1. Home Page

PushDataToClient()

Call "ProductController" to get home page data "GetHomePageData" to return "Featured Products (ProductCurrency)"

2. Product Deatail Page

BindData()

Call "ProductController" to get product detail data "GetProductDetailData" to return "Product Detail (ProductCurrency), Product's category detail (procategory), Store's categories (CategoryProduct)"

3. WebSiteUrl

Web.config

Backend: Helper.GetHostURL()

Frontend:

var MasterPageDataJson = $('\*[id\*=MasterPageDataDiv]')[0].innerHTML;

if (MasterPageDataJson) {

var masterPageData = jQuery.parseJSON(MasterPageDataJson);

currentHostUrl = masterPageData.HostUrl;

}

4. ResultCode: This will help to show alert message for every task on client side.

- It has ResultCode.cs in App\_Code

- Each result code has its message and code as ID

- To get all result code, call method GetResultSet()

- To use, call ResultCode.Value[0] to get the code, and call ResultCode.Value[1] to get the message. Eg: *model.ResultCode.Value[0]*

5. Show aller message on client:

- Call method ShowMessage in Javascript with parameter ResultCode. Eg: ShowMessage(model.ResultCode);

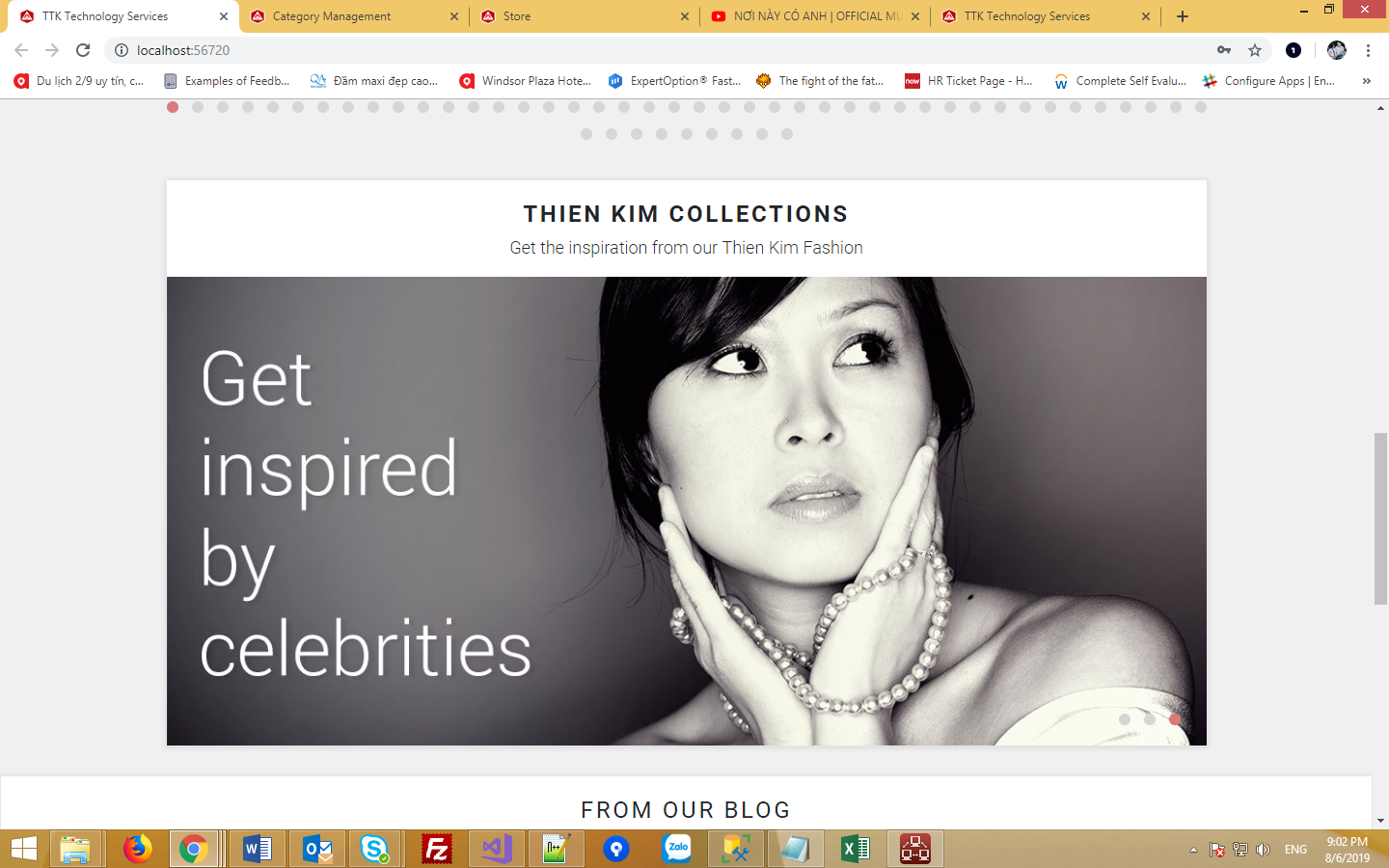
- If message is available when call ShowMessage, it will be show by default. Eg: ShowMessage(model.ResultCode, message);

6. How to make a menu item link to a store?

- Enter /admin/category.aspx to create a new category.

- Set Publish = true; Is Menu = true; Category URL = “/category.aspx?store\_id=2” (Store\_id is the id of store)

7. How to change home page collection?



Home page collection is a category that has property is\_collection = true.

It link to a store by category\_url = /category.aspx?store\_id=3

8. Add product to a basket

* Condition to add product to basket:
  + Product to add must exist, and user must be logged in
  + product\_id > 0 && userId != null
* Find Product by Id
* Find Category by Product Id
* Find order by User Id and order status **new** (We only add new order if no “**new”** order has found – User’s orders have different status, but only one “**new”** order is accepted to be existed.
* Find order’s items by Order Id
  + Update founded order item with following options:

item.quantity = item.quantity + 1;

item.list\_price = product.list\_price;

item.shaipping\_fee = product.shipping\_fee;

item.free\_shipping\_amount = product.free\_shipping\_amount;

item.tax = product.tax;

item.taxed\_countries = product.taxed\_countries;

* + Add existing order item with following options:

order\_id = order.order\_id,

product\_id = product\_id,

quantity = 1,

list\_price = product.list\_price,

shipping\_fee = product.shipping\_fee,

free\_shipping\_amount = product.free\_shipping\_amount,

tax = product.tax,

taxed\_countries = product.taxed\_countries

* If we cannot find order with status “new” and belong to the user, then we add new order and add new product to order:
  + New order with following options:

customer\_id = userId,

order\_status = (int)OrderStatus.Status.New,

order\_date = DateTime.UtcNow,

required\_date = DateTime.UtcNow,

store\_id = category.store\_id == null ? 0 : category.store\_id.Value

* + New order’s item which has new added order id and product id:

order\_id = new\_order.order\_id,

product\_id = product\_id,

quantity = 1,

list\_price = product.list\_price,

shipping\_fee = product.shipping\_fee,

free\_shipping\_amount = product.free\_shipping\_amount,

tax = product.tax,

taxed\_countries = product.taxed\_countries

8. Get basket for customer only

* Request URL: <http://localhost:56720/basket.aspx>
* Note: If the request has product id, then it will add a new product to order and get basket detail.
* Condition to get order for basket:
  + Order must have status “New” and Customer Id = Logged User Id
  + Get Order’s items by the same condition, joined with table order and then check for status “New” and Customer Id = Logged User Id
* After getting order and order’s items, we have these calculation for order total:
  + For loop all order’s items
    - Currency of all order’s items must be the same, so I only take one order’s item currency to be used as order’s currency code: currencyCode = order\_item.currency\_code;
    - Discount will be available for each order’s item (It will be insert or update via function UpdateOrderDiscount). It will effect total prize of each order’s item in a for-loop all order’s items.
    - Tax, each order has its own tax. But it has to compare the customer’s country with the current list of countries with item record, just for which are apply to be charged. If no country from the list similar to the customer’s country, then default tax from web.config will be applied.
    - Shipping fee: If the amount of product is higher than “free shipping amount” column of item’s record, it will equal to zero. In case the “free shipping amount = 0” the shipping fee will be applied no matterwhat.
    - Total prize for each item will be calculated as: totalPrice = totalPrice - (double)discount;
    - shippingTotal will be plus for each item.
    - Tax total will be plus for each item taxTotal += (totalPrice \* tax) / 100;
    - Order total will be also plus for each item
    - totalItem will be also plus for each item
  + grandTotal = orderTotal + shippingTotal + taxTotal;
  + The order its self also has its own discount. Thus, if the order’s discount greater than zero, it will be also applied for the grand total (In reallity, if the the order’s disccount is greater than zero, the discount for each order’s item won’t be applied). Also, if the order\_discount\_is\_fixed = true, the discount will be used as the fixed prize but not in percentage:

if (order\_discount > 0)

{

bool order\_discount\_is\_fixed = model.Order.order\_discount\_is\_fixed == null ? true : (bool)model.Order.order\_discount\_is\_fixed;

if (order\_discount\_is\_fixed)

{

grandTotal = grandTotal - order\_discount;

}

else

{

grandTotal = grandTotal - (grandTotal \* order\_discount / 100);

}

}

8. Apply coupon for customer

* URL: <http://localhost:56720/update-basket.aspx?coupon_code=abc&order_id=1>
* Only order with status “New” and Order ID = Requested Order Id is updated with coupon.
* Note: One order accept only one coupon. If new coupon applied, the old one will be remove.
* Function UpdateOrderDiscount
* Look for existed voucher: var voucher = context.vouchers.SingleOrDefault(w => w.voucher\_code == couponCode);
* Below are the field and comment for voucher fields:

//The number of uses currently

int voucher\_uses = 0;

Int32.TryParse(voucher.voucher\_uses + "", out voucher\_uses);

//The max uses this voucher has

int voucher\_max\_uses = 0;

Int32.TryParse(voucher.voucher\_max\_uses + "", out voucher\_max\_uses);

//How many times a user can use this voucher.

int voucher\_max\_uses\_user = 0;

Int32.TryParse(voucher.voucher\_max\_uses\_user + "", out voucher\_max\_uses\_user);

//The type can be: voucher, discount, sale. What ever you want.

VoucherType.Type voucher\_type = voucher.voucher\_type == null? VoucherType.Type.Voucher : (VoucherType.Type)voucher.voucher\_type;

//The amount to discount by (in pennies) in this example.

int voucher\_discount\_amount = 0;

Int32.TryParse(voucher.voucher\_discount\_amount + "", out voucher\_discount\_amount);

//Whether or not the voucher is a percentage or a fixed price.

bool voucher\_is\_fixed = false;

bool.TryParse(voucher.voucher\_is\_fixed + "", out voucher\_is\_fixed);

//When the voucher begins

DateTime voucher\_starts\_at = DateTime.UtcNow;

DateTime.TryParse(voucher.voucher\_starts\_at + "", out voucher\_starts\_at);

//When the voucher ends

DateTime voucher\_expires\_at = DateTime.UtcNow;

DateTime.TryParse(voucher.voucher\_expires\_at + "", out voucher\_expires\_at);

//Product id this voucher apply to

int voucher\_product\_id = 0;

Int32.TryParse(voucher.voucher\_product\_id + "", out voucher\_product\_id);

//User id this voucher apply to

string voucher\_user\_id = voucher.voucher\_user\_id;

* Condition for update order with discount:
  + voucher\_discount\_amount > 0 && voucher\_max\_uses > 0 && voucher\_expires\_at > voucher\_starts\_at && voucher\_starts\_at <= DateTime.UtcNow && voucher\_expires\_at >= DateTime.UtcNow
  + If voucher has product ID, then it will update the order’s items’ discount that has order = voucher product ID:
  + If voucher has no product ID but has customer ID, then it will update order’s dicount if current logged in user id = voucher customer ID
  + If the voucher doesn’t have product Id, customer ID, then it will update the order’s discount eventually nomatterwhat.