Developer Technical Questions.

1. Taking the following EDIFACT message text, write some code to parse out the all the LOC segments and populate an array with the 2nd and 3rd element of each segment.

Note: the ‘+’ is an element delimiter

UNA:+.? '

UNB+UNOC:3+2021000969+4441963198+180525:1225+3VAL2MJV6EH9IX+KMSV7HMD+CUSDECU-IE++1++1'

UNH+EDIFACT+CUSDEC:D:96B:UN:145050'

BGM+ZEM:::EX+09SEE7JPUV5HC06IC6+Z'

LOC+17+IT044100'

LOC+18+SOL'

LOC+35+SE'

LOC+36+TZ'

LOC+116+SE003033'

DTM+9:20090527:102'

DTM+268:20090626:102'

DTM+182:20090527:102'

String LOC = LOC+17+IT044100';

List <> items = new List<>();

String []segmants = LOC.Split(‘+’).ToArray;

items.Add(new Tuple(segmants[1], segmants[2]));

1. Taking the following XML document, write code to extract the RefText values for the following RefCodes: ‘MWB’, ‘TRV’ and ‘CAR’

<InputDocument>

<DeclarationList>

<Declaration Command="DEFAULT" Version="5.13">

<DeclarationHeader>

<Jurisdiction>IE</Jurisdiction>

<CWProcedure>IMPORT</CWProcedure>

<DeclarationDestination>CUSTOMSWAREIE</DeclarationDestination>

<DocumentRef>71Q0019681</DocumentRef>

<SiteID>DUB</SiteID>

<AccountCode>G0779837</AccountCode>

<Reference RefCode="MWB">

<RefText>586133622</RefText>

</Reference>

<Reference RefCode="KEY">

<RefText>DUB16049</RefText>

</Reference>

<Reference RefCode="CAR">

<RefText>71Q0019681</RefText>

</Reference>

<Reference RefCode="COM">

<RefText>71Q0019681</RefText>

</Reference>

<Reference RefCode="SRC">

<RefText>ECUS</RefText>

</Reference>

<Reference RefCode="TRV">

<RefText>1</RefText>

</Reference>

<Reference RefCode="CAS">

<RefText>586133622</RefText>

</Reference>

<Reference RefCode="HWB">

<RefText>586133622</RefText>

</Reference>

<Reference RefCode="UCR">

<RefText>586133622</RefText>

</Reference>

<Country CodeType="NUM" CountryType="Destination">IE</Country>

<Country CodeType="NUM" CountryType="Dispatch">CN</Country>

</DeclarationHeader>

</DeclarationList>

</InputDocument>

For (RefCodes == ‘MWB’, ‘TRV’, ‘CAR’)

Return RefText;

1. Write a webservice that accepts the following XML document as the payload:

<InputDocument>

<DeclarationList>

<Declaration Command="DEFAULT" Version="5.13">

<DeclarationHeader>

<Jurisdiction>IE</Jurisdiction>

<CWProcedure>IMPORT</CWProcedure>

<DeclarationDestination>CUSTOMSWAREIE</DeclarationDestination>

<DocumentRef>71Q0019681</DocumentRef>

<SiteID>DUB</SiteID>

<AccountCode>G0779837</AccountCode>

</DeclarationHeader>

</Declaration>

</DeclarationList>

</InputDocument>

The webservice should respond with a status code which is based on parsing the contents of the XML payload

1. If the XML document is given here is passed then return a status of ‘0’ – which means the document was structured correctly.
2. If the Declararation’s Command <> ‘DEFAULT’ then return ‘-1’ – which means invalid command specified.
3. If the SiteID <> ‘DUB’ then return ‘-2’ – invalid Site specified.

XmlDocument NewSourceXML = new XmlDocument();

NewSourceXML.LoadXml(SourceXML);

If (Declaration Command==” DEFAULT”)

return {statusCode : -1};

if (SiteID == “DUB”)

return {statusCode : -2};