READ JSON COUPON DATA WITH CHECKING TWO DATA LIST WITH COUPON ID AND TOTAL DISCOUNT AND ITEM OBJECTS USING ADO.NET

FolderName: JSONFiles

fileName: Applied.json

{

"NewApplied": [

{

"couponId": "9060",

"externalId": "9060",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

},

{

"lineId": "8",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.6,

"reducesTax": true

},

{

"couponId": "9061",

"externalId": "9061",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

},

{

"lineId": "8",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9062",

"externalId": "9062",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

},

{

"lineId": "8",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9064",

"externalId": "9064",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

},

{

"lineId": "8",

"discount": 0,

"quantity": 1

},

{

"lineId": "9",

"discount": 0,

"quantity": 1

},

{

"lineId": "10",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9065",

"externalId": "9065",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9066",

"externalId": "9066",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9067",

"externalId": "9067",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "5",

"discount": 0,

"quantity": 1

},

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9068",

"externalId": "9068",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "1",

"discount": 0,

"quantity": 1

},

{

"lineId": "2",

"discount": 0,

"quantity": 1

},

{

"lineId": "3",

"discount": 0,

"quantity": 1

},

{

"lineId": "4",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

}

],

"OldApplied": [

{

"couponId": "9060",

"externalId": "9060",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

},

{

"lineId": "8",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.6,

"reducesTax": true

},

{

"couponId": "9061",

"externalId": "9061",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

},

{

"lineId": "8",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.6,

"reducesTax": true

},

{

"couponId": "9063",

"externalId": "9063",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

},

{

"lineId": "8",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9064",

"externalId": "9064",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

},

{

"lineId": "8",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9065",

"externalId": "9065",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

},

{

"lineId": "8",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9066",

"externalId": "9066",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9067",

"externalId": "9067",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "4",

"discount": 0,

"quantity": 1

},

{

"lineId": "5",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

},

{

"couponId": "9068",

"externalId": "9068",

"receiptAlias": "$0.20 OFF Body Armor 16oz Bottles",

"items": [

{

"lineId": "5",

"discount": 0,

"quantity": 1

},

{

"lineId": "6",

"discount": 0,

"quantity": 1

},

{

"lineId": "7",

"discount": 0,

"quantity": 1

}

],

"totalDiscount": 0.7,

"reducesTax": true

}

]

}

Models:

Coupon.cs:

using JWTRoleAuthentication.CommonLayer.Models;

namespace JWTRoleAuthentication.CommonLayer.Models

{

public class Coupon

{

public string CouponId { get; set; }

public string ExternalId { get; set; }

public string ReceiptAlias { get; set; }

public List<Item> Items { get; set; }

public double TotalDiscount { get; set; }

public bool ReducesTax { get; set; }

//public override bool Equals(object? obj)

//{

// if (obj is Coupon coupon)

// {

// return CouponId == coupon.CouponId &&

// TotalDiscount == coupon.TotalDiscount &&

// Items.SequenceEqual(coupon.Items);

// }

// return false;

//}

//public override int GetHashCode()

//{

// return HashCode.Combine(CouponId, TotalDiscount, Items);

//}

}

public class Item

{

public string LineId { get; set; }

public double Discount { get; set; }

public int Quantity { get; set; }

public override bool Equals(object obj)

{

return obj is Item item &&

LineId == item.LineId &&

Discount == item.Discount &&

Quantity == item.Quantity;

}

public override int GetHashCode()

{

return HashCode.Combine(LineId, Discount, Quantity);

}

}

public class CouponResponse

{

public List<Coupon> Applied { get; set; }

public string ErrorCode { get; set; }

public string ErrorDesc { get; set; }

public string Status { get; set; }

}

public class AppliedData

{

public List<Coupon> NewApplied { get; set; }

public List<Coupon> OldApplied { get; set; }

}

}

public class AppliedCouponComparer : IEqualityComparer<Coupon>

{

public bool Equals(Coupon x, Coupon y)

{

if (x is null || y is null) return false;

return x.CouponId == y.CouponId &&

x.TotalDiscount == y.TotalDiscount

&& x.Items.SequenceEqual(y.Items);

}

//public int GetHashCode(Coupon obj)

//{

// if (obj == null) return 0;

// //return (obj.CouponId + obj.TotalDiscount).GetHashCode();

// return (obj.CouponId + obj.TotalDiscount + obj.Items).GetHashCode();

//}

public int GetHashCode(Coupon obj)

{

if (obj == null) return 0;

int hashCouponId = obj.CouponId == null ? 0 : obj.CouponId.GetHashCode();

int hashTotalDiscount = obj.TotalDiscount.GetHashCode();

int hashItems = obj.Items == null ? 0 : obj.Items.Aggregate(0, (hash, item) => hash ^ item.GetHashCode());

//return hashCouponId ^ hashTotalDiscount;

return hashCouponId ^ hashTotalDiscount ^ hashItems;

}

}  
  
  
Repo.cs:

public async Task<CouponResponse> GetCouponsData()

{

CouponResponse response = new CouponResponse();

try

{

var json = await File.ReadAllTextAsync("D:\\csharpprojects\\JWTRoleAuthentication\\JWTRoleAuthentication\\JSONFiles\\Applied.json");

var dataResult = JsonConvert.DeserializeObject<AppliedData>(json);

var newAppliedList = dataResult?.NewApplied.ToList();

var oldAppliedList = dataResult?.OldApplied.ToList();

if(newAppliedList != null && oldAppliedList != null && dataResult != null)

{

// var nonMatchedItemForNewApplied = newAppliedList.Except(oldAppliedList).ToList();

// var nonMatchedItemsForOldApplied = oldAppliedList.Except(newAppliedList).ToList();

//var nonMatchedItemForNewApplied = newAppliedList.Where(newItem => !oldAppliedList.Any(oldItem => newItem.CouponId == oldItem.CouponId && newItem.TotalDiscount == oldItem.TotalDiscount && newItem.Items.SequenceEqual(oldItem.Items))).ToList();

//var nonMatchedItemsForOldApplied = oldAppliedList.Where(oldItem => !newAppliedList.Any(newItem => oldItem.CouponId == newItem.CouponId && oldItem.TotalDiscount == newItem.TotalDiscount && oldItem.Items.SequenceEqual(newItem.Items))).ToList();

// find non-matched items for NewApplied:

var nonMatchedItemForNewApplied = newAppliedList.Except(oldAppliedList, comparer: new AppliedCouponComparer()).ToList();

// find non-matched items for OldApplied:

var nonMatchedItemsForOldApplied = oldAppliedList.Except(newAppliedList,comparer: new AppliedCouponComparer()).ToList();

var updateOldAppliedListWithTotalDiscount = nonMatchedItemsForOldApplied.Select(x =>

{

x.TotalDiscount = -x.TotalDiscount;

return x;

}).ToList();

// combined together:

var finalResultList = nonMatchedItemForNewApplied.Concat(nonMatchedItemsForOldApplied).ToList();

//// find out max Items objects for each couponId and TotalDiscount.

//var maxItemForNonMatchedItems = finalResultList

// .GroupBy(x => new {x.CouponId, x.TotalDiscount,x.Items})

// .SelectMany(g => g.Where(x => x.Items.Count == g.Max(y => y.Items.Count)))

// .ToList();

//// Add - sign for oldApplied TotalDiscount:

//var updateOldAppliedListWithTotalDiscount = nonMatchedItemsForOldApplied.Select(x =>

//{

// x.TotalDiscount = -x.TotalDiscount;

// return x;

//}).ToList();

//// Update final result with oldApplied list with - sign:

//var finalResultList = maxItemForNonMatchedItems.Select(item =>

//{

// var matchedOldItem = updateOldAppliedListWithTotalDiscount

// .FirstOrDefault(oldItem => oldItem.CouponId == item.CouponId && oldItem.TotalDiscount == -item.TotalDiscount);

// if(matchedOldItem != null)

// {

// item.TotalDiscount = -matchedOldItem.TotalDiscount;

// }

// return item;

//}).ToList();

// select final NewApplied list array from final result:

// var maxItemsForNewApplied = nonMatchedItemForNewApplied.Where(item => finalResultList.Any(newItem => newItem.CouponId == item.CouponId && newItem.TotalDiscount == item.TotalDiscount && newItem.Items.SequenceEqual(item.Items) )).ToList() ;

// select final oldApplied list array from final result:

// var maxItemsForOldApplied = nonMatchedItemsForOldApplied.Where(item => finalResultList.Any(oldItem => oldItem.CouponId == item.CouponId && oldItem.TotalDiscount == item.TotalDiscount && oldItem.Items.SequenceEqual(item.Items))).ToList();

// finalResultList = maxItemsForNewApplied.Concat(maxItemsForOldApplied).ToList();

response.Applied = finalResultList;

response.ErrorCode = "";

response.ErrorDesc = "";

response.Status = "Successful";

return response;

}

response.Applied = null;

response.ErrorDesc = "";

response.ErrorCode = "";

response.Status = "Failed.";

}

catch (Exception ex)

{

// response.Applied = null;

response.ErrorCode = "500";

response.ErrorDesc = "";

response.Status = "Something went wrong. Please try again later.";

}

return response;

}

StartUp.cs:

public class StartUp

{

private readonly IWebHostEnvironment \_env;

public IConfiguration Configuration { get; }

public StartUp(IConfiguration configuration, IWebHostEnvironment env )

{

Configuration = configuration;

\_env = env;

}

public void ConfigureServices(IServiceCollection services)

{

// read jsonfile config:

// ConfigJsonFiles

var path = Path.Combine(Environment.CurrentDirectory, @"JSONFiles\");

// // for RSAClient.json

IConfiguration config = new ConfigurationBuilder()

.SetBasePath(path)

.AddJsonFile("RSAClientData.json")

.AddJsonFile("Applied.json")

.Build();

var rSAClients = config.GetSection("RSAClientData").Get<List<RSAClient>>();

var coupons = config.GetSection("NewApplied").Get<List<Coupon>>();

// // Register the list of items as a singleton

services.AddSingleton(rSAClients);

services.AddSingleton

(coupons);  
  
}  
  
}