HTTP概述与RESTful API

厦门大学信息学院 赵江声

2023-10

目录

Content

厦大赵江声

01

HTTP概述

Hyper Text Transfer Protocol

（超文本传输协议）

1.1 什么是HTTP协议　　1.1.1 概念

参考资料：https://zhuanlan.zhihu.com/p/187541438

HTTP协议是Hyper Text Transfer Protocol（超文本传输协议）的缩写，是用于从万维网（WWW:World Wide Web）服务器传输超文本到本地浏览器的传送协议。

- HTTP是一个基于TCP/IP通信协议来传递数据。
- HTTP是一个属于应用层的面向对象的协议。
- HTTP协议工作于客户端-服务端架构上。
  - 浏览器作为HTTP客户端通过URL向HTTP服务端即WEB服务器发送所有请求；
  - Web服务器根据接收到的请求后，向客户端发送响应信息。

厦大赵江南

1.1.2 HTTP特点

- **简单快速。** 客户向服务器请求服务时，只需传送请求**方法**和**路径**。
- **灵活。** HTTP允许传输**任意类型**的数据对象。
- **无状态。** HTTP协议是**无状态**协议。
- **支持B/S与C/S架构。**

厦大赵江声

1.2 HTTP的路径

uri：统一资源标识符(URI)，uniform resource identifier的缩写，是HTTP简单的、可扩展的、统一的资源标识方式。分为url和urn。

- url
  - Uniform Resource Locator， 统一资源定位符，是互联网上用来标识某一处资源的地址。
  - 样例：百度百科的HTTP词条的url ： https://baike.baidu.com/item/HTTP/243074?fr=ge_ala
- urn
  - Uniform Resource Name，统一资源命名，是通过名字来标识资源。相对于url，urn比较少用。
  - 样例： mailto:zjs@xmu.edu.cn

厦大软工

1.3 HTTP请求的方法

访问url，就是一次http request（请求），如果url的服务可达，会收到一个response。

- HTTP Request 方法一般包括：get、post、put、delete、patch、head、option、connection、trace等。
- HTTP Request 方法的定义主要 考虑安全性和幂等性。
  - 安全性：方法不改变资源本身。
  - 幂等性：用户对于同一操作发起的一次请求或者多次请求的结果是一致的，不会因为多次点击而产生了副作用。

厦大出江南

1.3.1 HTTP请求的方法详解（1）

主要了解get、post、put、delete、patch；

- **GET**
  - 特点：安全、幂等。
  - 说明：从服务器端获取数据，请求body在地址栏上。
  - 作用：获取资源。
- HEAD
  - 特点：安全、幂等。
  - 说明：与get方法类似，但不返回message body内容，仅仅是获得获取资源的部分信息（content-type、content-length）。
  - 作用：RESTful框架中较少使用。
- **POST**
  - 特点：非安全、非幂等。
  - 说明：向服务器端提交数据，请求数据在报文body里；发送一个修改数据的请求，需求数据要重新创建。
  - 作用：用于创建子资源。创建、更新、删除、查询资源均可使用。
- **PUT**
  - 特点：非安全、幂等。
  - 说明：向服务器端提交数据，请求数据在报文body里；发送一个修改数据的请求，需求数据更新（全部更新）。
  - 作用：用于创建、更新资源。

厦大赵江声

1.3.1 HTTP请求的方法详解（2）

- **DELETE**
  - 特点：非安全、幂等。
  - 说明：向服务器端提交数据，请求数据在报文body里；发送一个删除数据的请求。
  - 作用：删除资源。
- OPTIONS
  - 特点：安全、幂等。
  - 作用：用于url验证，验证接口服务是否正常。
- TRACE
  - 特点：安全、幂等。
  - 说明：维基百科“回显服务器收到的请求，这样客户端可以看到（如果有）哪一些改变或者添加已经被中间服务器实现。”
  - 作用：restful框架中较少使用。
- PATCH
  - 特点：非安全、幂等。
  - 说明：向服务器端提交数据，请求数据在报文body里；与PUT类似，发送一个修改数据的请求，区别在于PATCH代表部分更新；后来提出的接口方法，使用时可能去要验证客户端和服务端是否支持；
  - 作用：用于创建、更新资源。局部更新，比如：user对象，只更改了name属性，那么他的其他属性值是不会变的，如果用post，那么其他属性值会被设置为null（全局更新）

1.3.2 常用方法特性表

HTTP Method	安全性	幂等性	解释
GET	安全	幂等	读操作安全，查询一次多次结果一致
POST	非安全	非幂等	写操作非安全，每多插入一次都会出现新结果
PUT	非安全	幂等	写操作非安全，一次和多次更新结果一致
DELETE	非安全	幂等	写操作非安全，一次和多次删除结果一致

知乎 @bigsai

1.3.3 状态码

```
200 OK                          //客户端请求成功
400 Bad Request                 //客户端请求有语法错误，不能被服务器所理解
401 Unauthorized                //请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起
403 Forbidden                   //服务器收到请求，但是拒绝提供服务
404 Not Found                   //请求资源不存在，eg：输入了错误的URL
500 Internal Server Error       //服务器发生不可预期的错误
503 Server Unavailable          //服务器当前不能处理客户端的请求，一段时间后可能恢复正常
```

厦大密匠

1.4 GET 与 POST的区别　1.4.1 基本概念

GET和POST是什么？HTTP协议中的两种发送请求的方法。

- get请求:从指定的资源请求数据，用于获取数据，一般用于搜索排序和筛选之类的操作。
- post请求:向指定的资源提交要被处理的数据，用于将数据发送给服务器，一般用于修改和写入数据。

get请求和post请求本质上就是TCP链接，并无差别。但是由于HTTP的规定和浏览器/服务器的限制，导致他们在应用过程中体现出一些不同。

厦大逆江青

1.4.2 请求过程

- get请求的过程：
  - ① 浏览器请求tcp连接（第一次握手）
  - ② 服务器答应进行tcp连接（第二次握手）
  - ③ 浏览器确认，并发送get请求头和数据（第三次握手，这个报文比较小，所以http会在此时进行第一次数据发送）
  - ④ 服务器返回200 OK响应
- post请求的过程：
  - ① 浏览器请求tcp连接（第一次握手）
  - ② 服务器答应进行tcp连接（第二次握手）
  - ③ 浏览器确认，并发送post请求头（第三次握手，这个报文比较小，所以http会在此时进行第一次数据发送）
  - ④ 服务器返回100 Continue响应
  - ⑤ 浏览器发送数据
  - ⑥ 服务器返回200 OK响应

厦大赵江虎

1.4.3 区别　1.4.3.1 post请求和get请求的区别

- post请求更安全（不会作为url的一部分，不会被缓存、保存在服务器日志、以及浏览器浏览记录中，get请求的是静态资源，则会缓存，如果是数据，则不会缓存）
- post请求发送的数据更大（get请求有url长度限制，http协议本身不限制，请求长度限制是由浏览器和web服务器决定和设置）
- post请求能发送更多的数据类型（get请求只能发送ASCII字符）
- 传参方式不同（get请求参数通过url传递，post请求放在request body中传递）
- get请求产生一个TCP数据包；post请求产生两个TCP数据包（get请求，浏览器会把http header和data一并发送出去，服务器响应200返回数据；post请求，浏览器先发送header，服务器响应100 continue，浏览器再发送data，服务器响应200 返回数据）

1.4.3.2 二者区别划重点

- post请求更安全
  - post不会作为url的一部分，不会被缓存、保存在服务器日志以及浏览器浏览记录中；
  - get请求的是静态资源，则会缓存；如果是数据，则不会缓存
- post请求发送的数据更大
  - get请求有url长度限制，
  - post请求，虽然http协议本身不限制，请求长度限制是由浏览器和web服务器决定和设置
- post请求能发送更多的数据类型
  - get请求只能发送ASCII字符
  - post请求可以发送更多的数据类型
- 传参方式不同
  - get请求参数通过url传递
  - post请求放在request body中传递
- get请求产生一个TCP数据包；post请求产生两个TCP数据包
  - get请求，浏览器会把http header和data一并发送出去，服务器响应200返回数据；
  - post请求，浏览器先发送header，服务器响应100 continue，浏览器再发送data，服务器响应200 返回数据

厦大赵江声

1.4.3.3 其他

阅读资料：https://zhuanlan.zhihu.com/p/114846445

get：类似于敞篷小货车

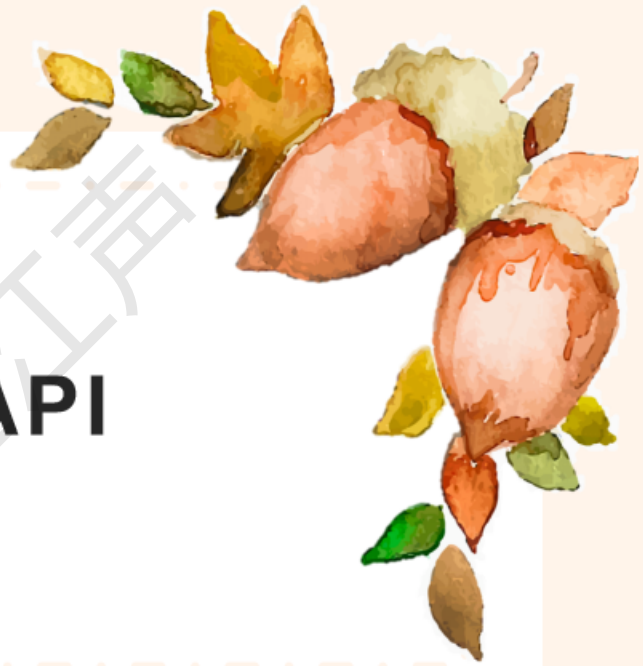
post： 类似于密封的大卡车，送货前一般要送一小部分样品，客户确认后再送大批的货物

厦大赵江喜

02

RESTful API

CRUD



厦大计协

2.1 基本概念

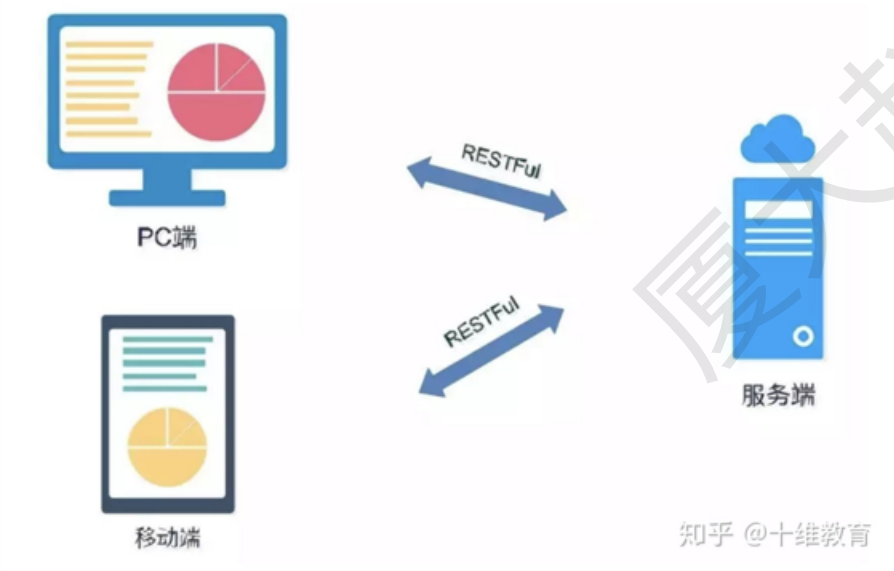
- 参考资料：https://blog.csdn.net/qq_41378597/article/details/85248848

- REST 全称：REpresentational State Transfer，英文翻译过来就是“表现层状态转化”。
  - Resource：资源，即数据。
  - Representational：某种表现形式，比如用JSON，XML，JPEG等；
  - State Transfer：状态变化。通过HTTP动词实现。
- RESTful：用URL定位资源、用HTTP动词（GET、POST、PUT、DELETE）描述操作。
- RESTful API：就是REST风格的API，即REST是一种架构风格，跟编程语言无关，跟平台无关，它采用HTTP做传输协议。

厦大计软

2.2 RESTful架构的优点

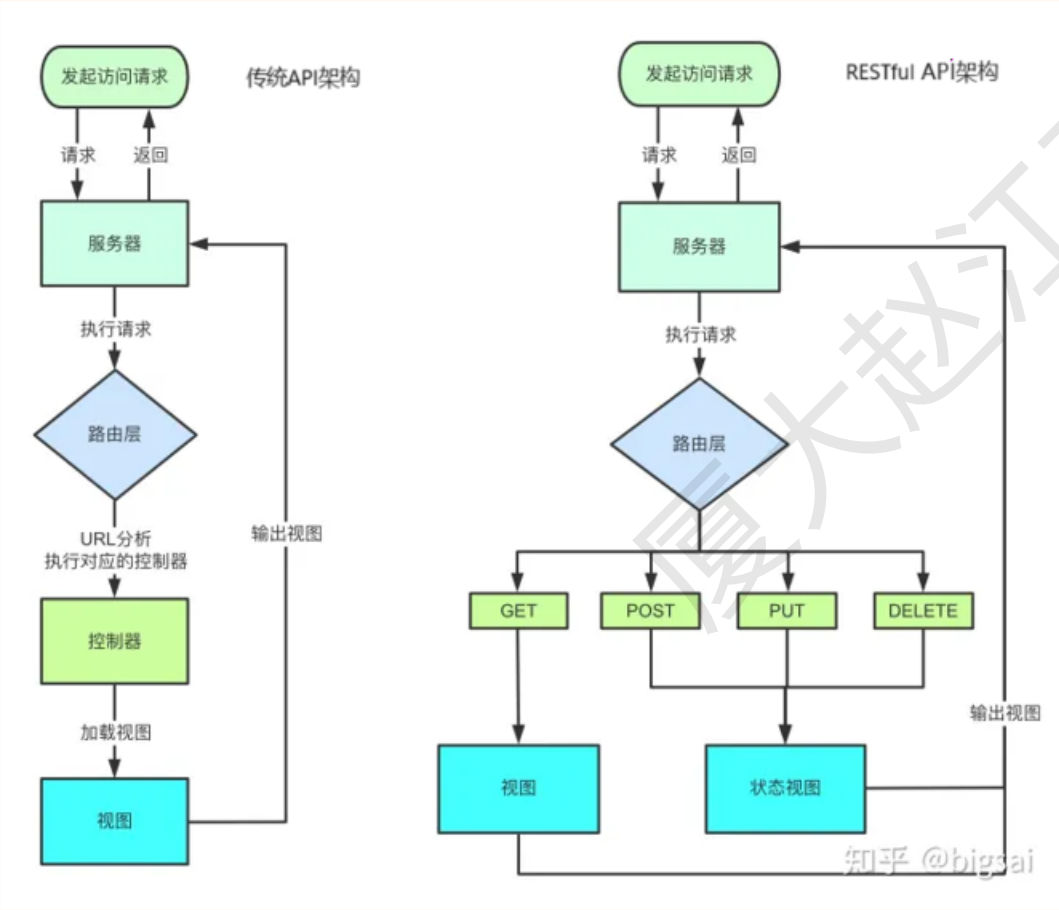
- 项目的分工更加明确
- 实现了前后端的解耦
- 可以将前端和后端部署到不同的服务器上来减轻服务器的压力
- 前后端代码在不同的服务器上，可以提高安全性



厦大赵江声

2.3 RESTful API架构

从请求的流程来看，RESTful和传统的API大致构架如下：



bigsai

2.4 设计原则　2.4.1 标准的REST约束应满足一下6个原则

- 客户端-服务端（Client-Server）：这个更专注客户端和服务端的分离，服务端独立可更好服务于前端、安卓、IOS等客户端设备。
- 无状态（Stateless）：服务端不保存客户端状态，客户端保存状态信息每次请求携带状态信息。
- 可缓存性（Cacheability）：服务端需回复是否可以缓存以让客户端甄别是否缓存提高效率。
- 统一接口（Uniform Interface）：通过一定原则设计接口降低耦合，简化系统架构，这是RESTful设计的基本出发点。
- 分层系统（Layered System）：客户端无法直接知道连接的到终端还是中间设备，分层允许你灵活的部署服务端项目。
- 按需代码（Code-On-Demand，可选）：按需代码允许我们灵活的发送一些看似特殊的代码给客户端例如JavaScript代码。

厦大逆江南

2.4.2 RESTful API设计原则（1）

（1）统一接口

RESTful风格的数据操作CRUD（create,read,update,delete）使用HTTP方法：GET，POST，PUT，DELETE等，这样就统一了数据操作的接口。

- GET（READ）：从服务器取出资源（一项或多项）。
- POST（CREATE）：在服务器新建一个资源。
- PUT（UPDATE）：在服务器更新资源（客户端提供完整资源数据）。
- PATCH（UPDATE）：在服务器更新资源（客户端提供需要修改的资源数据）。
- DELETE（DELETE）：从服务器删除资源。

厦大软江

2.4.2 RESTful API设计原则（2）

（2）以资源为基础

资源可以是一张图片、音乐、一个XML格式、HTML格式或者JSON格式等网络上的一个实体。**JSON是现在最常用的资源表现形式。**

（3）URI

最典型的URI就是URL

（4）无状态

所谓无状态即所有的资源都可以URI定位，而且这个定位与其他资源无关，也不会因为其他资源的变化而变化。

厦大赵江声

2.5 RESTful API 设计规范（1）

（1）URL设计规范

URI = scheme "://" host ":" port "/" path ["?" query]["#" fragment]

厦大赵江声

2.5 RESTful API 设计规范（2）

（2）HTTP动词

- GET /collection：从服务器查询资源的列表（数组）
- GET /collection/resource：从服务器查询单个资源
- POST /collection：在服务器创建新的资源
- PUT /collection/resource：更新服务器资源
- DELETE /collection/resource：从服务器删除资源

厦大数江南

2.5 RESTful API 设计规范（3）

（3）状态码和返回数据

状态码	说明
200	（成功）服务器已成功处理了请求。
201	（已创建）请求成功并且服务器创建了新的资源。
204	（无内容）服务器成功处理了请求，但没有返回任何内容。
301	（永久移动）请求的网页已永久移动到新位置。
302	（临时移动）服务器目前从不同的位置响应请求。
400	（错误请求）服务器不理解请求的语法。
401	（未授权）请求要求身份验证。
403	（禁止）无权限，服务器拒绝请求。
404	（未找到） 服务器找不到请求的资源
408	（超时） 请求超时
422	（验证错误） 请求参数未通过验证
429	（被限制）请求次数过多
500	（服务器内部错误） 服务器遇到错误，无法完成请求。
501	（尚未实施） 服务器不具备完成请求的功能。
502	（错误网关） 服务器作为网关或代理，从上游服务器收到无效响应。
503	（服务不可用） 服务器目前无法使用（由于超载或停机维护）。 通常，这只是暂时状态。
504	（网关超时） 服务器作为网关或代理，但是没有及时从上游服务器收到请求。
505	（HTTP 版本不受支持） 服务器不支持请求中所用的 HTTP 协议版本。

厚大赵江声

2.6 RESTful API 实战（1）

参考资料：https://learn.microsoft.com/zh-cn/aspnet/core/tutorials/min-web-api?view=aspnetcore-7.0&tabs=visual-studio

使用 ASP.NET Core 创建最小 API

（0）目标

API	描述	请求正文	响应正文
`GET /todoitems`	获取所有待办事项	None	待办事项的数组
`GET /todoitems/complete`	获取已完成的待办事项	None	待办事项的数组
`GET /todoitems/{id}`	按 ID 获取项	None	待办事项
`POST /todoitems`	添加新项	待办事项	待办事项
`PUT /todoitems/{id}`	更新现有项	待办事项	None
`DELETE /todoitems/{id}`	删除项	None	无

2.6 RESTful API 实战（2）

（1）创建

- 选择“ASP.NET Core 空”模板
- 检查Program.cs，然后运行

（2）添加 NuGet 包

- 在“工具”菜单中，选择“NuGet 包管理器”>“管理解决方案的 NuGet 包”
- 添加 Microsoft.EntityFrameworkCore.InMemory 包
- 添加 Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore 包

（3）模型和数据库上下文类

（4）添加 API 代码

（5）安装 Postman 以测试应用

厦大林江

谢谢！