

2010

Uprava Carina
Sektor za
informacione
tehnologije

[ELEKTRONSKO PODNOŠENJE DOKUMENATA – EPD SERVIS] Uputstvo za programere

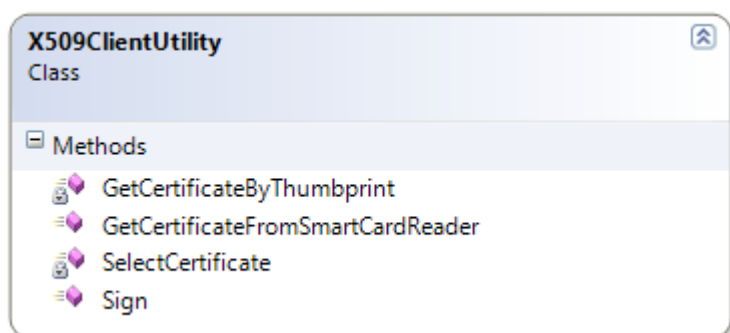
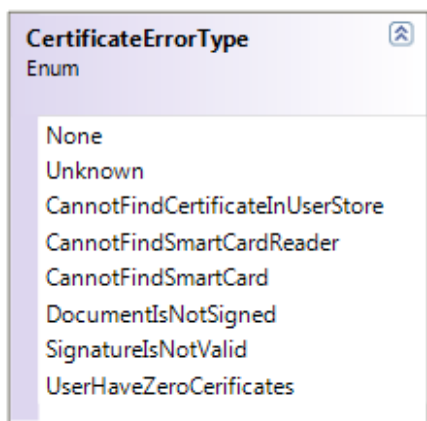
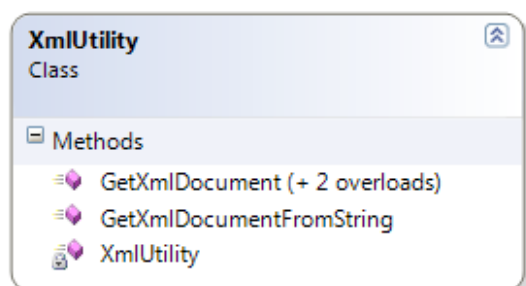
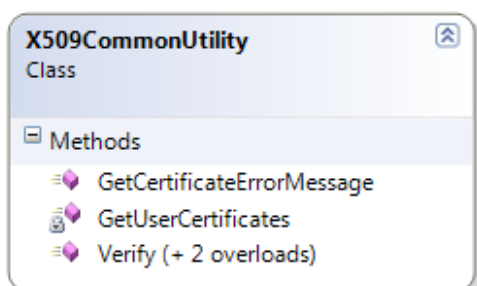
Sadržaj

X509CommonUtility i X509ClientUtility biblioteke	3
Metode klase X509CommonUtility	4
Metode klase XmlClientUtility	4
Metode klase XmlUtility	4
Primer provere validnosti potpisa dokumenta	4
Metode web servisa	5
GetUserInfo	5
SendData	7
UpdateCodes	8
Izmene klijentskog koda	10
Instanciranje web servisa	10
Poziv metode UpdateCodes	12
Poziv metode SendData	12
Poziv metode GetUserInfo	14

Radi povećanja sigurnosti pristupa svojim resursima, Uprava carina je implementirala novu verziju servisa za elektronsko podnošenje deklaracija. Servisu će moći da pristupe samo korisnici koji imaju validne kvalifikovane sertifikate izdate od strane ovlašćenih sertifikacionih tela Republike Srbije, a koji imaju dozvolu Uprave carina za njegovo korišćenje. Sva dokumenta koja se šalju od strane klijenta moraju biti potpisana sertifikatom ovlašćenog agenta.

X509CommonUtility i X509ClientUtility biblioteke

Implementirane su dve nove .net biblioteke u kojima su enkapsulirane metode za rad sa sertifikatima, kao i pomoćna biblioteka za rad sa xml dokumentima.



Metode klase X509CommonUtility

Metode Verify proveravaju da li je xml dokument potpisan odgovarajućim kvalifikovanim sertifikatom i vraćaju u out parametru rezultat provere. U prvom overloadu sa šalje sertifikat, a u drugom korisničko ime i ldap putanja, na osnovu čega se sertifikat pronalazi iz Active Directory.

```
public static bool Verify(X509Certificate2 certificate, XmlDocument xmlDoc, out CertificateErrorType error)
```

```
public static bool Verify(string userName, string ldapPath, XmlDocument xmlDoc, out CertificateErrorType error)
```

Sledeća metoda služi da se na osnovu tipa greške vrati latinična poruka o grešci.

```
public static string GetCertificateErrorMessage(CertificateErrorType error)
```

Metode klase XmlClientUtility

Metoda Sign se koristi za potpisivanje dokumenta kvalifikovanim klijentskim sertifikatom.

```
public static bool Sign(XmlDocument xmlDoc, out CertificateErrorType error)
```

Metoda GetCertificateFromSmartCardReader služi za čitanje kvalifikovanog sertifikata sa smart kartice.

```
public static X509Certificate2 GetCertificateFromSmartCardReader(out CertificateErrorType error)
```

Metode klase XmlUtility

Metode GetXmlDocument su pomoćne metode koje na osnovu različitih ulaznih parametara (putanja do xml fajla, binarni sadržaj dokumenta i xml string) vraćaju odgovarajući xml dokument.

```
public static XmlDocument GetXmlDocument(string fileName)
```

```
public static XmlDocument GetXmlDocument(byte[] byteArray)
```

```
public static XmlDocument GetXmlDocumentFromString(string xmlDoc)
```

Primer provere validnosti potpisa dokumenta

Sledeći deo koda, koji koristi gore opisane metode, izvršava se na serveru sa ciljem da proveri da li je dokument koji je stigao sa klijenta potpisan kvalifikovanim sertifikatom ulogovanog korisnika.

```
var xmlDoc = XmlUtility.GetXmlDocumentFromString(declarationDataXml);  
  
CertificateErrorType error;  
  
if (!X509CommonUtility.Verify(CurrentUser.Username, LdapPath, xmlDoc, out error))  
{  
    var errorMessage = X509CommonUtility.GetCertificateErrorMessage(error);  
    throw new SoapException(errorMessage, SoapException.ServerFaultCode);  
}
```

```
XmlNode signatureNode =  
xmlDocument.GetElementsByTagName("Signature")[0];  
xmlDocument.DocumentElement.RemoveChild(signatureNode);  
declarationDataXml = xmlDocument.OuterXml;
```

Metode web servisa

U ovom poglavlju opisaćemo web metode koje su dostupne klijentu. U prilogu je prikazan xml sa opisom servisa.

GetUserInfo

Ova metoda nema ulazne parameter, a kao rezultat vraća objekat tipa *UserInfo*, i to na osnovu podataka sa klijentskog sertifikata korisnika koji pristupa sistemu. Objekat *UserInfo* ima dva atributa tipa string – *AgentCode* i *TaxNumber*. U nastavku su prikazane SOAP deklaracije poziva web metode i rezultata koji metod vraća klijentu. U narednim poglavljima prikazaćemo poziv ove i ostalih metoda sa .net klijenta.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /ElSubWs/ElSubWS.asmx HTTP/1.1  
Host: localhost  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
SOAPAction: "http://tempuri.org/GetUserInfo"  
  
<?xml version="1.0" encoding="utf-8"?  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <GetUserInfo xmlns="http://tempuri.org/" />  
  </soap:Body>  
</soap:Envelope>  
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <GetUserInfoResponse xmlns="http://tempuri.org/">  
      <GetUserInfoResult>  
        <AgentCode>string</AgentCode>  
        <TaxNumber>string</TaxNumber>  
      </GetUserInfoResult>  
    </GetUserInfoResponse>  
  </soap:Body>  
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /ElSubWs/ElSubWS.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetUserInfo xmlns="http://tempuri.org/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetUserInfoResponse xmlns="http://tempuri.org/">
      <GetUserInfoResult>
        <AgentCode>string</AgentCode>
        <TaxNumber>string</TaxNumber>
      </GetUserInfoResult>
    </GetUserInfoResponse>
  </soap12:Body>
</soap12:Envelope>
```

HTTP POST

The following is a sample HTTP POST request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /ElSubWs/ElSubWS.asmx/GetUserInfo HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<UserInfo xmlns="http://tempuri.org/">
  <AgentCode>string</AgentCode>
  <TaxNumber>string</TaxNumber>
</UserInfo>
```

SendData

Ova metoda se koristi za slanje potpisanih XML dokumenata na servis. Ulazni parametar je string koji u stvari predstavlja sadržaj xml dokumenta koji se šalje na potpisivanje. Kao rezultat sistem vraća xml string koji može da sadrži najrazličitije informacije, npr. listu grešaka, konzistentnost podataka u odnosu na XSD šemu itd... U nastavku su prikazane SOAP deklaracije poziva web metode i rezultata koji metod vraća klijentu.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /ElSubWs/ElSubWS.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/SendData"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendData xmlns="http://tempuri.org/">
      <declarationDataXml>string</declarationDataXml>
    </SendData>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendDataResponse xmlns="http://tempuri.org/">
      <SendDataResult>string</SendDataResult>
    </SendDataResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /ElSubWs/ElSubWS.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```

```
<soap12:Body>
  <SendData xmlns="http://tempuri.org/">
    <declarationDataXml>string</declarationDataXml>
  </SendData>
</soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendDataResponse xmlns="http://tempuri.org/">
      <SendDataResult>string</SendDataResult>
    </SendDataResponse>
  </soap12:Body>
</soap12:Envelope>
```

HTTP POST

The following is a sample HTTP POST request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /ElSubWs/ElSubWS.asmx/SendData HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length

declarationDataXml=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

UpdateCodes

Ova metoda služi za update šifarnika na klijentskom računaru. Ulaz u metodu je parametar tipa string koji predstavlja verziju šifarnika na klijentskom računaru. Rezultat izvršenja web metode je xml string sa šifarnicima, ukoliko se njegova verzija razlikuje od one sa servera. U nastavku su prikazane SOAP deklaracije poziva web metode i rezultata koji metod vraća klijentu.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /ElSubWs/ElSubWS.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
```


SOAPAction: "http://tempuri.org/UpdateCodes"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateCodes xmlns="http://tempuri.org/">
      <version>string</version>
    </UpdateCodes>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateCodesResponse xmlns="http://tempuri.org/">
      <UpdateCodesResult>string</UpdateCodesResult>
    </UpdateCodesResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /ElSubWs/ElSubWS.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UpdateCodes xmlns="http://tempuri.org/">
      <version>string</version>
    </UpdateCodes>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UpdateCodesResponse xmlns="http://tempuri.org/">
```

```
<UpdateCodesResult>string</UpdateCodesResult>
</UpdateCodesResponse>
</soap12:Body>
</soap12:Envelope>
```

HTTP POST

The following is a sample HTTP POST request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /ElSubWs/ElSubWS.asmx/UpdateCodes HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length

version=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

Izmene klijentskog koda

U nastavku ćemo prikazati izmene koje smo implementirali na klijentskoj aplikaciji ElSub u skladu sa izmenama na web servisu. Daćemo uporedni prikaz starog i izmenjenog koda.

Instanciranje web servisa

Sledeći kod pokazuje nam kako se instancirao web servis u staroj ElSub aplikaciji:

```
public static ElSubWS.ElSubWS ws = null;

public static bool CreateWSLink(string url) {
    if (ws == null) {
        try {
            if (url != null && url != string.Empty) {
                ws = new ElSubWS.ElSubWS(url);
                return true;
            }
            throw new Exception("Nepravilan URL servera!");
        } catch (Exception ex) {
            MessageBox.Show(ex.Message, "OBAVEŠTENJE",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
            return false;
        }
    } else {
        return true;
    }
}
```

Sledi prikaz instanciranja web servisa u novoj verziji aplikacije. Prilikom prvog poziva web servisa, sistem će od korisnika tražiti da ubaci smart karticu sa kvalifikovanim klijentskim sertifikatom u čitač, kako bi ga servis autentifikovao.

```

using Fusion.SerbianCustoms.EPD.ClientSecurity;
using Fusion.SerbianCustoms.EPD.EpdCommonSecurity;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;

private static ElSubWS cWs = null;

public static ElSubWS ws
{
    get
    {
        if (cWs == null)
        {
            try
            {
                string serviceUrl =
Util.GetConfigProperty(string.Format("EndpointUrl{0}",
Util.GetConfigProperty("ApplicationMode")));
                if (!string.IsNullOrEmpty(serviceUrl))
                {
                    cWs = new ElSubWS
                    {
                        Url = serviceUrl
                    };
                }
            }
            catch (Exception ex)
            {
                ServicePointManager.ServerCertificateValidationCallback += new
System.Net.Security.RemoteCertificateValidationCallback(ValidateWebServiceCal
l);
            }
            else
            {
                throw new Exception("Nepravilan URL servera!");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "OBAVEŠTENJE",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    if (cWs != null)
    {
        CertificateErrorType error;
        cWs.ClientCertificates.Clear();
        var certificate =
X509ClientUtility.GetCertificateFromSmartCardReader(out error);
        if (error != CertificateErrorType.None)
        {
            cWs = null;
        }
        MessageBox.Show(X509CommonUtility.GetCertificateErrorMessage(error));
    }
    else
    {
        cWs.ClientCertificates.Add(certificate);
    }
}
}

```

```
        return cWs;  
    }  
}
```

```
private static bool ValidateWebServiceCall(  
    object sender,  
    X509Certificate certificate,  
    X509Chain chain,  
    SslPolicyErrors sslPolicyErrors)  
{  
    return true;  
}
```

Poziv metode UpdateCodes

U staroj i novoj verziji aplikacije praktično je samo promenjen broj parametara koji se šalje na server. U staroj se šalju sledeći parametri: username, partycode, token i verzija šifarnika na klijentu.

```
string response = ws.UpdateCodes(currentUser.UserName, currentUser.PartyCode,  
currentUser.Token,  
codes.SelectSingleNode("//NewDataSet").Attributes["Version"].InnerText);
```

U novoj verziji šalje se samo parametar verzija šifarnika na klijentu.

```
string response =  
ws.UpdateCodes(codes.SelectSingleNode("//NewDataSet").Attributes["Version"].I  
nnerText);
```

Poziv metode SendData

Sledeći kod prikazuje poziv metode za slanje dokumenata na servis. Šalju se sledeći parametri: username, partycode, token i xml string dokumenta koji se šalje.

```
cusDec = InitializeDocument(xml);  
  
//Send (validate) cusDec  
string response = string.Empty;  
  
try {  
    response = ProDek.ws.SendData(ProDek.currentUser.UserName,  
ProDek.currentUser.PartyCode, ProDek.currentUser.Token, cusDec.OuterXml);  
} catch (System.Web.Services.Protocols.SoapException se) {  
    HandleSexception(se);  
    this.Cursor = Cursors.Default;  
    return;  
} catch (Exception ex) {  
    MessageBox.Show(ex.Message);  
    this.Cursor = Cursors.Default;  
    return;  
}
```

Poziv metode u novoj verziji ima samo jedan ulazni parameter – xml string dokumenta, ali je neophodno da se taj dokument pre slanja potpiše kvalifikovanim klijentskim sertifikatom. U nastavku je prikazan kod za potpisivanje dokumenta i za poziv samog web servisa. Za potpisivanje se koristi biblioteka X509ClientUtility i poziva se njena metoda Sign kojoj se šalje dokument na potpisivanje, a u out parametrima se vraća enumerator CertificateErrorType, koji ima sledeće vrednosti:

- None (nema greške),
- Unknown (nepoznata greška prilikom čitanja sertifikata sa kartice),
- CannotFindCertificateInUserStore (nije moguće pronaći sertifikat),
- CannotFindSmartCardReader (ne postoji konekcija sa čitačem kartica, čitač nije spojen),
- CannotFindSmartCard (ne postoji konekcija sa čitačem kartica, kartica nije ubačena u čitač),
- DocumentIsNotSigned (dokument nije potpisan),
- SignatureIsNotValid (potpis nije validan),
- UserHaveZeroCerificates (korisnik nema instaliran nijedan sertifikat).

Ukoliko je dokument validno potpisan, poziva se web metoda za slanje dokumenta. U slučaju greške sistem prikazuje odgovarajuću poruku.

```
cusDec = InitializeDocument(xml);

if (cusDec != null)
{
    //Send (validate) cusDec
    string response = string.Empty;
    try
    {
        CertificateErrorType error;
        if (X509ClientUtility.Sign(cusDec, out error))
        {
            response = ProDek.ws.SendData(cusDec.OuterXml);
        }
        else
        {
            MessageBox.Show(X509CommonUtility.GetCertificateErrorMessage(error));
        }
    }
    catch (System.Web.Services.Protocols.SoapException se)
    {
        HandleSexception(se);
        this.Cursor = Cursors.Default;
        return;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        this.Cursor = Cursors.Default;
        return;
    }
}
```

Poziv metode GetUserInfo

Sledeći kod prikazuje poziv metode GetUserInfo. Ova metoda nema ulazne parametre. Na osnovu sertifikata sa kartice metoda vraća odgovarajuće podatke o korisniku.

```
private static UserInfo cCurrentUser;

internal static UserInfo CurrentUser
{
    get
    {
        if (cCurrentUser == null)
        {
            try
            {
                cCurrentUser = ws.GetUserInfo();
            }
            catch (WebException)
            {
                cWs = null;
                MessageBox.Show("Problem u komunikaciji sa servisom.
Pozovite administratora sistema.");
            }
        }
    }

    return cCurrentUser;
}
}
```