SAML u .NET aplikacijama

U ovom primjeru koristi se OIOSAML.NET programski alat za implementaciju SAML2 protokola u .NET web aplikacijama. Ovaj programski alat inicijalno je nastao na temelju projekta Danske vlade za integracijom vladinih servisa u sustav jedinstvene autentikacije, a u sustavu AAI@EduHr koristi se kao autentikacijski modul za povezivanje .NET aplikacija s AAI@EduHr Single Sign-On servisom.

Potrebna programska podrška

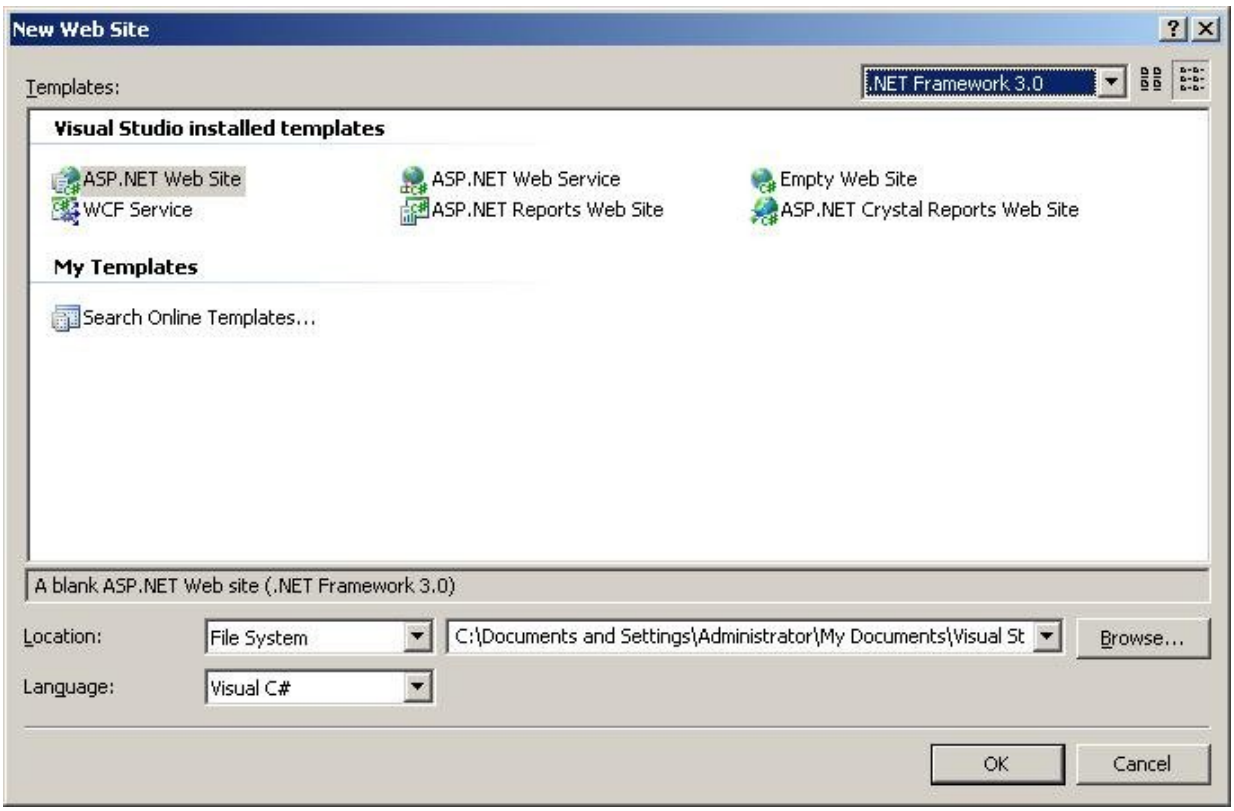
Upute u nastavku realizirane su uporabom sljedeće programske podrške:

- Microsoft Windows Server 2008 R2 SP1 / Windows 7 Professional;
- IIS sa ASP.NET 2.0 modom rada;
- .NET 4.5 framework ili noviji;
- Microsoft Visual Studio 2005 ili noviji;
- Potrebno je instalirati `OIOSAML.NET`, verziju 1.7.9 (direktna poveznica na dk.nita.saml20 nuget paket);
- Odgovarajuća DLL datoteka `dk.nita.saml20.dll` prilagođena sustavu AAI@EduHr;
- Datoteka `log4net.dll` - log4net omogućuje napredno logiranje događaja, koje možemo i ne moramo konfigurirati ovisno o tome želimo ili ne dodatno logiranje, ali ova komponenta mora postojati;
- Windows distribucija OpenSSL-a za generiranje certifikata;

Procedura za implementaciju OIOSAML autentikacijskog modula

1. Pokrenite `Visual Studio` i kreirajte novi `ASP.NET` projekt:

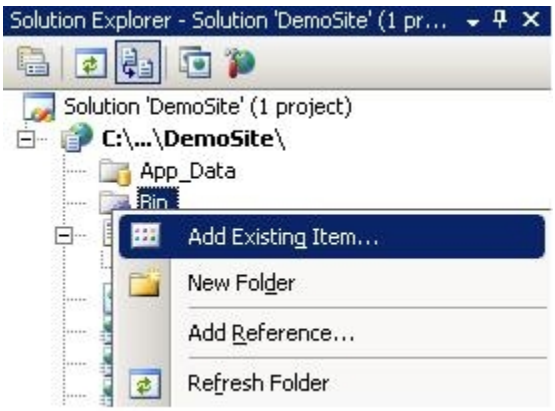
`File -> New -> Web Site...`



2. Instalirajte dk.nita.saml20 nuget paket.

3. Dodajte prilagođenu datoteku **`dk.nita.saml20.dll`** (preuzetu sa ovih uputa) u projekt:

`Projekt -> Add Existing Item...`

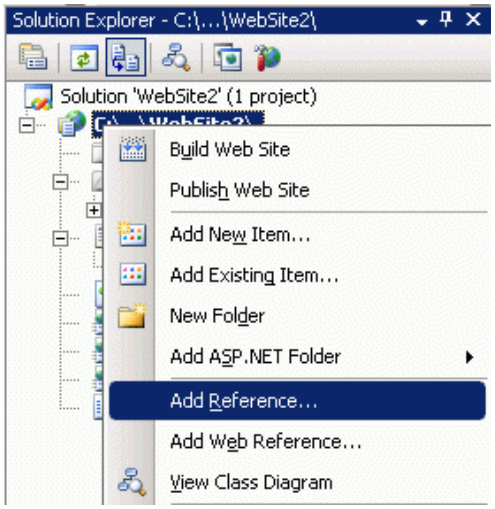


Odaberite datoteku **`dk.nita.saml20.dll.`**

4. Dodajte poveznicu na datoteku **`dk.nita.saml20.dll`:**

- Uklonite referencu na postojeći DLL dk.nita.saml20.dll koji je došao sa instalacijom paketa
- Dodajte referencu na prilagođeni DLL:

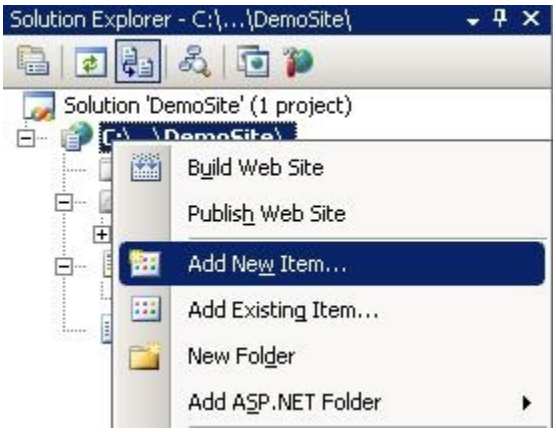
`Projekt -> Add Reference...`



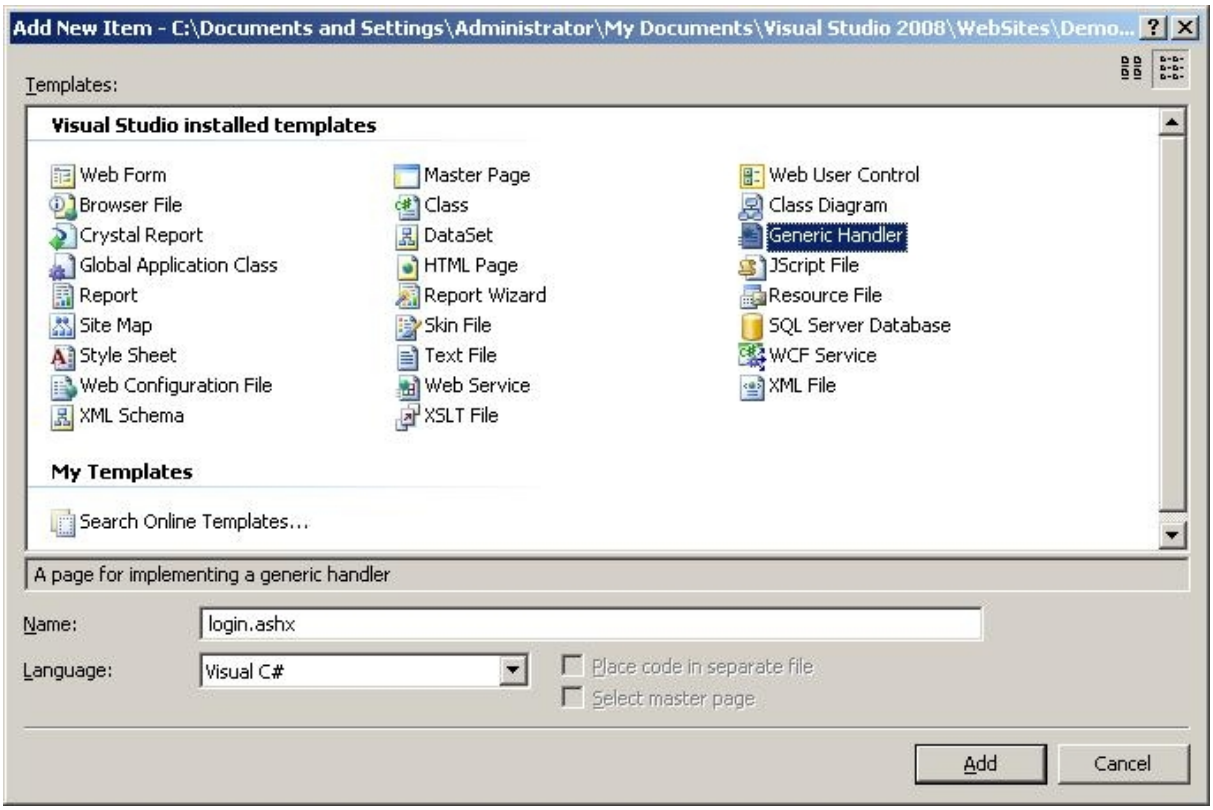
Odaberite datoteku **`dk.nita.saml20.dll.`**

5. Dodajte ASP.NET handlere login.ashx, logout.ashx i metadata.ashx:

`Projekt -> Add New Item...`



Iz ponuđenog prozora odaberite **`Generic Handler`** i upišite ime datoteke **`login.ashx`**



Ponovite postupak za preostala 2 handlera: **`logout.ashx`** i **`metadata.ashx`**.

6. Promjenite sadržaje generičkih handlera na sljedeći način (svaki postojeći kod handlera treba zakomentirati ili obrisati):

U **`login.ashx`** postaviti:

```
<%@ WebHandler Class="dk.nita.saml20.protocol.Saml20SignonHandler" %>
```

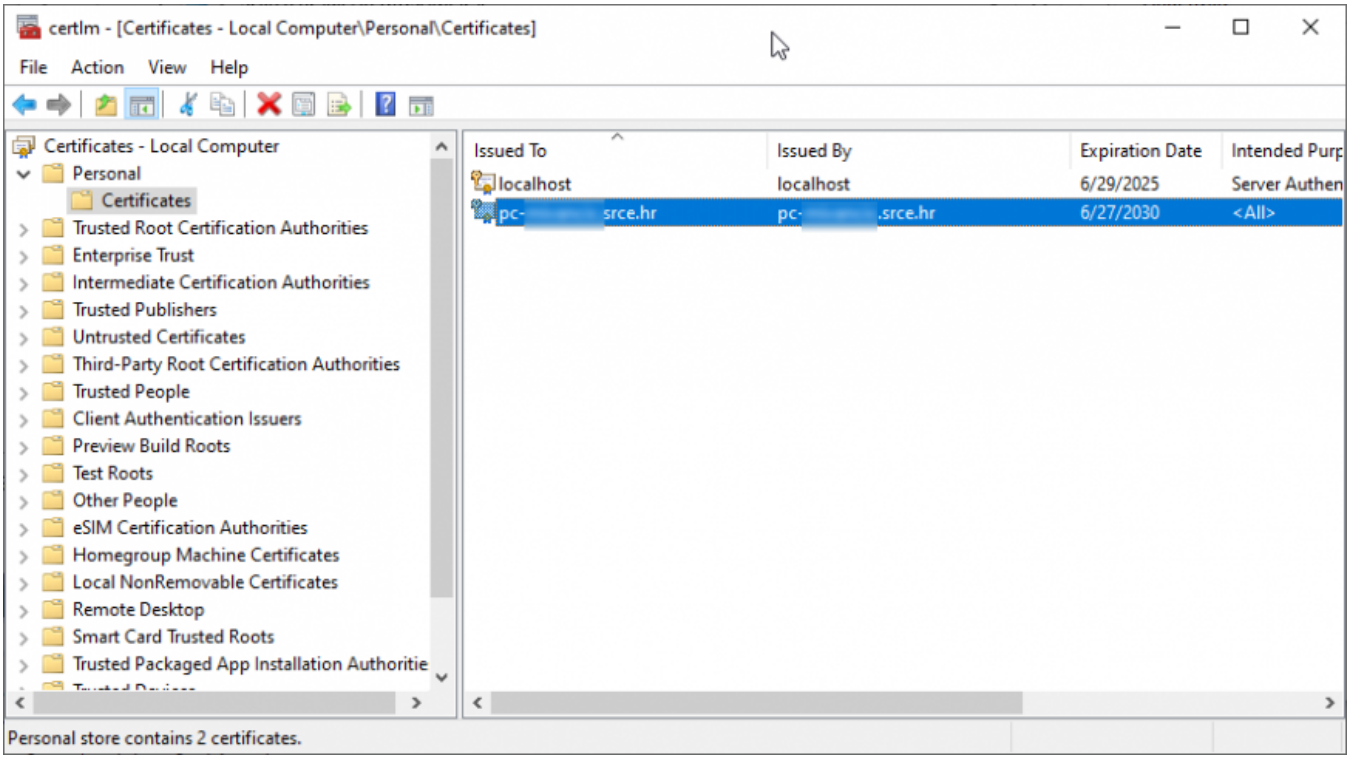
U **`logout.ashx`** postaviti:

```
<%@ WebHandler Class="dk.nita.saml20.protocol.Saml20LogoutHandler" %>
```

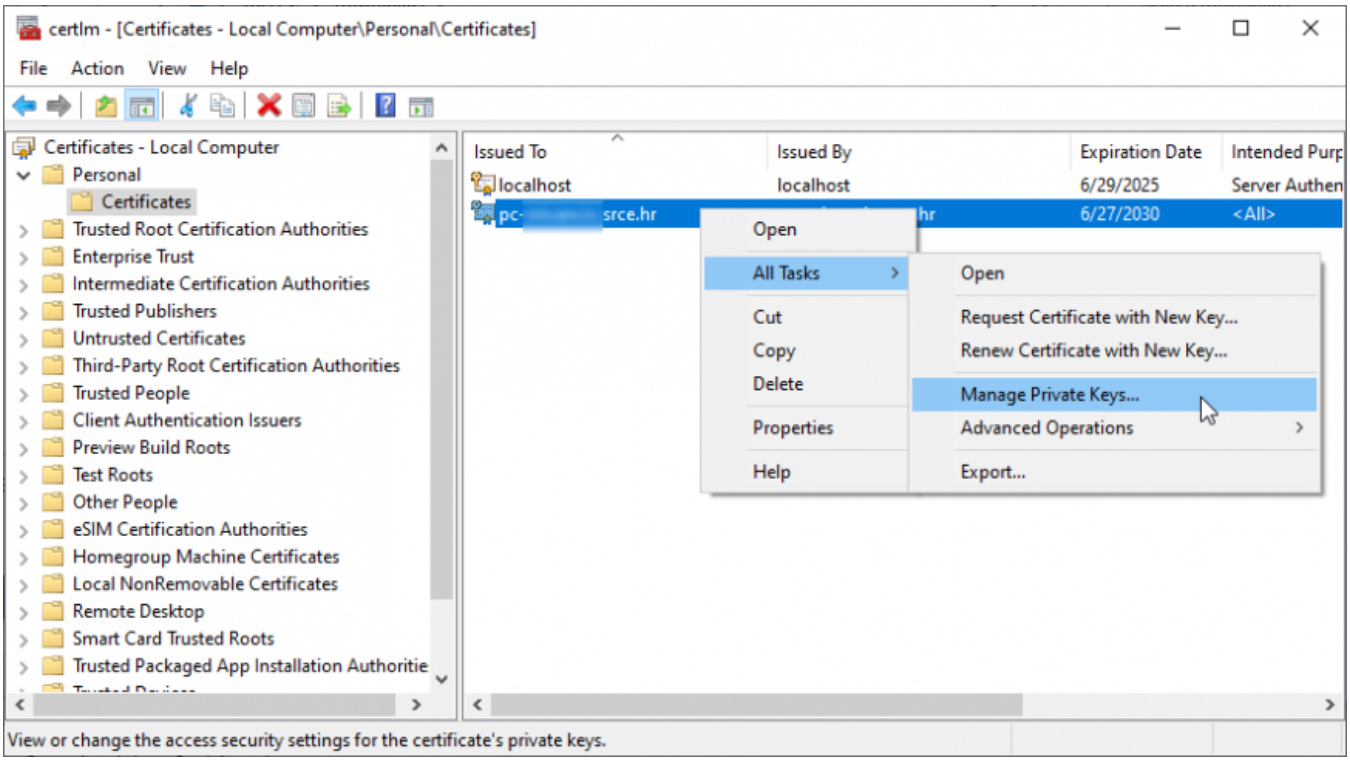
U **`metadata.ashx`** postaviti:

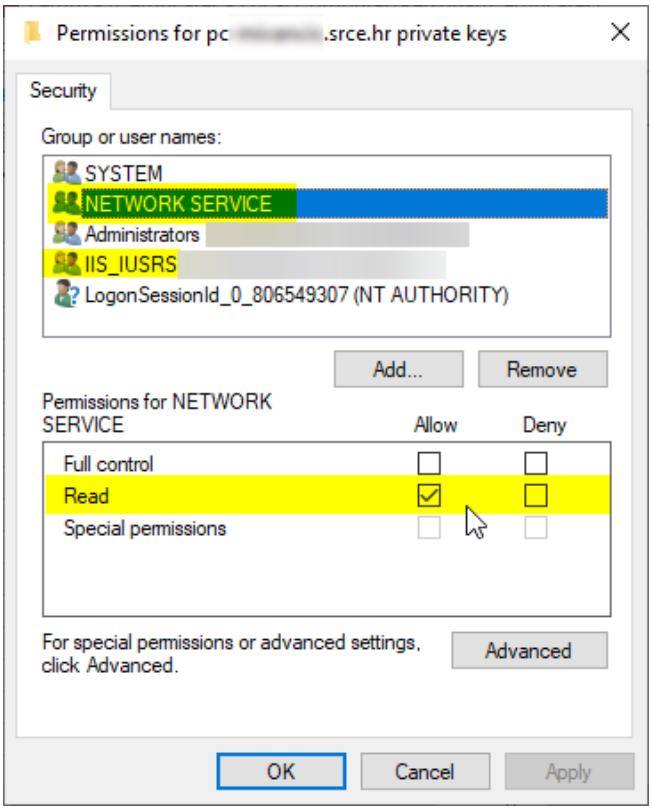
```
<%@ WebHandler Class="dk.nita.saml20.protocol.Saml20MetadataHandler" %>
```

7. Instalirajte certifikat (postupak generiranja certifikata) kojim će se potpisivati zaglavlje. Certifikat treba instalirati u **`Computer Account`** koristeći **`MMC Certificates`**.



8. Odaberite certifikat i postavite prava čitanja za **`Network Service`** i **`IIS_IUSR`**



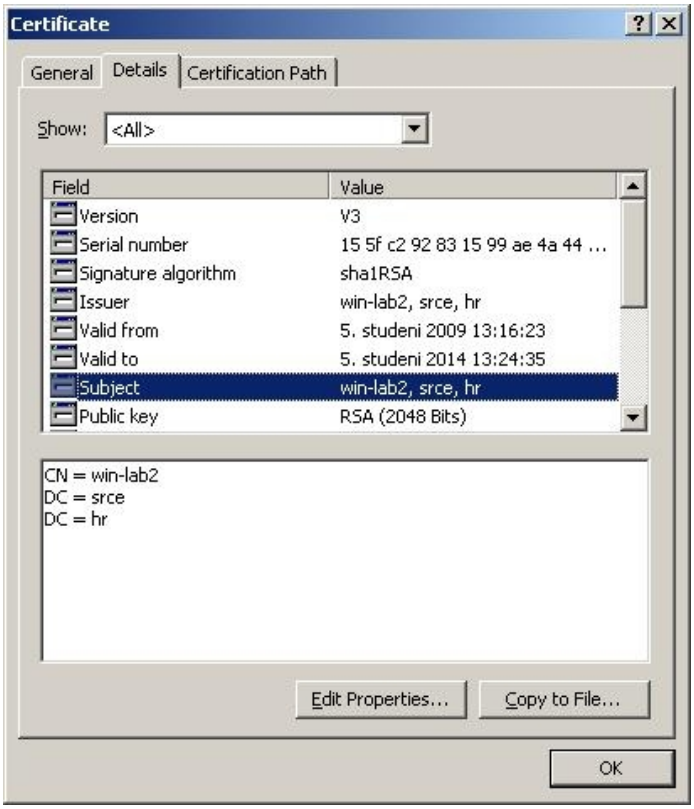


9. Koristite prilagođenu konfiguracijsku `web.config` datoteku:

Preuzmite datoteku i preimenujte je iz web.config.example u web.config. U **`web.config`** datoteci pronađite redak:

```
<SigningCertificate
        findValue="CN=virtualni-host, DC=organization, DC=hr"
        storeLocation="LocalMachine"
        storeName="My"
        x509FindType="FindBySubjectDistinguishedName"
        validonly="no"/>
```

Zamjenite sadržaj atributa **`findValue`** vrijednošću atributa **`Subject`** u certifikatu kojim će se potpisivati zaglavlja (kriterij pretraživanja MS Certificate Store-a putem DN). Navedeno se može pronaći dvostrukim klikom na certifikat te pregledom polja **`Subject`** (slika je samo za primjer):



Dalje, u **`web.config`** datoteci pronađite redak:

```
<ServiceProvider id="demoSP" server="http://virtualni-host/">
```

Zamjenite **`id`** imenom kojim će se vaša aplikacija predstavljati Single Sign-On servisu, dok **`server`** treba biti stvarno ime poslužitelja.

Dalje, u **`web.config`** datoteci pronađite redak:

```
<Audience>demoSP<Audience>
```

U vrijednost **`Audience`** upišite istu vrijednost koja je upisana u polje **`id`** u prethodnom koraku.

10. U korijenu **`c:\`** kreirajte direktorij **`metadata`** i postavite mu prava čitanja za **`Network Service`** i **`IIS_IUSR`**. Ukoliko direktorij **`metadata`** treba biti na nekom drugom mjestu, u konfiguracijskoj datoteci **`web.config`** treba promjeniti putanju do direktorija:

```
<IDPEndPoints metadata="C:\metadata\">
        <add id="https://login.aaiedu.hr/ms/saml2/idp/metadata.php">
                <CertificateValidation>
                <add type="dk.nita.saml20.Specification.SelfIssuedCertificateSpecification, dk.nita.saml20"/>
                </CertificateValidation>
        </add>
</IDPEndPoints>
```

11. Dohvatite metapodatke AAI@EduHr Single Sign-On servisa i spremite ih u gore definiran direktorij `metadata` pod bilo kojim imenom (npr. **`aaiedu_metadata.xml`**). U slučaju testiranja autentikacije u testnom AAI@EduHr okruženju, potrebno je dohvatiti metapodatke za AAI@EduHr Lab Single Sign-On servis. U tom slučaju, prilikom prelaska u produkciju, potrebno je zamjeniti testne metapodatke sa produkcijskim.

12. Objavite kompajliran projekt na web serveru.

13. Registrirajte vašu aplikaciju u sustavu AAI@EduHr prema uputama na web stranici Registar resursa.

SAML metapodatke koje je potrebno unijeti prilikom registracije možete pronaći na web adresi (URL) na kojoj se nalazi metadata handler vaše aplikacije, npr. **`https://vas_posluzitelj.ustanova.hr/metadata.ashx.`** SAML modul resursa koji koriste OIOSAML.NET u registru resursa treba u parametru AuthService imati postavljenu vrijednost OIOSAML.NET.

14. (opcionalno) Prema početnim postavkama, OIOSAML.NET paket za sve tipove grešaka javlja generičku poruku tipa 'Unable to validate SAML message'. Tijekom razvoja i testiranja aplikacije moguće je uključiti detaljni prikaz poruke greške na način da se postavi konfiguracija 'ShowError'. Ovu opciju potrebno je onemoućiti u produkciji.

```
<SAML20Federation xmlns="urn:dk.nita.saml20.configuration">
    ...
    <ShowError>true</ShowError>
    ...
</SAML20Federation>
```

Generiranje certifikata

Postupak generiranja certifikata (**potrebna je `OpenSSL` distribucija za Windowse**):

1. Generirajte 1024 bitni ključ:

```
openssl genrsa -out openssl_key.pem 1024
```

2. Generirajte certifikat sljedećom naredbom (sve u jednom retku, zamjenite serverName.realm stvarnim DNS nazivom vašeg poslužitelja):

```
openssl req -new -x509 -key openssl_key.pem -out openssl_crt.pem -outform pem -days 3650 -subj "
/CN=serverName.realm"
```

3. Snimite certifikat u pk12 formatu, zajedno s privatnim ključem:

```
openssl pkcs12 -export -in openssl_crt.pem  -inkey openssl_key.pem -out certificate_in_pk12.p12
```

Zapisivanje događaja

`dk.nita.saml20` omogućuje zapisivanje događaja na dvije osnovne razine:

- **`Error`** - zapisuje samo pogreške;

- **`Information`** - zapisuje događaje unutar sustava i služi za analizu ponašanja sustava u najsitnije detalje;

Podrazumijevana konfiguracija zapisuje samo pogreške u **`C:\logs\saml2.trace.log`**. Kako direktorij **`C:\logs`** standardno ne postoji, potrebno ga je kreirati i postaviti prava pisanja za **`Network Service`**.

Vrstu zapisivanja događaja i odredišni direktorij moguće je promjeniti u datoteci **`web.config`** u odjeljku diagnostics:

```
<system.diagnostics>
        <trace autoflush="true"></trace>
        <sources>
                <source name="dk.nita.saml20" switchValue="Error">
                        <listeners>
                                <add name="trace"/>
                        </listeners>
                </source>
        </sources>
        <sharedListeners>
                <add         name="trace"
                        type="System.Diagnostics.XmlWriterTraceListener"
                        initializeData="C:\logs\saml2.tracelog"/>
        </sharedListeners>
</system.diagnostics>
```

Uporaba

Podaci o autenticiranom korisniku dohvaćaju se korištenjem property-ja **`dk.nita.saml20.identity.Saml20Identity.Current`**, primjerice:

```
using dk.nita.saml20.identity;
...
if (Saml20Identity.Current != null)
{
        Saml20Identity test = Saml20Identity.Current;
        Response.Write("Ispisujem vrijednost za hrEduPersonUniqueID: ");
        string[] values_ID = test["hrEduPersonUniqueID"][0].AttributeValue;
        Response.Write(values_ID[0]);
        Response.Write("<br>");
        Response.Write("Ispisujem vrijednost za mail: ");
        string[] values_mail = test["mail"][0].AttributeValue;
        Response.Write(values_mail[0]);
        Response.Write("<br>");
        Response.Write("Ispisujem vrijednost za hrEduPersonPrimaryAffiliation: ");
        string[] values_hrEduPersonPrimaryAffiliation =
                                test["hrEduPersonPrimaryAffiliation"][0].AttributeValue;
        Response.Write(values_hrEduPersonPrimaryAffiliation[0]);
        Response.Write("<br> <p size=20>it works</p>");
}
else
{
        Response.Write("Neautenticirani korisnik");
        Response.Redirect("login.ashx");
}
```

Alternativno, može se koristiti metoda IsInitialized(). Za provjeru postojanja atributa može se iskoristiti metoda HasAttribute().

```
if (Saml20Identity.IsInitialized())
  {
        string userInfo = "Korisnik je autenticiran. ";
        Saml20Identity user = Saml20Identity.Current;
        if (user.HasAttribute("hrEduPersonUniqueID"))
        {
            userInfo += "Korisnika oznaka: ";
            userInfo +=  user["hrEduPersonUniqueID"][0].AttributeValue[0];
        } else
        {
            userInfo += "Korisnika oznaka nije meu atributima. ";
        }
  }

...
// Dodatni primjer redirekcije u sluaju da korisnik nije autenticiran.
if (! Saml20Identity.IsInitialized())
{
    Response.Redirect("login.ashx");
}
```

Korisnik se odjavljuje sa sustava pozivanjem **`logout.ashx`**:

```
Response.Redirect("logout.ashx");
```

Redirekcija na stranicu različitu od **`default.aspx`** se postiže uporabom koda:

```
if (dk.nita.saml20.identity.Saml20Identity.Current == null)
{
        dk.nita.saml20.config.SAML20FederationConfig.GetConfig().ServiceProvider.SignOnEndpoint.RedirectUrl =
                "subweb.aspx";
        dk.nita.saml20.config.SAML20FederationConfig.GetConfig().ServiceProvider.SignOnEndpoint.endpointType
=
                EndpointType.SIGNON;
        Response.Redirect("login.ashx");
}
```

Tipične greške

```
Stack Trace:
[CryptographicException: Keyset does not exist
...
]
```

U tom slučaju certifikat ne sadrži potrebni privatni ključ za generiranje potpisa ili nisu dobro postavljena prava za čitanje. Ponovite korak 7 ili ponovite postupak generiranja certifikata.

```
Error: dk.nita.saml20.Profiles.DKSaml20.DKSaml20FormatException: The DK-SAML 2.0 profile requires that an
attribute's "Name" is an URI.
```

Format atributa 'Name' kojeg isporučuje AAI@EduHr je 'urn:oasis:names:tc:SAML:2.0:attrname-format:basic'. Prema početnim postavkama, OIOSAML.NET 1.7.9 očekuje format 'urn:oasis:names:tc:SAML:2.0:attrname-format:uri'. U DLL datoteci 'dk.nita.saml20.dll' koja je dostupna za preuzimanje u ovim uputama, napravljene su preinake koje omogućuju ispravnu validaciju SAML odgovora. Provjerite jeste li u vašem projektu uspješno postavili poveznicu (referencu) na prilagođeni DLL koji je dostupan u ovim uputama.

```
System.Security.Cryptography.CryptographicException: Invalida algorithm specified.
```

Za rješavanje ove greške, prilikom postupka generiranja certifikata, u zadnjem (trećem) koraku unesite sljedeću naredbu (u naredbi je dodan parametar -CSP):

```
openssl pkcs12 -export -in openssl_crt.pem -inkey openssl_key.pem -out certificate_in_pk12.p12 -CSP
"Microsoft Enhanced RSA and AES Cryptographic Provider"
```

Nakon toga ponovno instalirajte certifikat i postavite prava čitanja.

```
dk.nita.saml20Exception: Your sesion has been disconected, please logon again...
```

Radi se o promjeni ponašanja Internet preglednika vezano za atribut 'SameSite' u kolačićima. Slijedite rješenje za .NET aplikacije.

Važno!

SAML modul resursa koji koriste OIOSAML.NET u registru resursa treba u parametru AuthService imati postavljenu vrijednost OIOSAML.NET.