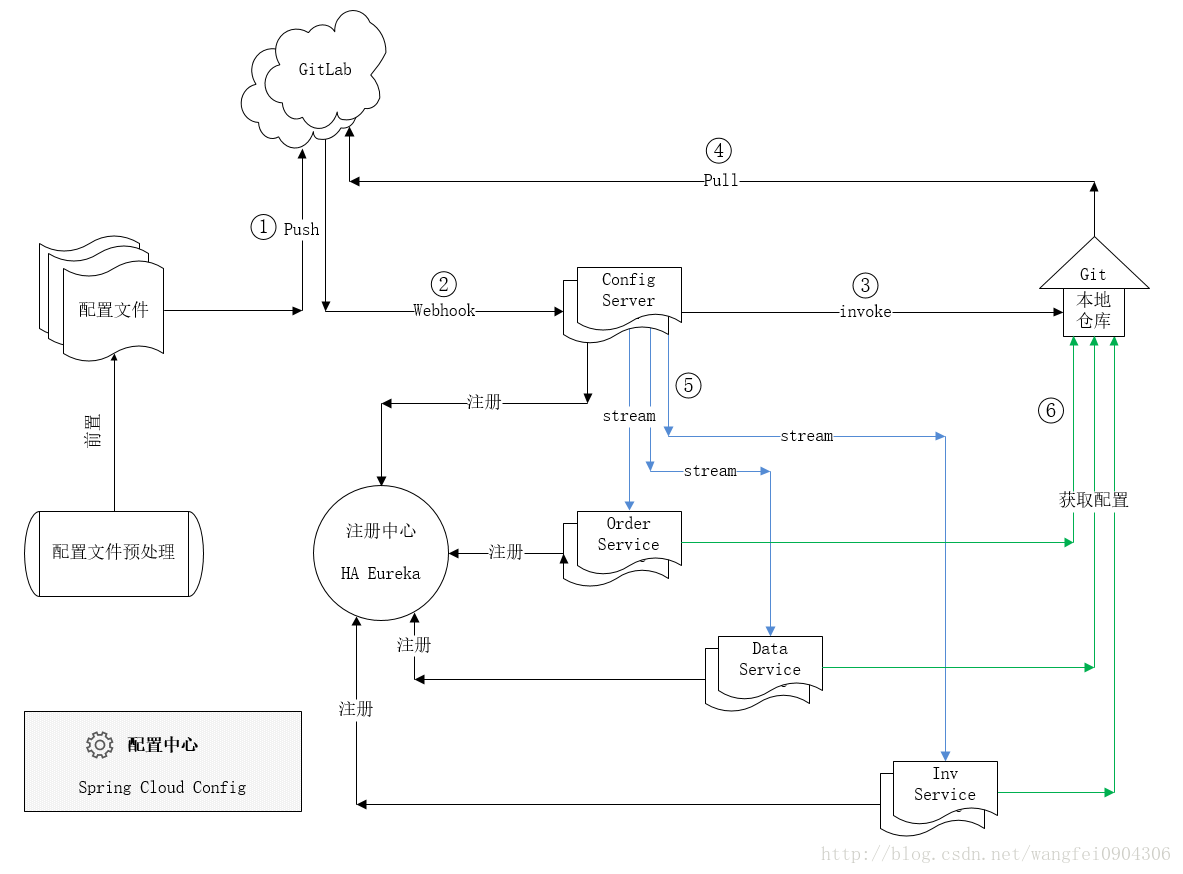
先安好GitLab（<http://blog.csdn.net/wangfei0904306/article/details/76177218>），如果已经有跳过这一步。

SpringCloud Config步骤流程如下图所示：



**第一步：上传文件到GitLab；**

**第二步：GitLab通过Webhook调用ConfigServer；**

**第三、四步：ConfigServer拉取配置到本地仓库；**

**第五步：ConfigServer通过Stream触发各ConfigClient；**

**第六步：ConfigClient获取配置并更新。**

**第一步就是Git的Push，毫无疑问都会，从第二步开始说。**

**第二步，GitLab通过Webhook调用ConfigServer；**

（1）首先要创建ConfigServer应用。

POM文件里的依赖：

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-cloud-starter-bus-amqp</artifactId>

</dependency>

启动类添加相应注解：

@SpringBootApplication

@EnableDiscoveryClient

@EnableConfigServer

public class ConfigServerApplication {

public static void main(String[] args) {

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

}

}

配置文件：

**server:**

**port:** 10900

**management:**

**security:**

**enabled: false** *# 关闭验证*

**spring:**

**application:**

**name:** config-server

**profiles:**

**active:** local

**rabbitmq:**

**addresses:** 192.168.14.97

**port:** 5672

**username:** guest

**password:** guest

**cloud:**

**config:**

**server:**

**git:**

**uri:** http://192.168.14.97/cibei/cibei-config-demo.git *# 配置git仓库的地址*

**search-paths:** config-center-ymls/config-repo *# git仓库地址下的相对地址，可以配置多个，用,分割。*

**username:** wangfei *# git仓库的账号*

**password:** Icibei123 *# git仓库的密码*

**eureka:**

**instance:**

**prefer-ip-address: true**

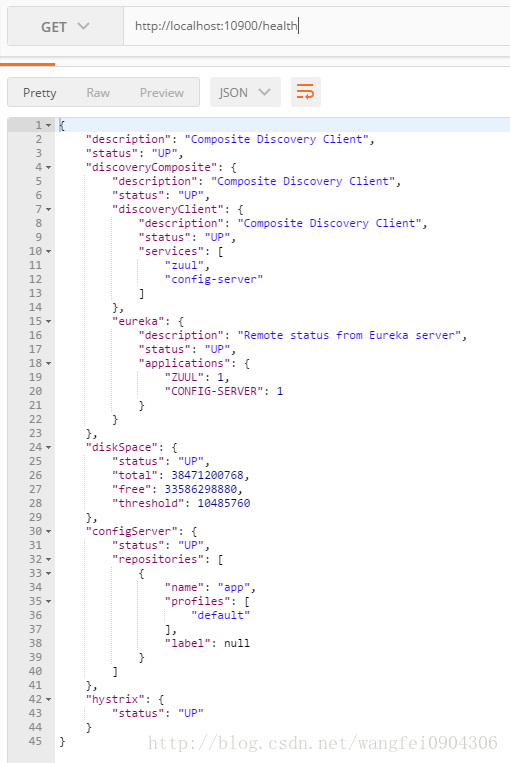
**client:**

**service-url:**

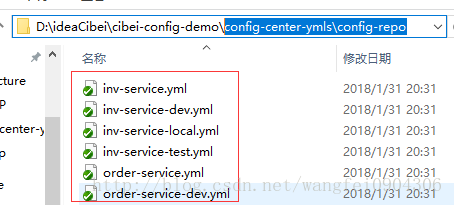
**defaultZone:** http://localhost:10001/eureka/

这样就生成了一个最简单的ConfigServer，启动它并打开健康检查网址：[http://localhost:10900/health](http://localhost:10900/health" \t "_blank)

出现如下图说明启动正确。

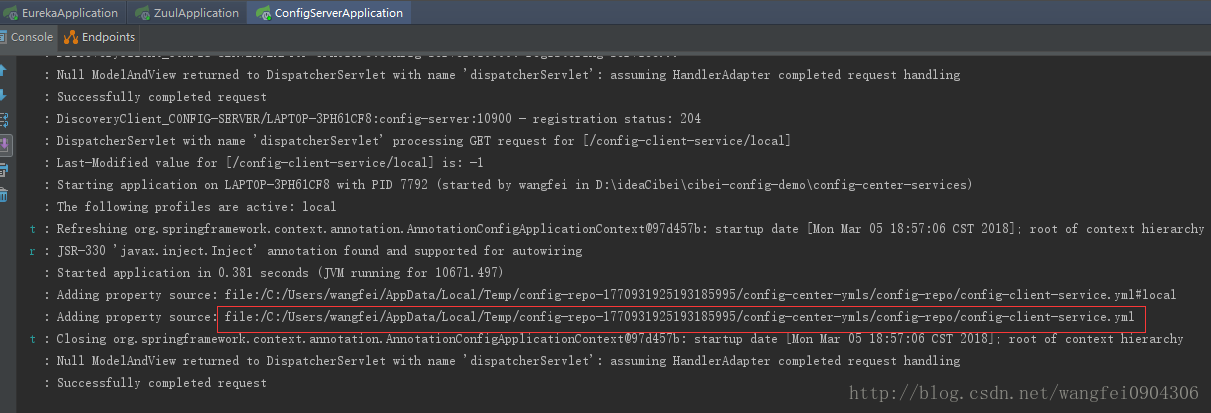


此时访问Git仓库中的配置文件

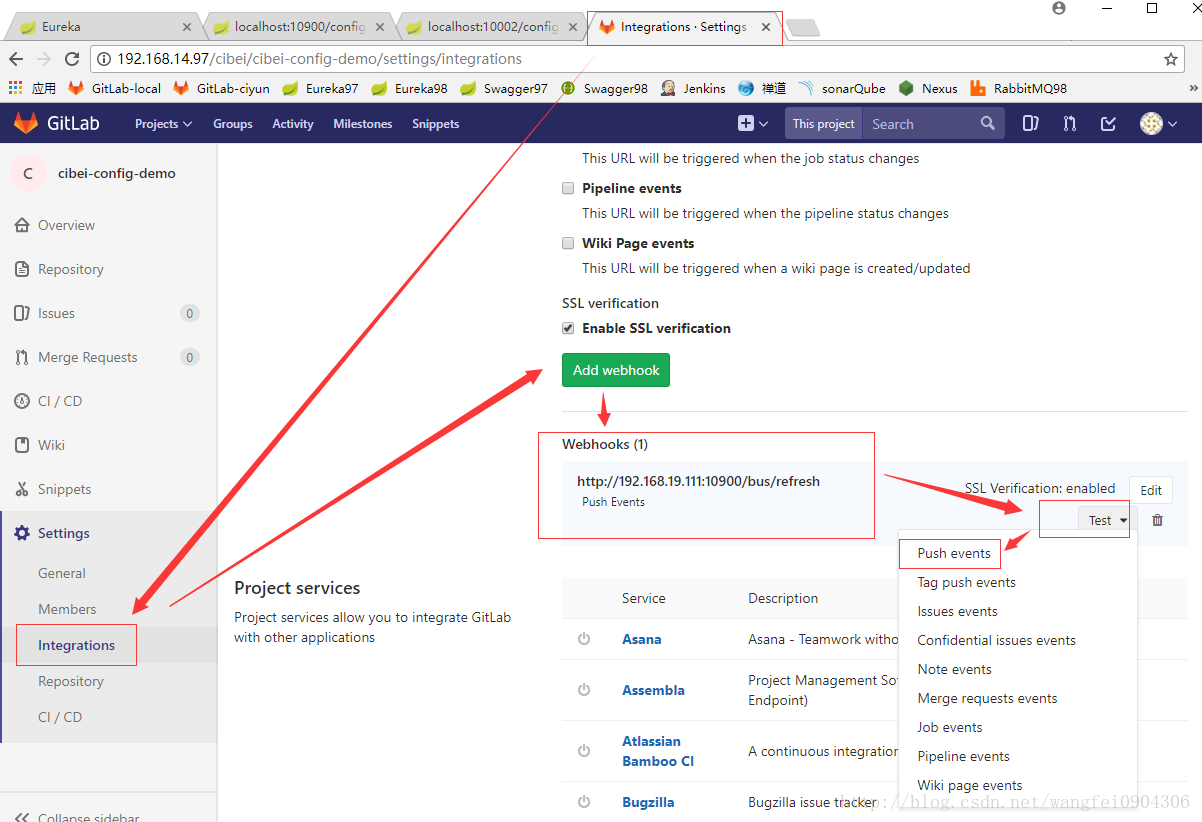


<http://localhost:10900/order-service.yml> 应该可以访问到相应的内容

（2）访问[http://localhost:10900/bus/refresh注意是POST调用。此时ConfigServer服务应有如下日志，说明调用成功。](http://localhost:10900/bus/refresh" \t "_blank)

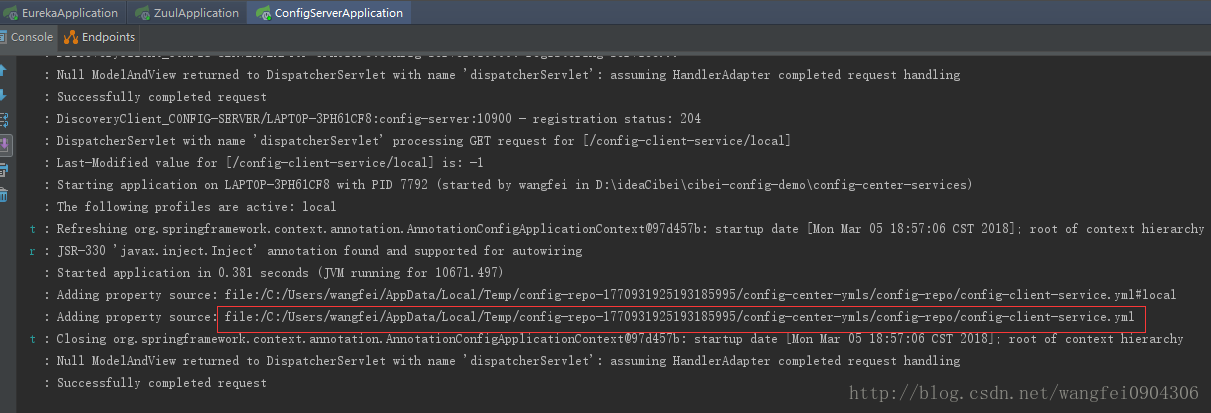


（3）将[http://localhost:10900/bus/refresh](http://localhost:10900/bus/refresh" \t "_blank)配置到GitLab相应的Weebhook。测试发现[ConfigServer服务会有同第二步一致的数据，说明配置成功，第二步完成。](http://localhost:10900/bus/refresh" \t "_blank)

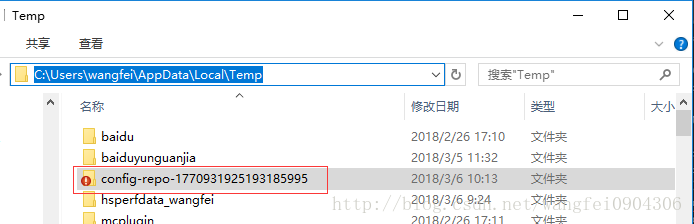


**第三、四步：ConfigServer拉取配置到本地仓库；**

我们回顾第二步第（2）条，调用[http://localhost:10900/bus/refresh](http://localhost:10900/bus/refresh" \t "_blank)[时的日志：](http://localhost:10900/bus/refresh)



红框线标示出来的位置，其实已经把数据拉到了本地Git仓库。查看相应的目录位置：



说明摘取到本地仓库成功。

**第五步：ConfigServer通过Stream触发各ConfigClient；**

实际上是ConfigServer发送消息给消息总线，消息总线再发送消息给各个ConfigClient，各个ConfigClient再去获取配置内容。

（1）注意ConfigServer有如下依赖：

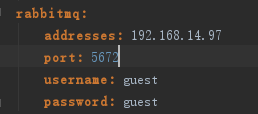
<dependency>

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

<artifactId>spring-cloud-starter-bus-amqp</artifactId>

</dependency>

并有相应的RabbitMQ配置



（2）新建config-client-service。

加入如下依赖：

<!--config center-->

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-cloud-starter-bus-amqp</artifactId>

</dependency>

<dependency>

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

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

</dependency>

加入配置bootstrap.yml，着重注意其spring.cloud.config.discovery配置，config-server为配置中心服务名。

**server:**

**port:** 10401

**spring:**

**application:**

**name:** config-client-service

**profiles:**

**active:** local

**cloud:**

**config:**

**discovery:**

**enabled: true**

**service-id:** config-server

**rabbitmq:**

**addresses:** 192.168.14.97

**port:** 5672

**username:** guest

**password:** guest

---

*################### 开发环境的profile ###################*

**spring:**

**profiles:** local

**eureka:**

**instance:**

*#hostname: ${spring.cloud.client.ipAddress}*

**client:**

**service-url:**

**defaultZone:** http://127.0.0.1:10001/eureka/

（3)启动config-client-service项目，启动Logo下出现如下字样表示配置成功。



**第六步：ConfigClient获取配置并更新。**

（1）config-client-service服务下创建controller，注意@RefreshScope，加上此注解，才能在运行时更新配置。

@RestController

@RequestMapping("/example")

@RefreshScope

public class ExampleController {

@Value("${config.center.test}")

private String test;

@GetMapping("/test")

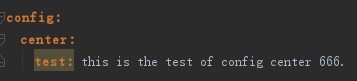
public String test(){

return "config center server test: " + test;

}

}

（2）在cibei-config-demo项目的config-center-ymls\config-repo目录下添加配置文件config-client-service.yml，config-client-service.yml文件中要有config.center.test值。

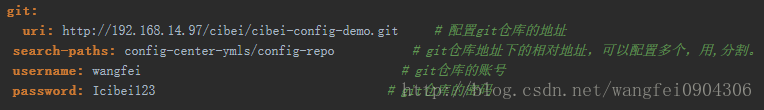


参考ConfigServer配置可以很明白：

cibei-config-demo --对应-- 仓库项目；

config-center-ymls\config-repo目录 --对应-- ConfigServer的search-paths；

配置文件名config-client-service.yml --对应-- 服务名



（3）调用controller中接口：[http://localhost:10002/config-client-service/example/test](http://localhost:10002/config-client-service/example/test" \t "_blank)

（4）修改上述的值并Push到GitLab，稍后再次调用（3）中接口，查看变化。