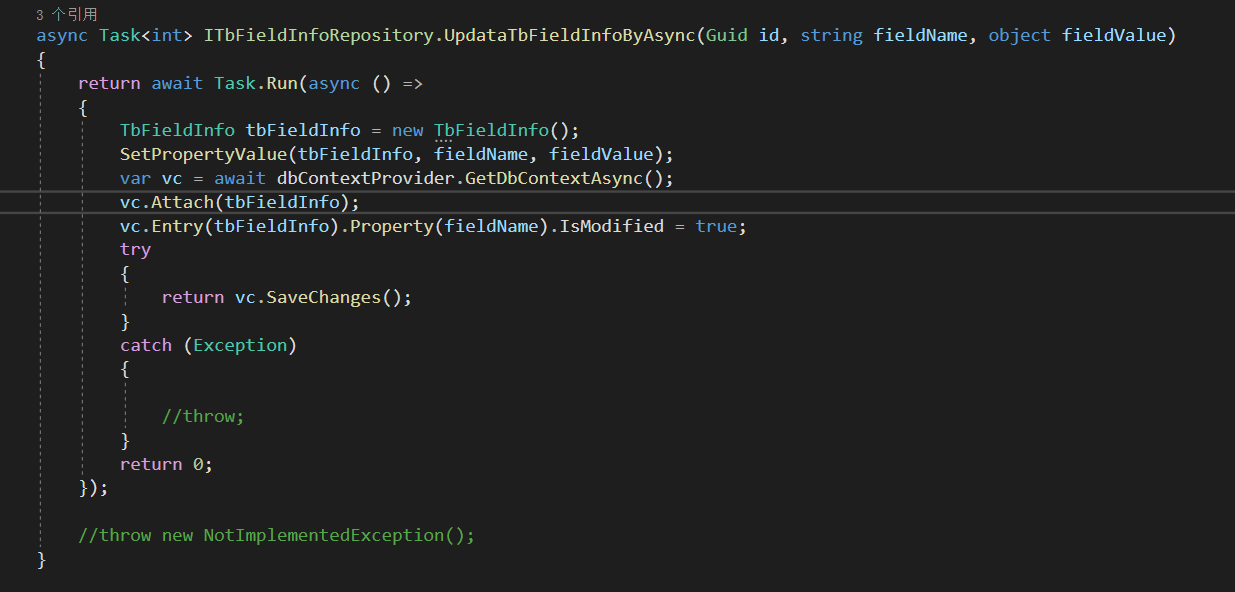
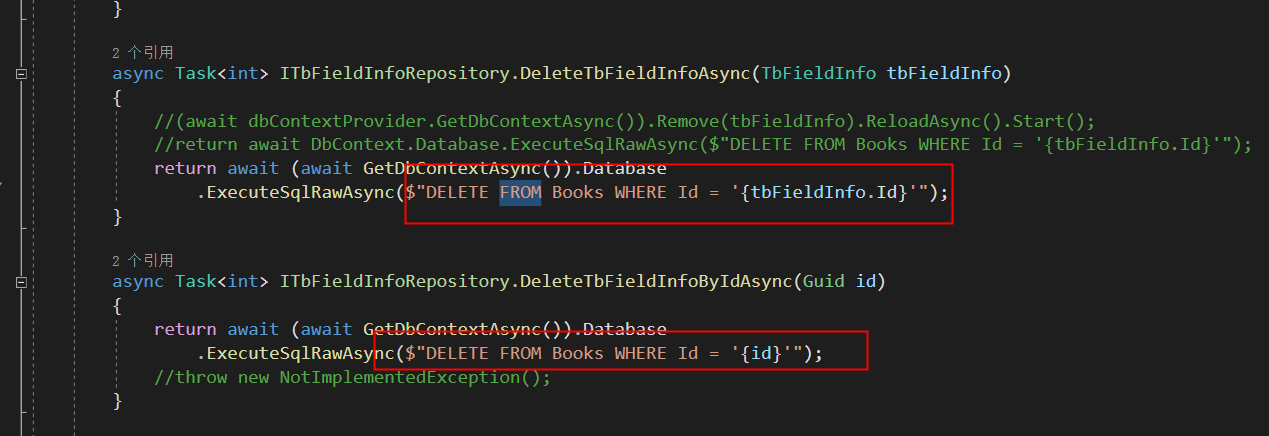


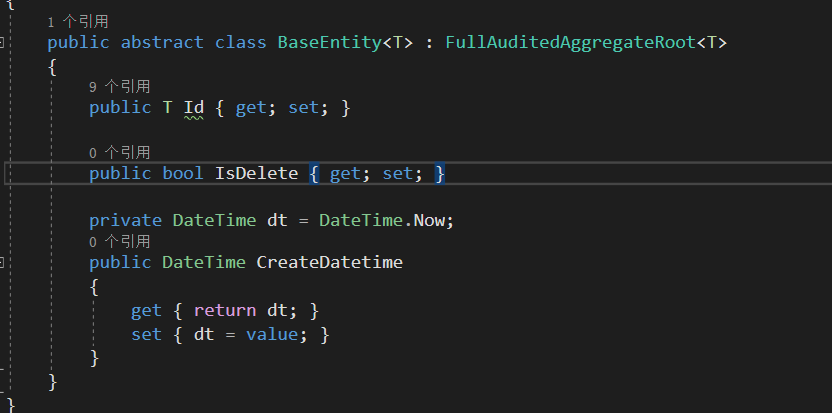
1. 账户需要用自己的昵称，不要用这种无法识别的账号



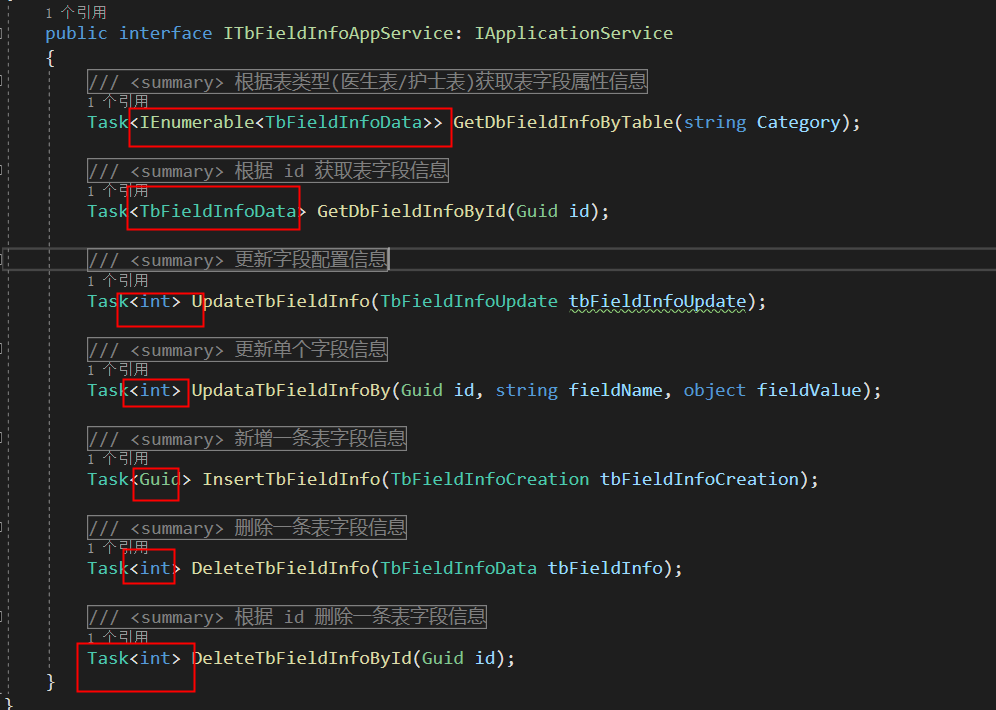
1. 仓储用EF的基本可以解决我们项目中存在的所有问题，不要扩展，这样会引入其他风险，查阅里面的方法发现这种代码快，希望不要这样操作

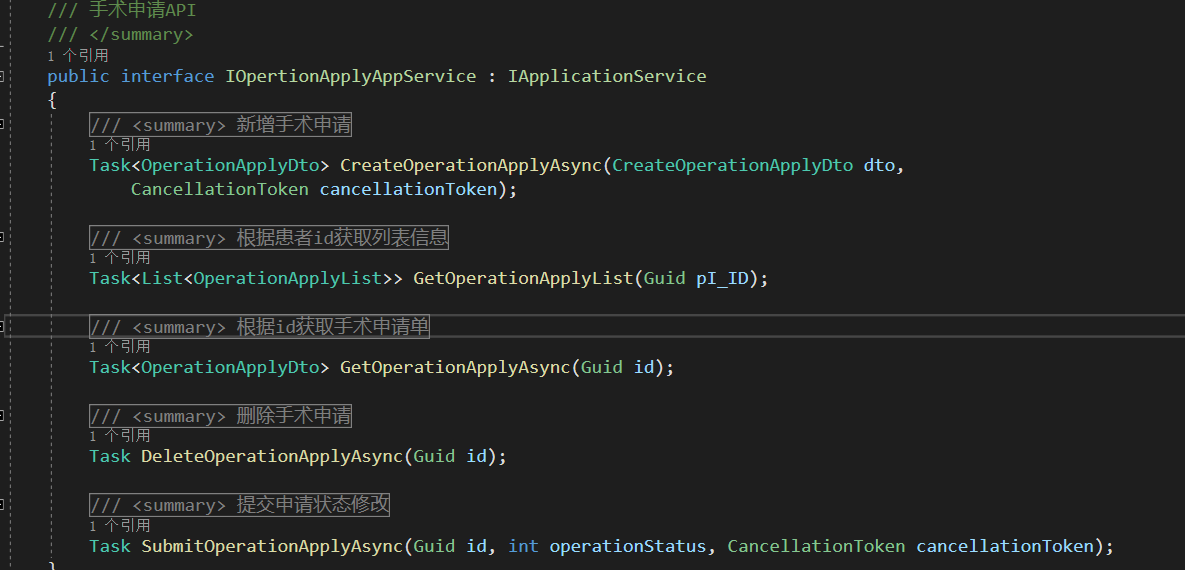


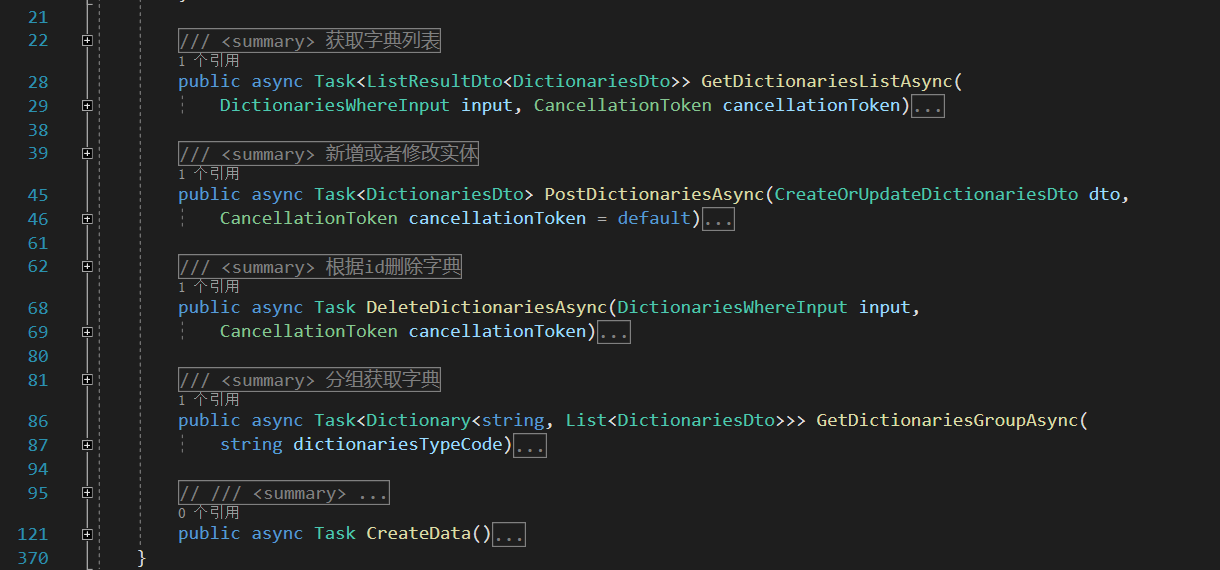
1. SQL不建议出现在项目里面，防止数据库迁移其他数据库的时候异常，可移植性不是很好。查看代码可能是简单软删除即可并无硬删除的需要。



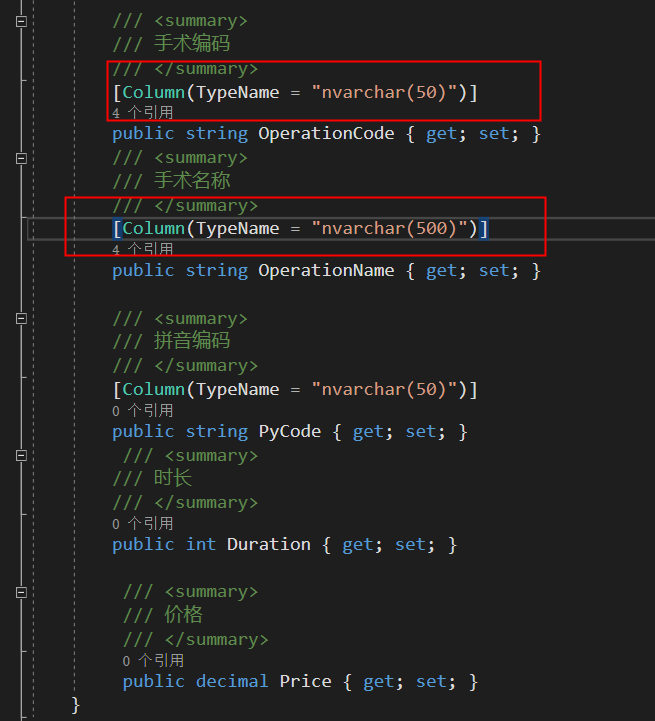
1. Id , IsDelete , CreateDatetime 在FullAuditeAggregateRoot 中都有，无需重新，再次封装意义不大







1. AppService 对外的api 需要包装一层统一的结构体，目标是提高前段开发效率，不要让前段处理各种各样的结构体，减少不必要的异常。规范化才可以提高团队的协作能力。

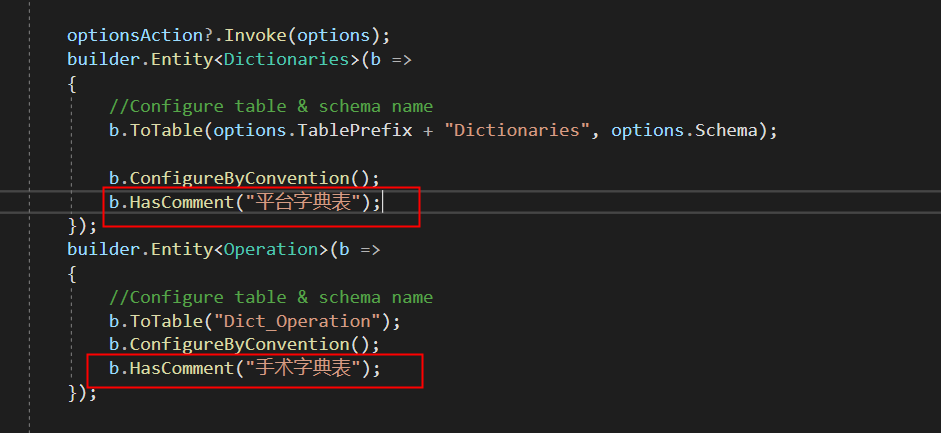


反



正

1. 表字段类型不要把类型写固定了，还是可移植性的问题，建议用 StringLength 这样的标签，字段类型交给ORM解决。

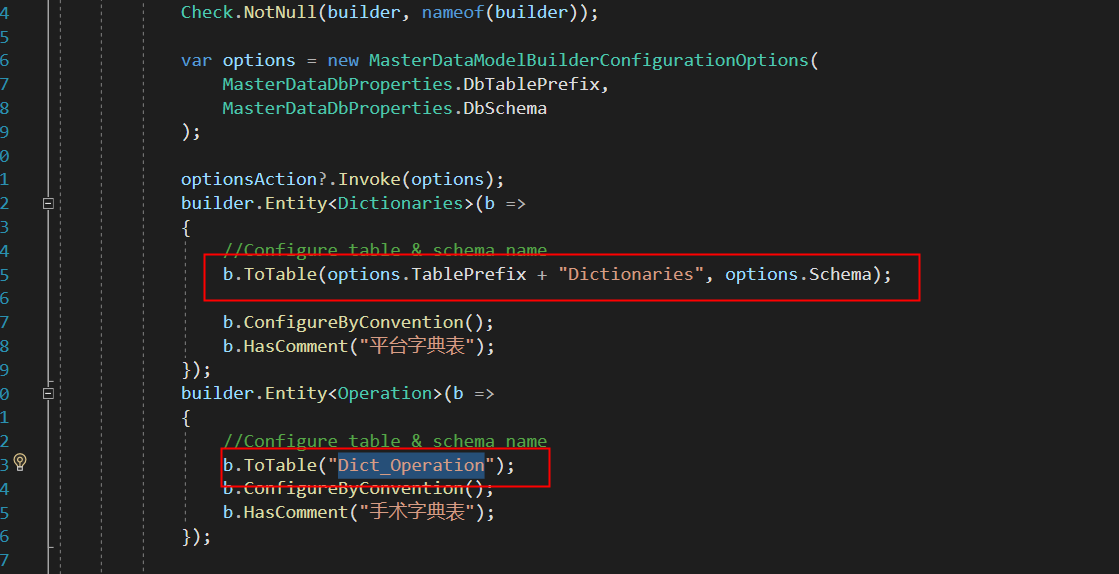


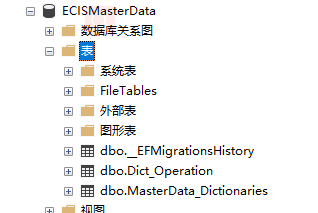
1. 表备注最好统一标在实体里面，放表里面增加可读性

引入

<PackageReference Include="Microsoft.EntityFrameworkCore.Abstractions" Version="5.0.8" />

然后 [Comment("xxx ")] 即可





1. 表前缀不一致的问题，建议统一

总结： 代码开发规范也属于代码质量的重要一环，希望大家足够重视。当代码规范统一了，团队就走向了成熟，希望大家每天都有新的进步，都有新的收获。

展望： 目前项目才开始，主要是走规范走查，未来会审逻辑甚至更细。未来，希望大家都能共同进步，交叉走查将成为可能。