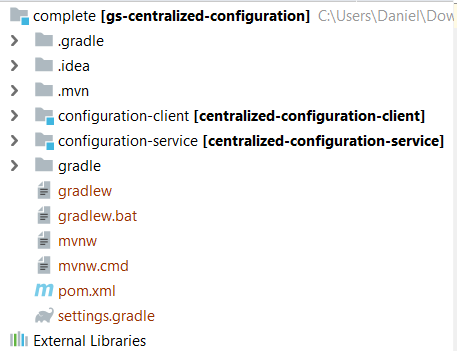
**PARA HACER UN PROJECTO NESTADO EN INTELLI-J**

**pom.xml**

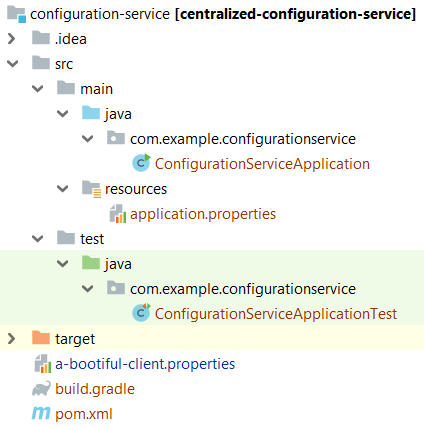
*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
  
 <**groupId**>org.springframework</**groupId**>  
 <**artifactId**>gs-centralized-configuration</**artifactId**>  
 <**version**>0.1.0</**version**>  
 <**packaging**>pom</**packaging**>  
  
 <**modules**>  
 <**module**>configuration-client</**module**>  
 <**module**>configuration-service</**module**>  
 </**modules**>  
  
</**project**>

…

**settings.gradle**

**rootProject.name = 'gs-centralized-configuration'  
include 'configuration-client'  
include 'configuration-service'**

**configuration-service**



**pom.xml**

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
 <**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>**spring-boot-starter-parent**</**artifactId**>  
 <**version**>2.1.8.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  
 <**groupId**>com.example</**groupId**>  
 <**artifactId**>**centralized-configuration-service**</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**name**>**centralized-configuration-service**</**name**>  
 <**description**>Demo project for Spring Boot</**description**>  
  
 <**properties**>  
 <**java.version**>1.8</**java.version**>  
 <**spring-cloud.version**>Greenwich.SR3</**spring-cloud.version**>  
 </**properties**>  
  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>**spring-cloud-config-server**</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>**spring-boot-starter-test**</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**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**>  
  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>**spring-boot-maven-plugin**</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
  
</**project**>

**build.gradle**

**plugins** {  
 id **'org.springframework.boot'** version **'2.1.8.RELEASE'** id **'io.spring.dependency-management'** version **'1.0.8.RELEASE'** id **'java'**}  
  
**group = 'com.example'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '1.8'  
  
repositories** {  
 mavenCentral()  
}  
  
**ext** {  
 set(**'springCloudVersion'**, **"Greenwich.SR3"**)  
}  
  
**dependencies** {  
 implementation **'org.springframework.cloud:spring-cloud-config-server'** testImplementation **'org.springframework.boot:spring-boot-starter-test'**}  
  
**dependencyManagement** {  
 imports {  
 mavenBom **"org.springframework.cloud:spring-cloud-dependencies:**${springCloudVersion}**"** }  
}

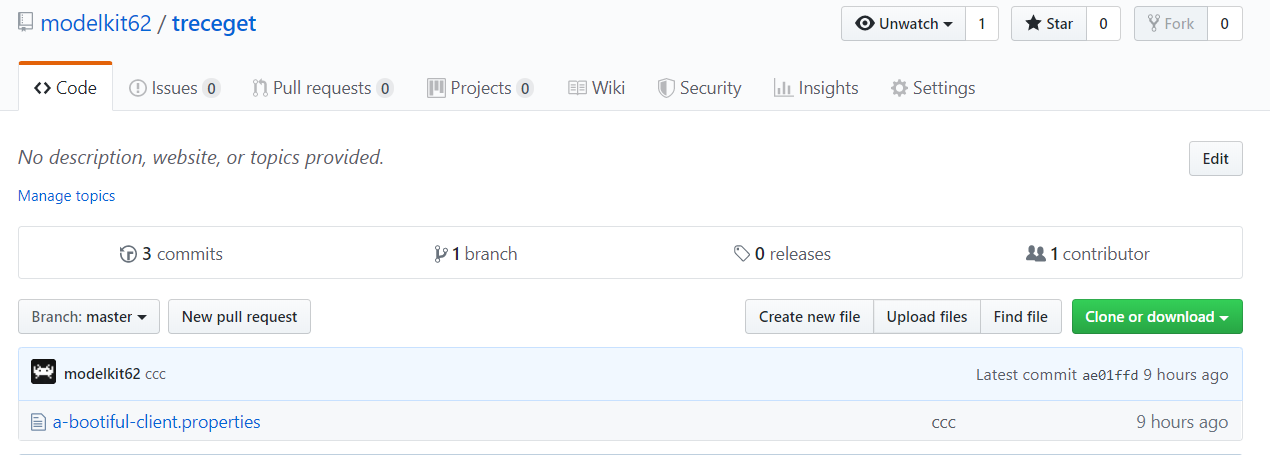
…

**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.config.server.EnableConfigServer;  
  
**@EnableConfigServer  
@SpringBootApplication  
public class ConfigurationServiceApplication** {  
  
 **public static void main**(String[] args) {  
 SpringApplication.*run*(ConfigurationServiceApplication.**class**, args);  
 }  
}

…

**application.properties**

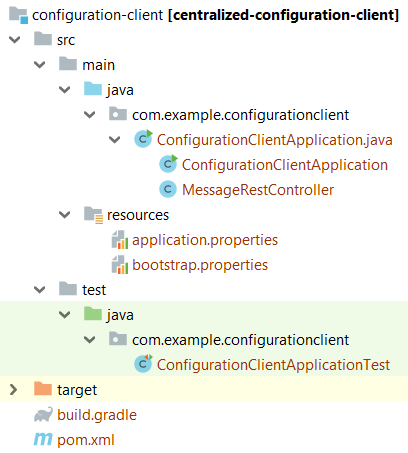
**server.port**=**8888  
spring.cloud.config.server.git.uri**=**https://github.com/modelkit62/treceget**



…

**import static** org.assertj.core.api.Assertions.*assertThat*;  
**import** org.junit.Test;  
**import** org.junit.runner.RunWith;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.boot.test.context.SpringBootTest;  
**import** org.springframework.cloud.config.server.environment.**EnvironmentController**;  
**import** org.springframework.test.context.junit4.SpringRunner;  
  
**@RunWith(SpringRunner.class)  
@SpringBootTest(properties = { "spring.profiles.active=native" })  
public class ConfigurationServiceApplicationTest** {  
  
 **@Autowired** **private EnvironmentController** **controller**;  
  
 **@Test** **public void contextLoads**() {  
 *assertThat*(**controller**).isNotNull();  
 }  
}

**configuration-client**



**pom.xml**

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
 <**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>**spring-boot-starter-parent**</**artifactId**>  
 <**version**>2.1.8.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  
 <**groupId**>com.example</**groupId**>  
 <**artifactId**>**centralized-configuration-client**</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**name**>**centralized-configuration-client**</**name**>  
 <**description**>Demo project for Spring Boot</**description**>  
  
 <**properties**>  
 <**java.version**>1.8</**java.version**>  
 <**spring-cloud.version**>Greenwich.SR3</**spring-cloud.version**>  
 </**properties**>  
  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>**spring-boot-starter-actuator**</**artifactId**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>**spring-boot-starter-web**</**artifactId**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>**spring-cloud-starter-config**</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>**spring-boot-starter-test**</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**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**>  
  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>**spring-boot-maven-plugin**</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
  
</**project**>

**build.gradle**

**buildscript** {  
 ext { springBootVersion = **'2.1.6.RELEASE'** }  
 repositories { mavenCentral() }  
 dependencies { classpath(**"org.springframework.boot:spring-boot-gradle-plugin:**${springBootVersion}**"**) }  
}  
  
apply **plugin**: **'java'**apply **plugin**: **'eclipse'**apply **plugin**: **'idea'**apply **plugin**: **'org.springframework.boot'**apply **plugin**: **'io.spring.dependency-management'  
  
bootJar** {  
 baseName = **'configuration-client'** version = **'0.0.1-SNAPSHOT'**}  
**sourceCompatibility = 1.8  
targetCompatibility = 1.8**  
**repositories** { mavenCentral() }  
  
**dependencies** {  
 compile(**'org.springframework.cloud:spring-cloud-starter-config'**)  
 compile(**'org.springframework.boot:spring-boot-starter-actuator'**)  
 compile(**'org.springframework.boot:spring-boot-starter-web'**)  
 testCompile(**'org.springframework.boot:spring-boot-starter-test'**)  
}  
  
**dependencyManagement** {  
 imports {  
 mavenBom **"org.springframework.cloud:spring-cloud-dependencies:Finchley.SR2"** }  
}  
  
eclipse {  
 classpath {  
 containers.remove(**'org.eclipse.jdt.launching.JRE\_CONTAINER'**)  
 containers **'org.eclipse.jdt.launching.JRE\_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.StandardVMType/JavaSE-1.8'** }  
}

…

**import** org.springframework.beans.factory.annotation.Value;  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.context.config.annotation.RefreshScope;  
**import** org.springframework.web.bind.annotation.RequestMapping;  
**import** org.springframework.web.bind.annotation.RestController;  
  
**@SpringBootApplication  
public class ConfigurationClientApplication** {  
  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(ConfigurationClientApplication.**class**, args);  
 }  
}  
  
**@RefreshScope  
@RestController  
class MessageRestController** {  
  
 **@Value("${message:Hello default}")** **private** String **message**;  
  
 **@RequestMapping("/message")** String getMessage() {  
 **return this**.**message**;  
 }  
}

…

**application.properties**

**management.endpoints.web.exposure.include**=**\***

…

**bootstrap.properties**

**spring.application.name**=**a-bootiful-client  
spring.cloud.config.uri**=**http://localhost:8888**

…

**import static** org.assertj.core.api.Assertions.\*;  
  
**import** org.junit.Test;  
**import** org.junit.runner.RunWith;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.boot.test.context.SpringBootTest;  
**import** org.springframework.boot.test.util.TestPropertyValues;  
**import** org.springframework.cloud.context.refresh.ContextRefresher;  
**import** org.springframework.core.env.ConfigurableEnvironment;  
**import** org.springframework.test.context.junit4.SpringRunner;  
  
**@RunWith(SpringRunner.class)  
@SpringBootTest  
public class ConfigurationClientApplicationTest** {  
  
 **@Autowired** **private ConfigurableEnvironment** **environment**;  
  
 **@Autowired** **private MessageRestController** **controller**;  
  
 **@Autowired**  
 **private ContextRefresher** **refresher**;  
  
 **@Test** **public void contextLoads**() {  
 *assertThat*(**controller**.getMessage()).isNotEqualTo(**"Hello test"**);  
 TestPropertyValues  
 .*of*(**"message:Hello test"**)  
 .applyTo(**environment**);  
 *assertThat*(**controller**.getMessage()).isNotEqualTo(**"Hello test"**);  
 **refresher**.refresh();  
 *assertThat*(**controller**.getMessage()).isEqualTo(**"Hello test"**);  
 }  
}

**PROYECTO SIMPLIFICADO**

**SERVER**

**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.config.server.EnableConfigServer;  
  
**@EnableConfigServer  
@SpringBootApplication  
public class ClientServerApplication** {  
 **public static void main**(String[] args) {  
 SpringApplication.*run*(ClientServerApplication.**class**, args);  
 }  
}

…

**application.properties**

**server.port**=**8083  
spring.cloud.config.server.git.uri**=**https://github.com/modelkit62/treceget**

// si despues hago un get en postman => **localhost:8083/application/default** me tiene que salir lo que hay en git

**build.gradle**

**plugins** {  
 id **'org.springframework.boot'** version **'2.2.0.RELEASE'** id **'io.spring.dependency-management'** version **'1.0.8.RELEASE'** id **'java'**}  
  
**group = 'com.example'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '1.8'  
  
repositories** {  
 mavenCentral()  
 maven { url **'https://repo.spring.io/milestone'** }  
}  
  
**ext** {  
 set(**'springCloudVersion'**, **"Hoxton.M3"**)  
}  
  
**dependencies** {  
 implementation **'org.springframework.cloud:spring-cloud-config-server'** testImplementation(**'org.springframework.boot:spring-boot-starter-test'**) {  
 exclude **group**: **'org.junit.vintage'**, **module**: **'junit-vintage-engine'** }  
}  
  
**dependencyManagement** {  
 imports {  
 mavenBom **"org.springframework.cloud:spring-cloud-dependencies:**${springCloudVersion}**"** }  
}  
  
test {  
 useJUnitPlatform()  
}

**CLIENT**

**import** org.springframework.beans.factory.annotation.Value;  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.context.config.annotation.RefreshScope;  
**import** org.springframework.web.bind.annotation.RequestMapping;  
**import** org.springframework.web.bind.annotation.RestController;  
  
**@SpringBootApplication  
public class ClientClientApplication** {  
 **public static void main**(String[] args) {  
 SpringApplication.*run*(ClientClientApplication.**class**, args);  
 }  
}  
  
**@RefreshScope  
@RestController  
class MessageRestController** {  
  
 **@Value("${message:Hello default}")** **private** String **message**;  
  
 **@RequestMapping("/message")** String **getMessage**() {  
 **return this**.**message**;  
 }  
}

…

**application.properties**

**management.endpoints.web.exposure.include=\* // para actuator**

…

**bootstrap.properties**

**spring.application.name=dummyconfig  
spring.cloud.config.uri=http://localhost:8083**

…

**bootstrap.yml**

**spring**:

**application**:

**name**: dummyconfig

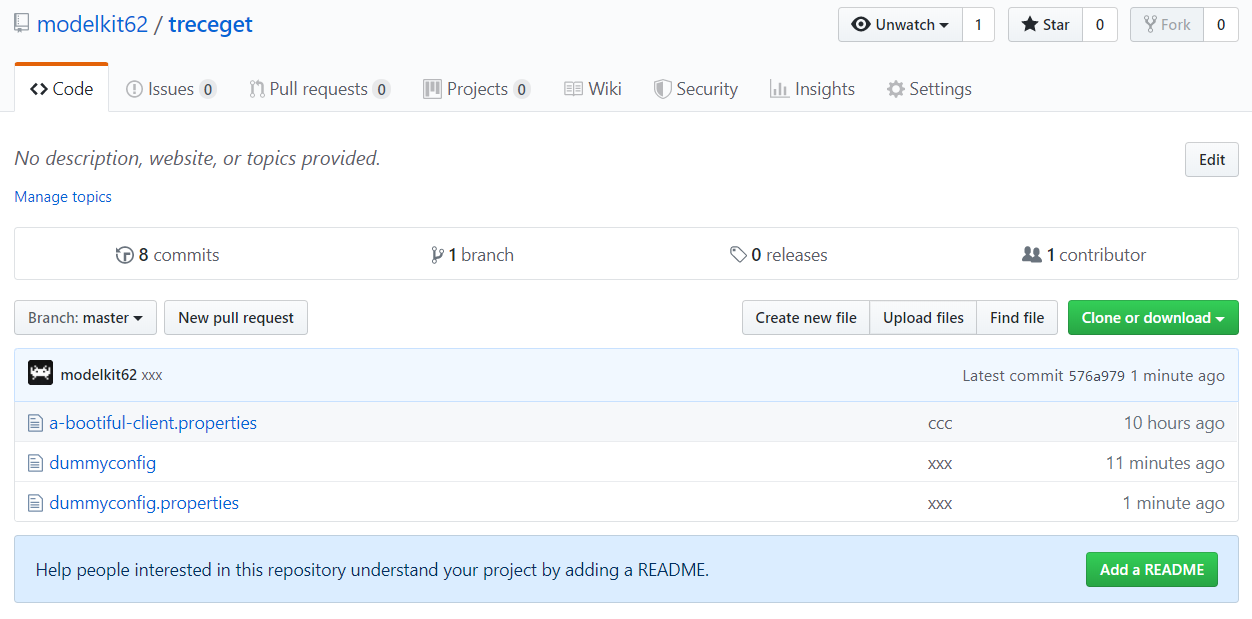
# N.B. this is the default:

**cloud**:

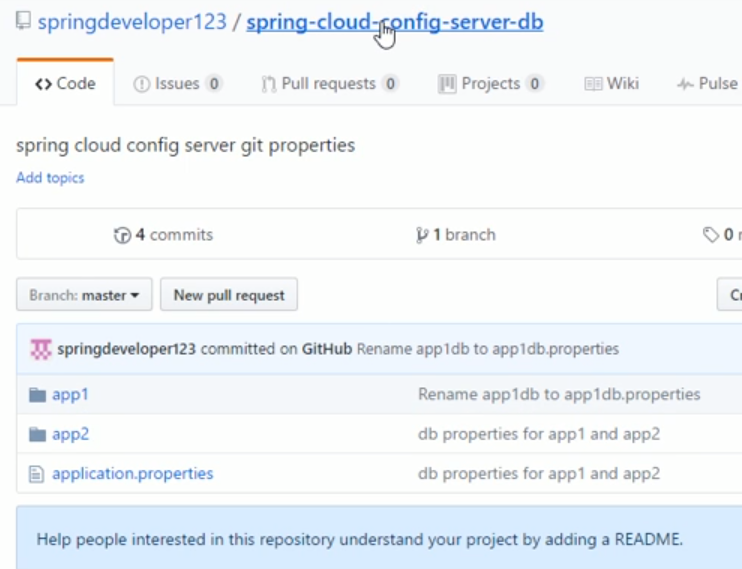
**config**:

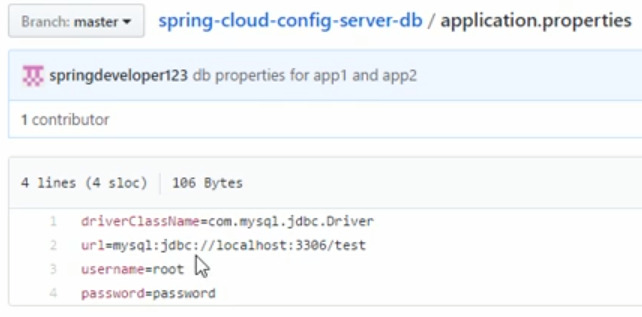
**uri**: http://localhost:8083

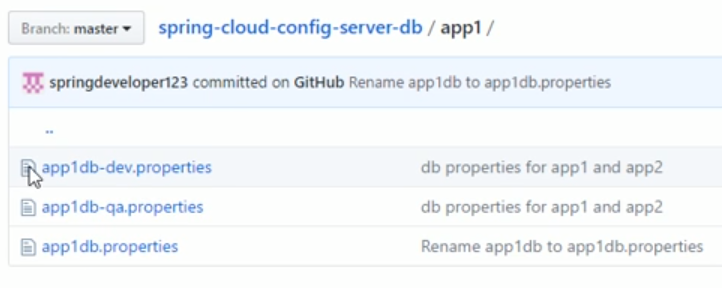
**\* En postman es: http://localhost:8080/message**

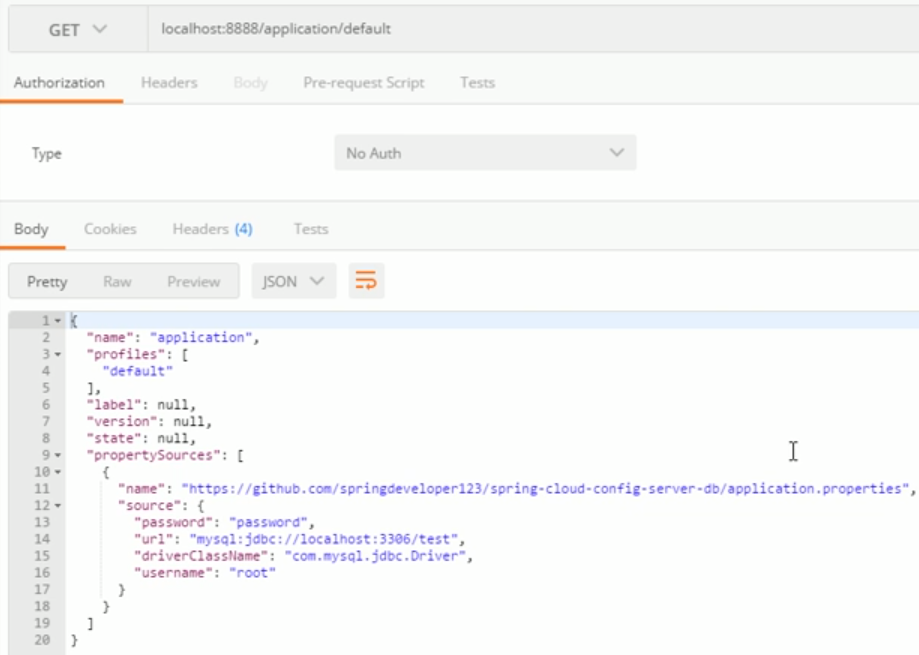


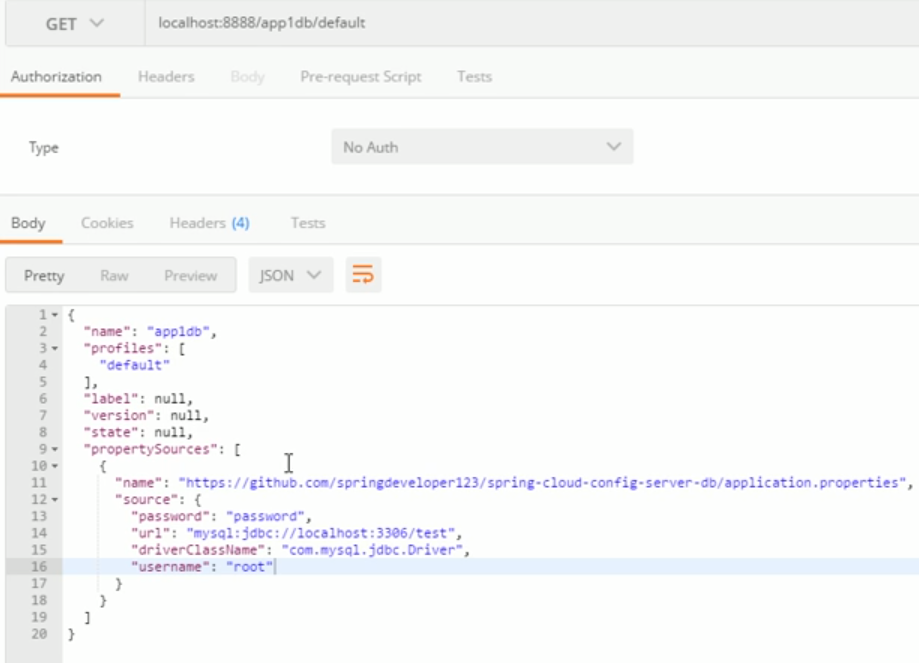
==============================

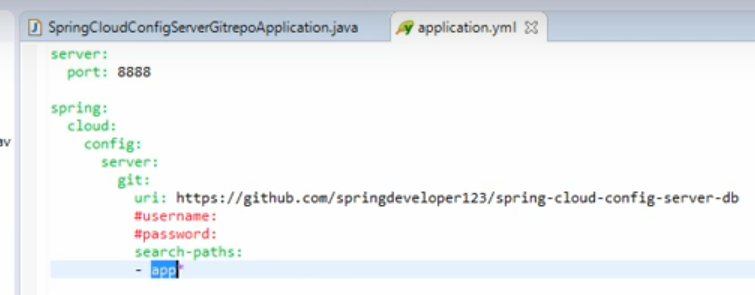


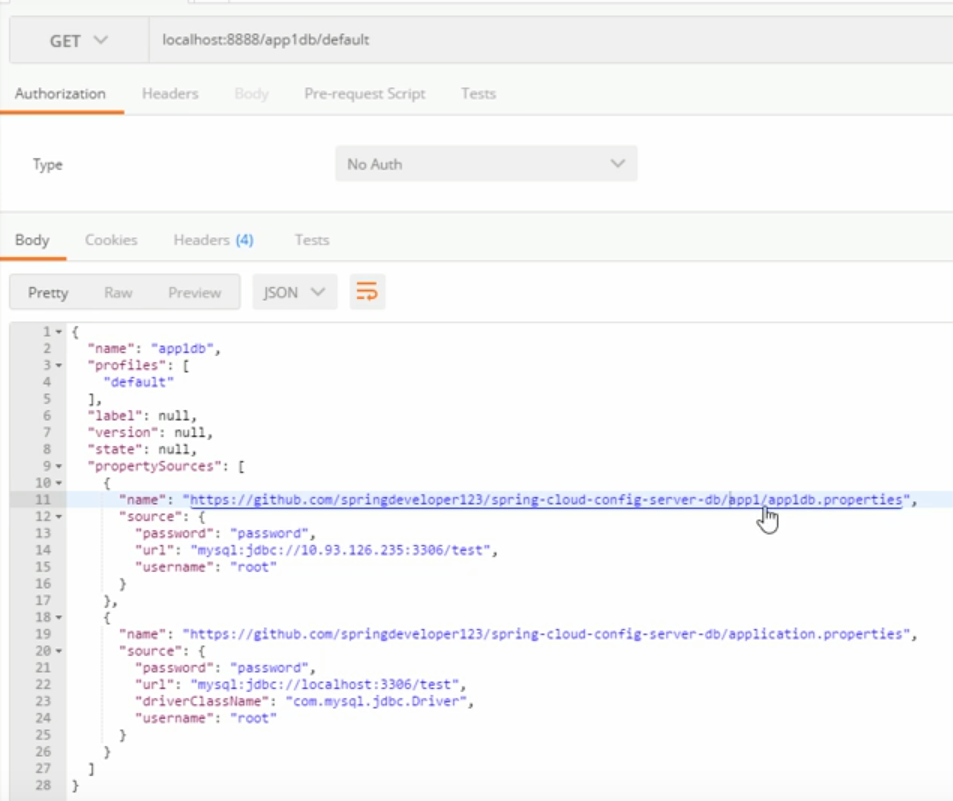


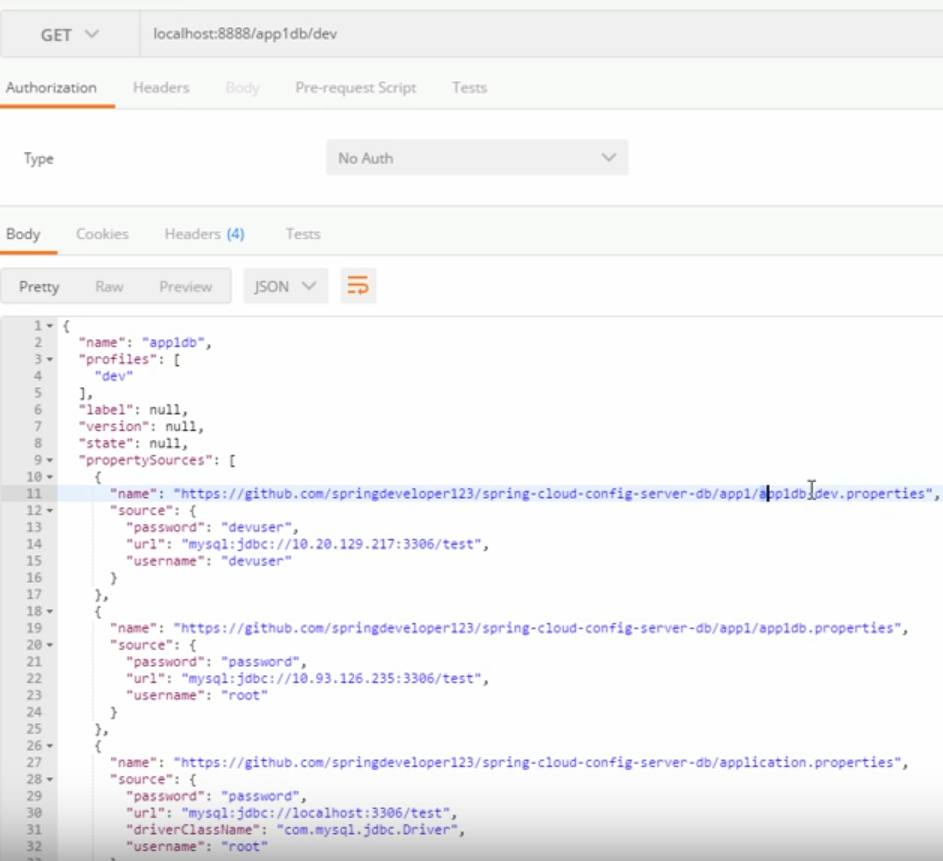


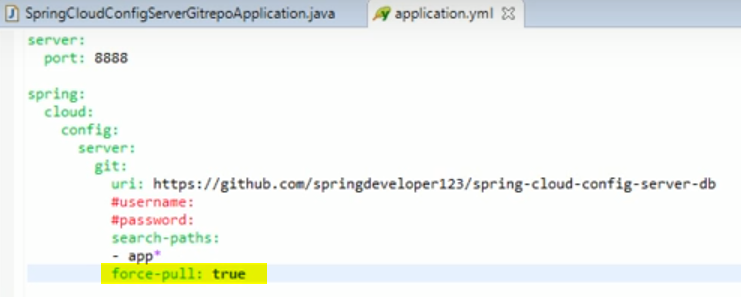




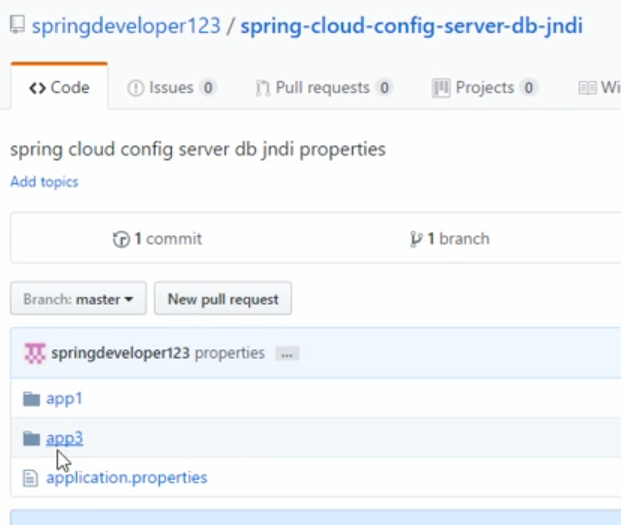


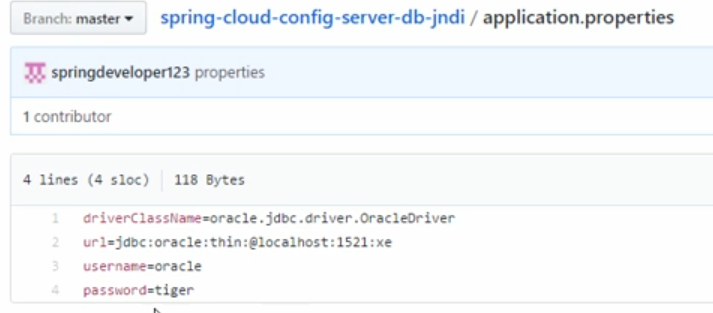


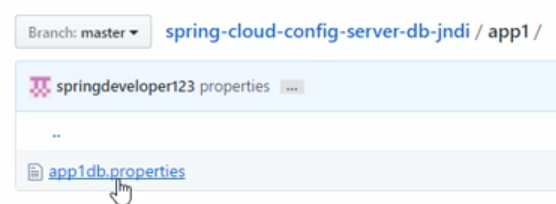


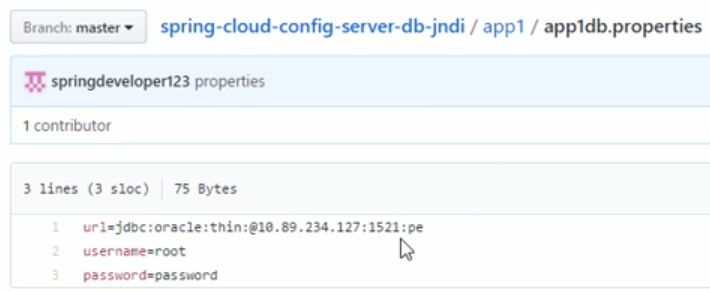


PARA TRAER DESDE MULTIPLE REPOS

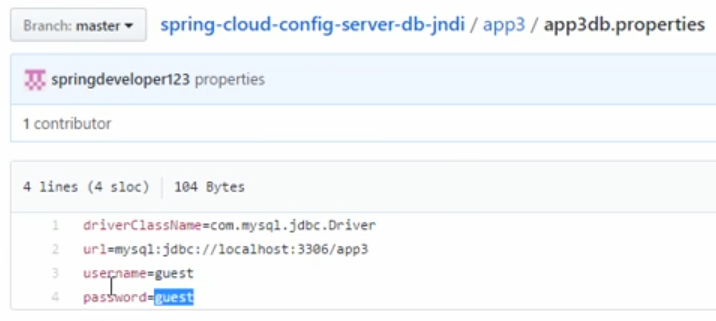




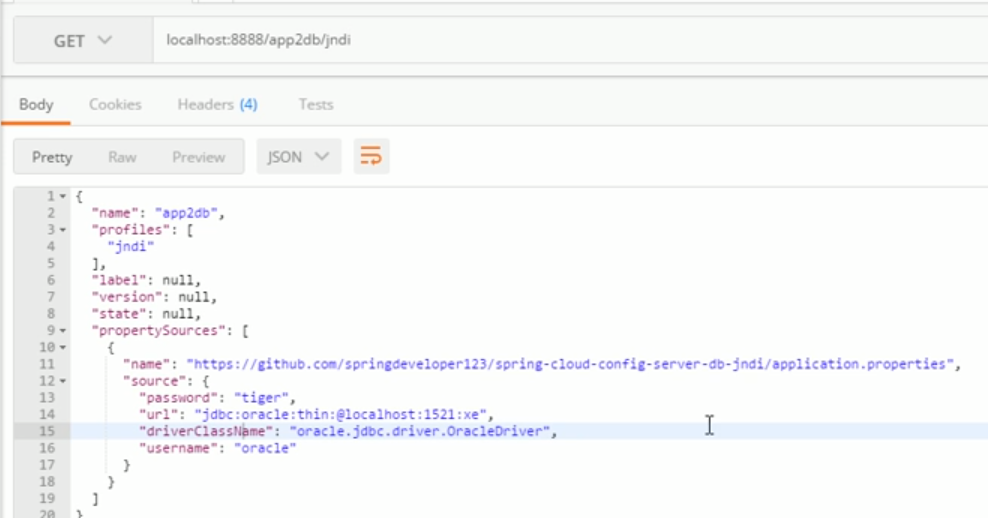


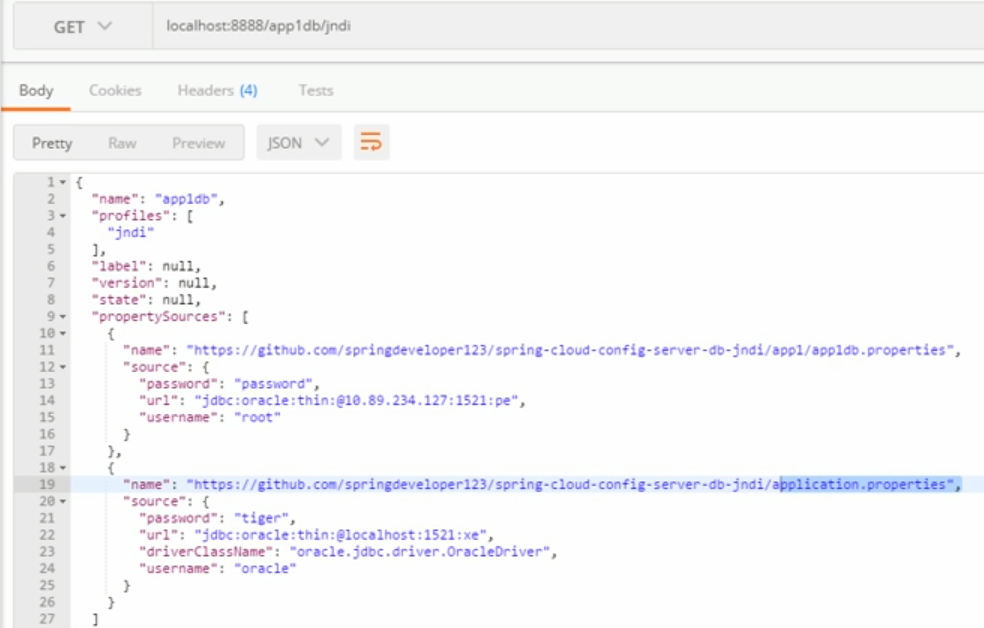




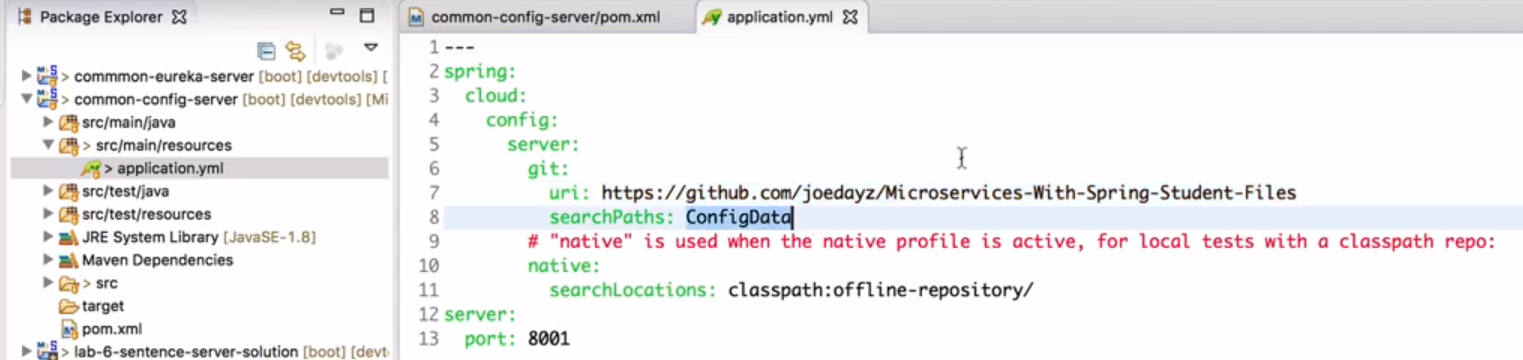


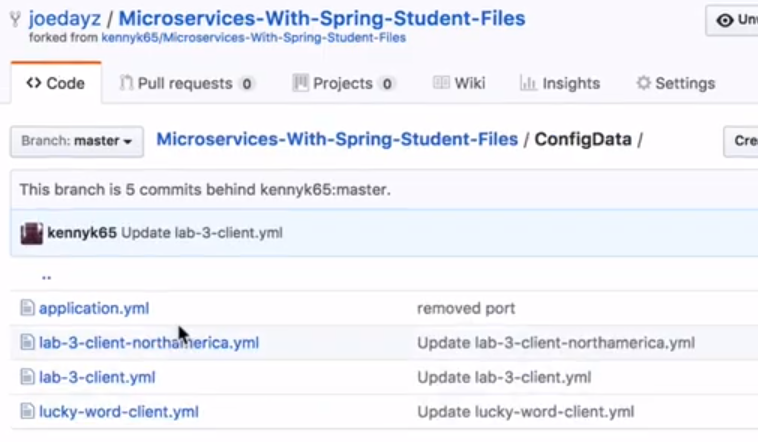


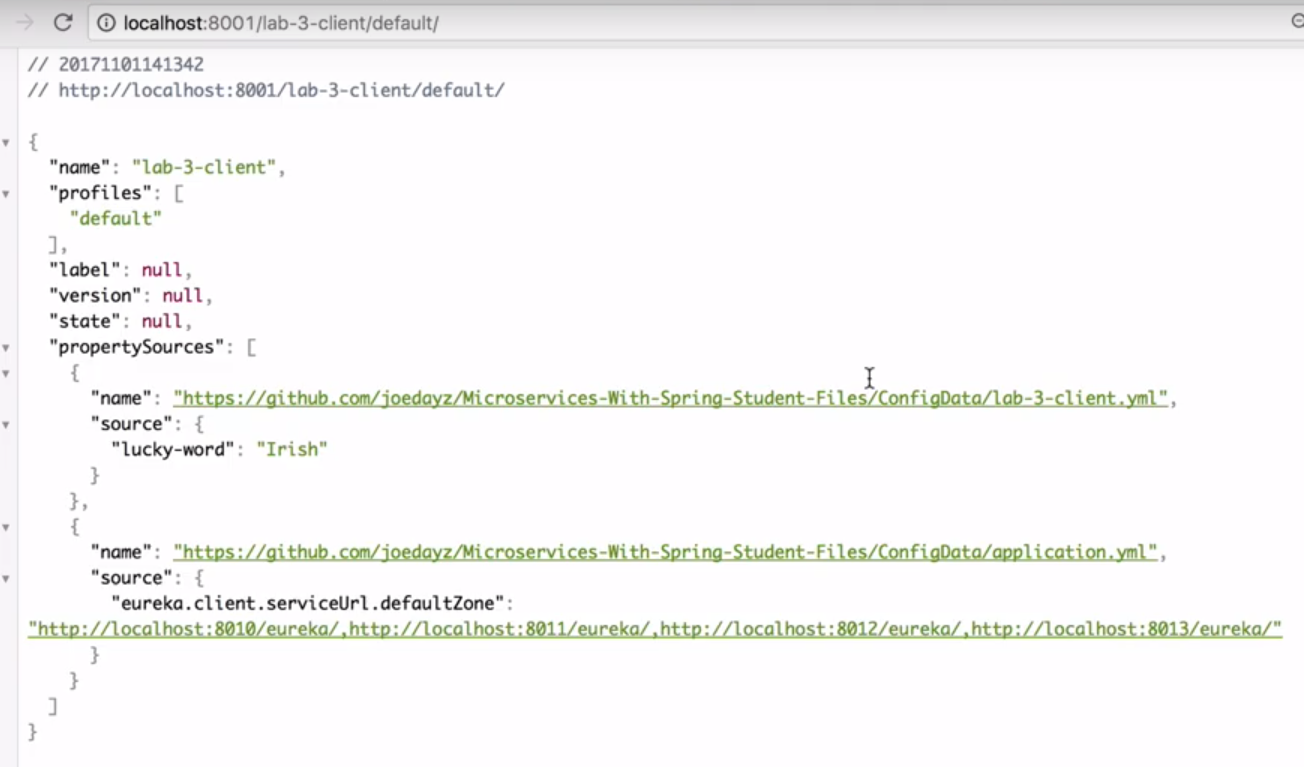




DE JOYDAZ







PROYECTO EUREKA



