Web的核心脉络—HTTP协议与Tomcat的使命

The Core of Web: HTTP Protocol & Tomcat's Mission

Java Web 开发 (Java Web Application Development)

本讲概要 (Outline)

1. **问题引入 - 一次网页请求的背后** (Introduction: Behind a Web Request)
2. **HTTP协议 - Web的通用语** (HTTP: The Universal Language of the Web)
3. **Tomcat - Web应用的家** (Tomcat: Home for Web Applications)
4. **课堂总结与展望** (Summary & Outlook)

教学目标 (Teaching Objectives)

知识目标 (Knowledge Objectives)

- 理解HTTP协议是Web通信的基石 (Understand HTTP as the cornerstone of web communication)
- 了解HTTP请求和响应的主要组成 (Know the main parts of HTTP requests/responses)
- 掌握Tomcat作为Servlet容器的核心作用 (Master Tomcat's core role as a Servlet container)

能力目标 (Ability Objectives)

- 能够使用浏览器开发者工具分析HTTP报文 (Analyze HTTP messages with browser dev tools)
- 能够独立安装和启动Tomcat (Install and start Tomcat independently)

问题引入：一次网页请求的背后 (Introduction: Behind a Web Request)

一个熟悉的场景 (A Familiar Scenario)

当我们在浏览器地址栏输入 `http://www.tyust.edu.cn` 并敲下回车时...

思考 (Think About It):

- 浏览器是如何"告诉"服务器它想要什么？ (How does the browser "tell" the server what it wants?)
- 服务器又是如何"回应"的？ (How does the server "respond"?)

> 它们之间需要一种"通用语言"进行沟通，这就是 **HTTP协议**。
> (They need a "common language" to communicate. This is the **HTTP Protocol**.)

HTTP协议 - Web的通用语 (HTTP: Universal Language of the Web)

1. HTTP 请求 (The Request)

> 客户端发送给服务器的"信件" (A "letter" from client to server)

- **请求行 (Request Line):** `GET /index.html HTTP/1.1`
  - 我想用 `GET` 方法，获取 `/index.html` 资源
  - (I want to `GET` the `/index.html` resource)
- **请求头 (Headers):** `Host: www.tyust.edu.cn`
  - 附加信息，如我从哪个主机来
  - (Additional info, like which host I'm from)
- **请求体 (Body):**
  - 发送的具体数据 (GET请求通常为空)
  - (Specific data being sent - usually empty for GET requests)

HTTP协议 - Web的通用语 (HTTP: Universal Language of the Web)

2. HTTP 响应 (The Response)

> 服务器回复给客户端的"回信" (A "reply letter" from server to client)

- **状态行 (Status Line):** `HTTP/1.1 200 OK`
  - 告诉你请求处理的结果 (Tells you the result of the request)
- **响应头 (Headers):** `Content-Type: text/html`
  - 附加信息，如我给你的是什么类型的文件
  - (Additional info, like the type of file I'm giving you)
- **响应体 (Body):**
  - 实际的HTML代码内容
  - (The actual HTML code content)

生动演示 (Live Demo)

浏览器开发者工具 (Browser Developer Tools)

- 按下 `F12` 打开开发者工具 (Press `F12` to open developer tools)
- 切换到 **`Network`** **(网络)** 面板 (Switch to the **`Network`** panel)
- 刷新页面，观察真实的HTTP请求与响应 (Refresh the page and observe real HTTP requests and responses)

> **实践出真知** (Seeing is Believing)
> 故意访问一个不存在的URL，看看会发生什么？(`404 Not Found`)
> (Try visiting a non-existent URL. What happens? - `404 Not Found`)

Tomcat - Web应用的家 (Tomcat: Home for Web Applications)

为什么需要Tomcat？ (Why Do We Need Tomcat?)

- HTML 文件: 浏览器可以直接打开 (Browsers can open HTML files directly)
- **包含Java代码的动态页面**: 浏览器**不能**直接执行Java代码
  - (Dynamic pages with Java code: Browsers **cannot** execute Java code directly)

结论 (Conclusion):

> 我们需要一个特殊的软件，它既能接收HTTP请求，又能运行Java代码。
> (We need special software that can both receive HTTP requests and run Java code.)
> 这个"超级服务器"就是 **Servlet容器**，Tomcat是其中最著名的一员。
> (This "super server" is a **Servlet Container**, and Tomcat is the most famous one.)

什么是Tomcat？ (What is Tomcat?)

核心使命 (Core Mission)

Tomcat是Apache基金会下一个开源、免费的轻量级Web服务器和**Servlet容器**。
(Tomcat is an open-source, free, lightweight web server and **Servlet container** from the Apache Foundation.)

> 它的家就是用来装载和管理我们的Java Web应用的。
> (Its home is used to load and manage our Java Web applications.)

它负责：

1. 接收HTTP请求 (Receive HTTP requests)
2. 运行我们写的Java程序 (Run our Java programs (Servlets))
3. 将程序生成的动态内容组合成HTTP响应，再发回浏览器 (Combine the dynamic content into an HTTP response and send it back)

Tomcat 核心目录 (Tomcat's Core Directories)

> 各司其职 (Each has its own duty)

- `bin`: **启动器 (Launcher)**
  - 存放启动和关闭脚本 (`startup.bat`, `shutdown.bat`)
- `conf`: **配置中心 (Config Center)**
  - 存放服务器配置文件 (`server.xml`)
- `webapps`: **公寓楼 (Apartment Building)**
  - 存放我们开发的Web应用的地方
- `logs`: **日志室 (Log Room)**
  - 记录服务器运行日志，排错必备

部署第一个Web应用 (Deploying Your First Web App)

1. **创建应用 (Create App)**
   - 在 `webapps` 目录下创建一个 `myapp` 文件夹
2. **编写主页 (Write Homepage)**
   - 在 `myapp` 内创建 `index.html` 文件，内容为:
   - `<h1>Hello, Tomcat!</h1>`
3. **启动服务 (Start Service)**
   - 进入 `bin` 目录，双击 `startup.bat`
4. **验证访问 (Verify Access)**
   - 在浏览器访问: `http://localhost:8080/myapp/`

课堂总结 (Summary)

1. **通信靠协议 (Communication via Protocol)**
   - 浏览器和服务器通过HTTP协议"对话"
   - (Browsers and servers "talk" via the HTTP protocol)
2. **动态靠容器 (Dynamics via Container)**
   - Java Web应用必须住在像Tomcat这样的Servlet容器里
   - (Java Web apps must live in a Servlet container like Tomcat)
3. **部署很简单 (Deployment is Simple)**
   - 将应用文件夹放入Tomcat的`webapps`目录即可
   - (Just put the application folder into Tomcat's `webapps` directory)

展望 (Outlook)

下一节课 (Next Class)

今天，我们展示了一个静态页面。但这仅仅是冰山一角。
(Today, we displayed a static page. This is just the tip of the iceberg.)

下一节课，我们将亲手编写第一个**Servlet**程序，让我们的Web应用能真正"听懂"**用户请求并做出**动态回应！
(Next time, we will write our first **Servlet** program, allowing our web app to truly "**understand**" user requests and make **dynamic responses**!)

Q & A

有问题请随时提问

(Feel free to ask questions anytime)