# 

# 小A创客论坛项目开发文档

2021级信息工程学院电子信息类13班黄一峰

[小A创客论坛项目开发文档 1](#_Toc102838711)

[一、项目简介 3](#_Toc102838713)

[二、设计思路 4](#_Toc102838714)

[三、具体功能介绍 5](#_Toc102838715)

[四、项目总结 21](#_Toc102838716)

一、项目简介

1. **考核简介**

二轮考核约12天，内容可按接口分为四个部分，分别是用户模块、动态模块、相册模块和评论模块。最终成果是完成一个可以进行简单交互的论坛式网页。

1. **验收要求**

a.在考核过程中，完成项目基本功能的实现，以及在有能力的情况下完成加分功能

b.考核过程中的学习笔记

c.最终答辩的情况

1. **各模块的功能要求**

a.用户模块：

- 登录/注册，且进行正则校验

- 修改密码

- 记住账号密码

- 上传/显示个人头像

- 展示/修改个人信息

- 展示个人动态

b.动态模块：

- 发布/删除个人动态

- 查看所有用户简略动态信息（展示头像，用户名，时间，简略信息）

- 查看所有用户详细动态信息

- 搜索指定动态

- 点赞/取消点赞

c.相册模块：

- 新建/删除相册

- 新建/删除照片

- 图片懒加载

d.评论模块：

- 新建/删除评论

- 新建/删除回复

- 查看所有评论and回复

1. **完成情况**

已完成的功能：

用户模块的基本功能、动态模块的基本功能、相册的基本功能（删除相片存在bug，图片懒加载未完成）、评论模块的新建和删除评论（新建回复和删除回复未完成、查看所有回复未完成，但可以筛选出评论，把回复隐藏）

已完成的加分功能：

七天免登陆、个人相册的大部分功能、动态分页展示、展示热门动态排行榜（可查看详细内容）。

未完成的加分功能（但有思路）：

动态更新后立刻提示、黑夜/白天模式、一键切换主题、动态评论。

个人完成情况：

由于我的学习进度原因，在二轮考核期间是边学边做，最终完成的可能大概有60%-70%吧0.0，我也不太确定。就完成了用户模块、动态模块、相册模块的基本功能。

1. **用到的知识**

在考核过程中，大范围来说，使用到的知识有HTML、CSS、JavaScript、AJAX 、es6

以及服务器存储等只是，具体来说，有用到JS事件委托，JS对于节点的操作，ES6中对字符串的插入、API fileReader的应用、AJAX的封装、cookie的存储在免登陆等功能的应用、localStorage在记住账号密码上的应用以及JS BOM中的计时器等知识、还有JS中的各种事件的运用等等。

1. **开发中遇到的困难**

开发中的困难还是有的，一边摸索一边前进吧，我的信念就是可以比人家慢，但一定要努力，所以做的过程中，遇到过传参不成功的，通过一步步调试去找到哪个地方传参出错；也试过传参后乱码的，后来去查了转码方法，又恢复正常的步伐。在评论回复那个模块研究了很久，但是可能还是对这方面知识比较欠缺，没研究透，也有可能是学习的过程中过于着急了，没有很好的把这部分只是巩固下来。同样，相册那里也是没有做得很好，比如自己想把封面换成相册里面第一张图片，想把样式写的更好等等，还有就是做cookie的时候没有设置好，导致后面的七天免登录没有做好，只是做到了判断cookie中有无密码来实现免登录，显然是不行的，没有考虑到用户的体验。没有优化好，是我在做项目中觉得很遗憾的一个地方。

## 二、设计思路

1. **分析页面需求，搭建页面结构**

拿到考核要求后，我首先对页面的需求进行了排序，搭建了页面的基本结构，我的规划是先把用户模块做好，其次是相册模块，然后是动态模块，最后是评论模块。有了基本的思路后，我决定按功能写多个页面，通过跳转页面实现功能间跳转，这样更能实现功能的模块化，当然，我也分离了CSS和JS的文件，把功能模块化。（其实在做这个决定之前，有想过通过display实现功能之间的切换，但后面觉得不妥，就选择了写多个页面）。留意到考核要求最终页面以index.html命名，我选了登录页面作为index.html，因为这是整个页面的开端，从这里开始可以遍历后面的所有功能。到此，我在项目中的页面结构就清晰了，按照功能去分离，同时实现HTML、CSS、JS三部分的编写，一步步去搭建小A创客论坛页面。

2. **设计中的思考**

在设计页面的制作流程时，被CSS样式部分困扰了许久，如何去美化页面，提高页面的观赏性是我思考很久的一个问题。在想到小A创新创客团队的主题色后，我选择使用了蓝底，以及在一些地方添加上小A的logo来美化页面，这取得了令我满意的效果，并且为了丰富页面色彩，我用了不同的颜色来渲染页面，比如相册页面我做成了一个镶嵌式的页面，就是想把页面的相册按照现实中的相册来仿真，以此来提升一点美感。但也不全是不同的颜色，例如按钮我基本上全是使用橙色来渲染，只有评论那里使用了浅蓝色（仿b站）。CSS部分差不多就这么多，最后页面其实还可以提升，通过hover、animation等属性还可以做得更加完美，但限于时间，没有完善好。。。

其次就是JS部分了，JS部分涉及到页面的交互，影响是很大的。在设计之初，我还是受限于我的学习进度，很多功能还是没想好怎么去实现，但后面边学边做，发现自己能做的越来越多，也就越来越顺利，但一开始思考的就是相册懒加载，和动态评论，多级回复这几点怎么做，因为就其他功能而言，这几个功能是在学习过程和制作demo过程中没遇到过的，也是很想去学习，很想去了解的知识，虽然后面还是没做出来，但是对于这几个功能的实现都有了一定的思路（答辩可讲）。

3. **对于JS的一些介绍**

事实上，我JS代码比较长，因为我没有封装对于动态生成的一些函数（本来是想完成所有功能后封装），我只封装了Cookie函数和AJAX函数，但是我将各功能的JS进行了分离，下面就JS的内容进行介绍。

首先，每个JS的命名都是对应相应的HTML页面的，除了共用的一些JS，例如ajaxFunction、cookieFunction、allElements、forum、search这些就属于共用的JS文件，其中allElements代表是所有的已获取的JS元素、forum代表是各功能的点击跳转、search代表是各页面的搜索跳转（夹杂传参）。之后的JS文件基本上就是与HTML页面命名一一对应，分别实现各部分的功能。

## 三、具体功能介绍

1. **登录**

首先对于登录部分的HTML部分，我采用了两个div 一个是装了小Alogo和欢迎语句的div，一个是装了登录功能的div（其中tab栏为li标签所做）。CSS部分背景色我采用了贝壳色，主要目的是为了让盒子内容显示更清晰，然后也是有一种柔和感。（此处由于邮箱登录和用户名登录基本一致，所以只展示用户名登录页）



JS部分呢，首先从tab栏说起，因为考核要求有用户名登录和邮箱登录，所以我做了个切换tab的功能，代码如下：



这部分主要的思路就是排他思想和display的隐藏与显示，既然要实现切换，就在点击另一个小li时，前面的样式要被清除，后面的要承接样式，由于此处比较简单，所以我直接就写个循环来实现（通过清除和改变类名来实现）。然后实现后，就是下面div内容的变化，这里我是用了display的none和block来做到内容切换。

然后就是登录框，首先是两个input text输入框，输入账号/邮箱和密码，输入密码旁边有一个眼睛是用来显示和隐藏密码的，然后下面就是修改密码的a标签和七天免登录的选择框，下面就是登录按钮与注册。下面来说一下JS的实现。首先显示与隐藏密码是用了一个函数，代码如下：



我的思路是设立一个函数，通过输入参数来判断是否隐藏，下面就是我传入的一些参数，其实就是四个按钮，来控制显示与隐藏密码和图标的变化。

然后说说登录按钮，登录按钮我给了一个点击事件，当点击时，就会获取输入栏的数据，然后传入数据到接口，再然后跳转到登录页面，实现登录功能，代码如下：



思路是，传到eolink后，拿到数据，并且设立cookie，使得实现七天免登录和记住账号密码。同时为了提高体验，每次登录后都会清除账号密码，使得用户如果要换号不需要清除之前的账号密码。

七天免登录就是利用cookie值和复选框的checked值进行判断，如果两者均符合条件，则下次进入直接跳转到主页，无需继续登录。

关于登录的正则，我没有以正则表达式的方式去写，而是通过输入后alert的提示来实现登录验证。

2. **注册**

首先，从登录页面进入到注册页面，原理同样是通过display来实现，给登录页面中的

注册a标签（如最上图所示）一个点击事件，使其点击后实现改变display的属性，从而实现内容的切换，代码如下：



进入到注册页面后，首先注册的结构和样式如下：



思路很简单，就是跟一般的注册页差不多，在输入框下进行了一些提示，提高用户体验。

JS部分，首先隐藏和显示密码是跟登录同样原理。其次，在注册部分我添加了正则表达式，通过用户输入的内容验证判断，提醒用户按照格式输入从而完成注册。思路就是给一个reg变量利用正则语法限制一定的范围，从而来对输入的内容进行test判断，再通过onblur和onfocus事件对提示信息进行实时的显示。由于思路比较清晰，代码基本都是重复的，就不展示代码了，展示一下效果：

：

接下来的是一个注册按钮，同样是给点击事件把注册输入框的数据传到eolink给服务器，并且根据报文响应状态码返回注册成功或失败的提示。思路同注册，同样有注册后清除输入框的数据，方便下一次注册，且注册成功后会弹回登录页面。

最后是一个返回主页的选择，如果用户注册到一半发现自己已经注册过了（或者不想注册了），就可以同个这里返回登录页，继续登录。

3. **修改密码（修改密码）**

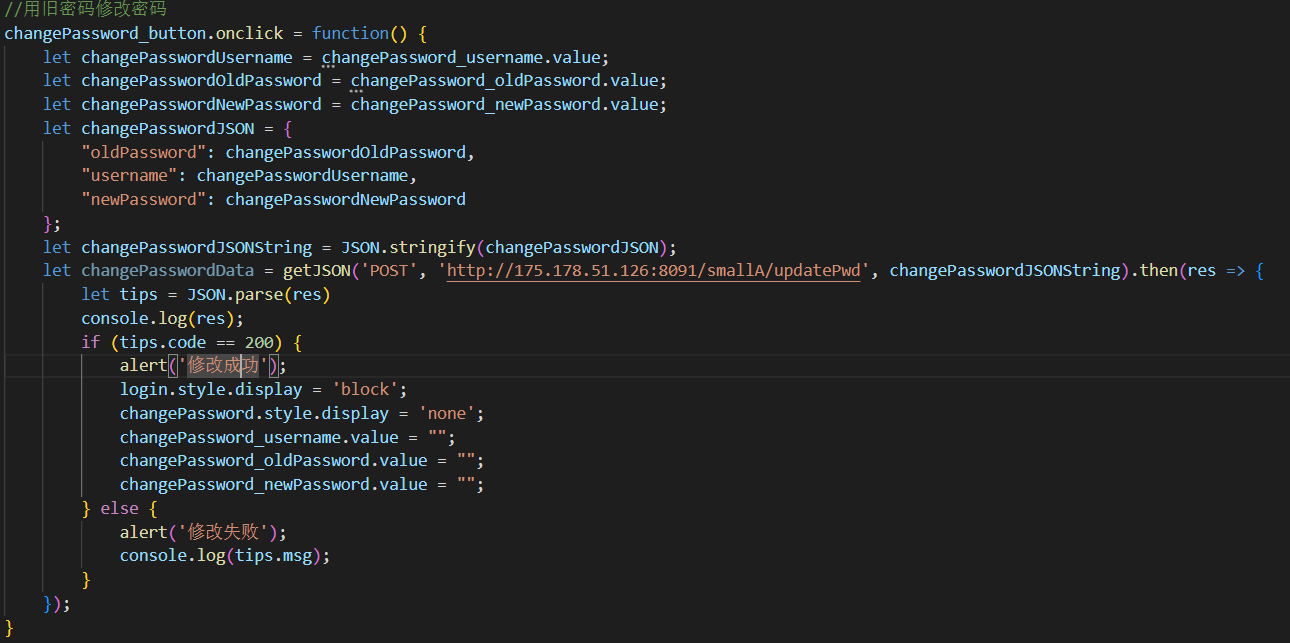
HTML和CSS效果如下：



同样是用了tab栏切换（原理同登录），然后下面的就是对数据的输入最后提交。

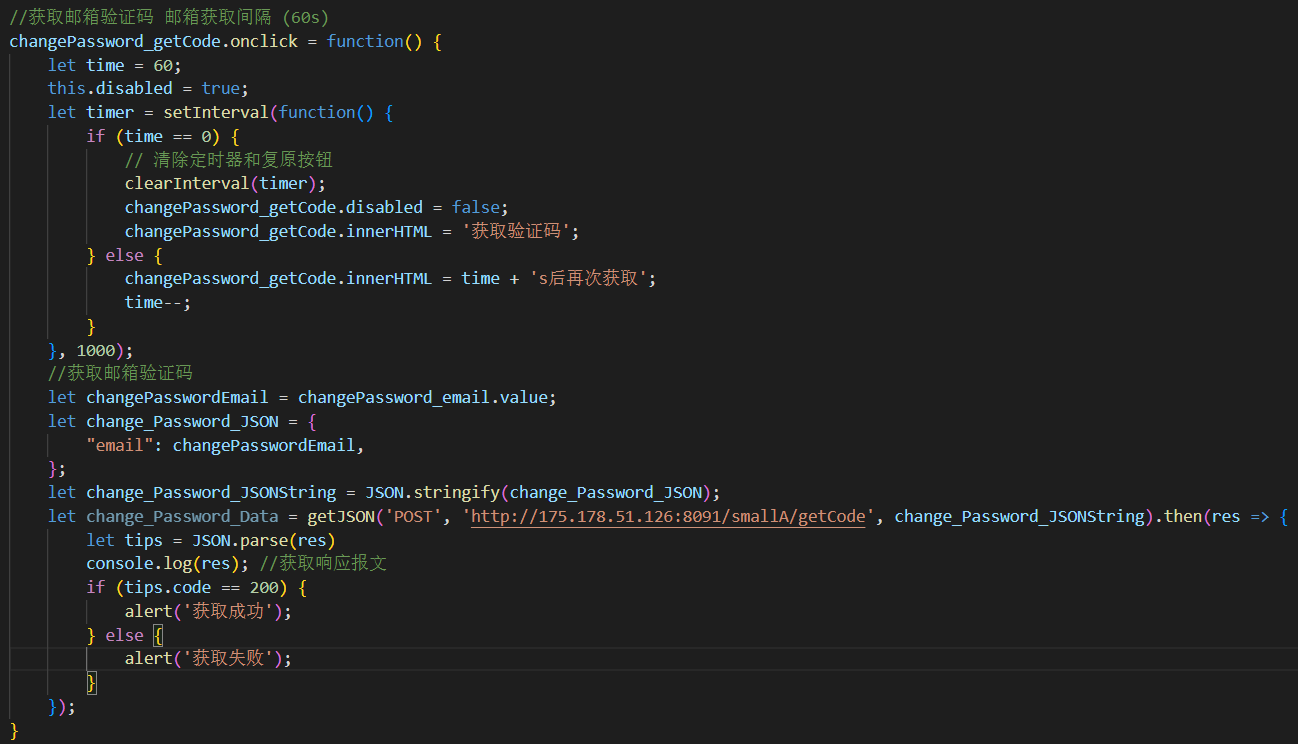
JS部分，首先仍是使用了display属性来对修改密码的弹出和隐藏登录页，然后相同于注册，修改密码页也是有回到账号登录的选择，设计思路也是同注册雷同，在用户想到密码时可以返回，没必要盯死在修改密码页。

然后在旧密码修改页，原理十分简单，就是通过确认修改按钮的点击事件，来获取输入框的内容，并传到eolink接口，实现密码的修改，代码如下：



思路：获取输入的数据，打包成JSON发送到eolink，根据返回报文的响应状态码来提示不同的返回信息，成功后会弹回登录框并且清除在输入框已输入的数据，失败则返回修改失败。

在邮箱验证码修改页，则是有一些不同，涉及到邮箱验证码获取按钮的样式以及获取验证码的时间间隔，代码如下：



此处的思路是，点击获取验证码按钮后，设立60s时间，并让获取验证码按钮不可选择（disabled），然后就设立计时器，通过判断时间是否为0，来对定时器决定是否删除，而且更改获取验证码的按钮样式。然后下面就是获取邮箱验证码，通过email传到eolink，然后返回验证码到邮箱，同样通过报文响应状态码来提示获取成功或失败。

后面就是修改数据的传入，跟旧密码思路相同，就不多论述了。

4. **个人菜单（由个人信息、个人相册、和退出登录组成）**

HTML、CSS效果如下：



a . **个人信息**

HTML、CSS效果如下：



这部分样式没做到很好，主要是因为一直忙于做后面的功能，把它遗忘了。。。这也给我个提醒，就是当天事当天毕，因为你永远不知道后面的事有多少。所以这部分就是简单上了个色，然后就是分为左右两部分，左边是上传头像的部分（可预览），右边就是展示个人信息部分和修改个人信息，删除账号部分。

下面展开说说他们的JS实现。首先是上传头像，上传头像我用的是input file按钮，此处令input file 隐藏，再写一个button，通过JS令点击button来触发input file的点击事件实现替代（更改input file的样式）；然后头像预览是通过监听其改变事件，获取其改变的值，并且通过window.URL.createObjectURL（）来获取他的url，再通过拼接的方法来获得带有src的img标签（str），再通过innerHTML来加到盒子中，实现头像预览。最后是创建FormData对象，将键值对加入到对象中，再进行上传（此处没有利用到ajax函数，是在下面重新写了一次微改的AJAX，所以就没封装FormData数据的函数）。具体代码如下：



接着是右边部分，是个人信息展示和修改个人信息，删除账号按钮。首先，个人信息展示是利用了立即执行函数和登录接口，通过登录接口获取登录账号的数据，然后把数据用innerHTML拼接在个人信息展示盒中，实现展示个人信息的效果。

修改个人信息页面和删除账号页面是通过给修改个人信息和删除账号按钮点击事件，更改display值实现跳转的，代码如下：



接下来是修改个人信息页面，页面如下：



此处思路也比较简单，就是获取填入的数据，然后通过确认修改点击按钮事件来将数据提交到eolink，代码基本跟之前相同，就是换了接口，同时，清除数据和返回功能也是支持的，所以在这不重复论述了。

最后是删除账号功能，页面如下：



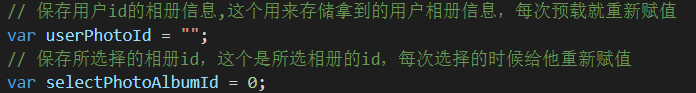
删除部分也相同地跟上面同样的思路了，只是换了接口

b． **个人相册**

进入相册首页是利用window.location来实现的跳转，包括上面的个人信息页面也是，因为是属于一个功能，所以使用页面跳转的方式来写。页面如下：

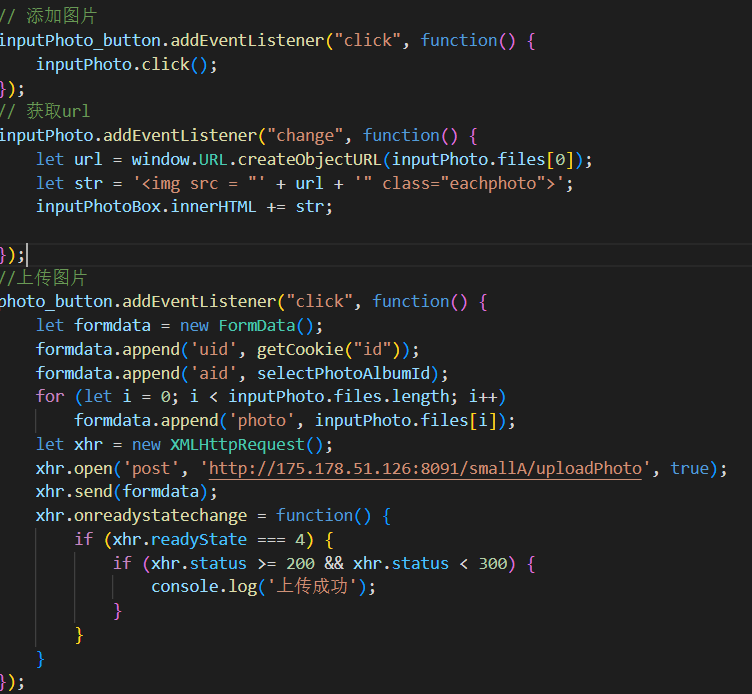


设计思路上面有说，在这就不多说了，直接说说JS部分思路，首先定义了全局变量拿到相册的信息和相册Id，方便下面的取值，减少不必要的操作。



然后呢对相册进行预载，通过cookie里面存的登录获取时id，来查询id中的所有相册，实现上面相册图的效果（通过for循环innerHTML动态插入新相册），这里做的有缺陷，没有把相片封面设成第一张图片还存在一个bug，点击删除相册会先进入到相册内部。紧接着新建相册，新建相册的内容也是跟前面的内容类似，都是获取输入框的内容，然后post上eolink，再通过预载函数实时刷新相册数。

打开相册，这是小功能，所以是改变display实现打开。进入相册后，会实时更新标题，通过获取相册的数据来更改相册名，并且可以添加图片（需要刷新），添加图片是新建FormData对象，然后来上传，删除的话则是通过事件委托用JSON传数据到eolink删除，不过删除部分没做好，会报错。代码如下：

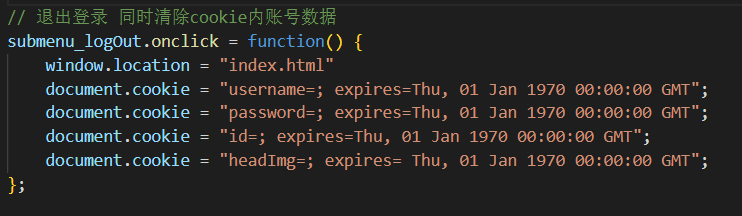




c． **退出登录**

退出登录思路：跳转回登录面，考虑到免登录的存在，把先前登录的cookie删除（删除即是改变过期时间，实现删除效果）

代码如下：



5. **动态（个人与所有人**）

首先主页页面效果如下：



JS部分，顶部导航栏都是可以实现跳转的，我将功能分开了，具体跳转代码（包括个人菜单的二级菜单显示）如下：

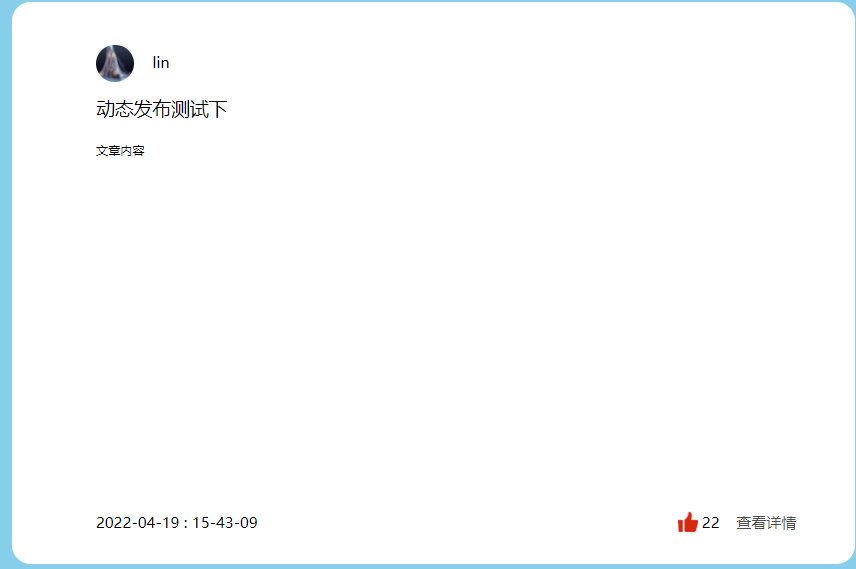


登录后的头像和用户名显示代码如下：

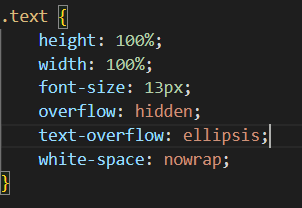


A．**动态生成简略动态**

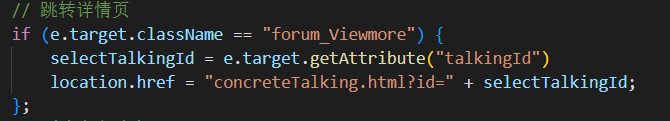
动态生成简略动态（代码有点长，就不展示了，具体可看vscode代码），主要讲讲思路，就是利用了appendChild（）方法，从最小装到最大，一级级装成一条简略动态的样式，然后通过获取后台数据，写for循环遍历数据，动态生成所有简略动态（已实现分页，分页后面说）[同时，还有第二种生成的方法，就是直接将html写好（保留大盒子）然后把大盒子下面的html复制到js中，利用大盒子.innerHTML = html部分，再在html部分中的相关要渲染的内容通过es6中的${}方法插入渲染，但是动态图片那里只能显示一张，这是个问题] 下面是单条动态效果：



补充对简略内容的处理：



下面是详情页，如果要点击详情页，右下角是有个查看详情提供查看功能的，其原理是跳转页面，为什么这里不是display呢？原因是，在里面需要添加评论的一些内容，如果display做会影响到原页面。需要注意的是，此次跳转是用了location.herf传参方法，把动态的id传到详情页，方便详情页的数据渲染。代码如下：



进入详情页后，首先会分解首页传来的链接，获得动态id（然后通过全局变量获取这个id），通过id查询接口，实现动态的单一显示，实现详情页的效果，同时，给点击详情按钮进入时，会重载页面，获取数据对页面进行渲染，同时也会查询评论（查询评论已经将prior值不是0的去掉，因为通过后台发现删除评论后仍存在回复的情况），我写固定的部分只有评论动态的评论框，其他都是动态生成的。由于宽高有点大，就不在这展示了，可以在答辩展示，代码也是TAT。下面的评论我是给了发表评论一个点击事件，给他设置了一个自定义属性来获取用户的id，点击后获取id，加上全局变量获取到的动态id，再加上输入内容进行评论（评论不能为空，为空就会显示不到评论）。然后对于评论的删除功能，就是给删除功能获取到评论的id，通过自定义属性获取，然后post到接口删除。然后查看所有回复和回复功能还没做好，所以评论只是做了一部分。。（补充：评论部分也是通过事件委托来实现）

B．**点赞动态**

点赞动态这里其实不难，就获取到每一条数据，渲染在动态盒子上，然后数据里面有是否点赞isLike和点赞数likeNum，设置个tag值等于isLike判断是否点赞，初始值为true，如果已经点赞了就tag变为false，没点赞就是true，然后把数据封装成JSONpost到eolink就行，然后这里点赞图标的显示，我是放在了动态生成里面，把图片的src改成以下形式likeImg.src = "../image/" + tips.data.list[i].isLike + "like.png";这样设的好处是可以直接通过isLike的值来对图标进行更改，而不需要额外写更改src，下面展示代码（同样有用事件委托）：



C. **删除动态**

删除动态一开始我是做全部都可以删除，只要你有他的动态ID，但后面觉得不妥，应该只能删个人的动态，所以我给每个动态判断他的用户id是否跟登录获取的id一致，如果一致就在右上角出现叉（给叉一个自定义属性获取动态id），点击叉就可删除动态。具体代码如下：

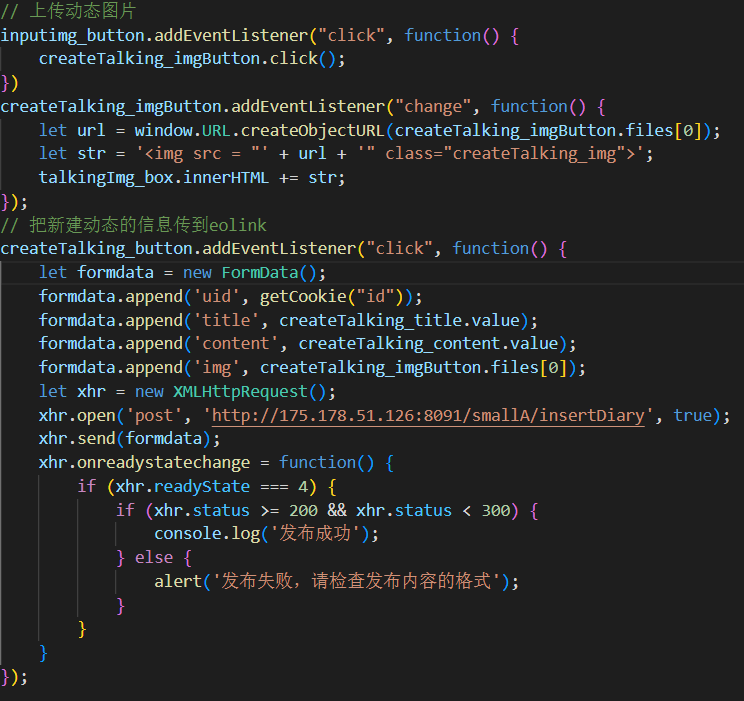


D. **发布动态**

其实跟之前的登录注册差不多，就是涉及到图片上传，所以要用到FormData对象，

思路就是创建FormData对象，一个一个append进去，然后postFormData对象去eolink

再重载页面实现刷新。代码如下：

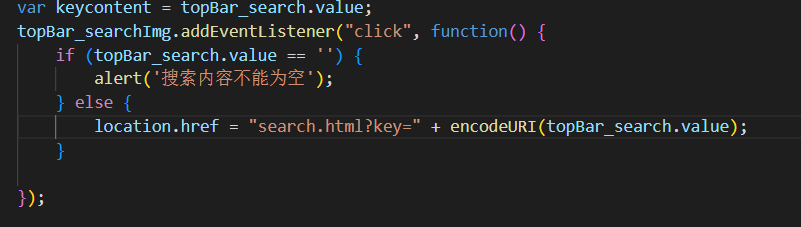


E. **个人所有动态**

个人所有动态页面通过点击头像进入，主要思路是给了头像一个点击事件和自定义属性，能够获取用户的id（也通过传参到另一页面，有编码操作），然后点击事件触发，获得用户id，和登录的id封装成JSON数据传到eolink，并跳转到个人所有动态页，这部分也用了事件委托，具体代码太长，在这里就不展示了0.0

F. **搜索动态**

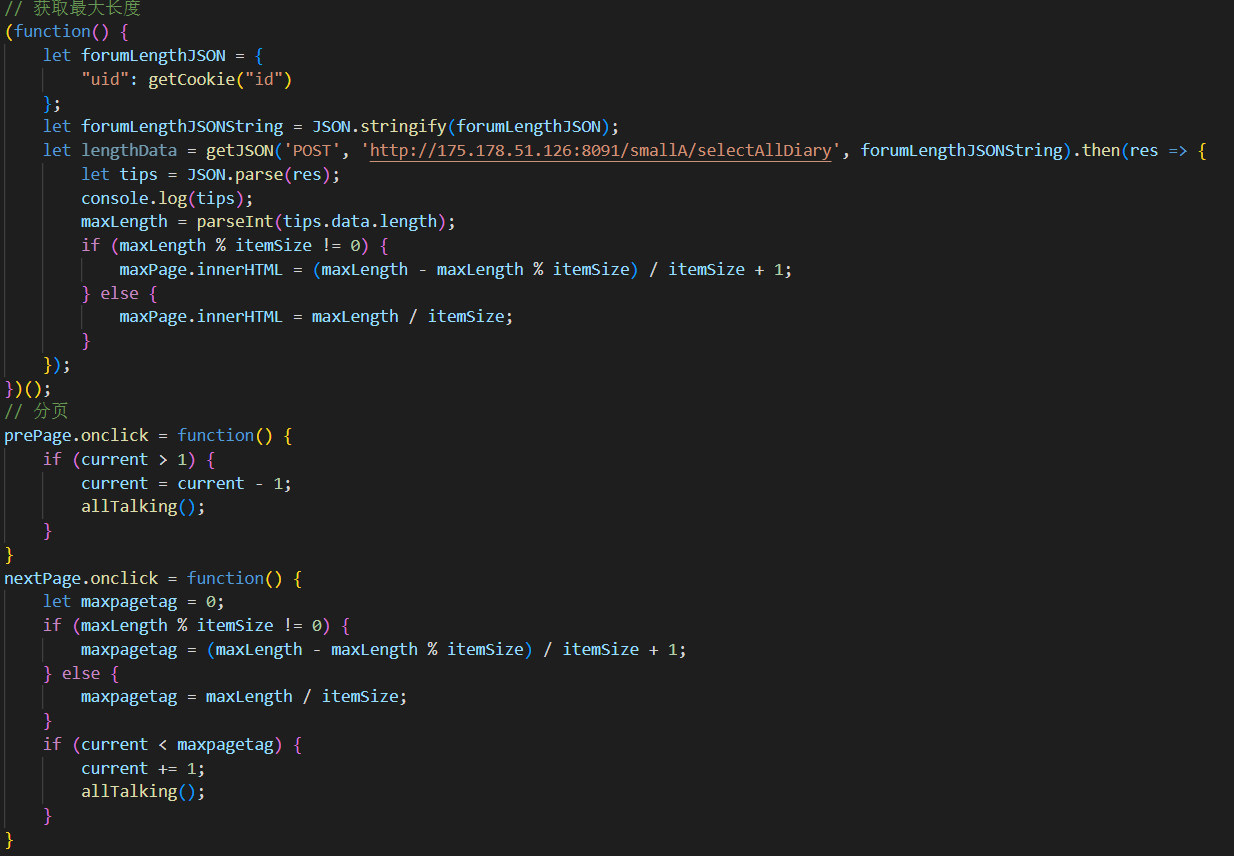
搜索动态的话，每个功能页面都引用了search.js文件，其中的作用就是利用location.herf将搜索内容传参到搜索页面，具体代码如下：



其中运用了encodeURI方法对输入内容进行编码，这一步主要是中文输入进行搜索会出现乱码，进行编码后就可以正常搜索，于此相对的decodeURI（）可以进行解码。同时也定义了全局变量来获得参数，方便传参。然后在搜索页面，便先利用未分页的接口获取最大长度，再利用分页接口获得所有数据进行分页，具体代码太长，就不在这展示了，大概思路跟展示所有动态相同。

1. 分页功能

以上的功能能用分页都已经用分页来做了的，对于分页有三种思路，一种是用循环来做，一种是数组的splice()和slice()方法，最后就是用接口，我选择用了接口，因为我时间不够0.0，太赶了，主要思路就是设置三个变量，分别为当前页数，最大页数，一页最大展示数据。然后通过未分页的查询所有动态的接口来获得所有数据的最大长度，也就是上面的变量之一，再根据每页的最大展示数据进行边界值的判断，从而获得current和size，再进行封装 上传到接口，就完成了分页，同时，在页面下设计当前页最大页和上一页下一页，从而能够实现切换，下图为具体代码，其他分页也雷同：



7. 封装AJAX和Cookie函数

1）AJAX函数

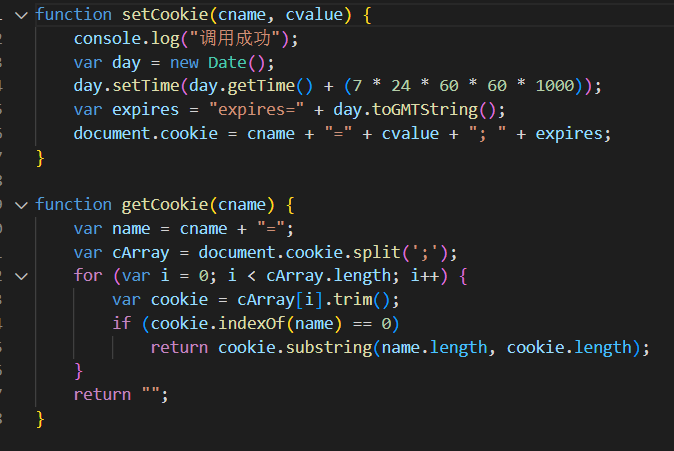
直接上图吧，注释应该挺清楚了



type为类型，url为接口地址，json是要传的JSON数据

2）Cookie函数

同样是上图



主要是设置一个Date对象来获取当前时间，再加上七天的毫秒数，来做到七天期限，然后expires就是ddl，后面的toGMTString（）方法可根据格林威治时间 (GMT) 把 Date 对象转换为字符串,并返回结果。下面的spilt是对字符串中按每个；进行分隔，用于获得各部分信息。trim（）方法是清除头尾空白符。substring（）用于获取两个长度之间的字符。

## 四、项目总结

1. 亮点

CSS部分可能渲染得好一点

全局变量除了用于获取id，数据和传参，没怎么滥用

在做搜索分页那里用了转码实现中文搜索（也有可能是我方法不太对，可能只有我的中文搜索不行）

没有融合所有代码在一起，将代码按照功能分开文件来写

通过事件委托，来解决动态问题和特定地操作动态生成中的某一元素

2. 缺点

JS部分很多代码都没封装（这些代码相同度高），使得代码很长，可读性不高（但有做注释）

对ES6部分学习不深，没怎么用到promise的地方

个人相册没做完

多级评论部分没做完

主题切换没做完

有些地方比如修改个人信息的正则忘做了

等等

3. 感想

这次考核在做的时候压力还是蛮大的，但是顶着压力去学去看代码这种学习感觉比平时空闲时间去学来的要好。整个考核给我的感觉就是一个流程，首先是怎么写好HTML和CSS部分，然后就考虑怎么去从后台获取数据，获取数据后怎么渲染，然后包括实现各种功能之间的跳转啊，一些事件的设立啊等等，最后就是补上一些该有的提示，考虑到用户的体验（这个自己走一轮自己的程序应该能感受到）。于此同时，我认识到前端确实是入门轻松，但纵深发展还是比较难的一门语言，尽管如此，这门语言很有意思，这次小项目就是一个例子，如果学的再深一点，我感觉我能做出更多有意思的东西。所以整体来说这次考核还是收获颇丰的，我会继续学习，继续进步！