

个人作业1-3 设计浏览器插件

问题发现

我们在使用浏览器的时候经常会出现以下的场景：



当我们要从一个页面跳转至另外一个页面的时候，会出现这样的一个拦截窗口。他的目的是让我们留意目标网页的安全性。但是事实上我们早就要前往那个页面，并不会因为这样的提示而停止我们的操作。我们检查这个url：

<https://link.csdn.net/?target=http%3A%2F%2Fwww.ruanyifeng.com%2Fblog%2F2014%2F11%2Fcompiler.html>

发现实际上是把目标链接放在了target里面。那么我们能不能实现一个浏览器插件直达target呢。

	组织	新建	打开	边
5.大三上 > 互联网 > gototarget				
名称	^			
manifest.json	修改日期	2021/11/28 16:47	类型	JSON File

创建一个gototarget的项目，在里面配置manifest的配置文件。

```
{
  "name": "gototarget",
  "description": "查询中间站点的target, 然后不显示直接访问",
  "version": "0.0.1",
  "manifest_version": 3,
  "permissions": ["storage", "tabs", "scripting"],
  "background": {
    "service_worker": "./background.js"
  },
  "action": {
    "default_popup": "popup.html",
  }
}
```

```

    "default_icon": {
      "16": "/pics/logo16.png",
      "32": "/pics/logo32.png",
      "48": "/pics/logo48.png",
      "128": "/pics/logo128.png"
    }
  },
  "icons": {
    "16": "/pics/logo16.png",
    "32": "/pics/logo32.png",
    "48": "/pics/logo48.png",
    "128": "/pics/logo128.png"
  }
}

```

然后找好图标按照对应文件名放进pics文件夹。然后新建一个入口文件：background.js，这样这个项目就已经可以放进浏览器里。



可以看到这里已经将插件移入了浏览器

编辑js文件：

```

chrome.runtime.onInstalled.addListener(() => {
  chrome.storage.sync.set({
    linkopen: true,
  });
});

chrome.tabs.onUpdated.addListener((tabId, changeInfo, tab) => {
  if (changeInfo.status === "complete" && /^http/.test(tab.url)) {
    chrome.scripting
      .executeScript({
        target: { tabId: tabId },
        files: ["/content-script.js"],
      })
      .then(() => {
        console.log("INJECTED SCRIPT SUCC.");
      })
      .catch((err) => console.log(err));
  }
});

```

代码逻辑也比较简单，就是用户首次安装插件的时候会执行一次这个文件，然后之后的逻辑会交给content-script。

```

chrome.storage.sync.get("linkOpen", ({ linkOpen }) => {
  if (linkOpen) {
    document.body.addEventListener("click", function (event) {
      const target = event.target;
      if (target.nodeName.toLocaleLowerCase() === "a") {
        const href = target.getAttribute("href");
        if (href.indexOf("://link") > -1) {
          event.preventDefault();
          const link = href.split("target=")[1];
          const url = decodeURIComponent(link);
          if (target.getAttribute("target") === "_blank") {
            window.open(url);
          } else {
            window.location.href = url;
          }
        }
      }
    });
  }
});

```

根据需求出发。这段代码会在每次点击的时候判断点击的是不是a标签，然后判断是不是中间链接，随后决定不打开这个链接，直接检查target的地址，然后前往目标地址。这样执行起来就再也不会被中间链接拦截。

然后我们再配置一下两个文件popup.html和popup.js。这样我们就可以在浏览器以开关的形式开启这个插件了。