1.Denomination Routine:

namespace ATM

{

class Program

{

static void Main(string[] args)

{

int[] cartridge = new int[] { 10, 50, 100 }; //storing all the denominations in an array

int[] payOut = new int[] {30,50,60,80,140,230,370,610,980 };//storing all the Pay Out amounts in an array

for (int payoutLength = 0; payoutLength < payOut.Length; payoutLength++)

{

Console.WriteLine();

Console.WriteLine("Pay Out For " + payOut[payoutLength].ToString());

Console.WriteLine();

denominationsNew(cartridge, payOut[payoutLength]);

}

}

public static void SingleTypeNotePrint(int cartridge, int notesprint)

{

Console.Write(cartridge + "\tx\t" + notesprint + "\t EUR" );

}

// Write a program which will calculate for each payout the possible combinations which the ATM can pay out.

public static void denominationsNew(int[] cartridge, int iputAmount)

{

int printConbination = 1;

for ( int i=0;i<cartridge.Length;i++)

{

if (iputAmount/cartridge[i]!=0 && iputAmount %cartridge[i]==0)

{

int remainder = 0;

int notes = Math.DivRem(iputAmount, cartridge[i], out remainder);

Console.WriteLine("Option " + printConbination.ToString());

SingleTypeNotePrint(cartridge[i], notes);

Console.WriteLine();

printConbination += 1;

for (int x=1;x<cartridge.Length;x++)

{

int l = notes - (cartridge[x]/cartridge[i]);

while (l > 0)

{

for (int y=1; y<(cartridge[x]/cartridge[i]);y++)

{

if (((cartridge[i]\*l) + (cartridge[x]\*y)) ==iputAmount)

{

;

Console.WriteLine();

Console.WriteLine("Option "+ printConbination.ToString());

Console.WriteLine( (cartridge[i].ToString() + "\t\*\t" + l).ToString() + "\t+\t" + (cartridge[x].ToString()+ "\t\*\t" + y.ToString()).ToString() + "\tEUR");

printConbination += 1;

}

}

l = l - 1;

}

}

}

}

}

}

}

2.REST server  
Customer Class File.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.ComponentModel.DataAnnotations;

//using ServiceStack.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace RestAPIService

{

public class Customer

{

[Key]

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public int age { get; set; }

}

}

CustomerArray.cs File

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.IO;

using Newtonsoft.Json;

namespace RestAPIService

{

public class CustomerArray

{

public int Size { get; set; }

public List <Customer> Customers { get; set; }

private string fileName = "Data/customers.json";

private bool InsertOne( Customer customer)

{

int id = this.Size + 1;

customer.Id = id;

if (this.Size == 0)

{

this.Customers.Insert(0,customer);

}

else

{

for (int i=0;i<Size;i++)

{

if(Compare(customer,Customers[i])<0)

{

Customers.Insert(i, customer);

// Size = ;

}

}

}

this.Size = Customers.Count();

return true;

}

private int Compare(Customer left,Customer right)

{

string leftKey = left.LastName + ", " + left.FirstName;

string RightKey = right.LastName + ", " + right.FirstName;

return leftKey.CompareTo(RightKey);

}

public bool InsertMany(Customer [] customers)

{

if (customers.Length<2)

{

throw new Exception("At least provide two customers details");

}

for (int i = 0; i < customers.Length; i++)

{

if (string.IsNullOrEmpty (customers[i].FirstName))

{

throw new Exception("Firstname can not be empty, please provide some value");

}

if (string.IsNullOrEmpty(customers[i].LastName))

{

throw new Exception("Lastname can not be empty, please provide some value");

}

if (string.IsNullOrEmpty(customers[i].age.ToString()))

{

throw new Exception("Age can not be empty, please provide some value");

}

if (customers[i].age < 18)

{ throw new Exception("Age must be greater than 18"); }

}

for (int i=0;i<customers.Length;i++)

{

InsertOne(customers[i]);

}

//save jason

File.WriteAllText(fileName, JsonConvert.SerializeObject(Customers));

return true;

}

public CustomerArray ()

{

if (File.Exists(fileName))

{

string json = File.ReadAllText(fileName);

this.Customers = JsonConvert.DeserializeObject<List<Customer>>(json);

Size = Customers.Count();

}

else

{

Customers = new List<Customer>();

Size = 0;

}

}

}

}

CustomerController.cs

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

// For more information on enabling Web API for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

namespace RestAPIService.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class CustomerController : ControllerBase

{

private CustomerArray customers = new CustomerArray();

// GET: api/<CustomerController>

[HttpGet]

public Customer[] Get()

{

return customers.Customers.ToArray();

}

// POST api/<CustomerController>

[HttpPost]

public void Post([FromBody] Customer [] value)

{

customers.InsertMany(value);

}

}

}

If you will run this program, please create a Data folder on project path for data persistent.