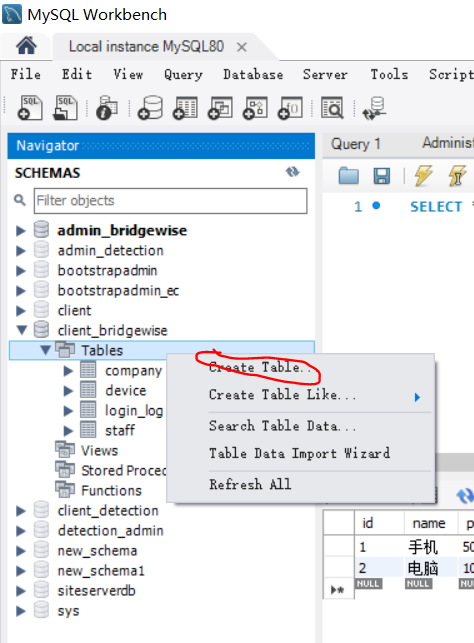
## Model

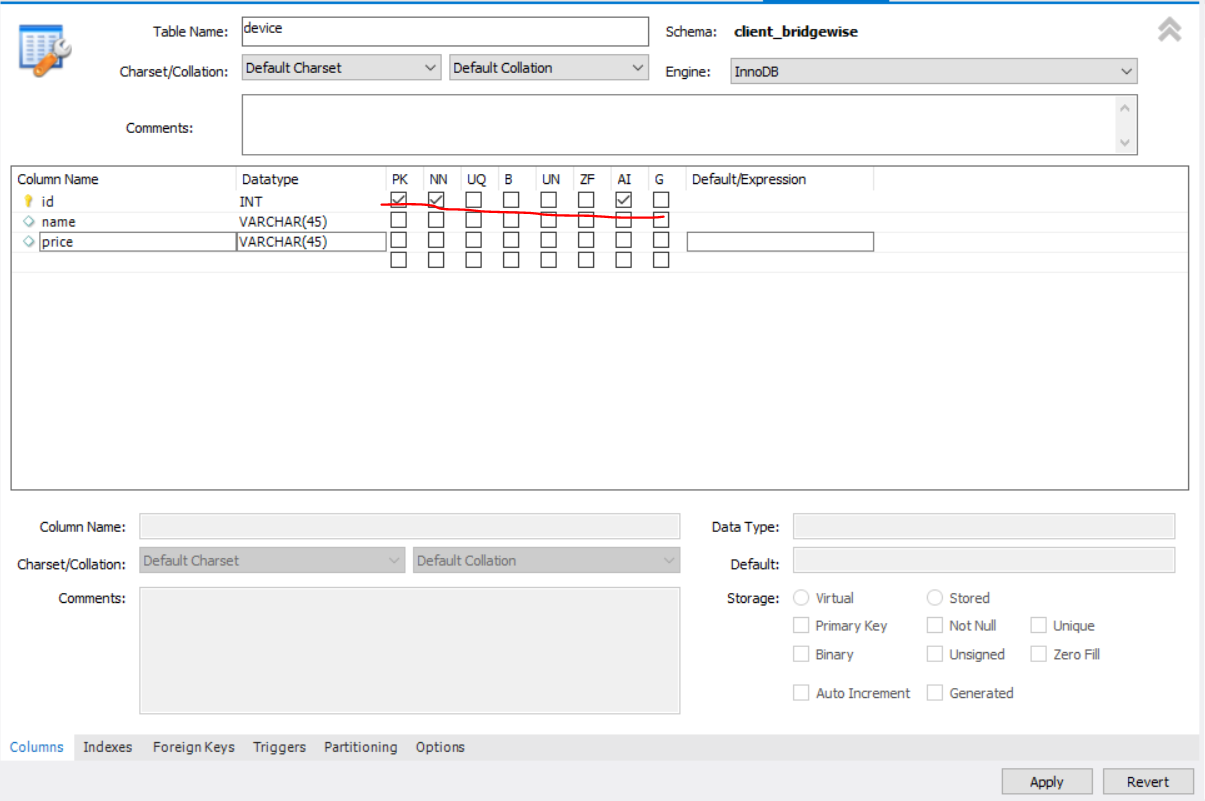
想好要建立一个什么样的实体类，比如设备，详细定义这个类有哪些字段，比如设备名称、设备类型等。然后，进行以下操作：

1、在数据库中建立表，比如device



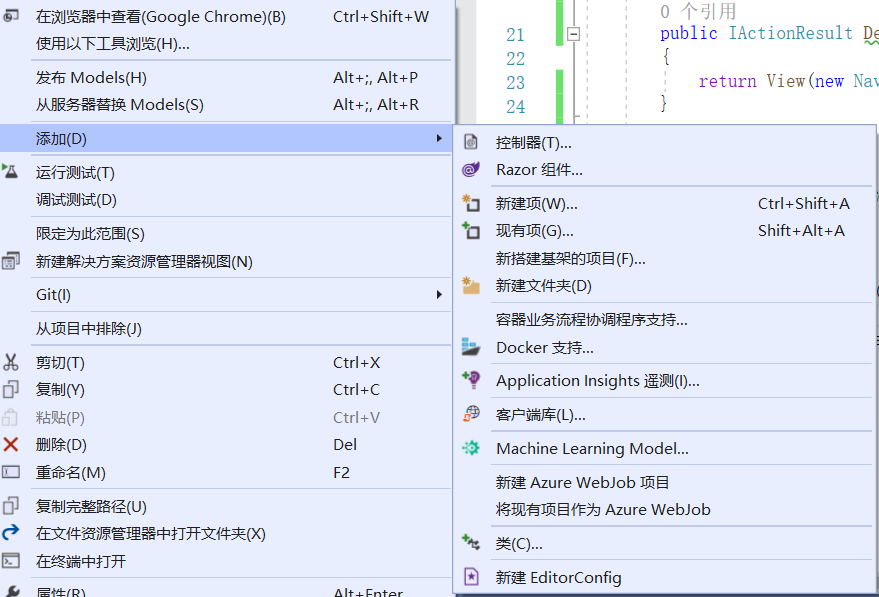
添加字段，一般都会有id这个字段，注意勾选pk，nn和ai

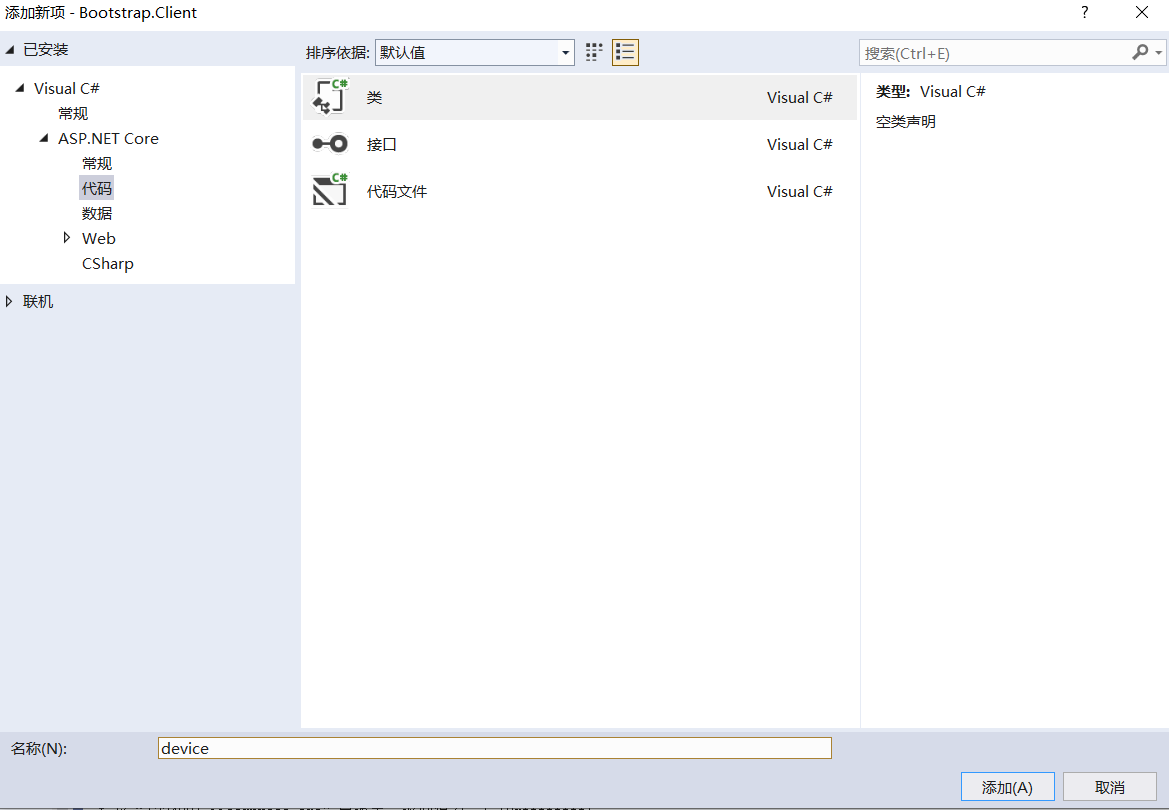
其中pk的意思是primary key，nn是not null，ai是aotu increase（自动加1）



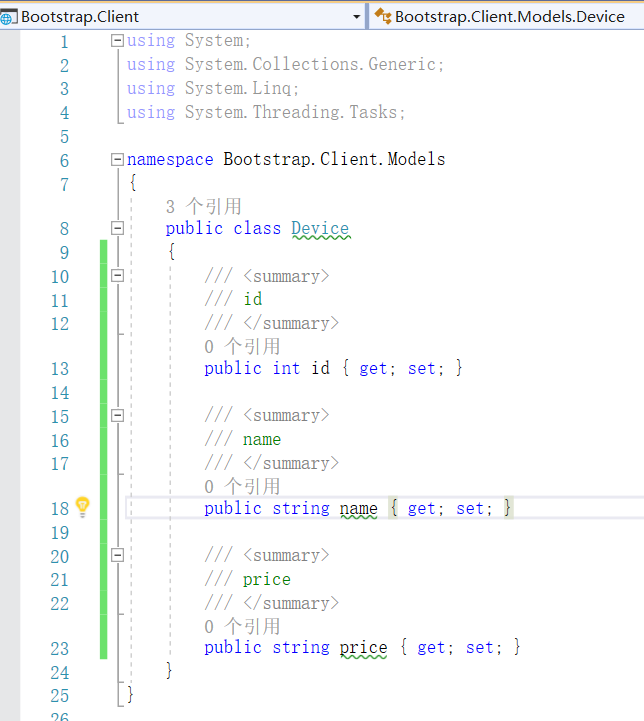
定义好了之后可以自行添加一些数据供后续测试用

2、在Model中添加新的类，比如device

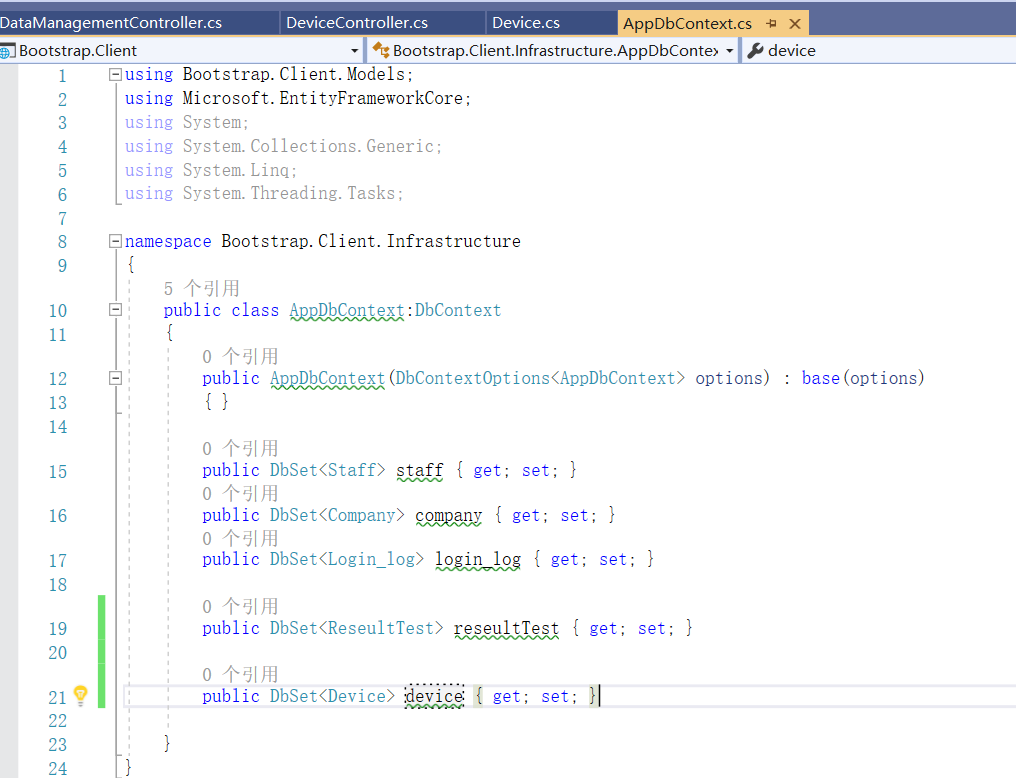




在device类中定义相应的成员



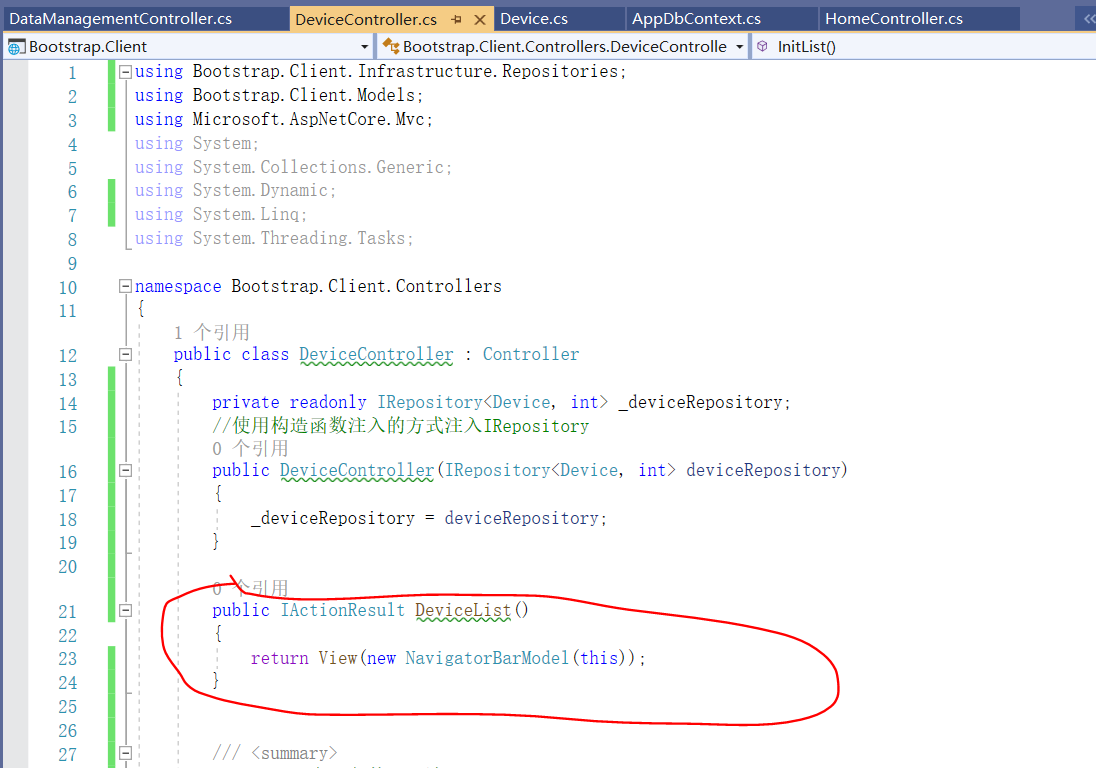
3、在AppDbContext.cs（在Infrastructure文件夹下面）中注册这个新的实体类，以实现数据库操作



## Control

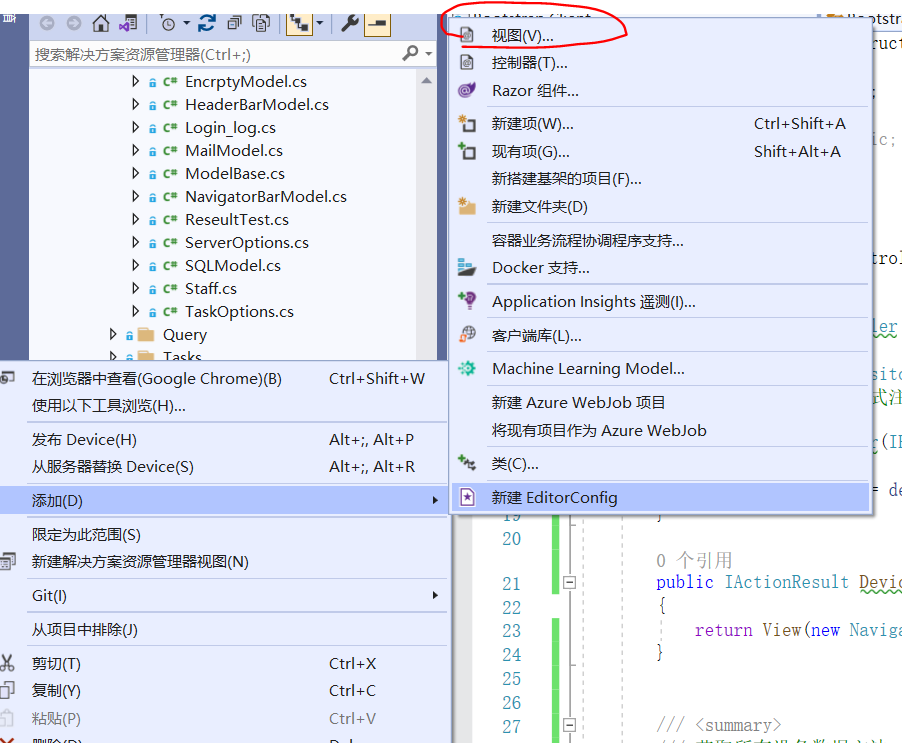
1、添加一个新的controller（右键Controllers文件夹，添加一个新的controller），比如DeviceController

2、自定义一个动作action，用于关联视图，比如DeviceList视图，注意使用demo中的写法，以使用母版



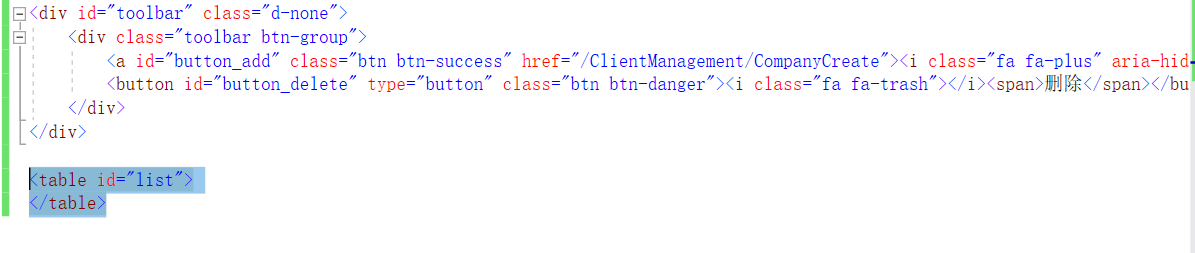
## View

1、新建一个视图，与DeviceController中的devicelist动作对应。在Views文件夹下面新建一个与controller对应的文件夹，比如Device文件夹，然后右键该文件夹，新建一个视图文件DeviceList.cshtml



2、在上述视图文件中用html、js等前端开发语言设计页面，比如想放一个表格显示设备列表，定义了一个<table></table>，需要注意的是，为了使用母版，请参考其他页面的写法



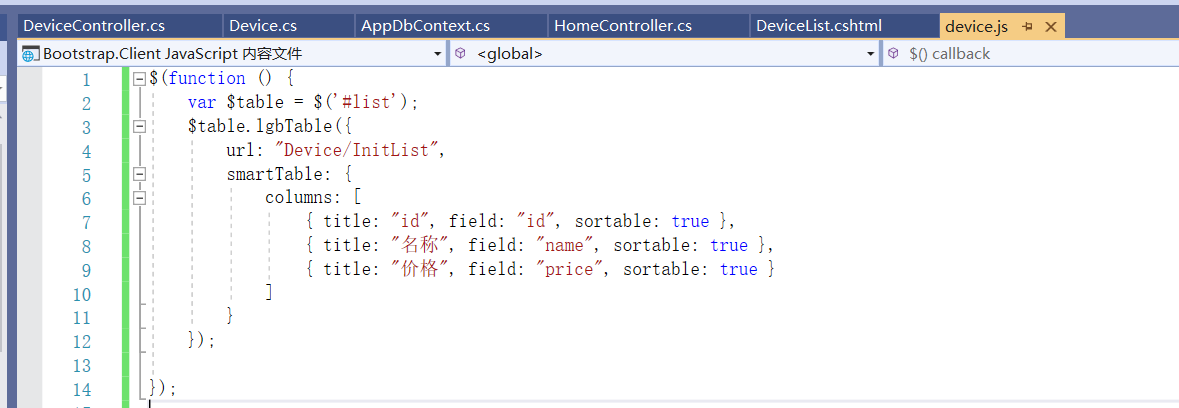


3、采用js的方式将前台后台关联起来，以实现列表的初始化，可以将js写在外部文件中，这样可以让整个前台页面看起来清爽，注意要在视图文件中将这个js文件导入

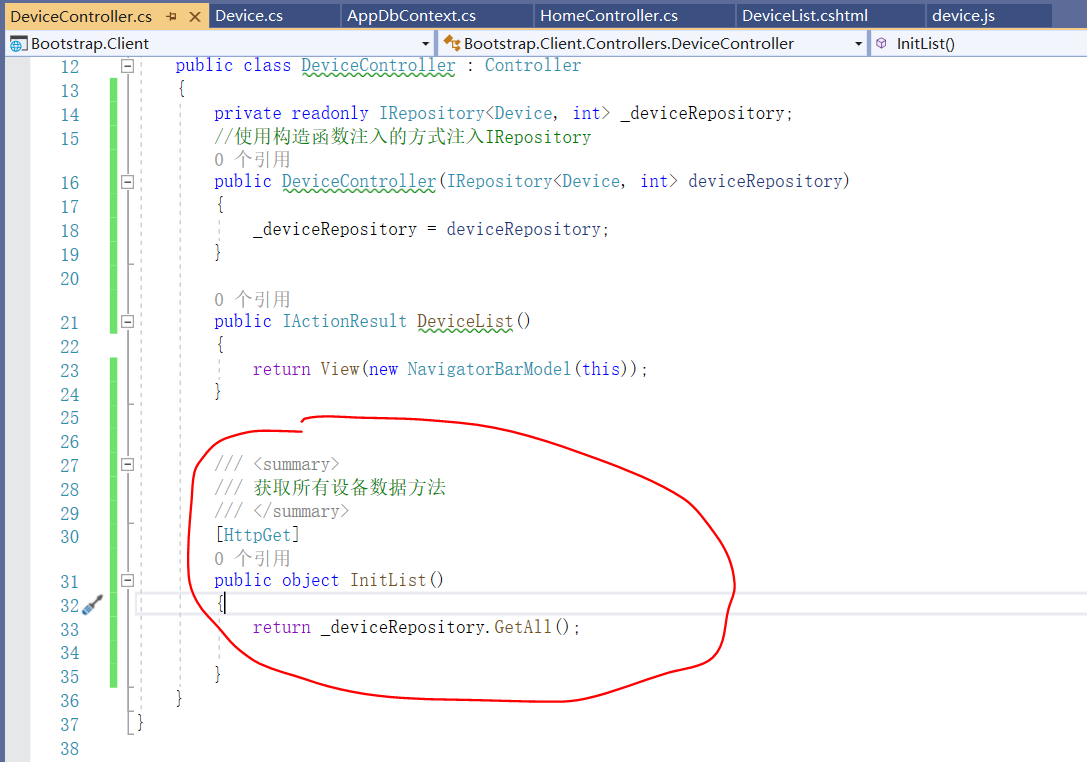
<script **src**="~/js/device.js" **asp-append-version**="true"></script>

在wwwroot/js文件夹下面添加一个device.js文件。

4、实现js文件，可以参考其他js文件的写法，进行表格初始化操作

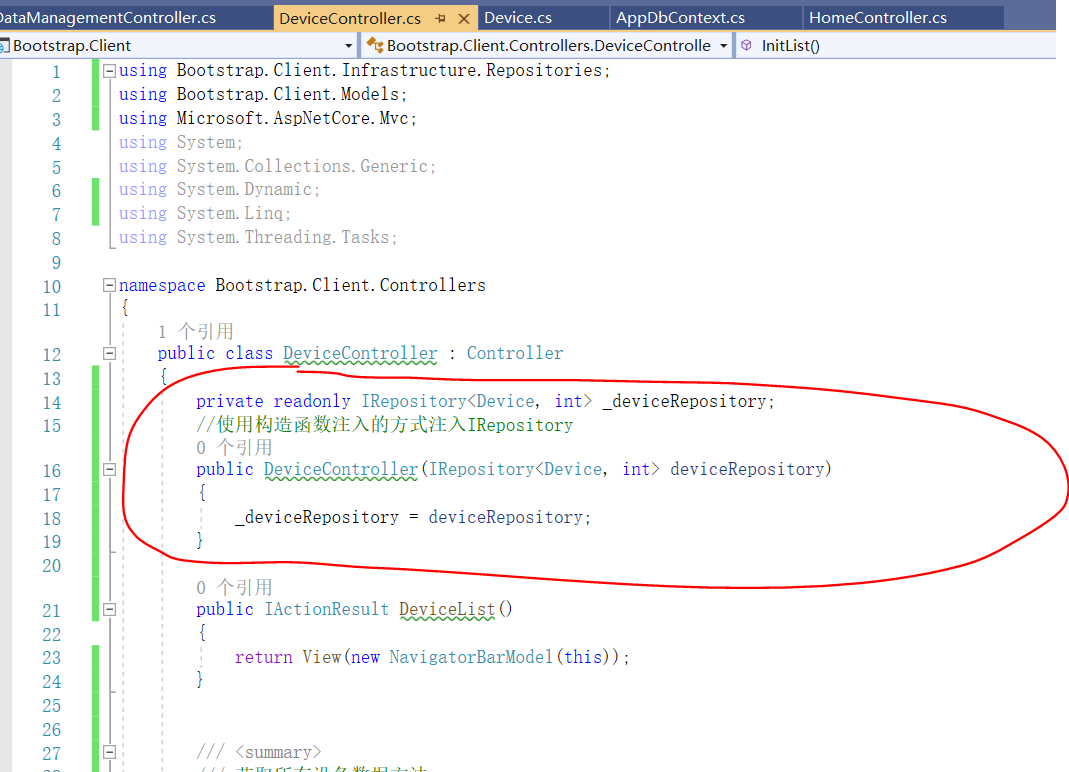


这里需要注意的是url，这是沟通前台后台的关键所在，url的写法是[controller/action]，对于这个设备管理，controller就是device，要初始化列表，可以新建一个动作action，比如InitList，采用Get的方法进行取值



5、在这个initlist动作中实现从数据库中获取所有设备，并返回到前台的功能。

采用依赖注入的方式，实现数据库操作接口



然后进行数据库操作



完成以上步骤后，就完成了设备列表显示的代码，接下来只需要在后台管理系统中添加一个新的菜单，就能将这个功能呈现在用户界面上！

