Projektpräsentation: CAMT053 - Webservice Berlin, den 17.07.2020

Ausbildungsbetrieb: a.b.s. Rechenzentrum GmbH
Marvin Viedt
Bornhagenweg 39
12309 Berlin
017687829915
marvin.viedt@googlemail.com

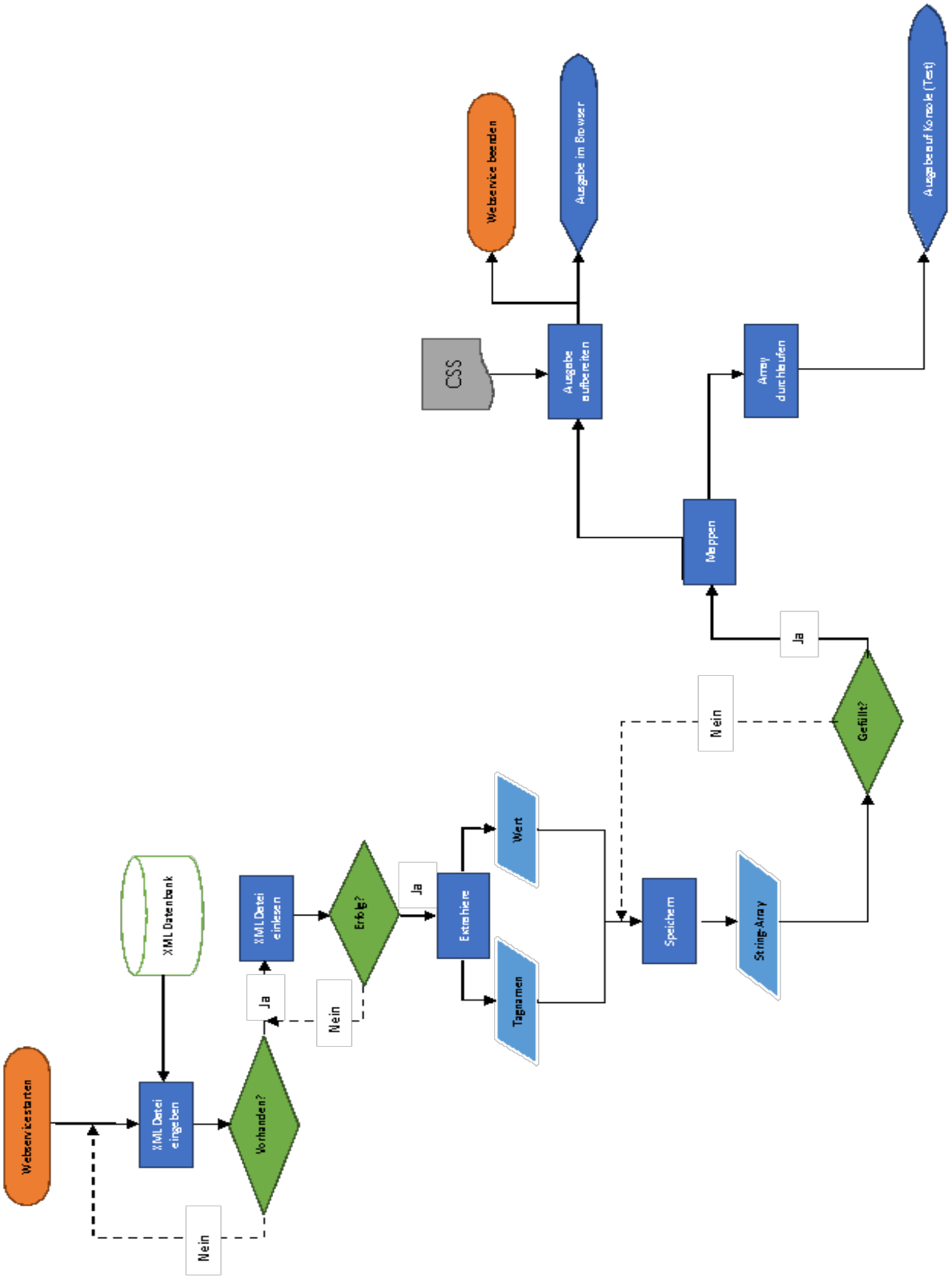
Handout

Inhaltsverzeichnis

CAMT053 Datei (Auszug)

```
<?xml version="1.0" encoding="UTF-8" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:camt.053.001.02" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <BkToCstmrStmt>
        <GrpHdr>
            <MsgId>[redacted]</MsgId>
            <CreDtTm>[redacted]</CreDtTm>
            <MsgRcpt>
                <Nm>[redacted]</Nm>
                <PstlAdr>
                    <AdrLine>[redacted]</AdrLine>
                    <AdrLine>[redacted]</AdrLine>
                </PstlAdr>
            </MsgRcpt>
            <MsgPgntn>
                <PgNb>[redacted]</PgNb>
                <LastPgInd>[redacted]</LastPgInd>
            </MsgPgntn>
        </GrpHdr>
        <Stmt>
            <Id>[redacted]</Id>
            <ElctrncSeqNb>[redacted]</ElctrncSeqNb>
            <CreDtTm>[redacted]</CreDtTm>
            <FrToDt>
                <FrDtTm>[redacted]</FrDtTm>
                <ToDtTm>[redacted]</ToDtTm>
            </FrToDt>
            <Acct>
                <Id>
                    <IBAN>[redacted]</IBAN>
                </Id>
                <Ccy>[redacted]</Ccy>
                <Ownr>
                    <Nm>[redacted]</Nm>
                    <PstlAdr>
                        <AdrLine>[redacted]</AdrLine>
                        <AdrLine>[redacted]</AdrLine>
                    </PstlAdr>
                </Ownr>
                <Svcr>
                    <FinInstnId>
                        <BIC>[redacted]</BIC>
                        <Nm>[redacted]</Nm>
                        <Othr>
                            <Id>[redacted]</Id>
                            <Issr>[redacted]</Issr>
                        </Othr>
                    </FinInstnId>
                </Svcr>
            </Acct>
            <Bal>
                <Tp>
                    <CdOrPrtry>
                        <Cd>[redacted]</Cd>
                    </CdOrPrtry>
                </Tp>
                <Amt Ccy=[redacted]>[redacted]</Amt>
                <CdtDbtInd>[redacted]</CdtDbtInd>
                <Dt>
                    <Dt>[redacted]</Dt>
                </Dt>
            </Bal>
            <Bal>
                <Tp>
                    <CdOrPrtry>
                        <Cd>[redacted]</Cd>
                    </CdOrPrtry>
```

```
                </Tp>
                <Amt Ccy=[redacted]</Amt>
                <CdtDbtInd>[redacted]</CdtDbtInd>
                <Dt>
                    <Dt>[redacted]</Dt>
                </Dt>
            </Bal>
            <Bal>
                <Tp>
                    <CdOrPrtry>
                        <Cd>[redacted]</Cd>
                    </CdOrPrtry>
                </Tp>
                <Amt Ccy=[redacted]</Amt>
                <CdtDbtInd>[redacted]</CdtDbtInd>
                <Dt>
                    <Dt>[redacted]</Dt>
                </Dt>
            </Bal>
            <Ntry>
                <Amt Ccy=[redacted]</Amt>
                <CdtDbtInd>[redacted]</CdtDbtInd>
                <Sts>[redacted]</Sts>
                <BookgDt>
                    <Dt>[redacted]</Dt>
                </BookgDt>
                <ValDt>
                    <Dt>[redacted]</Dt>
                </ValDt>
                <AcctSvcrRef>[redacted]</AcctSvcrRef>
                <BkTxCd>
                    <Domn>
                        <Cd>[redacted]</Cd>
                        <Fmly>
                            <Cd>[redacted]</Cd>
                            <SubFmlyCd>[redacted]</SubFmlyCd>
                        </Fmly>
                    </Domn>
                    <Prtry>
                        <Cd>[redacted]</Cd>
                        <Issr>[redacted]</Issr>
                    </Prtry>
                </BkTxCd>
                <NtryDtls>
                    <TxDtls>
                        <Refs>
                            <EndToEndId>[redacted]</EndToEndId>
                            <TxId>[redacted]</TxId>
                            <MndtId>[redacted]</MndtId>
                            <ClrSysRef>[redacted]</ClrSysRef>
                        </Refs>
                        <BkTxCd>
                            <Domn>
                                <Cd>[redacted]</Cd>
                                <Fmly>
                                    <Cd>[redacted]</Cd>
                                    <SubFmlyCd>[redacted]</SubFmlyCd>
                                </Fmly>
                            </Domn>
                            <Prtry>
                                <Cd>[redacted]</Cd>
                                <Issr>[redacted]</Issr>
                            </Prtry>
                        </BkTxCd>
                        <RltdPties>
                            <Dbtr>
```


Webservicestarten
XML Datei eingeben
Vorhanden?
Nein
Ja
XML Datenbank
XML Datei einlesen
Erfolg?
Nein
Ja
Extrahiere
Wert
Tagnamen
Speichern
String-Array
Gefüllt?
Nein
Ja
Mappen
Array durchlaufen
Ausgabe aufbereiten
CSS
Webservice beenden
Ausgabe im Browser
Ausgabe auf Konsole (Test)

```
Algorithmus readxmlfile(check, filename, r, w)

Ausgabe: Gebe Test2 auf der CMD aus (Kontrolle zum Funktionsstart)

Deklariere Variablen pwd, err und Zuweisung des Dateipfades der XML Datei

wenn Dateipfad ist falsch dann

      Ausgabe: XML Datei nicht gefunden

      beende

wenn_ende

Ausgabe: Dateipfad

Deklariere Variablen getxmlcontent, _ und öffne XML Datei

Deklariere parsexmlcontent, err und parse die XML Datei

wenn Parsen ist Fehlgeschlagen dann

      Ausgabe: Parsen der XML Datei ist fehlgeschlagen

      beende

wenn_ende

Deklariere Variable grphdr und suche das erste Element (MsgId) in GrpHdr

Deklariere Array grphdrarr und Aufruf des Algorithmus getxmlvalues(grphdr, r, w)

Deklariere Variable stmt und suche das erste Element (Id) in Stmt

Deklariere Array stmtarr und Aufruf des Algorithmus getxmlvalues(stmt, r, w)

Ausgabe: Gebe das Array stmtarr auf der CMD aus (Kontrolle)

wenn Check ist true dann

      Aufruf des Algorithmus printaussugscontent(w, grphdrarr, stmtarr)

sonst

      Aufruf des Algorithmus printxmlcontent(w, grphdrarr, stmtarr)

      beende

wenn_ende

Gebe den Rückgabewert nil zurück

Ende
```

```
func readxmlfile(check bool, filename string, r *http.Request, w http.ResponseWriter) error {

    fmt.Println("Test 2\n")

    pwd, err := filepath.Abs("./XML/" + filename + ".xml")
    if err != nil {
        fmt.Println("Die XML Datei konnte nicht gefunden werden: %s", err)
        return err
    }

    fmt.Println("Der Dateipfad ist: %s\n", pwd)
    getxmlcontent, _ := os.Open(pwd)

    parsexmlcontent, err := xmlquery.Parse(getxmlcontent)
    if err != nil {
        fmt.Println("Öffnen der XML Datei ist fehlgeschlagen: %s", err)
        return err
    }

    fmt.Println("-----------------------------------------------------------------")

    grphdr := xmlquery.FindOne(parsexmlcontent, "//BkToCstmrStmt/GrpHdr/MsgId")
    grphdrarr := getxmlvalues(grphdr, r, w)

    stmt := xmlquery.FindOne(parsexmlcontent, "//BkToCstmrStmt/Stmt/Id")
    stmtarr := getxmlvalues(stmt, r, w)

    //fmt.Fprintf(w, "%s\n", Map)
    fmt.Println("%s\n", stmtarr)

    if check == true {
        printaussugscontent(w, grphdrarr, stmtarr)
    }
    if check == false {
        printxmlcontent(w, grphdrarr, stmtarr)
    }

    return nil
}
```

sandbox.go Imports off Syntax off

```go
package main

import (
    "encoding/xml"
    "fmt"
)

type pruefling struct {
    Prueflingsnummer string `xml:"prf_nr"`
    IHK_Name         string `xml:"ihk_name"`
    Name             string `xml:"name"`
    Alter            string `xml:"alter"`
    Ort              string `xml:"ort"`
    Ausbildungsberuf string `xml:"aberuf"`
    Firma            string `xml:"firma"`
}

func main() {
    pruefling := &pruefling{
        Prueflingsnummer:   "100000",
        IHK_Name:           "IHK Musterstadt",
        Name:               "Max",
        Alter:              "27",
        Ort:                "Musterstadt",
        Ausbildungsberuf:   "Fachinformatiker Anwendungsentwicklung",
        Firma:              "Muster GmbH",
    }

    xml, _ := xml.MarshalIndent(pruefling, "", " ")

    fmt.Println(string(xml))

}
```

Reset Format Run

```
<pruefling>
 <prf_nr>100000</prf_nr>
 <ihk_name>IHK Musterstadt</ihk_name>
 <name>Max</name>
 <alter>27</alter>
 <ort>Musterstadt</ort>
 <aberuf>Fachinformatiker Anwendungsentwicklung</aberuf>
 <firma>Muster GmbH</firma>
</pruefling>

Program exited.
```