

Praktikum 8

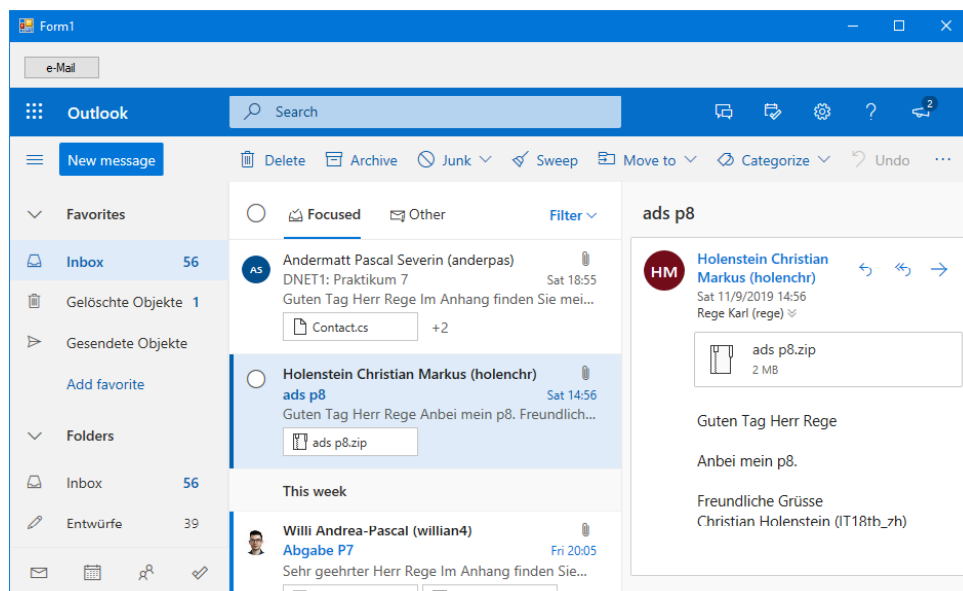
Fortsetzung NextChange

Aufgabe 1 – Internet Explorer Einbindung und Fernsteuerung

Es gibt auch die Möglichkeit, den Kalender zentral auf einem Server zu führen. Dabei stellt sich jedoch das Problem des entfernten Zugriffs. Eine einfache Möglichkeit ist die Darstellung des Kalender in einem Browser Fenster wie es z.B. in Outlook Web Access realisiert wurde, mit der URL:

```
https://outlook.office365.com/owa/?realm=zhaw.ch&exsvurl=1&ll-cc=1033&modurl=0&path=/mail/inboxch
```

Diese Lösung hat jedoch den Nachteil, dass der Benutzer u.U. mehrere Anwendungen gleichzeitig bedienen und sich auch mehrmals anmelden muss. Es gibt jedoch die Möglichkeit in .NET den Browser als sog. ActiveX Element einzubinden und fernzusteuern (siehe Script). In Visual Studio steht bereits ein vorbereitetes WebBrowser Control zur Verfügung. Hinweis: Anchors setzen, damit das Browser Control frei bewegbar wird (Docking wird disabled).



Nach dem Einbinden erfolgt der Aufruf einer Seite einfach mittels:

```
webBrowser1.Navigate (url);
```

Da jedoch der Outlook Passwort geschützt ist (man kann nur auf den eigenen Kalender zugreifen), wird man beim ersten Aufruf auf eine Login-Seite verwiesen, in der man das Passwort eingeben muss. Falls der OWA Teil einer laufenden Anwendung sein soll ist dies lästig. Um diese Passwort-Eingabe zu automatisieren, möchten wir das Login Formular deshalb automatisch ausfüllen (mit <StudentenName> und Passwort). Dafür müssen wir einen DocumentComplete Handler folgendermassen anmelden (vor dem Navigate-Aufruf) .

```
webBrowser1.DocumentCompleted += new  
WebBrowserDocumentCompletedEventHandler(webBrowser1_DocumentCompleted);
```

Die mitgegebene Methode wird nach erfolgreichem Laden der Seite aufgerufen.

Sie muss folgende Signatur besitzen:

```
private void webBrowser1_DocumentCompleted(Object sender,  
WebBrowserDocumentCompletedEventArgs e)
```

In dieser Methode kann mittels:

```
e.Url.ToString().StartsWith("https://sts.zhaw.ch/adfs/ls/")
```

überprüft werden ob wir uns wirklich auf der Login Seite befinden.

Wurde die Seite vollständig geladen können wir mittels:

```
HtmlDocument doc = webBrowser1.Document;
```

auf das HTML Dokument zugreifen. Der relevante Teil sieht folgendermassen aus:

```
<div id="formsAuthenticationArea">  
  <div id="userNameArea">  
    <label id="userNameInputLabel" for="userNameInput" class="hidden">User Account</label>  
    <input id="userNameInput" name="UserName" type="email" value="" tabindex="1" class="text fullWidth"  
      spellcheck="false" placeholder="someone@example.com" autocomplete="off"/>  
  </div>  
  
  <div id="passwordArea">  
    <label id="passwordInputLabel" for="passwordInput" class="hidden">Password</label>  
    <input id="passwordInput" name="Password" type="password" tabindex="2" class="text fullWidth"  
      placeholder="Password" autocomplete="off"/>  
  </div>  
  
  <div id="kmsiArea" style="display:none">  
    <input type="checkbox" name="Kmsi" id="kmsiInput" value="true" tabindex="3" />  
    <label for="kmsiInput">Keep me signed in</label>  
  </div>  
  
  <div id="submissionArea" class="submitMargin">  
    <span id="submitButton" class="submit" tabindex="4" role="button"  
      onKeyPress="if (event && event.keyCode == 32) Login.submitLoginRequest();"   
      onclick="return Login.submitLoginRequest();">Sign in</span>
```

```
</div>
</div>
```

Setzen von Feldern

Die Textfelder des Passwort-Formulars greifen wir folgendermassen zu:

```
HtmlElement e1 = doc.GetElementById("userNameInputLabel");
e1.SetAttribute("value", "<StudentenName>");
```

Drücken des Knopfes

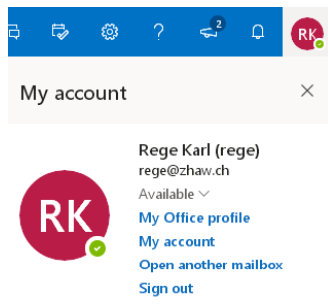
Das Formular wird durch einfaches aktivieren des Submit-Knopfes abgeschickt:

```
foreach (HtmlElement elem in doc.GetElementsByTagName("div"))
{
    if (elem.GetAttribute("role") == "button")
        {elem.InvokeMember("click");}
}
```

Beim Nachfrage Dialog ob Sie angemeldet bleiben wollen, setzen Sie false und aktivieren Sie ebenfalls den Knopf

Logout

Wenn Sie die Applikation beenden, dann soll ein sauberes Logout durchgeführt werden



Einfach durch den Aufruf folgender URL

```
https://login.microsoftonline.com/common/oauth2/logout?ui_locales=en-US&mkt=en-US
```

Aufgabe 2 – Low Level Programmierung

Für einen Kalender ist die exakte Uhrzeit wichtig. Leider sind die Uhren der meisten Computer ziemlich ungenau und sie sollte deshalb periodisch überprüft und allenfalls korrigiert werden. Es gibt im Internet für diesen Zweck sogenannte NIST Internet Time Server, die auf Anfrage die exakte Zeit liefern. Eine Zusammenstellung der verfügbaren Time Server findet man unter <http://tf.nist.gov/tf-cgi/servers.cgi#>

Leider kann in .NET nicht direkt die Uhr der eigenen Maschine gesetzt werden. Es gibt jedoch in der Kernel32.dll einen entsprechenden Aufruf; siehe Zusatzdokument (pdf)

Schreiben Sie eine Methode, welche die Uhr mit der NIST Zeit synchronisiert.

Hinweis: es muss folgendes struct mit den zu C++ analogen Datentypen als **ref** dem Aufruf übergeben werden.

```
public struct SystemTime // win32 struct
{
    public short wYear;
    public short wMonth;
    public short wDayOfWeek;
    public short wDay;
    public short wHour;
    public short wMinute;
    public short wSecond;
    public short wMilliseconds;
};
```

Hinweise

- Sie müssen den Aufruf unter Admin Rechten ausführen; Rechts-Klick und runas
- Oder mittels Anleitung <http://www.codeproject.com/KB/cs/zetaimpersonator.aspx> (Quelle finden Sie auf der .NET Seite).

Aufgabe 3 – COM Einbindung und Fernsteuerung (optional)

Wandeln Sie Ihre NIST Klasse in eine COM Komponente um; diese soll dann via einem VBA Script in Excel aufgerufen werden und das Resultat des Aufrufs soll in eine Excel Zelle gespeichert werden.

Alternativ kann der Aufruf der COM Komponenten und Excel via ein Python Script erfolgen. D.h. das Python Script ruft zuerst die COM Komponente auf und speichert dann das Resultat in ein Excel Document.

Hinweise:

- Verwenden Sie die pywin32 Bibliothek; sowohl Office als auch Python müssen in derselben Architektur (32/64 Bit) vorliegen
- <https://sourceforge.net/projects/pywin32/files/pywin32/>