Q=in which case we will use application.yml and in which case bootstrap.yml?

Q=what is yml file?

Q=how to add our own dependency in spring boot application?

Q=how to handle exception handling in spring rest?

Ans=@ControllerAdvice(basePackages = "my.chosen.package") //Global exception handler

@ControllerAdvice(annotations = RestController.class)

<https://dzone.com/articles/global-exception-handling-with-controlleradvice>

<https://dzone.com/articles/exception-handling-spring-rest>

Q=what is Eureka in spring cloud?

Q=what in spring cloud?

Q=who to call one service to another service?

Q=what is difference between zoon and fignclient ?

Q=what is the sequence of execution, Filter, DispatherServler, Inhterceptor, Normal Controller?

Ans= Filter -> DispatcherServlet -> Interceptor -> NormalController

Q=

15.1 Importing Additional Configuration Classes You need not put all your @Configuration into a single class. The @Import annotation can be used to import additional configuration classes. Alternatively, you can use @ComponentScan to automatically pick up all Spring components, including @Configuration classes. 15.2 Importing XML Configuration If you absolutely must use XML based configuration, we recommend that you still start with a @Configuration class. You can then use an @ImportResource annotation to load XML configuration files.

16.2 Disabling Specific Auto-configuration Classes If you find that specific auto-configuration classes that you do not want are being applied, you can use the exclude attribute of @EnableAutoConfiguration to disable them, as shown in the following example: import org.springframework.boot.autoconfigure.\*; import org.springframework.boot.autoconfigure.jdbc.\*; import org.springframework.context.annotation.\*; @Configuration @EnableAutoConfiguration(exclude={DataSourceAutoConfiguration.class}) public class MyConfiguration { } If the class is not on the classpath, you can use the excludeName attribute of the annotation and specify the fully qualified name instead. Finally, you can also control the list of auto-configuration classes to exclude by using the spring.autoconfigure.exclude property