**Exp-13: Spring Cloud Integration**

**If we have 2 or more applications, and common configuration for all applications can be stored in the Github.**

**This configuration properties can be accessed by all applications with help of server.**

**Github Token Password Generation: (This password will be used in the application.properties file in the server project)**

**Login into Github.**

**Profile->Settings->Developer Settings->Personal Access Tokens->Token(Classic) ->**

**Generate New Token -> Generate New Token(Classic)-> provide password -> Note (JFSD) -> expiration date, choose 30 day->**

**Select Scope (Repo, User, Project, Adminrepo\_Hook)-> Generate Token -> Copy the Generated Token.**

**My generated Token: ghp\_y4WaSjkQjnFqG1TfqczKXg8bojDER517L48f**

**Install the Git :**

**=====================**

**Download the Git for Windows installer from the official site:**

**👉 https://git-scm.com/download/win**

**Run the downloaded .exe file.**

**Follow the installation steps: (Keep other options as default)**

**Click Next through the setup.**

**Click Install and wait for the process to complete.**

**C:\Program Files\Git\bin (If Git exists there, it means it's installed but not added to the PATH.)**

**Add following two paths in the Environment variables:**

**C:\Program Files\Git\bin**

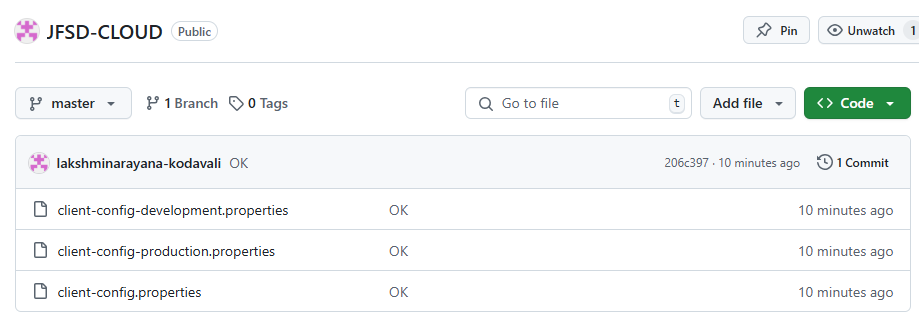
**C:\Program Files\Git\cmd**

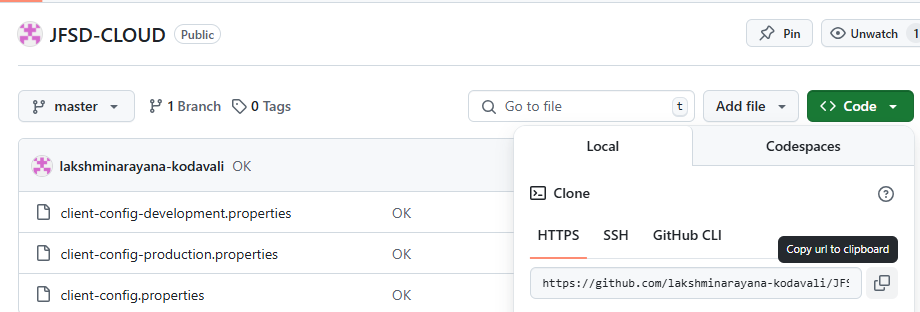
**D:\>git --version**

**git version 2.48.1.windows.1**

**========================================================**

**Run the commands provided below such that your files from (**<https://github.com/balajee-rm/JFSD-CLOUD.git>**) should be push into the specified repository(JFSD-CLOUD).**

****

****

**Copy the above HTTPS URL to use as a remote origin to push.**

**===========================================**

**Git Commands:**

**========================================**

**create a repository in Github, its name should be JFSD-CLOUD**

**1. git init**

**2. git add .**

**If you get error:**

**git config --global --add safe.directory "D:/MERN stack/clswrk"**

**git add .**

**3. git config user.email kodavali.lakshmi@gmail.com**

**4. git config --global user.name kodavali.lakshmi@gmail.com**

**5. git remote add origin https://github.com/lakshminarayana-kodavali/JFSD-CLOUD.git**

**(Above link you may copy from the hithub repository)**

**6. git status**

**7. git commit -m "Initial commit"**

**8. git branch -M master**

**9. git push -u origin master**

**1. Creating a configuration Server (Springboot Project)**

**Dependencies**

1. **Config server**
2. **Spring Boot Dev Tools**
3. **Spring Web**

**Application.properties:**

spring.application.name=SC-SERVER

server.port=8081

spring.cloud.config.server.git.uri=https://github.com/lakshminarayana-kodavali/JFSD-CLOUD.git

spring.cloud.config.server.git.username=kodavali.lakshmi@gmail.com

spring.cloud.config.server.git.password=ghp\_y4WaSjkQjnFqG1TfqczKXg8bojDER517L48f

spring.cloud.config.server.git.default-label=master

spring.cloud.config.server.git.clone-on-start=true

management.security.enabled=false

**ServerApplication.java**

**-------------------------------------**

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.config.server.EnableConfigServer;

@SpringBootApplication

@EnableConfigServer

public class ScServerApplication {

public static void main(String[] args) {

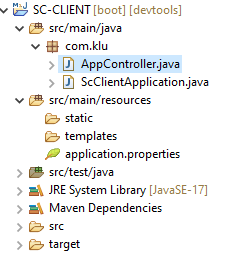
SpringApplication.run(ScServerApplication.class, args);

}

}

======================================================

**2. SC-Client Project: (Springboot Project)**



**Dependencies**

1. **Config Client**
2. **Spring Boot Dev Tools**
3. **Spring Web**

**AppController.java**

**package** com.klu;

**import** org.springframework.beans.factory.annotation.Value;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.RestController;

@RestController

**public** **class** AppController {

@Value("${msg: Not Connected to Server}")

String msg;

@GetMapping("/clientfun")

**public** String fun1() {

**return** msg;

}

}

Application.properties

spring.application.name=client-config

#application.name must be same as github pushed files prefix name

server.port=8082

#spring.profiles.active=production

#spring.profiles.active=development

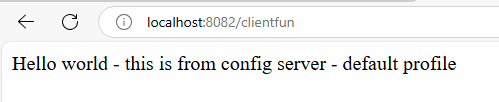
spring.config.import=configserver:http://localhost:8081

**Execution Procedure:**

**Step-1:**

Localhost:8082/clientfun

Output: Third property message is extracted from the git



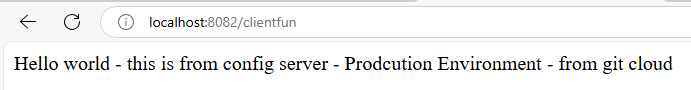
**Step2:**

Uncomment the line from application.properties of Client

spring.profiles.active=production

And run the project: Localhost:8082/clientfun

Output:Second property message is extracted from the git



**Step3:**

Uncomment the line from application.properties of Client

spring.profiles.active=development

comment the following line

#spring.profiles.active=production

and run: Localhost:8082/clientfun

Output:First property message is extracted from the git

