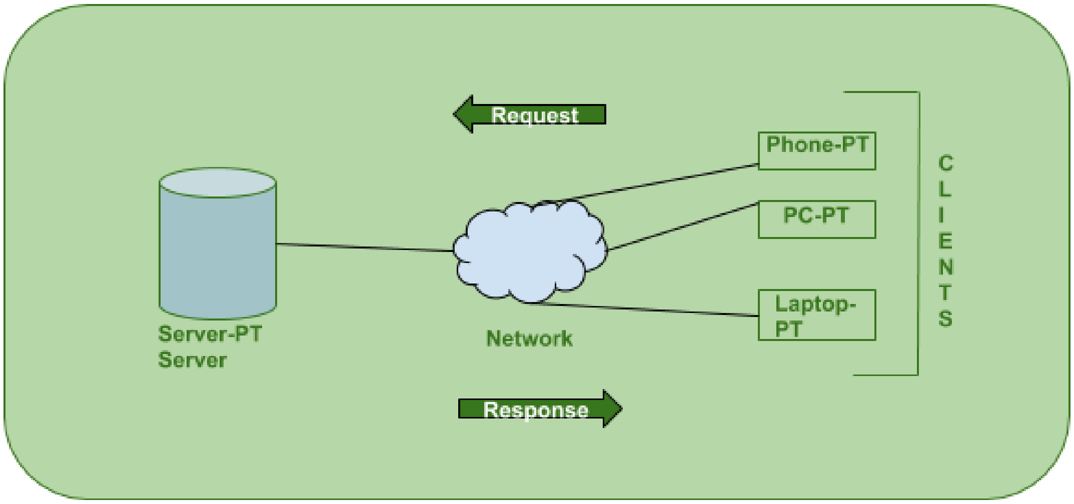
Contents

- Introduction to Server and Client Model
- Introduction to Spring Boot
- How to run application as a Server
- What are Embedded Servers (Jetty , Tomcat)
- Logging Levels In Spring Boot
- Running Spring Boot applications in different spring profiles
- Running Spring Boot applications on terminal

INTRODUCTION TO SERVER AND CLIENT MODEL : The Client-server model is a distributed application structure that partitions task or workload between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client. Clients do not share any of their resources. Examples of Client-Server Model are Email, World Wide Web, etc.



Challenge 1 : What is an application server ?
When an application runs as a server , it becomes an entity which runs continuously which you can make requests to get some output out of it . eg : amazon.in

Challenge 2 : When we type amazon.in in the url window of Google , does this request hit Amazon server or Google server ?

Challenge 3 : Let's explore Google Chrome Developer Tools and inspect the requests in the Network tab .
HINT : When we search MS Dhone in google.in . Basically the frontend is converting your search into a GET API request .The server returns a response inform of json object . This json response from the backend gets parsed on the frontend .

Challenge 4 : What is an API and why Rest api is called Idempotent while SOAP is not ?
HINT : API is just cient b/w client and server to communicate and get a response .
https://www.geeksforgeeks.org/application-programming-interfaces-api-and-its-types/

Challenge 5 : What are the components of an API ?
HINT : https://www.bigcommerce.com/blog/what-is-an-api/#what-about-rest-soap-apis

Challenge 6 : What is the meaning of HTTP request ? Is rest api request not HTTP request ?
HINT : HTTP is a protocol . HTTP request is a way to communicate with the server following the rules of HTTP protocol . This request is stateless . REST refers to a set of attributes of a particular architectural style, while HTTP is a well-defined protocol that happens to exhibit many features of a RESTful system.

Challenge 7 : What is the difference b/w web API and web service ?
HINT : https://www.geeksforgeeks.org/differences-between-web-services-and-web-api/
https://stackoverflow.com/questions/808421/api-vs-webservice

Challenge 8 : Find the ip address of google.com .
HINT : host google.com
Now put the ip address of google.com in the url bar and press enter .
This ip address is not of google server , it is of load balancer . (distributor which routes the request to an application server)

Challenge 9 : What is scaling and its types ?
HINT : https://www.cloudzero.com/blog/horizontal-vs-vertical-scaling

Challenge 10 : What is load balancer and how is it helpful in horizontal scaling ?
HINT : https://www.geeksforgeeks.org/load-balancer-system-design-interview-question/

Challenge 11 : What means when we say Restful API is stateless ?
HINT : response will be the same irrespective of how many times we make the request .

References :

https://www.geeksforgeeks.org/client-server-model/

https://medium.com/@maneesha.wijesinghe1/what-happens-when-you-type-an-url-in-the-browser-and-press-enter-bb0aa2449c1a

https://www.geeksforgeeks.org/introduction-to-apis/

https://stackoverflow.com/questions/19336347/what-is-the-difference-between-a-web-api-and-a-web-service

https://blog.uptrends.com/technology/the-anatomy-of-an-api-call/

https://blog.yellowant.com/rest-api-calls-made-easy-7e620e4d3e82

https://www.geeksforgeeks.org/proxy-server/

https://www.nginx.com/resources/glossary/reverse-proxy-vs-load-balancer

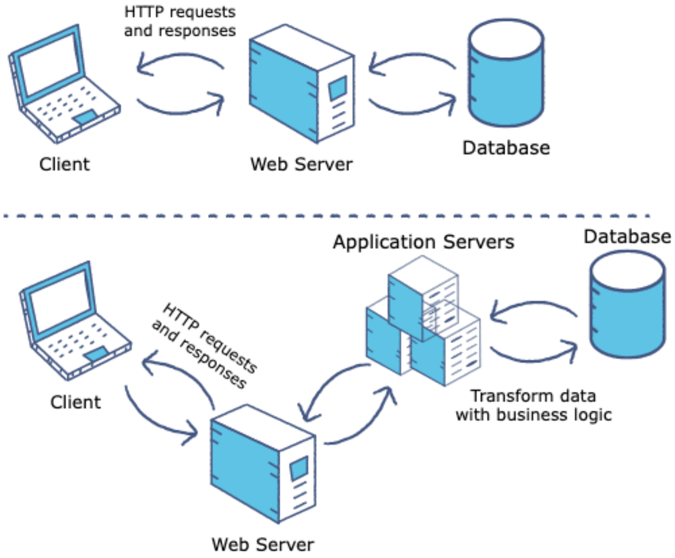
https://www.geeksforgeeks.org/state-the-core-components-of-an-http-response/

https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/

INTRODUCTION TO SPRING BOOT : Spring is widely used for creating scalable applications. For web applications Spring provides Spring MVC which is a widely used module of spring which is used to create scalable web applications. But the main disadvantage of spring projects is that configuration is really time-consuming and can be a bit overwhelming for the new developers. Making the application production-ready takes some time if you are new to the spring. Solution to this is Spring Boot. Spring Boot is built on the top of the spring and contains all the features of spring.

Challenge 1 : What is an embedded server ?
HINT : It gives you low level code that helps to start the server . An embedded server is embedded as part of the deployable application. If we talk about Java applications, that would be a JAR. The advantage with this is you don't need the server pre-installed in the deployment environment. With SpringBoot, the default embedded server is Tomcat (built by Apache) . Other options available are Jetty(built by Eclipse) and UnderTow(built by Redhat) .

Challenge 2 : What is the difference b/w Web server and Application server ?
HINT : For static websites we call it a web server and for dynamic websites ,we have an application server .
https://www.geeksforgeeks.org/difference-between-web-server-and-application-server/

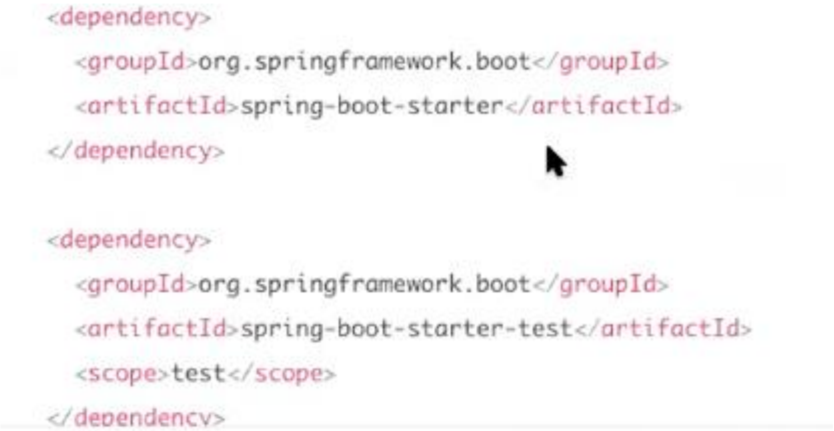


Challenge 3 : What is a database server ?
HINT : Application server uses database server to persist data . App server has code logic running and does not have much space to store all data , thus , we need the Database server .

Challenge 4 : What is a virtual machine ?
HINT : https://azure.microsoft.com/en-in/overview/what-is-a-virtual-machine/

Challenge 5 : Let's make our first spring boot project with maven using start.spring.io . (Don't add any dependencies yet)

Challenge 6 : What are the two dependencies that are added to the POM.xml by default by start.spring.io and why ?
HINT : spring-boot-starter is needed to simply run your Spring Boot Application . (we are not even talking about running as server)



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

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

Challenge 7 : Which dependency to add to POM.xml to allow our spring boot application to run as a server ?
HINT : spring-boot-starter-web (note : The moment we add this to POM.xml and load changes , spring-boot-starter dependency disappears . Why : because spring-boot-starter-web has spring-boot-starter as a parent dependency so no need to have it separately)

Challenge 8 : Now how to load our downloaded spring boot project into IntelliJ IDE ?

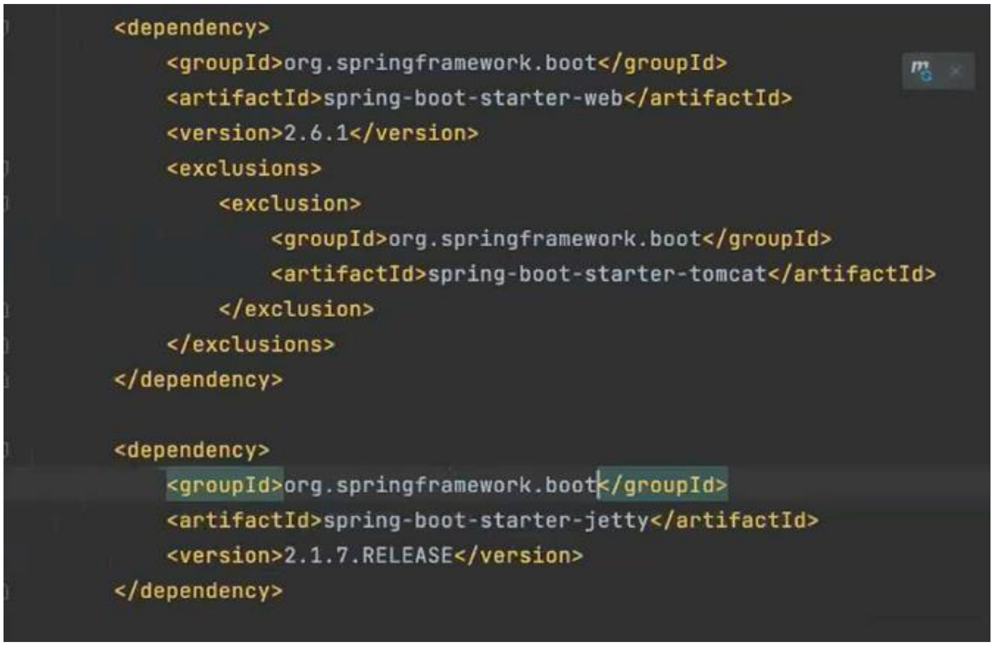
Challenge 9 : Let's explore the POM of our project and check the parent POM of each dependency by clicking on artifact id .

Challenge 10 : Let's explore the structure of our spring boot project and External Libraries .

Challenge 11 : Run the application . Now lets see the difference between spring-boot-starter and spring-boot-starter-web by replacing spring-boot-starter-web from our POM.xml with spring-boot-starter . Run the application once again after loading the changes .
HINT : On exploration of the parent POM of spring-boot-starter-web , we can see that it has an embedded server TOMCAT as a parent dependency which allows the spring boot application to run as a server .

Challenge 12 : Now add spring-boot-starter-web back again and remove spring-boot-starter from our POM . Run the application . This time ,let us read the logs and observe together .

Challenge 13 : Is it possible to use any other embedded servers for running our spring boot application as a server ? How ?
HINT : Once added and changes are loaded , we can see Jetty in the External Libraries. Run.



```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <version>2.6.1</version>
    <exclusions>
        <exclusion>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-tomcat</artifactId>
        </exclusion>
    </exclusions>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jetty</artifactId>
    <version>2.1.7.RELEASE</version>
</dependency>
```

Challenge 14 : What is a port ? Which port does TOMCAT run on ?
HINT : https://www.geeksforgeeks.org/tcp-ip-ports-and-its-applications/
Supposedly you own a building. The street address corresponds to the IP address. Also, the way there is a port number, each building also has a building number.

Challenge 15 : What will happen if we don't exclude Tomcat and also add Jetty as a dependency in our POM.xml . Now which port will be used and which server will run the spring boot application ?
HINT : As Tomcat is default embedded server and if it's present , other embedded server dependencies become redundant .

Challenge 16 : (HomeWork) Explore the class ServletWebserverApplicationContext by doing Ctrl+Shift+F and click on Scope .
HINT : This is the class responsible for loading the web server into application context .

Challenge 17 : How to change the port on which your tomcat application server runs ?
HINT : server.port = 8081 (application.properties)
Search for server.port by Ctrl + Shift + F and look in scope to check where the default value 8080 is defined for it . While Spring Boot is able to override its default values in autoconfigure jar by using application.properties .This functionality is missing in Spring .

Challenge 18 : (HomeWork) If we exclude Tomcat from our project and add Jetty and Undertow dependencies in POM.xml in no specific order . When we run the application , which server will run ? Why ?
HINT : Jetty will run in every case . Order does not matter .

Challenge 19 : If we have all three embedded servers in our POM.xml . Which server will the application run on ?
HINT : Tomcat is default .

Challenge 20 : (HomeWork) Which embedded server takes the least time to start among Jetty , Tomcat and Undertow ?

References :
https://www.geeksforgeeks.org/difference-between-spring-and-spring-boot/ IMP

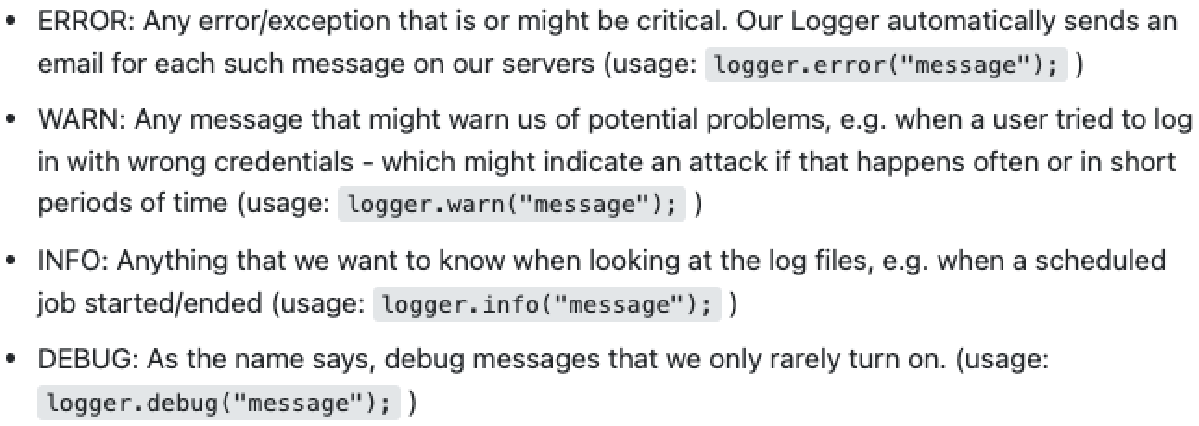
https://www.geeksforgeeks.org/introduction-to-spring-boot/ IMP

https://www.educative.io/edpresso/web-server-vs-application-server

https://www.baeldung.com/java-jar-war-packaging IMP

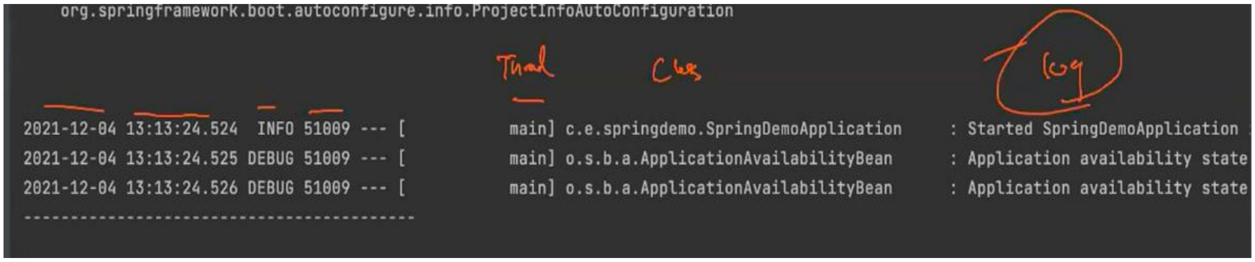
https://stackoverflow.com/questions/39632667/how-do-i-kill-the-process-currently-using-a-port-on-localhost-in-windows

LOGGING LEVELS IN SPRING BOOT : A good logging infrastructure is necessary for any software project as it not only helps in understanding what's going on with the application but also to trace any unusual incident or error present in the project.



- ERROR: Any error/exception that is or might be critical. Our Logger automatically sends an email for each such message on our servers (usage: `logger.error("message");`)
- WARN: Any message that might warn us of potential problems, e.g. when a user tried to log in with wrong credentials - which might indicate an attack if that happens often or in short periods of time (usage: `logger.warn("message");`)
- INFO: Anything that we want to know when looking at the log files, e.g. when a scheduled job started/ended (usage: `logger.info("message");`)
- DEBUG: As the name says, debug messages that we only rarely turn on. (usage: `logger.debug("message");`)

Challenge 1 : Why do we need a logger when we have System.out.println() ?
HINT : Debugging (debug) , checking threads and heartbeat of servlets (Trace)



```
    org.springframework.boot.autoconfigure.info.ProjectInfoAutoConfiguration

  2021-12-04 13:13:24.524  INFO 51009 --- [           main] c.e.springdemo.SpringDemoApplication     : Started SpringDemoApplication
  2021-12-04 13:13:24.525 DEBUG 51009 --- [           main] o.s.b.a.ApplicationAvailabilityBean      : Application availability state
  2021-12-04 13:13:24.526 DEBUG 51009 --- [           main] o.s.b.a.ApplicationAvailabilityBean      : Application availability state
  ----------------------------------------
```

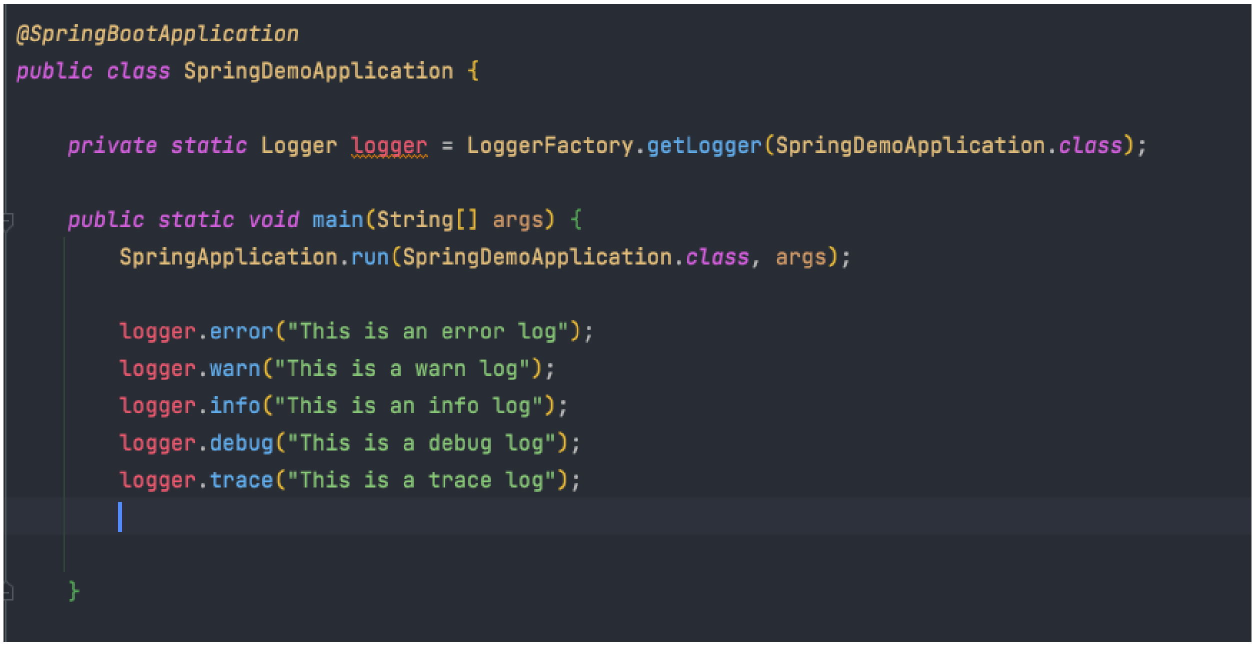
Challenge 2 : What do we mean by Logging Levels ?
HINT : In terms of severity : Error > Warning > Info > Debug > Trace

Challenge 3 : What is the by default logging level and how to change it ?
HINT : Default logging level is INFO . We can change it by defining property in the application.properties file : logging.level.<package_name> = debug
Example : logging.level.root = debug where root stands for applying the logging level to all packages in the application

Challenge 4 : If logging level is debug , what all levels of logs will be printed ?
HINT : All logging levels above and over debug in terms of severity will be printed ie Error , Warning , Info , Debug . Trace logs will not be printed .

Challenge 5 : Which logging level will have the most number of logs ?
HINT : Trace

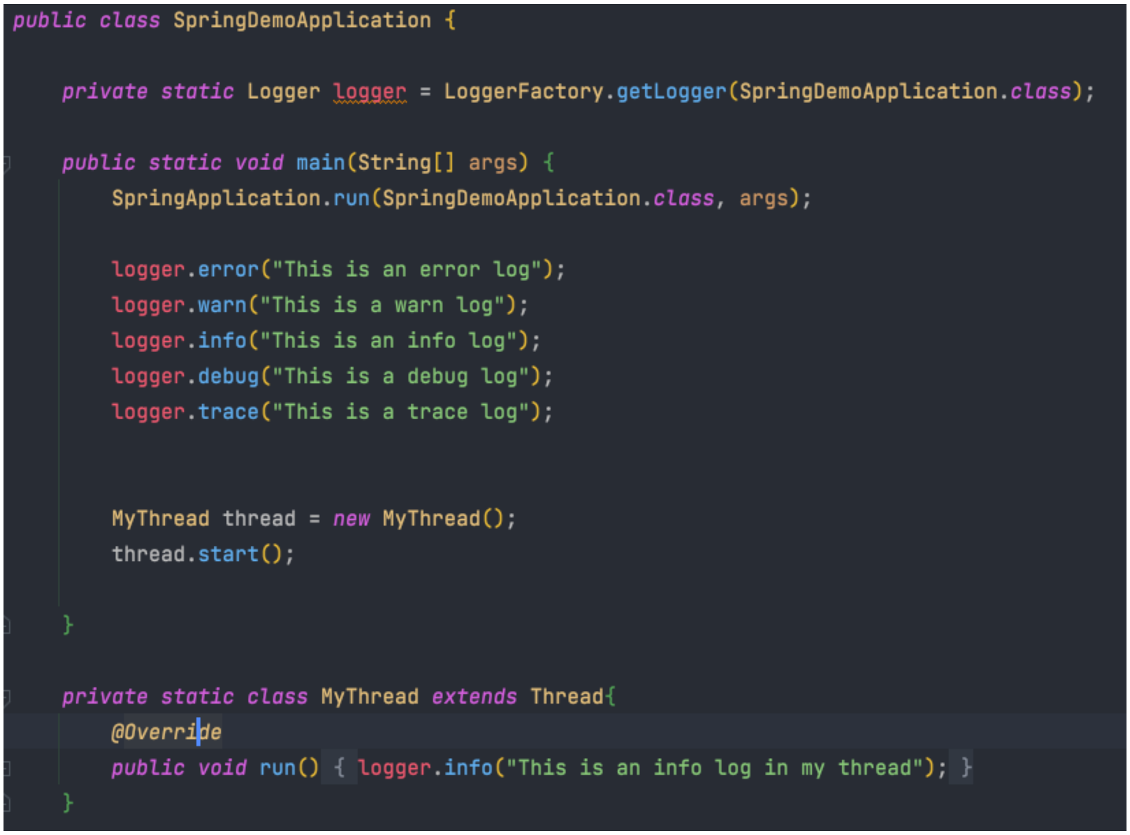
Challenge 6 : For the below code , if my logging level is error in the application.properties file . Then how many logs will be printed ?



```java
@SpringBootApplication
public class SpringDemoApplication {

    private static Logger logger = LoggerFactory.getLogger(SpringDemoApplication.class);

    public static void main(String[] args) {
        SpringApplication.run(SpringDemoApplication.class, args);

        logger.error("This is an error log");
        logger.warn("This is a warn log");
        logger.info("This is an info log");
        logger.debug("This is a debug log");
        logger.trace("This is a trace log");


    }
```

Challenge 7 : Can we log something from the run function of a thread ?

HINT :

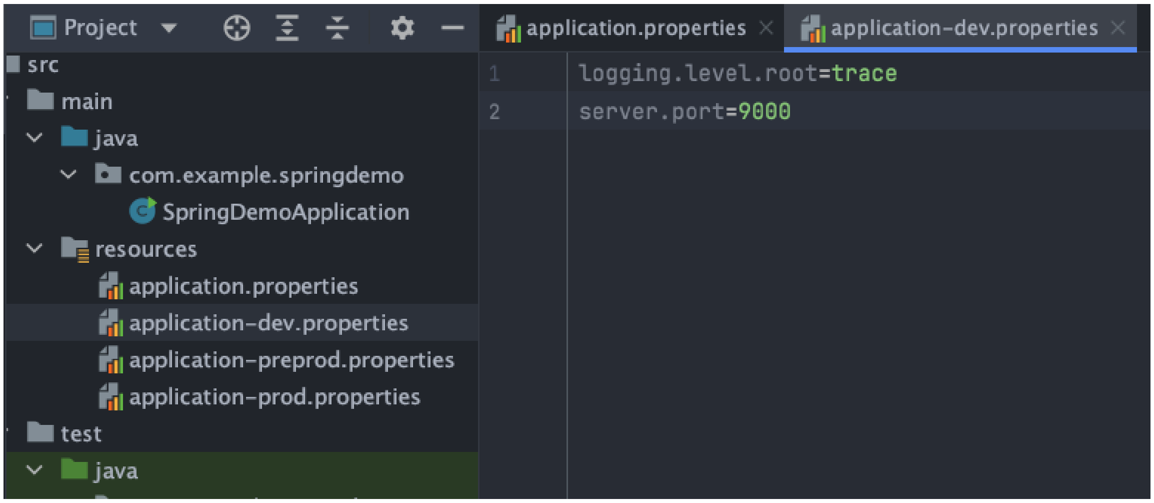


```
public class SpringDemoApplication {

    private static Logger logger = LoggerFactory.getLogger(SpringDemoApplication.class);

    public static void main(String[] args) {
        SpringApplication.run(SpringDemoApplication.class, args);

        logger.error("This is an error log");
        logger.warn("This is a warn log");
        logger.info("This is an info log");
        logger.debug("This is a debug log");
        logger.trace("This is a trace log");


        MyThread thread = new MyThread();
        thread.start();

    }

    private static class MyThread extends Thread{
        @Override
        public void run() { logger.info("This is an info log in my thread"); }
    }
```

Challenge 8 : Can we have logging level as Trace / Debug in Production ?
HINT : Cost of storing such a huge number of logs is high and memory requirements are high .Not suitable for debugging .

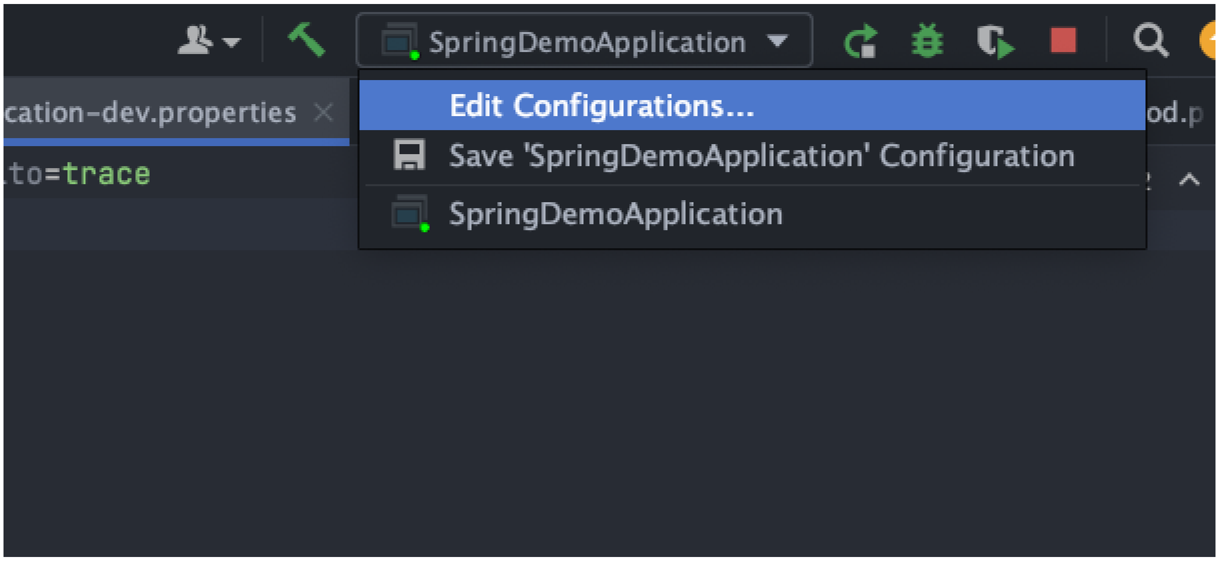
Challenge 9 : How can we specify different levels of logging for different environments (dev ,qa , sit , preprod , prod) ?
HINT : We can have a separate config file (application.properties) file for each environment .

Step 1 - Create the application-<env>.properties file and define the port number and the logging level you require when the application runs in that environment .

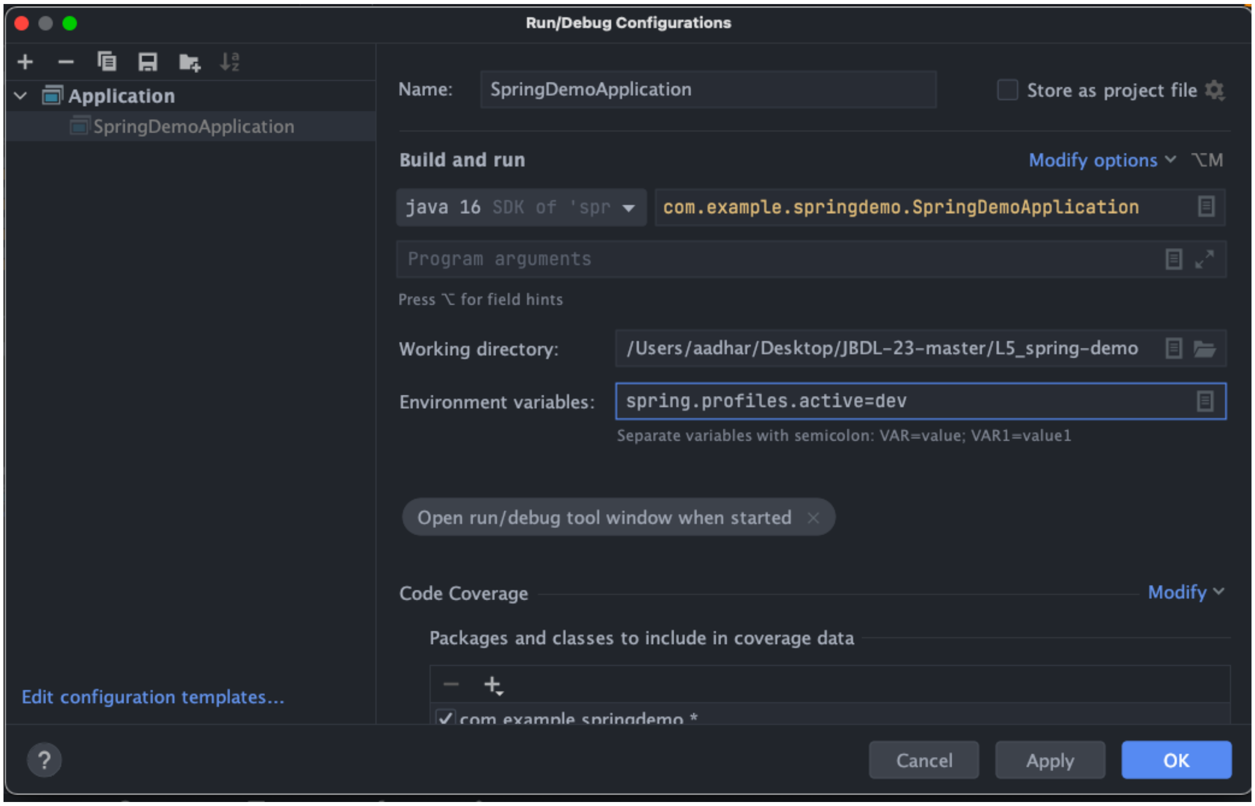


```
logging.level.root=trace
server.port=9000
```

Step 2 - Open Edit Configuration for the project as shown below and define which spring profile should the application run as out of all spring profiles we have in our application.properties



```
to=trace
```



Name: SpringDemoApplication

java 16 SDK of 'spr

com.example.springdemo.SpringDemoApplication

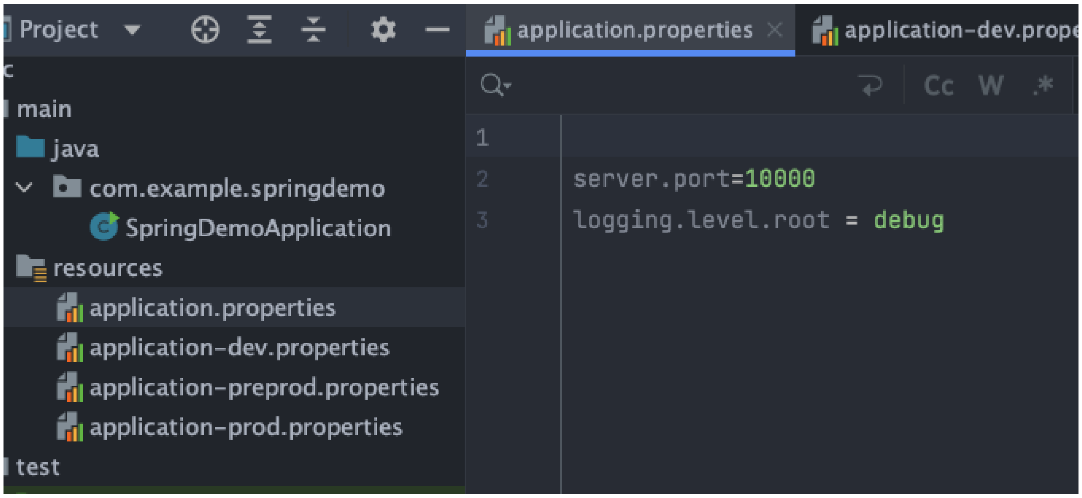
Working directory: /Users/aadhar/Desktop/JBDL-23-master/L5_spring-demo

Environment variables: spring.profiles.active=dev

☑ com.example.springdemo.*

When we run our application now , we can see that its running on dev profile and thus trace and greater severity logs are getting printed in the console . This is how we can maintain different profiles for different environments .

Challenge 10 : If I have three spring profiles i.e. have three application-<env>.properties files defined . These are for dev , prod and preprod . Now when I edit configurations , I give spring.profiles.active=production . Below is my application.properties .

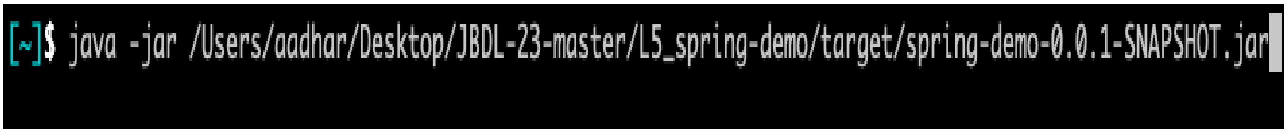


```

server.port=10000
logging.level.root = debug
```

Answer the following questions :

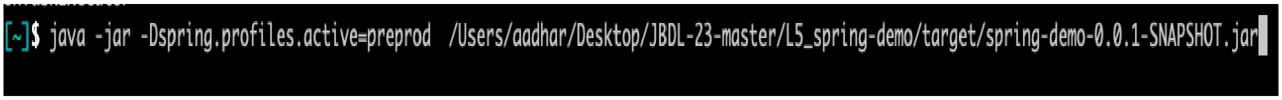
(i)Instead of application-dev.properties , can we have a file with the name dev.properties ?
(ii) At which port will my app run ?
(iii) What is the logging level ?
HINT : As active spring profile is not found , so it follows the application.properties file
https://dzone.com/articles/spring-boot-profiles-1

Challenge 11 : How to run a spring boot application from terminal ?
HINT : jar -jar <path of jar in target folder>
java -jar /Users/aadhar/Desktop/JBDL-23-master/L5_spring-demo/target/spring-demo-0.0.1-SNAPSHOT.jar



```
[~]$ java -jar /Users/aadhar/Desktop/JBDL-23-master/L5_spring-demo/target/spring-demo-0.0.1-SNAPSHOT.jar
```

Challenge 12 : Can we run spring boot project from terminal in any desired spring profile that we have defined an application-<env>.properties for ?
HINT :
java -jar -Dspring.profiles.active=preprod /Users/aadhar/Desktop/JBDL-23-master/L5_spring-demo/target/spring-demo-0.0.1-SNAPSHOT.jar



```
[~]$ java -jar -Dspring.profiles.active=preprod  /Users/aadhar/Desktop/JBDL-23-master/L5_spring-demo/target/spring-demo-0.0.1-SNAPSHOT.jar
```

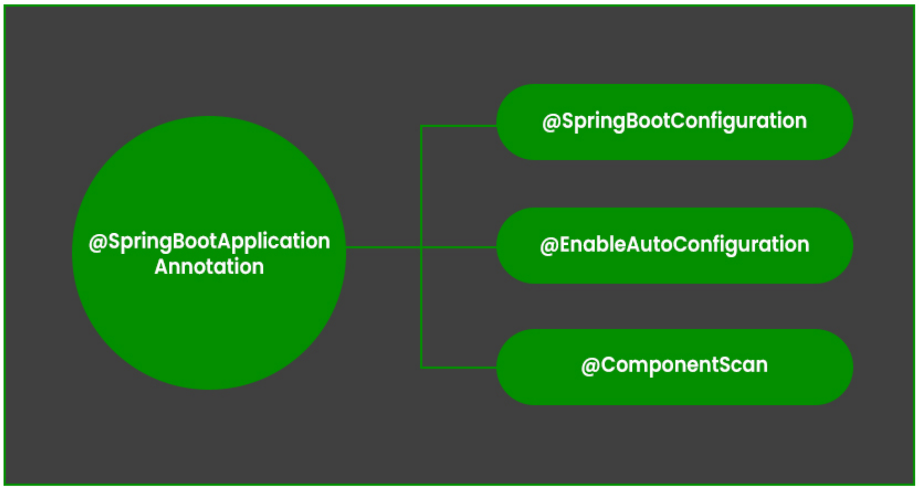
Challenge 13 : Can we run the same spring boot application on different profiles at the same time ?
HINT : port in use is different

Challenge 14 : Can we have different types of databases in different profiles ?

Challenge 15 : Can we define different logging levels for different packages ?

Challenge 16 : What is the utility of the @SpringBootApplication ?
HINT : This annotation is used to mark the main class of a Spring Boot application. It encapsulates @SpringBootConfiguration, @EnableAutoConfiguration, and @ComponentScan annotations with their default attributes.

https://www.geeksforgeeks.org/spring-boot-annotations/
https://learnjava.co.in/the-springbootapplication-annotation-explained/



References :
https://www.geeksforgeeks.org/spring-boot-logging/

https://www.youtube.com/watch?v=lGrcZsw-hKQ

https://stackoverflow.com/questions/5817738/how-to-use-log-levels-in-java

https://docs.spring.io/spring-boot/docs/2.1.13.RELEASE/reference/html/boot-features-logging.html

https://dzone.com/articles/spring-boot-profiles-1 IMP

https://www.geeksforgeeks.org/http-headers/ V Good for Request and Response Headers