最早由Roy T. Fielding于2000年在其博士论文中提出，全名是REpresentational State Transfer(表述性状态转移)。

架构约束：

1、客户-服务端(client-Server) 请求-响应模式

2、无状态(Stateless) 通信的会话状态应该全部由客户端维护

3、缓存(Cache) 响应内容可以在通信链的某处被缓存，以改善网络效率

4、统一接口(Uniform Interface) 通信链的组件之间通过统一的接口相互通信，以提高交互的可见性

5、分层系统(Layered System) 通过限制组件的行为（即，每个组件只能“看到”与其交互的紧邻层），将架构分解为若干等级的层。

6、按需代码(Code-On-Demand 可选) 支持通过下载并执行一些代码（例如Java Applet、Flash或JavaScript），对客户端的功能进行扩展。

关键字：

资源(Resource) 抽象概念

资源的表述(Represetation) 一段对于资源在某个特定时刻的状态的描述，可以在客户端和服务器之间转移(交换)。资源表述的格式有很多，比如HTML/图片/文件/XML/JSON等

状态转移(State Transfer) 在客户端和服务器端之间转移（transfer）代表资源状态的表述，即通过转移和操作资源的表述，来间接操作资源

统一接口(Uniform Interface) ：

7个HTTP方法，GET/POST/PUT/DELETE/PATCH/HEAD/OPTIONS

HTTP头信息(可自定义)

HTTP响应状态码(可自定义)

内容协商机制

缓存机制

客户端身份认证机制

此外，操作语义不能放在消息体中。这样做是为了提高交互的可见性，以便于通信链的中间层实现缓存、审计等功能。

超文本驱动(Hypertext Driven)

特性：

面向资源，可寻址(通过URI)，连通性(各资源通过超链接相互关联)，无状态，统一接口，超文本驱动。