技术选型理由：

现在个人电脑使用的多是win10系统，开发win10的wpf应用符合用户的需求。

wpf应用现在只支持VS和C#开发，使用微软开发的语言和IDE开发微软的操作系统应用当然是最好的选择。经过这几年的发展，wpf社区也已经逐渐壮大。

架构：

MVVM是微软面对wpf应用提出的前后端分离框架，借鉴了传统桌面软件设计的MVC框架。使用MVVM利于软件的升级。

模块划分：

网络查询相关类写在NetworkHelper文件夹下。

查询返回数据类写在NetworkData.Query文件夹下

网络异常类写在Exceptions文件夹下

强制转换函数写在Converter文件夹下，其下的UI文件下写xaml绑定中用到的converter，Json文件夹下写Json数据读写时的类型转化函数。

默认设置写在Config文件下

Data下储存证书，保存车站编码的Json文件，和保存URL编码的Json文件。

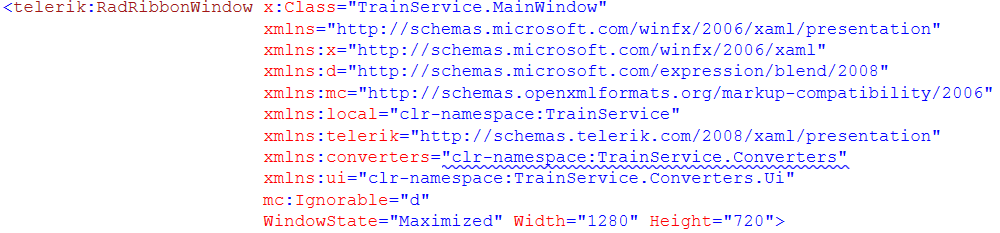
LocalData下写本地数据的类，以及从本地的保存车站编码的Json文件中读取数据的函数。

Uihelper文件夹下写Ui界面上进行排序和过滤的函数

使用的具体技术：

1.Telerik控件：

能够快速构建UI界面，并且实现多行数据的排列。



上图为部分节选，所有控件代码位于MainWindow.xaml

2.MVVM框架：

LocalData文件夹以及NetworkData文件夹下类是Model

MainWindow.xaml是View

MainWindowViewModel.cs是ViewModel

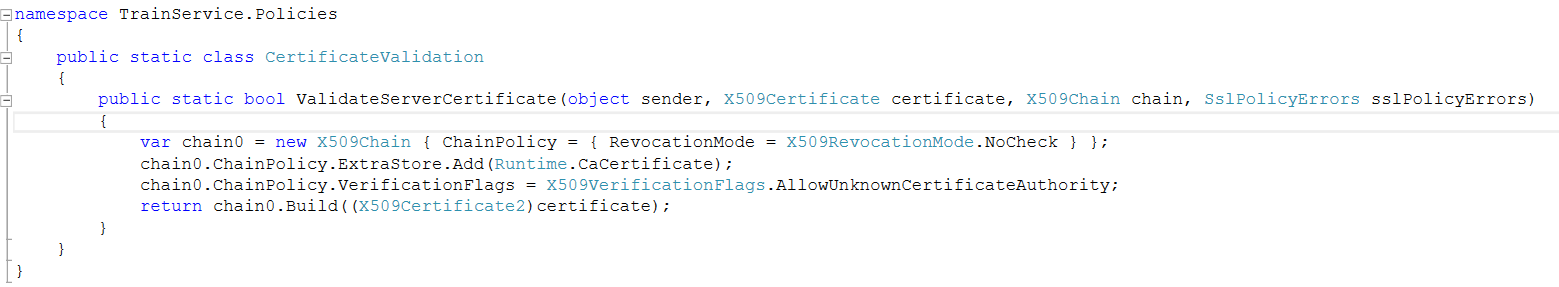
3.Json.net以及其序列化用法：

出现在LocalData文件夹以及NetworkData文件夹以及Json文件夹下所有类



上图截选自NetworkData.Query下的QueryResponse.cs

运行时添加12306证书：



上图截选自Policies文件夹下CertificateValidation.cs