**Soa(面向服务架构)**

它将应用程序的不同功能单元（称为服务）通过这些服务之间定义良好的**接口**和契约联系起来。接口是采用中立的方式进行定义的，它应该**独立于**实现服务的硬件平台、操作系统和编程语言。

面向服务架构，它可以根据需求通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用。

SOA是一种思想 SOA的精髓是严格的松散耦合 不允许shared memory，不允许back door，不允许直接访问其它服务的数据这都好理解，因为它破坏了封装性，造成了一种内部依赖。

正因为SOA架构实现不依赖于技术，因此能够被各种不同的技术实现。

例如： SOAP, RPC REST DCOM CORBA OPC-UA Web services DDS Java RMI WCF (Microsoft's implementation of web services now forms a part of WCF) Apache Thrift SORCER 因此**REST、SOAP、RPC、RMI、DCOM**等都是SOA的一种实现而已。

Soa大部分基于webservice实现

WebService有两种方式：SOAP和REST

REST（英文：Representational State Transfer，简称REST）

Web 应用程序最重要的 REST 原则是，客户端和服务器之间的交互在请求之间是**无状态**的。从客户端到服务器的每个请求都必须包含理解请求所必需的信息。

**REST 指的是一组架构约束条件和原则。满足这些约束条件和原则的应用程序或设计就是 RESTful。**

在服务器端，应用程序状态和功能可以分为各种资源。资源是一个有趣的概念实体，它向客户端公开。资源的例子有：应用程序对象、数据库记录、算法等等。每个资源都使用 URI (Universal Resource Identifier) 得到一个唯一的地址。所有资源都共享统一的接口，以便在客户端和服务器之间传输状态。使用的是标准的 HTTP 方法，比如 GET、PUT、POST 和 DELETE。

（1）每一个**URI**代表一种资源；

（2）客户端通过四个HTTP动词，对服务器端资源进行操作，实现"表现层状态转化"。

另一个重要的 REST 原则是分层系统，这表示组件无法了解它与之交互的中间层以外的组件。

GET /zoos：列出所有动物园

POST /zoos：新建一个动物园

GET /zoos/ID：获取某个指定动物园的信息

PUT /zoos/ID：更新某个指定动物园的信息（提供该动物园的全部信息）

PATCH /zoos/ID：更新某个指定动物园的信息（提供该动物园的部分信息）

DELETE /zoos/ID：删除某个动物园

GET /zoos/ID/animals：列出某个指定动物园的所有动物

DELETE /zoos/ID/animals/ID：删除某个指定动物园的指定动物

**Soap**

RESTful Web 服务使用标准的 HTTP 方法 (GET/PUT/POST/DELETE) 来抽象所有 Web 系统的服务能力，而不同的是，SOAP 应用都通过定义自己个性化的接口方法来抽象 Web 服务.

所有的 SOAP 消息发送都使用 **HTTP POST** 方法，并且所有 SOAP 消息的 **URI** 都是一样的，这是基于 SOAP 的 Web 服务的基本实践特征。

基于xml传输

清单 5. getUserList SOAP 消息

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <?xml version="1.0" encoding="UTF-8" standalone="no"?>  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">      <soap:Body>          <p:getUserList xmlns:p="http://www.exmaple.com"/>      </soap:Body>  </soap:Envelope> |

客户端将使用 HTTP 的 POST 方法，将上述的 SOAP 消息发送至 http://localhost:8182/v1/soap/servlet/messagerouterURI，SOAP SERVER 收到该 HTTP POST 请求，通过解码 SOAP 消息确定需要调用 getUserList 方法完成该 WEB 服务调用，返回如下的响应：

清单 6. getUserListResponse 消息

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <?xml version="1.0" encoding="UTF-8" standalone="no"?>  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">      <soap:Body>              <p:get                  UserListResponse xmlns:p="http://www.exmaple.com">                  <Users>                  <username>tester<username>                  <username>tester1<username>                  ......                  </Users>                  <p: getUserListResponse >      </soap:Body>  </soap:Envelope> |

微服务架构

CAP原理