

## Lernziel

Der Student soll

- # über ein WCF Felder und userdefinierte Objekte an eine Konsole übermitteln können,
- # Objektfelder und Felder einfachen Datentyps clientseitig verwalten können.
- # **ACHTUNG:** Sämtliche Datenoperationen laufen über Felder, in denen die Objekte gespeichert sind. Der Zugriff auf Informationen aus der GUI ist notwendig, um die Elemente aus den Feldern zu spezifizieren. Beispiel: Index eines Controls referenziert die Information aus dem entsprechenden Feld.
- # Die Methoden für ASPX und HTML sind selbständig zu definieren und dazu ihre Übergabeparameter! Die GUI ist als Schablone bekannt.
- # Die HTML/JavaScript-Informationen sind stateless. D.h. nach jedemPostBack sind die Inhalte dieser Controls leer. Man muss also darauf achten, dass beim Laden der Seite nach einemPostBack die HTML-Controls wieder ihre Informationen erhalten.
- # *Wissen, wann man Feder benutzen soll und wie sie in Verbindung mit Hiddenfields stehen!*

## Aufgabenstellung

In einer XML-Datei (Abb. 1) sind Firmen mit ihrem Sortiment enthalten. Es ist eine restful WCF-Anwendung zu implementieren, die auf ein Assembly (DLL = Bibliothek) zugreift und Interface-Methoden zur Verfügung stellt. Eine zu entwickelnde Web-Anwendung soll über ein Web-Service WCF auf die DDL zugreifen. Die Anwendung besteht aus 3 Web-Seiten:

### 1. GUI und User-Stories zum ersten GP

Firmen und ihr Warensortiment sind anzuzeigen. Es sind einerseits alle Artikel aus dem gesamten Sortiment aller Firmen anzuzeigen, andererseits das Sortiment mit den Namen der Lieferfirmen. Da verschiedenen Firmen teilweise dieselben Artikel in ihrem Sortiment anbieten, sollen diese in Auswahl-Controls nur einmal aufscheinen! Auch sollen die Firmen mit ihren Adressen angezeigt werden.

Das Repository ist dasselbe wie im TS 03, also die XML-Datei (Abb. 1)!

```
<?xml version="1.0"?>
<Bestellungen>
  - <Firma Ort="Graz" PLZ="8010">
    Huber
    - <Artikelliste>
      <Artikel Rabatt="5" RabMenge="50" Menge="120" Preis="2.50" ArtID="100">Heft</Artikel>
      <Artikel Rabatt="10" RabMenge="50" Menge="100" Preis="1.80" ArtID="120">Kuli</Artikel>
    </Artikelliste>
  </Firma>
  - <Firma Ort="Wien" PLZ="1010">
    Weber
```

Abb. 1: Auszug aus der XML-Datei

GUI (Abb. 2): Sie besteht aus folgenden Controls:

- ✓ HTML-select-Control „Artikel“: Enthält alle Artikel sämtlicher Firmen nur einmal!
- ✓ 2 HTML-TextAreas: Enthalten die Artikel mit Firmennamen bzw. die Lieferanten
- ✓ HTML-Input (Button): Zur Ausgabe der Firmen in die 2. TextArea
- ✓ ASPX-GridView: Zeigt nur die Artikel
- ✓ ASPX-GridView: Zeigt nur die Firmen
- ✓ 3 ASPX-Schalter: siehe unten.

### Funktionalitäten / User-Stories:

Für clientseitige Datenmanipulationen sind immer Felder zu verwenden!

**Beim Laden:** Ermittelt das gesamte Artikelsortiment mit den dazugehörigen Firmennamen (=erweiterte Artikelliste) aus der XML-Datei über das WCF. Die Artikel werden genau einmal im Select-Control angezeigt. Das gesamte Sortiment mit den Firmennamen in der 1. TextArea. Die 2.

TextArea und die GRVs sind leer (Abb. 2). In der Folge wird nicht mehr explizit auf den WCF-Zugriff hingewiesen. Achtung: Hier werden einmal das Sortiment ohne und einmal mit Firmennamen erstellt.

ASPX: Zeige Sortiment (ohne Firmen): Zeigt das gesamte Sortiment in der GRV – ohne die Firmennamen - an (Abb. 3).



Abb. 2: GUI beim Laden der Seite



Abb. 3: Gesamts Sortiment aus der XML-Datei

ASPX: Zeige alle Firmen: Ermittelt die Firmen aus dem BL und zeigt sie in der 2. GRV an (Abb. 4).

HTML: Zeige alle Firmen (Lieferanten): Zeigt die vorher ermittelten Firmen in der 2. TextArea an (Abb. 5).

**Achtung:** Es ist auf folgendes zu beachten: Betätigt man einen ASPX-Schalter, so ist die 2. TextArea LEER(!), da beim Laden der HTML-Seite diese ja „stateless“ ist!!! Was ist zu tun, damit der Inhalt der 2. TextArea angezeigt wird?



Abb. 4: Alle Firmen in der GRV



Abb. 5: Alle Firmen in der 2. TextArea

ASPX: Navigiere zur nächsten Seite: Ruft die Seite für den 2. GP auf. Dort werden die Firmennamen, die Artikel-Bezeichnungen (clientseitig) und die erweiterte Artikelliste benötigt (serverseitig) benötigt.

## 2. Klassen der Bibliothek BL\_Firma, HTML und des WCF

In diesem Beispiel besteht der BL aus den Klassen Firma und DAL. Das WCF greift auf die Klasse Firma zu und die Klasse Firma auf den DAL!

### Die Klassen im BL:

**Klasse Firma:** Sie ist eine sogenannte Transferklasse. D.h. sie implementiert die Methoden, die vom WCF aufgerufen werden.

**Eigenschaften: m\_Dal::** Ein Objekt der Klasse DAL

#### *Methoden:*

`GetAllArticlesBL(): List<ArticleComp>::` Ermittelt über den DAL die Artikel des gesamten Sortiments inklusive der Firmennamen. Gibt ein Feld vom Typ `ArticleComp` zurück (siehe unten).

`GetAllCompaniesBL(): List<Company>::` Ermittelt über den DAL die Firmen und gibt sie als Liste zurück.

**Klasse DAL:** Zugriff auf die XML-Datei.

**Eigenschaften:** `m_Doc`:: DOM des XML-Datei

#### *Methoden:*

`GetAllArticlesWithComp(): List<ArticleComp>::` Ermittelt aus der XML-Datei die Artikel des gesamten Sortiments und fügt den Firmennamen hinzu. Gibt ein Feld vom Typ `ArticleComp` zurück (siehe unten).

`GetAllCompanies(): List<Company>::` Ermittelt aus der XML-Datei die Firmen und gibt sie als Liste zurück.

**Klasse Company:** Verwaltet die Firmendaten (Name, Ort, PLZ)

**Klasse Article:** Verwaltet die Artikeldaten (ArtID, Bezeichnung, Menge, Preis).

**Klasse ArticleComp:** Verwaltet die Artikeldaten inklusive des Firmennamens (CompName, ArtID, Bezeichnung, Menge, Preis). Ableitung???

#### Das WCF implementiert eben 2 Methoden:

- ✓ `GetAllArticlesSvcWithCompSvc(): List<ArticleComp>::` Ermittelt eine Liste von Objekten vom Typ `ArticleComp`.
- ✓ `GetAllCompaniesSvc(): List<Company>::` Ermittelt eine Liste von Objekten vom Typ `Company`.

#### JavaScript implementiert die Klassen/Methoden:

- ✓ Methode (Klasse) `Company`: Verwaltet die Firmendaten (Name, Ort, PLZ). Implementiert die To-String-Methode `ShowMe()`
- ✓ Methode (Klasse) `ArticleComp(...)`: Verwaltet die „erweiterten“ Artikeldaten (ArtID, Menge, Bezeichnung, Preis und Firmenname). Implementiert die To-String-Methode `ShowMe()`
- ✓ Methode `Onload(...)`: Ausgabe der Artikelobjekte in der 1. TextArea. Ausgabe der Artikel-Bezeichnungen im Select-Control ohne Redundanz! Gewährleistet, dass die Firmendaten in der 2. TextArea bei PostBacks erhalten bleiben.
- ✓ Methode `OnShowCompanies(...)`: Ausgabe der Firmen in der 2. TextArea.

### **3. GUI und User-Stories zum zweiten GP**

Die Artikel-Bezeichnungen und die Firmen sollen dargestellt werden, weiter das gesamte Sortiment mit Firma. Durch Auswahl eines Artikels und Auswahl einer Firma soll dieser Artikel, wie er im Gesamt-Sortiment aufgelistet ist aber ohne Firma, in einer Bestellliste ausgegeben werden.

**GUI (Abb. 6):** Sie besteht aus folgenden Controls:

- ✓ 2 HTML-select-Controls: Enthalten die Artikelbezeichnungen bzw. die Firmennamen aus Startseite.
- ✓ 2 ASPX-GridViews: Zeigen das Gesamt-Sortiment mit Firmen aus Startseite bzw. die bestellten Artikel.
- ✓ 2 ASPX-Schalter „Auswahl in die Bestellung“ und „Navigiere zu Final“.

#### **Funktionalitäten / User-Stories:**

Für clientseitige Datenmanipulationen sind immer Felder zu verwenden!

**Beim Laden:** Übernimmt das gesamte Artikelsortiment aus der Start-Seite und gibt es in der GRV aus. Übernimmt die Firmennamen und die Artikel aus der Start-Seite und gibt sie in den Select-Controls aus (Abb. 6).

**ASPX-Schalter „Auswahl in die Bestellung“:** Auszuwählen sind ein Artikel und die Firma zu diesem Artikel; hier Artikel „Heft“ und Firma „Huber“, siehe Abb. 7. Durch Betätigen des Schalters wird zur dieser Firma und diesem Artikel aus dem BL und damit aus der XML-Datei der Artikel ermittelt und in der GRV angezeigt (Abb. 8).

**ASPX-Schalter „Navigiere zu Final“:** Damit wird zur 3. Seite navigiert. Es sind alle 3 Firmen als Parameter zu übergeben.



Abb. 6: GUI der 2. Seite „NextPage“



Abb. 7: Auswahl einer Firma und eines Artikels

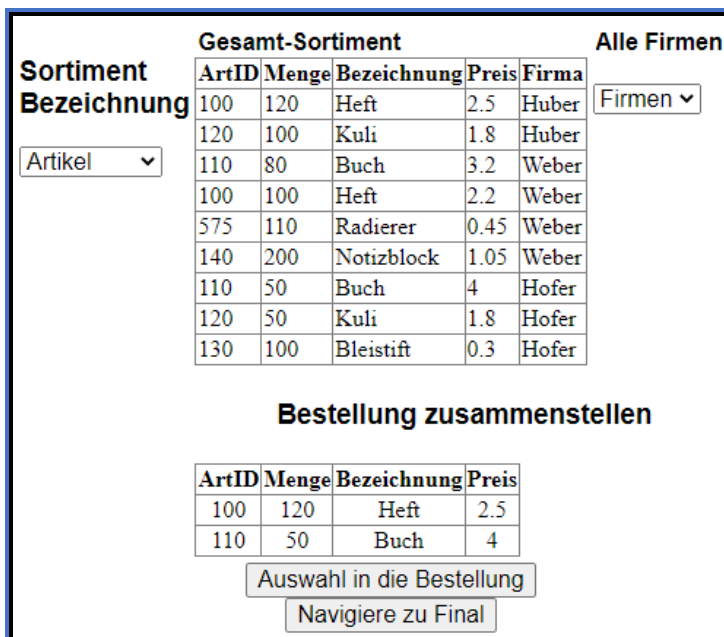


Abb. 8: 2 ausgewählte Artikel aus dem BL

#### 4. Klassen der Bibliothek BL\_Firma, HTML und des WCF

In diesem Beispiel besteht der BL aus den Klassen Firma und DAL. Das WCF greift auf die Klasse Firma zu und die Klasse Firma auf den DAL!

Die Klassen im BL sind bekannt. Folgende Ergänzungen sind zu implementieren:

**Klasse Firma:**

**Methode:**

GetArticleBL(fa, artBez: string): Article:: Ermittelt über den DAL den Artikel zur gegebenen Firma und Artikelbezeichnung. Gibt den Artikel zurück.

**Klasse DAL:**

**Methoden:**

GetArticle(fa, artBez: string): Article:: Ermittelt aus der XML-Datei den Artikel zur gegebenen Firma und Artikelbezeichnung.

#### Das WCF:

✓ GetArticleSvc(fa, artBez: string): Article:: Ermittelt den Artikel über den BL.

#### JavaScript:

✓ Methode onload: Befüllt die Select-Controls.

## 5. GUI und User-Stories zum dritten GP

Hier soll die Bestellung aus dem 2. GP nochmals korrigiert werden können. Der User möchte sämtliche Firmennamen sehen, die im BL vorhanden sind (XML-Datei). Die Bestellung soll alle Artikel vollständig aus dem 2. GP anzeigen, und zwar ArtID, Menge, Bezeichnung und Preis. In dieser Auflistung soll durch Auswahl eines Artikels und unter Angabe eines neuen Preises und einer neuen Menge der Artikel in der Auflistung korrigiert werden.

**GUI** (Abb. 6): Sie besteht aus folgenden Controls:

- ✓ 1 ASPX TextBox: Zeigt die alle Firmen.
- ✓ 1 ASPX-GridView: Zeigen die Bestellung aus GP 2.
- ✓ 1 ASPX-DDL: Enthält das Sortiment
- ✓ 2 HTML-Input (Text): Für Eingabe von Menge und Preis
- ✓ 1 ASPX-Schalter „Ändere Artikel“.

### **Funktionalitäten / User-Stories:**

Beim Laden: Zeigt die Firmen in der TextBox an, weiters die Bestellung in der GRV und die Artikelbezeichnungen in der DDL (Abb. 9).

ASPX-Schalter „Ändere Artikel: Auszuwählen sind ein Artikel und einzugeben sind die neue Menge und der neue Preis zu diesem Artikel. Durch Betätigen des Schalters werden die Änderungen an diesem Artikel in der GRV aktualisiert. Es ist darauf zu achten, dass der Preis und die Menge in den HTML-Feldern nach dem Button-Klick weiterhin sichtbar sind.

ArtID	Menge	Bezeichnung	Preis
100	120	Heft	2.5
110	80	Buch	3.2

Sortiment Preis:

Artikel

Menge:

Ändere Artikel

Abb. 9: Seite beim ersten Aufruf

ArtID	Menge	Bezeichnung	Preis
100	33	Heft	1.1
110	80	Buch	3.2

Sortiment Preis:

Heft

Menge:

Ändere Artikel

Abb. 10: Korrigierter Artikel Heft