Consultingwerk
software architecture and development

The Browser Within Your GUI: Seamlessly Integrating Modern Web Browsers in your OpenEdge Application

Mike Fechner
Consultingwerk

The full stack modernization framework
SmartComponent Library
by Consultingwerk

Mike Fechner

- Director, Lead Modernization Architect and Product Manager, Architect of the SmartComponent Library and WinKit
- Specialized on object-oriented design, software architecture, desktop user interfaces and web technologies
- 35 years of Progress experience (V5 … OE12)
- Active member of the OpenEdge community
- Frequent speaker at OpenEdge related conferences around the world

Consultingwerk Application Modernization Solutions

Progress Partner
PREMIER SERVICE DELIVERY
#ProgressProud

- Independent IT consulting organization
- Focusing on **OpenEdge** and **related technology**
- Located in Cologne, Germany, subsidiaries in UK, USA and Romania
- Customers in Europe, North America, Australia and South Africa
- Vendor of developer tools and consulting services
- Specialized in GUI for .NET, Angular, OO, Software Architecture, Application Integration
- Experts in OpenEdge Application Modernization

SmartComponent Library

- The tool to improve **developer productivity**
- Full stack **modernization framework** for OpenEdge – focusing on strong architectural foundation
- Backend as **future-proof** home for business logic
- Relational and **object-relational** (ORM)
- **RESTful** out of the box
- Multiple **user interface options**: Desktop, Web and Mobile
- **Application Framework**: Authentication, Localization, Menu, Workflows, …
- **Integration** with existing OpenEdge applications and frameworks

An urgent reminder!

- OpenEdge 11.7 has retired **9 days** ago!
- Windows 10 will retire October 25th 2025
- Windows 12 expected around that time
- No security fixes for OpenEdge 11.7 after April 1st
- No bug fixes for OpenEdge 11.7 after April 1st
- No new platform certification after April 1st
- No formal Windows 12 support for OpenEdge 11.7
- …

Consultingwerk
software architecture and development

Agenda

- **CefSharp**
- CefSharp vs. Microsoft WebView2 SDK
- Usage of the Web Browser Control in your application
- Hybrid application architecture
- Navigating within a SPA (Angular)
- Interaction with the SPA (Angular)
- Authentication
- Integrating local web content as pretty controls

CEF

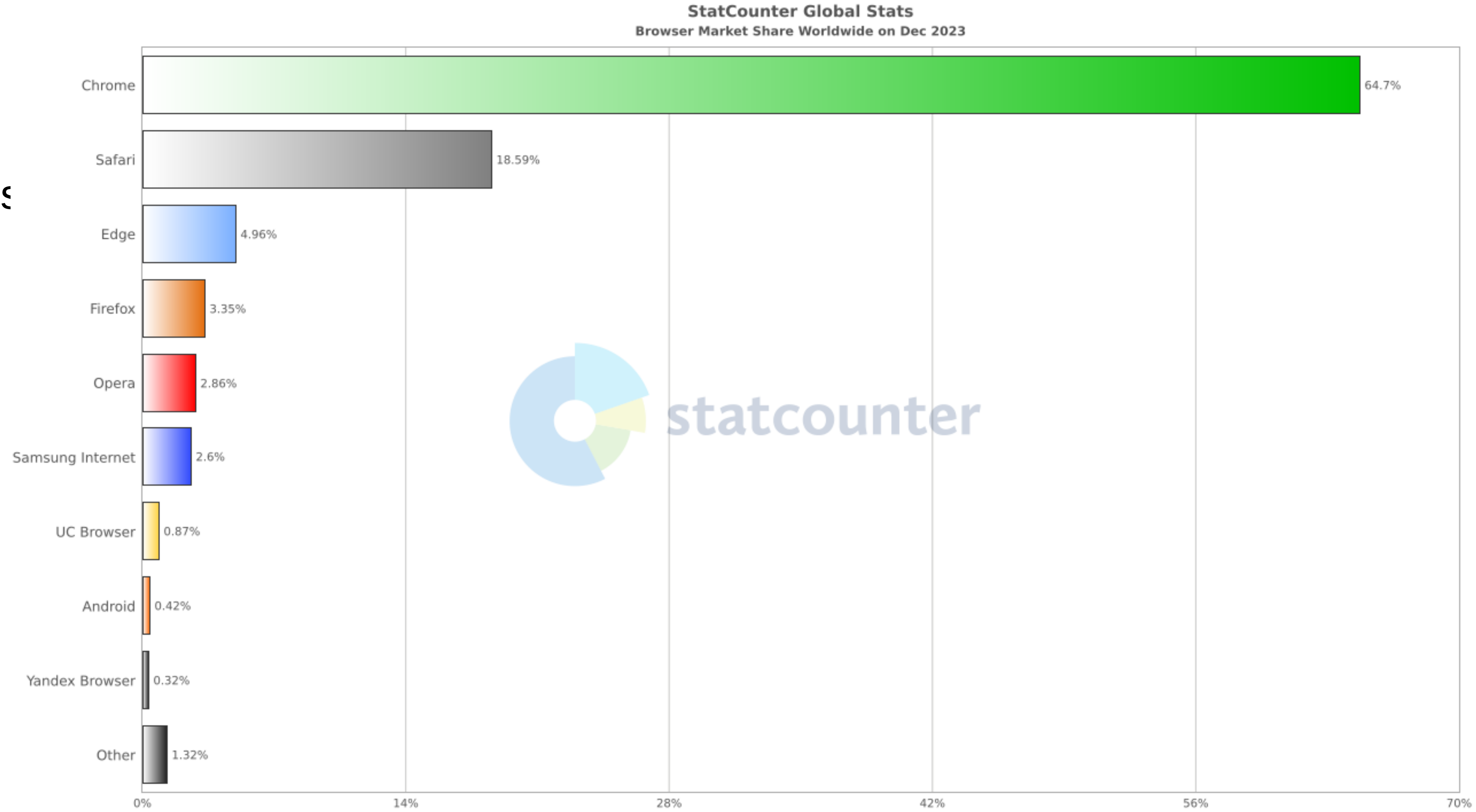
- **C**hromium
  - Based on open-source code base of the Google Chrome Web Browser
- **E**mbedded
  - Browser component to be embedded into your application
- **F**ramework
  - Allows developing our own web browser application
  - But not an independent web browser application

CefSharp

- Chromium Embedded Framework (CEF)
- Open-source project, launched in 2009
- BSD license
- CefSharp is a popular open-source project embedding CEF into .NET applications
- Based on Google‘s open-source Chromium, the foundation of Chrome
- CEF is developed in C++; CefSharp is developed in C#
- **WinForms** and WPF Web Browser Control
- OpenEdge GUI application can use the WinForms Control

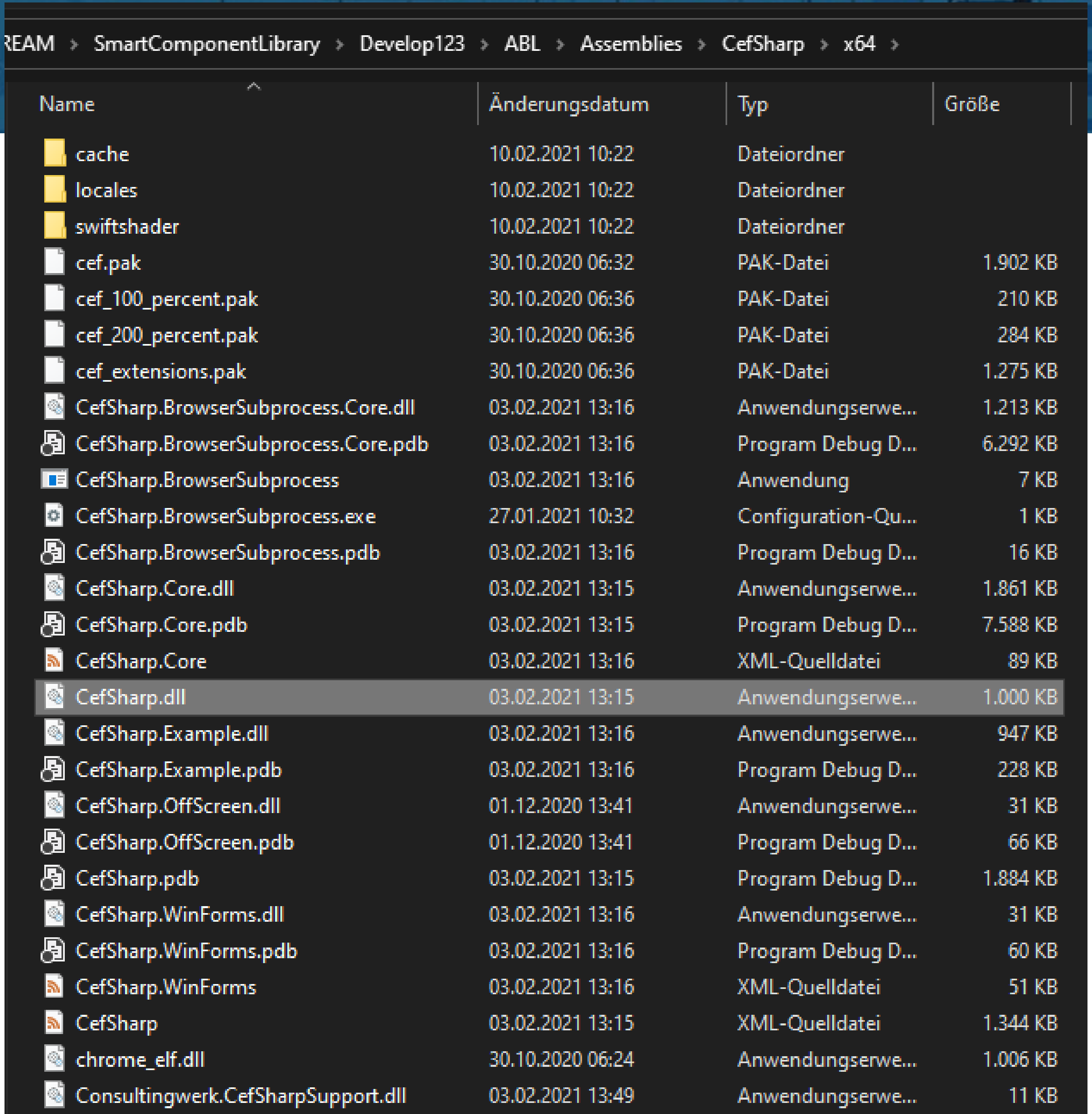
Web Browser market share 2023

- Source: Statcounter
- https://en.wikipedia.org/wiki/Usage_share_of_web_browsers



CefSharp

- CEF/CefSharp defacto standard to embed web browser into applications
- Used in applications and games such as: Adobe Autodesk, Amazon, Battle.net, Bitdefender, BlueStacks, Evernote, Facebook Messenger, GTA Online, Kaspersky, Intel League of Legends, MATLAB, Power BI, QuarkXPress, Second Life, Solidworks, Spotify, Steam, Unity3D, Unreal Engine etc.



REAM › SmartComponentLibrary › Develop123 › ABL › Assemblies › CefSharp › x64 ›

Name	Änderungsdatum	Typ	Größe
cache	10.02.2021 10:22	Dateiordner	
locales	10.02.2021 10:22	Dateiordner	
swiftshader	10.02.2021 10:22	Dateiordner	
cef.pak	30.10.2020 06:32	PAK-Datei	1.902 KB
cef_100_percent.pak	30.10.2020 06:36	PAK-Datei	210 KB
cef_200_percent.pak	30.10.2020 06:36	PAK-Datei	284 KB
cef_extensions.pak	30.10.2020 06:36	PAK-Datei	1.275 KB
CefSharp.BrowserSubprocess.Core.dll	03.02.2021 13:16	Anwendungserwe...	1.213 KB
CefSharp.BrowserSubprocess.Core.pdb	03.02.2021 13:16	Program Debug D...	6.292 KB
CefSharp.BrowserSubprocess	03.02.2021 13:16	Anwendung	7 KB
CefSharp.BrowserSubprocess.exe	27.01.2021 10:32	Configuration-Qu...	1 KB
CefSharp.BrowserSubprocess.pdb	03.02.2021 13:16	Program Debug D...	16 KB
CefSharp.Core.dll	03.02.2021 13:15	Anwendungserwe...	1.861 KB
CefSharp.Core.pdb	03.02.2021 13:15	Program Debug D...	7.588 KB
CefSharp.Core	03.02.2021 13:16	XML-Quelldatei	89 KB
CefSharp.dll	03.02.2021 13:15	Anwendungserwe...	1.000 KB
CefSharp.Example.dll	03.02.2021 13:16	Anwendungserwe...	947 KB
CefSharp.Example.pdb	03.02.2021 13:16	Program Debug D...	228 KB
CefSharp.OffScreen.dll	01.12.2020 13:41	Anwendungserwe...	31 KB
CefSharp.OffScreen.pdb	01.12.2020 13:41	Program Debug D...	66 KB
CefSharp.pdb	03.02.2021 13:15	Program Debug D...	1.884 KB
CefSharp.WinForms.dll	03.02.2021 13:16	Anwendungserwe...	31 KB
CefSharp.WinForms.pdb	03.02.2021 13:16	Program Debug D...	60 KB
CefSharp.WinForms	03.02.2021 13:16	XML-Quelldatei	51 KB
CefSharp	03.02.2021 13:15	XML-Quelldatei	1.344 KB
chrome_elf.dll	30.10.2020 06:24	Anwendungserwe...	1.006 KB
Consultingwerk.CefSharpSupport.dll	03.02.2021 13:49	Anwendungserwe...	11 KB

Embedded Browser

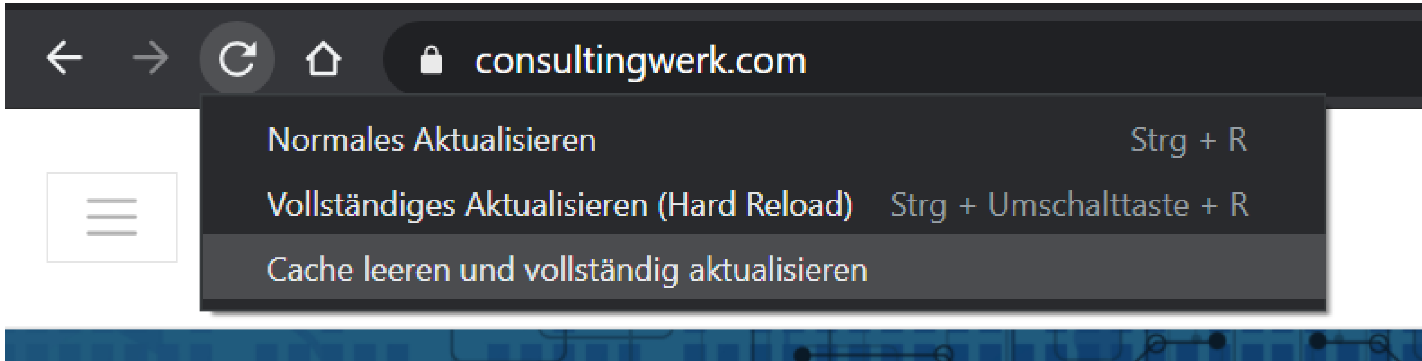
- C++ and C# DLL‘s in application folder
- Browser Sub-Process in application folder
- No dependencies on existing Google Chrome installation
- Only dependencies are Visual C++ Runtime and .NET Framework

Benefits of an embedded web browser

- Close integration with your own application possible (as demoed later in this presentation)
- Modern Web Browser
- Release Upgrades under your own control
  - No dependency on the end user
  - Delay browser updates when required
    - No surprises through sudden introduction of Secure-Cookie requirements
    - No surprises through sudden revocation of SSL root certificates
    - Test of the own web application with new web browser release prior to rollout of updated embedded web browser component

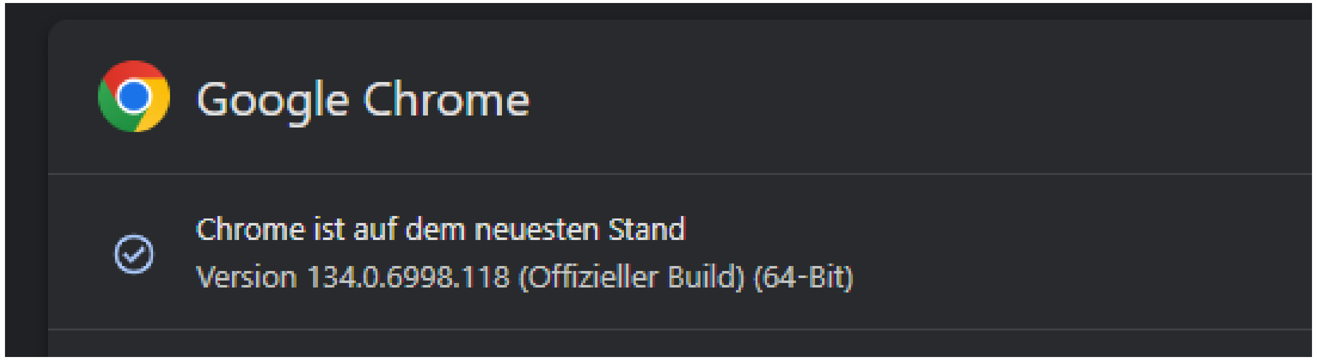
Full control on web browser cache

- Web browser cache of importance for application performance
- Web developer to tester: “have you emptied the browser cache?” – “have you really emptied the browser cache completely?” – “open the browser debugger tools and right click the refresh button!”
- CefSharp config controls the cache folder
- Can be emptied when needed programmatically, e.g. when the web application is updated
- CefSharp can avoid cache on disk completely


consultingwerk.com
Normales Aktualisieren
Strg + R
Vollständiges Aktualisieren (Hard Reload)
Strg + Umschalttaste + R
Cache leeren und vollständig aktualisieren

Stable versions available briefly after Google Chrome

- As per 19.03.2025

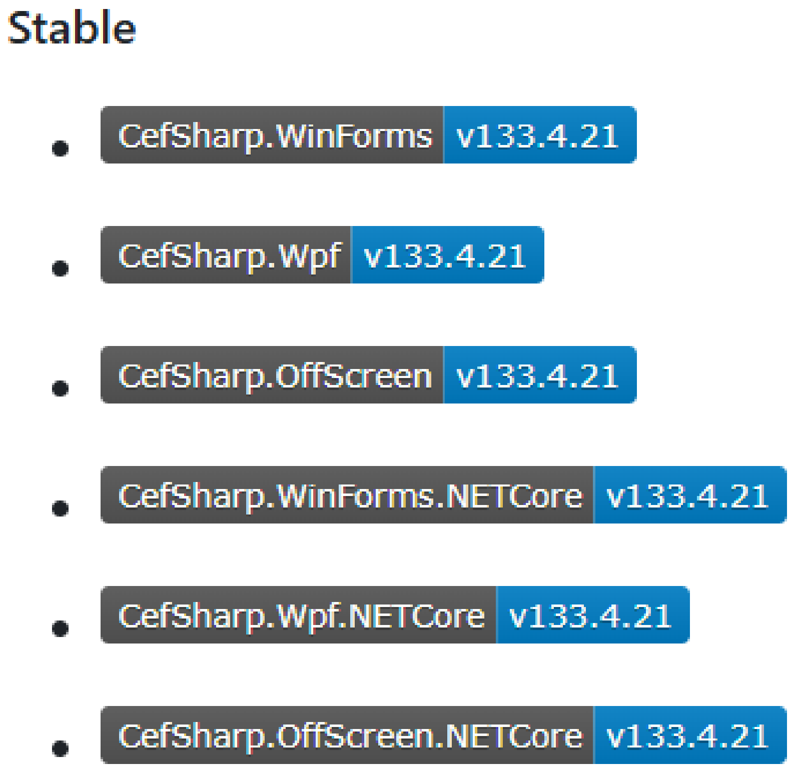


Google Chrome

Chrome ist auf dem neuesten Stand
Version 134.0.6998.118 (Offizieller Build) (64-Bit)

- Chrome Version 134.0 was released on March 4th 2025

CefSharp Builds:



Stable

- CefSharp.WinForms v133.4.21
- CefSharp.Wpf v133.4.21
- CefSharp.OffScreen v133.4.21
- CefSharp.WinForms.NETCore v133.4.21
- CefSharp.Wpf.NETCore v133.4.21
- CefSharp.OffScreen.NETCore v133.4.21

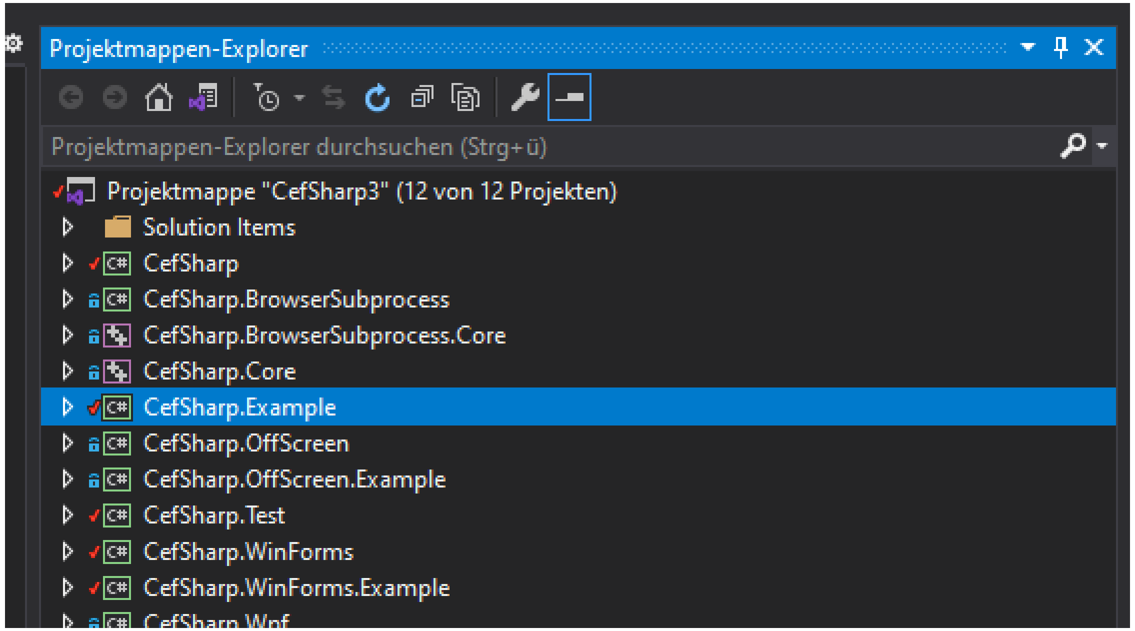
Download CefSharp

- Source-code via git clone of https://github.com/cefsharp/CefSharp
- Binaries https://www.nuget.org/packages/CefSharp.WinForms/

- We‘re using the source-code – debugging sample application helpful for understanding usage of web browser control
- Contains sample application as reference
- For integration into OpenEdge we have developed our own Support .NET Assembly
  - for Event-Synchronization out of background threads
  - Settings wrapper, etc.

Visual Studio

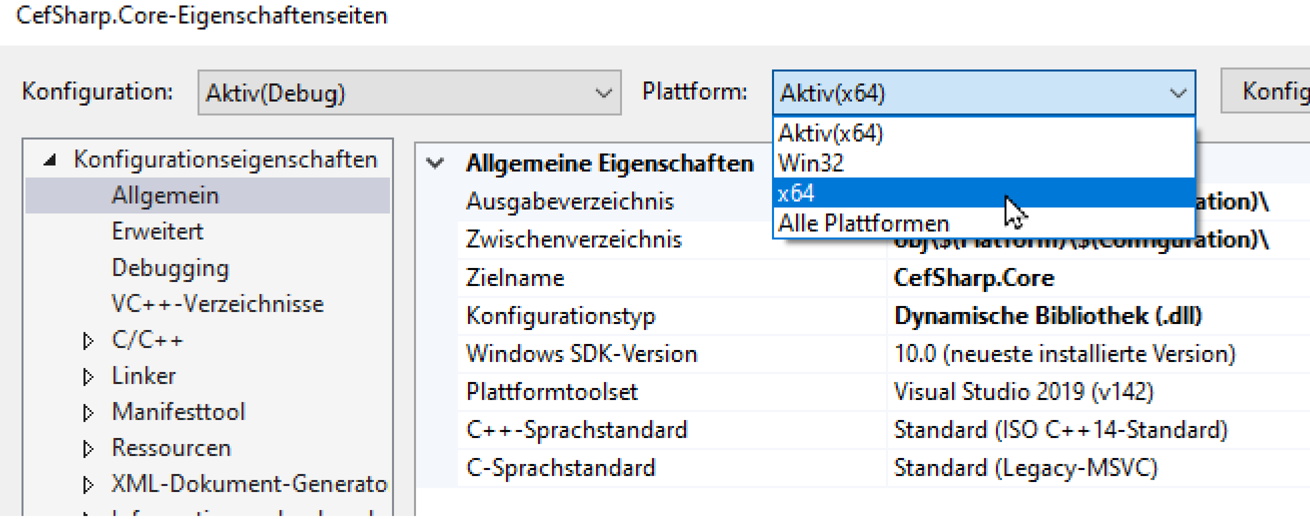
- CefSharp Solution
- Pretty uncomplicated to build! Even in the Visual Studio Community Edition
- Projects must be built separately for 64- or 32-bit runtime (prowin.exe or prowin32.exe), separate set of DLL‘s



Projektmappen-Explorer

Projektmappen-Explorer durchsuchen (Strg+ü)

- Projektmappe "CefSharp3" (12 von 12 Projekten)
  - Solution Items
  - CefSharp
  - CefSharp.BrowserSubprocess
  - CefSharp.BrowserSubprocess.Core
  - CefSharp.Core
  - CefSharp.Example
  - CefSharp.OffScreen
  - CefSharp.OffScreen.Example
  - CefSharp.Test
  - CefSharp.WinForms
  - CefSharp.WinForms.Example
  - CefSharp.Wpf



CefSharp.Core-Eigenschaftenseiten

Konfiguration: Aktiv(Debug)

Plattform: Aktiv(x64)

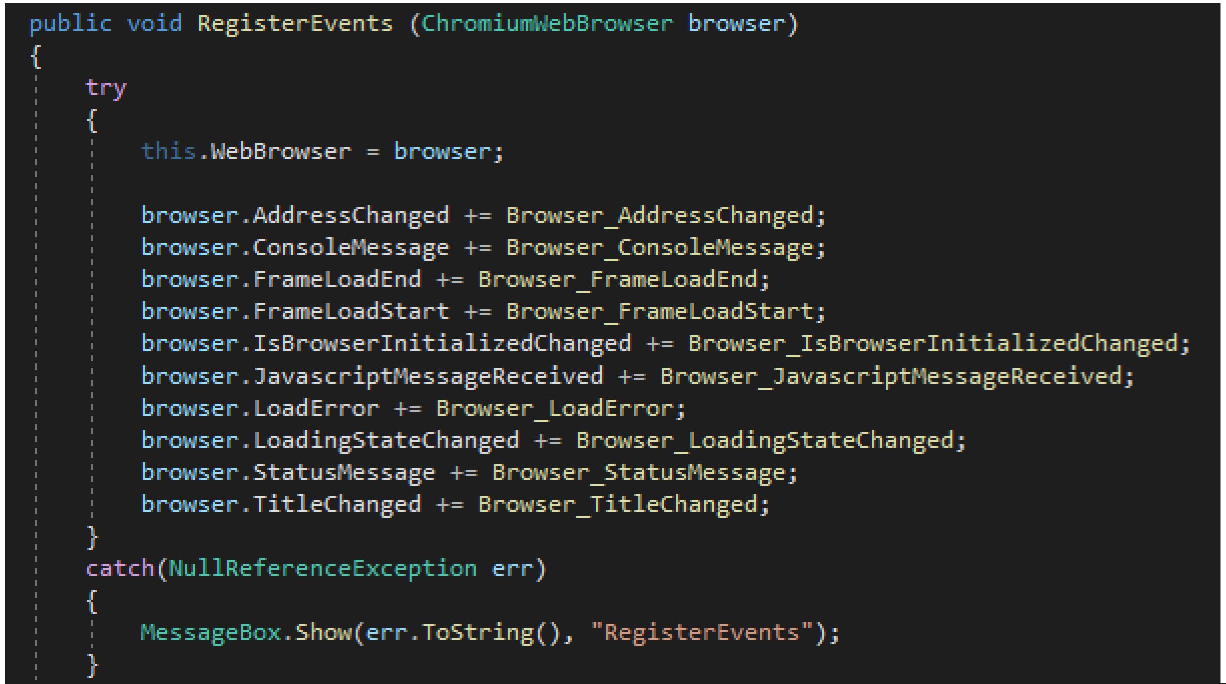
- Aktiv(x64)
- Win32
- x64
- Alle Plattformen

Konfig

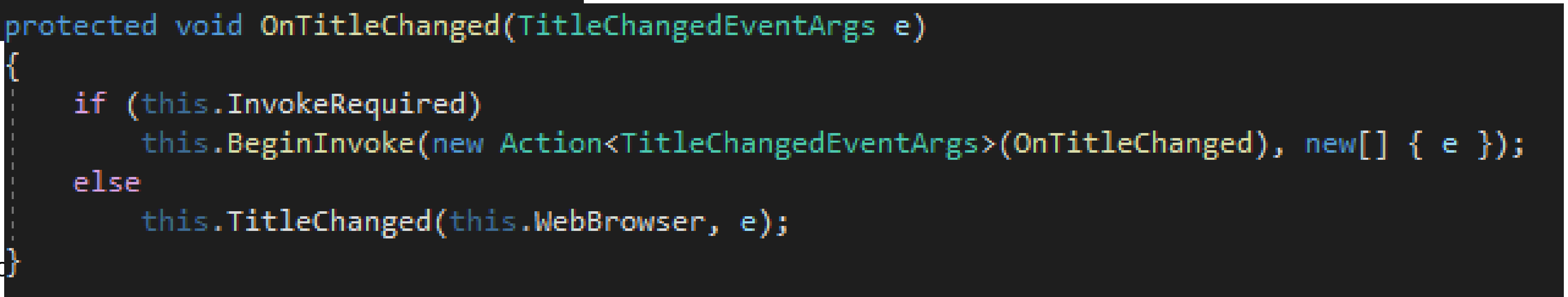
- Konfigurationseigenschaften
  - Allgemein
  - Erweitert
  - Debugging
  - VC++-Verzeichnisse
  - C/C++
  - Linker
  - Manifesttool
  - Ressourcen
  - XML-Dokument-Generato

Allgemeine Eigenschaften	
Ausgabeverzeichnis	ation)\
Zwischenverzeichnis	obj\$(Platform)\$(Configuration)\
Zielname	CefSharp.Core
Konfigurationstyp	Dynamische Bibliothek (.dll)
Windows SDK-Version	10.0 (neueste installierte Version)
Plattformtoolset	Visual Studio 2019 (v142)
C++-Sprachstandard	Standard (ISO C++14-Standard)
C-Sprachstandard	Standard (Legacy-MSVC)

Thread-synchronization of Events (WinForms Control)



```
public void RegisterEvents (ChromiumWebBrowser browser)
{
    try
    {
        this.WebBrowser = browser;

        browser.AddressChanged += Browser_AddressChanged;
        browser.ConsoleMessage += Browser_ConsoleMessage;
        browser.FrameLoadEnd += Browser_FrameLoadEnd;
        browser.FrameLoadStart += Browser_FrameLoadStart;
        browser.IsBrowserInitializedChanged += Browser_IsBrowserInitializedChanged;
        browser.JavascriptMessageReceived += Browser_JavascriptMessageReceived;
        browser.LoadError += Browser_LoadError;
        browser.LoadingStateChanged += Browser_LoadingStateChanged;
        browser.StatusMessage += Browser_StatusMessage;
        browser.TitleChanged += Browser_TitleChanged;
    }
    catch(NullReferenceException err)
    {
        MessageBox.Show(err.ToString(), "RegisterEvents");
    }
}
```



```
protected void OnTitleChanged(TitleChangedEventArgs e)
{
    if (this.InvokeRequired)
        this.BeginInvoke(new Action<TitleChangedEventArgs>(OnTitleChanged), new[] { e });
    else
        this.TitleChanged(this.WebBrowser, e);
}
```

Additional CefSharp features

- Integration with Google Chrome Website Debugger Console
  - Browser Log Console
  - Tracing of http-Requests
  - Debugging of JavaScript Code
- Head-less Browsing support, usage of the browser without user interface
  - Used as foundation of unit-test frameworks for websites
- Free choice of the „User-Agent“ (browser identification)
- Typical controls such as Back-Button, address bar are not part of CefSharp and need to be implemented as part of the application

Consultingwerk
software architecture and development

Agenda

- CefSharp
- **CefSharp vs. Microsoft WebView2 SDK**
- Usage of the Web Browser Control in your application
- Hybrid application architecture
- Navigating within a SPA (Angular)
- Interaction with the SPA (Angular)
- Authentication
- Integrating local web content as pretty controls

WebView

- Classic WebView: Active X Control or .NET Control based on Internet Explorer
- Commonly used in OpenEdge applications, also to view PDF files
- No support for modern web applications
- Version dependent on installed Internet Explorer
- Retired, certainly since the release of Microsoft Edge

WebView2

- Based on Google‘s Chromium like Microsoft Edge
- Similar integration features as CefSharp
- **Seems** a bit simpler in usage …
- No web-site Debugger available, just a browser console
- Not wide-spread, WebView2 .NET available since mid January ´21, C++ component (not usable in OpenEdge) since end of October ´20
- No embedded engine, installed as OS component
- Dependency on OS installed Edge Browser is a pro and a con
- No control on browser-version

WebView2

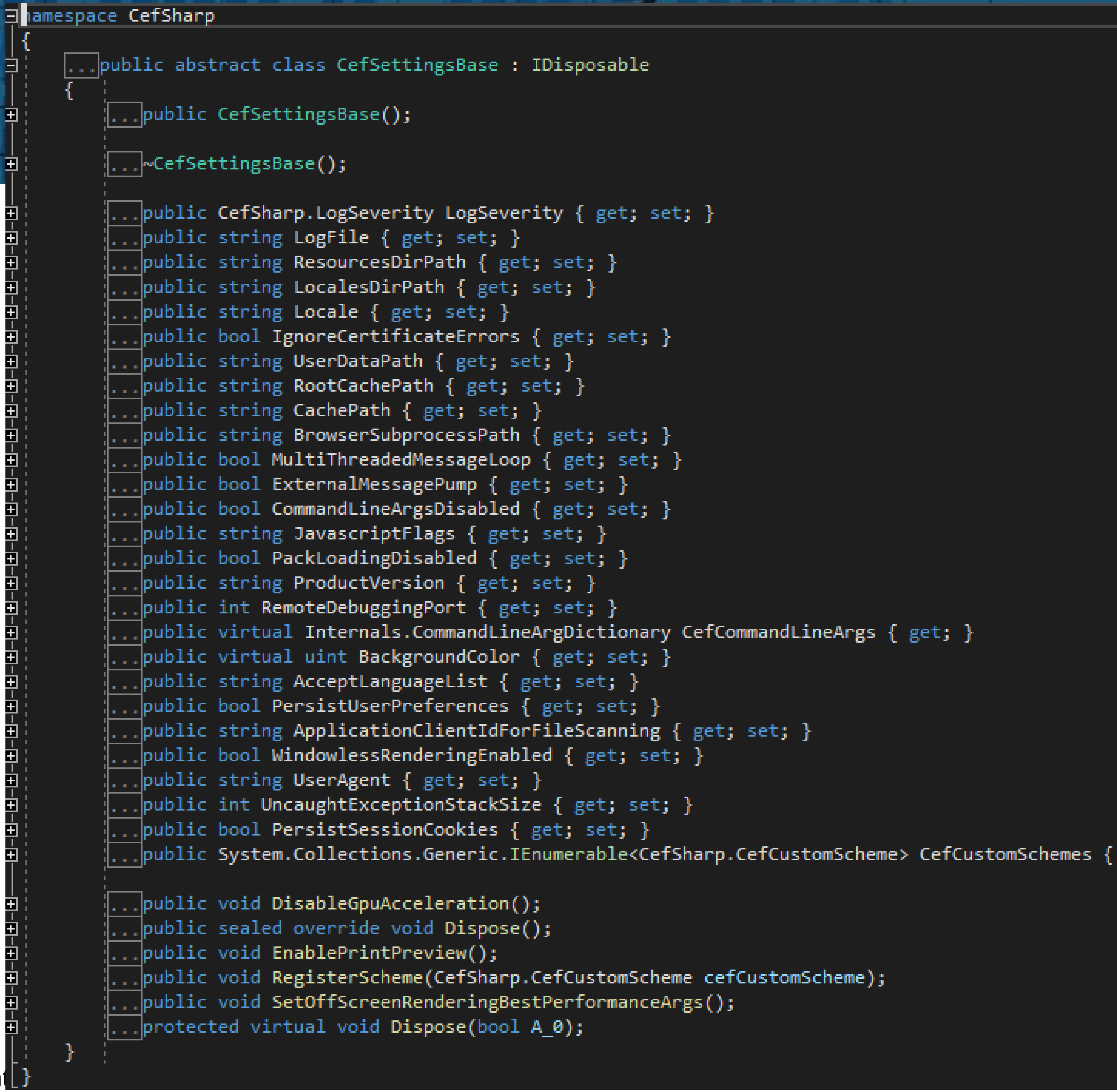
- Limitations in
  - Launching JavaScript code from the host application
  - DOM manipulation from the host application (change HTML without page reload)
    - https://github.com/ChromiumDotNet/WebView2.DevTools.Dom
    - Currently trying adoption of this SDK with another customer – has some challenges

Agenda

- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- **Usage of the Web Browser Control in your application**
- Hybrid application architecture
- Navigating within a SPA (Angular)
- Interaction with the SPA (Angular)
- Authentication
- Integrating local web content as pretty controls

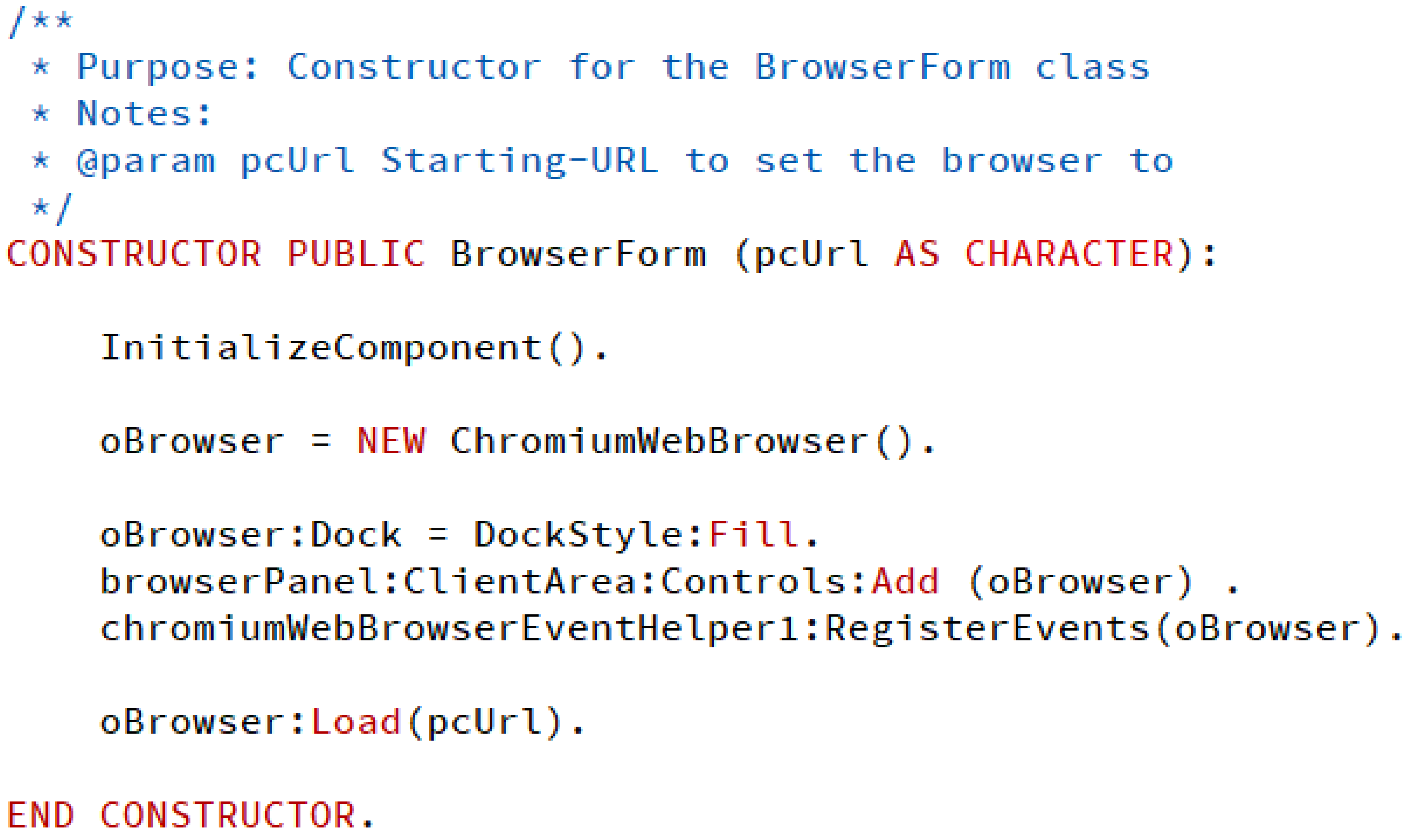
Usage of CefSharp

- Initialization using CefSettings class



```
namespace CefSharp
{
    public abstract class CefSettingsBase : IDisposable
    {
        public CefSettingsBase();

        ~CefSettingsBase();

        public CefSharp.LogSeverity LogSeverity { get; set; }
        public string LogFile { get; set; }
        public string ResourcesDirPath { get; set; }
        public string LocalesDirPath { get; set; }
        public string Locale { get; set; }
        public bool IgnoreCertificateErrors { get; set; }
        public string UserDataPath { get; set; }
        public string RootCachePath { get; set; }
        public string CachePath { get; set; }
        public string BrowserSubprocessPath { get; set; }
        public bool MultiThreadedMessageLoop { get; set; }
        public bool ExternalMessagePump { get; set; }
        public bool CommandLineArgsDisabled { get; set; }
        public string JavascriptFlags { get; set; }
        public bool PackLoadingDisabled { get; set; }
        public string ProductVersion { get; set; }
        public int RemoteDebuggingPort { get; set; }
        public virtual Internals.CommandLineArgDictionary CefCommandLineArgs { get; }
        public virtual uint BackgroundColor { get; set; }
        public string AcceptLanguageList { get; set; }
        public bool PersistUserPreferences { get; set; }
        public string ApplicationClientIdForFileScanning { get; set; }
        public bool WindowlessRenderingEnabled { get; set; }
        public string UserAgent { get; set; }
        public int UncaughtExceptionStackSize { get; set; }
        public bool PersistSessionCookies { get; set; }
        public System.Collections.Generic.IEnumerable<CefSharp.CefCustomScheme> CefCustomSchemes {

        public void DisableGpuAcceleration();
        public sealed override void Dispose();
        public void EnablePrintPreview();
        public void RegisterScheme(CefSharp.CefCustomScheme cefCustomScheme);
        public void SetOffScreenRenderingBestPerformanceArgs();
        protected virtual void Dispose(bool A_0);
    }
}
```

Code-Sample



```
/**
 * Purpose: Constructor for the BrowserForm class
 * Notes:
 * @param pcUrl Starting-URL to set the browser to
 */
CONSTRUCTOR PUBLIC BrowserForm (pcUrl AS CHARACTER):

    InitializeComponent().

    oBrowser = NEW ChromiumWebBrowser().

    oBrowser:Dock = DockStyle:Fill.
    browserPanel:ClientArea:Controls:Add (oBrowser) .
    chromiumWebBrowserEventHelper1:RegisterEvents(oBrowser).

    oBrowser:Load(pcUrl).

END CONSTRUCTOR.
```

Demo

- DHL tracking with passing of parcel tracking ID
- PDF Viewer

Consultingwerk
software architecture and development

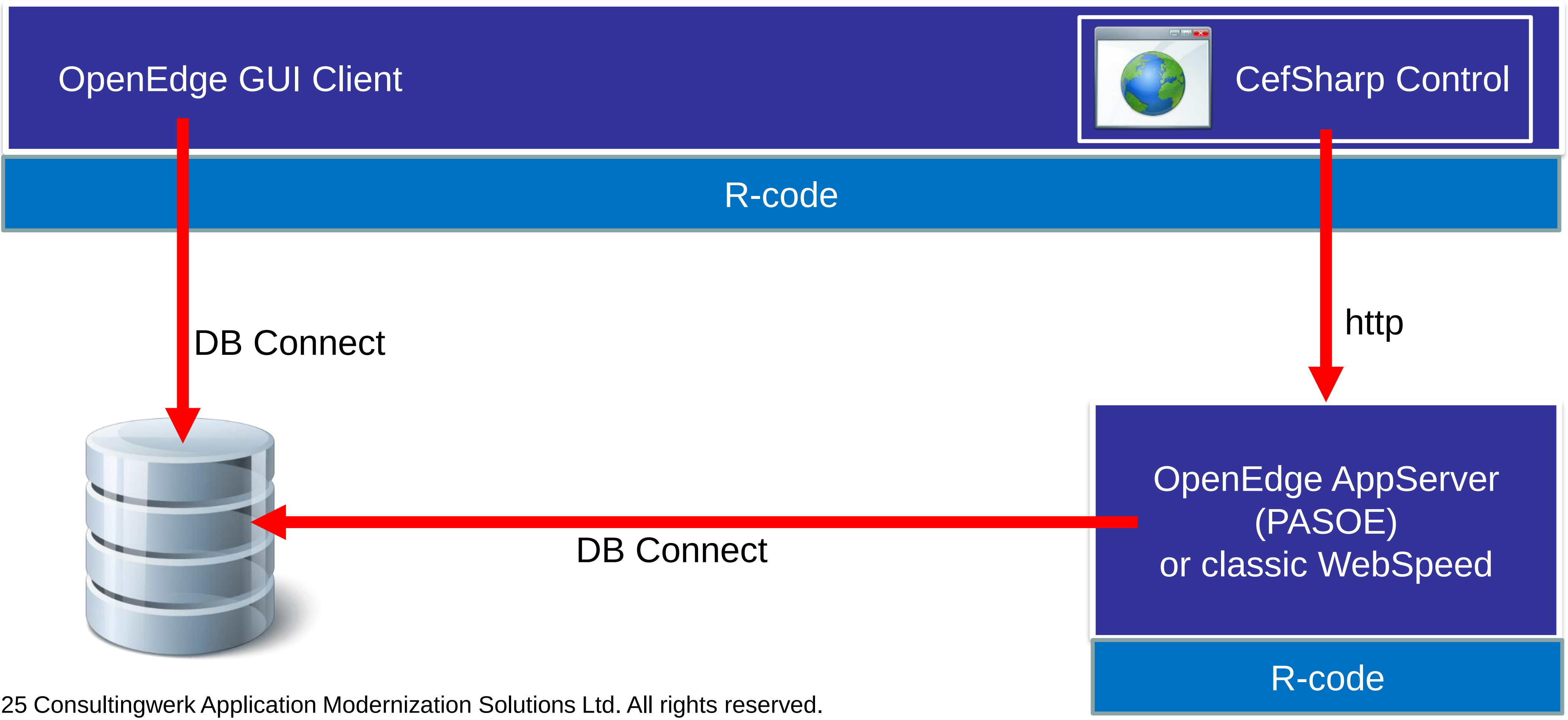
Agenda

- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- Usage of the Web Browser Control in your application
- **Hybrid application architecture**
- Navigating within a SPA (Angular)
- Interaction with the SPA (Angular)
- Authentication

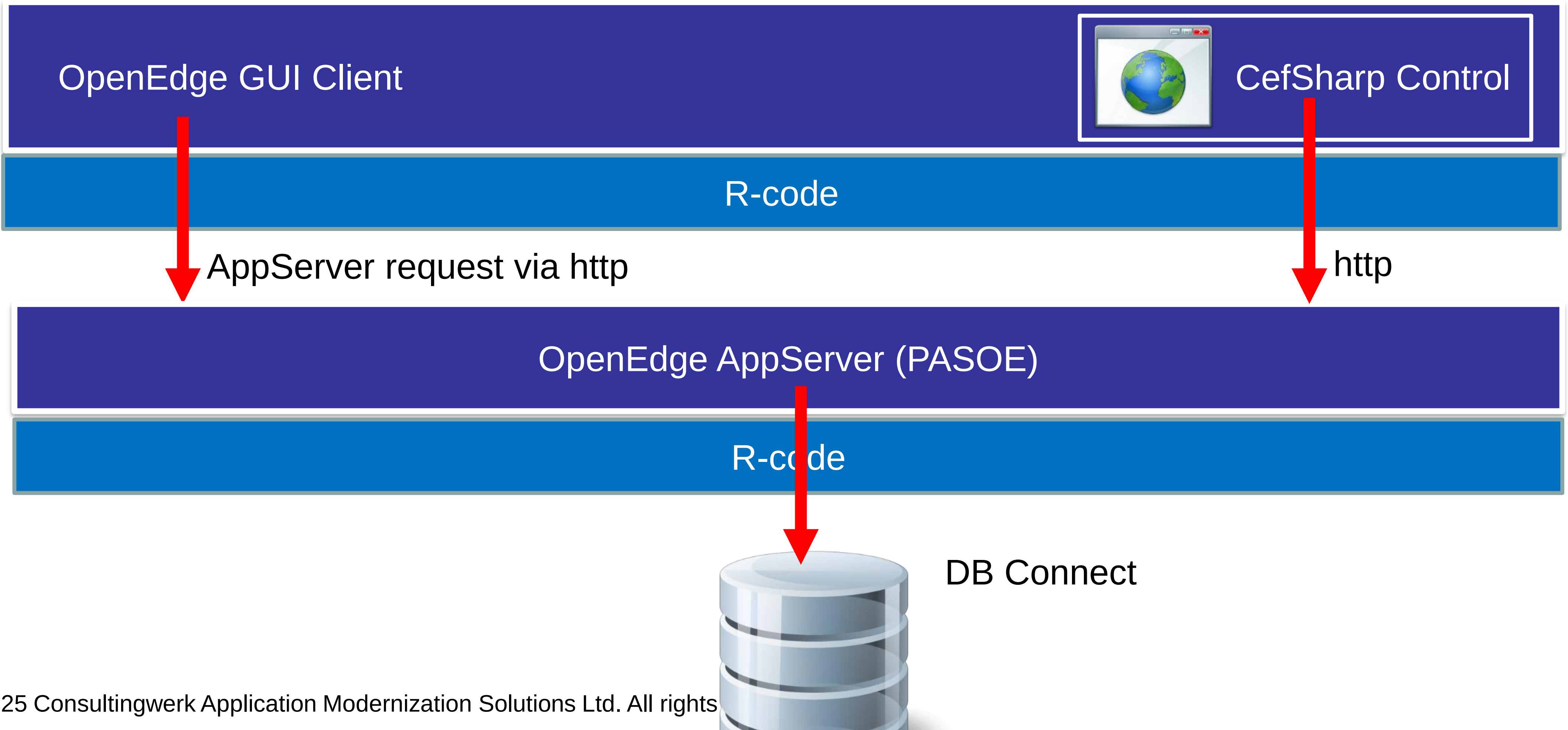
Hybrid application architecture

- Assumption ABL application migrated to modern web application
  - Angular
  - Vue
  - React
  - …
- PASOE backend for web application
- ABL application may be fat client (with DB) or thin client (with AppServer) or mixed

Architecture Integration in Fat Client OpenEdge GUI



Architecture Integration in Thin Client OpenEdge GUI



Agenda

- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- Usage of the Web Browser Control in your application
- Hybrid application architecture
- **Navigating within a SPA (Angular)**
- Interaction with the SPA (Angular)
- Authentication
- Integrating local web content as pretty controls

Demo

- Embedding Angular Web Application in OpenEdge Desktop
- Toolbar integration
  - Toolbar in Angular application replaced by Toolbar in Desktop
  - Toolbar state synchronized

Navigation within an Angular SPA

- Modern web applications typically **s**ingle **p**age web **a**pplication
- Navigation between application screens via *Router* component
- Navigation typically performed via JavaScript functionality within the SPA
- Screen layout and client-side code (JavaScript/TypeScript) either already loaded in browser or load on demand
- Screens can also be addressed via URL

- http://localhost:3000/#/(view:**user-maintenance**)

Hash Based Routing

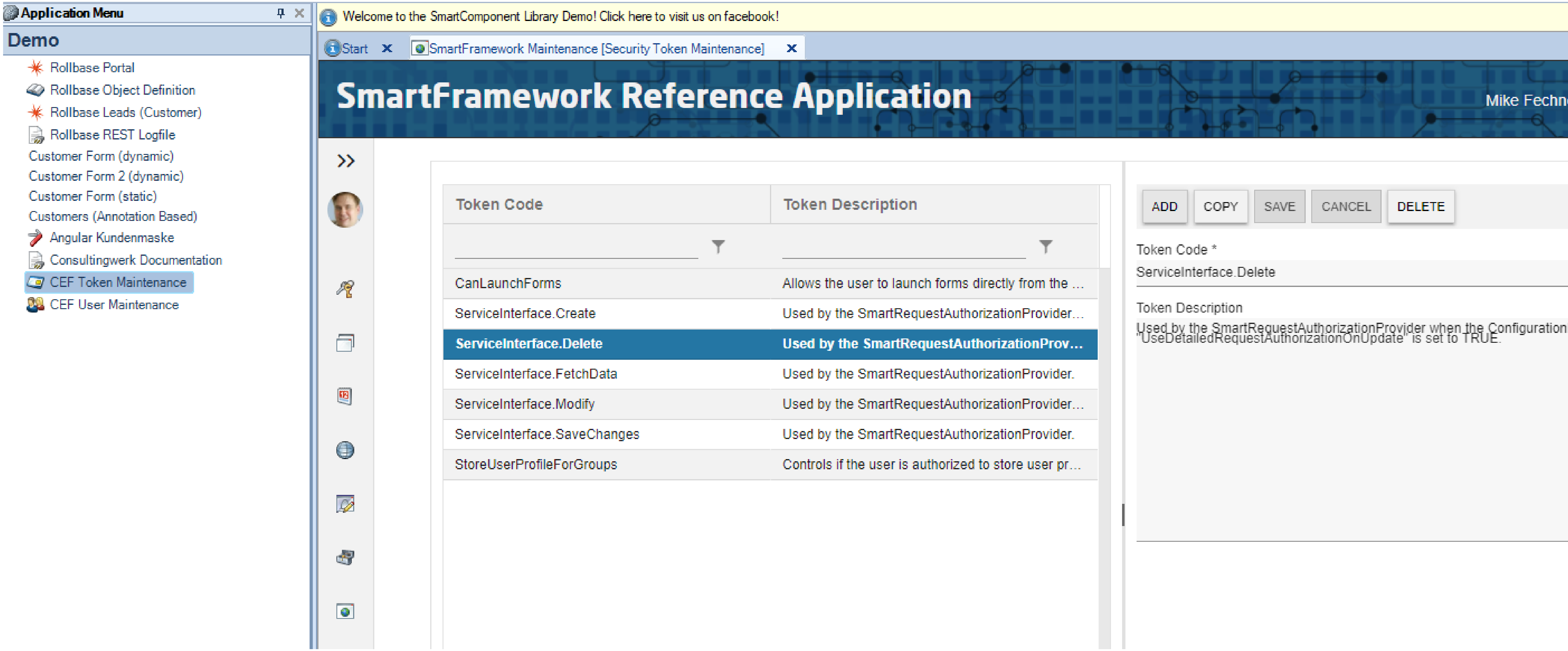
- # character in URL typically used for anchors in (long) web pages, e.g.: https://en.wikipedia.org/wiki/OpenEdge#**History**
- Supports navigation within web page via URL – **without reload**
- Router components of SPA frameworks use this behavior to address application screens (routes) via anchor
- Anchor not part of the HTML page

Hash based routing

- Navigation from

  http://localhost:3000/#/(view:user-maintenance)
  to
  http://localhost:3000/#/(view:security-token-maintenance)

- Navigation to screen/route without full page reload of SPA support prompt changing between screens, screen context may remain available
- Supports integration of Angular web screens into the menu of the desktop application


Application Menu
Demo
Rollbase Portal
Rollbase Object Definition
Rollbase Leads (Customer)
Rollbase REST Logfile
Customer Form (dynamic)
Customer Form 2 (dynamic)
Customer Form (static)
Customers (Annotation Based)
Angular Kundenmaske
Consultingwerk Documentation
CEF Token Maintenance
CEF User Maintenance
Start
SmartFramework Maintenance [Security Token Maintenance]
SmartFramework Reference Application
Mike Fechn
ADD
COPY
SAVE
CANCEL
DELETE

Welcome to the SmartComponent Library Demo! Click here to visit us on facebook!

Token Code	Token Description
CanLaunchForms	Allows the user to launch forms directly from the ...
ServiceInterface.Create	Used by the SmartRequestAuthorizationProvider...
ServiceInterface.Delete	Used by the SmartRequestAuthorizationProv...
ServiceInterface.FetchData	Used by the SmartRequestAuthorizationProvider.
ServiceInterface.Modify	Used by the SmartRequestAuthorizationProvider...
ServiceInterface.SaveChanges	Used by the SmartRequestAuthorizationProvider.
StoreUserProfileForGroups	Controls if the user is authorized to store user pr...

Token Code *
ServiceInterface.Delete

Token Description
Used by the SmartRequestAuthorizationProvider when the Configuration
"UseDetailedRequestAuthorizationOnUpdate" is set to TRUE.

Demo

- Demo – Navigation between Angular Forms from the desktop menu

Agenda

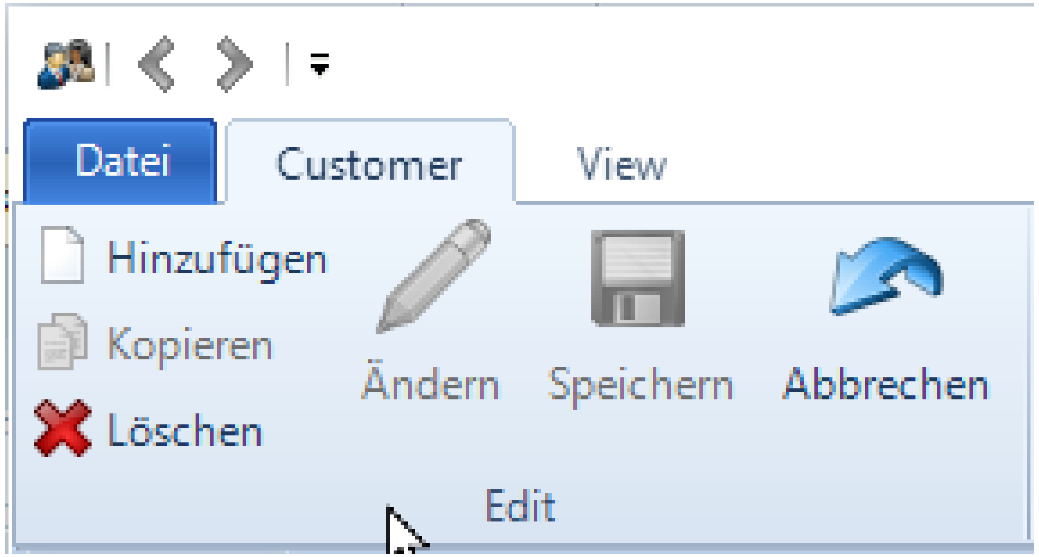
- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- Usage of the Web Browser Control in your application
- Hybrid application architecture
- Navigating within a SPA (Angular)
- **Interaction with the SPA (Angular)**
- Authentication
- Integrating local web content as pretty controls

Interaction with the SPA (Angular)

- CefSharp supports bidirectional communication between the Desktop application and the SPA
  - Execute JavaScript Function from the Desktop application
  - Raise an event from the browser application and handle the event in the Desktop application
- Interaction directly through the web browser control, no need to use any external messaging like web-sockets or JMS

Interaction with the SPA (Angular)

- Use case: Replace buttons in the web application with buttons in the Ribbon of the Desktop application
- Button click event and enabling/disabling of the Buttons depending on the screen status

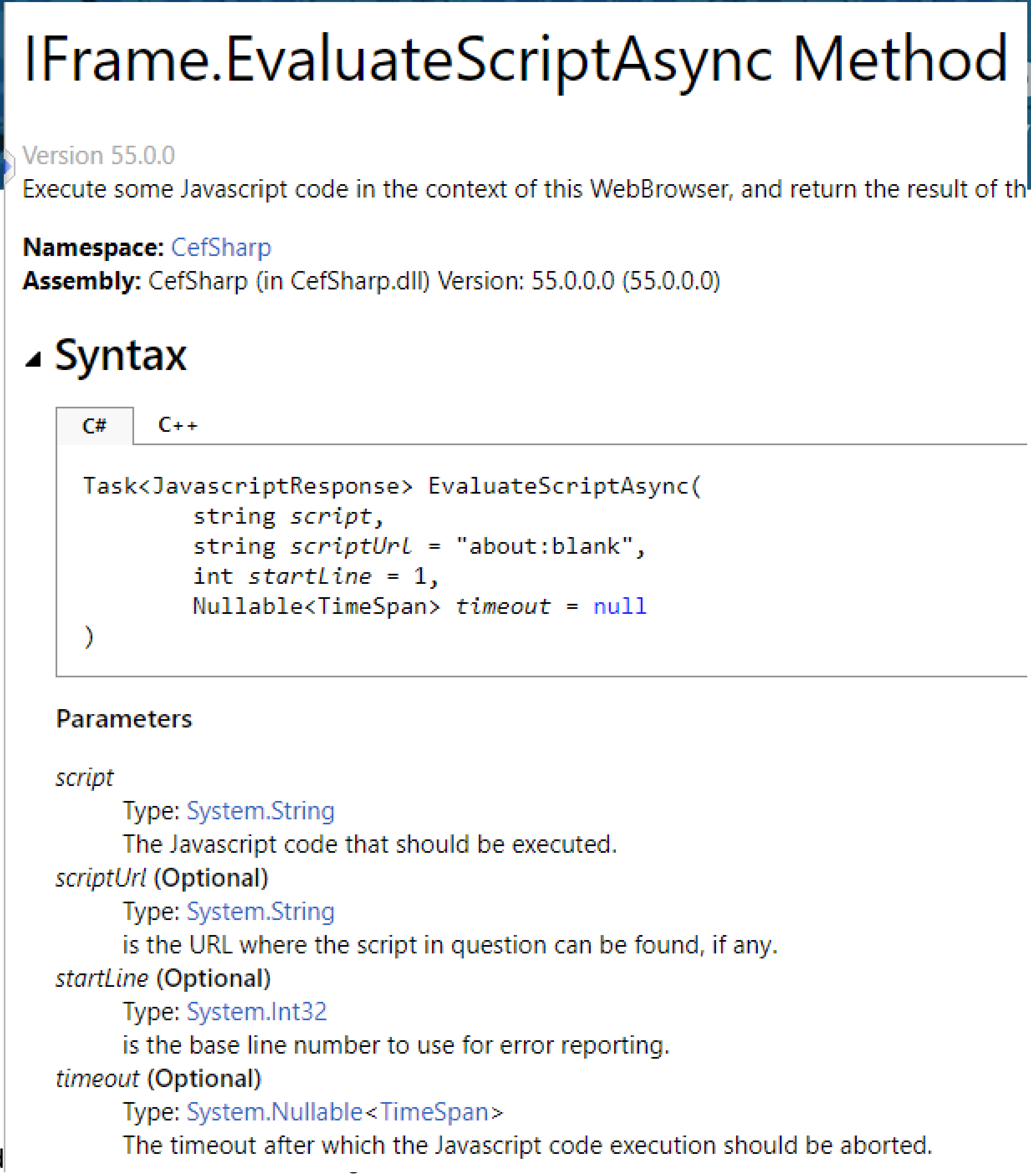


Execution of JavaScript function via CefSharp

- Based on EvaluateScriptAsync function of the Frame object of CefSharp (Web pages may be framesets)
- e.g.

```
EvaluateScriptAsync('cefsharpCall("ButtonClick",
                                  {
                                     "toolbarInstanceName": “DefaultToolbar",
                                     "buttonName": "SaveChanges"
                                   });') .
```

- ***cefsharpCall*** is a global JavaScript function which calls into the ***ButtonClick*** handler of the referenced Buttons
- http://cefsharp.github.io/api/55.0.0/html/M_CefSharp_IFrame_EvaluateScriptAsync.htm



IFrame.EvaluateScriptAsync Method

Version 55.0.0

Execute some Javascript code in the context of this WebBrowser, and return the result of th

Namespace: CefSharp
Assembly: CefSharp (in CefSharp.dll) Version: 55.0.0.0 (55.0.0.0)

Syntax

```
Task<JavascriptResponse> EvaluateScriptAsync(
        string script,
        string scriptUrl = "about:blank",
        int startLine = 1,
        Nullable<TimeSpan> timeout = null
)
```

Parameters

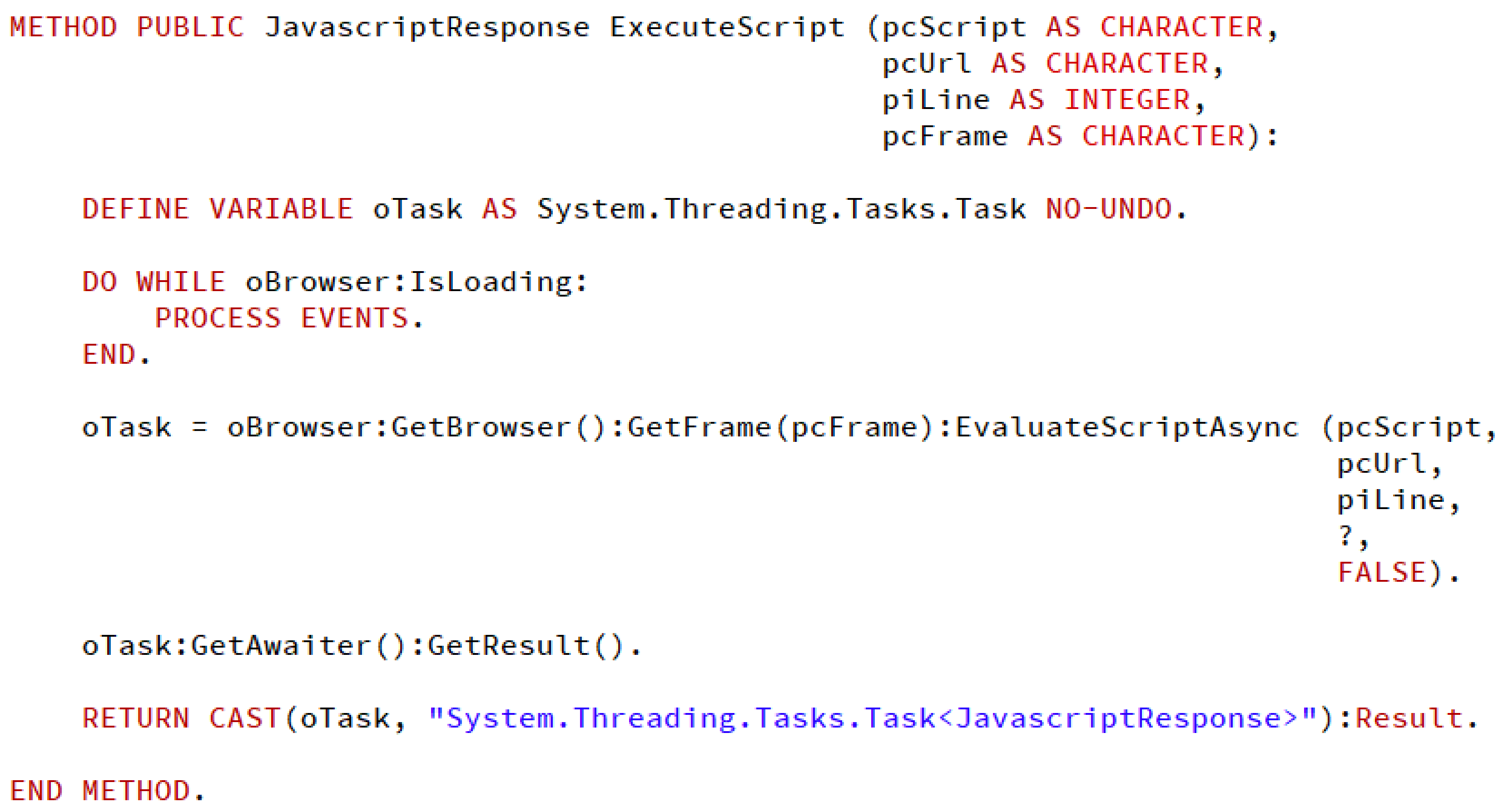
script
    Type: System.String
    The Javascript code that should be executed.

scriptUrl **(Optional)**
    Type: System.String
    is the URL where the script in question can be found, if any.

startLine **(Optional)**
    Type: System.Int32
    is the base line number to use for error reporting.

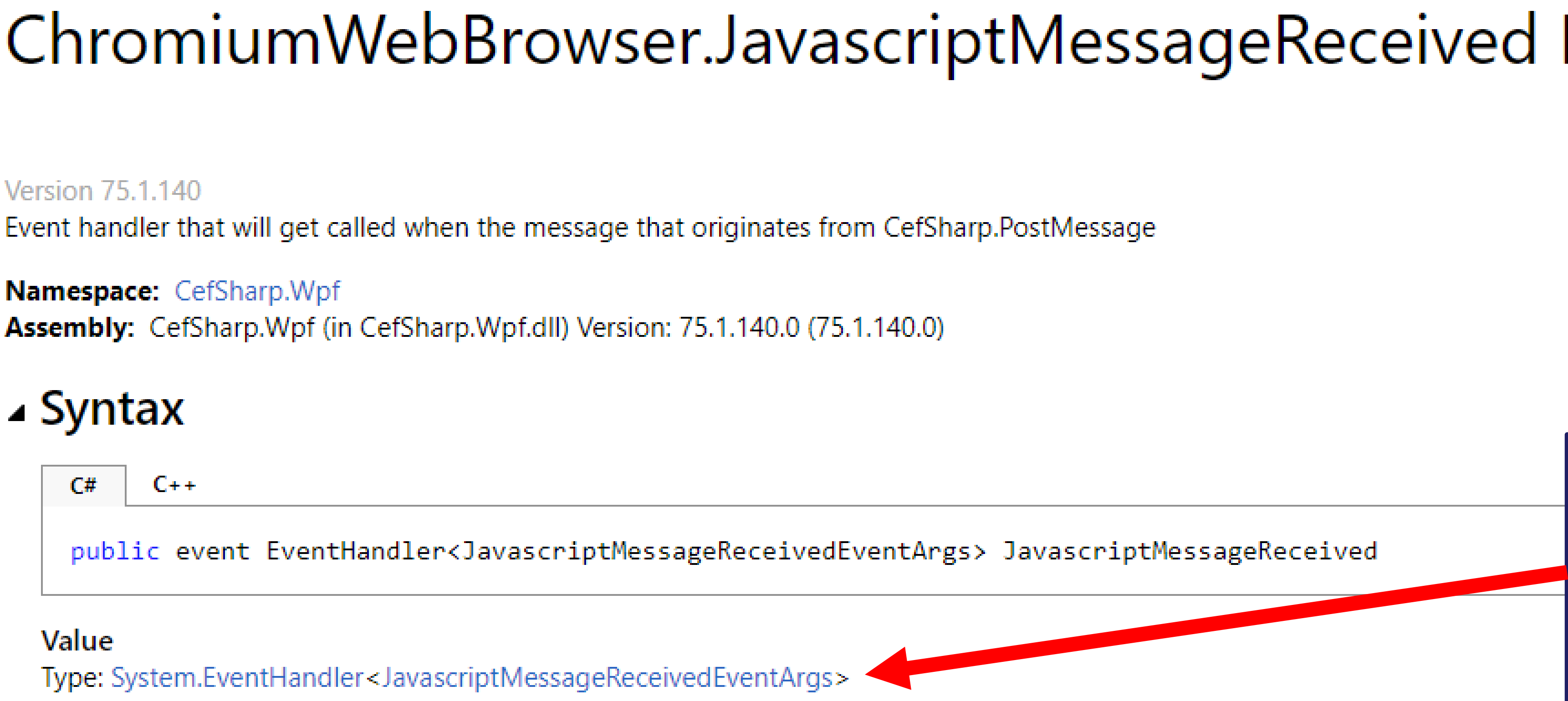
timeout **(Optional)**
    Type: System.Nullable<TimeSpan>
    The timeout after which the Javascript code execution should be aborted.

Invoking async .NET method from the ABL …



```
METHOD PUBLIC JavascriptResponse ExecuteScript (pcScript AS CHARACTER,
                                                pcUrl AS CHARACTER,
                                                piLine AS INTEGER,
                                                pcFrame AS CHARACTER):

    DEFINE VARIABLE oTask AS System.Threading.Tasks.Task NO-UNDO.

    DO WHILE oBrowser:IsLoading:
        PROCESS EVENTS.
    END.

    oTask = oBrowser:GetBrowser():GetFrame(pcFrame):EvaluateScriptAsync (pcScript,
                                                                         pcUrl,
                                                                         piLine,
                                                                         ?,
                                                                         FALSE).

    oTask:GetAwaiter():GetResult().

    RETURN CAST(oTask, "System.Threading.Tasks.Task<JavascriptResponse>"):Result.

END METHOD.
```

Handling of JavaScript Event through CefSharp



ChromiumWebBrowser.JavascriptMessageReceived Event

Version 75.1.140

Event handler that will get called when the message that originates from CefSharp.PostMessage

Namespace: CefSharp.Wpf
Assembly: CefSharp.Wpf (in CefSharp.Wpf.dll) Version: 75.1.140.0 (75.1.140.0)

Syntax

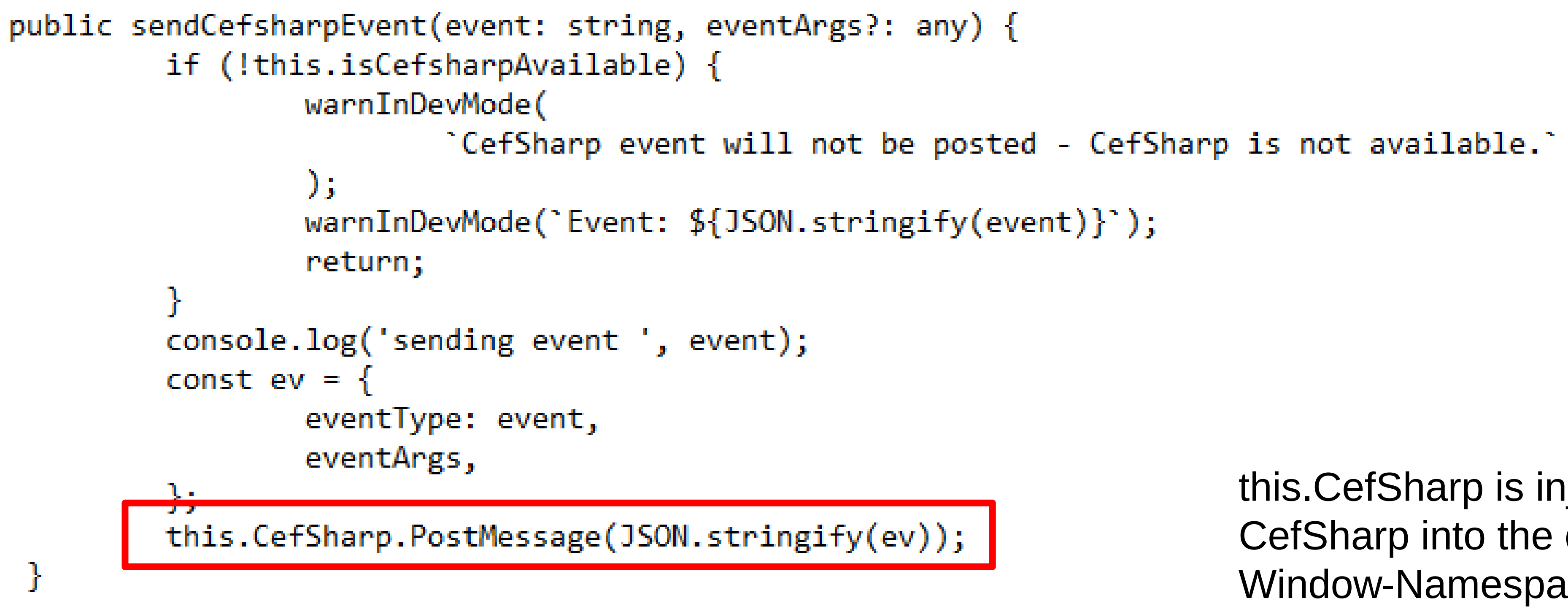
```
public event EventHandler<JavascriptMessageReceivedEventArgs> JavascriptMessageReceived
```

Value
Type: System.EventHandler<JavascriptMessageReceivedEventArgs>

JavaScriptMessageReceived
EventArgs Class
returns„Value“ as
System.Object, e.g. as
System.String
(ABL CHARACTER)

- http://cefsharp.github.io/api/75.1.x/html/E_CefSharp_Wpf_ChromiumWebBrowser_JavascriptMessageReceived.htm

Raise JavaScript Event through PostMessage



```
public sendCefsharpEvent(event: string, eventArgs?: any) {
        if (!this.isCefsharpAvailable) {
                warnInDevMode(
                        `CefSharp event will not be posted - CefSharp is not available.`
                );
                warnInDevMode(`Event: ${JSON.stringify(event)}`);
                return;
        }
        console.log('sending event ', event);
        const ev = {
                eventType: event,
                eventArgs,
        };
        this.CefSharp.PostMessage(JSON.stringify(ev));
 }
```

this.CefSharp is injected by CefSharp into the den JavaScript Window-Namespace

Agenda

- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- Usage of the Web Browser Control in your application
- Hybrid application architecture
- Navigating within a SPA (Angular)
- Interaction with the SPA (Angular)
- **Authentication**
- Integrating local web content as pretty controls

Authentication

- Technically the desktop application and the embedded web application are two separate applications
- From a user's point of view both applications should be seen as a single application
- Multiple authentication - first on the desktop application and then in the embedded web application – is disturbing
- Worst case, multiple authentication in the browser application, when CefSharp window would be closed

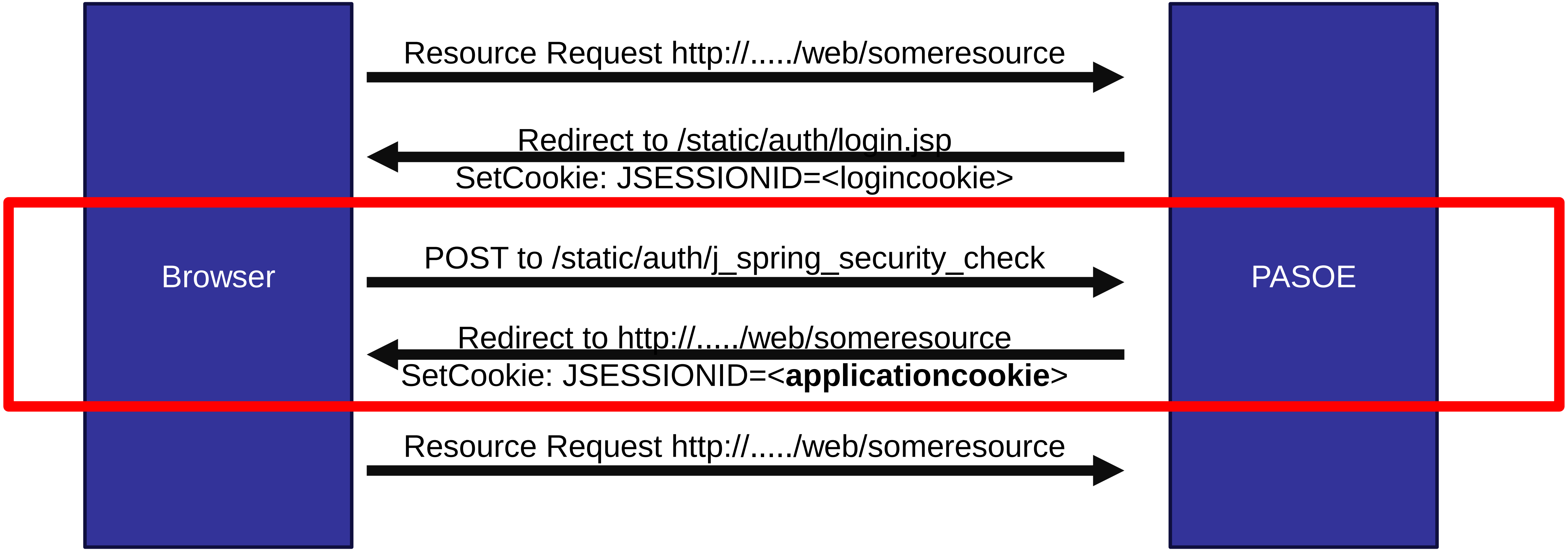
“Single sign-on” options

- Web application needs to bypass its own login prompt
- Desktop application needs to inject authentication into requests made from CefSharp control
  - Cookies (e.g. JSESSIONID cookie)
  - Headers (via ResourceRequestHandler implementation)
    - Authentication header (e.g. bearer, OAuth2, SAML)

Sample –PASOE‘s form-based authentication

- Browses challenges authentication by accessing **resource**
- PASOE redirects to login form, provides JSESSIONID for login session
- Browser posts username/domain and password to login endpoint
- PASOE redirects on successful login to **resource**, provides JSESSIONID for application session, login session voided
- Browsers accesses **resource** using JSESSIONID of application session

Sample –PASOE‘s form-based authentication



Resource Request http://...../web/someresource

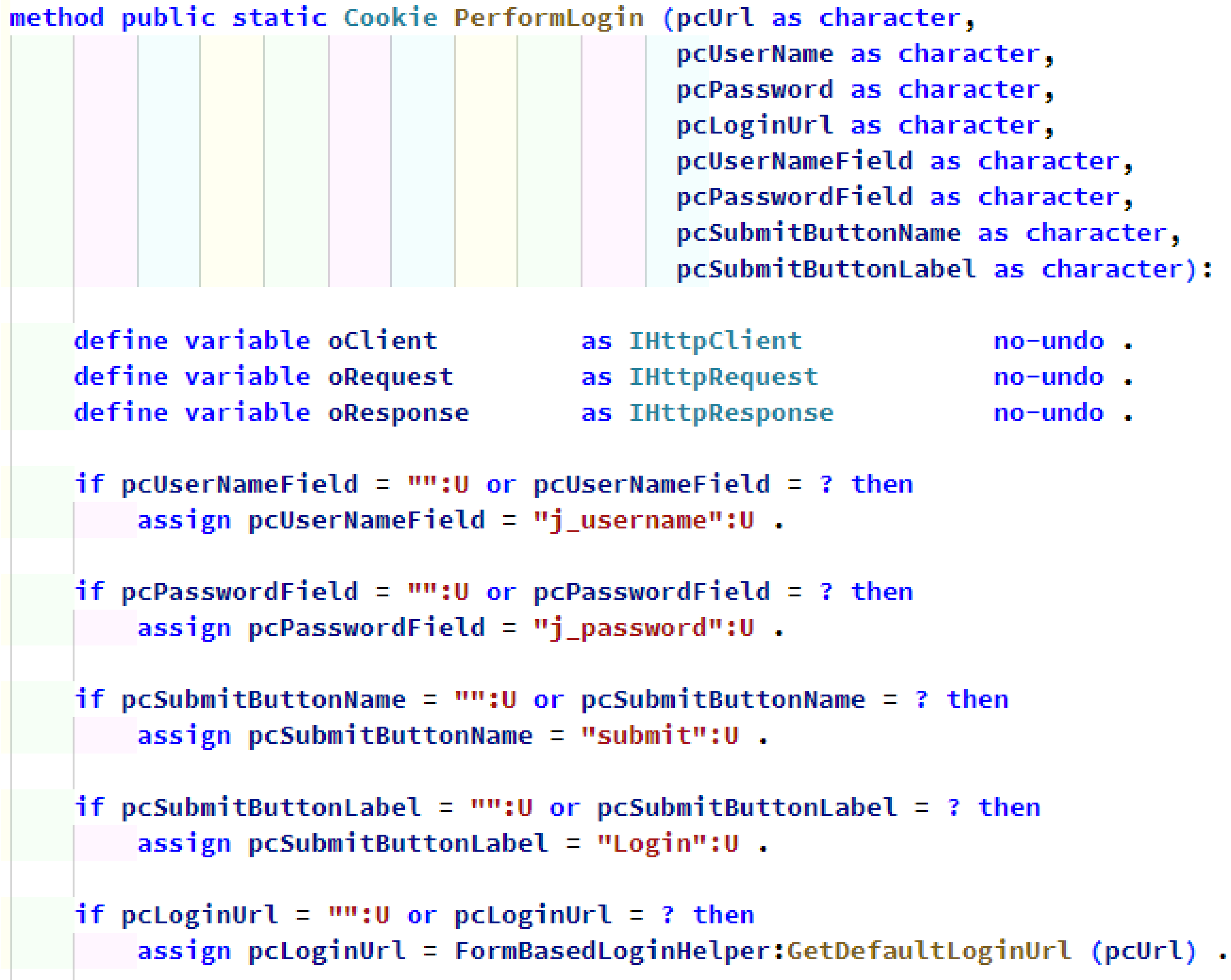
Redirect to /static/auth/login.jsp
SetCookie: JSESSIONID=<logincookie>

POST to /static/auth/j_spring_security_check

Redirect to http://...../web/someresource
SetCookie: JSESSIONID=<**applicationcookie**>

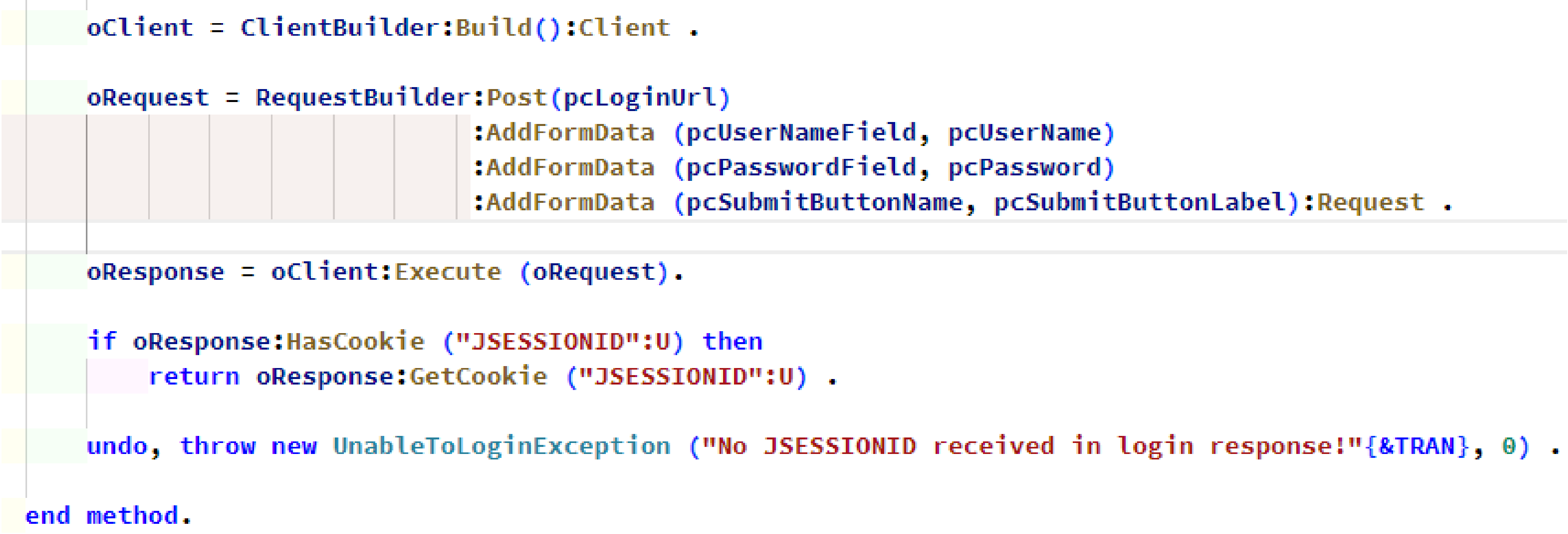
Resource Request http://...../web/someresource

Use ABL HTTP Client to perform form-based auth



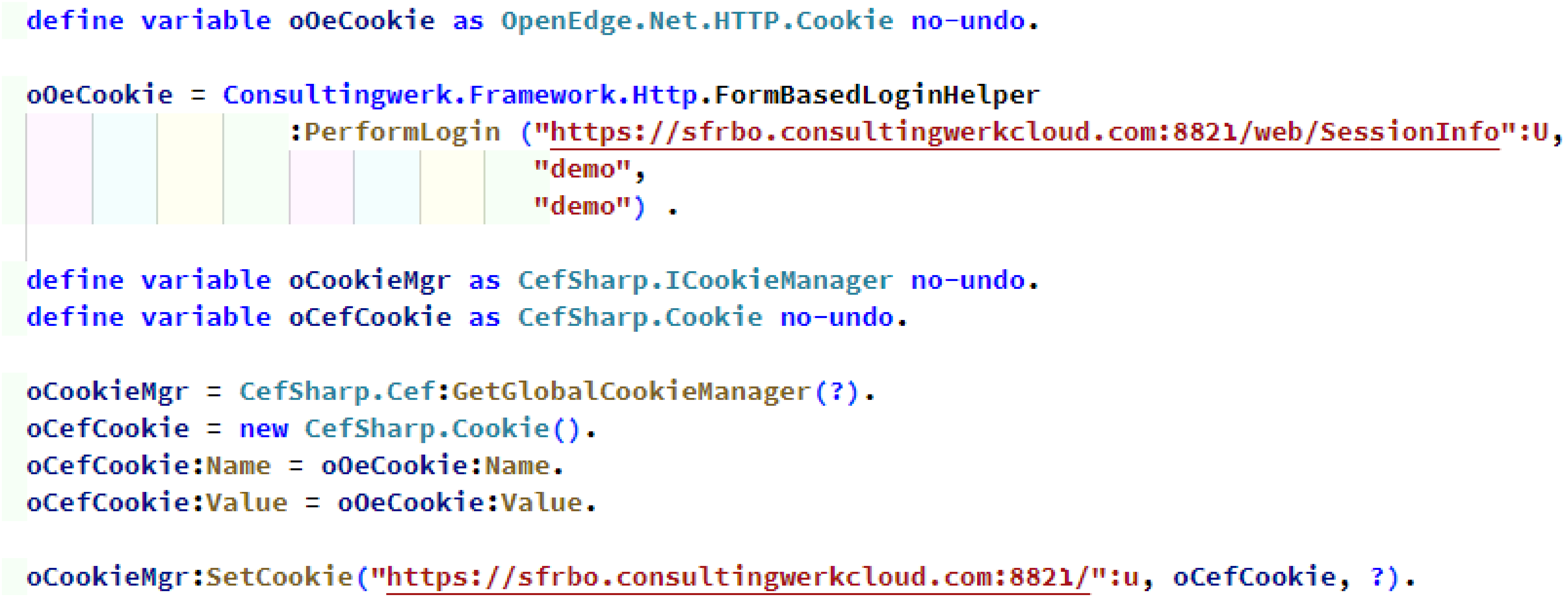
```
method public static Cookie PerformLogin (pcUrl as character,
                                          pcUserName as character,
                                          pcPassword as character,
                                          pcLoginUrl as character,
                                          pcUserNameField as character,
                                          pcPasswordField as character,
                                          pcSubmitButtonName as character,
                                          pcSubmitButtonLabel as character):

    define variable oClient         as IHttpClient          no-undo .
    define variable oRequest        as IHttpRequest         no-undo .
    define variable oResponse       as IHttpResponse        no-undo .

    if pcUserNameField = "":U or pcUserNameField = ? then
        assign pcUserNameField = "j_username":U .

    if pcPasswordField = "":U or pcPasswordField = ? then
        assign pcPasswordField = "j_password":U .

    if pcSubmitButtonName = "":U or pcSubmitButtonName = ? then
        assign pcSubmitButtonName = "submit":U .

    if pcSubmitButtonLabel = "":U or pcSubmitButtonLabel = ? then
        assign pcSubmitButtonLabel = "Login":U .

    if pcLoginUrl = "":U or pcLoginUrl = ? then
        assign pcLoginUrl = FormBasedLoginHelper:GetDefaultLoginUrl (pcUrl) .
```

Use ABL HTTP Client to perform form-based auth



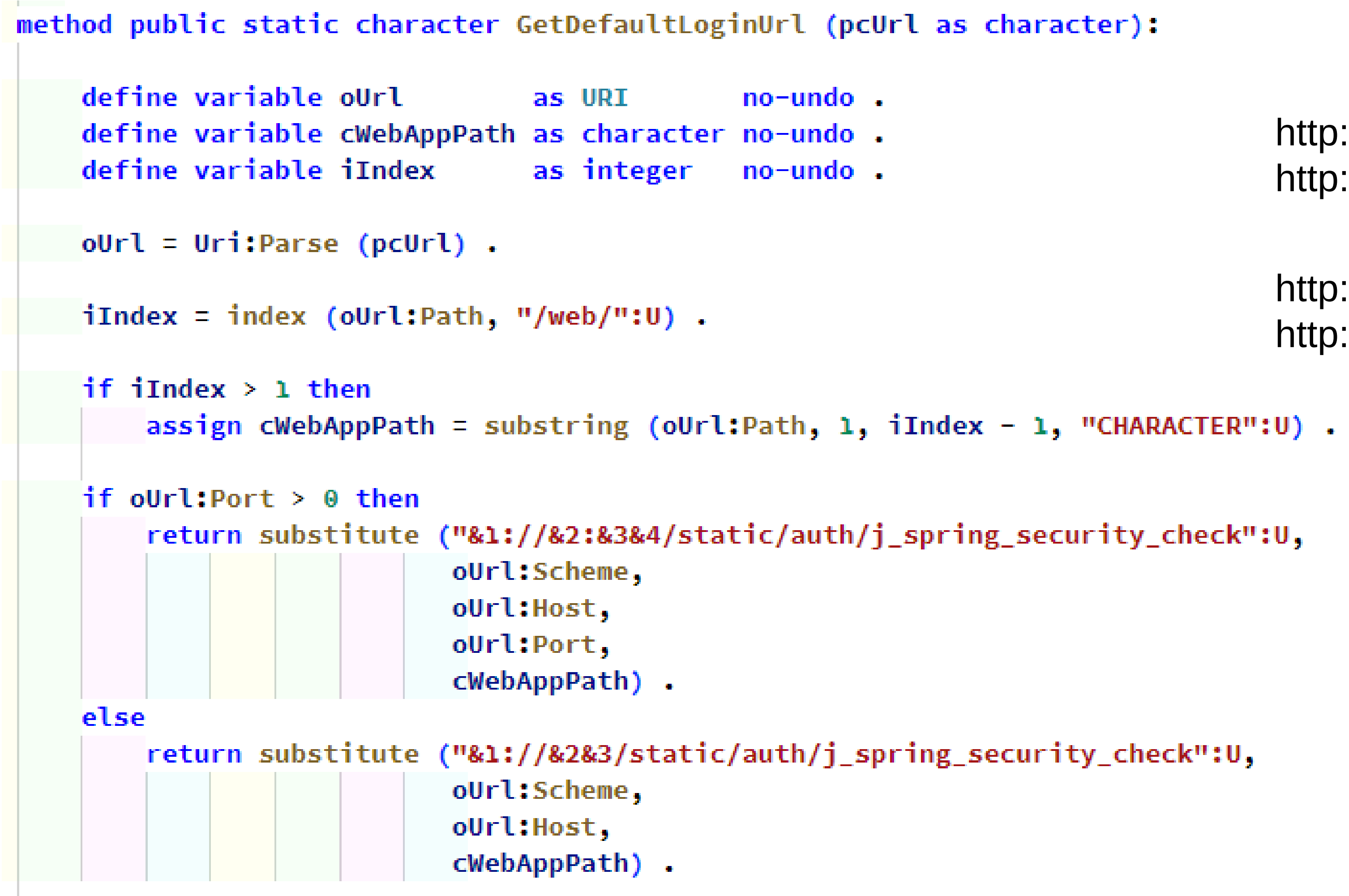
```
        oClient = ClientBuilder:Build():Client .

        oRequest = RequestBuilder:Post(pcLoginUrl)
                                 :AddFormData (pcUserNameField, pcUserName)
                                 :AddFormData (pcPasswordField, pcPassword)
                                 :AddFormData (pcSubmitButtonName, pcSubmitButtonLabel):Request .

        oResponse = oClient:Execute (oRequest).

        if oResponse:HasCookie ("JSESSIONID":U) then
            return oResponse:GetCookie ("JSESSIONID":U) .

        undo, throw new UnableToLoginException ("No JSESSIONID received in login response!"{&TRAN}, 0) .

    end method.
```

Provide Cookie to CefSharp



```
define variable oOeCookie as OpenEdge.Net.HTTP.Cookie no-undo.

oOeCookie = Consultingwerk.Framework.Http.FormBasedLoginHelper
                :PerformLogin ("https://sfrbo.consultingwerkcloud.com:8821/web/SessionInfo":U,
                               "demo",
                               "demo") .

define variable oCookieMgr as CefSharp.ICookieManager no-undo.
define variable oCefCookie as CefSharp.Cookie no-undo.

oCookieMgr = CefSharp.Cef:GetGlobalCookieManager(?).
oCefCookie = new CefSharp.Cookie().
oCefCookie:Name = oOeCookie:Name.
oCefCookie:Value = oOeCookie:Value.

oCookieMgr:SetCookie("https://sfrbo.consultingwerkcloud.com:8821/":u, oCefCookie, ?).
```

Use ABL HTTP Client to perform form-based auth



```
method public static character GetDefaultLoginUrl (pcUrl as character):

    define variable oUrl        as URI       no-undo .
    define variable cWebAppPath as character no-undo .
    define variable iIndex      as integer   no-undo .

    oUrl = Uri:Parse (pcUrl) .

    iIndex = index (oUrl:Path, "/web/":U) .

    if iIndex > 1 then
        assign cWebAppPath = substring (oUrl:Path, 1, iIndex - 1, "CHARACTER":U) .

    if oUrl:Port > 0 then
        return substitute ("&1://&2:&3&4/static/auth/j_spring_security_check":U,
                           oUrl:Scheme,
                           oUrl:Host,
                           oUrl:Port,
                           cWebAppPath) .
    else
        return substitute ("&1://&2&3/static/auth/j_spring_security_check":U,
                           oUrl:Scheme,
                           oUrl:Host,
                           cWebAppPath) .
```

http://.../web/resource ->
http://.../static/j_spring_security_check

http://.../webapp/web/resource ->
http://.../webapp/static/j_spring_security_check

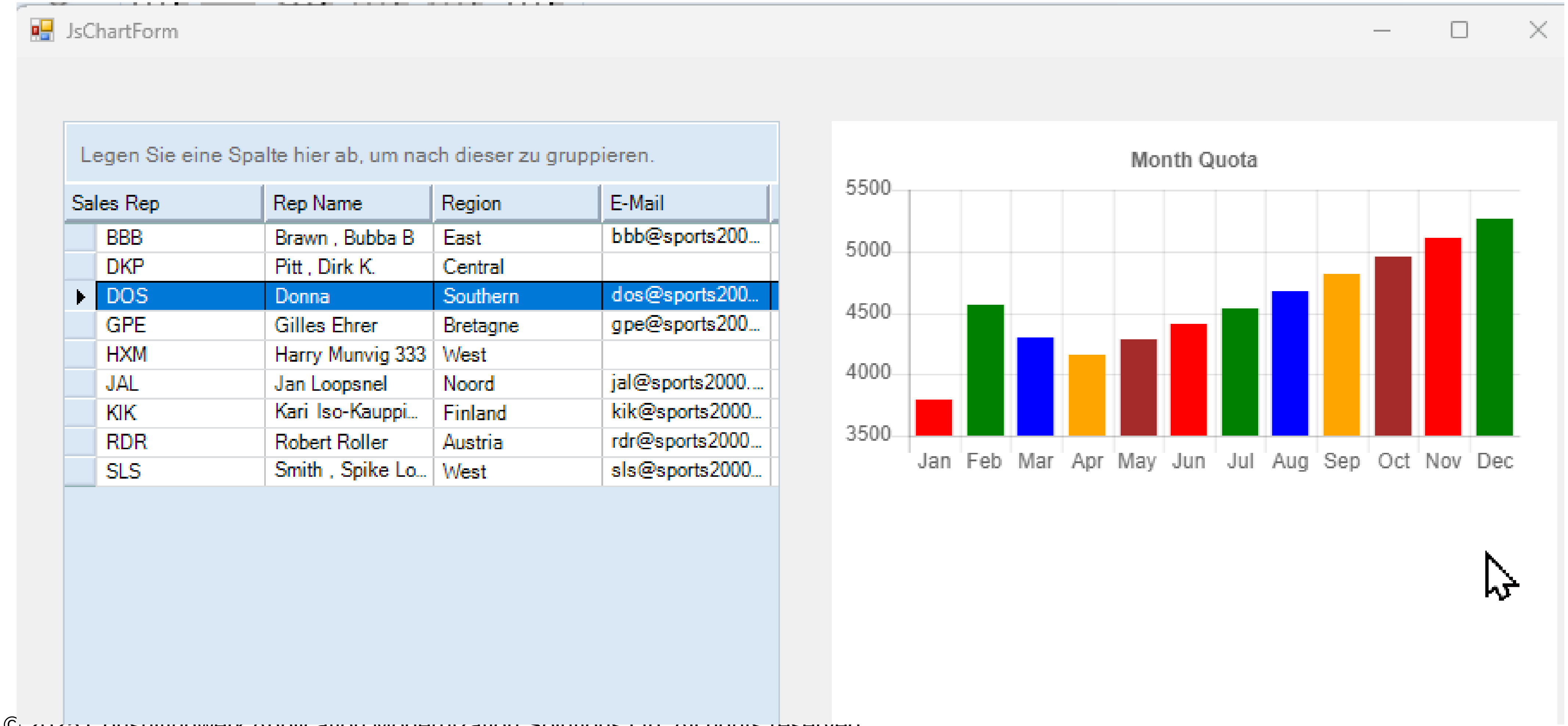
Agenda

- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- Usage of the Web Browser Control in your application
- Hybrid application architecture
- Navigating within a SPA (Angular)
- Interaction with the SPA (Angular)
- Authentication
- **Integrating local web content as pretty controls**

Demo: Using CefSharp to integrate Chart Controls

- Free JavaScript Chart Component
- https://www.w3schools.com/ai/ai_chartjs.asp
- CefSharp integrated as Control in „regular“ .NET Form
- Multiple CefSharp Controls can be integrated in a single page

Demo: Using CefSharp to integrate Chart Controls



Legen Sie eine Spalte hier ab, um nach dieser zu gruppieren.

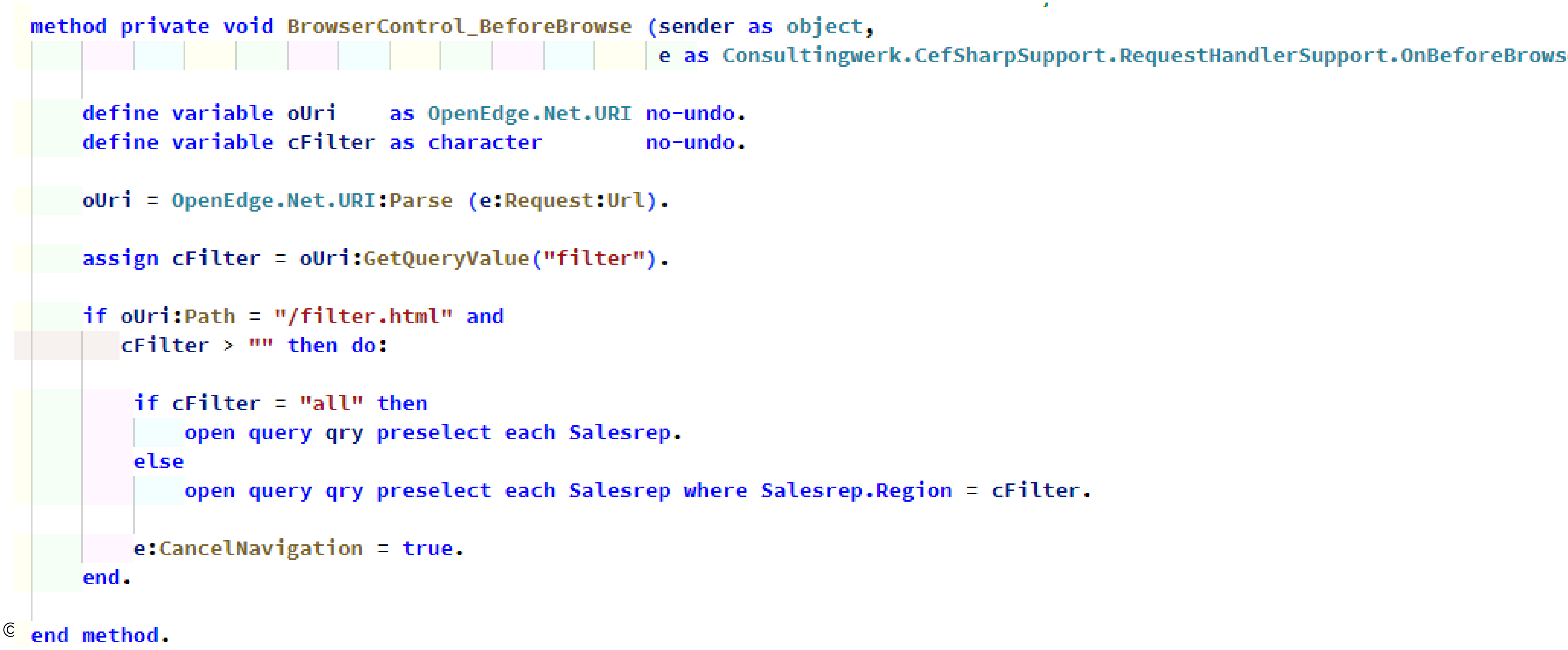
Sales Rep	Rep Name	Region	E-Mail
BBB	Brawn , Bubba B	East	bbb@sports200...
DKP	Pitt , Dirk K.	Central	
DOS	Donna	Southern	dos@sports200...
GPE	Gilles Ehrer	Bretagne	gpe@sports200...
HXM	Harry Munvig 333	West	
JAL	Jan Loopsnel	Noord	jal@sports2000....
KIK	Kari Iso-Kauppi...	Finland	kik@sports2000...
RDR	Robert Roller	Austria	rdr@sports2000...
SLS	Smith , Spike Lo...	West	sls@sports2000...

Populating Chart

- Complete HTML source of Chart generated by ABL code on the fly
- No file on disk, no file on a web server

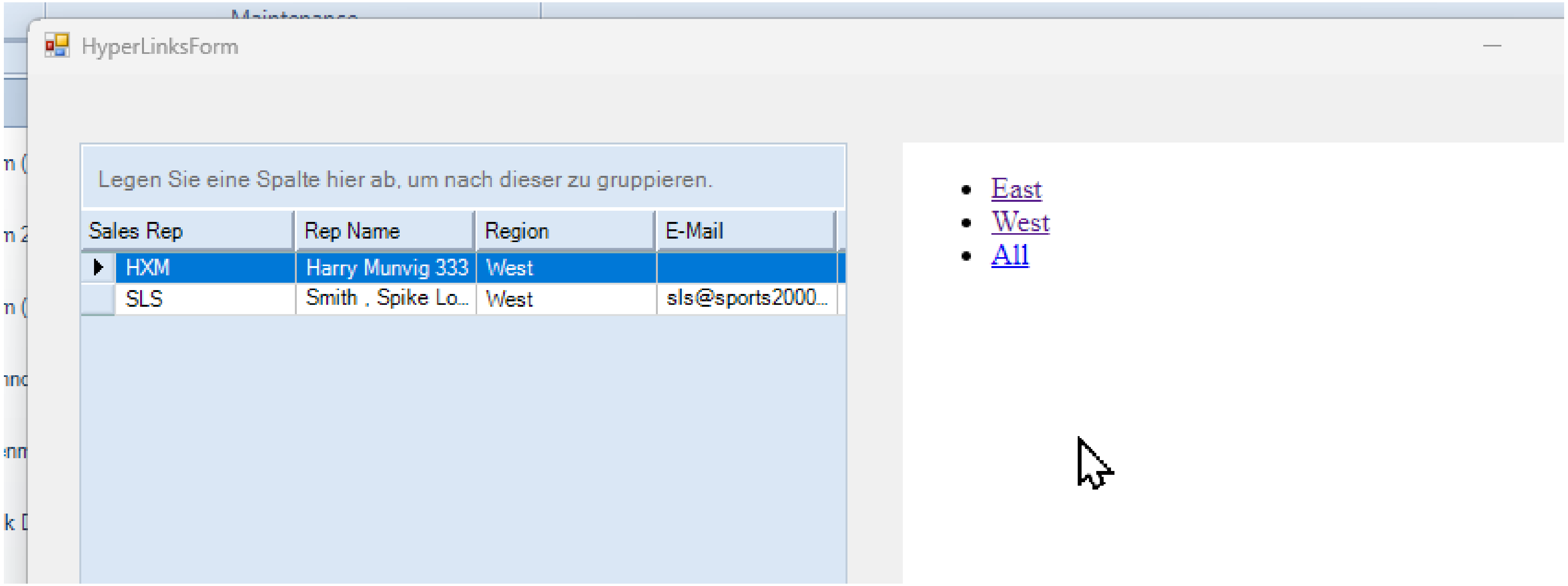
```
WebBrowserExtensions:LoadHtml
     (browserControl1:ChromiumWebBrowser,
      cHtml,
      "http://embeddedbrowser/":u).
```

Demo: Use HTML components as „pretty“ menu option



```
method private void BrowserControl_BeforeBrowse (sender as object,
                                                 e as Consultingwerk.CefSharpSupport.RequestHandlerSupport.OnBeforeBrows

    define variable oUri    as OpenEdge.Net.URI no-undo.
    define variable cFilter as character        no-undo.

    oUri = OpenEdge.Net.URI:Parse (e:Request:Url).

    assign cFilter = oUri:GetQueryValue("filter").

    if oUri:Path = "/filter.html" and
       cFilter > "" then do:

        if cFilter = "all" then
            open query qry preselect each Salesrep.
        else
            open query qry preselect each Salesrep where Salesrep.Region = cFilter.

        e:CancelNavigation = true.
    end.

end method.
```

Demo: Use HTML components as „pretty“ menu option



HyperLinksForm

Legen Sie eine Spalte hier ab, um nach dieser zu gruppieren.

Sales Rep	Rep Name	Region	E-Mail
HXM	Harry Munvig 333	West	
SLS	Smith , Spike Lo...	West	sls@sports2000...

- East
- West
- All

Questions

PRODUCTIVE
PROFESSIONAL
PROBLEMSOLVING
SMART
EDUCATION
LEADING
TECHSAVVY
RELIABLE
INNOVATIVE
GEEKY
WORLDWIDE
PROACTIVE
REMARKABLE
KNOWLEDGABLE
CONSULTINGWERK