

# Client di esempio .NET per Edec

Breve documentazione

## In breve

- GUI con WPF
- Gli inserimenti sono memorizzati come Property-Settings
- Viene scansionato un Soap-Envelope completo e inviato con un `HttpWebRequest`
- L'utente può copiare o memorizzare la risposta del server

## Certificati

È molto semplice creare un certificato in C#. A tale proposito si utilizza il Namespace `System.Security.Cryptography.X509Certificates`. Vi sono varie possibilità; è utile conoscere la traccia del certificato. Ciò permette di istanziare un oggetto come segue:

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

Se il certificato è protetto da una password, anche quest'ultima può essere aggiunta nel costruttore:

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

## X509Certificate oder X509Certificate2

Quando si integra il suddetto Namespace, si constata che con `X509Certificate` e `X509Certificate2` esistono due classi molto simili. In pratica: `X509Certificate2` è la versione più nuova da raccomandare in linea di massima.

In primo luogo è possibile utilizzare i certificati più nuovi e, in secondo luogo, anche lo sviluppo risulta più semplice.

## Esempio

```
// diminuisce la leggibilità
new X509CertificateCollection(new[] { new X509Certificate() });

// Struttura Array non necessaria
new X509Certificate2Collection(new X509Certificate2());
```

## HttpWebRequest

Per eseguire la request in .NET Http-Requests, serve un oggetto `HttpWebRequest`.

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

Ora è possibile precisare la Request:

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

Sono disponibili anche altre proprietà, che tuttavia non sono tutte necessarie per eseguire una Edec-Soap-Request.

Per effettuare un invio, utilizzare il 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);  
}
```

In seguito si può leggere la risposta:

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

#### Attenzione

Il metodo `GetResponse()` può comportare una `WebException`. È comunque possibile consultare la response: `webException.Response`.