Product price （ProductDetail页面）

如果有DisplayPrices权限，则 model.ProductPrice.HidePrices = false;

否则

model.ProductPrice.HidePrices = true;

model.ProductPrice.OldPrice = null;

model.ProductPrice.Price = null;

当product不是CustomerEntersPrice或CallForPrice时，会计算价格：

计算价格

\_taxService.GetProductPrice 计算出价格

WorkContext.TaxDisplayType 值 （IncludingTax 或 ExcludingTax）

如果taxSettings AllowCustomersToSelectTaxDisplayType 且 当前用户！=null

找genericAttributeService 中 Customer.TaxDisplayTypeId 属性的值

否则返回taxSettings.TaxDisplayType（默认为ExcludingTax）

decimal taxRate;

decimal oldPriceBase = \_taxService.GetProductPrice(product, product.OldPrice, out taxRate);

decimal finalPriceWithoutDiscountBase = \_taxService.GetProductPrice(product, \_priceCalculationService.GetFinalPrice(product, \_workContext.CurrentCustomer, includeDiscounts: false), out taxRate);

decimal finalPriceWithDiscountBase = \_taxService.GetProductPrice(product, \_priceCalculationService.GetFinalPrice(product, \_workContext.CurrentCustomer, includeDiscounts: true), out taxRate);

VAT(增值税) 计算

TaxService.CreateCalculateTaxRequest

VAT 可以基于BillingAddress， ShippingAddress， DefaultAddress计算

默认 taxSettings.**TaxBasedOn中为**BillingAddress

如果EU VAT Eupore Union Value Added Tax 启用：

//new EU VAT rules starting January 1st 2015

//find more info at http://ec.europa.eu/taxation\_customs/taxation/vat/how\_vat\_works/telecom/index\_en.htm#new\_rules

//EU VAT enabled?

if (\_taxSettings.EuVatEnabled)

{

//telecommunications, broadcasting and electronic services?

if (product != null && product.IsTelecommunicationsOrBroadcastingOrElectronicServices)

{

//January 1st 2015 passed?

if (DateTime.UtcNow > new DateTime(2015, 1, 1, 0, 0, 0, DateTimeKind.Utc))

{

//Europe Union consumer?

if (IsEuConsumer(customer))

{

//We must charge VAT in the EU country where the customer belongs (not where the business is based)

basedOn = TaxBasedOn.BillingAddress;

}

}

}

}

PriceCalculationService计算产品价格，includeDiscounts参数可以计算有没有折扣下的价格：

public virtual decimal GetFinalPrice(Product product,

Customer customer,

decimal additionalCharge = decimal.Zero,

bool includeDiscounts = true,

int quantity = 1)