# 拦截请求mock接口的数据

**拦截真实的网络请求，用我们自己定义的虚假数据（mock 数据）来代替接口的真实返回，从而实现不依赖后端服务也能调试前端功能。**

### 举个例子理解：

你可以把 Fiddler 想象成一个 "快递中转站"。

正常流程是：你的手机 / App（前端） → 发请求给服务器（比如请求 "我的余额"） → 服务器返回真实数据（比如 "100 元"） → 手机显示结果。

当你用 Fiddler"拦截请求并 mock 数据" 时：

1. 手机发的 "查余额" 请求，先到了 Fiddler 这个中转站

1. Fiddler 不把这个请求发给真正的服务器，直接自己编一个数据（比如 "10000 元"）

1. Fiddler 把这个编造的数据假装成服务器的回复，发给手机

1. 手机收到后，会当真的一样显示 "10000 元"

### 技术场景中的作用：

1. **后端接口没开发完**：前端可以先用 mock 数据调试页面展示、交互逻辑。

1. **模拟异常情况**：比如强制返回500错误、空数据，测试前端的错误处理能力。

1. **隔离环境干扰**：不需要依赖真实接口的不稳定数据（如实时变化的天气、股票接口）。

### Fiddler 实现的大致思路：

1. **拦截请求**：Fiddler 捕获到前端发送的目标接口请求（如/api/getUserInfo）。

1. **阻止真实请求**：不让这个请求真的发送到后端服务器。

1. **返回 mock 数据**：用我们提前写好的虚假数据（如{"name":"测试用户","age":20}）作为响应返回给前端。