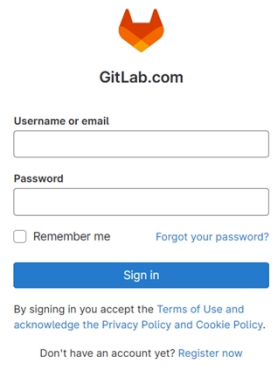
个人作业 1——Web 前端初探

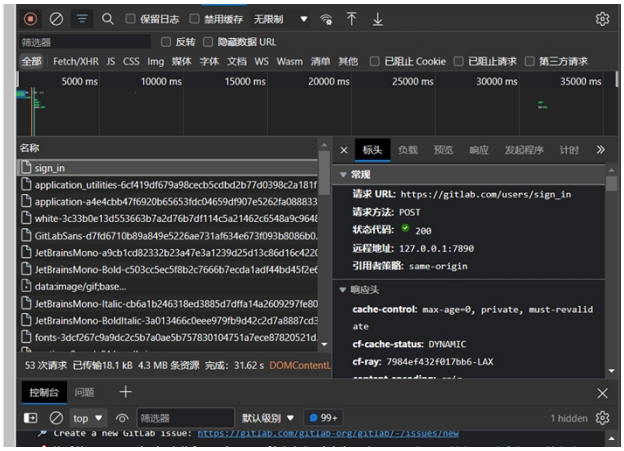
——黄逸轩 2012067

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

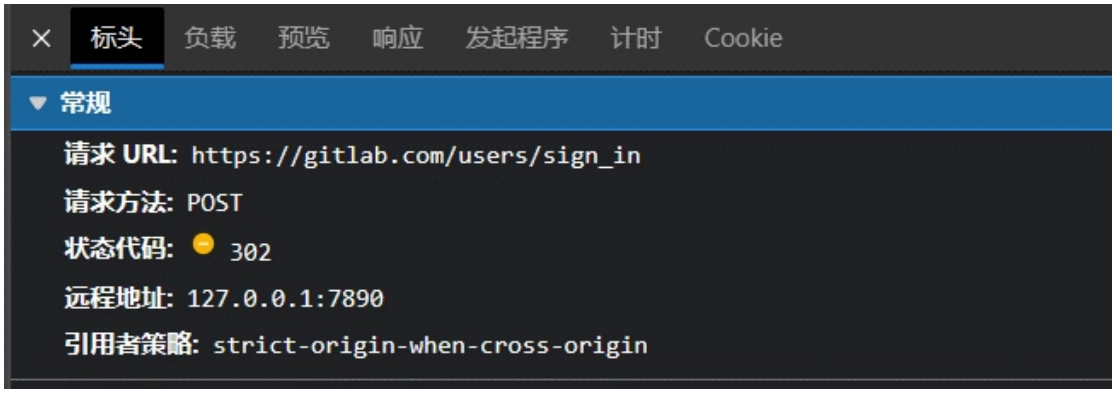
对于 gitlab 网站：

1）其登录页面可以看到请求方式为 post 请求



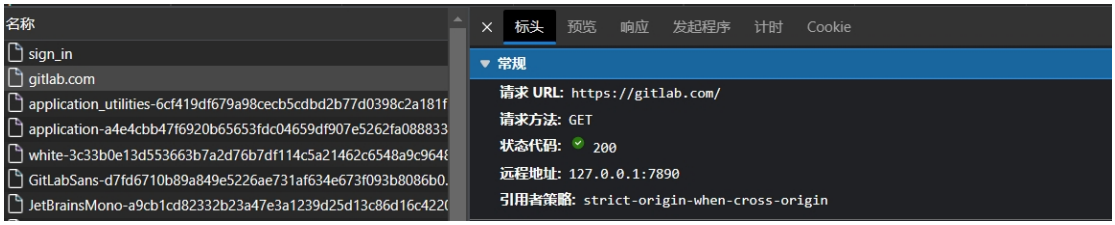


其具体请求内容如下：

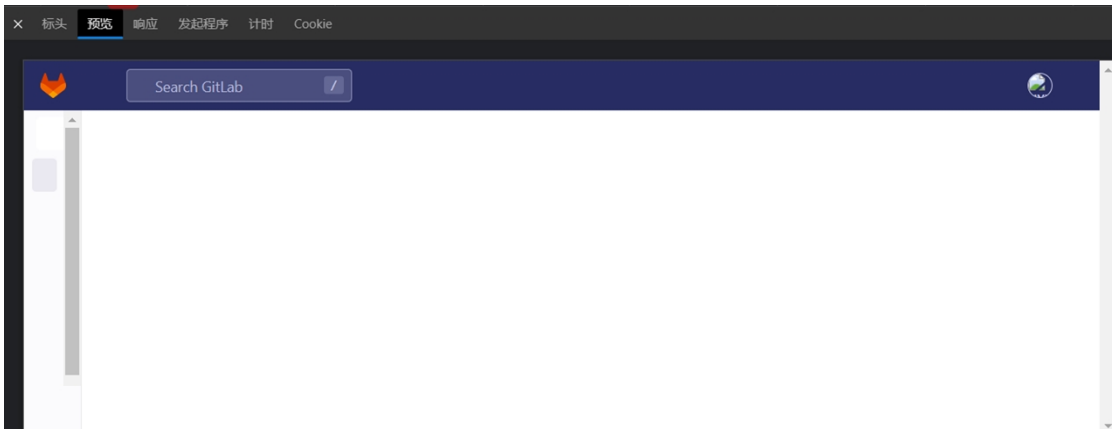


其中 302 是 http 协议中的一个状态码。可以简单的理解为该资源原本确实存在，但已经被临时改变了位置。网络资源地址临时性重定向，用户访问一个 url 地址时，被临时重定向了另一个 url 地址上，一般用于临时性网站页面跳转。于负载中可以看到提交的 POST 请求的具体内容，即账号密码：

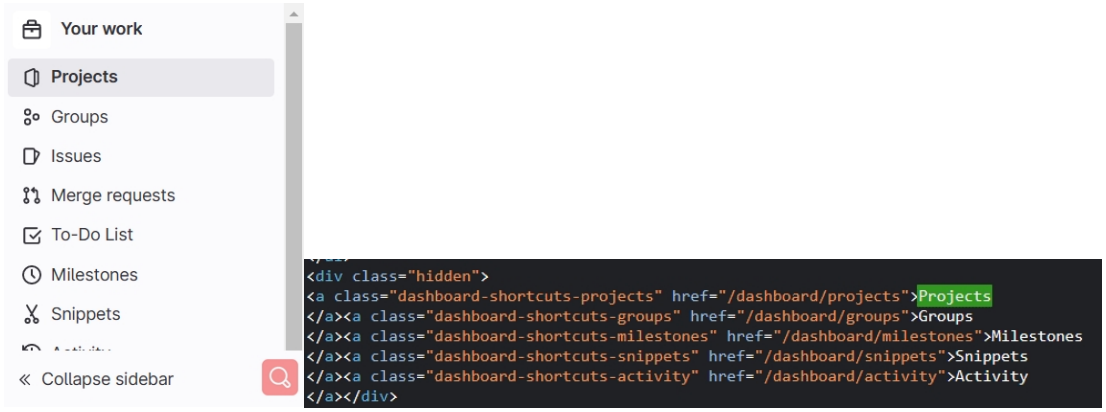
2）对于 gitlab 登录后的基本页面，为 GET 请求：



其可视化内容可以通过预览来查看：



即页面的一个基本内容，可以看到如下图所示，相应页面中的内容与代码内容一致：



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，截图相应前后不同的状态

下面给出网页的代码：

C: > Users > yukhu > Desktop > test.html > html

```
<!DOCTYPE html>
<html>
<head>
    <title>事件示例</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(function() {
            $("#button1").click(function() {
                alert("按钮1被点击");
            });

            $("#input1").focus(function() {
                $(this).css("background-color", "#e6e6e6");
            });

            $("#input1").blur(function() {
                $(this).css("background-color", "#fff");
            });

            $("#button2").dblclick(function() {
                alert("按钮2被双击");
            });

            $("#input2").on("keyup", function() {
                var value = $(this).val();
                $("#output").text(value);
            });
        });
    </script>
</head>
```

```
<body>
    <h1>事件示例</h1>

    <button id="button1">点击我</button>
    <button id="button2">双击我</button>

    <p>输入框:</p>
    <input type="text" id="input1" placeholder="在此输入文本">
    <input type="text" id="input2" placeholder="在此输入文本">
    <p>输出:</p>
    <div id="output"></div>

</body>
</html>
```

- 当点击#button1 按钮时，会弹出一个警告框；
- 当#input1 输入框获得焦点时，它的背景颜色会改变；
- 当#input1 输入框失去焦点时，它的背景颜色会恢复到原来的颜色。
- 当双击#button2 按钮时，会弹出一个警告框；

- 当#input2 输入框中的文本发生变化时，它的值将显示在 #output 元素中

下图是网页的视图：

点击第一个按钮后，弹出窗口：

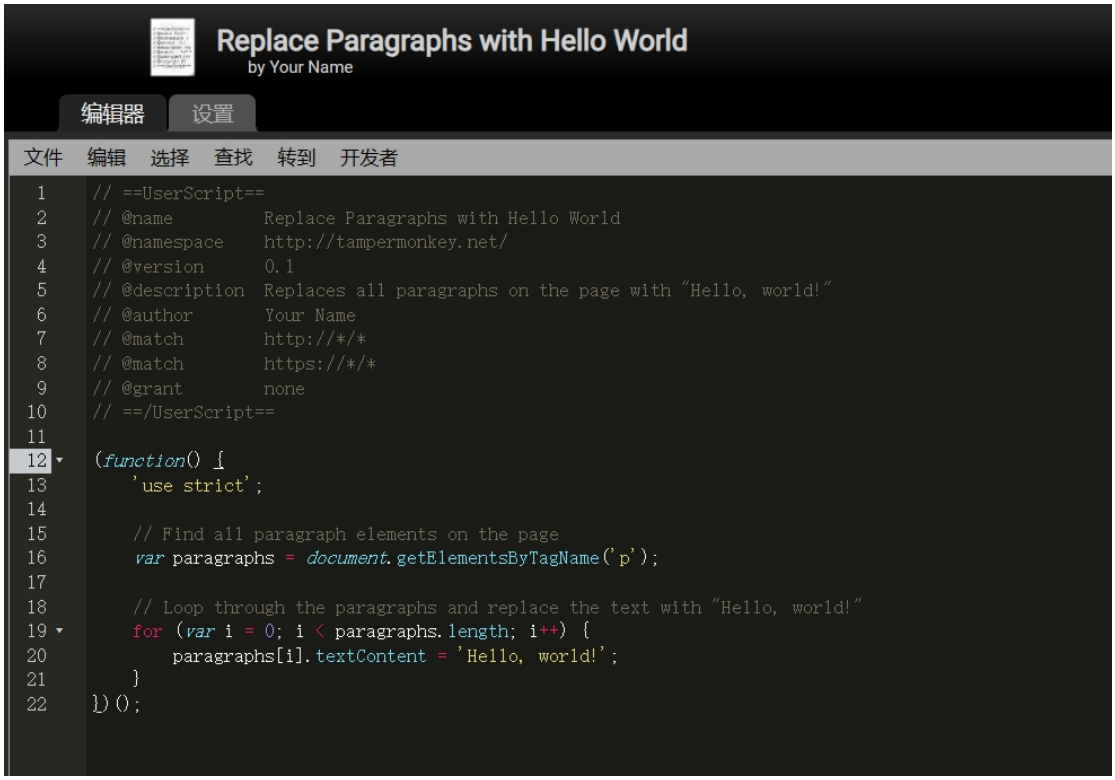
双击第二个按钮后，弹出窗口：

在左侧第一个输入框输入内容时，其背景色变灰：

在右侧文本框输入内容时，会同步被输出到下方空白处：

3、完成一个浏览器插件，功能不限，文档中写明功能及代码

下面脚本可以将页面中的所有脚本替换成 hello world：

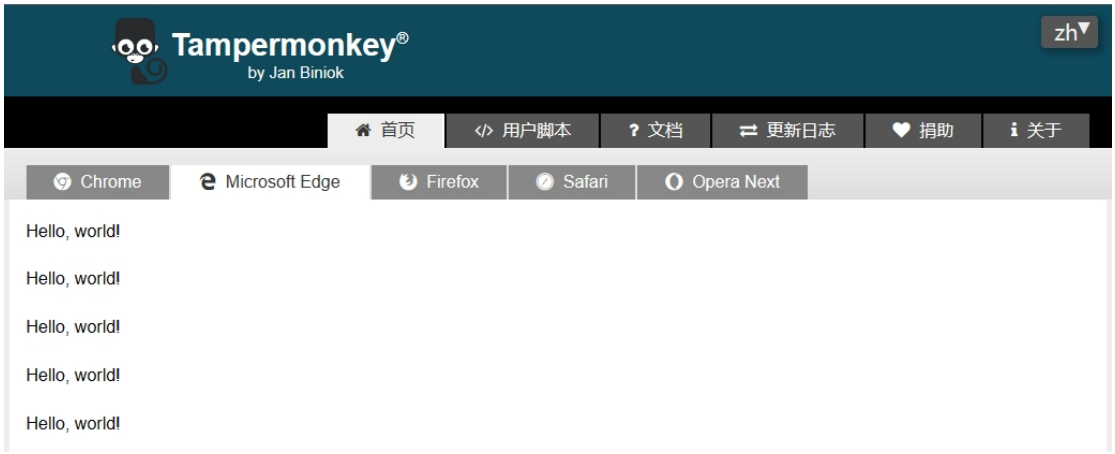


- // ==UserScript== 和 // ==/UserScript== 之间的代码是脚本的元数据信息，例如脚本名称、版本、描述等。这些信息是浏览器插件管理器使用的，通常不会直接在脚本中使用。
- (function() {...})(); 是一个自执行函数，用于封装脚本的代码。这可以确保脚本中定义的变量和函数不会与页面上的其他代码发生冲突。
- var paragraphs = document.getElementsByTagName('p'); 查找页面上的所有段落元素。
- for (var i = 0; i < paragraphs.length; i++) {...} 循环遍历所有段落元素。
- paragraphs[i].textContent = 'Hello, world!'; 用“Hello, world!”替换每个段落的文本内容。

下面是网站页面使用改脚本前后的对比：

使用前：

使用后：



可以看到，每一个段落内容都被替换为了 Hello，world！