# 作业一：

针对知乎网站，调研该网站下的不同网络请求方式。

## GET请求：

* 打开5月24日的[知乎热榜](https://www.zhihu.com/hot)之后，呈现如图1-1所示的部分界面。选取排名第一的热点话题，在点击该热点之前，打开浏览器的控制开，切换到“Network”选项条。



* 1. 知乎热榜部分截图
* 如图1-2为chrome浏览器控制台的Network界面，可以看到，当点击图1-2中的热点新闻时，页面跳转到[www.zhihu.com/question/397290358](http://www.zhihu.com/question/397290358)地址。在该独立热点详情页面下，在Network选项下勾选XHR选项，此时，可以看到浏览器与服务器之间的XMLHttpRequest（XHR）请求。网页的第一个请求question\_up?question\_token=397290358携带该页面热点问题的标识字段397290358向后台发出请求。下一步我们点击这一请求查看具体详情。

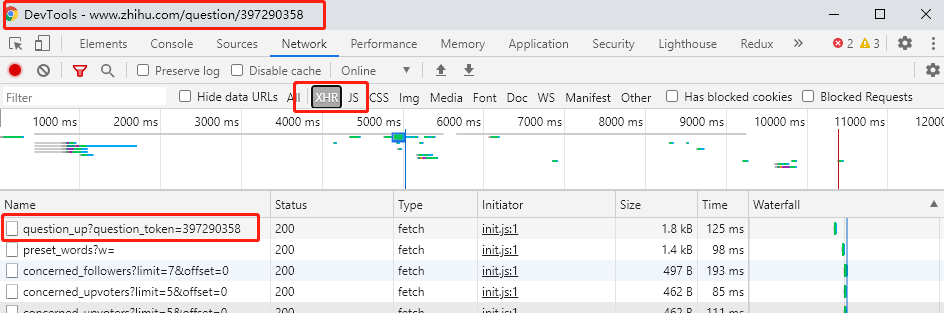


图1-2 chrome浏览器控制台的Network界面

* 图1-3为浏览器在访问question\_up?question\_token=397290358链接时的头部详情，浏览器的头部信息包括通用头部信息（general header）、请求头部信息（request header）和响应头部信息（response header），这些信息用以描述在浏览器发出网络请求到接受响应过程中的相关细节。从通用首部字段中可以看到，question\_up?question\_token=397290358网络请求属于GET请求类型，图1-4为此次请求头部信息的详情。请求头部字段包括本次请求的后台路径:path，网站的cookies，页面路由referer以及浏览器的信息user-agent等信息。

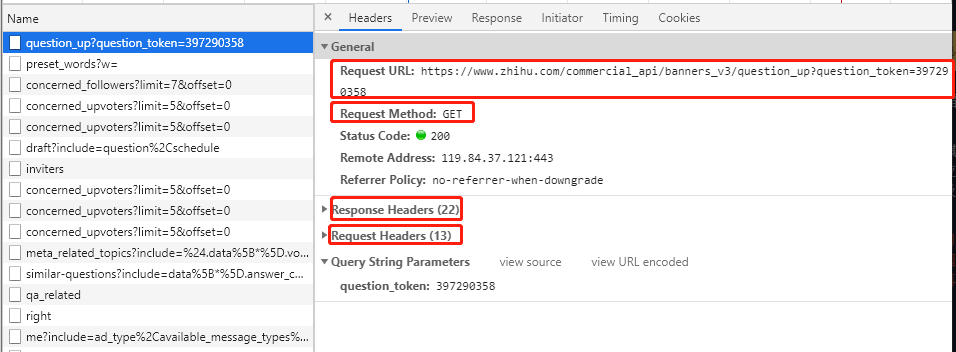


图1-3请求详情

图1-4请求首部字段详情

* 对于本次请求的响应，服务器会返回一定格式的响应，同样在Network的Response选项下，可以查看服务器的返回结果。分析该响应的结果得知，该请求为话题右侧的广告信息，也就是说再跳转到热点话题的详情页面时，会首先从后台的广告接口加载响应的广告。

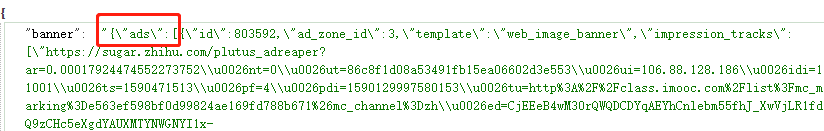


图1-5服务器响应的部分内容

## POST请求：

为了调研浏览器的POST请求方式，我随机选择一个知乎问题进行了回答，图1-6为回答问题的详情。





1-6知乎前端提交问题回答页面

* 同样，打开浏览器的Network选项，在点击提交回答按钮之后，页面通过浏览器发出一系列的请求，图1-7为该次提交问题回答的主要提交内容，网页通过浏览器以POST的请求方式向图中蓝色背景指示的URL提交网络请求，该次请求的主要内容如字段Request Payload字段所示，请求不仅向浏览器上传了本次回答问题的主要内容，还提交了本次提交内容的相关描述。

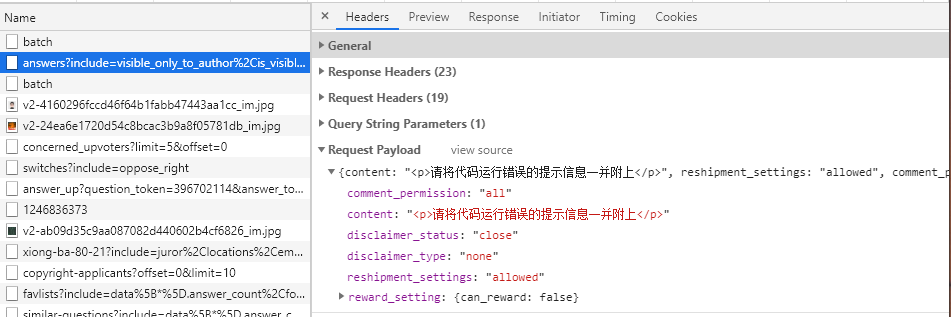


图1-7 回答知乎问题所产生的一个POST请求

* 服务器针对图1-7的POST请求产生响应的应答，如图1-8所示为前端根据服务其响应内容所展示的页面，图1-9为服务器响应的详情。

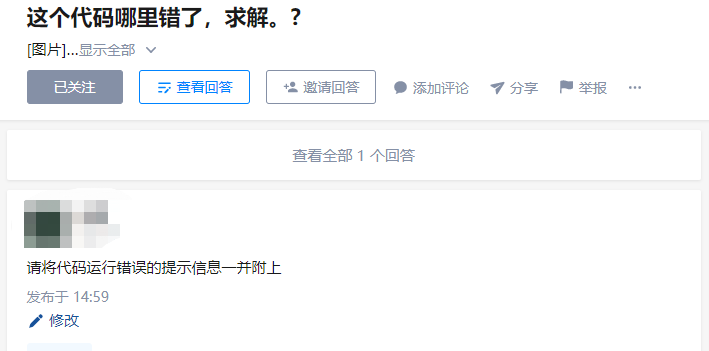


图1-8前端根据服务器响应生成的页面



图1-9服务器后台返回的数据详情

* 用户在前端填写的数据你并不是直接显示在对应的页面上，实际上，用户填写的数据需要经过服务器后台的验证之后，根据服务器的返回结果再写入到前端响应的页面中，上面所阐述的过程正验证了现象。

## jQuery事件：

为了完成这一任务，我基于豆瓣的API接口数据开发了一个建议网页，该网页页面如图1-10所示。

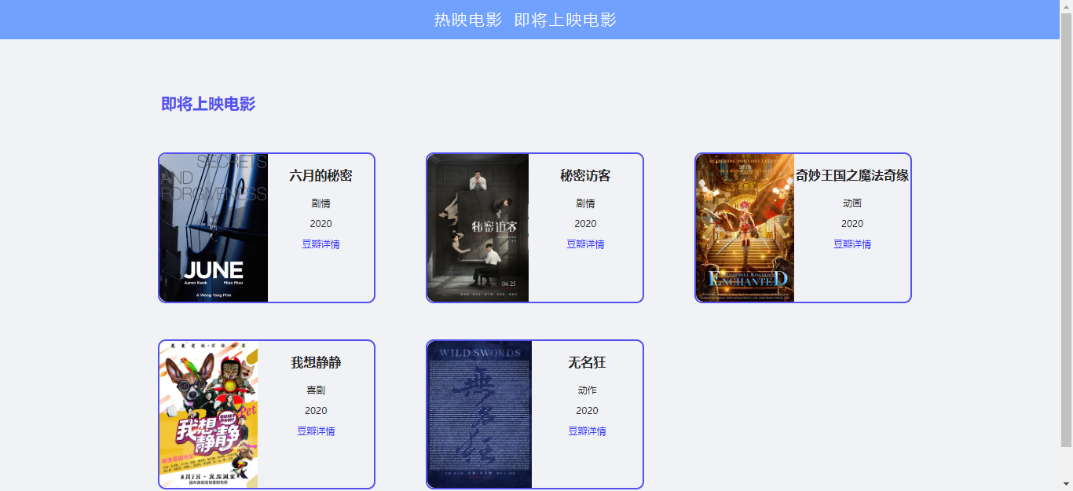


图1-10使用豆瓣API和jQuery开发的页面

* 在页面中有“热映电影”和“即将上映电影”两个主要界面，页面默认显示的是“热映电影界面”，可以通过鼠标点击事件触发两个页面之间进行转换。图1-11为页面的主要逻辑，当页面的HTML元素加载完成之后，图中的代码被调用，首先调用的是getHotMovies()函数，该函数请求豆瓣的热映电影API，获取相应电影的信息，同时更新页面的标题，指示显示的内容。接着，代码监听页面的点击事件，当鼠标点击“热映电影”时，程序会先更改指示标题为“热映电影”，将页面原先显示的内容清除的同时，加载热映电影的信息。

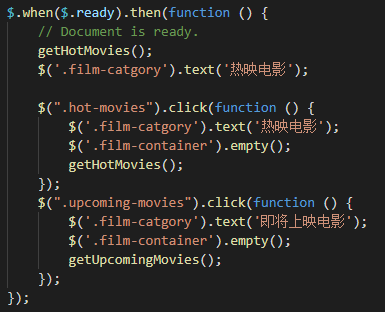


图1-11点击事件的主要逻辑

* 下图为点击“即将上映电影”前后网页页面元素的变化，左图为页面默认状态下加载的“热映电影”内容，右图为鼠标点击“即将上映电影”之后页面渲染出的HTML元素内容。



图1-12点击事件触发前后页面HTML元素状态

## 插件：

* 为了完成这部分的内容，我将上面开发的页面封装成浏览器插件发布，发布的平台是chrome浏览器。核心内容是在项目开发的目录下配置插件设置文件manifest.json。该文件是插件开发过程中必不可少的内容，用以描述插件的信息以及配置插件的相关信息。

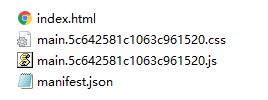


图1-13插件文件目录

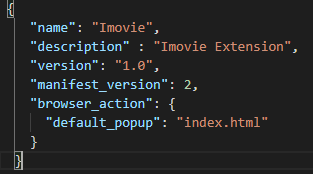


图1-14manifest.json文件的基本配置

* 在chrome浏览器的扩展程序界面，在“加载已解压的扩展程序”选项下，选择图1-13中的根文件。如图1-15所示。



图1-15添加插件的界面

* 成功添加插件之后，会在浏览器右上角生成一个图标，如图1-16所示，点击该图标可以打开插件，并进行相关使用。

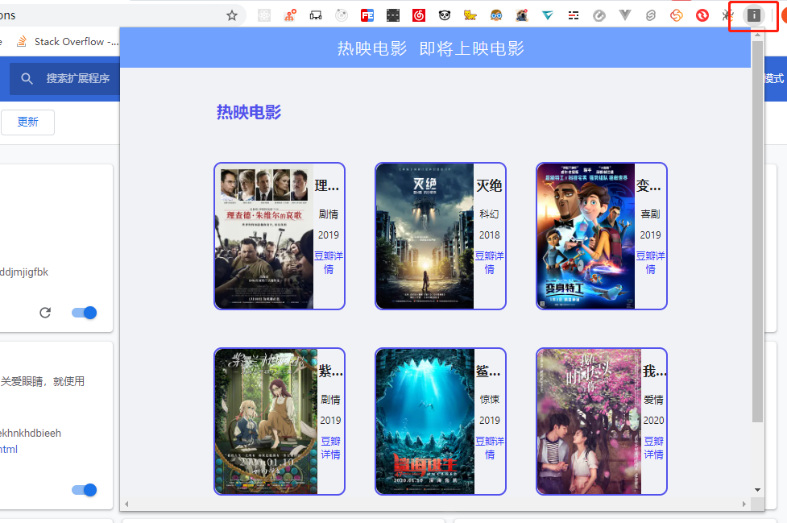


图1-17插件界面

* 插件功能

插件一共提供了两个功能：（1）查看热映电影（2）查看即将上映电影。

1. 查看热映电影功能：

这部分功能依赖豆瓣提供的API，<https://douban.uieee.com/v2/movie/in_theaters>，主要代码如下所示，函数getHotMovies()实现了该部分的主要逻辑。通过jQuery的ajax()方法获取接口数据，并按照一定的格式进行展示。

function getHotMovies() {

var request = $.ajax({

url: "https://douban.uieee.com/v2/movie/in\_theaters",

method: "get",

})

request.done(function (data) {

$.each(data.subjects, function (indx, item) {

$('.film-container').append('\

<div>\

<img src=' + item.images.small

.split('.').slice(0, -1).join('.') + '.webp' +

' alt="movie image">\

<p class="title">' + item.title +

'</p>\

<p class="genres">' + item.genres[0] +

'</p>\

<p class="publish">' + item.year +

'</p>\

<p><a href="' + item.alt +

'" target="\_blank">豆瓣详情</a></p>\

</div>')

})

})

request.fail(function (jqXHR, textStatus) {

alert("Request failed: " + textStatus);

})

1. 查看即将上映电影功能：

该部分内容与上一部分内容基本一致，不同的是请求的接口发生的变化，该功能下的接口为https://douban.uieee.com/v2/movie/coming\_soon

function getUpcomingMovies() {

var request = $.ajax({

url: "https://douban.uieee.com/v2/movie/coming\_soon",

method: "get",

})

request.done(function (data) {

$.each(data.subjects, function (indx, item) {

console.log(item);

$('.film-container').append('\

<div>\

<img src=' + item.images.small

.split('.').slice(0, -1).join('.') + '.webp' +

' alt="movie image">\

<p class="title">' + item.title +

'</p>\

<p class="genres">' + item.genres[0] +

'</p>\

<p class="publish">' + item.year +

'</p>\

<p><a href="' + item.alt +

'" target="\_blank">豆瓣详情</a></p>\

</div>')

})

})

request.fail(function (jqXHR, textStatus) {

alert("Request failed: " + textStatus);

})

}