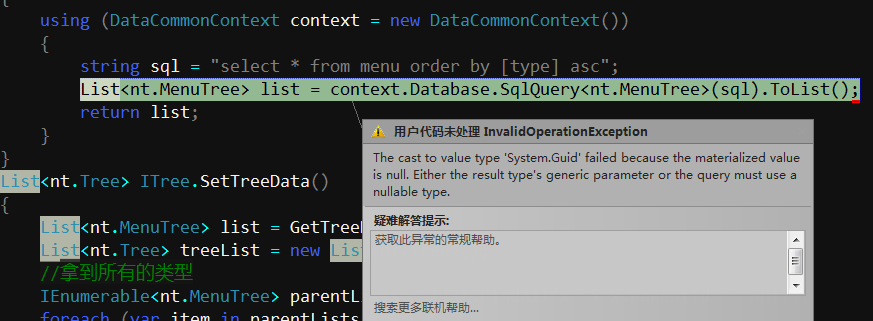
[EF-InvalidOperation系列](http://www.cnblogs.com/dunitian/p/4488224.html)

1.InvalidOperation强制转换为值类型 'System.Guid' 失败，因为实例化的值为 null。结果类型的泛型参数或查询必须使用可以为 null 的类型

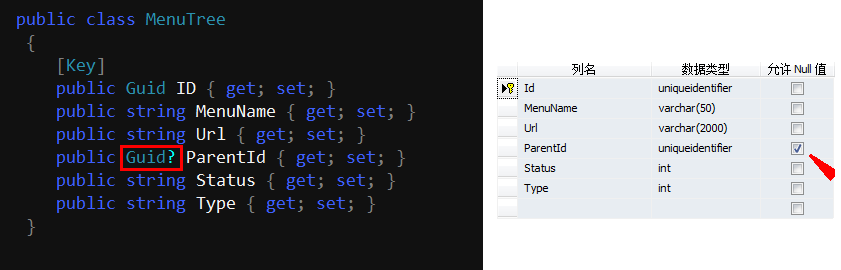


错误信息：

The cast to value type 'System.Guid' failed because the materialized value is null. Either the result type's generic parameter or the query must use a nullable type

强制转换为值类型 'System.Guid' 失败，因为实例化的值为 null。结果类型的泛型参数或查询必须使用可以为 null 的类型

解决：把guid设置为guid? 具体看图



————————————————————————————————————————————————————————————————————————————————————————————

2.【折中的解决方案】System.InvalidOperationException因为相同类型的其他实体已具有相同的主键值

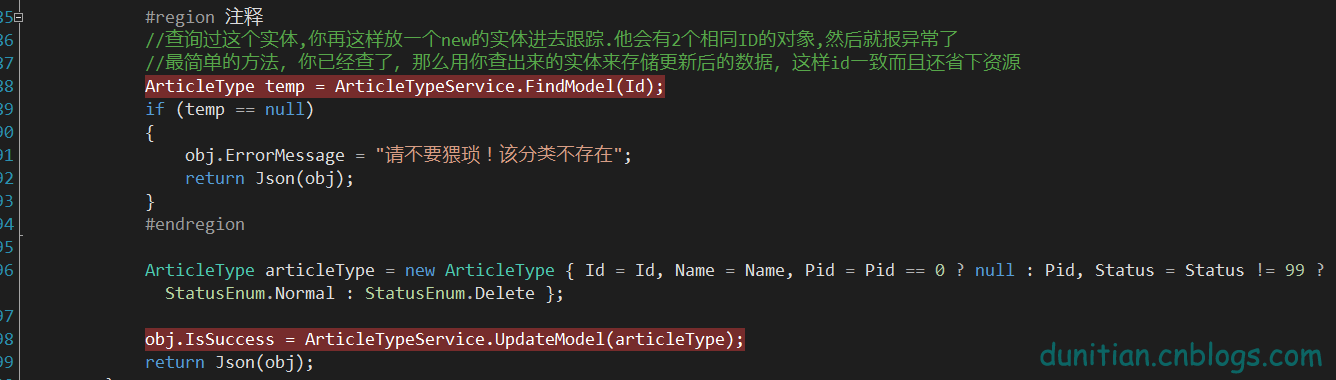
**【错误信息】：System.InvalidOperationException: 附加类型“LoT.Model.ArticleType”的实体失败，因为相同类型的其他实体已具有相同的主键值。在使用 "Attach" 方法或者将实体的状态设置为 "Unchanged" 或 "Modified" 时如果图形中的任何实体具有冲突键值，则可能会发生上述行为。这可能是因为某些实体是新的并且尚未接收数据库生成的键值。在此情况下，使用 "Add" 方法或者 "Added" 实体状态跟踪该图形，然后将非新实体的状态相应设置为 "Unchanged" 或 "Modified"。**

**【原因+解决】在更新之前通过这个id查询过Model，然后再更新了，EF不知道你到底想干什么（有两个model），所以就报错了**

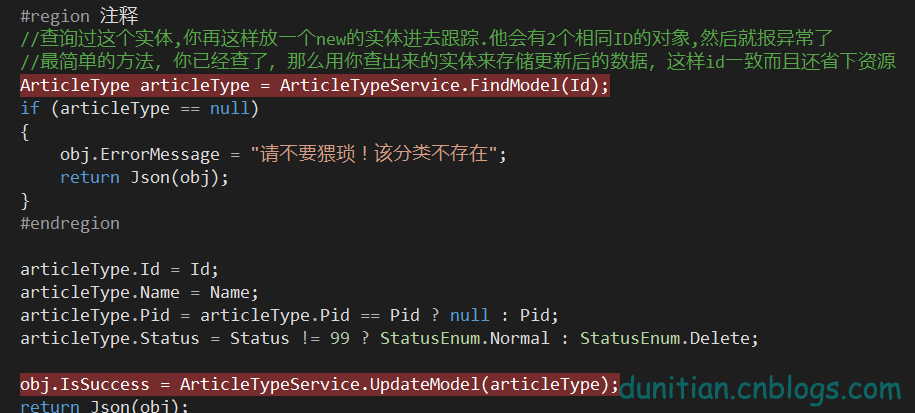
**最简单的方法，你已经查了，那么用你查出来的实体来存储更新后的数据，这样id一致而且还省下资源**

 具体看图：

原来写法：



修改后：



-------------------------Log\_Header-------------------------  
2015-03-19 00:05:00,598 [12] ERROR log - System.InvalidOperationException: 附加类型“LoT.Model.ArticleType”的实体失败，因为相同类型的其他实体已具有相同的主键值。在使用 "Attach" 方法或者将实体的状态设置为 "Unchanged" 或 "Modified" 时如果图形中的任何实体具有冲突键值，则可能会发生上述行为。这可能是因为某些实体是新的并且尚未接收数据库生成的键值。在此情况下，使用 "Add" 方法或者 "Added" 实体状态跟踪该图形，然后将非新实体的状态相应设置为 "Unchanged" 或 "Modified"。  
在 System.Data.Entity.Core.Objects.ObjectContext.VerifyRootForAdd(Boolean doAttach, String entitySetName, IEntityWrapper wrappedEntity, EntityEntry existingEntry, EntitySet& entitySet, Boolean& isNoOperation)  
在 System.Data.Entity.Core.Objects.ObjectContext.AttachTo(String entitySetName, Object entity)  
在 System.Data.Entity.Internal.Linq.InternalSet`1.<>c\_\_DisplayClass2.<Attach>b\_\_1()  
在 System.Data.Entity.Internal.Linq.InternalSet`1.ActOnSet(Action action, EntityState newState, Object entity, String methodName)  
在 System.Data.Entity.Internal.Linq.InternalSet`1.Attach(Object entity)  
在 System.Data.Entity.Internal.InternalEntityEntry.set\_State(EntityState value)  
在 System.Data.Entity.Infrastructure.DbEntityEntry`1.set\_State(EntityState value)  
在 LoT.Dal.BaseDal`1.UpdateModel(T model) 位置 c:\Users\DNT\Desktop\LoTCode\LoTBlog\LoT.Dal\BaseDal.cs:行号 77  
在 LoT.Service.BaseService`1.UpdateModel(T model) 位置 c:\Users\DNT\Desktop\LoTCode\LoTBlog\LoT.Service\BaseService.cs:行号 63  
在 LoTBlog.Back.Controllers.ArticleTypeController.Update(String Name, Nullable`1 Pid, Int32 Status, Int32 Id) 位置 c:\Users\DNT\Desktop\LoTCode\LoTBlog\LoTBlog.Back\Controllers\ArticleTypeController.cs:行号 147  
在 lambda\_method(Closure , ControllerBase , Object[] )  
在 System.Web.Mvc.ActionMethodDispatcher.Execute(ControllerBase controller, Object[] parameters)  
在 System.Web.Mvc.ReflectedActionDescriptor.Execute(ControllerContext controllerContext, IDictionary`2 parameters)  
在 System.Web.Mvc.ControllerActionInvoker.InvokeActionMethod(ControllerContext controllerContext, ActionDescriptor actionDescriptor, IDictionary`2 parameters)  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass42.<BeginInvokeSynchronousActionMethod>b\_\_41()  
在 System.Web.Mvc.Async.AsyncResultWrapper.<>c\_\_DisplayClass8`1.<BeginSynchronous>b\_\_7(IAsyncResult \_)  
在 System.Web.Mvc.Async.AsyncResultWrapper.WrappedAsyncResult`1.End()  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.EndInvokeActionMethod(IAsyncResult asyncResult)  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass37.<>c\_\_DisplayClass39.<BeginInvokeActionMethodWithFilters>b\_\_33()  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass4f.<InvokeActionMethodFilterAsynchronously>b\_\_49()  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass37.<BeginInvokeActionMethodWithFilters>b\_\_36(IAsyncResult asyncResult)  
在 System.Web.Mvc.Async.AsyncResultWrapper.WrappedAsyncResult`1.End()  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.EndInvokeActionMethodWithFilters(IAsyncResult asyncResult)  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass25.<>c\_\_DisplayClass2a.<BeginInvokeAction>b\_\_20()  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass25.<BeginInvokeAction>b\_\_22(IAsyncResult asyncResult)

-------------------------Log\_Header-------------------------  
2015-03-19 00:10:20,296 [7] ERROR log - System.InvalidOperationException: 附加类型“LoT.Model.ArticleType”的实体失败，因为相同类型的其他实体已具有相同的主键值。在使用 "Attach" 方法或者将实体的状态设置为 "Unchanged" 或 "Modified" 时如果图形中的任何实体具有冲突键值，则可能会发生上述行为。这可能是因为某些实体是新的并且尚未接收数据库生成的键值。在此情况下，使用 "Add" 方法或者 "Added" 实体状态跟踪该图形，然后将非新实体的状态相应设置为 "Unchanged" 或 "Modified"。  
在 System.Data.Entity.Core.Objects.ObjectContext.VerifyRootForAdd(Boolean doAttach, String entitySetName, IEntityWrapper wrappedEntity, EntityEntry existingEntry, EntitySet& entitySet, Boolean& isNoOperation)  
在 System.Data.Entity.Core.Objects.ObjectContext.AttachTo(String entitySetName, Object entity)  
在 System.Data.Entity.Internal.Linq.InternalSet`1.<>c\_\_DisplayClass2.<Attach>b\_\_1()  
在 System.Data.Entity.Internal.Linq.InternalSet`1.ActOnSet(Action action, EntityState newState, Object entity, String methodName)  
在 System.Data.Entity.Internal.Linq.InternalSet`1.Attach(Object entity)  
在 System.Data.Entity.Internal.InternalEntityEntry.set\_State(EntityState value)  
在 System.Data.Entity.Infrastructure.DbEntityEntry`1.set\_State(EntityState value)  
在 LoT.Dal.BaseDal`1.UpdateModel(T model) 位置 c:\Users\DNT\Desktop\LoTCode\LoTBlog\LoT.Dal\BaseDal.cs:行号 77  
在 LoT.Service.BaseService`1.UpdateModel(T model) 位置 c:\Users\DNT\Desktop\LoTCode\LoTBlog\LoT.Service\BaseService.cs:行号 63  
在 LoTBlog.Back.Controllers.ArticleTypeController.Update(String Name, Nullable`1 Pid, Int32 Status, Int32 Id) 位置 c:\Users\DNT\Desktop\LoTCode\LoTBlog\LoTBlog.Back\Controllers\ArticleTypeController.cs:行号 146  
在 lambda\_method(Closure , ControllerBase , Object[] )  
在 System.Web.Mvc.ActionMethodDispatcher.Execute(ControllerBase controller, Object[] parameters)  
在 System.Web.Mvc.ReflectedActionDescriptor.Execute(ControllerContext controllerContext, IDictionary`2 parameters)  
在 System.Web.Mvc.ControllerActionInvoker.InvokeActionMethod(ControllerContext controllerContext, ActionDescriptor actionDescriptor, IDictionary`2 parameters)  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass42.<BeginInvokeSynchronousActionMethod>b\_\_41()  
在 System.Web.Mvc.Async.AsyncResultWrapper.<>c\_\_DisplayClass8`1.<BeginSynchronous>b\_\_7(IAsyncResult \_)  
在 System.Web.Mvc.Async.AsyncResultWrapper.WrappedAsyncResult`1.End()  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.EndInvokeActionMethod(IAsyncResult asyncResult)  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass37.<>c\_\_DisplayClass39.<BeginInvokeActionMethodWithFilters>b\_\_33()  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass4f.<InvokeActionMethodFilterAsynchronously>b\_\_49()  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass37.<BeginInvokeActionMethodWithFilters>b\_\_36(IAsyncResult asyncResult)  
在 System.Web.Mvc.Async.AsyncResultWrapper.WrappedAsyncResult`1.End()  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.EndInvokeActionMethodWithFilters(IAsyncResult asyncResult)  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass25.<>c\_\_DisplayClass2a.<BeginInvokeAction>b\_\_20()  
在 System.Web.Mvc.Async.AsyncControllerActionInvoker.<>c\_\_DisplayClass25.<BeginInvokeAction>b\_\_22(IAsyncResult asyncResult)