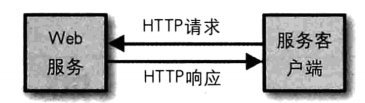
# Web服务入门

## 杂项

### 定义

Web Service；包括服务和客户端（消费者、请求者），是一种网络化的应用程序，通常通过HTTP/HTTPS协议交付；是一种分布式的软件系统，其组件可以在不同的物理设备上部署和执行。

### 会话模式



1. 单向模式（one-way）：客户端到服务器的无服务器的响应的模式
2. 通知模式（notification）：服务器到客户端的无客户端响应的模式
3. **请求/响应（request/response）：客户端发出请求期望服务器响应；**
4. 要求/响应（solicit/response）：服务器发出请求期望客户端响应；
5. 复杂模式：由请求/响应和要求/响应组成

### 风格

1. SOAP：一种XML语言，语法指定了一个SOAP文档必须具备的结构；客户端请求发送SOAP消息，服务端响应发送SOAP消息；SOAP有标准、工具包和丰富的软件库；
2. REST：将HTTP视为一个API，作为一个传输基础设施，同时作为一套设计服务请求和服务响应的指南；REST没有正式的标准，较少的工具和软件库；**通过HTTP传送的基于SOAP的服务可以看作是REST式服务的一种特例**。

### 跨语言

1. 无论是SOAP服务还是REST服务，客户端很少是一个网页浏览器，而通常是一个没有图形界面的应用程序；
2. Web服务和客户端尽管分别由不同编程语言编写，但仍然可以互相调用；
3. 存在一个中间层处理服务和客户端语言之间的不同数据类型：XML和JSON（JavaScript Object Notation，JavaScript对象符号）都是数据交换格式

### 区别于其他分布式应用软件系统

1. 开放式架构：

Web服务的部署方式采用工业标准，独立于特定供应商协议，像已经广泛应用和易于理解的HTTP和XML技术。基于现有网络技术、数据格式化技术、安全技术，以及其他已经存在的基础设施，Web服务可以有效降低成本投入，并且可以进一步促进不同服务资源之间的互相整合利用。

1. 语言透明：

Web服务和客户端尽管分别由不同编程语言编写，但仍然可以互相调用。像C/C++、C#、Java、Perl、Python、Ruby等计算机编程语言，以及其他库文件、工具类，甚至是一些提供Web服务支持的框架。

1. 模块化设计：

Web服务在设计上可以模块化，这意味着一个新的服务可以通过集成整合或层次化现有的服务而产生。假设有这样一个例子，一个库存跟踪服务通过整合在线订单服务而衍生出一个新的服务，而这个服务可以适时地根据库存的级别自动下达库存订货订单请求。

## 优点

1. 现实中的问题：
   1. 多语言开发；
   2. 多平台部署；
   3. 不同系统间交互
2. Web服务应用而生：
   1. 语言和平台的无关性
   2. 通信协议的无关性：不一定基于HTTP
   3. 通信单元多样化：结构化文本（XML、JSON）、二进制等
   4. 系统集成

## Web服务与SOA

1. Web服务以一种自然且重要的方式为任何SOA系统的核心提供服务；
2. SOA：
   1. SOA是一种思维方式，整合可通过网络访问的服务产生的应用程序是可互操作的；在SOA系统中，作为构成块组件的服务可以被描述为无关联和松散耦合的。
   2. 服务操作是一个函数调用：该函数接受０个或多个参数，并返回０个或多个值（集合）
   3. 无状态（stateless）函数调用：函数调用的返回值仅依赖于传递给它的参数，则该调用是无状态的
   4. SOA环境中，Web服务被区分为**提供者**（providers）和**消费者**（consumers）

## 发展历史

### DCE/RPC

1. DCE（Distributed Computing Environment）：分布式计算环境；RPC（Remote Procedure Call）：远程过程调用
2. 分布式对象系统第一代架构：CORBA（公共对象请求代理架构）、DCOM（微软的分布式COM）、RMI（java的远程方法调用）
3. 分布式对象系统第二代架构：Java EE、.Net

### XML-RPC

1. 轻量级的RPC系统，支持基本数据类型和一些简单的命令
2. 使用XML封送/解封实现语言中立性和传输对HTTP的依赖
   1. 封送：将内存中的对象转换为其它格式，如XML文档；
   2. 解封：指生成一个内存中的对象的逆过程
3. 与DCE/RPC区别
   1. XML-RPC的有效载荷是文本，易于编辑，而DCE/RPC的有效载荷是二进制；
   2. XML-RPC传输使用HTTP而不是一个专有系统，要支持XML-RPC编程语言只需要一个标准的HTTP库，连同生成、解析、转换XML和对其进行其它处理的库；
   3. SOAP是直接从XML-RPC导出的XML语言

### 分布式对象架构：RMI实例

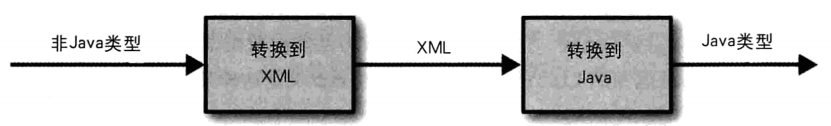
1. java.rmi.Remote



1. 分布式计算中存根和框架具有复杂性
2. 需要专有格式编码的一些对象的序列化和反序列化

### Web服务

1. 客户端和服务交换同等文本文件，如XML等；亦可非文本字节替代
2. 交换的文本可以使用现成的、非专有的，而且通常是使用免费的工具来检查、验证、转化、维护，以及进行其他处理；
3. 客户端和服务各方，只需要一个本地软件库，它绑定特定语言的类型



1. 客户端和服务端的处理过程一样，只需要在本地有可用的库和实用程序，这样依赖，复杂性被分离在端点
2. 相对简单的使用标准传输协议（如HTTP）基于文本的表示形式的消息交换，消息本身是中性的，它们没有对特定的语言的偏见