



哈尔滨工业大学 国家示范性软件学院

第二章 软件服务开发技术

杨大易

2021/12/14



本章内容



1. XML-RPC
2. SOAP Service (WSDL、SOAP、UDDI)
3. RESTful Web Service

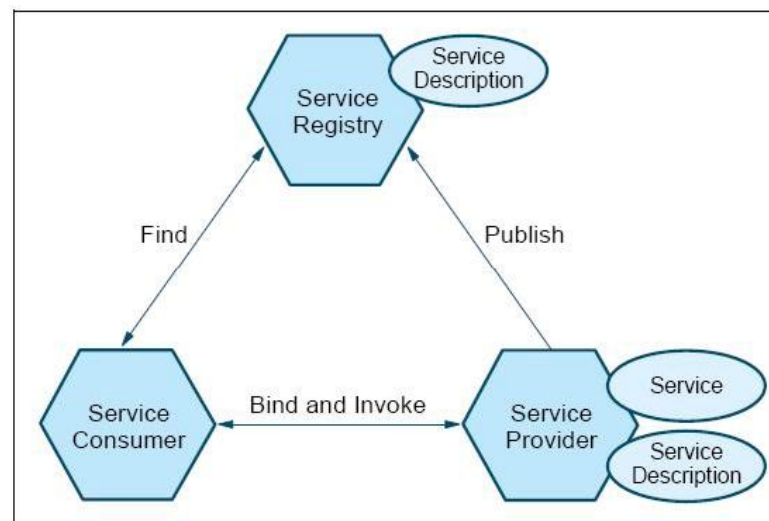


3.1 SOAP Service



❖ Web Service

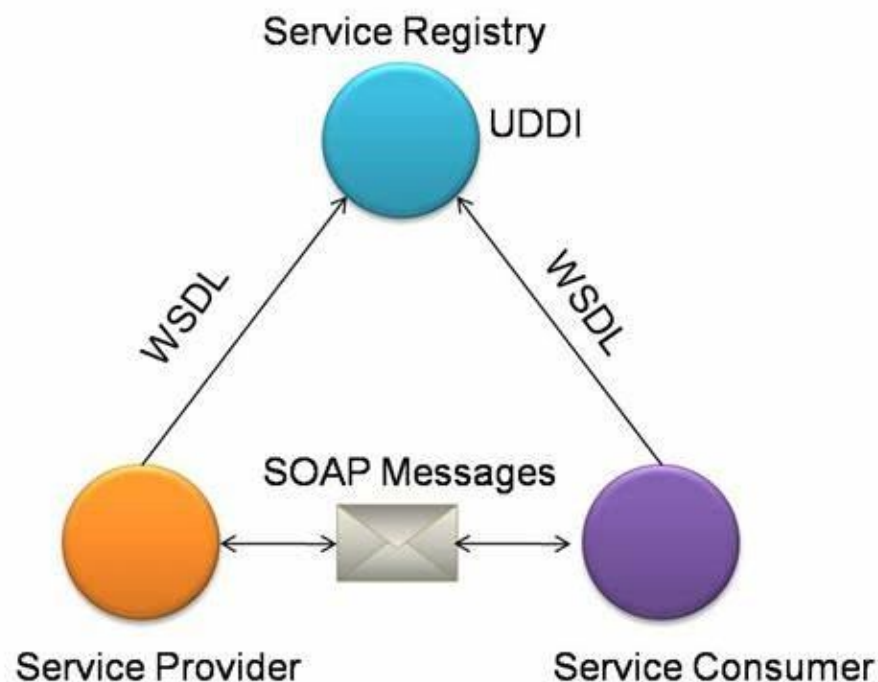
- 是应用程序组件
- 使用开放协议进行通信
- 独立的并可自我描述
- 可被其他应用程序使用
- XML 是 Web Service 的基础



3.1 SOAP Service



- ❖ **SOAP**：用于访问 Web 服务的协议。
- ❖ **WSDL**：描述服务的基本属性
 - 服务所提供的操作
 - 如何访问服务
 - 服务位于何处（URL）
- ❖ **UDDI**：通用的描述、发现以及整合。



3.1 SOAP Service



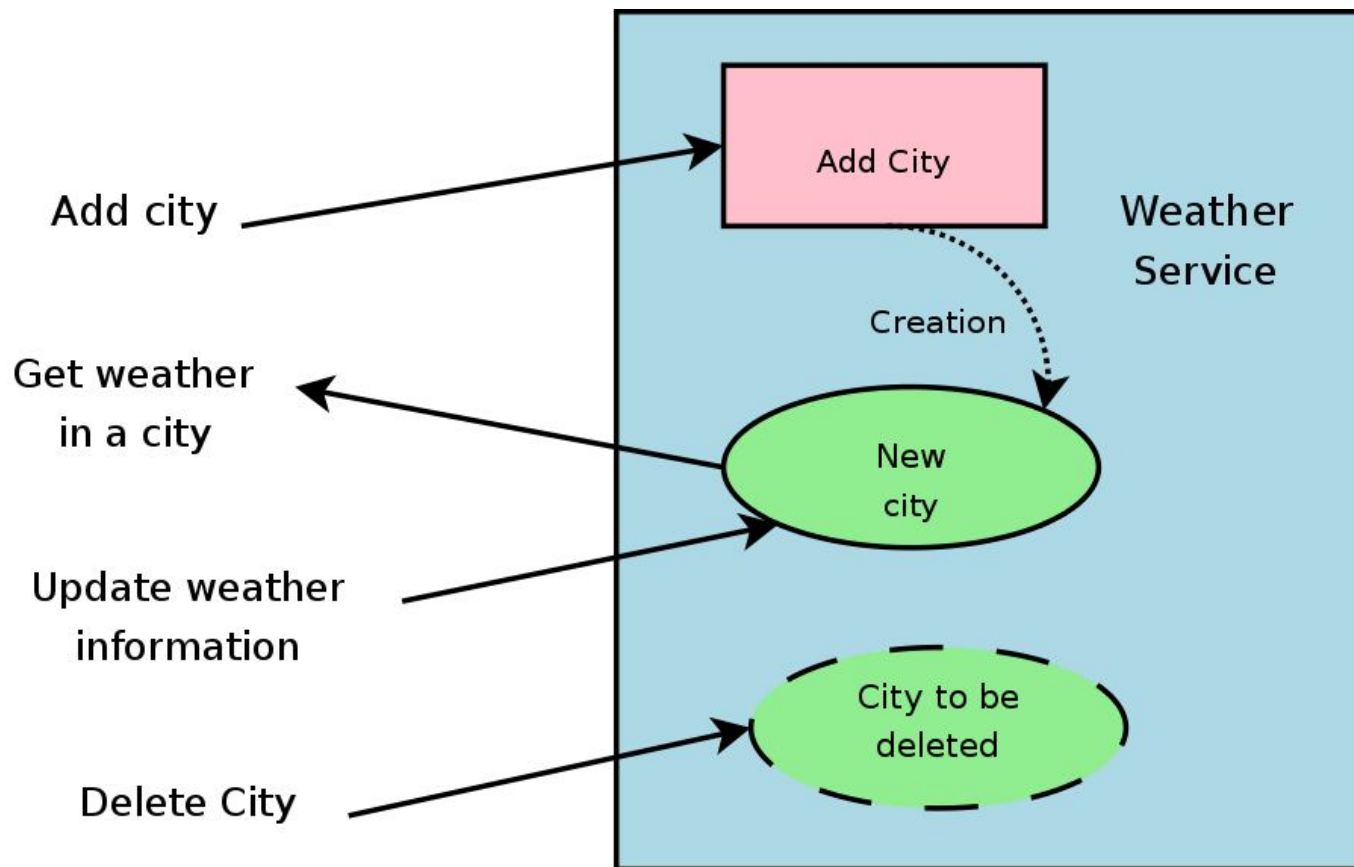
- ❖ 为什么用XML封装数据？
- ❖ 为什么不使用HTTP？
- ❖ 为什么使用URL，而不是URI？
- ❖ 2000年，Roy Fielding在博士论文中提出了REST
(Representational State Transfer)



3.2 REST的特性



❖ 一个Web服务应用的例子



3.2 REST的特性



- ❖ 客户端使用**GET**、**POST**、**PUT**、**DELETE**，4个表示操作方式的动词对服务端资源进行操作；
- ❖ 使用**URI**来定位每个资源；
- ❖ 资源的表现形式是**HTML**或**XML**；
- ❖ 客户端与服务端之间的交互在请求之间是无状态的。
- ❖ **REST**是一种网络应用程序的设计风格。



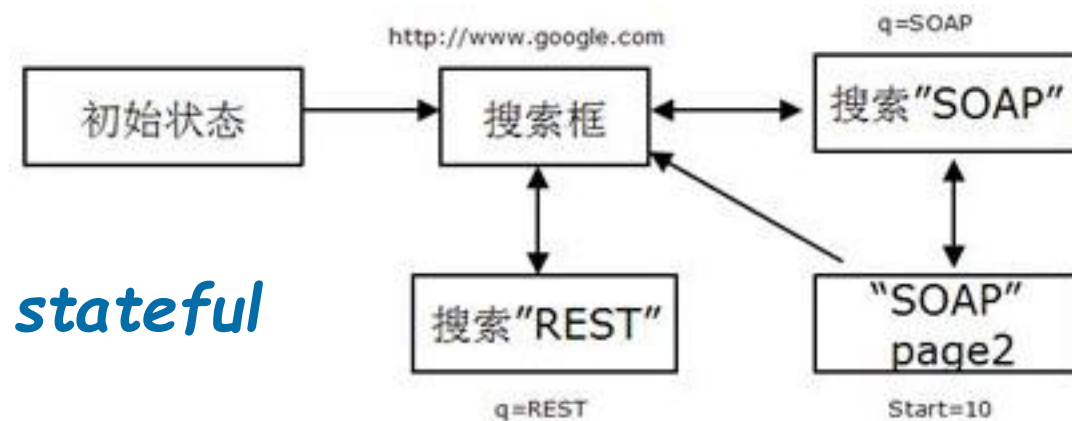
3.2 REST的特性



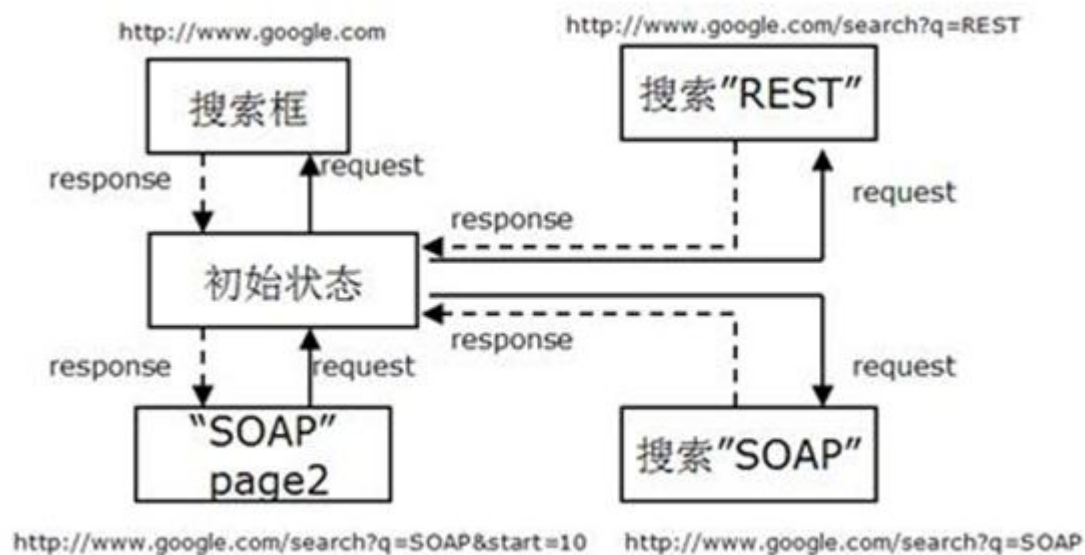
- ❖ **REST用URI来代表资源**
- ❖ **每个资源必须至少有一个URI**
- ❖ **URI应具有良好的结构及描述性（易于阅读）**
 - **`http://www.ex.com/software/releases/latest.tar.gz`**
 - **`http://www.ex.com/search/cs578`**
 - **`http://www.ex.com/sales/2012/Q1`**
 - **`http://www.ex.com/relationships/Alice;Bob`**



3.2 REST的特性



stateless



3.2 REST的特性



❖ REST的无状态性

- 每一个请求都必须是独立的，请求之间是完全分离的；
- 服务端没有保存客户端的状态信息，所以客户端发送的请求必须包含有能够让服务器理解请求的全部信息，包括自己的状态信息；
- 改善了分布式系统的可见性、可靠性以及可伸缩性，同时有效的降低了客户端与服务端之间的交互延迟；
- 无状态的请求有利于实现负载均衡，在分布式web系统下，有多个可的服务端，每个服务端都可以处理客户端发送的请求。



3.3 REST统一的操作接口



CRUD Operations

4 main HTTP methods

	Verb	Noun
Create (Single)	POST	Collection URI
Read (Multiple)	GET	Collection URI
Read (Single)	GET	Entry URI
Update (Single)	PUT	Entry URI
Delete (Single)	DELETE	Entry URI



3.3 REST统一的操作接口



3.3 REST统一的操作接口



- ❖ **GET:** 请求指定的资源。
- ❖ **POST:** 向指定资源提交数据，请求服务器进行处理。
- ❖ **PUT:** 向指定资源位置上传其最新内容。
- ❖ **DELETE:** 请求服务器删除所指定的资源。



3.4 REST的优点



❖ RPC Web Service

- 少量的URI，大量的操作方法
 - `musicPort.getRecordings("beatles")`
- 用HTTP传输SOAP消息

❖ RESTful Web Service

- 大量的资源（URI），统一固定的操作
 - `GET /music/artists/beatles/recordings`
- 将HTTP协议利用到了极致



3.4 REST的优点



❖ RESTful Web Service开发更加简便

- 客户端不需要使用专门的API
- 可基于标准HTTP协议
- 可以使用浏览器进行调试

❖ 统一性

- 使用URI对资源进行统一描述
- 使用统一的HTTP操作接口

❖ 脚本语言更加友好



3.4 REST的优点



- ❖ **REST**提供了一种面向资源架构（**ROA**）的系统构造方案
 - 可寻址：资源通过**URI**暴露，可被外界访问；
 - 无状态：服务端不保存状态，易于提升应用的规模，具有更高的可靠性；
 - 统一接口：统一性使系统具有更好的兼容性；
 - 连通性：资源通过其表示可以彼此连接。



3.4 REST的优点



❖ 知名的RESTful Web Service

- Amazon S3 (Simple Storage Service)
- Yahoo!
- 提供Atom发布协议及其变型的服务



3.5 开发实例



❖ 物流接口服务

❖ 项目要求

- 以服务的形式发布物流数据信息，可供互联网上的其他应用系统请求。

❖ 功能需求

- 根据快递单号查询物流信息
- 根据快递单号更新物流信息
- 删除快递单
-



3.5 开发实例



❖ RESTful Web Service开发步骤

1. 分析业务需求
2. 定义资源、访问方法
3. 描述资源
4. 选择媒体类型
5. 编码实现
6. 发布



3.5 开发实例



1. 分析业务需求

- 根据快递单号查询物流信息
- 根据快递单号更新物流信息

2. 定义资源、访问方法

- 快递单信息 GET
- 更新物流信息 POST

3. 描述资源

- 快递单信息 `flowinfo/orderid`
- 更新物流信息 `flowadd/orderid/flowinfo`



3.5 开发实例



4. 选择媒体类型

- JSON

5. 编码实现

- 服务端实现
- 客户端实现

6. 发布



3.6 软件服务开发技术对比



❖ **CORBA** (Common Object Request Broker Architecture)

- 公共对象请求代理体系结构，分布的、可以互操作的对象利用**ORB**构造可以互操作的应用

❖ **DCOM** (Distributed Component Object Model)

- 分布式组件对象模型标准，支持在局域网、广域网甚至Internet上不同计算机的对象之间的通讯



3.6 软件服务开发技术对比



❖ XML-RPC

- 远程过程调用协议，通过HTTP传输XML来实现远程过程调用的RPC
- 兼容性更好，能够跨域不同的操作系统、不同的编程语言进行远程过程调用
- 适合有一定接口要求或安全性要求的服务

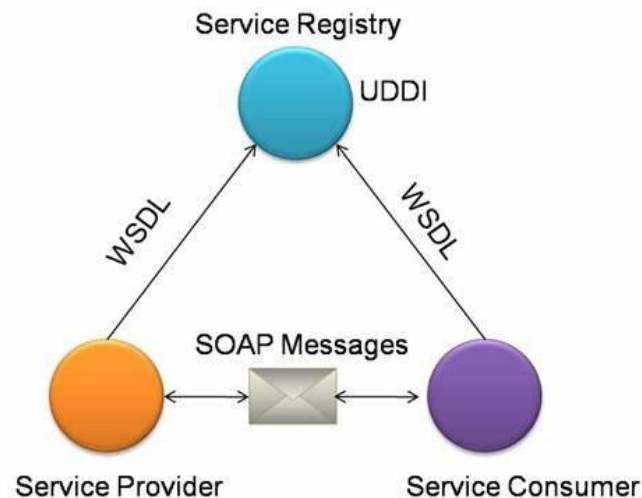


3.6 软件服务开发技术对比



❖ SOAP Service

- 一种跨编程语言和跨操作系统的远程调用技术
- 平台独立的，低耦合的，基于Web的应用程序
- 可使用开放的XML来描述、发布、发现、协调和配置这些应用程序



3.6 软件服务开发技术对比



❖ RESTful

- 一种软件架构风格，而不是标准，只是提供了一组设计原则和约束条件
- 结构简单，易于开发
- 适合安全要求不高的资源型信息发布服务





哈尔滨工业大学 国家示范性软件学院

谢谢！