1. Restful协议 1

1.1 nodejs-express框架的参数的传递问题 1

2. webservice 协议 2

2.1 简介 2

3. RPC(远程程序调用) 2

4. RMI(远程方法调用) 2

3.1 步骤(以java为例) 2

4.2 相关协议 2

4.2.1 JRMP (relevant to java) 3

4.2.1 Corba (relevant to common) 3

5. SOAP 协议 3

5.1 简介 3

5.2 涉及webservice 3

6. 负载均衡 3

6.1 简介 3

6.2 terminology 3

6.3 CDN(Content-Delivery-NETWORK) 3

7. 代理服务器 3

7.1 正向代理服务器 3

7.1.1 简介 3

7.2 反向代理服务器 3

7.2.1 简介 3

8. 加密/通信 4

简介 4

A. 加密 4

8.1 对称加密算法 (symmetric encryption algorithm) 4

8.1.1 类型 4

7.1.1.1流加密密文 4

8.1.1.2块加密密文 4

8.1.2 相关算法 4

8.2 非对称加密算法(asymmetric encryption algorithm) 4

8.2.1 rsa 4

8.2.1.1 introduction 4

8.2.2 Utilization 4

ssh、https、bitcoin、pgp & gpg 4

8. 3 七层协议-加密协议 4

8.3.1 运输层加密协议 4

8.3.1.1 ssl 4

8.3.1.1 tls 4

8.3.n 应用层 5

8.通信 5

8.1 无线通信 5

8.1.1 简介 5

8.2 有线通信 5

8.3 Duplex 5

8.3.1 Introduction 5

9. 压缩／解压原理 5

9.1 http 使用的压缩方式 5

9.1.1 lossy有损压缩 5

9.1.2 lossless 无损压缩 5

10. 编程 6

10.1 duck typing 6

# 1. Restful协议

## 1.1 nodejs-express框架的参数的传递问题

req.query：存在url中由?进行引导的参数获取

req.params: 存在与url中的参数，如: /http://localhost:8080/api/v1/budget/:id/refresh

常见问题：

跨域问题（cross-origin sharing standard）：

reference： <https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Access_control_CORS>

1. 简介

处于安全考虑， 浏览器会禁止程序（脚本）发起跨域http请求，例如XMLHttpRequest(ajax)和Fetch api 遵循同源策略。对于域名、端口号、协议不同，都有可能引起跨域问题。有些是浏览器直接拦截请求，有些则是根据发想服务器端的试探请求(preflight request)，从而确定是否允许跨域请求。

1.1 简单请求的跨域问题

使用简单请求，无需发送preflight request，去获取服务器的请求处理结果。

简单请求方法有： GET, HEAD, POST

满足简单请求的headers字段：

Accept

Accept-Language

Content-Language

Content-Type: 可选取值为： text/plain， multipart/form-data， application/x-www-form-urlencoded

DPR

Downlink

Save-Data

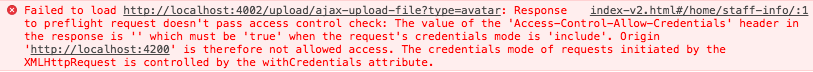
Viewport-Width

Width

1.2 复杂请求的跨域问题

2. preflight request

2.1 简介



请求的credentials mode为include，则 服务器端需要在response中设置头信息 ‘Access-Control-Allow-Credentials’ 为true， 此时’Access-Control-Allow-Origin’必须不能为’\*’

request headers存在的信息：

Access-Control-Request-Headers

Response headers存在的信息

Allow-Control-Allow-Headers

# 2. webservice 协议

## 2.1 简介

目的是实现应用程序之间的通信。 两种程序通信方式： rpc远程过程调用和消息传递。

RPC系统试图达到一种位置上的透明性：服务器暴露出远程对象的接口，而客户端就好像在本地使用的这些对象的接口一样，这样就隐藏了底层的信息，客户端也就根本不需要知道对象是在哪台机器上

三要素： 使用soap、wsdl、uddi框架进行通信

# 3. RPC(远程程序调用)

# 4. RMI(远程方法调用)

reference： <https://en.wikipedia.org/wiki/Java_remote_method_invocation>

## 3.1 步骤(以java为例)

1，客户对象调用客户端辅助对象上的方法

2，客户端辅助对象打包调用信息（变量，方法名），通过网络发送给服务端辅助对象

3，服务端辅助对象将客户端辅助对象发送来的信息解包，找出真正被调用的方法以及该方法所在对象

4，调用真正服务对象上的真正方法，并将结果返回给服务端辅助对象

5，服务端辅助对象将结果打包，发送给客户端辅助对象

6，客户端辅助对象将返回值解包，返回给客户对象

7，客户对象获得返回值

## 4.2 相关协议

### 4.2.1 JRMP (relevant to java)

java remote method protocol

### 4.2.1 Corba (relevant to common)

Common Object Request Broker Architecture, 解决异构分布式环境中的硬件、软件互联方案。

# 5. SOAP 协议

## 5.1 简介

Simple object access protocol(简单对象访问协议)。基于xml的协议。

## 5.2 涉及webservice

参考： <https://baike.baidu.com/item/%E7%AE%80%E5%8D%95%E5%AF%B9%E8%B1%A1%E8%AE%BF%E9%97%AE%E5%8D%8F%E8%AE%AE/3841505?fr=aladdin&fromid=4684413&fromtitle=SOAP>

soap是webservice三要素之一。分别是soap、WSDL( WebServiceDescriptionLanguage)、UDDI(UniversalDescriptionDiscovery and Integration)。

其中soap用来描述传递信息的格式， WSDL 用来描述如何访问具体的接口， uddi用来管理，分发，查询webService 。

soap协议更好的集成http、smtp、mime（多用途网际邮件扩充协议）等多种协议，同时也可以与tcp、udp协议进行绑定。

如：

jingli-v1与qmtrip的rpc调用，可以实现为rpc请求映射到http请求中。Rpc应答映射到http应答中。

# 6. 负载均衡

## 6.1 简介

可用服务器的处理能力

服务器的就近原则

http的headers信息检查

## 6.2 terminology

server farms 服务器群组

## 6.3 CDN(Content-Delivery-NETWORK)

内容分发网络

# 7. 代理服务器

代理服务器，即，客户端不与服务器端直接交互，通过中间服务器实现资源的请求。

## 7.1 正向代理服务器

### 7.1.1 简介

Forward proxy: 从客户端的角度考虑(代理客户端) , 正向代理隐藏真实的客户端

## 7.2 反向代理服务器

### 7.2.1 简介

Reverse proxy： 从服务器端的角度考虑(代理服务器端)， 反向代理隐藏真实的服务器端. 一般，反向代理， 其提供真实服务的服务器有多个，需要代理服务器判断请求该如何分发到相应的服务器。即需要处理负载均衡。

7.2.2 优点

\*保护、隐藏原始资源服务器

\*减少原始资源服务器的请求量，负载均衡

\*代理服务使用缓存技术，实现web-acceleration

\*代理服务器添加一层验证，更加安全。

# 8. 加密/通信

## 简介

### A. 加密

分为信息摘要加密 ---验证信息的完整性

## 8.1 对称加密算法 (symmetric encryption algorithm)

通信双方使用相同的加密密钥（也是该加密算法的主要弊端。）

### 8.1.1 类型

#### 7.1.1.1流加密密文

#### 8.1.1.2块加密密文

64bit的块作为加密单元，是常用的一种方式。

### 8.1.2 相关算法

twofish、AES(Advanced Encryption Standard)、Blowfish、IDEA等

## 8.2 非对称加密算法(asymmetric encryption algorithm)

### 8.2.1 rsa

#### 8.2.1.1 introduction

public key and its corresponding private key are linked together through their internal mathematical structure;

trapdoor Permutation: is applying to encrypt message; trapdoor is used to decrypt the message.

Let result = Math.cbrt( m^3 % n): 计算m的三次方取n的余数的立方根

N是质数(prime integer)的积(product)

Composite integer指的是可分解的整数 6 = 2 \* 3;

笛卡尔积--- Cartesian product

### 8.2.2 Utilization

##### ssh、https、bitcoin、pgp & gpg

PGP: Pretty Good Privacy, 采用idea的散列演算法作为加密认证之用。

GPG: GNU Privacy Guard, 加密软件

总结：

symmetric encryption相较asymmetric encryption，速度要快， computational cost要小。

## 8. 3 七层协议-加密协议

### 8.3.1 运输层加密协议

#### 8.3.1.1 ssl

安全套接层(secure socket layer), 属于网络传输层。

#### 8.3.1.1 tls

运输层安全协议(transport layer secure)s

### 8.3.n 应用层

8.3.1 https

https is encrypted、secure hypertext transfer protocol。

CA—certificate authority, used to issue digital certificate, is commonly used in data transferring based on https protocol, or to issue identity cards used in electronically signing documents by governments.

Organizations which bestows CA certificates:

1. Commercial organizations
2. Non-profit organizations: CAcerts, Let’s Encrypt.

# 8.通信

## 8.1 无线通信

### 8.1.1 简介

802.11 协议，其中重要的一项子协议便是wifi标准协议。关于wifi接入的方式是通过AP接入点（Acess Point）将无线设备接入有线网络。

802.11协议，规定了通信报：源地址、目的地址、基本服务集标识（basic service set id，bssid）。

早期的保护协议： wep, 使用40位或者104位公开密钥，安全性低

WPA: wifi-protected acess, 两种模式，pre-shared key预共享模式和企业模式。

Pre-shared key：

认证过程完成4次handshake。

## 8.2 有线通信

WEP-Wired Equivelant Privacy(有线等效保密协议)

## 8.3 Duplex

#### 8.3.1 Introduction

A duplex communication system is a point-to-point system composed of two connected parties or devices that can communicate with one another in both directions.

8.3.2 Full-Duplex System---全双工

Both parties can communicate with each other simultaneously.

8.3.2.1 Time-division Duplexing

8.3.2.2 Frequency-division Duplexing

8.3.3 Half-Duplex System---半双工

Both parties can communicate with each other, but not simultaneously;

# 9. 压缩／解压原理

## 9.1 http 使用的压缩方式

### 9.1.1 lossy有损压缩

### 9.1.2 lossless 无损压缩

gzip

deflate

# 10. 编程

## 10.1 duck typing

更严格的类型检查instance of