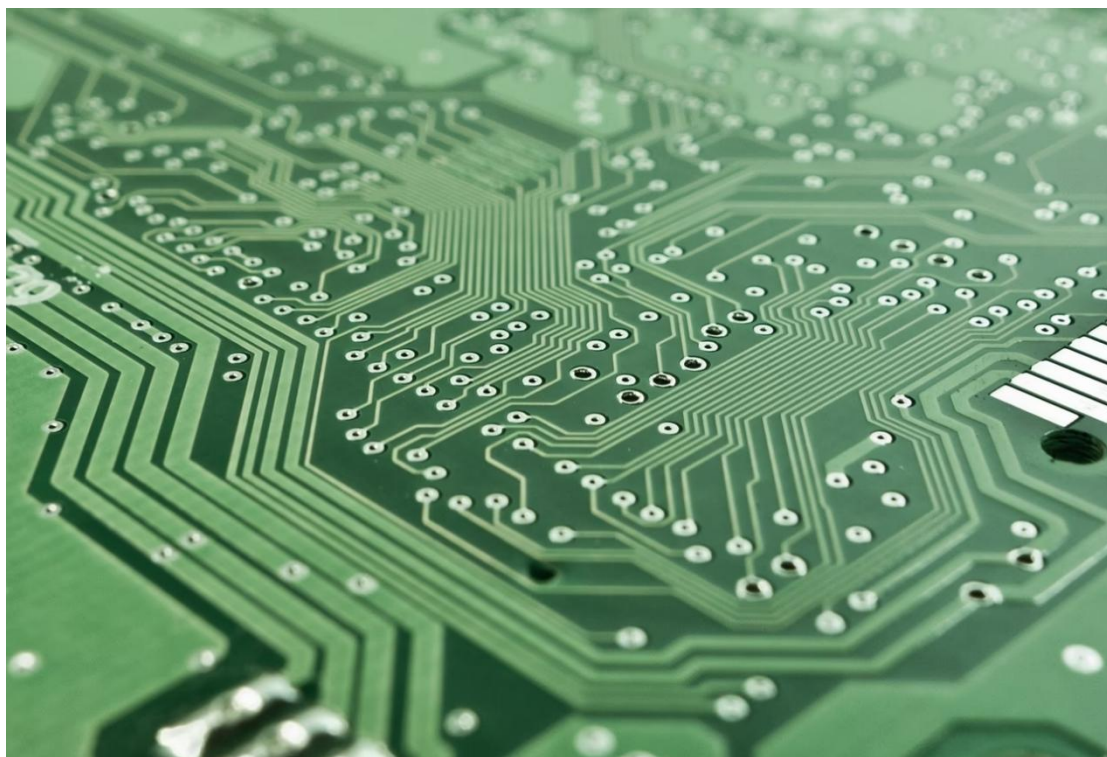


产品经理必备技术基础知识

作为产品经理，我们试图用产品的语言来说明技术问题，但产品的理解也并不都一定正确，因为技术是一个太宽太广的领域，一个人终其一生也不过在某一技术领域有所精湛，做软件的不精通硬件技术，做硬件上层技术的不精通底层技术等，但我们在工作中确实会需要到一些相关的知识，才能更好的和开发进行沟通，当然作为产品经理，我们也不需要具体理解技术的实现，仅需要知道有这项技术即可，知道其基本原理更好。



一. 产品为什么要懂技术

产品经理在思考产品时会优从用户使用产品的场景出发，从场景出发设计产品功能，包含了用户常规的使用场景及异常使用场景，而技术人员会优先从产品功能的实现步骤角度出发，实现某一功能需要

的步骤分别是什么，每一步又可以拆分成子步骤，子代码等，粒度更细，逻辑性更严谨。所以路径推理的技术思维和用户场景的产品思维，必然会各自站在各自的角度去分析和讨论问题，难免会产生差异，产品经理讲的是 A，技术理解的是 B，就会使产品的最终开发出现问题。所以产品经理如果能从技术的角度多考虑一些，会让产品和技术的沟通更加顺畅一些，懂技术不是产品经理的必备项，但是一定是加分项。

二. 产品的技术架构

逻辑层：把产品需求翻译成逻辑，即流程图、状态图、逻辑图、时序图。理清楚业务流程与功能模块；

实现层：具体的函数方法，即代码。

接口层：各功能、页面、模块之间交互的通道。产品往往会涉及到多个业务模块的交互，就是你传给我什么数据，我给你什么数据以及状态的确认。

数据层：程序执行的结果，即通过写入或读取对数据库的交互操作，需要了解一些数据库知识。

架构层：技术的技术架构、调用关系、技术规范等。我们需要了解技术的组织结构即可，可适当了解公司的技术架构。越是高级人员，架构能力越强，越需要。

所以对于产品，逻辑层是产品人员和技术人员建立连接的第一层，需要我们一定要将产品逻辑梳理清晰，接口层、数据层、架构层是能让彼此沟通更加顺畅，实现层了解即可。

三. 技术基础知识

计算机的构成

1. 最底层可以说是硬件，通过硬件来支持上层操作系统的运行，计算机系统的硬件结构主要由四部分组成：控制器、运算器、内存和输入输出设备，其中，控制器和运算器统称为中央处理器，简称 CPU，它是计算机硬件系统的指挥中心，包括控制器和运算器两个部件，其中，控制器的功能是控制计算机各部分协调工作，运算器则是负责计算机的算术运算和逻辑运算。

2. 中间层是操作系统，是管理计算机硬件与软件资源的计算机程序，同时也是计算机系统的内核与基石。操作系统需要处理如管理与配置内存、决定系统资源供需的优先次序、控制输入设备与输出设备、操作网络与管理文件系统等基本事务。同时操作系统也提供一个让用户与系统交互的操作界面。操作系统的类型非常多样，不同机器安装的操作系统的类型非常多样，不同机器安装的操作系统可从简单到复杂，可从移动电话的嵌入式系统到超级计算机的大型操作系统，常见的有 Windows、MAC、IOS、Android。

3. 上层是浏览器、桌面程序、APP 程序、应用程序，然后基于网络服务实现数据的交互，而互联网产品也是在这一层发挥的作用，产品经理负责这一层产品的设计，技术人员负责产品程序的开发。

程序

程序（program）是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。程序的执行是为了实现预期目的而进行操作的一系列语句和指令，一般分为系统程序和应用程序两大类。

技术架构

那用户通过程序访问数据，数据是如何实现传输的呢？这时我们就要了解下 B/S 结构和 C/S 结构了。目前我们常见的是 B/S 结构，即浏览器和服务器的架构模式，它是随着 Internet 技术的兴起，对 C/S 结构的一种变化或者改进的结构。C/S 结构体系是一种软件系统体系结构，通过将任务合理分配到 Client 端和 Server 端，降低系统的通讯开销，可以充分利用两端硬件环境的优势。早期的软件系统多以此作为首选设计标准。B/S 架构是 WEB 兴起后的一种网络架构模式，WEB 浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。用户通过电脑、手机、平板电脑访问互联网，互联网通过服务器向数据库获取数据并将计算结果再通过网络吐给电脑、手机、平板电脑的浏览器，浏览器再展示给用户查看。

前端与后端

前端：凡是运行在用户设备上的技术都可以称为前端技术。如浏览器网页、PC 客户端，Android/iOS 应用程序 APP 等。

后端：是负责将页面需要的内容通过数据包的形式通过网络传送到前端，同时后端还可以对前端提供的数据进行计算和保存。通常包含 HTTP 服务器、应用服务器、数据库。

前台和后台的区分：都是指前端，只不过针对的用户不一样，前台主要针对的是企业外部用户，后台主要针对的是企业内部用户，前台和后台是一个相对概念，如电商的前台和商家的后台和企业后台。

前台主要做的就是内容展示、页面布局、页面交互、提交信息等，后台主要负责管理，包括任务流页面（条件搜索、列表页、详情页（审核）、增删改查导）、权限管理等；前端包含前台、后台，都是给用户进行使用的地方，后端是技术的地盘，一般我们都看不到，包含服务端、数据库、运维等内容。

常见编程语言

编程语言是程序设计人员与计算机进行交互的指令集，在计算机中任何逻辑和表达都可以通过编程语言来实现。技术人员就是通过编程语言实现前端/后端不同程序的执行的，那前端语言有哪些？前端语言主流的包含：HTML、CSS、Javascript、jQuery、Bootstrap、Ajax，下面我们认识下：

Html：全称 HyperText Markup Language，是一组标签和文本的组合，是搭建网页的基础语言。代码简单，功能强大，什么平台都能用，什么电脑都能用，它已经包含了网页常见的所有元素，后面我们还会讲到。

CSS：CSS 又称叠层样式表，能够对网页中元素位置的排版进行像素级精确控制，支持几乎所有的字体字号样式，拥有对网页对象和模型样式编辑的能力。通过 CSS 能让页面排版更精美，是目前基于文本展示最优秀的表现设计语言。

JavaScript：通过嵌入到 Html 中，使网页拥有更加炫酷的交互，增加网页的动态功能，可以在多平台下运行，如 Windows、Linux、Mac、Android、iOS 等。Javascript 就是用来给 HTML 网页增加动

态功能，实现更炫酷的交互。现在我们看到的 PC 端或 APP 内部需要加载的页面信息，几乎都是通过以上三种语言开发完成的。

jQuery: 是一个 Javascript 库，主要面向查询的。简单理解就是 javascript 里面那些需要用一行行代码实现的体力活，在 jQuery 里面可以直接打包成模块，调取对应的接口使用，解放了开发者更多的时间。这种模块化的使用方式让开发者可以很快就开发出酷炫的页面。jQuery 使用户能更方便地处理 HTML，它能够使用户的 HTML 页面保持代码和 HTML 内容分离，通过 jQuery，可以不用在 HTML 里面插入一堆 JS 来调用命令，只需要定义 ID 即可。

Bootstrap: 是针对 Javascript 进行封装的库，它在 jQuery 的基础上进行更加人性化的完善，使用起来更方便。它有很多现成的组件，比如导航栏、下拉菜单、按钮，都定义好了样式和交互，直接成套拿来用就行。

Ajax: 是指一种创建交互式网页应用的网页开发技术，一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。通过在后台与服务器进行少量数据交换，Ajax 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。传统的网页在不使用 Ajax 时，需要更新内容，必须重载整个网页页面。



响应式布局

响应式布局是一种网页设计和开发技术，旨在使网站能够适应不同设备上的不同屏幕尺寸和分辨率。它的目标是提供一种无论用户使用何种设备访问网站都能获得最佳体验的方法。

传统的网页设计通常是基于固定的布局，这意味着设计师会针对特定的屏幕尺寸设计网站，并使用固定的像素值设置元素的大小和位置。然而，这种方法在移动设备的普及和各种屏幕尺寸的出现之后变得不再适用。

相比之下，响应式布局采用了一种灵活的方法来适应不同的屏幕尺寸。它使用相对单位（如百分比、em、rem）而不是固定像素值来定义元素的大小和位置。同时，通过媒体查询（media queries）和CSS3的弹性盒子布局（flexbox），响应式布局可以根据屏幕尺寸动态地调整元素的样式、排列方式和布局结构。

响应式布局的优点包括：

用户体验改善：无论用户使用台式机、笔记本电脑、平板电脑还是智能手机访问网站，都能获得更好的浏览体验。

节省开发成本：通过一个响应式设计，可以避免为不同设备开发多个独立的网站或应用程序，从而减少开发和维护的工作量。

简化内容管理：由于只需要维护一个网站版本，所以更新和管理内容变得更加简单和高效。

要实现响应式布局，开发者需要注意以下几点：

使用流式布局：将元素的宽度和位置设置为百分比或弹性单位，使其能够根据屏幕尺寸自适应调整。

应用媒体查询：使用 CSS 的媒体查询功能，通过检测屏幕尺寸、方向和像素密度等属性来应用不同的样式。

优化图片和媒体：使用自适应图片技术（如 `srcset` 和 `picture` 元素）和媒体资源查询（`media queries`）来加载适合不同屏幕尺寸的图像和媒体文件。

设计友好的用户界面：考虑不同设备上的交互方式和触控操作，使用户能够轻松地浏览和使用网站。响应式布局是一种灵活和可适应的网页设计方法，可以提供一致的用户体验，无论用户使用什么设备来访问网站。它是现代网页设计中的重要技术，以适应不断变化的设备和屏幕尺寸。

HTTP 服务器

作为传输数据通道，简单可以认为是网通、电信、移动提供的数据服务，有了这个服务我们才能实现数据的对接传输。HTTP 服务器基础协议就是 TCP/IP，是 Internet 最基本的通信协议，通过这个来实现点对点的通信。我们访问的网站/App 就是通过通信协议和数据交互格式的约定，进行数据的传输流转的，通信协议有 HTTP/HTTPS、WAP、websocket 等，数据交互格式主要有三种：PB、JSON、XML。

XML：提供统一的方法来描述和交换独立于应用程序或供应商的结构化数据。XML 的简单使其易于在任何应用程序中读写数据；

PB：Protocol Buffer，是 Google 定义的一个数据传输协议，以简洁，省流，易用的二进制数据进行传输。作为一种效率和兼容性都很优秀的二进制数据传输格式，可以用于诸如网络传输、配置文件、数据存储等诸多领域。

JSON，JavaScript Object Notation，JS 对象标记是一种轻量级的数据交换格式，易于人阅读和编写，同时也易于机器解析和生成。

通俗点说 Web 端大部分数据交互是通过 http 的数据包进行数据交互，移动端大部分是通过 JSON 的数据包进行数据交互。

应用服务器

对数据进行逻辑运算的部分，也就是技术负责的主要部分，比如阿里云、腾讯云或自建的机房都属于应用服务器，我们的服务端的技术就是将写好的代码上传到应用服务器上被使用的。在互联网公司服务端的开发一般能占到开发团队一半的人数。而产品经理在设计产品时，更多的是客户端产品，但客户端之间的信息互动和数据传输需

要通过服务端完成，服务端起到了中间核心处理的作用，负责处理复杂的业务逻辑并对数据进行管理。所以我们更应该了解服务端的技术都在用什么编程语言写代码。目前主流的服务端编程语言如下：

那有没有不通过后端服务实现数据的访问的？当然有，那就是无后端，无后端（noBackend）致力于让构建一个应用的过程变得更简单，实际上是通过前端代码抽象成后端接口，一些移动应用初期可以没有后端，实现项目的快速启动，如果后期业务逻辑复杂之后再寻找合适的后端工程师迁移也不迟。无后端的工具有 Facebook Parse, Google Firebase 以及国内的 LeanCloud。

数据库(DB)

运行在服务器上，按照数据结构对数据进行存储、组织和管理，通过程序可对数据库进行增删改查等操作。在公司一般叫做 DBA（DataBase Administrator，简称 DBA），它就是后厨仓库，里面存放着各种原材料（白菜、萝卜、西红柿等），并按某种关系分门别类的进行摆放管理，映射到数据库就是将不同类型的数据存储于不同的数据库，并通过不同的映射关系进行信息的管理。

数据库的形式分为关系型和非关系型。

关系型数据库：是建立在关系模型基础上的数据库，简单说，关系型数据库是由多张能互相关联连接的二维行列表格组成的数据库。类似于多张有关系的 excel 表格。我们最常见的关系型数据库是 MySQL，MySQL 是一个关系型数据库管理系统，可以通过 SQL 语句（即结构化查询语言，是一种操作关系数据库的编程语言）进行数据获取。

当前主流的关系型数据库还有 Oracle、DB2、PostgreSQL、Microsoft SQL Server、Microsoft Access、MySQL、浪潮 K-DB 等。

优点是体积小、速度快、成本低，一般中小型网站的开发都选择 MySQL 作为网站数据库；

缺点是在海量访问并发和海量数据管理时的力不从心，而且对数据库的升级和扩展很麻烦，往往需要停机维护和数据迁移。

非关系型数据库：即 NoSQL，泛指非关系型的数据库，是一种相对松散且不可按照严格的结构规范进行存储的数据库。它是随着互联网 web2.0 网站的兴起而产生的，传统的关系数据库在应付 web2.0 网站，特别是超大规模和高并发的 SNS 类型的 web2.0 网站时已经显得力不从心，暴露了很多难以克服的问题，而非关系型的数据库则由于其本身的特点得到了非常迅速的发展。非关系型数据库以人为单位，数据可增加个人元素，这个我们可以比作每个人的档案袋，每个人的基础数据是一样的，又可以在基础数据的基础上增加个性的档案数据信息。NoSQL 数据库的产生就是为了解决大规模数据集合多重数据种类带来的挑战，尤其是大数据应用难题。

优点：性能高，可扩展性强；

缺点：维护成本高、不提供对 SQL 的支持、不提供关系型数据库对事物的处理；

非关系型数据库有：MongoDB、CouchDB 等；

当然数据库还有介于关系型和非关系型之间的，当然这里的内容太深奥，就不再延展，我们知道有这样的数据库即可。

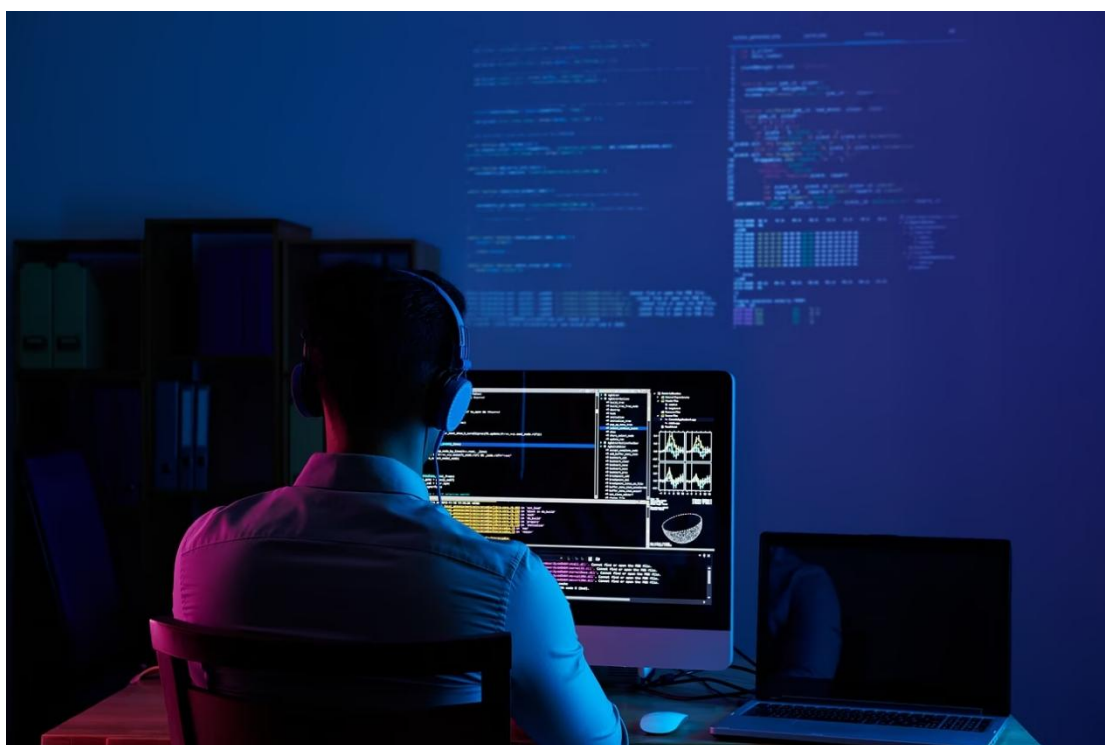
Redis

Redis 也是我们经常从技术口中听到的一个词，Redis 是一个开源的、使用 C 语言编写的、支持网络交互的、可基于内存也可持久化的 Key-Value 数据库，可用作数据库，高速缓存和消息队列代理，并提供多种语言的 API。

Redis 是一种开源的内存数据结构存储系统，它具有高性能和灵活性，被广泛用作缓存、消息队列和数据库等多种用途。“Redis”是“Remote Dictionary Server”的缩写。

Redis 支持多种数据结构，包括字符串、哈希表、列表、集合和有序集合。它提供了丰富的命令集，允许开发人员使用这些数据结构进行各种操作，如读取、写入、更新和删除数据。由于 Redis 将数据存储在内存中，它能够提供非常快速的响应时间。

除了基本的键值对存储之外，Redis 还具有一些特殊功能。例如，它可以通过发布-订阅模式实现消息传递，允许不同的进程或系统之间进行实时通信。此外，Redis 还提供了事务支持，可以保证一系列操作的原子性。另一个 Redis 的重要特点是其持久性选项。虽然 Redis 主要将数据保存在内存中以提供高性能，但它可以配置为定期将数据异步保存到磁盘上的持久化文件。这样可以在服务器重启后恢复数据，并且可以防止数据丢失。Redis 是功能强大的数据存储系统，通过使用内存来提供快速访问和操作数据，使其成为许多应用程序的理想选择。



API 接口

API 泛指实体把自己提供给外界的一种抽象化物，用以由内部操作分离出外部沟通的方法，使其能被内部修改而不影响外界其他实体与其交互的方式，是数据传输的一个通道。好比楼房里的电线，看不见，但需要有电线才能通电。而产品的功能和功能之间需要数据交互，建立连接的通道就叫 API 接口，而 MOA 目前是让开发可以快速开发 API 的类后端；接口调用方式包含同步调用、回调和异步调用，同步和异步的区别就在于在下达了执行任务的命令后，是等到执行完成之后才能得到结果呢，还是马上就知道了结果；大部分情况下技术之间的沟通障碍都是因为接口问题导致的，所以产品要把握住核心接口，接口中都包含什么数据字段，也就是定义好接口传输的信息。

JDK/Eclipse/Java

JDK 即 Java Development Kit, JDK 是 Java 语言的软件开发工具包, 主要用于移动设备、嵌入式设备上的 java 应用程序。JDK 是一切 java 应用程序的基础, 所有的 java 应用程序是构建在这个开发工具之上的。

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台, 安卓程序的一种集成开发环境。

Java 是安卓程序的主要开发语言, 安卓应用程序后缀名为 Apk, 发布时需要一个特殊的签名文件为这个安装包文件签名, 签名的目的是确保这个安装包的唯一性和安全性。顺便说一下 Windows 的应用程序的后缀名为 exe。关于系统权限控制, 安卓授权发生在应用安装阶段, 在安装时提示用户同意相关权限协议才能安装, IOS 则是在用户使用产品的过程中进行授权的。从用户体验和安全上来说, IOS 的权限管理会更好, 当具体用到某一个需要授权的功能时, 再提示授权通知, 能引起用户的关注。

H5

H5 即 HTML5, HTML 是万维网 (WWW) 的核心语言, H5 是 HTML 的第五次重大修改。可以只编写一次就跨平台运行, 更快做跨屏适配, 效果很酷炫, 但也有一些功能的局限, 由于 HTML5 必须要经过浏览器这个中间层, 所以在性能上多少会有些损失, 所以如果你的应用对性能特别敏感, 原生就会是比较好的选择。

混合式

混合式 (Hybird APP) 就是将 H5 页面嵌入到 APP 的原生器中，目前大部分的 APP 都是混合式的。通常开发周期为 3-4 周一个版本迭代。

对于一些用户使用频率高、体验要求高的产品模块，使用 Native 实现会比较合适，对于使用频率低、内容变化频繁的模块使用 H5 实现，如果需要修改 Native 的模块就需要发版更新解决了。

cookie/session/token

在公司团队里一般是用户服务组会提供注册和登录服务，提供用户的注册、登录授权标识，而这几个服务就与 cookie/session/token 三个技术密切相关：

Cookie：是一种数据，由服务器生成，发送到你的浏览器，然后由浏览器保存到本地的某个文件夹里，等你下次再登陆这个网站时，浏览器就会把你的 Cookie 发送给服务器，这时服务器就知道了“原来又是你在访问了”。一条 Cookie 的生命一般是一个轮回，就是当你第二次登录网站时，第一次登陆的那条 Cookie 会被覆盖，同时也可以设定周期，比如“一个月内自动登陆”等。

Session：则是一种时间的概念，就是你打开一个网站到关闭这个网站之间的时间，在这段时间里，你在网站上的动作都会被上传，然后以 Session 的方式保存在服务器中，比如你点击加购物车的动作，就会被 Session 保存在服务器上，关闭了网站怎么办？你的 Session 会被编上号，以 SessionID 的形式发送到浏览器，以 Cookie 的形式保存在本地，这就是 Cookie 和 Session 的亲密合作。

Token: 在计算机身份认证中的令牌，在一些数据传输之前，要先进行核对，不同的令牌被授权不同的数据操作，它就像一把钥匙只能开一把锁。产品会如何使用到 Token 呢，比如强制用户下线操作，重新登录操作都会用到，使用 token 进行的身份认证，安全性好且是唯一的。使用 Token 可以实现单点登录、单设备登录。单点登录就是单一标记登录，一个账号只需要登录一次就可以访问不同页面或产品。例如登录 QQ，可以访问 QQ 空间、QQ 邮箱等其他服务。单设备登录：就是只能在一个设备上登录，若同时在其他设备登录，先前登录的用户会被提醒或踢下线，例如微信只能在一台手机设备上登录，登录其他手机设备时会被提醒并踢下线。

其他常见技术术语

渲染: 是通过一些列计算并呈现的过程，其中包括测量、排版、绘制。我们看到浏览器加载出来的以各种不同排版的方式的数据就是通过渲染得来的。

写死: 将数据集写死在本地，省去数据接口的开发，可以快速实现效果，但不利于更新；

UA 信息: User Agen 的简称，中文名为用户代理，它是一个特殊字符串头，使得服务器能够识别客户使用的操作系统及版本、CPU 类型、浏览器及版本、浏览器渲染引擎、浏览器语言、浏览器插件等，从而判断用户是使用电脑浏览还是手机浏览，让网页作出自动的适应。

URL: 统一资源定位符(URL, 英语 Uniform / Universal Resource Locator 的缩写) 也被称为网页地址，是因特网上标准的资源的地址

(Address)。每个网页都对应着一个 URL 地址，类似于一个真实世界中的门牌地址一样。网址标识了一个 Web 页面所在的互联网里面的真实地址(这个页面处于 `www.baidu.com/file/1.html`, 处于 baidu 服务器 file 路径下的 1 这个文件)。

缓存：缓存就是数据交换的缓冲区（称作 Cache），当某一硬件要读取数据时，会首先从缓存中查找需要的数据，如果找到了则直接执行，找不到的话则从内存中找，由于缓存的运行速度比内存快得多，故缓存的作用就是帮助硬件更快地运行。

HTTP 请求：是指从客户端到服务器端的请求消息，包含了请求时长、QPS 等；

队列：在结构上分为队尾和队头，只能在队头执行出队操作，在队尾执行入队操作。技术人员经常会提及此词，以一个队列的形式给你吧，就是数据的一种交互方式，了解即可。

索引：即数据表的目录，比如要加入某个搜索条件，技术会告诉你没加索引，即没有目录，只能到表里面查找。

脚本：可以被计算执行的文件，文件里是一系列计算机指令，这些指令会按顺序被计算机解析并执行，如批量冻结 10W 个账号；

明文：明文，是指没有加密的文字（或者字符串），一般人都能看懂的意思，属于密码学术语。

密文：是加了密的的文字，明文是加密之前的文字。密文是对明文进行加密后的报文

加密：加密是以某种特殊的算法改变原有的信息数据，使得未授权的用户即使获得了已加密的信息，但因不知解密的方法，仍然无法了解信息的内容，对数据安全尤为重要。

Bug：因为代码或者逻辑出错而导致的功能性错误，如用户无法登陆，无法获取某一信息或接口写错参数，通常会导致产品功能的中断，从而影响业务流程进展。

面向对象：指任何事物需要抽象出事物的各个属性值或元素。对世界万物的状态与行为进行归纳与分类，以此分析个体与个体间的相互作用与影响方法。

SDK：Software Development Kit，一般都是一些软件工程师为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件时的开发工具的集合；

APP 推送：是通过推送技术自动传送信息给用户，根据用户的兴趣、偏好等定期推给用户，从而帮助产品运营人员更高效地实现运营目标，维持 APP 留存率。推送的原理一种叫 pull，也叫轮询，一种叫 push，目前常用的就是 push；

React Native：是 Facebook 推出的一个用 JavaScript 语言就能同时编写 ios，android 以及后台应用程序，并且它可以做到实时热更新，就像网页一样，改了一个字体，随时可上线；

AB 测试：是为 Web 或 App 界面或流程制作两个(A/B)或多个(A/B/n)版本，在同一时间维度，分别让组成成分相同（相似）的访客群组随

机的访问这些版本，收集各群组的用户体验数据和业务数据，最后分析评估出最好版本正式采用；

模拟器：用以在一台计算机上模拟另一台指令不兼容或者体系不同的计算机，如安卓模拟器。

抓包：（packet capture）就是将网络传输发送与接收的数据包进行截获、重发、编辑、转存等操作，也用来检查网络安全。

代理服务器：指在我们上网的过程中访问某个服务器的时候，并不是亲自访问真正的服务器，而是先找了一个代理，由它向真正的服务器发出请求。代理服务器架在客户端和真正服务器中间，干的是替客户端访问真正服务器的工作。

轻量级虚拟机：是一个容器，也可以说是 一个软件集装箱，这个箱子里面可以塞入特定版本的操作系统、数据库、服务器程序和 web 应用，这样一套完整的 web 服务就集成在这个箱子里面了，当要发布服务的时候，直接将这个集装箱放在我们的服务器船上。