# FeatureedProducts

IPagedList<Product> featuredProducts = null;

featuredProducts = \_productService.SearchProducts(

categoryIds: new List<int> { category.Id },

storeId: \_storeContext.CurrentStore.Id,

visibleIndividuallyOnly: true,

featuredProducts: true);

hasFeaturedProductsCache = featuredProducts.TotalCount > 0;

Category 下的Product

//products

IList<int> alreadyFilteredSpecOptionIds = model.PagingFilteringContext.SpecificationFilter.GetAlreadyFilteredSpecOptionIds(\_webHelper);

IList<int> filterableSpecificationAttributeOptionIds;

var products = \_productService.SearchProducts(out filterableSpecificationAttributeOptionIds,

true,

categoryIds: categoryIds,

storeId: \_storeContext.CurrentStore.Id,

visibleIndividuallyOnly: true,

featuredProducts: \_catalogSettings.IncludeFeaturedProductsInNormalLists ? null : (bool?)false,

priceMin: minPriceConverted,

priceMax: maxPriceConverted,

filteredSpecs: alreadyFilteredSpecOptionIds,

orderBy: (ProductSortingEnum)command.OrderBy,

pageIndex: command.PageNumber - 1,

pageSize: command.PageSize);

model.Products = PrepareProductOverviewModels(products).ToList();

//specs

model.PagingFilteringContext.SpecificationFilter.PrepareSpecsFilters(alreadyFilteredSpecOptionIds,

filterableSpecificationAttributeOptionIds != null ? filterableSpecificationAttributeOptionIds.ToArray() : null,

\_specificationAttributeService,

\_webHelper,

\_workContext,

\_cacheManager);