

Hospital Front Desk

Problem Statement:

A Patient/caller calls a Hospital front desk to enquire about the details of a specialist, to schedule appointments with a specialist, and to enquire about the admission (based on number of Beds Available through an online system). As a software consultant/developer, build an application using Restful Web services with Spring Boot Core to ease this process.

Requirement 1 - Retrieve Specialist Details

The patient/caller calls a Hospital Front Desk to enquire about the list of specialists using the hospital name and the types of specialist. The caller should be provided with the specialists' details in a two different response format: XML and JSON format.

Also improve the performance of these calls by maintain the specialist details in cache.

Rules:

1. Use the appropriate annotations to implement the solution.
2. The POJO class should be annotated to return both JSON and XML response.
3. Enable the caching feature using annotations.
4. Throw business exceptions for the invalid input hospital name, no specialist details found for the specified hospital name.
5. Use Controller Advice Annotation and Exception Handler Annotation to handle the exceptions across all requests to the Controller.
6. Frame the request URI only from **application.properties** ,**specialist.properties**

Input Format:

Method should have the Hospital Name, Specialist Type arguments

Output Format

XML Response Format should be as below

```
<List>
  <item>
    <type>Dentist</type>
    <name>sandhya</name>
    <availableday>Monday,Wednesday</availableday>
    <availableTime>5 to 6</availableTime>
    <isAvailable>Y</isAvailable>
    <hospitalId>946</hospitalId>
```

```
</item>
<item>
  <type>Dentist</type>
  <name>saranya</name>
  <availableday>Monday,Thursday</availableday>
  <availableTime>6 to 8</availableTime>
  <isAvailable>N</isAvailable>
  <hospitalId>946</hospitalId>
</item>
</List>
```

JSON Format :

```
[
  {
    "type": "Dentist",
    "name": "sandhya",
    "availableday": "Monday,Wednesday",
    "availableTime": "5 to 6",
    "isAvailable": "Y",
    "hospitalId": 946
  },
  {
    "type": "Dentist",
    "name": "saranya",
    "availableday": "Monday,Thursday",
    "availableTime": "6 to 8",
    "isAvailable": "N",
    "hospitalId": 946
  }
]
```

]

Requirement 2– Return a New Appointment

The patient/caller calls a hospital's front desk to enquire about the availability of a specialist on a specific day, if the specialist is available; Front Desk schedules an appointment for the caller.

Rules:

1. There should be an external property file 'specialist.properties' which should contains the details like the specialist's name, available day, available time.
2. Read the specialist's values from the specialist property file using Type Safe Configuration and create a new appointment containing the appointment day, appointment time, specialist's name, and patient name.
3. Throw business exceptions for the invalid input hospital name, no specialist details found for the specified hospital name.
4. Use the controller Advice Annotation and Exception Handler Annotation to handle the exceptions across all requests to the Controller.

Input Format:

Method should have the Specialist Name, Appointment Day, Patient Name arguments

Output Format

1. Appointment details should be returned to the caller.
2. Response should be in below JSON format.

```
{ "specialistName": "Edward",  
  "patientName": "suganthi",  
  "appointmentDay": "Monday",  
  "appointmentTime": "5 to 8"  
}
```

Requirement 3:– Return Number of beds available in the hospital for Admission

Front desk gets the details of the patients who are being discharged and retrieves the vacancy count to have details of number of beds available for admission.

Rules:

- 1) Get the hospital details for input hospital name (Use local methods to set the hospital details)
- 2) Count the number of beds available for admission from the fetched hospital details and compare the status as **"DISCHARGE"** against each **PatientStatus**
- 3) Return number of beds with message **"Number of Beds Available is = "**
- 4) Throw the business exception for any invalid **hospital name** and also if beds are not available for **Admission**.

Input Format:

Method should have the hospital Name as argument.

Output Format

Number of Beds Available is = 8

Requirement 4- Writing REST Client

Write a REST client using Rest Templates to test the Requirement 1.

Rules:

- 1) Write a REST client to test the Requirement 1 for both XML and JSON response
- 2) Implement this using Rest Template
- 3) Load URL details from property file from **application.properties**
- 4) Use Rest Templates to implement the REST Client to fetch List of **Specialist** details in XML and JSON Format
- 5) Use controller **Advice Annotation** and **Exception Handler Annotation** to handle the exceptions across all requests to the Controller

Input Format:

Method should have the port, environment, type, url, hospital Id, specialist type as argument.

Output Format

ResponseEntity of Specialist Details

Requirement 5 – Enabling REST Actuator Endpoints

Configure Spring Boot Actuator in the application to enable Actuator Endpoints such as health, info, beans and metrics.

Output Format

When health URL is accessed below output should be displayed.

{"status":"UP"}