* Generate a unique Case number

C#

public int RandomNumber()

{

Random random = new Random();

return random.Next();

}

* Calculate reschedule Days based on Zip Code

C#

public DateTime RescheduleDays(String zipCode, DateTime nextPaymentDate)

{

if (nextPaymentDate != null && !string.IsNullOrEmpty(zipCode))

{

switch (zipCode)

{

case "85026":

nextPaymentDate = nextPaymentDate.AddDays(2);

break;

case "75014":

nextPaymentDate = nextPaymentDate.AddDays(3);

break;

case "75063":

nextPaymentDate = nextPaymentDate.AddDays(4);

break;

case "76014":

nextPaymentDate = nextPaymentDate.AddDays(5);

break;

default:

break;

}

}

else

{

return DateTime.Today.AddDays(2);

}

return nextPaymentDate;

}

* Calculate Next Payment Date

public DateTime NextPaymentDate(DateTime nextPaymentDate)

{

if(nextPaymentDate!=null)

{

nextPaymentDate = nextPaymentDate.AddDays(2);

return nextPaymentDate;

}

return DateTime.Today.AddDays(2);

}

* Convert Pipe delimited to JSON

If you are using .NET Core

<ItemGroup>

<PackageReference Include="Newtonsoft.Json" Version="9.0.1" />

</ItemGroup>

using Newtonsoft.Json;

public string ConvertPipeDelmitedFileToJsonObject(string path)

{

var pipeLines = new List<string[]>();

var lines = File.ReadAllLines(path);

foreach (string line in lines)

pipeLines.Add(line.Split('|'));

var properties = lines[0].Split('|');

var listObjResult = new List<Dictionary<string, string>>();

for (int i = 1; i < lines.Length; i++)

{

var objResult = new Dictionary<string, string>();

for (int j = 0; j < properties.Length; j++)

objResult.Add(properties[j], pipeLines[i][j]);

listObjResult.Add(objResult);

}

return JsonConvert.SerializeObject(listObjResult);

}