

Client exemple .NET pour e-dec

Documentation succincte

En bref

- GUI avec WPF
- Les entrées sont enregistrées en tant que paramètres de propriété.
- Une enveloppe SOAP prête est lue et envoyée avec `HttpWebRequest`.
- L'utilisateur peut copier ou enregistrer la réponse du serveur.

Certificats

Dans C#, il est très simple d'établir un certificat. On utilise à cet effet le Namespace (espace de nommage) `System.Security.Cryptography.X509Certificates`. Différentes possibilités existent. Il est pratique de connaître le chemin d'accès au certificat. Un élément de certificat peut donc être instancié comme suit:

```
var path = @"C:\path\to\certificate.pem";
var certificateA = new X509Certificate2(path);
```

Si le certificat est en plus protégé par un mot de passe, on peut également le définir dans le constructeur:

```
var password = "A really strong password";
var certificateB = new X509Certificate2(path, password);
```

X509Certificate ou X509Certificate2

Si on intègre le Namespace susmentionné, on remarque qu'il existe, avec les `X509Certificate` et `X509Certificate2`, deux classes très similaires. En bref, `X509Certificate2` est la version la plus récente, qui doit être en principe recommandée.

Cela facilite premièrement le traitement des certificats récents et, deuxièmement, le développement en est ainsi un peu plus confortable.

Exemple

```
// diminue la lisibilité
new X509CertificateCollection(new[] { new X509Certificate() });

// aucun tableau requis
new X509Certificate2Collection(new X509Certificate2());
```

HttpWebRequest

Pour exécuter des requêtes Http dans .NET, nous avons besoin d'un élément `HttpWebRequest`.

```
var httpWebRequest = WebRequest.CreateHttp("http://example.com/service");
```

On peut préciser maintenant comment la requête doit être déterminée:

```
httpWebRequest.Method = "POST";  
httpWebRequest.ContentType = "text/xml; charset=UTF-8";  
httpWebRequest.KeepAlive = true;  
httpWebRequest.ClientCertificates = new  
X509Certificate2Collection(certificate);
```

Il y a encore d'autres propriétés. Mais elles ne sont pas toutes nécessaires pour exécuter une Edec-SOAP-Request (requête SOAP e-dec).

Nous voulons envoyer quelque chose. Pour ce faire, nous utilisons le `RequestStream`:

```
var body = File.ReadAllBytes(@"C:\path\to\soap\envelope.xml");  
  
httpWebRequest.ContentLength = body.LongLength;  
  
using (var stream = httpWebRequest.GetRequestStream())  
{  
    stream.Write(body, 0, body.Length);  
}
```

Enfin, nous pouvons lire la réponse:

```
// hier ohne Exception-Handling  
using (var responseStream = httpWebRequest.GetResponse().GetResponseStream())  
{  
    using (var reader = new StreamReader(responseStream))  
    {  
        Console.WriteLine(reader.ReadToEnd());  
    }  
}
```

Attention

La méthode `GetResponse()` peut déclencher une `WebException`. On peut tout de même accéder à la réponse: `webException.Response`.