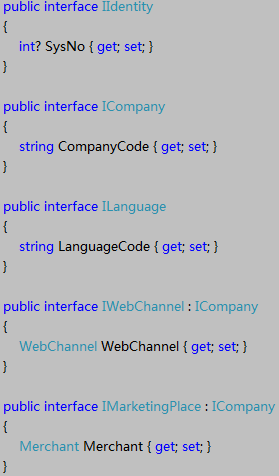
1. **概述**

在ECCentral.BizEntity的根目录下，有一个BaseInterface.cs的文件，其中定义了5个接口：



这5个接口是用来让BizEntity去继承实现的，分别具有不同的业务含义；

总的来讲，这5个接口定义出来的作用，不是为了来使用多态技术以实现系统扩展性的，也就是根本没有任何的Consumer代码来消费这5个接口，从代码层级来看，有没有这5个接口其实都一样的；

而这5个接口定义的目的，纯粹的是为了起到一个业务契约或者说业务记号的作用，实现了这些接口的BizEntity就仿佛是被打上了相应的记号，表示出某种业务设计意图，以尽量达到BizEntity代码本身就能体现出业务模型设计的业务意图的目的。

1. **接口IIdentity**

该接口是一个纯粹领域模型设计概念的接口，并不和具体的某个业务有关系；

实现该接口的BizEntity类型应该都是领域模型里的实体模型，将拥有唯一的业务标识和自己的状态，其自身内容或状态会根据某个维度（比如时间维度）的发展而变化，就仿佛一部电影，从开始到结束；而没有实现该接口的BizEntity将是领域模型里的值模型，不需要唯一的业务标识，也不会有状态的说法，或者说其只有一瞬间的状态，就像一张照片，是定格在某个瞬间的静止，对它来讲不会有变化和发展的概念；

1. **接口ICompany**
2. **业务背景**

Company在我们系统的业务概念是一个独立、完整的运营体，也就是可以独立运营的，业务完整的单元；一个Company下可以运营多个销售渠道，也可以开展Marketing Place模式；

而我们的系统定位，是需要支持多公司的业务模式，该模式会有2种比较典型的情况：

(1). 一个大的集团，可能分为多个公司来分别运营电商的情况，每个公司会有自己的商品、订单等数据，比如沃尔玛的多门店模式，在线下每个门店都是有自己独立的商品编码，所以把每个门店就视为一个独立的运营体，门店就对应到这里的Company概念；

(2). SaaS模式，在搭建一个SaaS平台时，需要同一个EC-Central的后台，可以租给不同的商家入驻来管理自己的电商后台运作，而每个商家就是一个独立的运营体，商家就对应到这里的Company概念；

1. **使用指导**

凡是需要归属于某个具体运营主体（Company）的业务模型，就需要继承自该接口，比如销售渠道、订单、库存等

1. **接口ILanguage**
2. **业务背景**

对于多语言，我们可以分为2个层面的支持：

(1). 软件产品本身的多语言支持

比如软件产品需要在多个语言的国家进行销售，那么对于使用某种语言的电商企业，软件产品可以很方便的切换为对应语言（主要就是软件界面上的各种文字信息、帮助文档等切换语言），对于该点我们需要做好软件产品内部资源文本和配置文本数据的多语言处理即可；

(2). 录入系统中的业务数据的多语言支持

比如某个电商客户，实施了EC-Central，他希望他的电商网站可以同时显示多种语言的内容，比如在东南亚的新加坡、马来等会有中文、英文作为常用语言，那么其电商网站也需要能支持显示内容为中文和英文，此时就需要在业务数据上考虑多语言的支持问题；

这里的ILanguage接口则是设计用来标识实现了该接口的业务模型，其自身的数据是需要支持多语言的；

1. **使用指导**

因为并不是所有业务数据都需要多套语言的文本展示支持，而只是部分业务数据需要，所以我们需要有选择性的添加ILanguage接口，需要识别出哪些模型需要添加ILanguage接口，可以按照如下方式来识别：

1. 由前台消费顾客录入的数据，是**不需要多语言**的，因为不可能让顾客录入多套数据
2. 由商家录入的但只是给商家内部自己看的数据，也**不需要多语言**的，因为后台运维团队识别一种语言就OK了
3. 由商家录入的并且需要发布出去给消费顾客查看的数据，就需要支持多语言了

所以只有第3)点的模型，才需要派生自ILanguage接口；

还有一种情况需要说明一下：有些模型的属性可能有LanguageCode，但是其没有继承ILanguage接口，此时则说明该模型本身的数据内容（形象一点就是数据库中该条记录的内容）不需要支持多语言，而其LanguageCode属性是有业务含义的，比如在Customer模型中的LanguageCode就表示客户最习惯的语言，在这种情况下，建议尽量将LanguageCode属性名换一个带有其业务含义的名字，比如Customer模型中的LanguageCode属性就可以改名为FavoriteLanguageCode；

1. **接口IWebChannel**
2. **业务背景**

在一个运营主体下，可能会有多个销售渠道，这些销售渠道间可能卖相同商品，也可能分别卖不同的商品，每个渠道可以定义自己的销售信息，比如商品售价、可售数量、促销活动等等；但这多个渠道的商品信息需要统一管理，多个渠道的销售单、退货单、发票等都也是统一来处理；

1. **使用指导**

因为并不是所有业务模型都需要派生自IWebChannel接口，还是需要有选择性的为某些模型添加IWebChannel接口，凡是需要或者可以按照WebChannel来划分数据和行为的模型，则需从IWebChannel接口派生；

1. **接口IMarketingPlace**
2. **业务背景**

也就是传统的Marketing Place模式（Seller Portal模式）；需要另外的Marketing Place（Seller Portal）系统的支持；

1. **使用指导**

凡是需要或者可以按照Marketing Place上的Merchant来划分数据和行为的模型，则需从IMarketingPlace接口派生；