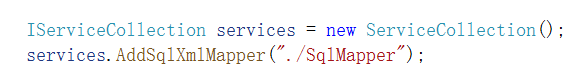
**SqlXmlMapper使用说明**

**使用场景**

针对业务场景中复杂查询的情况，使用EF虽然可以满足实现功能但是无法保证sql语句的可预料性。所以引入了将复杂Sql查询语句的xml化，编写规则符合ado.net的编程习惯，使用也比较简单。

**引入方式**

在需要使用的项目中引入IServiceCollection扩展方法AddSqlXmlMapper，其中传递的参数为sqlxml所在的文件夹路径

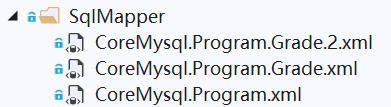


支持传递多个xml文件夹路径

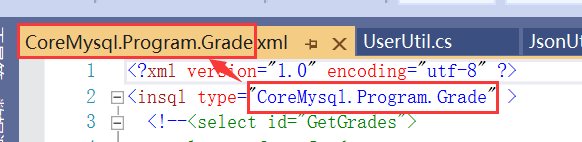


**编写SqlXml**

①Sql语句是存放到上文配置的xml文件夹中的（xml文件要在vs上配置为始终复制）。



②Xml文件的名称必须以xml节点中insql type的值开头



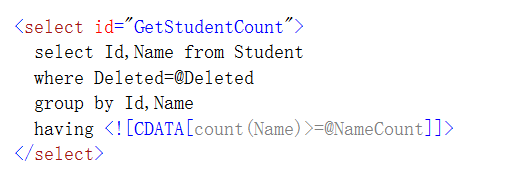
③目前支持多个xml共享一个命名空间type的形式，但是必须保证其中Type+SqlId组成Sql节点唯一标识

大致编写形式如下



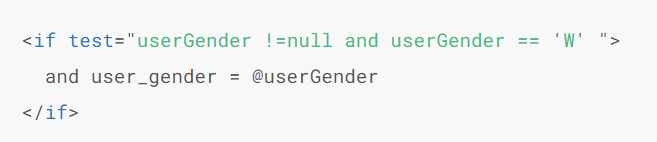
支持动态where和if判断，sql语句的形式和原生的ado.net写法保持一致即可

如果不需要使用动态条件判断的时候直接写sql语句即可

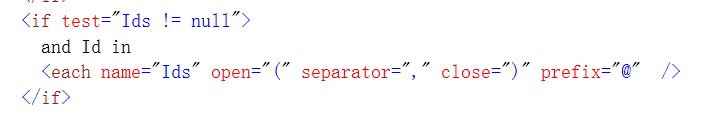


xml 中的配置语法类似于 Mybatis 的配置语法，目前支持以下配置节：

**If**



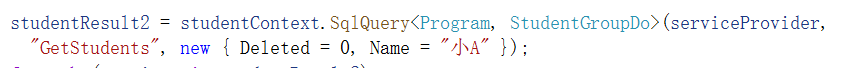
**In**



**Where**



**使用DbContext扩展**



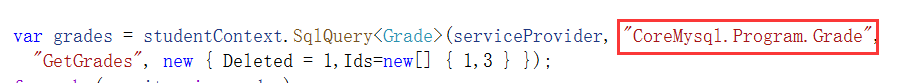
①泛型第一个参数为命名空间即和xml里的声明的type保持一致，标识一组sql的标识

比如这里可以是我们项目中的某个具体的Repository

②泛型的第二个参数表示查询返回结果对应的实体类型

③参数列表第一个表示IServiceProvider实例，第二个参数表示Sql的唯一标识Id，第三个参数表示要传递的参数查询条件

如果是在一个Repository中想查询别的Repository命名空间下的sql,但是并不像直接引入这个Repository类，可以直接通过参数传递命名空间(比如下图红框中可以是sqlxml中的type名称)



直接写入xml里的type名称