Given one DataDescriptor Map and a JSON payload you should be able to transform the JSON payload

to the object described in the DataDescriptor Object, the result of said transformation should be another JSON object.

For this exercise I strongly recommend the use of the Newtonsoft JSON library as it contains a

classes and helper methods that can help you with the deserialization of the data descriptor jsons in this document and the result of the mapping method to implement (to convert the result to instances of the business classes).

You are not limited in any way, you can make research online, consult online documentation.

Let's imagine that one of our carriers creates an shipment order JSON object and transfers it to our services, this is how the information is received, we receive infomation from various sources and structure may differs from our native classes, take this as your payload for the exercise:

EXAMPLE PAYLOAD

{

"FedexGuide": "ORDER\_USMX00001",

"InvoiceNumber": "INVOICE\_G20230107USMX000001", "ShippedAt": "2023-01-07T22:00:00",

"GoesFromAddress": {

"Id" : "ADDRESS\_MX4982178",

"Strt": "P.za de la Constitucion S/N, Centro HistOrico   
"PC": "06066",

"Localization": "Ciudad de Mexico",   
"PoliticalZone": "CDMX",

"Country": "Mexico",   
"CountryCode": "MX"

}

"GoesToAddress": {

"Id" : "ADDRESS\_US3218793",

"Strt": "1600 Pennsylvania Avenue NW, Washington",   
"PC": "20500",

"Localization": "Washington",   
"PoliticalZone": "DC",

"Country": "United States",   
"CountryCode": "US"

1,

de la Cdad. de Mexic

"Items" : [

{

"Length": 15,

"Width": 15,

"Height": 15,

"Weight": 400,

"WUnit": "GRAM",

"WLHUnit": "CENTIMETER" ,

"PackageType": "BOX",

"Qty": 1,

"Product": {

"Id" : "P\_0232121",

"Name": "MONTBLANC 1858 POCKET WATCH EDICION",

"Sku": "118486" }

} , {

"Length": 1 .25,

"Width": 1 .25,

"Height": 1 .25,

"Weight": 1000,

"WUnit": "KILOGRAM",

"WHLUnit": "METER" ,

"PackageType": "PALLET",

"Qty": 2,

"Product": {

"Id" : "PF\_43428",

"Name": "VERO PICA FRESA 8/100ct",

"Sku": "43428" }

} ]

**1**

**I**

Define Classes The following classes:

Business Classes:

Class ShipmentOrder

Property: CarrierShipmentld Type: String

Property: ReceiverReference Type: String

Property:

Property:

Property:

Property:

Class Address   
 Property:

Property:

Property:

Property:

Property:

Property:

Property:

Class Package   
 Property:

Property:

Property:

Property:

Property:

Property:

Property:

Property:

Class Product   
 Property:

Property:

Property:

Mapper classes:

ShippingStartDate Type: Datetime

FromAddress Type: Address

ToAddress Type: Address

Packages Type: Package

Id Type: String

Street Type: String

PostalCode Type: String

City Type: String

RegionStateProvidence Type: String

Country Type: String

CountryCode Type: String

Weight Type: Double

Height Type: Double

Length Type: Double

Width Type: Double

DimenstionsUnit Type: String

WeightUnit Type: String

Packaging Type: String

Quantity Type: Int

Id Type: String

Description Type: String

ProductCode Type: String

Class DataDescriptor

Property: Name

Property Description: Name of

Property: Alias

Property Description: Name of

Property: MapDescription

Type: String

the Map or field (If this data descriptor is

Type: String

the field or Type that this DataDescriptor pc

Type: String

Property Description: Describes this field or Type defined in this data de:

Property: Primitive Type: Bool

Property Description: Flag that says if the data type describes in this Dal

Property: Multiple: Type: Bool

Property Description: Flag that indicates if Field described in this DataDE

Property: Fields Type: DataDescriptor[] (Array or Collectior

PropertyDescription: Fields (if needed) of the object type described in the

■

Data Descriptors for this exercise (import these jsons to your project and deserialize them to Data Descritor objects):

Use these map descriptors and create a method that takes the EXAMPLE PAYLOAD json and

transform it to a ShipmentOrder instance, the method should read the Data Descriptors to know how to map the data in the payload, the resulting JSOn as described at the beggining of this   
document should be then deserialized to the ShipmentOrder class,

The JSON deserialization can be done with the:

Newtonsoft.Json.JsonConvert.DeserializeObject.Deserialize()

method

Fedex Shipment Order Data Descriptor for Fedex shipment orders

{

"Name": "FedexShipment", "Alias": "ShipmentOrder",

"MapDescription": "This is a Fedex data descriptor that   
"Primitive": false,

"Multiple": false,   
"Fields": [

{

"Name": "FedexGuide",

"Alias": "CarrierShipmentId",   
"Type": "String",

"Primitive": true,   
"Multiple": false,

} ,

{

can be converted to a

"Name": "InvoiceNumber",

"Alias": "ReceiverReference", "Type": "String",

"Primitive": true,   
 "Multiple": false, } ,

{

"Name": "ShippedAt",

"Alias": "ShippingStartDate", "Type": "Datetime",

"Primitive": true,   
 "Multiple": false, } ,

{

"Name": "GoesFromAddress", "Alias": "FromAddress",   
"Type": "FedexAddress",   
"Primitive": false,   
"Multiple": false,

} ,

{

"Name": "GoesToAddress",   
 "Alias": "ToAddress",   
 "Type": "FedexAddress",   
 "Primitive": false,   
 "Multiple": false,   
}

{

"Name": "Items",

"Alias": "Packages", "Type": "FedexItem", "Primitive": false,

"Multiple": true, }

]

}

Fedex Address data descriptor for Fedex addresses

{

"Name": "FedexAddress",

"Alias": "Address",

"MapDescription": "This is a Fedex data descriptor   
"Primitive": false,

"Multiple": false,   
"Fields": [

{

"Name": "Id",   
"Alias": "Id",

"Type": "String",   
 "Primitive": true,   
 "Multiple": false   
} ,

{

"Name": "Strt",   
 "Alias": "Street",   
 "Type": "String",   
 "Primitive": true,   
 "Multiple": false   
1 ,

{

"Name": "PC",

"Alias": "PostalCode",   
"Type": "String",

"Primitive": true,   
 "Multiple": false   
1 ,

{

"Name": "Localization",   
"Alias": "City",

"Type": "String",   
 "Primitive": true,   
 "Multiple": false   
1 ,

{

"Name": "PoliticalZone",

"Alias": "RegionStateProvidence",   
"Type": "String",

"Primitive": true,   
 "Multiple": false   
1 ,

{

"Name": "Country",   
"Alias": "Country",

that can be converted to an

"Type": "String",   
 "Primitive": true,   
 "Multiple": false 1 ,

{

"Name": "CountryCode",

"Alias": "CountryCodee", "Type": "String",

"Primitive": true,   
 "Multiple": false }

]

}

Fedex Item data descriptor for Fedex Shipment Item (a package)

{

"Name": "Fedexltem", "Alias": "Package",

"MapDescription": "This is a Fedex data descriptor   
"Primitive": false,

"Multiple": false,   
"Fields" : [,

{

"Name": "Length",   
 "Alias": "Length",   
 "Type": "Double",   
 "Primitive": true,   
 "Multiple": false   
} ,

{

"Name": "Width",   
 "Alias": "Width",   
 "Type": "Double",   
 "Primitive": true,   
 "Multiple": false   
} ,

{

"Name": "Height",   
"Alias": "Height",   
"Type": "Double",

that can be converted to an

"Primitive": true,   
 "Multiple": false 1 ,

{

"Name": "Weight", "Alias": "Weight", "Type": "Double",

"Primitive": true,   
 "Multiple": false 1 ,

{

"Name": "WUnit",

"Alias": "WeightUnit", "Type": "String",   
"Primitive": true,   
"Multiple": false

1 ,

{

"Name": "WLHUnit",

"Alias": "DimensionsUnit", "Type": "String",

"Primitive": true,   
 "Multiple": false 1 ,

{

"Name": "PackageType", "Alias": "Packaging", "Type": "Strig",

"Primitive": true,   
 "Multiple": false 1 ,

{

"Name": "Qty",

"Alias": "Quantity", "Type": "Int",

"Primitive": true,   
 "Multiple": false }

}

Product data descriptor from fedex

{

"Name": "FedexProduct", "Alias": "Product",

"MapDescription": "This is a Fedex data descriptor that can be converted to an

"Primitive": false, "Multiple": false,

"Fields" : [

{

"Name": "Id",

"Alias": "Id",

"Type": "String",   
 "Primitive": true,   
 "Multiple": false 1 ,

{

"Name": "Name",

"Alias": "Description", "Type": "String",

"Primitive": true,   
 "Multiple": false } ,

{

"Name": "Sku",

"Alias": "ProductCode", "Type": "String",

"Primitive": true,   
 "Multiple": false }

]

}

**I** ■

You have one week after this exercise has been sent to you.

Best regards.