Spring Cloud Config Server and Client-2022

## Config Server

**Maven Pom.xml**

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.5</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-config-server</artifactId>

</dependency>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

## Configuration(application.properties)

server.port=8888

**spring.cloud.config.server.git.skipSslValidation=true**

**spring.cloud.config.server.git.username=mishrd5**

**spring.cloud.config.server.git.password=Abcd@9900**

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

**spring.cloud.config.server.git.default-label=develop**

**spring.cloud.config.server.git.uri=https://gitlab.ddlab.com/business/services/products/ddlab-app-config.git**

**spring.cloud.config.server.git.search-paths=integration/business-service**

# Main Application

@EnableConfigServer

@SpringBootApplication(exclude = SecurityAutoConfiguration.**class**)

**public** **class** ConfigServerApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(ConfigServerApplication.**class**, args);

}

}

# Config Server

## Maven (pom.xml)

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-config</artifactId>

</dependency>

## Configuration(application.properties)

spring.application.name=survey-business-service  
spring.profiles.active=integration  
management.security.enabled=false  
spring.config.import=optional:configserver:http://localhost:8888

**Controller Layer**

@RefreshScope  
@RestController  
class MessageRestController {  
  
 @Value("${msg:Config Server is not working. Please check...}")  
 private String msg;  
  
 @Value("${basicAuth.userName:Looks like username is not available...}")  
 private String authUserName;  
  
 @Value("${basicAuth.password:Secret Agent is not available}")  
 private String secretAgent;  
  
 @GetMapping("/msg")  
 public String getMsg() {  
 return this.msg;  
 }  
  
 @GetMapping("/info")  
 public ResponseEntity<String> getInfo() {  
 return new ResponseEntity<>(authUserName, HttpStatus.OK);  
 }  
  
 @GetMapping("/secret")  
 public ResponseEntity<String> getSecret() {  
 return new ResponseEntity<>(secretAgent, HttpStatus.OK);  
 }  
}

# Main Application

@SpringBootApplication  
public class SpringConfigClientApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(SpringConfigClientApplication.class, args);  
 }  
   
}