

Web 基础概念入门

2025 软件工程课程

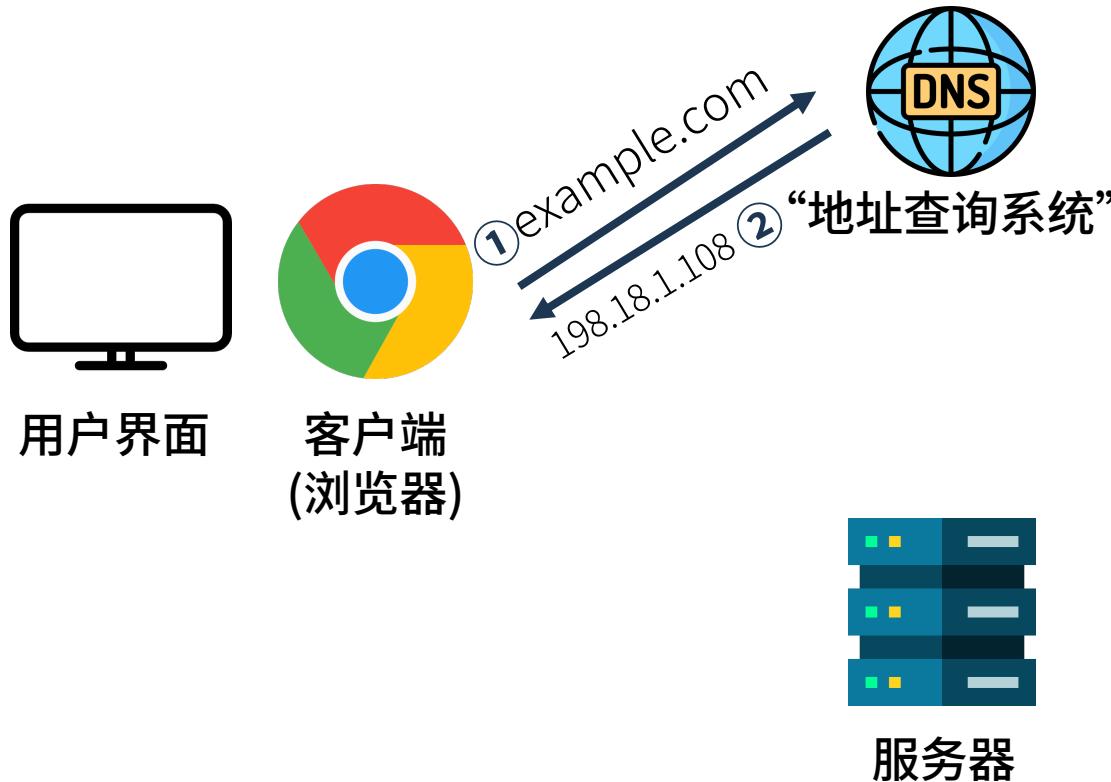
Web Concepts 101
2025 Software Engineering Course

软件工程实验室 林奕铖



当你打开一个网站时，发生了什么？

Step 1 解析服务器所在位置

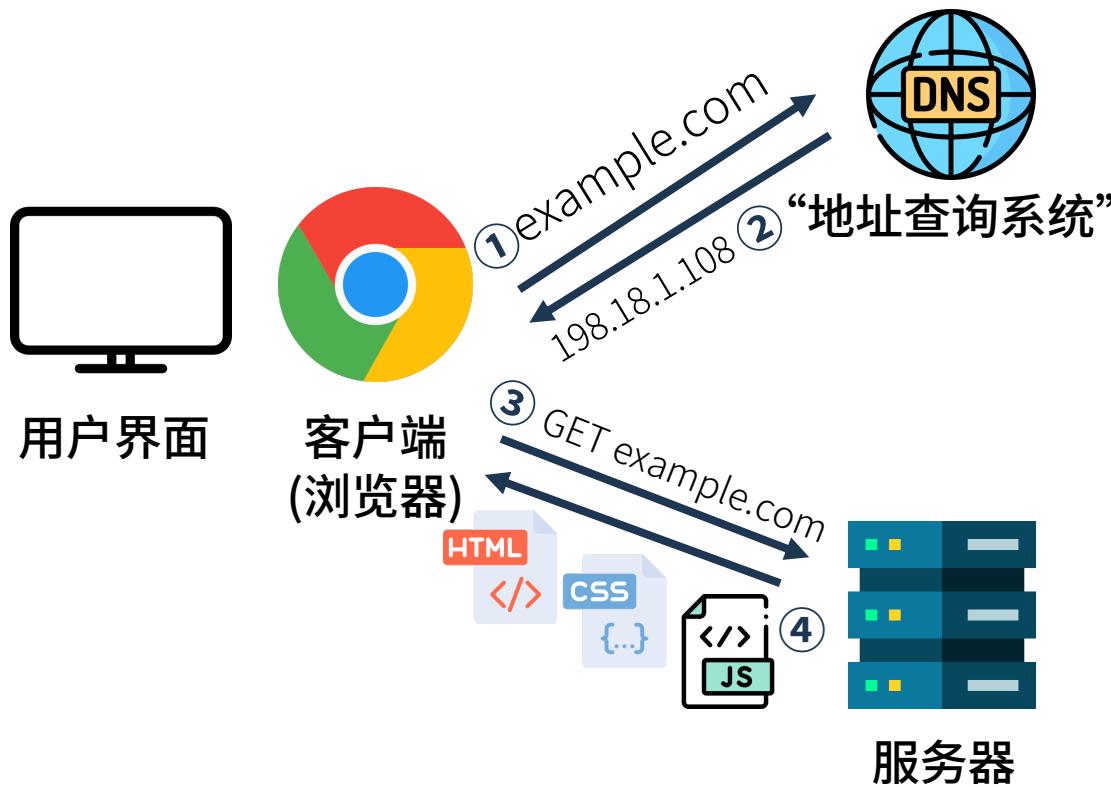


在地址栏输入网址并按下回车，浏览器先询问“地址查询系统”将域名解析成IP地址。

IP就相当于互联网中的地址与门牌号。

当你打开一个网站时，发生了什么？

Step 2 服务器处理请求并返回资源

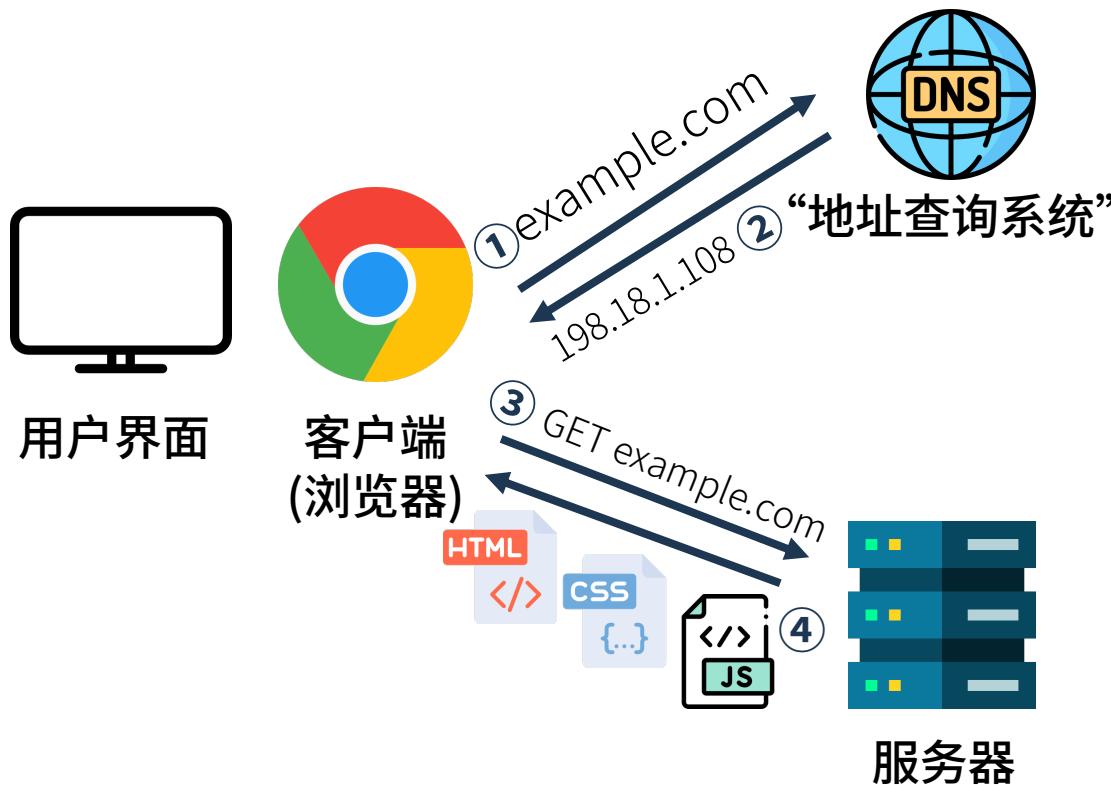


浏览器：向服务器发送请求，请求指定网页的内容。

服务器：根据请求路径，返回相应的文件，包括HTML、CSS和JavaScript文件。

当你打开一个网站时，发生了什么？

Step 2
服务器处理请求并返回资源



HTML 文件

网页有什么。

定义网页的结构，比如网页中有两段文字，每段文字配有一张图片。

CSS 文件

网页长什么样。

控制网页的外观，如字体、颜色、布局等。

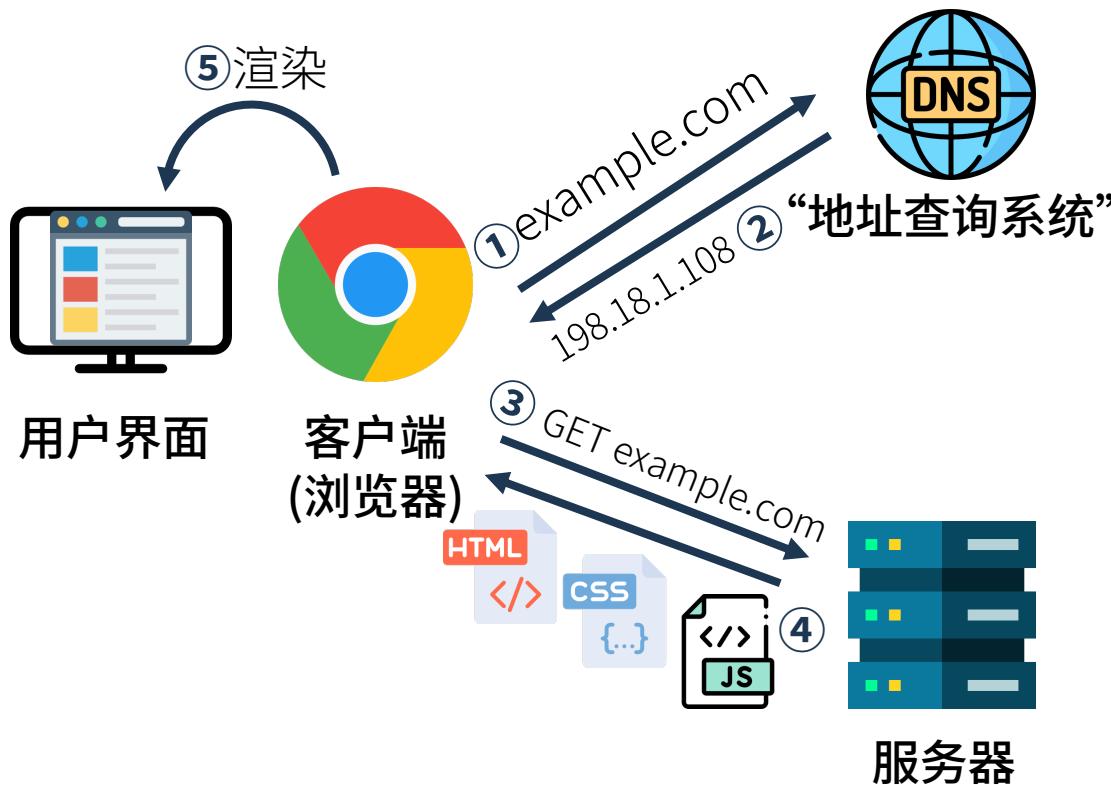
JavaScript 文件

网页有什么用。

让网页具有交互性，能处理点击、输入、动画等动态效果。

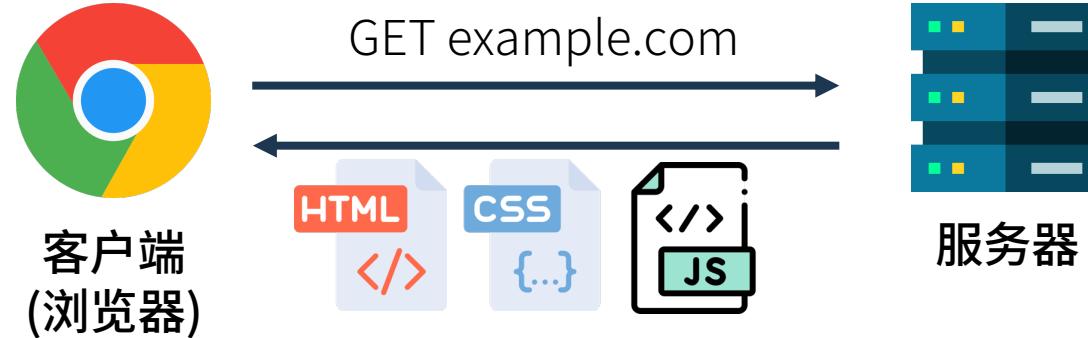
当你打开一个网站时，发生了什么？

Step 3 浏览器解析资源并呈现网页



浏览器根据服务器返回的文件，渲染网页并展示给用户。

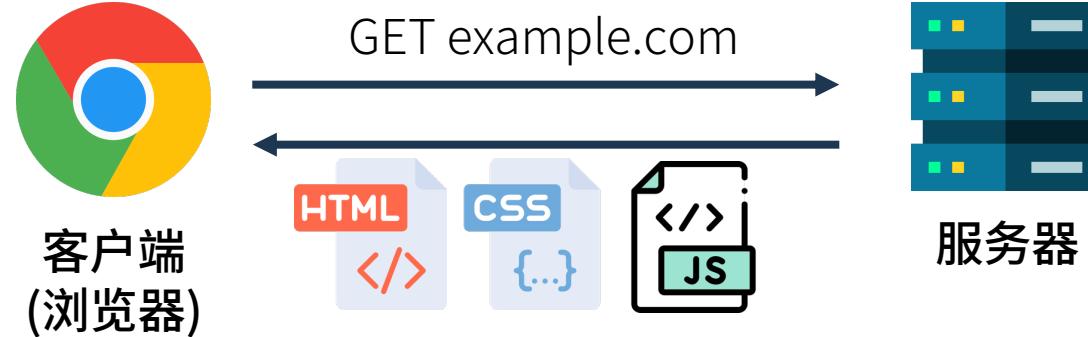
当你打开一个网站时，发生了什么？



Q1 浏览器起到了什么作用？

- 负责向服务器请求网页内容。
- 将服务器响应的文件解析并渲染为用户可以看到的页面。
- 负责用户交互，执行 JavaScript 代码实现动态效果。

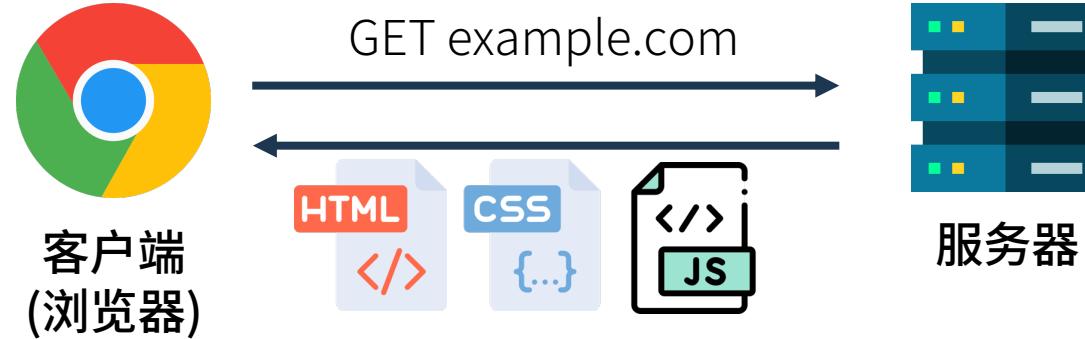
当你打开一个网站时，发生了什么？



Q2 服务器是什么，有什么作用？

服务器可以看作是一台特殊的电脑，它专门用来处理请求。不同于普通电脑，它通常需要24小时不停运转，接受其他设备的请求，并返回相应的数据。

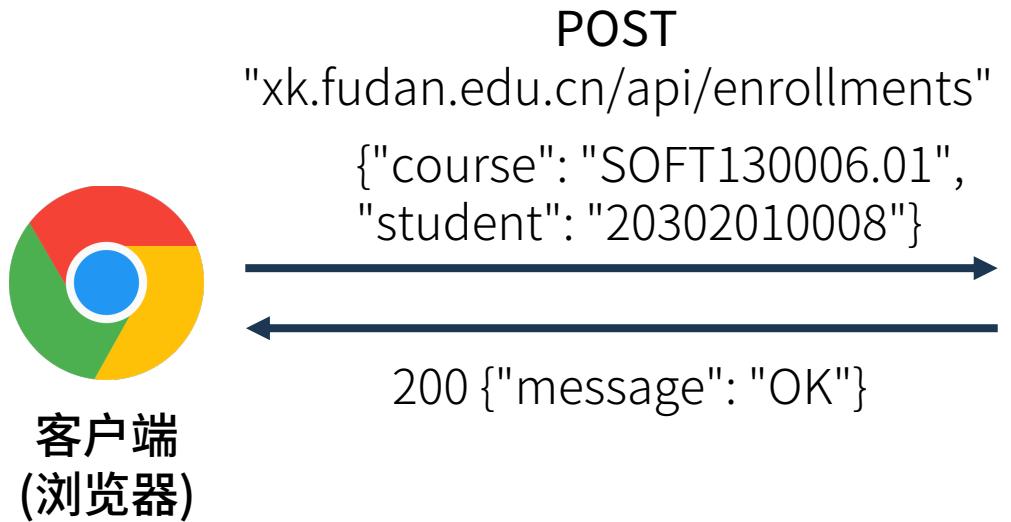
当你打开一个网站时，发生了什么？



Q3 什么是前端开发？

前端负责实现用户可以看到和互动的部分，包括网页设计、布局、动画等。
即提前准备好这些HTML、CSS和JavaScript文件放在服务器上，供用户获取。

当你点击选课按钮时，发生了什么？



网页光展示信息是不够的，还需要处理用户的请求。

当用户在网页上进行交互（如点击“选课”按钮）时，浏览器会向服务器发送HTTP请求，服务器处理后返回相应的结果，比如选课是否成功或者失败的原因。

当你点击选课按钮时，发生了什么？



POST "/api/enrollments"
200 {"message": "OK"}



Q1 服务器做了哪些事情？

1. 解析请求

提取请求中的内容，例如选课的课程序号和用户学号。

2. 处理逻辑

检查选课规则，如是否已满员、是否满足专业保护、是否与已有课程冲突等。

3. 访问数据库

查询课程信息，扣减课程余量，将选课记录写入数据库。

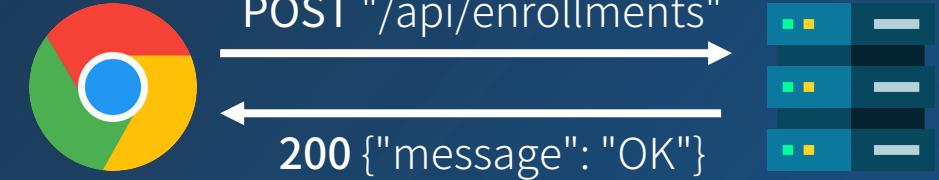
4. 返回响应

将选课结果（成功或失败）返回给客户端。

5. 安全性管理

包括验证用户身份、限制权限，防止恶意数据攻击。

当你点击选课按钮时，发生了什么？



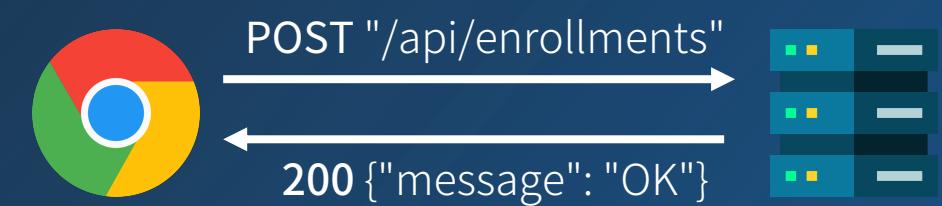
Q2 什么是后端开发？

后端负责处理网站的逻辑与数据，通过提供 **API** (Application Programming Interface, 应用程序接口)让前端访问。在Web开发的语境中，可以简单地将API理解为**后端提供操作的入口**。

当用户在前端（页面）进行操作时，

- 前端会向后端提供的API发送请求
- 后端接收请求、访问数据库、处理业务
- 把结果返回给前端

当你点击选课按钮时，发生了什么？



Q3 API是什么？客户端与服务端之间是如何沟通的？

HTTP (Hypertext Transfer Protocol, 超文本传输协议)
每次沟通包含并且只能包含一次**请求**和一次**响应**

一问一答

先问后答

服务器不能主动联系客户端

当你点击选课按钮时，发生了什么？



Q3 API是什么？客户端与服务端之间是如何沟通的？

HTTP (Hypertext Transfer Protocol, 超文本传输协议)

每次沟通包含并且只能包含一次**请求**和一次**响应**

一问一答

先问后答

服务器不能主动联系客户端

✓ **请求** | 响应

◆ URL地址

指定请求的目标，即访问的资源路径。

<https://example.com/api/user/20302010008>.

◆ HTTP方法

表示请求的类型

GET (获取) POST (创建)
PUT (更新) DELETE (删除)
等。

◆ 请求体

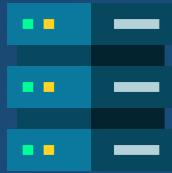
存放客户端要发送的数据，通常是JSON形式。

选课时包括学号、课程序号等。

当你点击选课按钮时，发生了什么？



POST "/api/enrollments"
200 {"message": "OK"}



Q3 API是什么？客户端与服务端之间是如何沟通的？

HTTP (Hypertext Transfer Protocol, 超文本传输协议)

每次沟通包含并且只能包含一次**请求**和一次**响应**

一问一答

先问后答

服务器不能主动联系客户端

请求 | 响应

◆ 状态码

表示请求的处理结果。

200 OK 请求成功

404 Not Found 请求的资源不存在

500 Internal Server Error 服务器内部错误

◆ 响应体

服务器返回的数据内容，通常是JSON形式。

例如，当请求用户信息时，响应体为
{"id": 1, "name": "Alice"}

当你点击选课按钮时，发生了什么？



Q3 API是什么？客户端与服务端之间是如何沟通的？

HTTP (Hypertext Transfer Protocol, 超文本传输协议)

每次沟通包含并且只能包含一次**请求**和一次**响应**

一问一答

先问后答

服务器不能主动联系客户端

API 是什么

API作为后端提供功能的入口，规定了URL、HTTP方法和请求数据格式。

A screenshot of a REST client interface. The top bar shows "POST /users". Below it are tabs for "请求" (Request), "响应定义" (Response Definition), "接口说明" (Interface Description), "预览文档" (Preview Document), "Mock" (Mock), and "registerUser" (with a play icon). The "Body" tab is selected, showing a JSON payload:

```
1 {
2   "name": "bob",
3   "password": "123456"
4 }
```

A screenshot of a REST client interface. The top bar shows "PUT /users/{userId}/username". Below it are tabs for "请求" (Request), "响应定义" (Response Definition), "接口说明" (Interface Description), "预览文档" (Preview Document), "Mock" (Mock), and "changeUsername" (with a play icon). The "Body" tab is selected, showing a JSON payload:

```
1 {
2   "name": "kiki"
3 }
```

当你点击选课按钮时，发生了什么？



Q4 什么是前后端分离？

前后端分离是一种开发方式，将网页的用户界面（前端）和后台逻辑（后端）分开开发。

- 各自独立开发和维护，提高效率。
- 前端和后端可以使用不同的技术栈，比如前端用 React/Vue，后端用 Java/Python，互不影响。
- 由于前后端通过API接口通信，只要遵循相同的接口规范，前端或后端可以独立替换或升级，而不会影响整个系统。