

头条飞书一面：

未整理

1. 说说分布式 Redis
2. 用过什么分布式消息队列？（说说 kafka）

已整理

3. 说说 api 网关？（微服务）

[API 网关](#)

API 网关是一个服务器，也可以说是进入系统的唯一节点。API 网关封装内部系统的架构，并且提供 API 给各个客户端。它可能还有其他功能，如授权、监控、负载均衡、缓存、请求分片和管理、静态相应处理等。

API 网关负责请求转发、合成和协议转换。所有来自客户端的请求都要先经过 API 网关，然后路由到这些请求对应的微服务，API 网关调用多个微服务来处理一个请求以及聚合多个服务的结果。它可以在 web 协议内部使用非 web 友好型协议进行转换，如 HTTP 协议、WebSocket 协议。

请求转发、响应合并、协议装换、数据转换、安全认证

API 网关方式的核心要点是，所有的客户端和消费端都通过统一的网关接入微服务，在网关层处理所有的非业务功能。通常，网关也是提供 REST/HTTP 的访问 API。服务端通过 API-GW 注册和管理服务。

4. API 网关有什么挑战

[基于 Kubernetes 的 API 网关面临两个最重要的挑战：如何扩展边缘管理并支持多种需求](#)

挑战 1：扩展边缘管理

挑战 2：支持各种范围的边缘要求

API 网关实现的注意事项:

- a) 可能产生的单点故障或者瓶颈。
- b) 由于通过 API 网关进行了额外的网络跳转以及复杂性风险,响应时间增长了。

5. 算法题:

返回二叉排序树第 k 大的节点 (1. 不能使用全局变量; 2. 使用递归)

[链接地址证明面试官错了, 面试官让我使用 Integer 代替全局变量](#)

头条飞书二面：

未整理

1. 你做的项目有没有什么难度
2. 缓存的那个 是一些非热点的 你缓存后以后还会请求吗
3. 实时数据怎么缓存的 用户一直看的 delay 的吗
4. 拉取间隔 数据量
5. Django-crontab 定时原理 这个也是单机的吗
6. 你是起了个线程一直查吗 单机吗 有分布式需求吗 分布式怎么搞
7. 登录场景怎么做

已整理

8. 用户怎么把账号密码发送到后端 为什么用 post

[Web 登录其实没有那么简单](#)

（推荐看看）[99%的人都理解错了 HTTP 中 GET 与 POST 的区别](#)

（强烈推荐看看）[GET 和 POST 到底有什么区别](#)

把账号密码发到后端

MD5 加密（对密码加密，防止密码被截获）+token 令牌（完成校验时，服务器将删除对应的令牌缓存，防止重放攻击）+数字签名（防止篡改）。

注意：用户在注册输入密码的时候，就存储 MD5(password)值，并且在 WEB 端先进行 MD5(password)，然后将密码传输至后台，与数据库中的密文进行比较。

GET 与 POST 本质上都是基于 TCP/IP 的，GET 也可以带 request body 但是不保证对方能收到数据（取决于接受服务器的策略）。

GET 产生一个 TCP 数据包，POST 产生两个 TCP 数据包。

对于 GET 请求，浏览器会把 http header 和 data 一起发出去，服务器响应 200；

而对于 POST，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。

并不是所有的浏览器都会在 POST 发送两次数据包，Firefox 就只发送一次。

9. spring springboot 请求流程

[理解 Servlet 和 Servlet 容器、Web 服务器等概念](#)

[图解 Spring: HTTP 请求的处理流程与机制](#)

[springboot 请求处理流程](#)

[HandlerAdapter 是什么，有哪些作用](#)

从 request 开始，依次进入 DispatcherServlet (前置分发器)->HandlerMapping (处理器映射)->Controller (控制器)->ModelAndView (模型和视图)->ViewResolver (视图解析器)->View (视图)->response (响应)

DispatcherServlet、HandlerMapping、ViewResolver 只需要在 xml 文件中配置即可。

ModelAndView 包含了模型 Model 对象和 View 视图对象名称。

一般 Spring 中会有多个 Controller handler 方法，所以一个请求过来肯定是要先找到匹配该请求的方法，其次请求过来的参数需要处理成 Controller 方法可以识别的参数格式，这样才能做下一步处理，最后 Controller 方法处理完后，可能还需要把返回值做一定的处理再返回给前端。

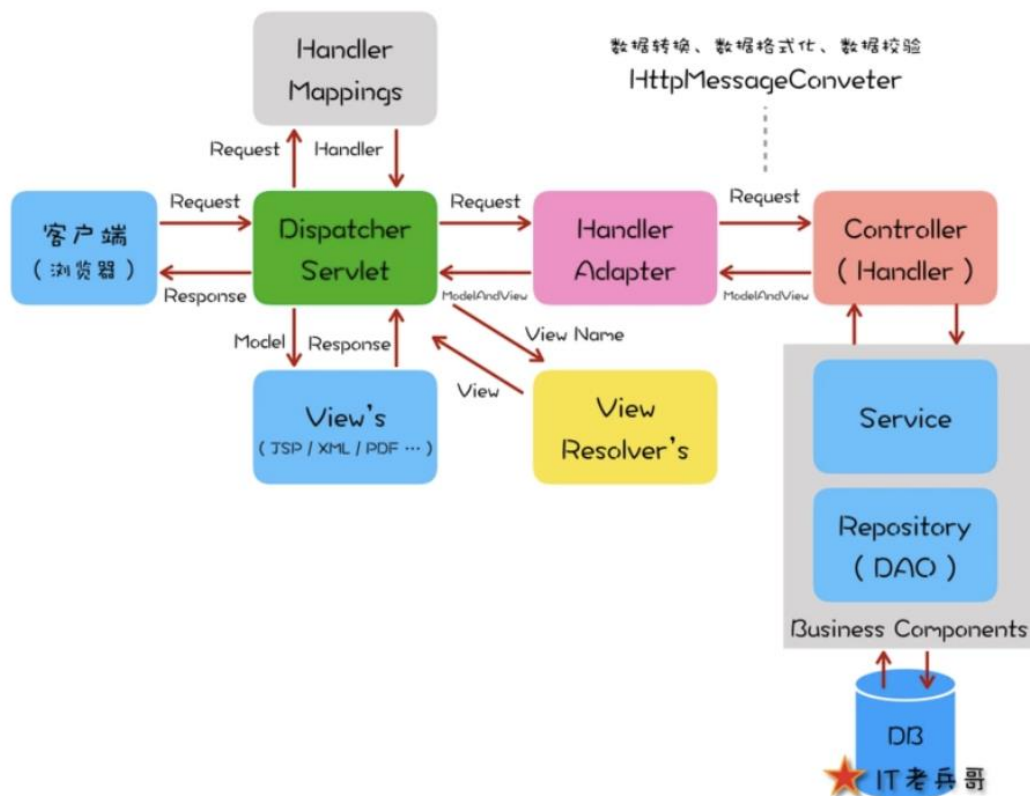
InvocableHandlerMethod 用于在某个请求被控制器方法处理时，包装处理所需的各种参数和执行处理逻辑。

ServletInvocableHandlerMethod 用于对返回值进行处理。

步骤	接口	实现方式
路径映射	HandlerMapping	在 DispatcherServlet 中，SpringBoot 注册了一个 HandlerMapping 列表。请求过来时，会循环该列表，来解析 url 获取 handle 方法，获取到之后跳出循环。
参数解析	HandlerMethodArgumentResolver	在 InvocableHandlerMethod 中，Spring 注入了一个 composite 对象，该对象中有多个解析器放在列表中，解析参数同样是循

		环列表,来找出第一个可以解析的解析器进行解析。
返回处理	HandlerMethodReturnValueHandler	在 ServletInvocableHandlerMethod 中, Spring 注入了一个 composite 对象,该对象中有多个返回值处理器在列表中,处理器返回值时,首先同样是循环列表,找到处理器,再进行返回值处理。

4.2 Spring 框架处理 HTTP 请求的流程



- 1) 前置分发器 DispatcherServlet 接收到 HTTP 请求后,将查找适当的控制器 Controller 来处理请求,它通过解析 HTTP 请求的 URL 获得 URI,然后再根据 URI 从处理器映射 HandlerMapping 当中获得该请求对应的处理器 Handler 和处理器拦截器 HandlerInterceptor,最后以 HandlerExecutionChain 形式返回。
- 2) 前置分发器 DispatcherServlet 根据获得的处理器 Handler 选择合适的适配器 HandlerAdapter。如果成功获得适配器 HandlerAdapter,在调用处理器 Handler 之前其拦截器的方法 preHandler()优先执行。
- 3) 方法 perHandler()提取 HTTP 请求中的数据填充到 Handler 的入参当中,然后开始调用处理器 Handler (即控制器 Controller) 相关方法。

- 4) 控制器 Controller 执行完之后, 向前置分发器 DispatcherServlet 返回一个模型与视图名对象 ModelAndView。
- 5) 前置分发器 DispatcherServlet 根据模型与视图名对象 ModelAndView 选择合适的视图解析器 ViewResolver, 前提是该视图解析器必须已经注册至 SpringIOC 容器当中。
- 6) 视图解析器 ViewResolver 将根据 ModelAndView 里面指定的视图名称获得特定的视图 View。
- 7) 前置分发器 DispatcherServlet 将模型数据填充至视图当中, 然后将渲染结果返回给客户端。

在填充处理器 Handler 入参的过程中, Spring 还会根据配置做如下处理:

- 1) **HttpMessageConveter:** 将请求消息 (Json、XML 等) 转换成 Java 对象。
- 2) **数据转换:** 对 HTTP 请求中的数据做类型转换, 例如: 将 String 转换成 Integer、Double 等。
- 3) **数据格式化:** 对 HTTP 请求中的特定数据做格式化, 例如将字符串转换成格式化数字或者格式化日期等
- 4) **数据验证:** 验证数据的有效性 (长度、格式等), 验证结果存储到 BindingResult 或者 Error 当中。

10. 怎么进行 db 的比对

11. 怎么避免 sql 注入

12. 用户登录态 session Cookie 的区别 怎么判断用户是否登录了

[前端 er 应该懂的登录态: Cookie、Session 和 Token](#)

[看完这篇 Session、Cookie、Token, 和面试官扯皮就没问题了](#)

[彻底弄懂 session, cookie, token](#)

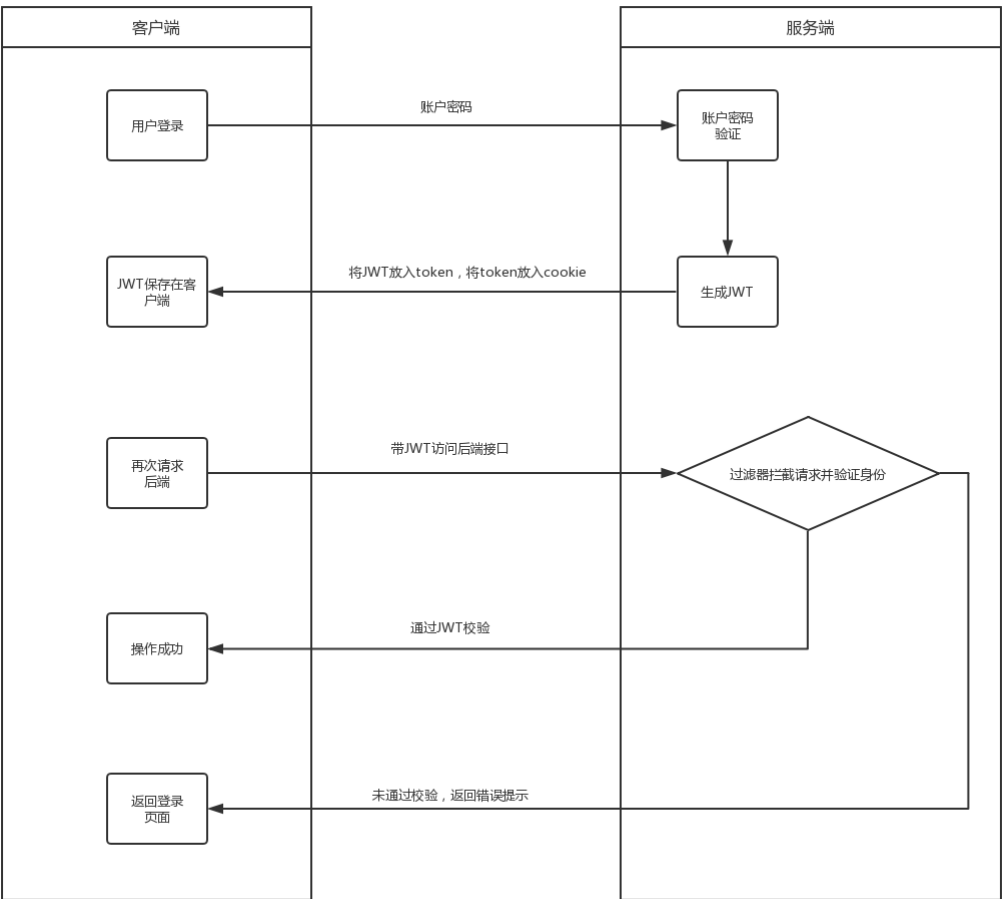
[JWT 超详细分析](#)

[详说 Cookie, LocalStorage 与 SessionStorage](#)

使用 JWT 维护登录态, 服务器不用再维护一个状态表, 服务器给仅客户端发送一个加密的 token, 每次请求带上这个加密的的数据, 服务器再解密验证是否合法即可。由于是加

密的，用户是无法修改的。

JWT 流程：



<https://blog.csdn.net/Sunhighlight>

Session Cookies 和 JWT

相同：

它们都既可以对用户进行身份验证，也可以用来在用户单击进入不同页面时以及登录网站或应用程序后进行身份验证。

不同点：

- JWT 具有加密签名，而 Session Cookies 则没有；
- JWT 是无状态的，因为 JWT 是保存在客户端，而 session 保存在服务端；
- 由于 JWT 是无状态的，在许多情况下，它们可以节省服务器资源。因此 JWT 要比 Session Cookies 具有更强的可扩展性；
- JWT 支持跨域认证；
- Session Cookies 只能用在单个节点的域或者它的子域中有效。如果它们尝试通

过第三个节点访问，就会被禁止。如果你希望自己的网站和其他站点建立安全连接时，这是一个问题。

使用 JWT 可以解决这个问题，使用 JWT 能够通过多个节点进行用户认证，也就是我们常说的跨域认证。

13. 怎么加解密 怎么用到公钥私钥

14. 什么时候对称加密

15. 算法题

[K 个一组翻转链表](#)

(hard, 面试出这题，有点过分，最过分的是，用这题在面评中说我 coding 有问题)

腾讯云一面：

未整理

1. 自我介绍
2. 介绍一下字节的经历 你们的数据是什么类型的 是什么东西 为什么慢 查询还是网络
3. 怎么优化的，
4. 什么数据 晚 1 分钟有问题吗 本地 `cache` 到什么地方 本地的有没有什么性能问题
5. 为什么不选 `memcache` 有没有考虑其他的 `db` (`redis`) 你这是在重复造轮子吗
6. 为什么用 `redis` 开发会慢呢
7. 你写的定时器是个什么东西 你这个 `python` 是个循环吗 你做的东西有人指导吗
8. `crontab` 听说吗 `linux` 下专门做定时器的东西 技术选型的时候导师有没有指导你
9. 有什么方案呢？
10. 说说你的阿里云的实习
11. 详细说一下推送的 (`supervisor` 进程管理) 你怎么解决的 解决的效果
12. 你的 `ip` 数据包的项目

13. ip 数据包 tcp 包的结构
14. 根据 ip+port 来确定一条数据流
15. 然后根据他的 seq 来进行排序
16. 三次握手
17. 为什么 3 次 4 次
18. tcp udp
19. http1.0、1.1、2
20. mysql
21. sql 慢查询
22. 算法题：

数组 int 很大 乱序 有重复 去重

扔 set bitmap

头条飞书三面：

未整理

1. MVC 之路径匹配规则

已整理

2. Controller 的作用

[Django MTV 和 MVC 的区别](#)

MTV 与 MVC：

MVC：

Model 代表数据存取层；

View 代表的是系统选择显示什么和怎么显示；

Controller 指的是系统中根据用户输入并视需要访问模型，决定使用哪个视图。

Django 中 M、T 和 V 各自的含义：

M：Model 数据存储层，处理所有数据相关的业务，和数据库交互，负责业务对象和数据库的关系映射(ORM)。

T：Template 模板层（也叫表现层）具体来处理页面的显示

V：View 业务逻辑层，处理具体的业务逻辑，它的作用是连通 Model 层和 Template。

3. 知道哪些 django 中间件

（CNM，要问什么中间件直接问，还让我选，我 TM 哪知道有什么中间件）

4. 算法题：

写一个工厂模式

腾讯二面：

未整理

1. 自己说简历的项目；

已整理

2. 说说 TCP，UDP，HTTP 的区别

TCP、UDP、HTTP 与 HTTPS 的异同

TCP 与 UDP 的区别：

TCP 是需要连接的传输协议（长连接），UDP 是不需要连接的；

TCP 对系统资源要求比较多，UDP 要求比较少；

TCP 的结构比较复杂，UDP 比较简单；

TCP 是流模式，UDP 是数据报模式；

TCP 保证数据正确性，UDP 可能丢包，TCP 保证数据顺序，UDP 不保证；

UDP 信息包的标题很短，只有 8 个字节，相对于 TCP 的 20 个字节信息包的额外开销很小。

Ping 是向对方主机发送 UDP 数据包。

HTTP 是建立在 TCP 协议之上的

特点：

客户/服务器模式；

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。请求方法常用的有 GET、POST、HEAD、PUT、DELETE、CONNECT、OPTIONS、TRACE；

灵活：HTTP 允许传输任意类型的数据对象；

无状态；

HTTPS

TLS 可以看做是 SSL3.0 版本的升级，但是对于大的协议栈来说，SSL 和 TLS 并没有

太大的区别。

TCP/IP 是一个协议组，分为网络层、传输层、应用层。

在网络层有 IP 协议、ICMP 协议、ARP 协议、RARP 协议和 BOOTP 协议。

在传输层有 TCP 协议和 UDP 协议。

在应用层有 FTP、HTTP、TELNET、SMTP、DNS 等协议。

3. 长连接

[TCP 短连接和长连接的区别](#)

<https://www.jianshu.com/p/f4005c1c728f>

中国电子系统一面：

未整理

1. 自我介绍
2. 实习对自己的主要提高有哪些

已整理

3. 为什么用 django

（我 TM 一实习生，哪知道为什么用 Django，技术选型轮得到我吗？）

4. http tcp
5. linux 常用命令和日志相关的
6. 跳表 红黑树

[什么是跳表](#)

7. linux top 系统负载
8. java 网络编程 nio bio

顺丰科技一面：

未整理

1. 项目哪些是用 java 做的
2. 有做过 java web 项目
3. spring mvc 后台逻辑（请求的处理流程）

已整理

4. 过滤器和拦截器的区别

[过滤器\(Filter\)和拦截器\(Interceptor\)的区别](#)

过滤器包裹住 servlet，servlet 包裹住拦截器。

5. 说说 Spring AOP

[面试被问了几百遍的 IoC 和 AOP ，还在傻傻搞不清楚？](#)

6. MySQL 两种引擎的区别（InnoDB 与 MyISAM）

[MySQL 索引背后的数据结构及算法原理](#)

InnoDB 是 SQL 默认的事务性存储引擎，采用 MVCC 支持高并发，默认的事物隔离级别是 RR（可重复读）；

InnoDB 能够在内存中创建哈希索引以加速读操作，还有自适应哈希索引等。

InnoDB 与 MyISAM 区别：

事务支持：InnoDB 支持事务，MyISAM 不支持事务；

锁机制不同：MyISAM 是表级锁，InnoDB 是行级锁；

InnoDB 是聚集索引，MyISAM 是非聚集索引；

InnoDB 支持外键，而 MyISAM 不支持；

InnoDB 不保存表的具体行数，而 MyISAM 保存，也就是说，InnoDB 需要扫描一

遍整个表才能计算表有多少行，而 MyISAM 不用；

如何选择：

是否要支持事务，如果要请选择 InnoDB，如果不需要可以考虑 MyISAM；

如果表中绝大多数都只是读查询，可以考虑 MyISAM，如果既有读写也挺频繁，

请使用 InnoDB；

系统崩溃后，MyISAM 恢复起来更困难，能否接受，不能接受就选 InnoDB；

7. Mysql 的 sql 优化方法

[最全的 SQL 优化策略送给你](#)

减少返回的列；

减少返回的行；

8. 项目里最有成就感的

9. 自己最大的优势