# 概述

# 知识点

## Spring Schema

参照Demo学习

## Dubbo SPI

参考官网样例

<http://dubbo.apache.org/zh-cn/blog/introduction-to-dubbo-spi.html>

博客

<https://cxis.me/2017/02/18/Dubbo%E4%B8%ADSPI%E6%89%A9%E5%B1%95%E6%9C%BA%E5%88%B6%E8%AF%A6%E8%A7%A3/>

### 概念

|  |  |
| --- | --- |
| 概念 | 描述 |
| 扩展点 | 被注解SPI标注的接口 |
| 扩展 | 扩展点的具体实现 |
| 扩展实例 | 某个扩展的具体实例（已经在内存被创建） |
| 扩展自适应实例 | 一种特殊的扩展实例，其扩展的具体实现是通过自动生成的。 |
| 扩展别名 | 一个扩展的别名，位于配置文件中 |
| ExtensionLoader | dubbo扩展机制的“调度中心”，类似于JDK SPI的ServiceLoader，负责扩展的加载和生命周期 |
| @SPI | 标注注解，用于指定扩展点，没有标注该注解的接口被ExtensionLoader加载会报异常 |
| @Adaptive |  |

### 原理

原理参照Demo逐步调试代码。

# 服务调用过程

参考官网<https://dubbo.apache.org/zh-cn/docs/source_code_guide/service-invoking-process.html>

# 模块

## 注册

参考：<https://juejin.im/post/5d5cc544e51d4561df7805a9>

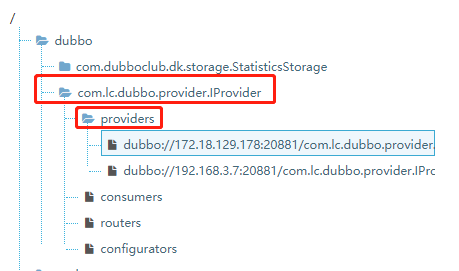
关键字：

AbstractRegistry—内存缓存和文件缓存，文件名格式dubbo-registry-${application.name}-${register\_address}.cache

FailbackRegistry—失败重试，重试机制就是一个定时线程池不断遍历配置的注册中心地址进行注册

### Zookeeper

在学习以Zookeeper作为注册中心之前，首先以一个例子开头，以此对zookeeper有个直观感受。同一个服务提供者，部署在两台服务器上并以同一个zookeeper作为注册中心时，zookeeper的树形结构如下所示：



可以看出dubbo提供中在注册服务时，提供的内网ip，不会提供公网ip。

# 参考

官网https://dubbo.incubator.apache.org/

API官网<https://dubbo.incubator.apache.org/books/dubbo-user-book-en/preface/>

Dubbo介绍<http://www.baeldung.com/dubbo>

Zookeeper简介<https://www.w3cschool.cn/zookeeper/zookeeper_installation.html>

博客<https://juejin.im/post/5bb18c8e5188255c6140de76>

<https://segmentfault.com/a/1190000016741532>（推荐）