\* web service学习笔记

\*

\* Web service是一个平台独立的，语言独立的，松耦合的，自包含的、基于可编程的web的应用程序，

\* 可使用开放的XML标准来描述、发布、发现、协调和配置这些应用程序，用于开发分布

\* 式的互操作的应用程序，是构建SOA应用的核心技术。

\* 目前两种主要的技术类型

\* 1、soap-based web service

\* 2、restful-based web service

\*

\* soap-based web service介绍

\* 这是一种目前阶段比较成熟的技术，有一整套的规范及标准

\* WSDL

\* web服务描述语言，以标准的方式向潜在的用户说明服务接口提供的功能，服务的地址

\* 交互时信息传输协议及信息的格式及编码，消息的交换模式，消息的数据类型及服务的其他

\* 非功能性特性（安全、事务、可靠消息传输、寻址等）

\* UDDI

\* 统一发现、描述和集成，一组规范，用于提供服务的企业把自己提供的服务注册到注册中心

\* 用户可以到注册中心查询该企业所提供的服务

\* ws-policy framework

\* 这是一个用于指定web服务中的实体的能力及特性的框架，web服务中每个特定域都是由一组策略

\* 来描述，这个框架指定策略的描述语言（策略表达式）及怎么把策略附加到ws中

\* 一个常见的策略的格式

\* 1 <wsp:policy>

\* 2 <wsp:ExactlyOne> 这个元素可以包含一个或是多个策略选择项，如多个<wsp:all>

\* 3 <wsp:All> 这个元素可以包含多个策略断言

\* 4 <sp:dd/> 策略断言

\* <.....>

\* 5 </wsp:All>

\* 6 <wsp:All>

\* 7 </wsp:All>

\* 8 <wsp:All>

\* 9 </wsp:All>

\* 10 </wsp:ExactlyOne>

\* 11 </wsp:policy>

\* 上面的xml片段代表了一个完整的策略

\* web 服务的每个特性都是用一个策略断言来描述的

\* 策略断言通过策略运算符组合在一起形成策略选择项

\*

\*

\* ws-I\*规范族

\* ws-I\*目前阶段主要包括,ws-security、ws-rm、ws-transaction、ws-addressing等

\* 只要是依照ws-policy的规范实现特定领域的web服务特性

\* SCA

\* ：一组规范。规定了服务的部署组合规范

\* SDO ： 规定了以统一的方式操作web服务中涉及的各种数据源数据的规范

\* java对该技术的支持

\* java语言中提供了标准spi,供各厂商实现，使用jax-ws开发web service可以使应用与具体实现技术独立

\* 目前主流实现

\* 1、axis

\* 2、cxf

\* 3、metro

\* 三种实现特性比较

\* 比较维度：性能、易开发测试

\* 性能：metro最好，cxf次之，axis最差

\* 易开发测试： metro最好，cxf次之，axis最差

\*

\* restful介绍

\* REST ( REpresentational State Transfer )

\* State Transfer 为 "状态传输" 或 "状态转移 "，Representational 中文有人翻译为"表征"、"具象"，

\* 合起来就是 "表征状态传输" 或 "具象状态传输" 或 "表述性状态转移"，不过，一般文章或技术文件都

\* 比较不会使用翻译后的中文来撰写，而是直接引用 REST 或 RESTful 来代表，因为 REST 一整个观念，

\* 想要只用六个中文字来完整表达真有难度。

\*

\* rest一词最早出现于一片博士论文《Architectural Styles and the Design of Network-based Software Architectures》

\* 从论文的名称就能看出rest的特点

\* 1、架构样式

\* 2、软件架构

\* 3、基于网络的

\* rest描述的是一种基于网络的软件架构的设计方式

\* rest把网络上的所有对象看作是资源，资源通过uri来标识 ，通过用HTTP的4种方法GET、POST、PUT、DELETE来实现对资源的操作

\*

\* rest遵循web标准，完全依赖于http协议，是http使用方式标准的典型的范例，所以他天然的

\* 继承了http的优缺点,易于开发、部署、测试，无状态

\*

\* rest目前的状态

\* 1、还没有形成统一的标准，成熟度不够

\* 2、rest基于请求-响应式，无状态，故不适用对会话及事务有要求的应用场景，这种方式的特点使得我们易于实现容错、缓存

\* 3、rest响应消息格式可以是xml,json,消息比较小，即使使用xml返回也比基于soap-based方式传输的数据要小，减轻了网络负载

\* 4、soap-based中ws-policy家族中所描述的web服务特性rest还不能很好的支持