Media

图片保存在~/content/images， 图片文件名为7位长度 picture.Id.ToString("0000000")

**0000005\_electronics.jpeg** (picture有seoname) **0000005.jpeg**

缩略图保存地址: ~/content/images/thumbs (参考GetThumbLocalPath方法)

**0000005\_electronics\_450.jpeg** **0000005\_450.jpeg**

如果mediaSettings.MultipleThumbDirectories 为true， 保存路径为:~/content/images/thumbs/[**000] :** picture.Id.ToString("0000000") 前三位

protected virtual string **GetThumbLocalPath**(string thumbFileName)

{

var thumbsDirectoryPath = CommonHelper.MapPath("~/content/images/thumbs");

if (\_mediaSettings.MultipleThumbDirectories)

{

//get the first two letters of the file name

var fileNameWithoutExtension = Path.GetFileNameWithoutExtension(thumbFileName);

if (fileNameWithoutExtension != null && fileNameWithoutExtension.Length > MULTIPLE\_THUMB\_DIRECTORIES\_LENGTH)

{

var subDirectoryName = fileNameWithoutExtension.Substring(0, MULTIPLE\_THUMB\_DIRECTORIES\_LENGTH); // 当前长度值为3

thumbsDirectoryPath = Path.Combine(thumbsDirectoryPath, subDirectoryName);

if (!System.IO.Directory.Exists(thumbsDirectoryPath))

{

System.IO.Directory.CreateDirectory(thumbsDirectoryPath);

}

}

}

var thumbFilePath = Path.Combine(thumbsDirectoryPath, thumbFileName);

return thumbFilePath;

}

Product Detail 页面对图片的处理：

参数 isAssociatedProduct 如果为true， 该产品是grouped 产品 不能单独显示， 会跳转到ParentGroupedProductId 对应的茶品

默认图片是否能点击放大（js popup 插件）

model.DefaultPictureZoomEnabled = \_mediaSettings.DefaultPictureZoomEnabled

\_mediaSettings.AssociatedProductPictureSize： **220**

\_mediaSettings.ProductDetailsPictureSize ： **550**

//default picture

var defaultPictureSize = isAssociatedProduct ?

\_mediaSettings.AssociatedProductPictureSize :

\_mediaSettings.ProductDetailsPictureSize;

读取product的产品的图片时 ， 第一张图片会当作默认图片来显示

// picture product\_Picture 两个表关联查询DisplayOrder排序

var pictures = \_pictureService.**GetPicturesByProductId**(product.Id);

var defaultPicture = pictures.FirstOrDefault();

var defaultPictureModel = new PictureModel

{

ImageUrl = \_pictureService.**GetPictureUrl**(defaultPicture, defaultPictureSize, !isAssociatedProduct),

FullSizeImageUrl = \_pictureService.**GetPictureUrl**(defaultPicture, 0, !isAssociatedProduct),

Title = string.Format(\_localizationService.GetResource("Media.Product.ImageLinkTitleFormat.Details"), model.Name),

AlternateText = string.Format(\_localizationService.GetResource("Media.Product.ImageAlternateTextFormat.Details"), model.Name),

};

Model中

ImageUrl： 为 defaultPictureSize 大小的图片 **~/content/images/thumbs** 中寻找， 没有则根据原始图片生成target大小的图片

FullSizeImageUrl： 为 原始图片大小， **~/content/images**/ 中寻找

Libraries\Nop.Services\Media\PictureService.cs

pictureService.**GetPictureUrl**

public virtual string GetPictureUrl(Picture picture,

int targetSize = 0,

bool showDefaultPicture = true,

string storeLocation = null,

PictureType defaultPictureType = PictureType.Entity)

如果图片entity没有对应的binary， 第三个参数showDefaultPicture是否显示默认的一张图片， 如果是grouped 就不显示， single product则显示

GetDefaultPictureUrl ：

switch (defaultPictureType)

{

case PictureType.Avatar:

defaultImageFileName = \_settingService.GetSettingByKey("Media.Customer.DefaultAvatarImageName", "default-avatar.jpg");

break;

case PictureType.Entity:

default:

defaultImageFileName = \_settingService.GetSettingByKey("Media.DefaultImageName", "default-image.png");

break;

}

如果PictureType是Entity, 默认返回图片~/content/images/[default-image.png]

如果PictureType是Entity, 默认返回图片~/content/images/[default-avatar.png]

targetSize == 0， 直接返回图片地址：

(!String.IsNullOrEmpty(storeLocation)

? storeLocation

: \_webHelper.GetStoreLocation())

+ "content/images/" + defaultImageFileName;

否则， 返回targetSize的图片

byte[] pictureBinary = null;

if (picture != null)

pictureBinary = LoadPictureBinary(picture);

if (picture == null || pictureBinary == null || pictureBinary.Length == 0)

{

if (showDefaultPicture)

{

url = GetDefaultPictureUrl(targetSize, defaultPictureType, storeLocation);

}

return url;

}