

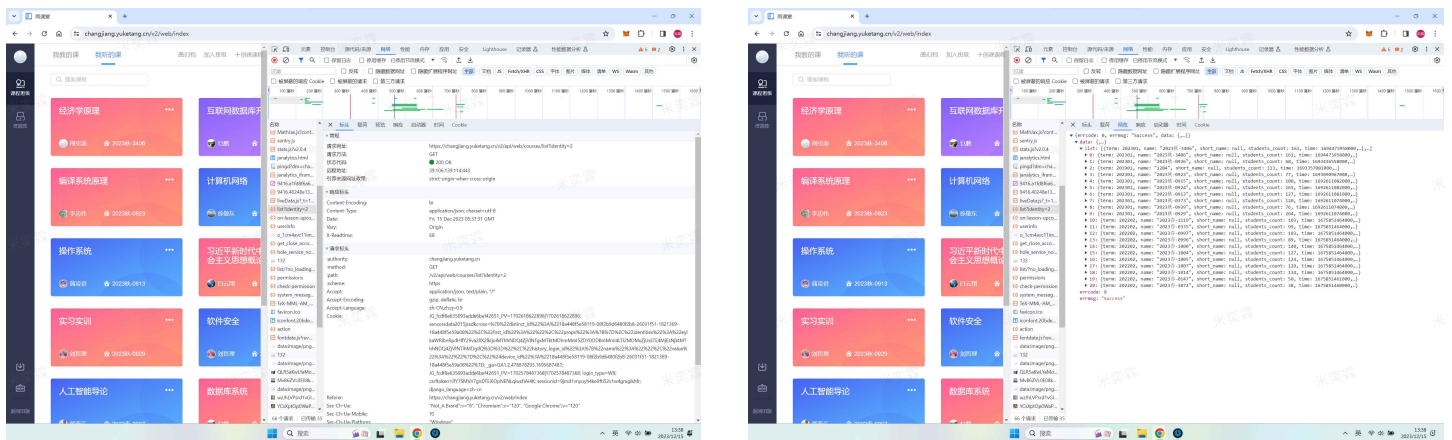
作业1

学号: 2111566 姓名: 米奕霖

1. 针对任意网页, 调研其不同方式请求, 至少包括get、post请求, 写出或截图其请求及相应数据包

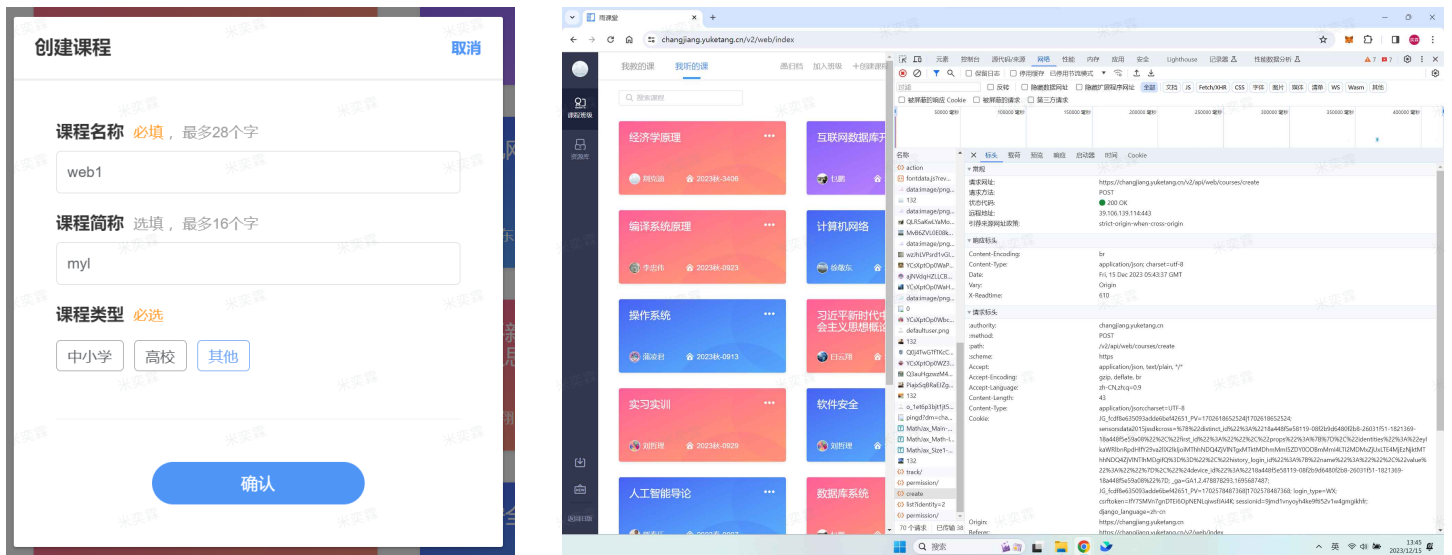
(1) get请求

对于与课堂界面, get请求list?identity=2我们发现是请求一些课程信息。点击预览可以查看响应的内容, 显然这就是页面上显示出来的课程的数据。我们可以查看本课程的相关信息, 可以得到classroom_id、students_count以及教师相关的一些信息。



(2) post请求

创建课程web1, 发现post请求create, 点击查看其代码, 发现创建课程的代码如下。



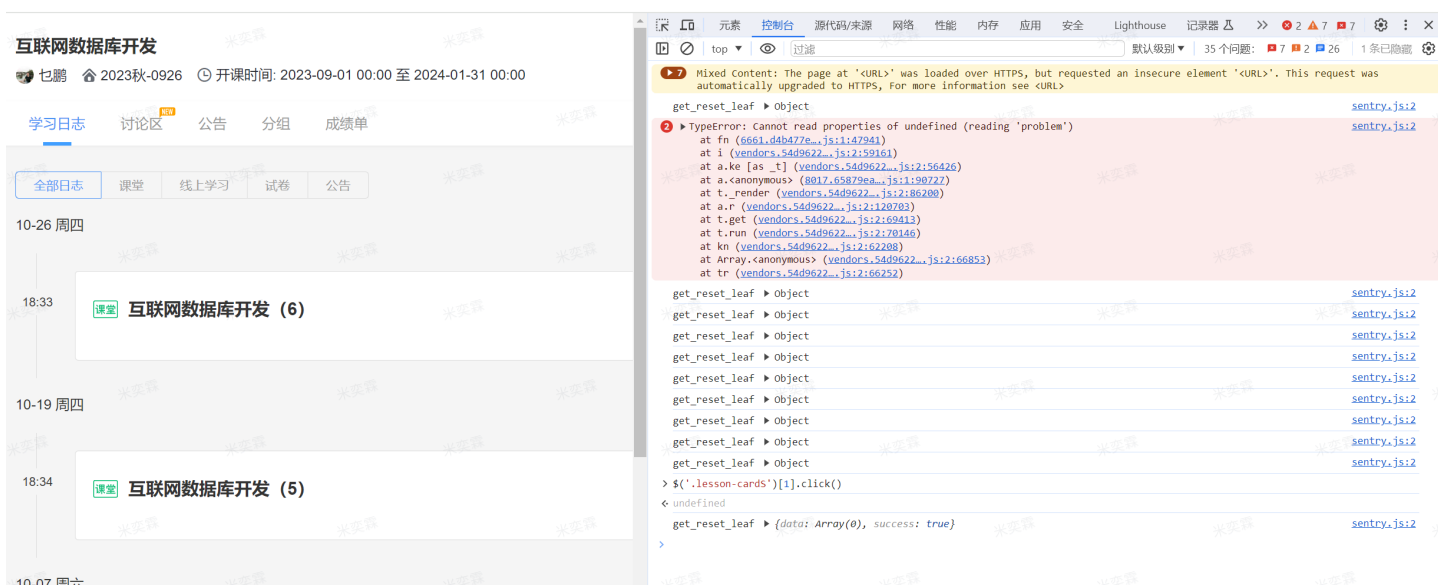


2. 针对任意网页，使用Jquery，能够触发某一事件， 写出至少三条语句，截图响应前后不同的状态

(1) 进入课程语句

`$('.lesson-cardS')[1].click()`

找到课程卡片的class，将对应的卡片进行点击得到如下效果，发现从雨课堂界面进入到了课程里面。



(2) 点击收藏语句

发现`$('.layout_right_switch').children().eq(0)`对应的为收藏按钮，对该按钮进行click达到收藏效果

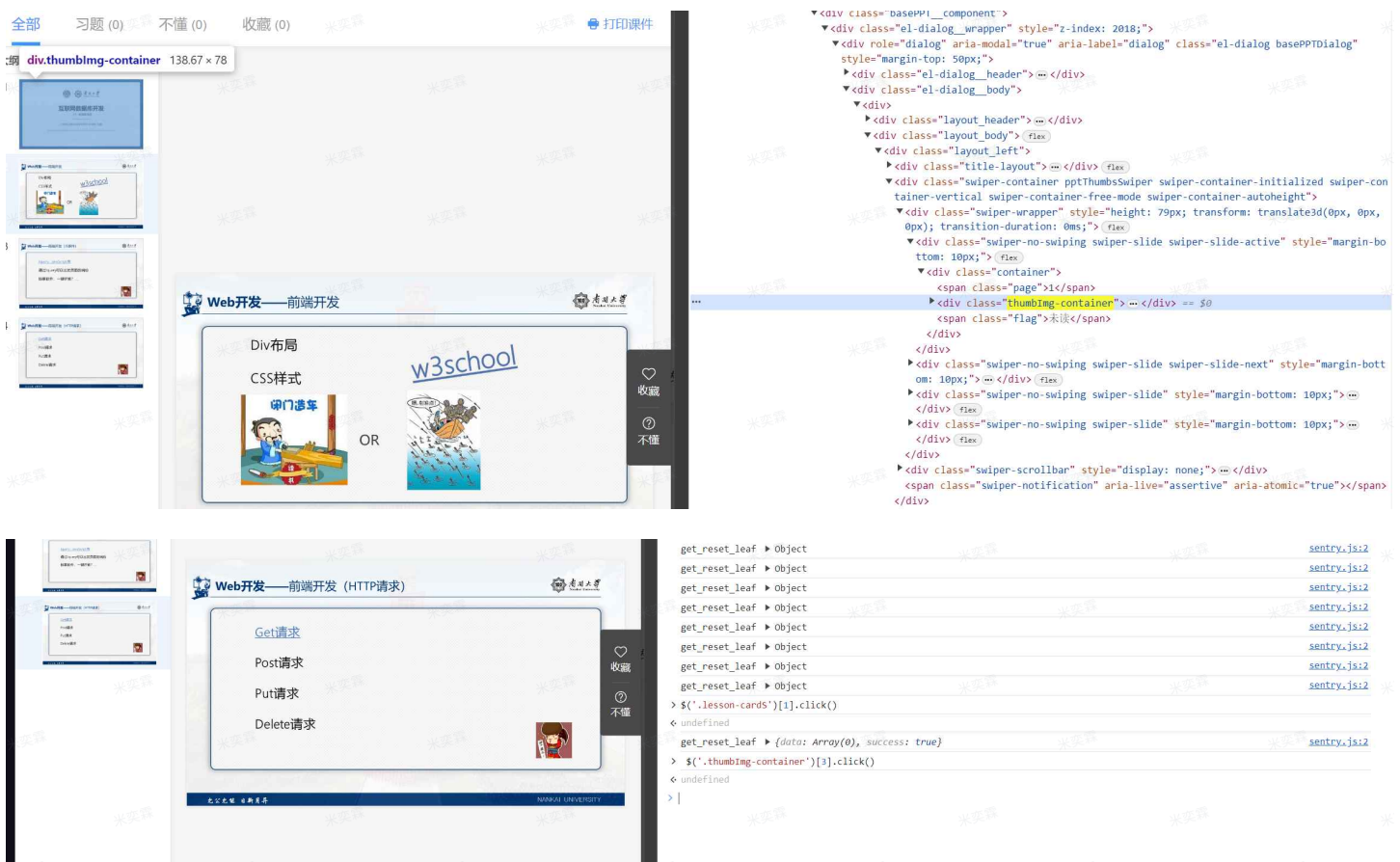
`$('.layout_right_switch').children().eq(0).click()`



(3) 切换页面语句





发现页面是thumbImg-container控制，如果我点击第四页则会跳转到第四页，执行语句实现页面跳转。

```
$('.thumbImg-container')[3].click()
```



3. 完成一个浏览器插件，功能不限，文档中写明功能及代码

代码讲解。代码结构如下

 background.js	2023/12/15 16:03	JS 文件	0 KB
 manifest.json	2023/12/15 16:22	JSON 文件	1 KB
 popup.html	2023/12/15 16:02	Microsoft Edge HT...	1 KB
 popup.js	2023/12/15 16:23	JS 文件	1 KB

(1).manifest.json

```
1 {
2   "manifest_version": 3,
3   "name": "background_color",
4   "version": "1.0",
5   "action": {
6     "default_popup": "popup.html",
7     "default_icon": {
8       "16": "16.png",
9       "48": "48.png",
10      "128": "128.png"
11    }
12  },
13  "background": {
14    "service_worker": "background.js"
15  },
16  "permissions": ["activeTab", "scripting"]
17 }
```

`manifest.json` 文件定义了扩展程序的元数据和行为。以下是文件内容的简要说明：

- `manifest_version`: 3 表示这是Manifest V3的扩展。
- `name`: "background_color" 是扩展程序的名称。
- `version`: "1.0" 是扩展程序的版本号。
- `action`: 定义了扩展的弹出行为，包括弹出页面 `popup.html` 和默认图标。
- `background`: 指定了后台服务工作线程 `background.js`，这是在Manifest V3中取代传统后台页面的方法。
- `permissions`: 列出了扩展所需的权限，包括修改当前活动标签页和执行脚本的能力。

(2)popup.js

```
1 document.addEventListener('DOMContentLoaded', function () {
2   var changeColorButton = document.getElementById('changeColor');
3   changeColorButton.addEventListener('click', function () {
4     var color = document.getElementById('colorPicker').value;
5     chrome.tabs.query({active: true, currentWindow: true}, function(tabs) {
6       chrome.scripting.executeScript({
7         target: {tabId: tabs[0].id},
```

```

8      function: setPageBackgroundColor,
9      args: [color]
10    });
11  });
12 }, false);
13 }, false);
14
15 function setPageBackgroundColor(color) {
16   document.body.style.backgroundColor = color;
17 }

```

1. 当文档加载完成时执行: `document.addEventListener('DOMContentLoaded', function () {...}, false);` 这部分代码会在弹出页面的DOM内容加载完成后执行。
2. 获取按钮并添加点击事件监听器:
 - 通过 `document.getElementById('changeColor')` 获取ID为 `changeColor` 的按钮。
 - 给这个按钮添加一个点击事件监听器。
3. 获取颜色并执行脚本:
 - 当按钮被点击时, 通过 `document.getElementById('colorPicker').value` 获取颜色选择器的值。
 - 使用 `chrome.tabs.query` 获取当前活动的标签页信息。
 - 通过 `chrome.scripting.executeScript` 在当前标签页上执行 `setPageBackgroundColor` 函数, 将页面的背景颜色更改为所选颜色。
4. 设置页面背景颜色的函数:
 - `function setPageBackgroundColor(color) {...}` 定义了一个改变背景颜色的函数, 它将文档的背景色设置为传入的颜色值。

(3)popup.html

```

1 <!doctype html>
2 <html>
3   <head>
4     <title>Change Background Color</title>
5   </head>
6   <body>
7     <input type="color" id="colorPicker">
8     <button id="changeColor">Change Color</button>
9     <script src="popup.js"></script>
10  </body>
11 </html>

```

1. HTML声明: `<!doctype html>` 表明这是一个HTML5文档。
2. HTML头部:
 - `<head>` 包含了页面的元数据, 如 `<title>Change Background Color</title>` 定义了弹出页面的标题。
3. HTML主体:
 - `<body>` 中包含了用户交互的元素, 包括一个颜色选择器 `<input type="color" id="colorPicker">` 和一个按钮 `<button id="changeColor">Change Color</button>`。
 - 弹出页面上有一个颜色选择器, 允许用户选择颜色, 并有一个按钮来应用这个颜色。
4. JavaScript引用:
 - `<script src="popup.js"></script>` 引入了 `popup.js` 文件, 它包含了处理用户交互和更改页面颜色的JavaScript代码。

(4)将扩展程序安装到chrome浏览器中

进行改变背景颜色。具体效果如下。

所有扩展程序

