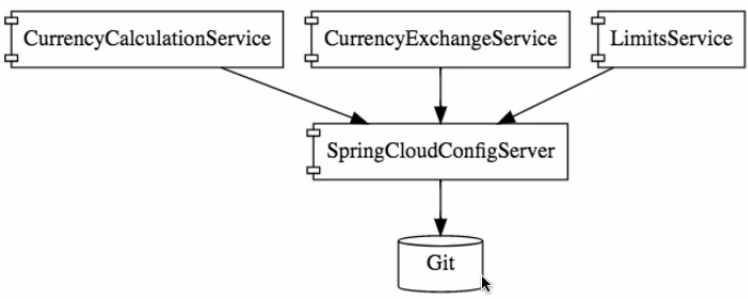
## Implementing config server

This config server must be a separate project



1. It’s a thought of externalizing the property files from our application and keeping it in a separate git repository, the great thing is all application properties files are present in 1 place,

* If u ask the config server with the name of property file , it will goto the git repository and fetch the related file and pick the respective environment file
* Its like a centralized configuration

# Reference

https://github.com/in28minutes/spring-microservices-v2/tree/main/03.microservices

This config server can store properties of all applications and all environments

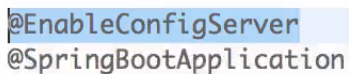
Running on diff port

-DServer.port=8001

U can get the port infor as -- 



1. add @EnableConfigServer annotation on top of that main method



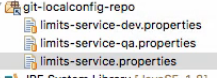
1. run this application on port 8888 , u can hit like below



1. create all properties like below with environment names,

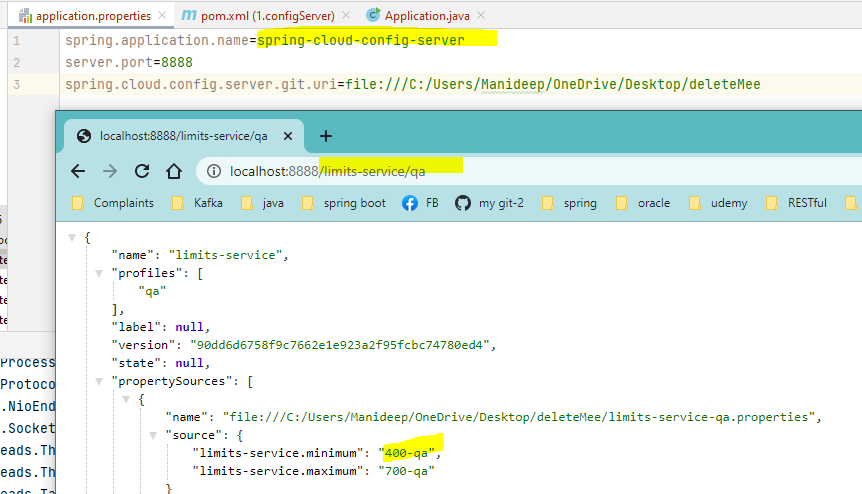
create a git repo and commit all like below

to create git repo click “**git init**” and commit command as 🡪 **git commit –m “abcd”**



In the above limits-qa.properties will have the highest priority , if the expected value is not present in that property file it will pick from the default property file.

Therefore, eventhough u have a context root for ur spring cloud config server , while searching we should not search with that context-root , we should search only with that property file and env name



## Course update

**If you are using 2.4.0, you need to add this dependency to the pom.xml:**

**HINT**: Make sure that you RESTART the server after saving your pom.xml

1. <dependency>
2. <groupId>org.springframework.cloud</groupId>
3. <artifactId>spring-cloud-starter-bootstrap</artifactId>
4. </dependency>

## Debugging help

<https://github.com/in28minutes/in28minutes-initiatives/tree/master/The-in28Minutes-TroubleshootingGuide-And-FAQ#debugging-problems-with-spring-cloud-config-server>

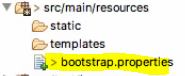
Connect cloud config server to git

Spring.cloud.config.server.git.uri= <file:///in28minutes/git/spring-microservices>

<This is the path where .git folder is present>

## Connect Limits-service to spring cloud config server

1. Rename the application.properties to bootstrap.properties



1. Place the config server git uri location in limits-service as below, but in latest versions the key name is changed



Now this will fetch values from limits-service-**dev**.properties file , because we have activated dev profile , but generally this is not the correct way to implement , provide the profile name in vm arguments

If u keep qa, then it will fetch limits-service-qa.properties file.

Or use below property

Spring.config.import=optional:configserver:http:localhost:8888

spring.config.import=optional:configserver:http://myconfigserver.com

From ur Dev appn connect to QA property file

Soring.profiles.active=dev

spring.cloud.config.profile=qa – eventhough as per above ,if ur appn is running on dev environment

if ur cloud profile is activated as QA, ur dev app will connect to qa config server/ it will load the QA config property file

From ur appn fetch diff appn property files

spring.application.name= santhoshi

spring.cloud.config.name=manideep ,now eventhough ur appn is santhoshi as per above , since ur cloud config name is manideep it will fetch the manideep related property file