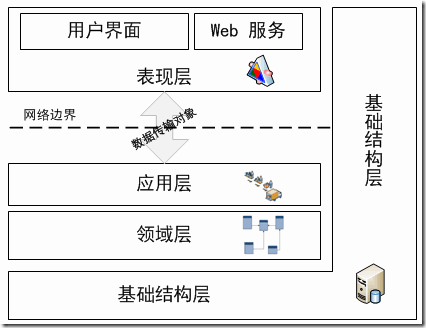
# 框架说明

框架基于Microsoft – Spain团队有一个很不错的“面向领域多层分布式项目”案例改造实现（案例在codeplex上的地址为：<http://microsoftnlayerapp.codeplex.com/>，关于此案例的分析也可参

考：<http://www.cnblogs.com/daxnet/archive/2011/03/01/1967896.html>）。

1、项目基本架构介绍 



1.1、结合项目的分层为

1.0-Presentation              表现层

1.1-Distributed Services  网络边界，数据传输

1.2-Application               应用层

1.3-Domain                     领域层

1.4-Infrastructure            基础结构层

1.0、表现层

项目定位为单服务端，多客户端的结构，有手机客户端和网站两种客户端，其中手机客户端又分为iOS，Android、WP客户端。在客户端，用户登录、注销、查看管理自己、别人定义的股票公式等操作。

1.1、数据传输

这里定义服务端跟客户端连接的接口。服务契约，服务宿主等信息。

1.2、应用层

这里定义了传输层对象Dto，以及与领域层对象Entity的转换，用Automapper实现。另外还调度和协调领域层的服务。

1.3、领域层

这里定义了领域层实体对象Entity以及领域服务。实体、聚合根、仓储接口，事务处理等。

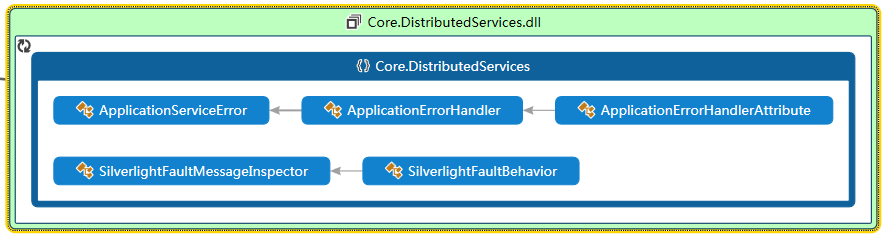
1.4、基础结构层

这里有仓储的具体实现，通过依赖注入的方式提供给领域服务调用。定义Log、加解密等在上面几个层都可能会用的与业务无关的东西。

2、代码分析

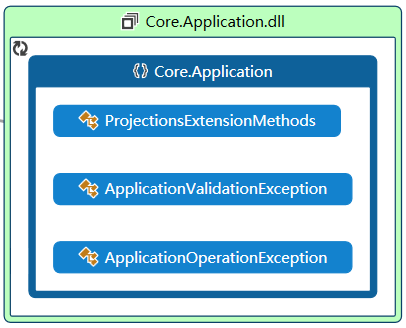
2.1、Framework.Core这个解决方案主要是对各个分层的抽象

Framework.Core的1.1-Distributed Services



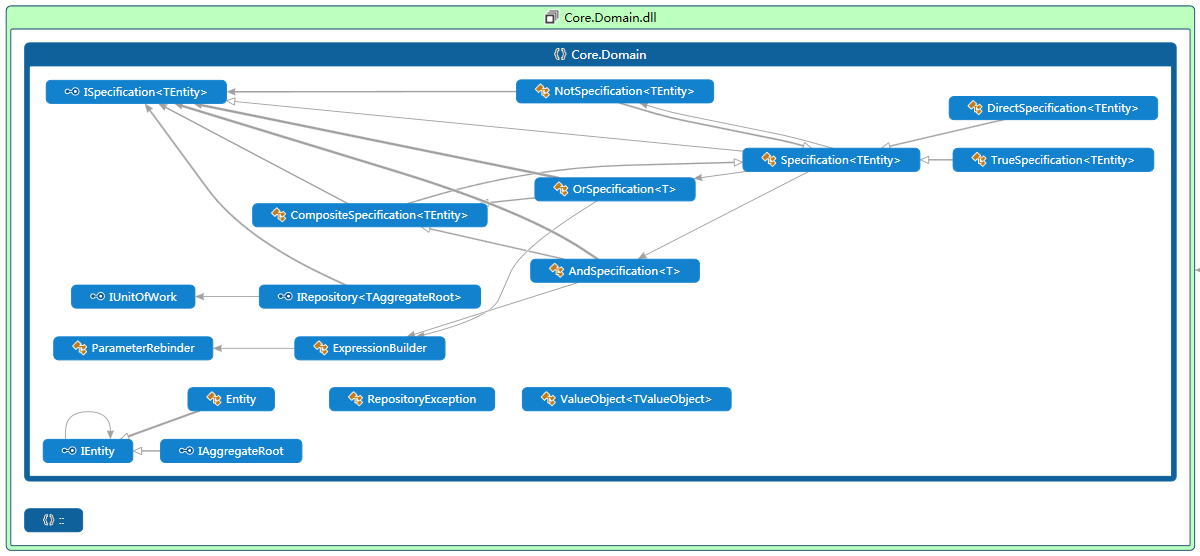
这里主要是

Framework.Core的1.2-Application



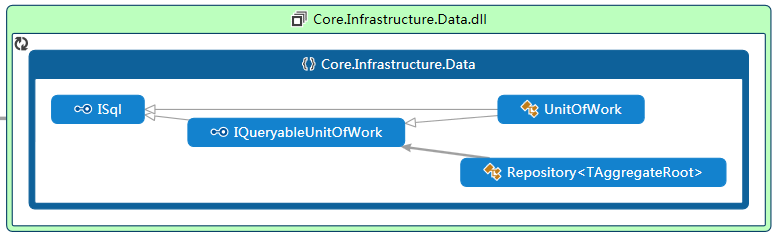
这里主要是

Framework.Core的1.3-Domain

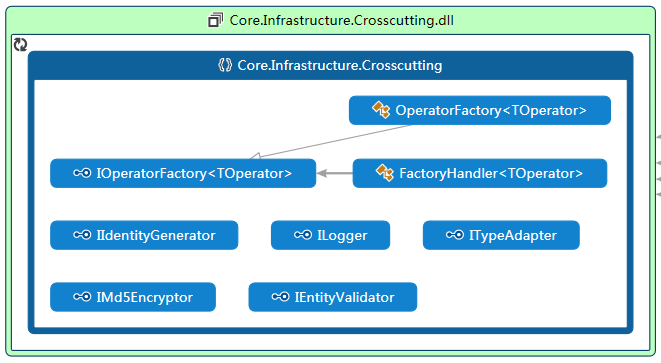


这里定义了实体接口，抽象类，聚合根，事务处理，已经查询用的多种表达式扩展

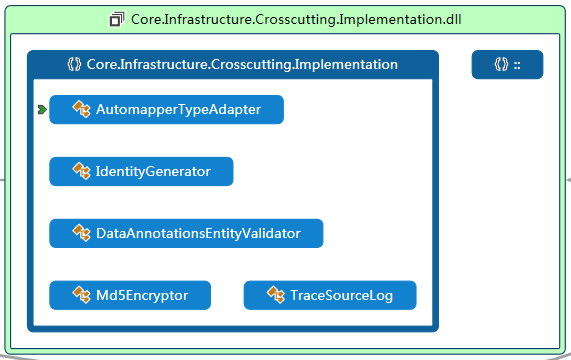
Framework.Core的1.4-Infrastructure



这个程序集主要用于数据的仓储，使用EF作为ORM框架，定义用EF进行仓储的接口以及抽象类，还有事务处理的实现。



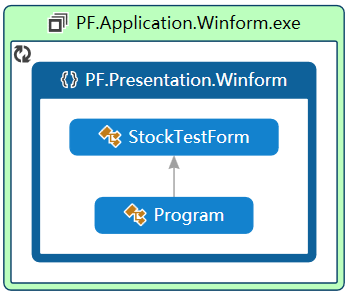
这个程序集定义了一些公共的服务接口，ID生成，Log记录，类型映射、加密，数据校验。



这个程序集是对上面公共服务接口的实现。

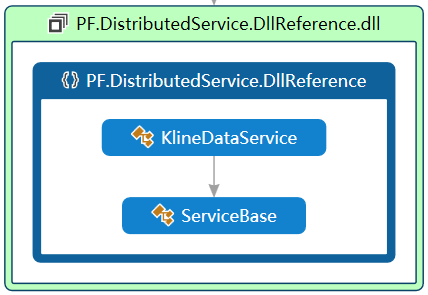
2.2、PredictFuture这个解决方案实现具体业务。解决方案现在仅将同花顺股票软件的数据解析按分层架构接入了进来。后续会有更多的业务接入进来。

PredictFuture的1.0-Presentation



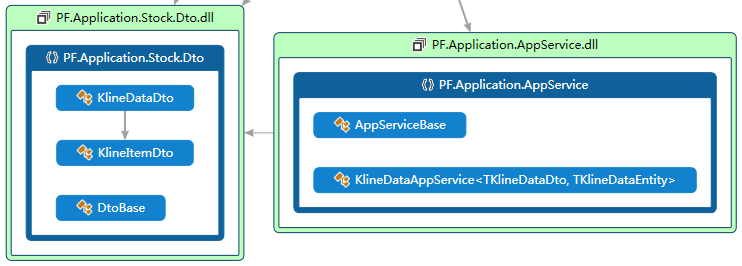
表现层的实现都在这里，现在这里仅有个用于同花顺股票数据解析的测试Form，后续Web客户端，手机客户端都将在这里管理。

PredictFuture的1.1-DistributedService



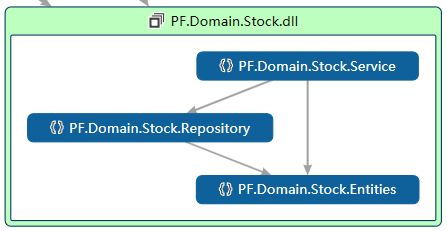
分布式服务写在这里。K线数据读取服务定义。

PredictFuture的1.2-Application



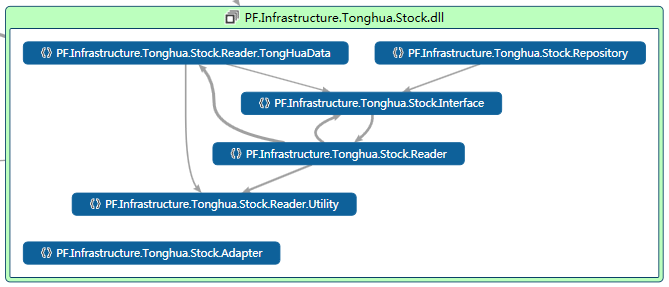
K线数据的Dto定义，领域层解析服务的调用。

PredictFuture的1.3-Domain



K线数实体定义、服务定义，仓储接口定义。

PredictFuture的1.4-Infrastructure



同花顺数据解析的具体逻辑，仓储的实现都在此。