

Spring Cloud Alibaba现阶段版本的特殊性

现在的Spring Cloud Alibaba由于没有纳入到Spring Cloud的主版本管理中，所以我们需要自己去引入其版本信息，比如之前教程中的例子：

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>Finchley.SR1</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-alibaba-dependencies</artifactId>
      <version>0.2.1.RELEASE</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

而不是像以往使用Spring Cloud的时候，直接引入Spring Cloud的主版本（Dalston、Edgware、Finchley、Greenwich这些）就可以的。我们需要像上面的例子那样，单独的引入spring-cloud-alibaba-dependencies来管理Spring Cloud Alibaba下的组件版本。

由于Spring Cloud基于Spring Boot构建，而Spring Cloud Alibaba又基于Spring Cloud Common的规范实现，所以当我们使用Spring Cloud Alibaba来构建微服务应用的时候，需要知道这三者之间的版本关系。

下表整理了目前Spring Cloud Alibaba的版本与Spring Boot、Spring Cloud版本的兼容关系：

Spring Boot	Spring Cloud	Spring Cloud Alibaba
2.1.x	Greenwich	0.9.x
2.0.x	Finchley	0.2.x
1.5.x	Edgware	0.1.x
1.5.x	Dalston	0.1.x

2.4之前的分组配置

先来看看2.4版本之前的分组配置，我们用下面这个例子来介绍：

```
spring:
  profiles:
    active: "dev"

---
spring.profiles: "dev"
spring.profiles.include: "dev-db,dev-mq"

---
spring.profiles: "dev-db"

db: dev-db.didispace.com

---
spring.profiles: "dev-mq"

mq: dev-mq.didispace.com
```

dev配置中使用了spring.profiles.include来引入其他配置信息

2.4的分组配置

在升级spring boot到2.4之后，再启动之前的应用，你会发现配置就没有生效了，这里不仅是因为spring.profiles失效的原因，即使我们将其都修改为spring.config.activate.on-profile，也依然无法激活dev-db和dev-mq的配置。因为在2.4版本之后，我们需要使用spring.profiles.group来配置了，同时组织结构也发生了变化。

```
spring:
  profiles:
    active: "dev"
  group:
    "dev": "dev-db,dev-mq"
    "prod": "prod-db,prod-mq"

---
spring:
  config:
    activate:
      on-profile: "dev-db"

db: dev-db.didispace.com

---
```

```
spring:
  config:
    activate:
      on-profile: "dev-mq"
```

```
mq: dev-mq.didispace.com
```

```
spring:
  config:
    activate:
      on-profile: "prod-db"
```

```
db: prod-db.didispace.com
```

```
spring:
  config:
    activate:
      on-profile: "prod-mq"
```

```
mq: prod-mq.didispace.com
```

可以看到，在2.4版本的配置中，不同环境的配置定义都在第一段默认配置中了，所有的环境定义都转移到了spring.profiles.group的key字段（上面配置了dev和prod），value字段则代表了每个环境需要加载的不同配置分组。

RESTful API具体设计如下：

请求类型	URL	功能说明
GET	/users	查询用户列表
POST	/users	创建一个用户
GET	/users/id	根据id查询一个用户
PUT	/users/id	根据id更新一个用户
DELETE	/users/id	根据id删除一个用户

