19   "area": "Ujjain",
20   "medicines": [
```

OnlineConsole

PostbotRunnerDesktop AgentCookiesTrash