观察网页结果

**一．观察开发者工具中显示的内容的发现**

筛选类型为 Fetch/XHR 的结果：请求类型：主要显示的是通过 Fetch 或 XHR（XMLHttpRequest）发起的异步请求。这些请求通常用于动态加载数据，如通过 AJAX 请求获取的 JSON 数据或其他资源。请求内容：这些请求通常与网页的动态内容相关，例如用户交互触发的数据加载、表单提交等。可以看到一些请求如 getThemeList.shtml、getdatalist.shtml 等，这些请求可能是用于获取网页的动态内容或用户交互数据。

筛选类型为 All 的结果：请求类型：显示所有类型的请求，包括 HTML、CSS、JavaScript、图片、字体、媒体文件等。请求内容：这些请求涵盖了网页加载过程中所需的所有资源，包括静态资源和动态资源。可以看到大量的请求，如 index3\_common.css、sjtu\_logo.png、swiper.min.js 等，这些请求分别对应网页的样式表、图片和 JavaScript 文件。

区别 请求数量：在 All 模式下，请求数量明显多于 Fetch/XHR 模式，因为前者包含了所有类型的资源请求。请求大小：不同类型的请求大小差异较大，例如图片和字体文件通常较大，而 CSS 和 JavaScript 文件相对较小。加载时间：不同资源的加载时间也不同，较大的文件（如图片）可能需要更长的时间加载。总结来说就是，Fetch/XHR：主要用于动态内容的加载，通常与用户交互或网页的动态更新相关。而All：涵盖了网页加载过程中所需的所有资源，包括静态和动态内容。

此外经过我的查阅还有一些其余的发现：

1. 请求的状态码（Status Code）

200 OK：大多数请求的状态码为200，表示请求成功。例如，index3\_common.css、sjtu\_logo.png 等资源的请求都返回了200状态码。

2. 请求的发起者（Initiator）

HTML 文件：通常由浏览器直接发起，作为网页的入口点。

CSS 和 JavaScript 文件：通常由 HTML 文件中的 <link> 和 <script> 标签发起。

图片和其他媒体：通常由 HTML 或 CSS 文件中的引用发起。

异步请求（XHR/Fetch）：通常由 JavaScript 代码发起，用于动态加载数据或更新内容。

3. 请求的大小和加载时间

大小：不同资源的大小差异较大。例如，图片文件（如 sjtu\_logo.png）通常较大，而 CSS 和 JavaScript 文件（如 index3\_common.css、swiper.min.js）相对较小。

加载时间：较大的文件（如图片）通常需要更长的时间加载，而较小的文件（如 CSS 和 JavaScript）加载时间较短。

4. 缓存的使用

磁盘缓存（Disk Cache）：许多资源的来源显示为“磁盘缓存”，这意味着这些资源之前已经被加载过，并且被缓存到了本地磁盘。这可以显著减少重复加载相同资源的时间。

禁用缓存：如果禁用了缓存，所有资源都将从服务器重新加载，这会导致加载时间增加。

5. 请求的顺序和依赖关系

顺序：浏览器通常按照 HTML 文件中引用的顺序加载资源。例如，先加载 CSS 文件，然后加载 JavaScript 文件，最后加载图片和其他媒体。

依赖关系：某些资源可能依赖于其他资源的加载。例如，JavaScript 文件可能依赖于某些 CSS 文件的加载，以确保正确的样式应用。

6. 网络性能分析

总请求数量：在 All 模式下，可以看到总请求数量较多，这反映了网页的复杂性。

总传输大小：总传输大小（如2.0 MB/6.6 MB）显示了网页加载过程中传输的数据量。

加载时间：总加载时间（如1.1分钟）反映了网页从开始加载到完全渲染所需的时间。

7. 资源类型和用途

CSS 文件：用于定义网页的样式和布局。例如，index3\_common.css 可能包含了网页的通用样式。

JavaScript 文件：用于实现网页的交互逻辑和动态内容加载。例如，swiper.min.js 可能是一个用于实现图片轮播的库。

图片文件：用于丰富网页的视觉内容。例如，sjtu\_logo.png 是网页的 logo 图片。

字体文件：用于定义网页中使用的字体。例如，DM-Sans-regular.woff2 是一个自定义字体文件。

**二．Response各部分内容的含义**

在HTTP通信中，Response由三个部分组成：状态行（Status Line）、响应头（Response Headers）和响应体（Response Body）。

1. 状态行（Status Line）

状态行是HTTP响应的第一行，包含三个部分：HTTP版本、状态码和状态消息。例如：

HTTP/1.1 200 OK

HTTP版本：表示服务器使用的HTTP版本，例如HTTP/1.1。

状态码：表示请求的处理结果，200表示请求成功。

状态消息：对状态码的简短描述，例如“OK”。

2. 响应头（Response Headers）

响应头包含服务器返回的附加信息，告诉客户端如何处理响应体。常见的响应头包括：

Content-Type：指定响应体的MIME类型，例如text/html; charset=utf-8表示HTML内容，字符编码为UTF-8。

Content-Length：指定响应体的字节长度。

Server：指定服务器软件的名称和版本，例如Apache/2.4.18。

Cache-Control：指定缓存机制，例如no-cache表示不缓存。

Date：指定响应的日期和时间。

3. 响应体（Response Body）

响应体是HTTP响应的实际内容，可以是HTML、JSON、图像、视频等。在这里响应体是一个完整的HTML页面。

3.1 HTML结构

HTML页面由<html>标签包裹，包含<head>和<body>两个主要部分。

3.1.1 头部（<head>）

头部包含页面的元数据和资源链接，例如：

字符集声明：<meta charset="utf-8">指定页面的字符编码为UTF-8。

视口设置：<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">用于响应式布局。

标题：<title>上海交通大学中文主页门户网站</title>指定页面的标题。

内联样式：<style>...</style>定义页面的内联CSS样式。

**CSS链接**：<link rel="stylesheet" href="/resource/assets/css/ETUI/ETUI3.min.css">等链接页面使用的CSS文件。

3.1.2 主体（<body>）

主体包含页面的可见内容，例如：

容器和布局：使用<div>标签创建容器和布局，例如<div id="main-container">。

导航和菜单：可能包含导航栏、菜单等，虽然在你的例子中没有详细展示。

内容区域：包含新闻、通知、学术信息等具体内容，使用<div>、<ul>、<li>等标签组织内容。

图片和链接：使用<img>和<a>标签插入图片和创建链接。

脚本：在<body>结束前引入JavaScript文件，例如<script src="/resource/assets/js/main.js"></script>，用于实现页面的动态功能。

3.2 JavaScript和CSS

页面中引入了多个JavaScript和CSS文件，用于实现页面的样式和功能：

JavaScript文件：例如/resource/assets/js/ETUI/jquery.min.1.12.4.js、/resource/assets/js/main.js等，用于实现页面的交互效果、动画、响应式布局等功能。

CSS文件：例如/resource/assets/css/ETUI/ETUI3.min.css、/resource/assets/css/style409.css等，用于定义页面的样式和布局。

3.3 图标和字体

页面中使用了图标字体（例如iconfont），通过<i class="iconfont icon-jiahao"></i>等标签插入图标，这些图标通常是从图标字体库中加载的。

3.4 内联脚本

页面中包含一些内联JavaScript代码，用于初始化页面组件、设置滚动效果、轮播图等功能。例如：

这段代码表示在页面加载完成后调用Main.init()函数初始化页面。

总结

这个HTTP响应返回的是一个完整的HTML页面，包含头部和主体内容。头部定义了页面的元数据和样式，主体包含页面的具体内容和结构。页面通过引入外部的CSS和JavaScript文件来实现复杂的样式和动态功能。响应头和状态行虽然在提供的内容中没有显示，但它们是HTTP响应不可或缺的部分，用于传达服务器的响应信息和控制客户端的行为。