Week 08 Questions

Theory

1. Discuss the rules to follow for an API to the RESTful.
2. Compare Spring And Spring Boot
3. Demonstrate with a simple code to secure REST APIs with Spring Security.
4. Discuss How REST API is best over SOAP API architecture.
5. Implement Inversion Of Control(IOC) container in Spring Boot With an Example.

Lab Questions

1. RestFull API supports CRUD operations to Objects. Demonstrate CRUD operations with Object.
2. DemonstrateCRUD operations using H2 database.
3. DemonstrateCRUD operations using MySQL.
4. DemonstrateCRUD operations using MongoDB.

Week 8 Theory Questions Answer

1. Discuss the rules to follow for an API to the RESTful.

For an API to be RESTful there are six rules that it needs to follow.The rules are as follows

1. Uniform interface
2. Client–server
3. Stateless
4. Cacheable
5. Layered system
6. Code on demand

1. Uniform Interface

The API should facilitate communication between the client and server as they exchangedata. To efficiently exchange data, we need a uniform interface.

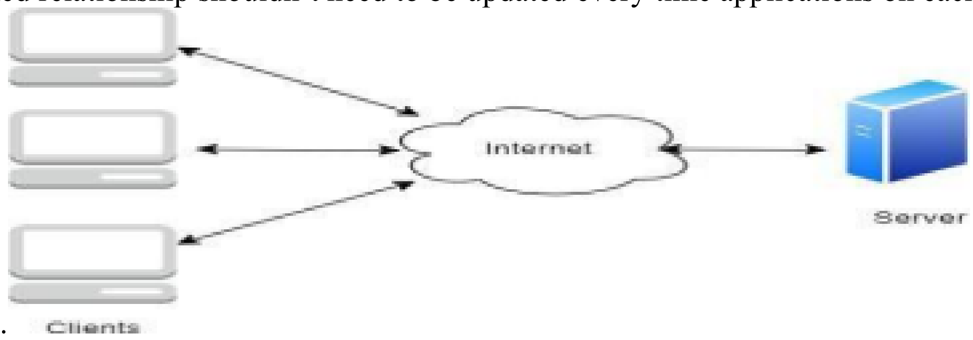
If our system is using well known protocols and techniques, it’s easily implemented. Datashould be

exchanged using standard formats JSON or XML.

2. Client-server architecture

The main purchase of an API is to connect two pieces of software – software might be custombuilt and run, off the shelf, or a Software as a Service. The client makes requests and the server gives responses – it's important that they stay separate and independent.

Well-designed relationship shouldn't need to be updated every time applications on each end change.



3. Stateless

It's important that each endpoint in the API be stateless which means each call must behandled independently and have no knowledge of what happened from other calls.

A stateless API means that the server receives everything from a client that they need toidentify them and what they want in each request.

The major advantages of a stateless API are:
They can handle more clients because less resources are used, and each request isindependent of previous ones.

4. Cacheable

APIs can have a lot of overhead when they process requests – making repeated requests for data that rarely changes or for the exact same data doesn't normally make sense.

A cache allows us to temporarily store data locally for a agreed upon period of time. So essentially, if the client goes to make the call again and the agreed upon time hasn't beenfully spent it will use the stored version.

5. Layered System

REST allows us to build a layered system architecture meaning that multiple servers may potentially respond to a request. A client shouldn't be able to easily tell what system is responding to their request especially if it's behind an API Gateway.

6. Code on Demand

Code-on-Demand (COD) is the only optional constraint in REST. It allows clients to improve its flexibility because, in fact, it is the server who decides how certain things willbe done. For example, client can download a javascript, java applet or even a flash application in order to encrypt communication so servers are not aware of any encryptionroutines / keys used in this process.

2. Compare Spring And Spring Boot

Spring	Spring Boot
Spring Framework is a widely used Java EE framework for building applications.	Spring Boot Framework is widely used to develop REST APIs.
It simplify Java EE development that makes developers more productive.	It shorten the code length and provide the easiest way to develop Web Applications.
The primary feature of the Spring Frameworkis dependency injection.	The primary feature of Spring Boot is Autoconfiguration. It automatically configuresthe classes based on the requirement.
	levelop loosely coupled applications. It helps to tand-alone application with less configuration.
The developer writes a lot of code (boilerplate code) to do the minimal task.	It reduces boilerplate code.
To test the Spring project, we need to set up the sever explicitly.	Spring Boot offers embedded server such as Jetty and Tomcat, etc.
It does not provide support for an in-memory database.	It offers several plugins for working with an embedded and in-memory database such as H2.

Developers manually define dependencies forthe Spring project in pom.xml.	Spring Boot use the concept of starter in pom.xml file that internally takes care of downloading the dependencies JARs based on Spring Boot Requirement.

3. Demonstrate with a simple code to secure REST APIs with Spring Security.

Spring Security is an application-level security framework which provides various security features like: authentication, authorization to create secure Java Enterprise Applications. To enable application-level security feature, add spring security dependency to the pom.xml or you can add while creating a maven spring boot application using spring initilizr. Spring security make sure that each and every API written in the controller to be authenticated.

The framework targets two major areas of application are authentication and authorization.

• Authentication is the process of knowing and identifying the user that wants to access. • Authorization is the process to allow authority to perform actions in the application. The application-level framework features provide Login and logout functionality. It allow/block access to URLs to logged in users. It also allow/block access to URLs to logged in users AND with certain roles.

• pox.xml code for spring security dependency

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

• SecurityController.java class

```
        @RestController
        public class SecurityController {
        @GetMapping("/")
        public String Welcome() {
        return ("<h1>Welcome to SpringBoot Security</h1>");
        } }
```

A default system generated security password is given by the spring boot security framework. The password is dynamic and changes every time we execute the application. Try to access the REST API using the URI mapping that is http://localhost:8080/ in any browser or application like postman or swagger. As the URI hits in the browser a default login form authentication is enabled. By default, User is created as username for authentication. Spring security also provide logout form and error handling / validation.

We can make the password static by setting its value in the application.properties file. Application.properties file is created under src/main/resource package. We can also create static user and static password by setting its value in the application.properties file as shown below.

```
spring.security.user.name=ABC
spring.security.user.password=XYZ
```

4. Discuss How REST API is best over SOAP API architecture.

There is no direct comparison between SOAP and REST APIs. But there are some points to be listed below which makes you choose better between these two web services. Here are:

- SOAP stands for **S**imple **O**bject **A**ccess **P**rotocol and REST stands for **R**epresentational **S**tate **T**ransfer.
- Since SOAP is a protocol, it follows a strict standard to allow communication between the client and the server whereas REST is an architectural style that doesn’t follow any strict standard but follows six constraints defined by Roy Fielding in 2000. Those constraints are – Uniform Interface, Client-Server, Stateless, Cacheable, Layered System, Code on Demand.
- SOAP uses only XML for exchanging information in its message format whereas REST is not restricted to XML and its the choice of implementer which Media-Type to use like XML, JSON, Plain-text. Moreover, REST can use SOAP protocol but SOAP cannot use REST.
- On behalf of services interfaces to business logic, SOAP uses @WebService whereas REST instead of using interfaces uses URI like @Path.
- SOAP is difficult to implement and it requires more bandwidth whereas REST is easy to implement and requires less bandwidth such as smartphones.

- Benefits of SOAP over REST as SOAP has ACID compliance transaction. Some of the applications require transaction ability which is accepted by SOAP whereas REST lacks in it.
- On the basis of Security, SOAP has SSL(**S**ecure **S**ocket **L**ayer) and WS-security whereas REST has SSL and HTTPS. In the case of Bank Account Password, Card Number, etc. SOAP is preferred over REST. The security issue is all about your application requirement, you have to build security on your own. It's about what type of protocol you use.
- SOAP cannot make use of REST since SOAP is a protocol without any architectural pattern. REST can make use of SOAP because it is an architectural pattern having protocol.

Differentiating between SOAP API and REST API

SOAP API	**REST API**
Relies on SOAP (Simple Object Access Protocol)	Relies on REST (Representational State Transfer) architecture using HTTP.

Transports data in standard XML format.	Generally transports data in JSON. It is based on URI. Because REST follows stateless model, REST does not enforces message format as XML or JSON etc.
Because it is XML based and relies on SOAP, it works with WSDL	It works with GET, POST, PUT, DELETE
Works over HTTP, HTTPS, SMTP, XMPP	Works over HTTP and HTTPS
Highly structured/typed	Less structured -> less bulky data
Designed with large enterprise applications in mind	Designed with mobile devices in mind

5. Implement Inversion Of Control(IOC) container in Spring Boot With an Exampl*e*.

Spring Application .java

```
ApplicationContext context=SpringApplication.run(SpringApplication.class,args);
Dev obj=context.getBean(Dev.class);
obj.build();
```

Dev.java

```
@Component
Class Dev{

                build(){

        S.O.P("Building Projct");
                }
}
```