# 我对searchui 项目的理解

这个项目是大体是一个node项目，又是webpack项目配合react前端框架，首先此文件夹由node管理，package.json文件是一些基于node项目配置信息,webpack.config.js是webpack的配置信息；

webpack.config.js指定了入口js文件为\frontend\src\main.js,main.js主要引用了SearchUIApp.js，这里配置读取了上层文件configuration.properties.js为配置信息在文件中是变量config;

其中方法：getMastheadNavTabs提供了搜索栏功能页配置，在这里可以通过修改配置可以改变searchui页面搜索栏内容展示；

新添加的export按钮，是通过修改C:\Users\Wangj01052\Desktop\searchui\frontend\src\pages\SearchUISearchPage.js,内部代码都是通过js模块化，很多变量可通过百度查询功能，还有部分变量是项目自己生成，

想要了解详情，可以查看其它js文件中的引用

# 1.给searchui 添加一个页面

参考：<https://answers.attivio.com/display/extranet55/Tutorial%3A+Adding+a+new+page+to+Search+UI>

## 1）下载 seachui开源项目<https://github.com/attivio/searchui>

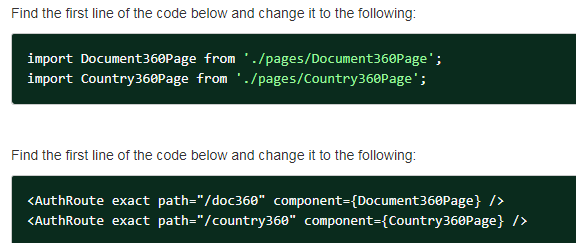
## 2）创建**Country360Page.js**

**在search-ui/search-ui-frontend/src/pages目录中创建Country360Page.js文件**



注意里面的空格，tab等空格符一定要服务规范，否则会打包失败，我之前就是没注意这个导致一直打包不成功，这里的js文件我已经做过处理空格符都没有问题

## 3）将这个新的Contry360Page添加到App中

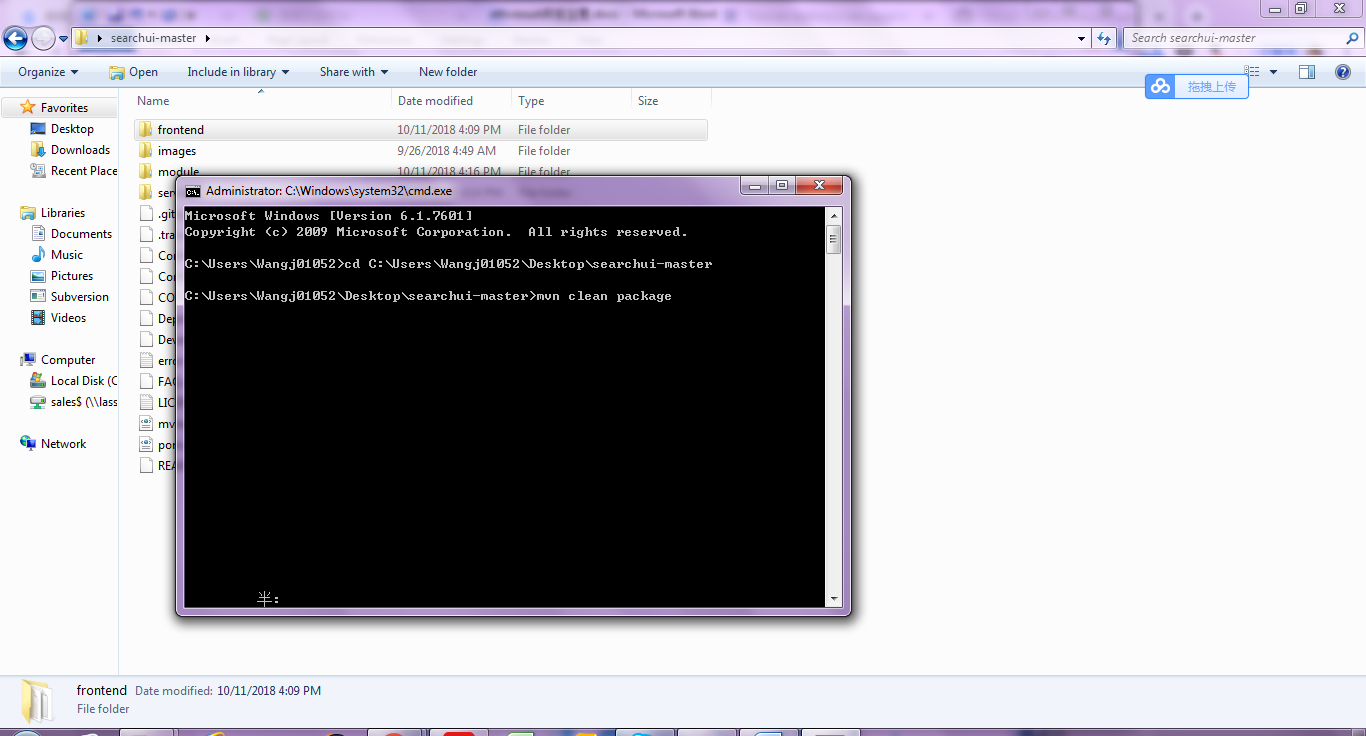
打开**search-ui/search-ui-frontend/src/SearchUIApp.js文件修改配置**

注意：这里粘贴时需要有空格

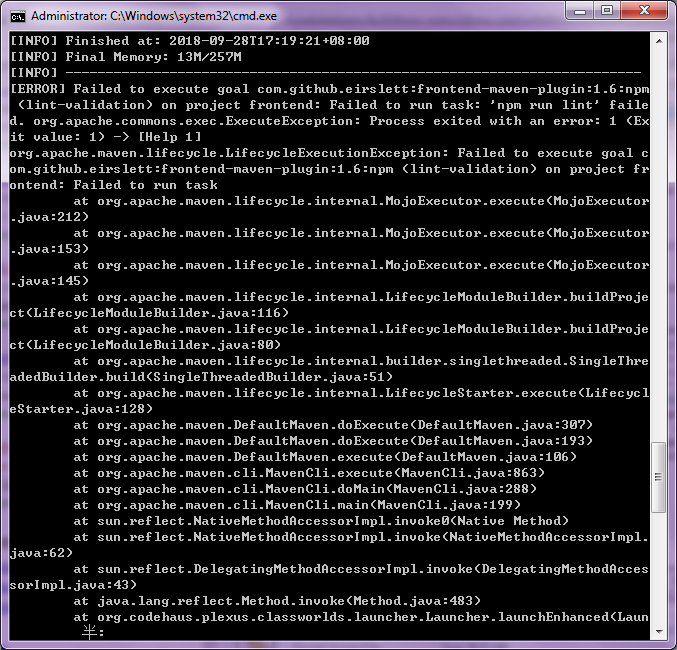


## 4）打包

在项目目录上执行打包命名：mvn clean package

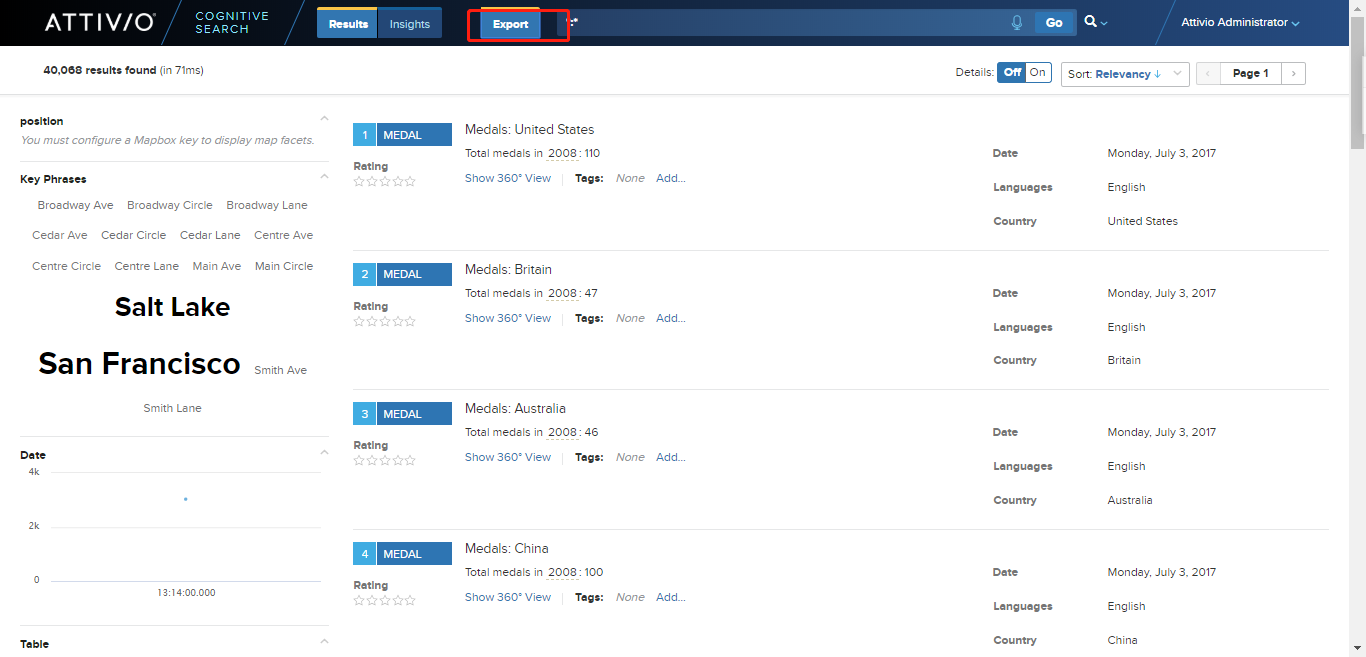


注意

打包时可能出现错误：

报这个错误的原因可能就是由于先添加的Contry360Page.js内容有问题，或者SearchUIApp.js里添加的代码有问题，我这里的问题是因为空格符导致的，这个项目是webpack项目，找到webpack项目根目录C:\Users\Wangj01052\Desktop\searchui-master\frontend，在这个根目录下执行cmd命令：npm run lint来检查项目文件代码是否有问题，如果有可根据提示调整代码！

# 2.在搜索结果页面添加export按钮如图



实现功能 点击export会获得搜索结果，之后会对搜索结果做二次处理

## 分析需求：

这个页面只需要修改\searchui-master\frontend\src\pages\SearchUISearchPage.js文件，

这个功能实际上就是调用http://127.0.0.1:17000/rest/searchApi/search接口从attivio服务端拿到搜索结果。

## 功能实现：

就是搜索栏一层，添加一个export （button）元素,之后添加上对应样式，我们这个项目已经引用了bootstrap框架，所以可以直接引用想要的样式，但点击export按钮时会触发点击事件，我们这里做一个函数，来监听这个事件即可，函数主要功能：调用<http://127.0.0.1:17000/rest/searchApi/search>接口获得搜索结果数据，并做二次处理



# 3.点击export按钮时，出现一个提示框，3s后自动关闭

## 分析需求：

可以分两种方式处理：直接在页面写alert，然后用定时任务，比较low；自己写一个模块实现提示框控件效果，可以操作；调用已有的模块，靠谱！

## 开始操作：

找到一个提示框模块rc-notification，调用也比较简单，具体可参考<https://www.jianshu.com/p/a84be2e98a8b；https://github.com/react-component/notification>；

在frontend文件夹，webpack根目录下执行npm install rc-notification ；安装完成在想要使用此模块页面引入import 'rc-notification/assets/index.less';import Notification from 'rc-notification';初始化notification 对象 let notification = null;Notification.newInstance({}, (n) => { notification = n; });使用方式特别简单，在需要有提示的地方：notification.notice({ content: '查询结果导出中，正在创建新表存入此结果', duration: 3, style: { background: '#d4b5b2' } });

打包结束！

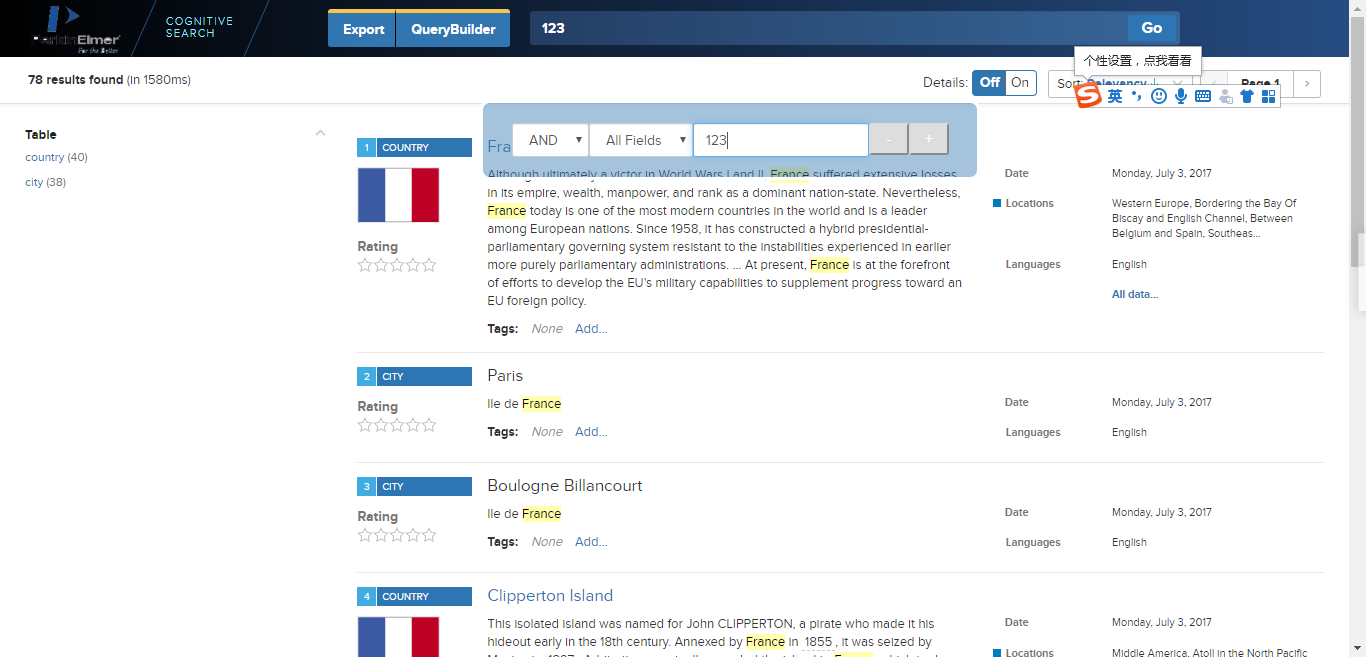
# 4.自定义查询增加QueryBuilder模块，如图

主要修改如下js



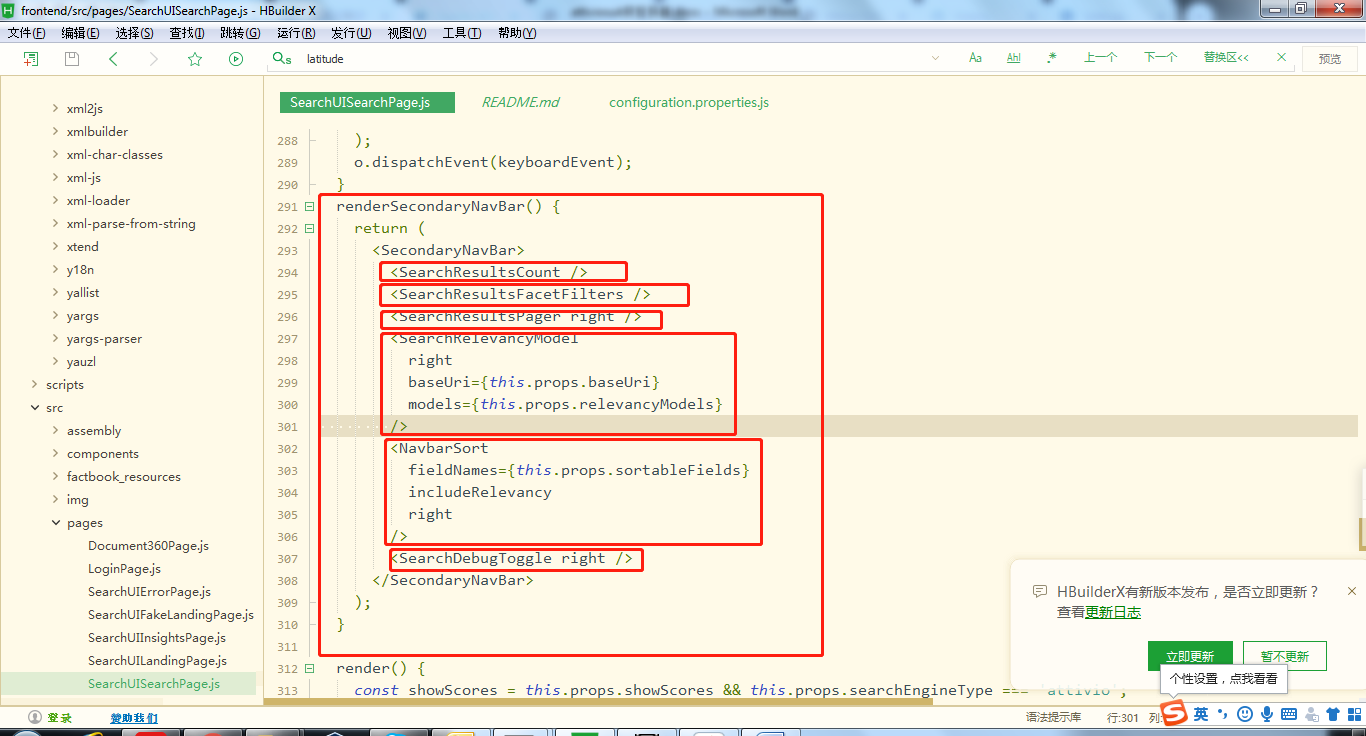
主要知识点与难点：对于这个功能主要难点在于其onchange事件上层控件不好补获，又由于框中下面输入框等控件是动态生成，对于未来元素控件react不好处理，解决方法时通过react的map通过state的动态改变而动态生成的，同时为了再点击删除控件时能删除指定控件，需要动态生成的每个控件还需要加上唯一的key属性，这样就能完美控制删除增加同时每个控件的onchange事件都能补获的到了，因为这里的动态生成的控件每次写入onchange事件时候react都会重新绘画页面（react有自己的特殊机制，不会改变state元素中没有变化的控件属性，只会改变指定特定的控件元素属性状态）；还有一个难点在于通过js脚本改变input中的值，并没有触发input元素的onchange事件，导致每次改变完input值后再点击页面其他地方又会恢复成原来的值，解决办法通过主动触发onchange事件：var event = new Event('input', { bubbles: true });element.dispatchEvent(event);

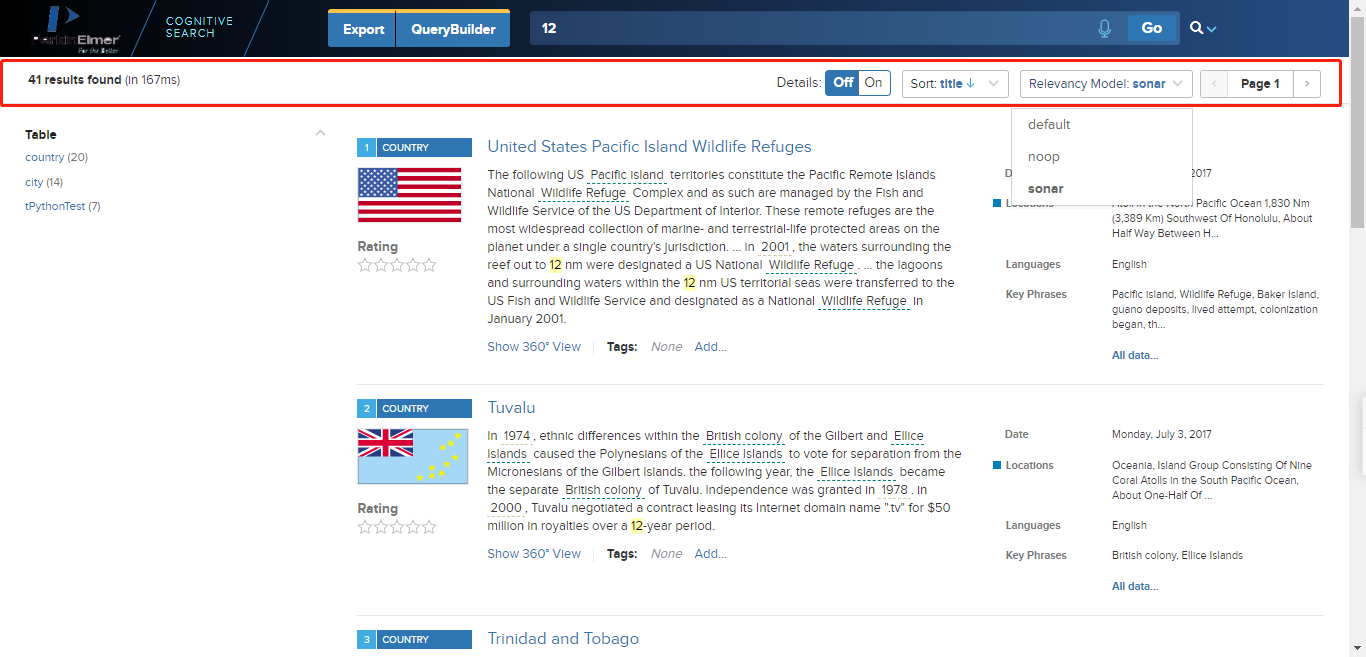




# 5．控制搜索栏二级栏目显示的控件情况

直接修改页面下图中文件的置顶区域





# 6．Searchui中react控件解读

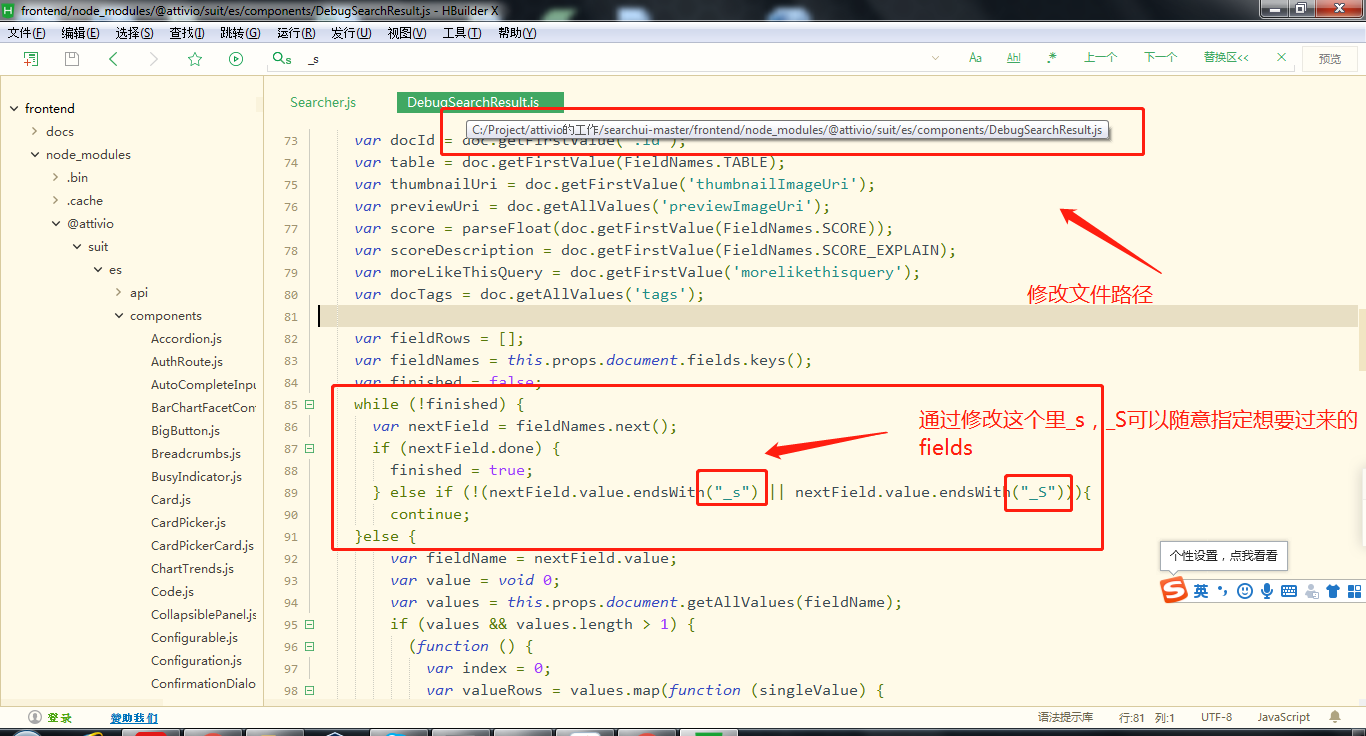
<https://attivio.github.io/suit/styleguide/index.html>

# 7.searchui 查询结果title部分显示异常

只要在生成表时提供一个title字段即可

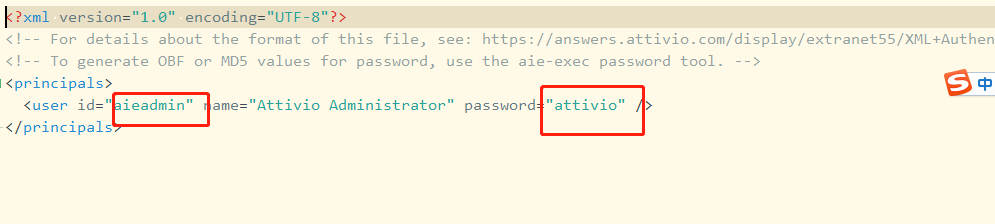
# 8.searchui 过滤fields（我们这里的例子是过滤field中有\_s的）

修改配置方法如下图



# 9.配置attivio searchui的登录密码

C:\java\apache-tomcat-9.0.0\searchui\users.xml，



Id：用户名，password:登录密码

# 10．Python集成到attivio中

<attivio接入python脚本.docx>

# 11．Attivio searchui中配置新的defaultSql机制

增加新的defalutSql机制是指：searchui接受非同源的数据到本地缓存字段defaultSql中，用户可配置是否使用这个defaultSql，使用的话，新的查询语句会根据普通，高级两种方式来拼接。，

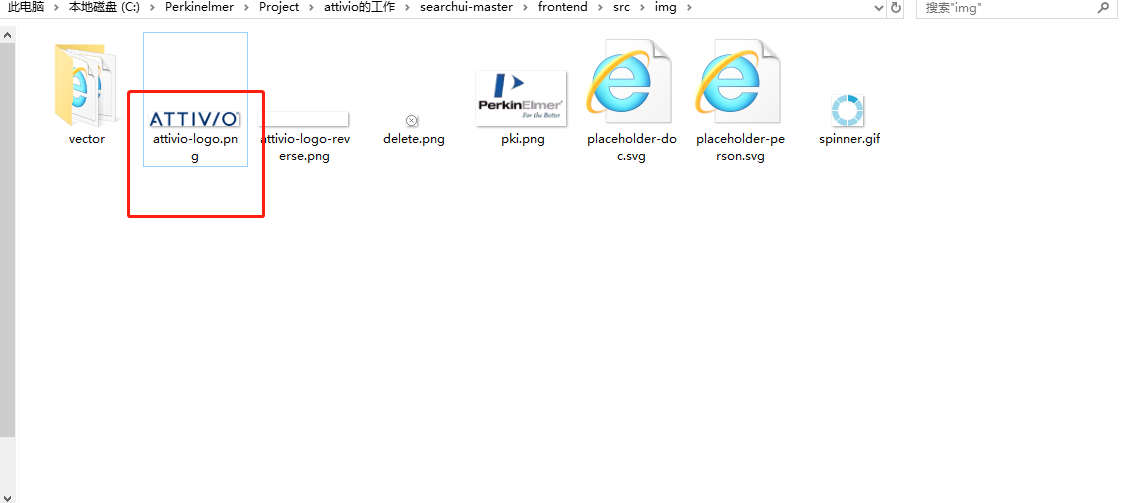
# 12.searchui中添加下载导出的csv文件

主要添加一个download按钮，并将之前export导出得内容按一定规则下载为csv文件，SearchUISearchPage.js自行看对应代码

# 13．Searchui将原来的export导出的字段增加了一个tab

# 14．Searchui查询主页面的LOGO变为Perkinelmer的logo

替换对应目录下的图片文件，再稍调样式即可



# 15.去掉eslint的限制

在你不要收eslint的限制写代码，你可以在你编写的文件最开始添加/\* eslint-disable \*/即可