Dubbo 官方文档:

用户指南 >> 示例 >> 启动时检查

启动时检查

(+)(#)

- ☑ Dubbo缺省会在启动时检查依赖的服务是否可用,不可用时会抛出异常,阻止Spring初始化完成,以便上线时,能及早发现问题,默认check=true。
- ⚠ 如果你的Spring容器是懒加载的,或者通过API编程延迟引用服务,请关闭check,否则服务临时不可用时,会抛出异常,拿到null引用,如果check=false,总是会返回引用,当服务恢复时,能自动连上。

可以通过check="false"关闭检查,比如,测试时,有些服务不关心,或者出现了循环依赖,必须有一方先启动。

关闭某个服务的启动时检查:(没有提供者时报错)

```
<dubbo:reference interface="com.foo.BarService" check="false" />
```

关闭所有服务的启动时检查:(没有提供者时报错)

```
<dubbo:consumer check="false" />
```

关闭注册中心启动时检查:(注册订阅失败时报错)

```
<dubbo:registry check="false" />
```

也可以用dubbo.properties配置:

```
dubbo.properties

dubbo.reference.com.foo.BarService.check=false
dubbo.reference.check=false
dubbo.consumer.check=false
dubbo.registry.check=false
```

也可以用-D参数:

```
java -Ddubbo.reference.com.foo.BarService.check=false
java -Ddubbo.reference.check=false
java -Ddubbo.consumer.check=false
java -Ddubbo.registry.check=false
```

注意区别

- dubbo.reference.check=false,强制改变所有reference的check值,就算配置中有声明,也会被覆盖。
- dubbo.consumer.check=false,是设置check的缺省值,如果配置中有显式的声明,如:
 dubbo:reference check="true"/>,不会受影响。
- dubbo.registry.check=false,前面两个都是指订阅成功,但提供者列表是否为空是否报错,如果注册订阅失败时,也允许启动,需使用此选项,将在后台定时重试。

引用缺省是延迟初始化的,只有引用被注入到其它Bean,或被getBean()获取,才会初始化。如果需要饥饿加载,即没有人引用也立即生成动态代理,可以配置:

<dubbo:reference interface="com.foo.BarService" init="true" />