

项目报告

项目名称：合肥公交线路查询软件

项目地址：DL2023Fall/project_1 at main · cxfann/DL2023Fall (github.com)

项目介绍：项目通过对合肥公交官网中信息的爬取 (<https://www.hfbus.cn/>)，提供了以下功能（运行示例图见最后一页）：

1. **公交线路查询。**依据线路路号（形式包括但不限于数字、汉字、字母及其组合）查询对应线路，输出对应线路运行区间、首末班车时间、站点数量、线路长度、各个站点名称以及显示实时车辆位置。对于双向运行的线路，支持在获取结果时更换结果所显示的线路运行方向；
2. **公交站点查询。**根据站点名称，查询出所有符合条件的站点，显示出站点名称、其地理位置（以区分同名异站）以及途经线路。对于匹配站点不止一个的搜索结果，支持逐个切换显示搜索结果；

项目一共设计了五个 GUI 界面：主菜单、线路查询界面、线路查询结果、站点查询界面、站点查询结果。通过用户操作实现界面间的切换和结果的输出。

数据来源：

实验从合肥公交官网爬取数据

- 1.公交线路数据来源于：

[https://www.hfbus.cn/map/MapIndex.html?type=line&linename=\[线路号对应 Unicode\]](https://www.hfbus.cn/map/MapIndex.html?type=line&linename=[线路号对应 Unicode])

- 2.公交站点数据来源于：

[https://www.hfbus.cn/map/MapIndex.html?type=station&stationname=\[站点名称对应 Unicode\]](https://www.hfbus.cn/map/MapIndex.html?type=station&stationname=[站点名称对应 Unicode])

所设计的函数和类：

由于篇幅限制以及函数输出输出内容较多，详细的函数参数说明，参数限制、返回数据已经体现在代码注释当中。

1. **txt_to_unicode(text)** 对汉字进行转码构造 url, 进而爬取信息。
2. **search_bus_route(bus_route)** 读取用户输入查询的线路信息, 爬取对应网站的 html。爬取时利用 Selenium 库模拟浏览器行为并进行虚拟点击, 以充分获得异步加载的内容。对 iframe 定位并且数据处理后, 输出线路查询结果列表。
3. **search_bus_stop(bus_stop)** 读取用户输入站点信息, 同样通过 Selenium 库模拟浏览器获取信息后输出查询结果列表。
4. 一共设计了 5 个 GUI 界面, 利用 QT Designer 设计完毕后生成了五个对应的类, 包含各个界面的结构信息。分别存放在 **Ui_main_window** 、 **Ui_search_route**、**Ui_search_station** 、 **Ui_route_result** 、 **Ui_station_result** 内
5. 为了实现 5 个界面的初始化以及各自之间的跳转逻辑、结果显示, 在 ui_jump.py 中设计了 **MainWindow** 、 **RouteWindow** 、 **StationWindow** 、 **RouteResultWindow** 、 **StationResultWindow** 五个类, 实现各自窗口下的相关功能;

实验中遇到的困难:

在实验中, 遇到的困难主要来自于两个方面。一方面是软件现实意义带来的困难, 还有一方面则是实现其功能带来的技术困难。

从现实意义角度来看, 有:

1. **合肥公交线路运行形式多样。**不仅有传统意义上的双向对发的线路, 还有很多单循环线 (起点终点位于同一站点, 且线路只有一个运行方向)。如何为多样输出结果设计一个统一的展示方式。

解决: 输出查询结果时, 只显示其中一个方向, 并增加了一个“更换方向”按钮来适应这种多样性。双向线路可以切换方向, 单循环线路无法切换。

2. **合肥公交站点允许站点重名。**即多个同名站点位于不同位置并服务不同线路。

因此，单纯输出站台名称是会带来混淆的。

解决：在站台名称后加入站台的地理位置（如：黄山路-北）保证站台指向明确。在输出站点查询结果时，同样安排了“下一个匹配站点”按钮，实现不同搜索结果之间的切换。

从技术角度看，遇到的困难更多：

1. 线路、站点查询结果的对应网页是**异步加载**的，单纯使用 get 方法得到的静态网页不包含所需信息。

解决：使用 Selenium 库模拟浏览器获取界面后定位 iframe 得到所需的信息；

2. 查询线路时，涉及到实时公交位置的查询。而这部分内容只有在网页中点击对应“线路运行方向”的 onclick 窗口后才会加载。直接获得的网页源码不会加载车辆位置，无法获取所需要的信息。

解决：在使用 Selenium 库模拟浏览器的基础上，定位了所需的 onclick 位置，进行模拟点击并等待相应资源加载完成再获取源码进行分析；

3. 在 GUI 设计方面，面临一个挑战是**文本数量的不均衡**。有些线路具有大量停站，而其他线路只有少数停站。当使用较大的字号时，停站较多的线路可能显示不完整，而使用较小字号时，停站较少的线路可能导致空间浪费。

解决：实施了自动判断线路站点数量并选择合适的字号进行显示，以提高界面的美观度和可读性。

结果分析：

通过对 GUI 界面的使用和交互测试，本次实验较好的实现了各项功能。但仍存在一定不足。例如，由于需要进行模拟点击以及等待数据加载完毕，网页读取

的速度较慢，搜索速度有待加强。而且，软件功能比较单一，还有提升空间。

心得与收获：

- 1. 我认识到了爬虫实践的复杂性，学会了对于异步加载信息的应对技巧，了解了模拟点击的实现方式；
- 2. 了解了 GUI 设计的复杂之处，尤其是如何创建吸引人、用户友好的界面和切换逻辑（感觉我自己设计的界面好丑啊）。

运行结果展示：



图 1



图 2



图 3



图 4



图 5



图 6



图 7



图 8



图 9

图 1：主菜单

图 2：线路搜索界面

图 3：站点搜索界面

图 4、5：一个双向线路示例（66 路）两个不同方向的查询结果，黑点表示该站点区间有实时车辆

图 6：一个单循环线路示例（543 路）查询结果，无法切换方向，黑点表示该站点区间有实时车辆

图 7、8：搜索“合肥南站”不同的匹配结果

图 9：搜索线路不存在时输出的提示词