开发思路

1. 根据 api 开发，例如搜索雾霾
   https://www.sogou.com/sie?ie=utf8&query=雾霾&pid=AWNb5-7772
2. 里面会有搜狗百科的条目
   根据搜狗百科搜索有用信息，
   比如搜索雾霾 ，返回的部分结果为

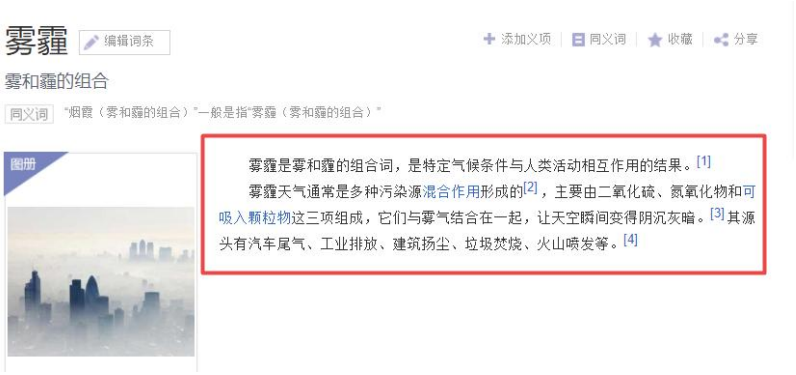
```
<div class="citeurl " vrcid="citeurl.celaee0">
    <span>搜狗百科 - baike.sogou.com/v...</span><span class="cite-date">- 2021-11-25</span></div></div>

  </div>

  </div>
<div class="r-sech ext_query r-sech-test_02 result_list" style="display:none" id="sogou_vr_30010097_sq_ext_1"
data-async="1" data-docid="3c28af542f2d49f7-efa638340bb86e2c-021b379e9767e0faac2e07a2dedea43d"
data-url="http://baike.sogou.com/v7724625.htm?fromTitle=%E7%83%9F%E9%9C%9E"><span><strong></strong>推荐您搜索：</span>
<div class="vrwrap">
  <div class="struct201102">
```

根据下面的链接 http://baike.sogou.com/v7724625.htm?fromTitle=%E7%83%9F%E9%9C%9E
可以得到雾霾词条的相关信息

3. 获取想要的结果



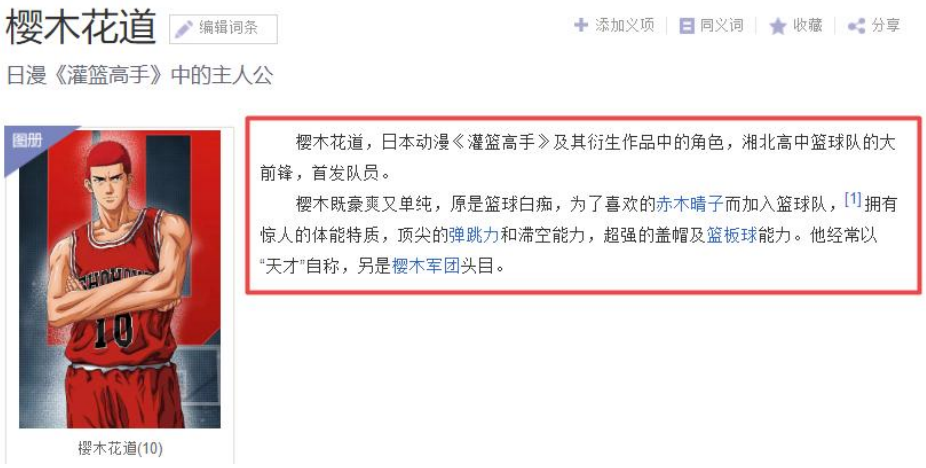
红色框体就是想要的结果，那么从 html 中提取，
首先看首部，这部分是不全的，也可以使用，不过不是很理想.

```
<!DOCTYPE html>
<!-- saved from url=(0065)https://baike.sogou.com/v7724625.htm?fromTitle=%E7%83%9F%E9%9C%9E -->
<html class="">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="save" content="history"><meta name="server" ip="unknown">
<meta name="keywords" content="生活术语 生活"><meta name="description"
content="雾霾是雾和霾的组合词，是特定气候条件与人类活动相互作用的结果。
雾霾天气通常是多种污染源混合作用形成的，主要由二氧化硫、氮氧化物和可吸入颗粒物这三项组成，
它们与雾气结合在一起，让天空瞬间变得阴沉灰暗。其源头有汽车尾气、工业排放、建筑扬尘、垃圾焚烧、火山喷发等。
雾霾，顾名思义是雾和霾。但是雾和霾的区别很大。
空气中的灰尘、硫酸、硝酸等颗粒物组成的气溶胶系统造成视觉障碍的叫霾。霾就是灰霾（烟霞）。
雾是">
<meta http-equiv="X-UA-Compatible" content="IE=edge"><link rel="Shortcut Icon" href="https://baike.sogou.com/favicon_n
    position: absolute;
    top: 16px;
    right: 16px;
    width: 24px;
    height: 24px;
    background: url(//hhy.sogoucdn.com/js/common/hhy/sprite_wap_baike_37443f3.png) no-repeat;
    text-indent: -999em;
    background-size: 84px;
    background-position: -63px 0;
}
```

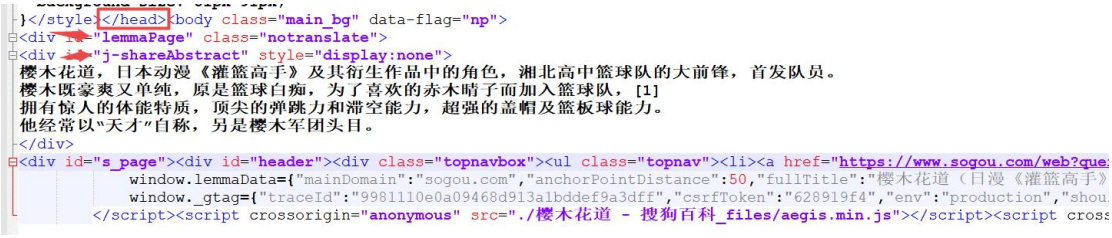
然后看下面的部分

```
    background-size: 81px 91px;
}</style>
</head>
<body class="main_bg" data-flag="np">
<div id="lemmaPage" class="notranslate">
<div id="j-shareAbstract" style="display:none">
雾霾是雾和霾的组合词，是特定气候条件与人类活动相互作用的结果。[1]
雾霾天气通常是多种污染源混合作用形成的[2]，
主要由二氧化硫、氮氧化物和可吸入颗粒物这三项组成，它们与雾气结合在一起，让天空瞬间变得阴沉灰暗。
[3]其源头有汽车尾气、工业排放、建筑扬尘、垃圾焚烧、火山喷发等。[4]
</div>
<div id="s_page">
<div id="header">
<div class="topnavbox">
<ul class="topnav"><li><a href="https://www.sogou.com/web?query=">网页</a></li><li><a href="https://weixin.sogou.com/w
            window.lemmaData={"mainDomain":"sogou.com","anchorPointDistance":50,"fullTitle":"雾霾（雾和霾的组合）","ti
            window._gtag={"traceId":"735ff56019834187b67213a62c2ce934","csrfToken":"628919f4","env":"production","shou
        </script><script crossorigin="anonymous" src="./雾霾 - 搜狗百科_files/aegis.min.js"></script><script crossorig
```

我们试图查看什么规律，现在再来一个搜索词条(樱木花道)



我们想要红色的部分，那么看 html 文件



直接看下面，还是一样的规律，那么大概我们可以猜到，这个词条的内容就是以

开始的一段文本，直到 结束
那么剩下的就是根据正则表达式就可以筛选出最终的结果.

开发实战

在了解原理之后，开始开发.
首先使用 qt 来实现网络的连接，发送请求和获取回复.
首先添加 Qt 模块
QT += network
包含头文件
#include<QNetworkAccessManager>
#include<QNetworkRequest>
#include<QNetworkReply>

我们需要用到的类为
QNetworkAccessManager *network_manager ; //发送请求搜狗搜索
QNetworkRequest *network_request;
//这里使用的类 QNetworkAccessManager 用于管理网络连接, QNetworkRequest 用来实现请求

创建类的对象

```
network_manager = new QNetworkAccessManager();
network_request = new QNetworkRequest();
```

写请求后得到回复的处理

```
connect(network_manager, SIGNAL(finished(QNetworkReply*)),
        this, SLOT(slot_replyFinished(QNetworkReply*)) );
```

发送 url 请求

```
network_request->setUrl(QUrl(url));    // url 是 Qstring 类型，表示要请求的 url 链接
network_manager->get(*network_request);
```

我们将以上部分使用 NetRequest 类进行封装

在构造函数中完成创建对象以及 connect 连接信号槽函数

实现发送请求函数

```
void NetRequest::slot_sendUrlRequest(QString url)
{
    network_request->setUrl( QUrl(url) );
    network_manager->get( *network_request );
}
```

通过上面的步骤可以将请求发送，一旦对方给出请求回复，那么会发出 finish 信号，进而有 connect 的槽函数 slot_replyFinished(QNetworkReply*)进行处理.

实现如下:

```
void NetRequest::slot_replyFinished(QNetworkReply *reply)
{
    if( reply->error() == QNetworkReply::NoError )
    {
//        qDebug() << __func__ <<"success";
        QByteArray bt = reply->readAll();
        Q_EMIT SIG_getResult( 200,  bt );
    }else{
//        qDebug() << "发生错误";
        QByteArray bt;
        Q_EMIT SIG_getResult( -1, bt );
    }
    reply->deleteLater();
}
```

返回的信息会以信号的形式传出，这样处理的类只需要连接信号，使用槽函数处理即可

对于返回信息的处理

```
#include<QCoreApplication>
#include<QFile>
#include<QRegExp>
/// Widget 是一个界面类 链接请求类( NetRequest )的对象分别为 netRequest1 , netRequest2(对应两个不同链接)
void Widget::slot_dealFirstReply( int code , QByteArray bytes )
{
    if( code < 0  ){
        qDebug() << "request fail ";
        return;
    }
//可以考虑先写入文件查看
//QString FileName = "/1.html";
//QString FilePath = QCoreApplication::applicationDirPath()+FileName;
//QFile file(FilePath);
//if( !file.open(QIODevice::WriteOnly))
//      return;
//file.write( bytes );
//file.close();
```

```
    QString result(bytes);   //转化为字符串
    //提取
    int idx = result.indexOf( QRegExp("data-url=\"https*://baike.sogou.com") );
    QString str = result.mid( idx );
    int beginIdx = str.indexOf(QRegExp("https*://") );
    int endIdx = str.indexOf("\">");
    str = str.mid( beginIdx , endIdx - beginIdx );

    qDebug() << str;
    //发送第二个链接请求
    netRequest2->slot_sendUrlRequest( str );
}
```

以上可以针对回复做出处理.
接下来的步骤与上面类似，下面给出第二个链接的回复处理

```
void Widget::slot_dealSecondReply( int code , QByteArray bytes )
{
    if( code < 0  ){
        qDebug() << "request fail ";
        return;
    }
    QString result(bytes);   //转化为字符串
    //提取
    int idx = result.indexOf( QRegExp("<div id=\"j-shareAbstract\" style=\"display:none\">") );
    QString str = result.mid( idx );
    int beginIdx = QString("<div id=\"j-shareAbstract\" style=\"display:none\">").length();
    int endIdx = str.indexOf("</div>");
    str = str.mid( beginIdx ,   endIdx - beginIdx );
    qDebug() << str;
    Q_EMIT SIG_searchResult( str );
}
```

可能遇到的问题

因为网络连接有的是 http，有的是 https，那么需要对 https 这种连接支持
Qt 默认是不支持 https 的连接访问的.
解决办法:
首先安装 openssl 库

Win32OpenSSL_Light-1_1_1q.exe

然后默认选项安装.
之后我们会得到 dll 的文件如下

那么只需要在 qt 编译好的 exe 路径下添加这两个 dll 文件即可.