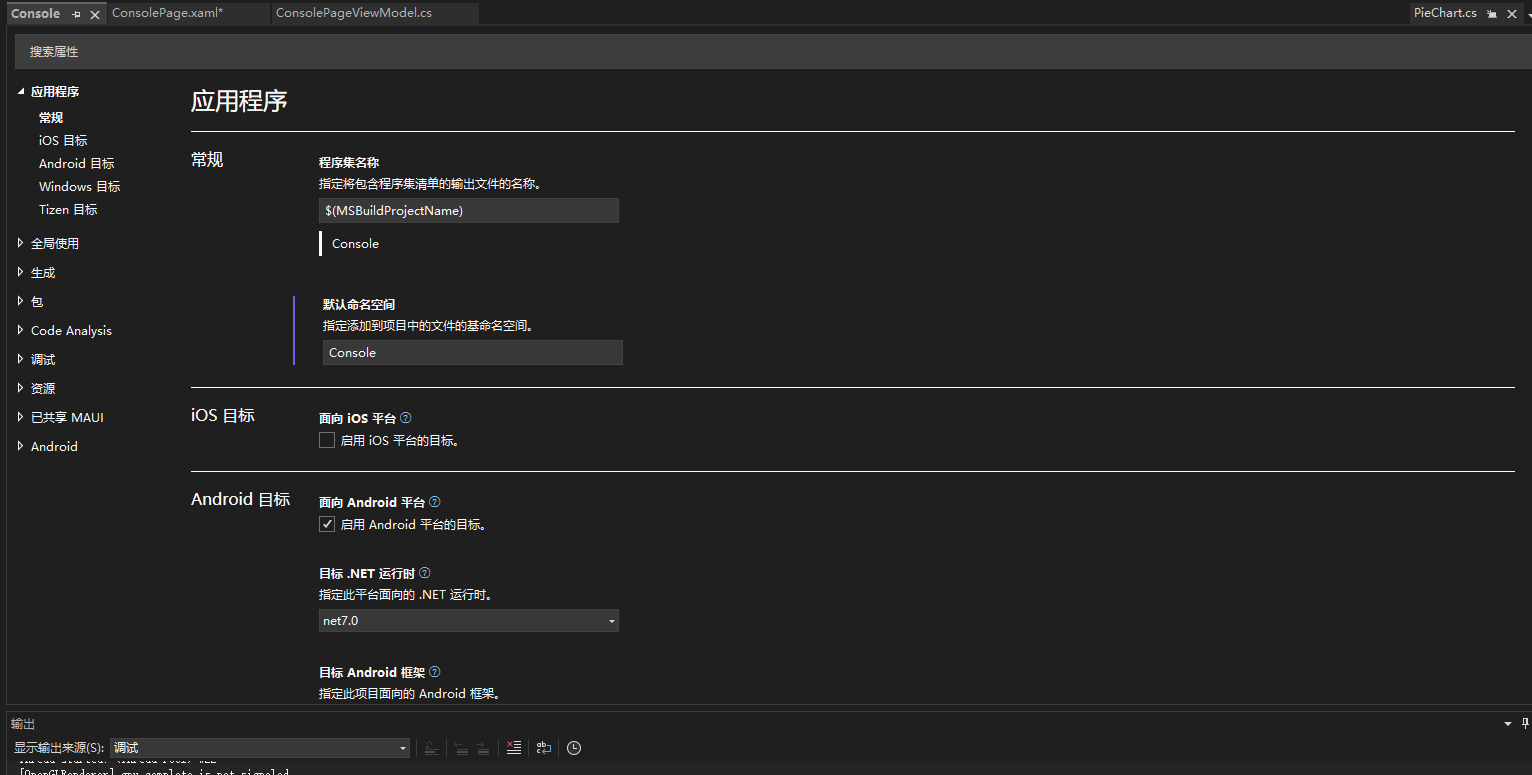
# .NET MAUI 开发文档

## Android开发环境配置

创建好项目后，在解决方案资源管理器中点击项目名称，右击后选择属性打开项目属性管理面板





1. 去掉ios目标，windows目标和Tizen目标三项
2. 在android目标中，选择最低目标android框架（推荐Android8.1 API级别27 可以适配目前绝大多数android系统）

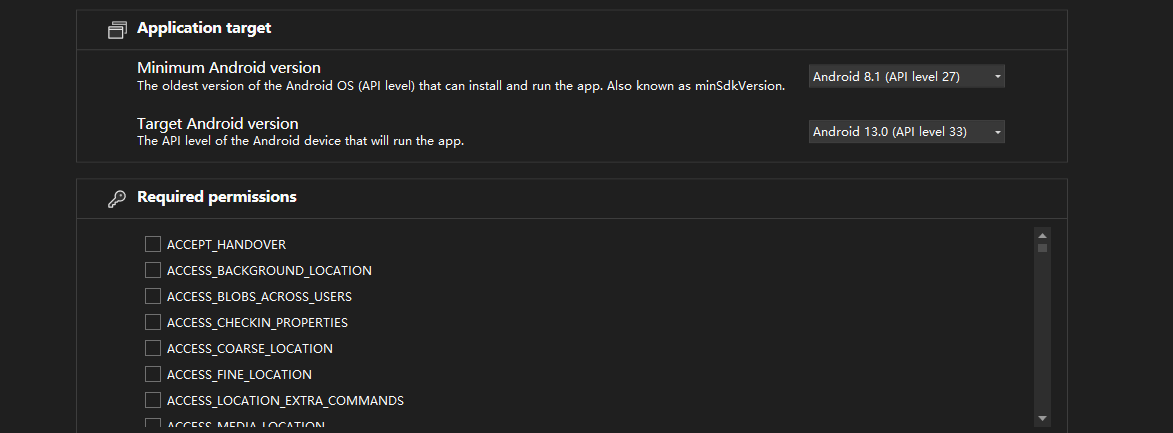


1. 在已共享MAUI项中，可设置应用名称，安装包名称，应用版本号等多个自定义项



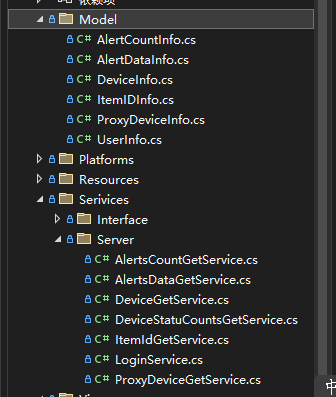
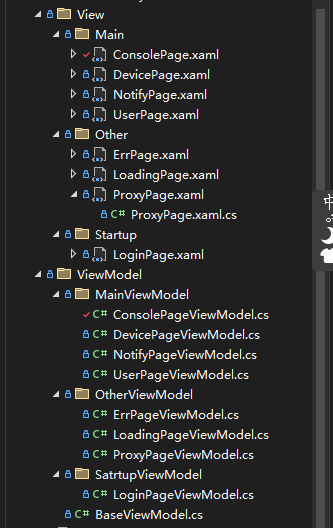
1. 展开项目文件夹，在Platform-android路径下，双击AndroidManifest.xml文件，打开android配置界面，在磁珠可以配置目标android版本，最低android版本和需要获取的权限等。推荐最低android版本Android 8.1 API level27，目标android版本选择最新版。配置需要的android权限勾选对应项即可。



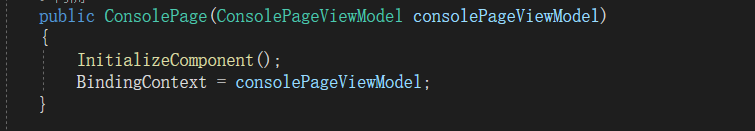


## 写法规范和MVVM

因为MVVM功能的存在，因此推荐在编写项目时将页面和页面对应的功能区分开，作为两个文件独立存在，即页面的Page和其对应的PageViewModel，以方便管理。同时通用的服务Services和通用类Model也建议做好单独存放来方便区分。



在Page对应的.cs文件的主函数（页面对用的函数，下图中的ConsolePage函数），添加PageViewModel参数（ConsolePageViewModel consolePageViewModel），再在函数中为页面绑定参数（BindingContext = consolePageViewModel）即完成Page和PageViewModel的绑定。

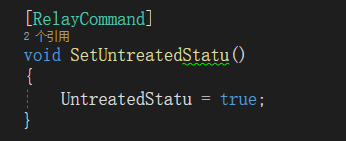


MVVM社区工具包简介：

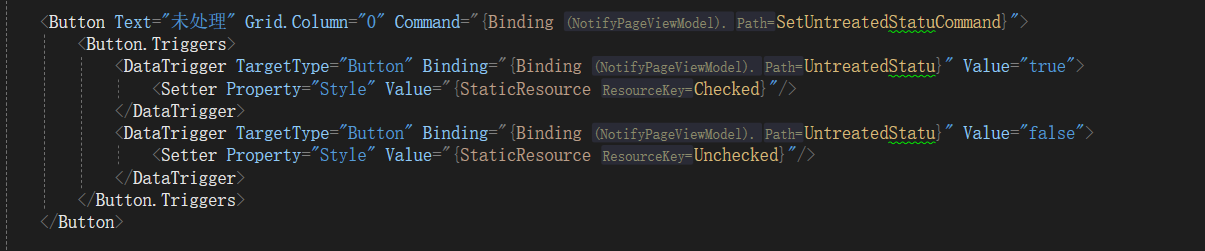
1. RelayCommand。RelayCommand作用于函数。在函数前添加[RelayCommand]即可调用。主要作用为为方法生成一个可供调用的Command方便控件或者行为调用。

Eg1：同步任务

同步任务即不占用主线程，不需要调用异步属性的方法。例如如下改变参数状态的方法SetUntreatedStatu，在使用RelayCommand后，会生成一个SetUntreatedStatuCommand方法以供调用。

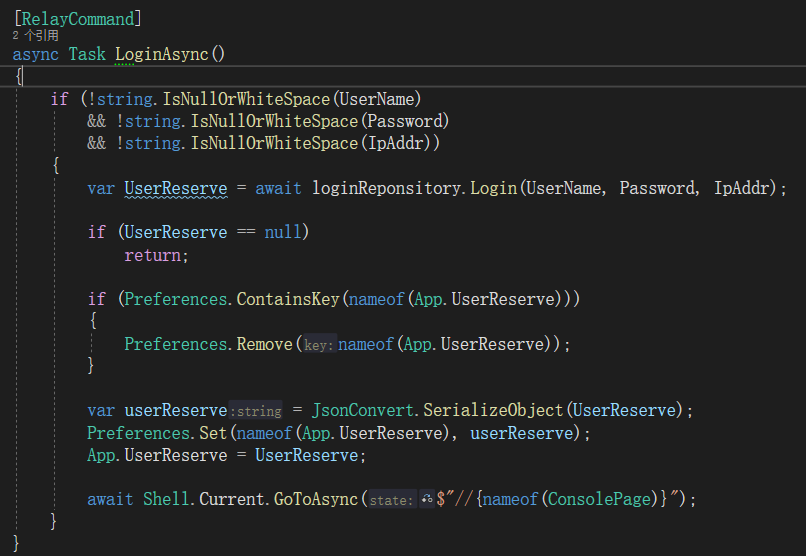


如下图所示，为一个Button绑定SetUntreatedStatuCommand方法后（Command="{Binding SetUntreatedStatuCommand}"），在点击按钮时，即可调用该方法。

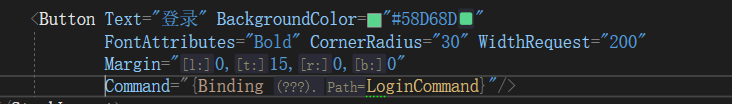


Eg2：异步任务

即需要占用主线程，调用异步属性的方法。例如如下登录函数，执行页面跳转时（await Shell.Current.GoToAsync($"//{nameof(ConsolePage)}");）需要占用主线程，因此需要异步属性（Async）。在函数名称末尾添加Async后，会生成一个LoginCommand方法以供调用。



如下图所示，为登录按钮绑定LoginCommand后，点击按钮即可执行LoginAsync方法。



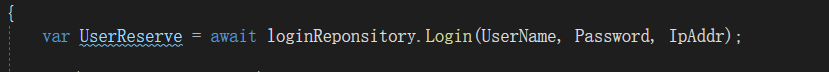
（2）ObservableProperty。ObservableProperty作用于可观测对象，例如实例化类，列表，char，int等类型的可变量。在参数前添加[ObservableProperty]即可调用。主要作用为对象添加一个观测方法，在对象发生变化时，通知调用改对象的控件，参数，方法等同步发生改变。

Eg：如下图所示，为string \_userName添加[ObservableProperty]后，会生成一个string UserName以供调用。



如下图，为Entry的Text属性绑定UserName后，当用户在Entry控件输入用户名后，UserName的值也会同步改变，LoginAsync函数只需要调用UserName即可。



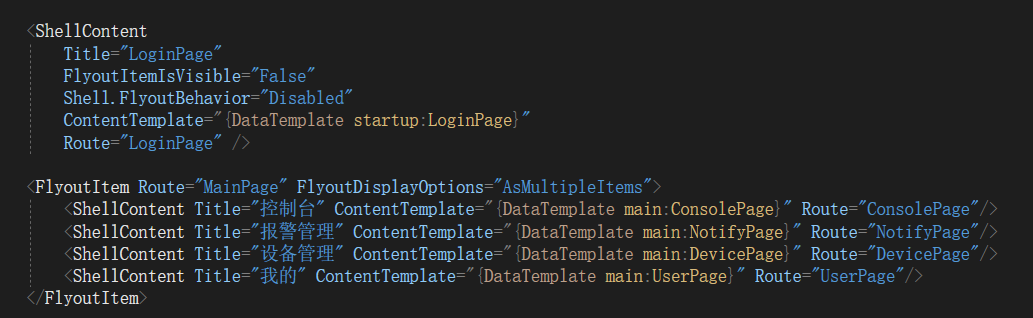


## Shell和页面跳转

所有的View，ViewModel和Service都需要在项目根目录的MauiProgram.cs文件中注册后才能使用，否则会无法调用。Services可以注册为Transient，View和ViewModel必须注册为Singleton。注册方式如下图所示，调用builde.Services服务即可。



想要为View实现跳转功能，还需要AppShell.xaml中为每一个页面注册。不同的注册方式可以实现不同的页面效果，例如飞出，导航栏，弹窗等等，具体注册方式见微软官方文档。需要注意的是，注册属性中的Route属性即为页面的路径，页面跳转均依赖设定的Route实现。下图中LoginPage为单独界面，ConsolePage，NotifyPage，DevicePage，UserPage四个页面组合生成导航栏。

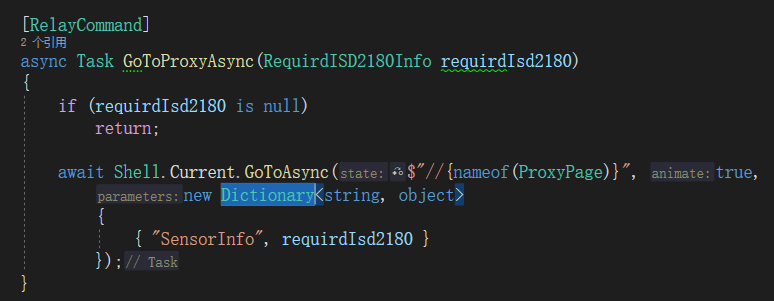


不传递参数的页面跳转直接通过Shell.Current.GoToAsync服务完成。下图即为完成登录服务后，跳转到ConsolePage页面的异步任务。



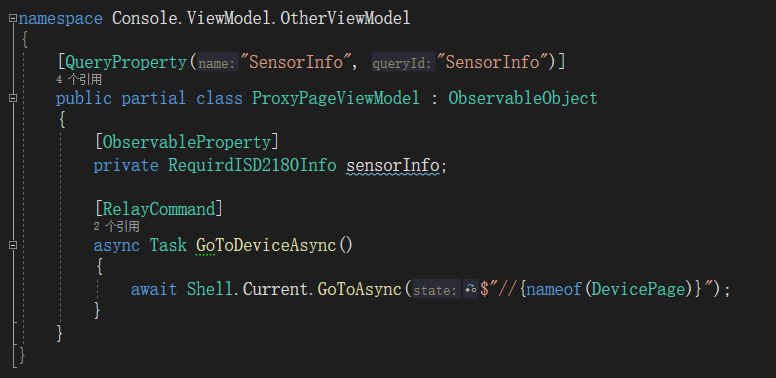
需要传递参数的页面跳转可通过在跳转服务中创建一个Dictionary实现，通过Dictionary可以传递多个参数给需要跳转的界面。

如下图所示，在页面跳转服务中创建了一个Dictionary，通过Dictionary将requirdIsd2180参数传递给下一个界面，同时为requirdIsd2180设置一个索引SensorInfo，跳转后的页面可以通过搜索SensorInfo，获取requirdIsd2180参数。



如下图，在跳转后的页面中，可以使用QueryProperty来搜索SensorInfo，同时将搜索到的SensorInfo对应的参数传递给在当前页面新建参数SensorInfo。如此，就完成了页面间带参数的页面跳转。

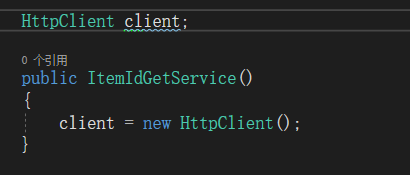
需要同时传递多个参数的话，在Dictionary中添加其他参数即可。



## HttpClient

使用HttpClient服务即可向服务器发送Http请求。

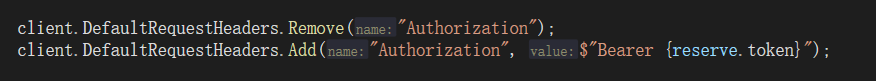
1. HttpClient。如下图所示，直接创建HttpClient服务，并为服务实例化即可创建一个HttpClient。



1. url。即请求地址，类型为string，直接输入需要请求的地址和携带的参数即可。下图即为请求项目数据的url。



1. Headers。对于需要添加Headers的请求，例如携带token的请求，可以通过DefaultRequestHeaders.Add服务为HttpClient添加Headers。下图即为分别为一个HttpClient服务client删除和添加Authorization 类型的Headers的操作，其中Bearer {reserve.token}为Authorization的值。

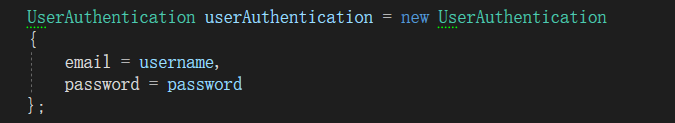


1. Get。直接调用GetAsyns即可完成get请求。下图即为client通过get请求url并将返回的数据传递给response的指令。



1. Post。Post请求需要通过PostAsJsonAsync实现。下图即为client通过post请求url并将返回的数据传递给response的指令，其中userAuthentication为像服务器传递的参数。userAuthentication是一个类型为类，将需要传递参数全部放入该类中即可。





1. JSON解析。C#中需要提前为返回的JSON创建好类才能正确解析服务器返回的JSON，也可以根据创建的类来筛选返回到的数据中需要保留的部分。解析通过ReadFromJsonAsync服务实现。

如下图，当返回值response的状态码正确时，开始调用ReadFromJsonAsync服务解析返回值response。其中UserResponse即为提前创建好的，需要保留的数据的类。userResponse为UserResponse类型，用来存放解析后返回的数据。

