



链滴

SPI-Service Provider Interface

作者: [eddy](#)

原文链接: <https://ld246.com/article/1486797924869>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

今天主要是结合代码简单介绍下SPI的功能以及SPI的应用。

SPI

SPI-Service Provider Interface的简写，对服务的一种扩展，可以对已知的服务进行实现并以类似插的方式进行加载，供服务定义方进行调用。

规范--服务的扩展

1. 实现服务接口
2. 在工程META-INF目录下新增services文件夹，如：META-INF/services
3. 在services文件夹内新增以接口全路径为文件名的文件，如：META-INF/services/com.example.interfaceName
4. 在文件内写上接口的实现类的全路径，如：com.example.impl.ClassName

规范--服务加载方

1. 通过 `java.util.ServiceLoader#load(java.lang.Class)` 方法来加载服务的实现。
2. load方法返回ServiceLoader，实现了Iterable，可以直接for-each来遍历服务的实现。

示例

- 利用三方jar包来利用注解生成META-INF/services, 注解: `@MetaInfServices`

```
org.kohsuke.metainf-services metainf-services
```

服务提供方:

```
@MetaInfServices
public class Say implements SayHello {

    public String say() { return "hello"; }}
```

服务调用方:

```
ServiceLoader services = ServiceLoader.load(SayHello.class);
for (SayHello sayHello : services) {
    sayHello.say();
}
```

示例代码地址: <https://github.com/Justice-love/spi>

问题

- spring boot 的auto-config和spring schema的扩展是使用的类似SPI的方式进行加载的还是使用O GI的方式呢?
- SPI和OSGI的差别?

原文地址<http://justice-love.com/SPI/>