

AQUABOT 前端网页技术文档

目录

1. 综述.....	3
2. 网站布局思路.....	4
3. 用户界面.....	4
3.1 用户基本信息	4
3.2 用户图表信息	4
4. 主界面.....	5
4.1 Good、Bad 按钮	5
4.2 主地图	5
4.3 Badlist 按钮	5
5. 统计图表.....	6
5.1 饼状图	6
5.2 季节图	6
5.3 地域图	6
6. 使用技术.....	6
7. Q&A.....	6
7.1 bad,good 按钮点以后只显示好或者差，再次点击还原	6
7.2 InitMap 函数.....	7
7.3 如何实时更新地图上的 marker.....	7
7.4 点击 marker A，A 跳动，再点击 marker B，A 停止跳动，B 跳动	7
7.5 点击 marker A，A 跳动，再次点击 A，A 不跳动.....	7
7.6 如何实现实时更新.....	7
7.7 Google Map 生成.....	7
7.8 Badlist 自动添加表格.....	7
7.9 Badlist 表格保存为 excel	7
7.10 如何在网站中同时显示两个 Google Map 地图.....	7
7.11 后端数据库连接不上（即在网页输入端口 api 报错：jdbc 数据库 xxxxx）	7
7.12 HighChart.js 对于图表中数据的更新.....	7
7.13 热像图的更新.....	8
7.14 Ajax 获取.....	8

7.15	Jsonp 理解	8
7.16	Badlist 方法.....	8
7.17	Badlist 里生成的表格右边的滑轮	8
7.18	表格 X 轴的伸缩	8
7.19	Navicat 破解注册机	8
7.20	Download 功能	8
8.	参考资料	8
8.1	Jsonp	8
8.2	Ajax.....	8
8.3	Chart.....	9
8.4	Highchart.....	9
8.5	Jquery+ajax	9
8.6	自动添加 table	9
8.7	地图 style.....	9
8.8	js 保存 table.....	9
8.9	两个地图.....	9
8.10	查看 geojson 效果	9

1. 综述

本文档用于 Aquabot 水质监测网站前端的工作记录。网站主要用于与 Aquabot 公司合作的运营商可以更加方便的对使用 Aquabot 公司产品的用户进行水质监测与管理。整个网站的开发是前后端分离的，后端采用 Spring Boot，数据库使用 MySQL。此文档主要针对前端网页的设计，实现，使用技术，遇到问题进行记录。

网站主要分为——左，中，右三个模块。左边模块用于显示单一用户基本信息，中间模块用于显示主地图相关信息，右边模块用于显示基于所有用户的统计信息。通过这三个模块，运营商可以全面有效的了解用户的水质情况。通过中间的主地图，运营商可以在地图上实时看到用户的水质状态好坏与否以及用户的位置，上面的 good，bad 按钮可以让地图只显示水质为“好”，或者“坏”的

用户，badlist 按钮点击后将显示当前水质为坏的用户信息列表，点击 download 将会下载 badlist 列表。网站左侧的用户信息将展示地图上每一个用户的基本信息，通过点击地图上的某一个用户，左侧上方将显示相应的用户基本信息，如地址，名字，etc，下侧将显示以时间为 x 轴，水质好坏为 y 轴的柱状图图表，其中 1 表示水质为好，-1 表示水质为差，图表下方显示水质相关的参数。网站的右侧分为三个图表，均是基于所有用户的统计数据。第一个图为饼状图，用于显示当前所有用户中好与坏的数量比。第二个图 x 轴是时间，y 轴是所有用户在同一时间中为好的百分比，从而运营商可以有效的看到某一个时间或者某一个月份好或者不好的整体情况，这是展示季节性的图表。第三个图是热像图，用于显示不好的用户在地图上的位置，运营商通过此图可以及时看到某一个地区的水质为“坏”的用户比较多，由此可以着重管理并监控这个地区。

网站前端技术上主要使用 html、css、javascript 完成布局与相关逻辑实现。地图 api 使用 Google Map Api，图表使用 HighCharts.js、Chart.js。

2. 网站布局思路

前端网页首先将网页分成 3 个部分，左边代表用户基本信息，中间是地图，右边是统计信息图表。左边在分成上下两部分，上面显示用户基本信息，下面显示用户图表以及相应的水质参数信息。中间分为 3 部分，上面是 good，bad 按钮，中间是地图，地图上有代表每个用户的 marker，下面是 badlist 按钮。右边分为 3 部分，上面是饼状图，中间是表示季节的折线图，下面是表示地域的热像图。

3. 用户界面

网站中左侧为用户界面，并分为上下两个部分，上面为用户基本信息，下面为用户图表信息。

3.1 用户基本信息

用户基本信息主要是当运营商点击地图上某一个 marker 的时候，要相应的在左侧显示这个 marker 所对应的用户信息。

实现：参考代码中 showInfoWindow 这个函数，通过添加事件响应，在每生成一个 marker 的时候，便对它添加相应的事件响应。因为是点击 marker 后才显示，所以 showInfoWindow 中使用 ajax 的 post 方法从而获取 marker 相应的用户信息。然后拿到后端传回来的信息后，用过 document.getElementById("userCode").innerHTML 来更新对应的值。

3.2 用户图表信息

用户图表信息主要是显示用户水质的历史情况，让运营商可以看到从使用设备开始到此时此刻的所有水质情况，以及相应的水质测量元素数值（如 PH，cl，tds，etc）。

实现：图表采用的是 Highchart.js，代码中是包装在 userChart 函数中，方便后面调用。每当用户点击地图上 marker 时，通过 showInfoWindow 中 ajaxpost 获取的数据，使用 HighChart.js 文档中提供的更新方法 update 来实现更新数据。Marker 点跳动的动画效果是通过 Google Map Api 中 marker

的 `getAnimation` 函数实现的。对于如何实现点击另一个 marker 当前 marker 不跳动，这是通过判断当前获得的 marker 信息中的对应的与用户基本信息中某一个值是否与当前用户信息的对应值相同，比如点击 marker A，左侧显示用户 A 的 code 是 123，当你在点击 marker B 的时候，将会获得新的 code，所以在你更新用户基本信息之前先判断这两个是不是相等，相等代表你点击的还是 A，不相等代表你点击的是 B，从而实现让 A 停止跳动，让 B 跳动。

4. 主界面

网站中间部分为主界面，主界面主要分成 3 个模块，最上面两个按钮，中间的地图，以及下面的一个按钮。

4.1 Good、Bad 按钮

通过点击 Good、Bad 按钮，地图上可以显示当前状态下只有好或者坏的额用户在地图上。

实现：参考代码中 `showBadMarker`，`showGoodMarker`，因为在生成主地图的时候，每生成一个 marker 我会讲这个 marker 对象存放在一个数组中，所以当用户点击 good 或 bad 按钮时，遍历整个数组，判断 marker 对象的水质好坏值，从而不显示好或者坏的 marker。Google Map Api 通过 `markerCount[i].setMap(null);` 的方法让某一个 marker 不显示。通过 `markerCount[i].setMap(map);` 指定 marker 显示在某张地图上。

4.2 主地图

主地图用于显示 Google 地图，并让所有用户显示在地图上，使用绿色 marker 表示水质为好的用户，使用红色 marker 表示水质为坏的用户。点击某一个 marker，将在左侧用户基本信息中，显示点击 marker 所代表用户的信息，并且点击的 marker 会持续跳动，再次点击同一个 marker 当前 marker 不再跳动，或者点击另一个 marker，当前 marker 也不再跳动。

实现：参考代码 `initMap()` 函数，以及 `fullMap` 函数。其中，`initMap` 函数用于生成地图，它需要申请一个个 Google Map Key，并通过回调函数生成，参考函数下方的 script 标签，或者通过 `ctrl-f` 搜索 `initMap` 找到，在这个 script 中要使用 `defer` 的方式加载不然无法同时加载两张地图。不能用 `async defer`，因为 `async` 是异步加载并且加载完就执行，`defer` 是先异步加载，等到 html 读完再执行。

4.3 Badlist 按钮

通过点击 badlist 按钮，将会生成当前状态下所有水质为坏的用户列表，并且会显示一个 download 下载按钮。

实现：参考代码 `badlist` 函数，点击以后通过 ajax 获取用户数据再通过 post 请求相应的用户信息。然后通过动态创建 html 表格生成列表

5. 统计图表

统计图表用于展示基于所有用户的统计信息，分成三个图显示。

5.1 饼状图

饼状图主要显示基于所有用户在当前状态下好的数量与坏的数量。

实现：参考代码 `versusChart` 函数，和 `verses` 函数（讲 `chart.js` 包含在 `verses` 中方便调用）使用 `Chart.js` 中的 `pie chart` 实现，每次 `ajax` 获取相关数据，显示在图表上。

5.2 季节图

季节图用于显示基于所有用户信息在同一时间水质为好的百分比，从而让运营商可以了解某一个时间段或某一个月份，水质好的用户多，或者水质差的用户多，从而对某一个月份着重管理。

实现：参考代码 `statisticChart2` 函数，`seasonalChart` 函数，前者将画图函数包含其中方便调用，后者每次通过 `ajax` 获得请求数据并显示在图表上

5.3 地域图

地域图用于显示基于所有用户并且水质为差，将其以热像图的方式显示在地图上，从而让运营商可以看到某一个地区水质为差的用户数量特别多，在热像图中就显得很红，人少不密集就会偏绿，以此运营商可以有效的管理不同地区。

实现：参考代码 `statisticChart3` 函数，以及 `Google Map Api` 中的 `heatMap` 图层，更新图层是时候数据格式记得存储在 `MVCArray` 中，不然普通数组会导致在更新好伸缩地图的卡顿。

6. 使用技术

地图：Google Map Api key: AlzaSyBlpSzqx999357qCKZ2ne9bMORF6T9Bgil

图表：HighChart.js, Chart.js

获取数据：jquery+ajax

7. Q&A

7.1 bad,good 按钮点以后只显示好或者差，再次点击还原

答：设置一个 `countBadClick`，`countGoodClick` 来记录点击次数，判断次数是 0 表示没点击过，1 为点击过，使用 `setMap (null)` 取消 marker 在地图上的显示，`setMap (map)` 指定 marker 在 map 上显示。

7.2 InitMap 函数

答：用于初始化地图，即只在网页指定部分生成空的地图，marker 是以及相应事件是通过 fullMap 函数添加上去的

7.3 如何实时更新地图上的 marker

答：每次 fullMap 获取成功 ajax 的数据时，都先将 markers 数组里的 marker.setMap (null) 在地图上清除，再将 markers 数组清空，然后重新添加新的 marker 以及在地图上显示，setMap 方法参考 google map 文档

7.4 点击 marker A, A 跳动, 再点击 marker B, A 停止跳动, B 跳动

答：在 showInfoWindow 函数中，因为我们可以获得当前用户基本信息栏中的某一个值，比如编号对应的 innerHtml，或者地址对应的 innerHTML 值，诸如次，并且 showInfoWindow 函数里的 ajax 成功后会获得当前点击的 marker 对应的编号或者地址等等的 innerHTML，判断这两个是否相等，从而知道点击的是当前的还是其他的 marker，如果不是同一个 setAnimation(null)

7.5 点击 marker A, A 跳动, 再次点击 A, A 不跳动

答：showInfoWindow 中最下面有判断当前 marker 是否在跳动

7.6 如何实现实时更新

答：通过 setInterval 的方法，设定时间，每多长时间调用一次函数实现，函数每次被调用都将之前放置的 marker 之类的清楚，再重新放置新的 marker

7.7 Google Map 生成

答：通过 script 标签引用，并添加 google map key，以及所需要的额库，比如生成热像图层需要 visualization 库，最后指定 callback 为 intiMap，但是由于网站是显示 2 个 map，所以不能仿照 google map 官方文档使用 async defer 加载，只能使用 defer，因为因为 async 是异步加载并且加载完就执行，defer 是先异步加载，等到 html 读完再执行。

7.8 Badlist 自动添加表格

答：因为 badlist 按钮时点击以后，在下方自动生成表格，所以需要使用 createElement 操作，自动添加 table 所需元素。参考：<https://www.jianshu.com/p/2d758fbf5449>

7.9 Badlist 表格保存为 excel

答：参考：<https://www.jianshu.com/p/eb32a6954b41>

7.10 如何在网站中同时显示两个 Google Map 地图

答：参考：<https://codepen.io/Marnoto/pen/VLjVZZ> , <https://codepen.io/jovanivezic/pen/dXjXNx>

7.11 后端数据库连接不上（即在网页输入端口 api 报错：jdbc 数据库 xxxxx）

答：可能是 mysql 数据库版本太高，但是后端 pom.xml 中数据库 connector 的依赖版本太低

7.12 HighChart.js 对于图表中数据的更新

答：使用 chart.update，这是官方文档中的方法，详细操作参考官方文档

7.13 热像图的更新

答：不能直接设置普通数据，然后更新数据从而更新热像图中的点，这样做在前几次刷新的实时更新中没有问题，但是随着更新次数过多，比如更新 10 次以上，在缩放热像地图的时候，会出现非常大的卡顿，所以要使用 MVCArray 代替普通的 array，这也是 Google Map 官方文档中的方法。

7.14 Ajax 获取

答对于 ajax 的获取，如果是跨域的，只能使用 get 方法，没有 post 方法，并且传回来的格式是 jsonp 格式，即在 json 文件的外部添加一个回调函数，如 `callback(json{a:"a"})` 并在 ajax 中的参数里指定回调函数名称。但是如果想要 post 请求数据但是又是跨域的话，需要在后端添加 cors，只要很少的代码，参考后端代码中的 `src/main/java/cn/controller/CorsConfig` 文件

7.15 Jsonp 理解

答：参考 <https://tonghuashuo.github.io/blog/jsonp.html>

7.16 Badlist 方法

答：在 badlist 按钮的函数中，我们先使用 get 然后根据每个用户的值判断是否使用 post，这里 ajax 的方法本身是默认使用 async 异步操作的，但是在这里的 post 方法要把异步设置成 false，但是前面的 getasync 依旧保持默认，不然会出错。

7.17 Badlist 里生成的表格右边的滑轮

这个是设置的 overflow 参数，设置成 auto，在 css 文件里 `mapView #mapList .badTable` 的属性

7.18 表格 X 轴的伸缩

答：Highchart.js 里将 chart 的属性设置成 `zoomType: 'x'`，这样可以对 X 轴进行伸缩操作

7.19 Navicat 破解注册机

答：windows 版 链接：链接：<https://pan.baidu.com/s/1wisePPHOoAG6U6OM2cJa4A> 密码：密码：fs3k，如果下载时候显示有病毒，这是因为注册机会让电脑误以为是病毒，暂时关闭杀毒软件，安装完以后再删除相应软件即可

7.20 Download 功能

答：txt 文件中不能通过 `\n` 换行，但是 cvs 可以，以及要注意 address 里的逗号会在 cvs 里作为空格

8.参考资料

8.1 Jsonp

<https://tonghuashuo.github.io/blog/jsonp.html>

8.2 Ajax

http://www.w3school.com.cn/example/ajax_examples.asp

8.3 Chart

<https://blog.csdn.net/ling369523246/article/details/53327686>

<https://www.highcharts.com/demo/line-time-series>

<https://segmentfault.com/a/1190000008498664>

<https://chartjs-doc.abingoal.com/axes/labelling.html>

8.4 Highchart

https://blog.csdn.net/bsh_csn/article/details/51700514

<https://api.hcharts.cn/highcharts#yAxis.labels>

8.5 Jquery+ajax

<http://spxiaomin.github.io/2017/04/06/%E4%BD%BF%E7%94%A8-jQuery-%E5%8F%91%E9%80%81-json%E8%AF%B7%E6%B1%82/>

<https://segmentfault.com/q/1010000000424040>

<http://spxiaomin.github.io/2017/04/06/%E4%BD%BF%E7%94%A8-jQuery-%E5%8F%91%E9%80%81-json%E8%AF%B7%E6%B1%82/>

8.6 自动添加 table

<https://www.jianshu.com/p/2d758fbf5449>

8.7 地图 style

<https://developers.google.com/maps/documentation/javascript/styling>

8.8 js 保存 table

<https://www.jianshu.com/p/eb32a6954b41>

8.9 两个地图

<https://codepen.io/Marnoto/pen/VLjVZZ>

<https://codepen.io/jovanivezic/pen/dXjXNx>

8.10 查看 geojson 效果

<http://geojson.io/>