SpringCloud第二季

1、课程内容（SpringCloud+SpringCloud alibaba）

2、技术要求：java8+maven+git(github)+Nginx+RabbitMQ+springboot2.0

什么是微服务

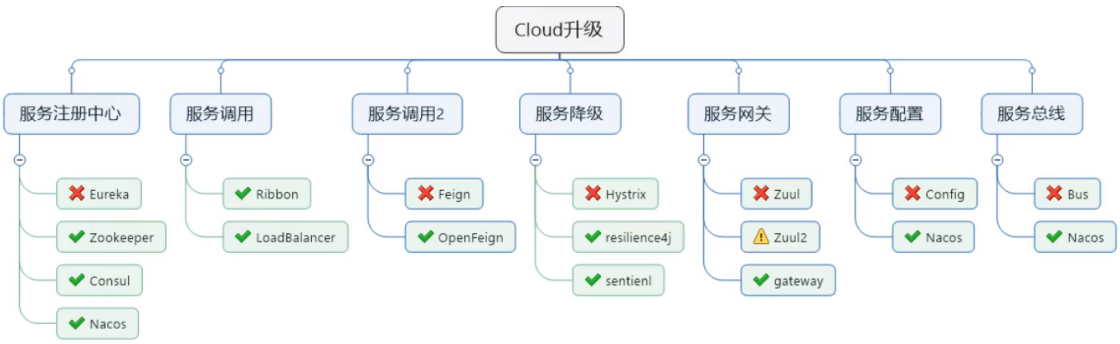
微服务架构是一种架构模式，它提倡将单一应用程序划分成一组小的服务，服务之间互相协调、互相配合，为用户提供最终价值。每个服务运行在其独立的进程中，服务与服务间采用轻量级的通讯机制互相协作(通常是基于HTTP机制的RESTful API)。每个服务都围绕着具体业务进行构建，并且能够独立的部署在生产环境、类生产环境

SpringCloud

分布式微服务架构的一站式解决方案，是多种微服务架构落地技术的集合体，俗称微服务全家桶。

版本选型：进入到spring官网https://spring.io/，Projects-->Spring Cloud-->LEARN下就可以看到最新的版本，点击 Reference Doc.就可以看到当前SpringCloud推荐的SpringBoot版本

SpringCloud与SpringBoot版本对应关系查看：
https://start.spring.io/actuator/info



参考资料：可以查看SpringCloud中文文档

父工程搭建

maven项目-->maven原型site-->.....搭建完成后；

设置：（1）字符编码设置setting-->Editor-->File Encodings，设置3个UTF-8，复选框打勾；

（2）注解生效激活：setting-->Build-->Compliler-->Annotation，勾上复选框Enable annotation

（3）java编译版本选8，setting-->Build-->Compliler-->Java Compliler，项目名后面选8

（4）File Type过滤器（可以不设置）：setting-->Editor-->File Type，最下面加上 *.idea;*.iml;

dependencyManagement

Maven 使用dependencyManagement 元素来提供了一种管理依赖版本号的方式。
通常会在一个组织或者项目的最顶层的父POM 中看到dependencyManagement 元素。

使用pom.xml 中的dependencyManagement 元素能让所有在子项目中引用一个依赖而不用显式的列出版本号。Maven 会沿着父子层次向上走，直到找到一个拥有dependencyManagement 元素的项目，然后它就会使用这个dependencyManagement 元素中指定的版本号。

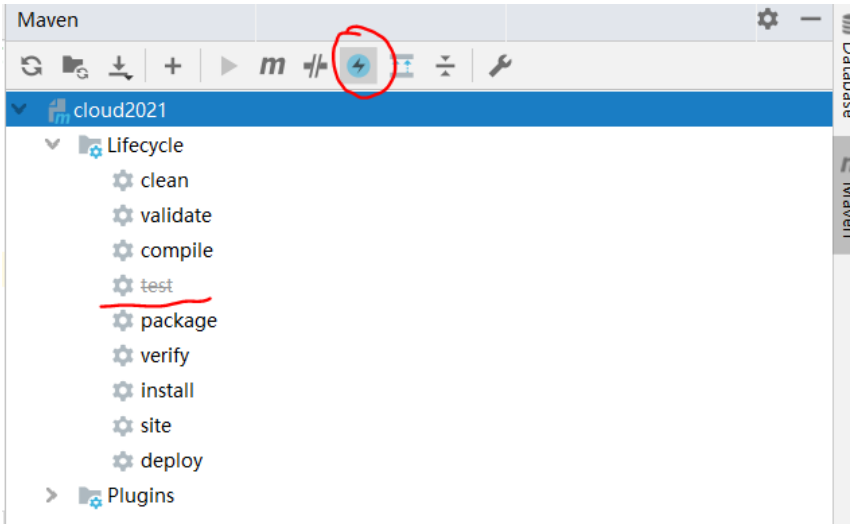
这样做的好处就是：如果有多个子项目都引用同一样依赖，则可以避免在每个使用的子项目里都声明一个版本号，这样当想升级或切换到另一个版本时，只需要在顶层父容器里更新，而不需要一个一个子项目的修改；另外如果某个子项目需要另外的一个版本，只需要声明version就可。

* dependencyManagement里只是声明依赖，***并不实现引入***，因此子项目需要显示的声明需要用的依赖。

* 如果不在子项目中声明依赖，是不会从父项目中继承下来的；只有在子项目中写了该依赖项，并且没有指定具体版本，才会从父项目中继承该项，并且version和scope都读取自父pom;

* 如果子项目中指定了版本号，那么会使用子项目中指定的jar版本。

Maven中跳过单元测试：



创建微服务模块

1.建model

2.改pom

3.写YML

4.主启动

5.业务类

热部署Devtools

开发时使用，生产环境关闭

1.Adding devtools to your project

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>
```

2.Adding plugin to your pom.xml

下段配置复制到聚合父类总工程的pom.xml

```
<build>
    <!--
    <finalName>你的工程名</finalName>（单一工程时添加）
    -->
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <configuration>
                <fork>true</fork>
                <addResources>true</addResources>
            </configuration>
        </plugin>
    </plugins>
</build>
```

3.Enabling automatic build

File -> Settings(New Project Settings->Settings for New Projects) ->Complier

下面项勾选

Automatically show first error in editor
Display notification on build completion
Build project automatically
Compile independent modules in parallel

4.Update the value of

键入Ctrl + Shift + Alt + /，打开Registry，勾选：

compiler.automake.allow.when.app.running

actionSystem.assertFocusAccessFromEdt

5.重启IDEA

RestTemplate

RestTemplate提供了多种便捷访问远程Http服务的方法，是一种简单便捷的访问restful服务模板类，是Spring提供的用于访问Rest服务的客户端模板工具集

使用：

使用restTemplate访问restful接口非常的简单粗暴无脑。
(url, requestMap, ResponseBean.class)这三个参数分别代表。
REST请求地址、请求参数、HTTP响应转换被转换成的对象类型。

Eureka

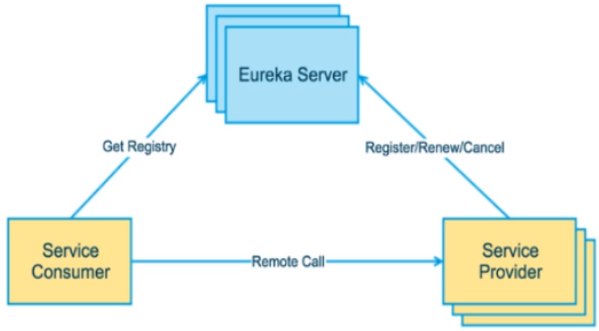
基础知识

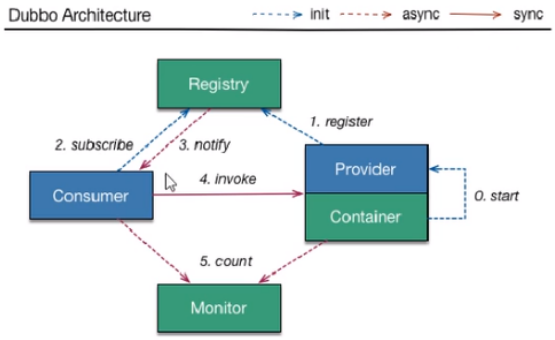
什么是服务注册与发现

Eureka采用了CS的设计架构，Eureka Sever作为服务注册功能的服务器，它是服务注册中心。而系统中的其他微服务，使用Eureka的客户端连接到 Eureka Server并维持心跳连接。这样系统的维护人员就可以通过Eureka Server来监控系统中各个微服务是否正常运行。

在服务注册与发现中，有一个注册中心。当服务器启动的时候，会把当前自己服务器的信息比如服务通讯地址等以别名方式注册到注册中心上。另一方(消费者服务提供者)，以该别名的方式去注册中心上获取到实际的服务通讯地址，然后再实现本地RPC调用。RPC远程调用框架核心设计思想：在于注册中心，因为使用注册中心管理每个服务与服务之间的一个依赖关系(服务治理概念)。在任何RPC远程框架中，都会有一个注册中心存放服务地址相关信息(接口地址)

下左图是Eureka系统架构，右图是Dubbo的架构，请对比





Eureka包含两个组件:Eureka Server和Eureka Client

Eureka Server提供服务注册服务

各个微服务节点通过配置启动后，会在EurekaServer中进行注册，这样EurekaServer中的服务注册表中将会存储所有可用服务节点的信息，服务节点的信息可以在界面中直观看到。

EurekaClient通过注册中心进行访问

它是一个Java客户端，用于简化Eureka Server的交互，客户端同时也具备一个内置的、使用轮询(round-robin)负载算法的负载均衡器。在应用启动后，将会向Eureka Server发送心跳(默认周期为30秒)。如果Eureka Server在多个心跳周期内没有接收到某个节点的心跳，EurekaServer将会从服务注册表中把这个服务节点移除（默认90秒)