

# Konverter für XML -> CSV

Kyuhyun Jo  
Yaser Said-Ahmet  
Alisina Yozbashi  
Nasim Shahbazi

# Logik für die Konversion. (XML -> CSV)

Step 1. Lesen XML File

Step 2. Schreiben einzel string Zeil

Step 3. Pflanzen den XML Node Baum

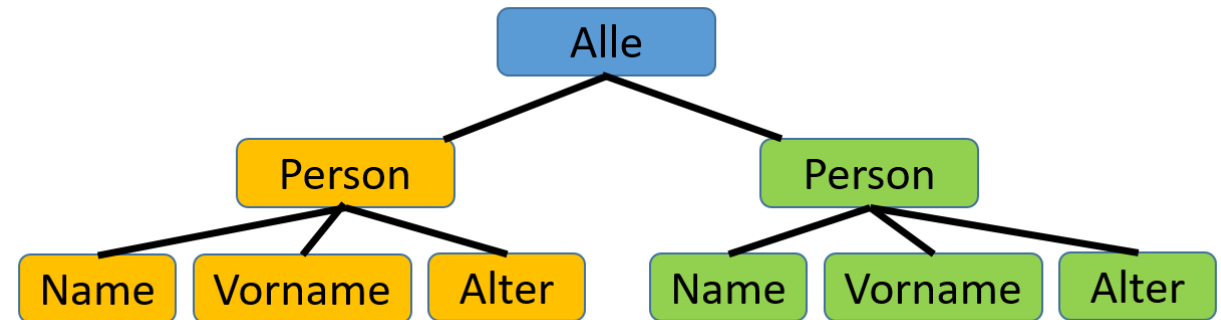
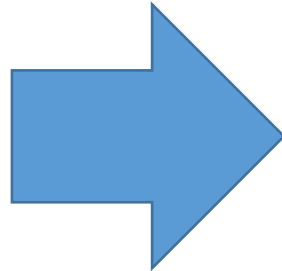
Step 4. Auswählen die Spalte des CSVs

Step 5. Schreiben CSV File

# Logik für die Konversion. (XML -> CSV)

Was ist der Node Baum ?

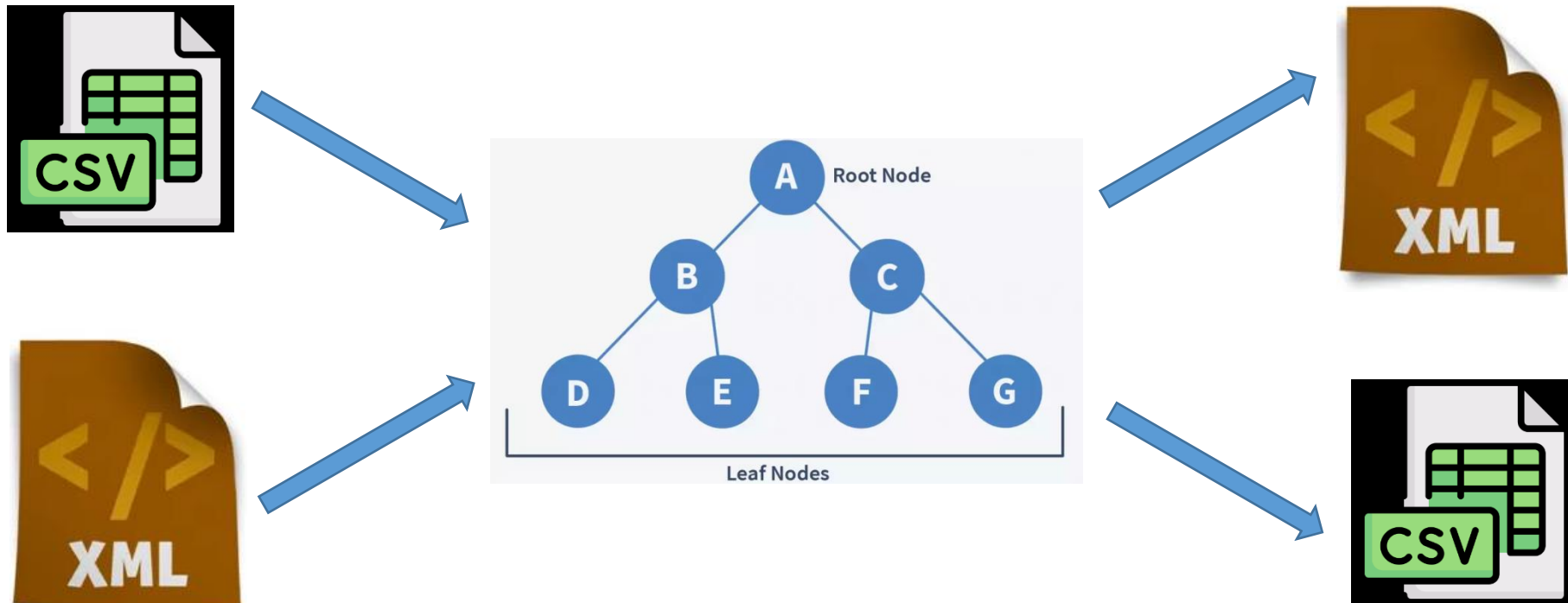
```
▼<Alle>
  ▼<Person>
    <Name>Schmidt</Name>
    <Vorname>Tim</Vorname>
    <Alter>18</Alter>
  </Person>
  ▼<Person>
    <Name>Mayer</Name>
    <Vorname>Tina</Vorname>
    <Alter>20</Alter>
  </Person>
</Alle>
```



# Logik für die Konversion. (XML -> CSV)

Warum muss man einen Node Baum pflanzen?

1. **Richtungsfreiheit** => Lesen der Datei und Schreiben einer Datei.  
**komplett getrennt!!!!!!**

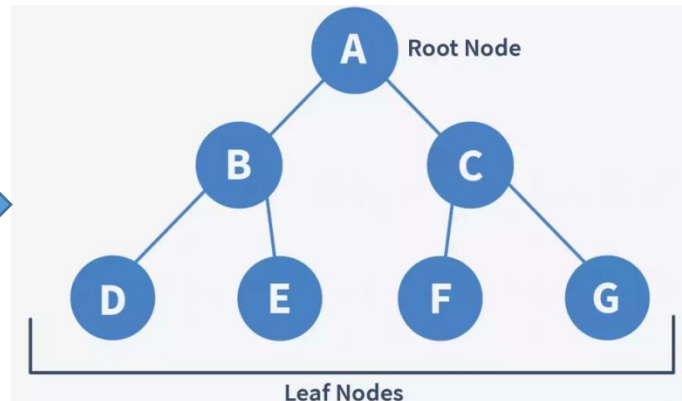


# Logik für die Konversion. (XML -> CSV)

Warum muss man einen Node Baum planen?

2. Programm-Erweiterbarkeit => Neue Dateityp können hinzugefügt werden\*  
(für INPUT, OUTPUT Beide seite)

Etwas Neue Dateityp



Etwas Neue Dateityp



# Step 1. Lesen XML File

## Lesen XML File

```
public string[] ReadXML(string Verzeichnis)
{
    string[] str = File.ReadAllLines(Verzeichnis);
    return str;
}
```

## Step 2. Schreiben einzel string Zeil

2.1 Schreibt ein Array von Zeichenfolgen in eine einzelne Zeichenfolge um.

```
public string MakeAStringFromStrArr(string[] strArr)
{
    string res = string.Empty;
    foreach (string str in strArr)
    {
        res += str;
    }
    return res;
}
```

# Step 3. Planzen den XML Node Baum

3.1 Ersetzen einen Node Instanz

3.2 Ausfüllen der Information des Nodes

3.3 Wenn der Node Kinder hat, rufen den Rekursive Method auf.

3.4 Geben der Node zurück



# Step 3. Planzen den XML Node Baum

## 3.1 Ersetzen einen Node Instanz

```
Node NodeBaumPlazen(GelesendeStringXML)
{
    Node theNode = new Node();
    ...
}
...
class Node
{
    List<Node> ChildNodes = new List<Node>();
    InfoOfNode info = new InfoOfNode();
}
```

# Step 3. Planzen den XML Node Baum

## 3.2 Ausfüllen der Information des Nodes

### <Inhalt der InfoOfNode Klass>

- 4 Index von Vor- und Hinter des Anfang und Ende Klammer
- Titel des Nodes
- element des Nodes
- Anzahl der KinderNodes

\*\* Eltern weiß nur der Anzahl der Kinder.

\*\* Eltern weiß gar nicht über Onkelkinder.

\*\* Kinder weiß nur der Eltern aber die Geschwester nicht.

```
Node NodeBaumPlazen(GelesendeStringXML)
{...
    TheNode.info = AusfuellenAllInfoOfNode(gelesendeneXMLstr);
...}
```

# Step 3. Planzen den XML Node Baum

## 3.3 Wenn der Node Kinder hat...

Falls der Node **mind. einen Kinder** hat,  
rufen den **\*\*Rekursive Method** auf!!

```
Node NodeBaumPlazen(GelesendeStringXML)
{
    ...
    for(int i = 0; i<der_Anzahl_der_Kinder; i++)
    {
        theNode.ChildNodes.Add(NodeBaumPlazen(schneidete_Inhalt_des_Node));
    }
    ...
}
```

# Step 3. Planzen den XML Node Baum

## 3.4 Geben den Node zurück

Nach diese Method allen ChildNode schreiben,  
gibt disen Rückgabe zurück.

```
Node NodeBaumPlazen(GelesendeStringXML)
{ ...
  return TheNode;
}
```

# Step 4. Auswählen die Spalte des CSVs

4.1 lesen Node Baum und geben allen Themen.

4.2 sortieren und zeigen allen mögliche Themen..(wenn es gibt.)

4.3 auswählen einen Them

=>benutzer. mit console.read()

4.4 geben "ausgewählter Them" zurück.

=> return "ausgewählter Them" : string

# Step 4. Auswählen die Spalte des CSVs

4.1 lesen Node Baum und geben allen Themen.

```
private string ZeigenAllenMoeglichenThemUndEinenAuswaehlen(Node
nodeTree)
{
    List<string> einPlatzFürMoeglichenThemen = new List<string>();
    List<string> AlleMoeglicheThemen =
        GebenAllenMoeglichenThemen(nodeTree,
        einPlatzFürMoeglichenThemen);
    ...
}
```

# Step 4. Auswählen die Spalte des CSVs

## 4.1 lesen Node Baum und geben allen Themen.

```
private List<string> GebenAllenMoeglichenThemen(Node node, List<string>
PlatzFürMoeglichenThemen)
{
    if (node.info.AnzahlKinderUndInhalts.anzahlDerKinderNode > 0)
    {
        PlatzFürMoeglichenThemen.Add(node.info.title_Node);
        foreach (Node ChildNode in node.ChildNodes)
        {
            GebenAllenMoeglichenThemen(ChildNode,
            PlatzFürMoeglichenThemen);
        }
    }
    return PlatzFürMoeglichenThemen;
}
```

Achtung!!  
Reklusiv Method

## Step 4. Auswählen die Spalte des CSVs

4.2 sortieren und zeigen allen mögliche Themen..(wenn es gibt.)

```
private string ZeigenAllenMoeglichenThemUndEinenAuswaehlen(Node
nodeTree)
{ ...
  AlleMoeglicheThemen = AlleMoeglicheThemen.Distinct().ToList();
  ..
}
```

Keine  
Duplizierte Themen



# Step 4. Auswählen die Spalte des CSVs

## 4.3 auswählen einen Them

```
private string ZeigenAllenMoeglichenThemUndEinenAuswaehlen(Node
nodeTree)
{ ...
    string NameDesAusgewaehleteThem =
        AuswaehlenEinenThem(AlleMoeglicheThemen);
    ...
}
```

# Step 4. Auswählen die Spalte des CSVs

## 4.3 auswählen einen Them

```
private string AuswaehlenEinenThem(List<string> AlleMoeglicheThemen)
{
    string NameDesAusgewaehleteThem = string.Empty;
    for (int i = 0; i < AlleMoeglicheThemen.Count; i++)
    {
        Console.WriteLine(i + ", " + AlleMoeglicheThemen[i].ToString() + "\n");
    }
    Console.WriteLine("welchen Them?\n");
    int ausgewaehltesteNummer = Console.Read();
    ausgewaehltesteNummer -= 48;
    NameDesAusgewaehleteThem = AlleMoeglicheThemen[ausgewaehltesteNummer].ToString();

    return NameDesAusgewaehleteThem;
}
```

Zeigen  
den alle  
Themen

//ASCII Code ,0' = 48, ,1' = 49...

## Step 4. Auswählen die Spalte des CSVs

4.4 geben "ausgewählter Them" zurück.

```
private string ZeigenAllenMoeglichenThemUndEinenAuswaehlen(Node  
nodeTree)  
{ ...  
    return NameDesAusgewaehleteThem;  
}
```

# Step 5. Schreiben CSV File

5.1 Nennen File\_Name von CSV File mit "ausgewählter Them"

5.2 Schreiben Einen Kompletten Verzeichnis

=>"VerzeichnisOrdner"+"\"+"ausgewählter Them" + ".csv,,

=>(Erste Parameter in File.WriteAllText(path,TXT))

5.3 Schreiben einen Text für CSV

=>(zweite Parameter in File.WriteAllText(path,TXT))

5.4 Schreiben Einen CSV File

# Step 5. Schreiben CSV File

5.1 Nennen File\_Name von CSV File mit "ausgewählter Them,,

5.2 Schreiben Einen Kompletten Verzeichnis

=>"VerzeichnisOrdner"+"\"+"ausgewählter Them" + ".csv,,

=>(Erste Parameter in File.WriteAllText(path,TXT))

```
string komplettenVerzeichnis
```

```
= VerzeichnisOrdner + "\\\" + choosedThema + ".csv";
```

# Step 5. Schreiben CSV File

## 5.3 Schreiben einen Text für CSV

=>(zweite Parameter in File.WriteAllText(path,TXT))

string eineZeile =

MakeTemporaryEineZeilForCSVFromNodeBaum(nodeTree, choosedThema);

# Step 5. Schreiben CSV File

## 5.3 Schreiben einen Text für CSV

=>(zweite Parameter in File.WriteAllText(path,TXT))

```
private string MakeTemporaryEineZeilForCSVFromNodeBaum(Node nodeTree, string choosedThema)
{
    1. Sammeln allen Node,der den title_Node(gleich mit "ausgewählter Them") hat,in dem NodeBaum.

    2. sortierende SpaltenCSV

    3. Dies schneidet erneut um das Semikolon herum und rekonstruiert ohne Duplizierung.

    4. schreiben der erste Zeil von Temporary string list->Spalten.

    5. fuegen von Zweite bis letzte Zeil hinzu => Contentes.

    6. Verketteten Sie die vorbereitete tempStrCSV zu einer einzelnen Zeichenfolge.

    7. geben der TXTFuerCSV zurück
}
```

# Step 5. Schreiben CSV File

## 5.4 Schreiben Einen CSV File

```
File.WriteAllText(kompletenVerzeichnis, eineZeileTXTCSV);
```

Endlich..

Ein CSV File Schreiben!!!!



Vielen Dank für Deine Aufmerksamkeit.

Und Der gesamte Quellcode ist hier.

(Dafür habe ich Viel zeit gearbeitet. Aber Ohne Diskussion kein Fortschritt. Wir freuen uns jederzeit auf den Austausch mit Ihnen.)

NextCloud -> E2FI3 -> string-> ConverterKJO-> V1.1

Fragen? Das nehmen wir sehr sehr gerne!