

Web 后端导论

教程作者：V3n0

欢迎阅读 HDU-CS-Wiki 的后端教程！

这篇导论会快速为你建立对于整个计算机领域的认识，并让你了解后端开发在整个开发领域的位置。之后，我会尽可能浅显直白地为你解答小白对于后端的大多数疑问。如果还有更多的问题，那么欢迎提问。

后端简介

深入“后端”这个角色之前，我们先把镜头拉远，看看计算机领域的全貌。这是为了你能更好的理解后端是什么。

计算机领域整体速通

我们可以把整个计算机领域想象成从零开始建造并运营一座现代化的城市：

- **硬件 - 城市的基建**

这是城市的根基。水电，钢筋水泥，道路交通，就是电脑的CPU, 内存，硬盘，主板等等。没有基建，城市的一切建设都是空谈。

硬件是让计算成为可能的物理基础。

对应到开发领域，硬件工程师负责了这些硬件的设计制造。此外还有嵌入式开发，为特定的硬件编写贴近硬件的软件（也就是固件）。

- **操作系统 - 城市的市政**

为了利用上面的基建，我们需要有一个“市政府” -- **操作系统**。市政府负责维护一些基础的公共服务，并且保证了城市的秩序。

Windows, MacOS, Linux这些操作系统就是市政府，这是连接物理硬件与上层软件的桥梁。它们会和硬件设备交互，管理硬件资源，维持各进程有条不紊地共用资源，并提供一些公共的基础设施，比如进程管理，内存管理，文件系统，设备管理等等。

- **应用软件 - 城市的万千商铺**

有了基建和市政，我们终于能开店做生意了。假设说我们想开一家大饭店。

餐厅的前厅，当然就对应了 **前端开发**。这是顾客能直接看到和体验到的一切。

装修、菜单、桌椅、灯光、服务员，这对应着应用的用户界面（UI）。按钮长什么样，图片放在哪里，页面跳转的动画效果，点击按钮后的反应等。服务员热情地为你点餐，就如同一个友好的交互界面。

而后端开发，就是餐厅的后厨。

后端的地位

餐厅的后厨是顾客看不见，但决定了饭店生死存亡的核心部分。

- 厨师团队 (**核心业务逻辑**) :

他们根据菜谱（业务规则）处理食材（数据），制作出菜品（响应）。比如，用户下了一个订单，后端的厨师就要执行“减库存”、“计算价格”、“生成订单记录”等一系列操作。

- 仓库和货架 (**数据库**) :

所有的食材（用户信息、商品信息、订单数据）都必须分门别类、井井有条地存放在这里。需要什么食材，厨师要能快速、准确地拿到。仓库必须安全、防盗、防火（数据安全与备份）。

- 点餐的小票 (**API**) :

服务员（前端）把顾客的点单通过一个小票传到后厨（后端），后厨做好菜后，再通过这个系统通知服务员取餐。这张小票就是 API，它定义了前后端沟通的格式和语言，是连接前厅和后厨的桥梁。

- 厨房的标准化装修 (**后端框架**):

如果每个厨师都自己决定炉子放哪、案板多大、调料摆哪，后厨就会乱成一锅粥。框架就像统一装修方案：炉子、冰箱、切菜台、调料架的位置都设计好了，还配好燃气管道和下水系统。厨师（开发者）来了就能直接开工，不用自己从水电走线开始搞。

- 厨师长和经理 (**服务器与架构**) :

他要确保后厨运转高效。顾客多的时候（高并发），要加派人手（服务器扩容）；要设计好流水线（应用架构），让切菜、炒菜、洗碗协同工作；还要保证炉灶不断火（服务器稳定运行）。

所以，后端开发就是这家饭店的“后厨团队+仓库管理员+厨师长”的集合体。

他们不直接服务顾客，但他们决定了：

- 饭店能提供什么菜品（核心功能）。
- 菜品味道好不好（算法与逻辑的优劣）。
- 上菜速度快不快（系统性能）。
- 饭店能接待多少客人而不会崩溃（系统并发能力）。
- 客人的信息和交易安不安全（系统安全）。

看完上面的描述，你一定已经对后端是个什么很清楚了。计算机领域虽然确实很浩大，但是别被吓到。我们只需要对这个领域有一个整体的认识即可。就像是我们不需要背下百科全书，只需要学会怎样查百科全书。