



# RESTful API的设计与开发

moonzwu



# 议程

1. RESTful的含义，目的以及现状
2. 理解RESTful架构
3. RESTful API设计要点
4. 如何调试RESTful API
5. 学习资源

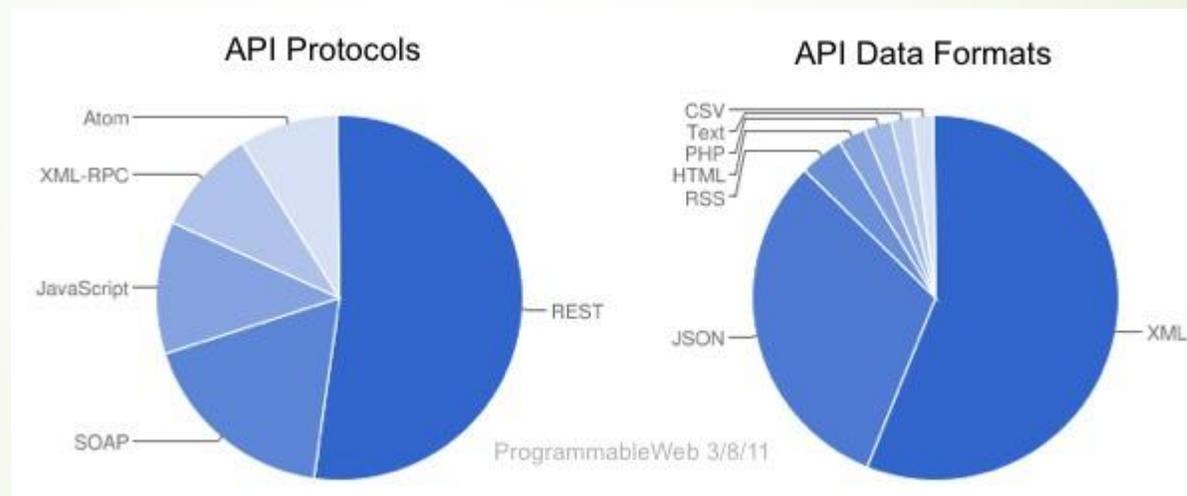
# RESTful的含义，目的以及现状

- ▶ 什么是RESTful ( **Representational State Transfer** )
  - ▶ 一种**软件架构风格**，**设计风格**而不是标准，只是提供了一组**设计原则**和**约束条件**。 - Roy Fielding
- ▶ 用RESTful来做什么
  - ▶ 它主要用于**客户端**和**服务器**交互类的软件。基于这个风格设计的软件可以更**简洁**，更有**层次**，更易于实现**缓存**等机制。

# RESTful的含义，目的以及现状

## RESTful的现状

- 2011年在Internet Web API上RESTful约占**52%**



- 截至目前为止 Internet Web APIs 总数为 12,727, 而支持 REST 与 JSON 约有 8,000, 比例约为 **62%**

# 理解RESTful架构

## ▶ RESTful的架构元素

- ▶ 资源 (resource) : 网络上的一个实体或者说是一个具体信息, 比如一个MP3文件, 一个网页或者一个视频。
- ▶ 统一资源定位符 (URI) : 一个资源的识别符或者说是一个地址, 通过URI你可以定位到特定的资源。
- ▶ 状态转换 (State Transfer) : 客户端与服务器互动的过程, 通常涉及到服务器端数据和状态的变化过程, 比如文件被修改, 访问数量增加等。Http标准中定义的最主要四个动词: GET、POST、PUT、DELETE。它们分别对应四种基本操作:
  - ▶ GET: 用来获取资源
  - ▶ POST: 用来新建资源
  - ▶ PUT: 用来更新资源
  - ▶ DELETE: 用来删除资源

# 理解RESTful架构

RESTful的架构与其他架构的比较

- ▶ 分布式对象（Distributed Objects，简称DO）
  - ▶ 架构实例有CORBA/RMI/EJB/DCOM/.NET Remoting...
- ▶ 远程过程调用（Remote Procedure Call，简称RPC）
  - ▶ 架构实例有SOAP/XML-RPC/Flash AMF ...

特性	DO	RPC	RESTful
跨语言	无	一般	完全跨语言
统一性	无	依赖中间件	基于HTTP标准
耦合性	紧	一般	松
数据流和管道	无	无	支持
交互表达性	基于API接口	基于协议	基于超文本

# RESTful API设计要点

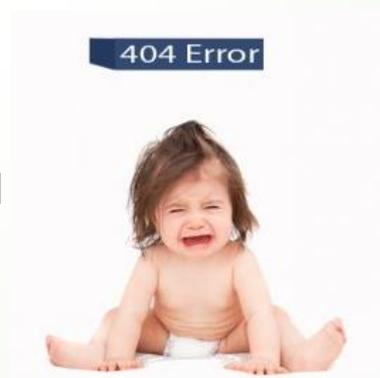
- ▶ 避免在URL里出现动词，使用GET，POST，PUT和DELETE代替



- ▶ ~~http://localhost:8088/rest/api/bmc/add/192.168.1.10~~

- ▶ POST: http://localhost:8088/rest/api/bmc/192.168.1.10

- ▶ 利用Http状态码来反映调用结果
  - ▶ 200 OK, 500 Internal



- ▶ 使用JSON，避免XML作为数据传输格式

# RESTful API设计要点

- ▶ 总是使用Https, 因为你永远不知道谁会使用你的API
- ▶ 版本化RESTful API
  - ▶ `https://api.example.com/v1/`
- ▶ 使用查询参数过滤信息, 因为有时候你并不想要全部的数据
  - ▶ `?limit=10`
- ▶ 尽可能使用Hypermedia API, 在返回的结果中带上下一步的链接

```
{ "link": {  
  "rel": "collection https://www.example.com/zoos",  
  "href": "https://api.example.com/zoos",  
  "title": "List of zoos",  
  "type": "application/vnd.yourformat+json"  
}}
```

# 如何调试RESTful API

使用三种方式来调试RESTful API

- ▶ 使用浏览器进行调试
  - ▶ 在浏览器里面手动输入REST URL进行调试
    - ▶ 优点：直接
    - ▶ 缺点：太麻烦，费事费力，无法模拟form的提交
- ▶ 使用浏览器插件进行调试
  - ▶ Firefox的Firebug，Chrome的Post Man
    - ▶ 优点：简单，可以模拟各种HTTP的请求，可以更改Http头
    - ▶ 缺点：只能在浏览器上工作
- ▶ 使用客户端软件进行调试
  - ▶ Fiddler
    - ▶ 优点：独立的安装包，支持所有的HTTP请求模拟
    - ▶ 缺点：只能在Windows上安装

# 学习资源

- 《RESTful Web APIs》
- HTTP协议: <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- Http状态码列表: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
- 简约的REST设计: <https://speakerdeck.com/akuzemchak/simple-api-development-with-laravel>
- RESTful API设计原则: <http://codeplanet.io/principles-good-restful-api-design/>
- JSON介绍: <http://www.json.org/>
- POCO一个C++的Http库: <http://pocoproject.org/>