

# SITHS eID Appväxling - Exempel för inbäddade webbläsare

Version	Datum	Författare	Kommentar
0.9	13 Oct 2020	<a href="#">Agerberg, Tobias</a>	Skapat exempelkod
1.0	24 Nov 2020	<a href="#">Christoffer Johansson</a>	Upprättat dokument
1.1	12 Jan 2021	<a href="#">Skarsgård, Grim</a>	Versionering

Under tidiga tester med Region Stockholm identifierades att deras Journalsystem, Take Care, jobbar med Chrome som en inbäddad webbläsare när man presenterar e-tjänsten (Webcert) och IdP:n (Ineras IdP).

Om en användare väljer metoden **SITHS eID på denna enhet** i IdP:n måste den tjänst/applikation som presenterar IdP:n tillåta att det externa SITHS-protokollet (siths://) får exekvera och därigenom begära start av SITHS eID app.

För standardwebbläsarna på respektive plattform kommer installationen av SITHS eID app säkerställa att detta fungerar. Men när en tjänst bäddar in en webbläsare brukar dessa funktioner avaktiveras.

## Exempel för lösning på Windows 10 med inbäddad Chrome

Följande ger en proof-of-concept på hur man kan tillåta SITHS-protokollet att exekvera/appväxla vid användning av CefSharp.

Beroende på vilket språk er applikation och ramverk som just er applikation använder så kan nedan kod behöva anpassas.

1. Först skapas en anpassad ResourceRequestHandler som tillåter protokollet "siths" men inget annat protokoll. Denna funktion anropas för alla protokoll som är okända, innan de tillåts.

```
class CustomResourceRequestHandler : ResourceRequestHandler
{
    protected override bool OnProtocolExecution(IWebBrowser chromiumWebBrowser, IBrowser browser, IFrame frame, IRequest request)
    {
        Uri uri = new Uri(request.Url);
        if (uri.Scheme == "siths")
        {
            return true;
        }
        return false;
    }
}
```

2. Sedan skapas en factory-klass som returnerar vår implementation.

```
public class CustomResourceRequestHandlerFactory : IResourceRequestHandlerFactory
{
    public bool HasHandlers => true; public IResourceRequestHandler GetResourceRequestHandler(IWebBrowser chromiumWebBrowser, IBrowser browser, IFrame frame, IRequest request, bool isNavigation, bool isDownload, string requestInitiator, ref bool disableDefaultHandling)
    {
        return new CustomResourceRequestHandler();
    }
}
```

3. Till sist pekar vi ut vår factory-klass för browser-instansen.

```
var browser = new ChromiumWebBrowser("https://enrollment.preacctest.ineratest.org")
{
    ResourceRequestHandlerFactory = new CustomResourceRequestHandlerFactory()
};
panel.Controls.Add(browser);
```

4. Därefter bör det fungera för er applikation att "appväxla" och starta SITHS eID Windows via en inbäddad Chrome.