

## My Docs Online Secure File Delivery API: C#

### Introduction

[My Docs Online](#) has provided HIPAA-compliant Secure File Sharing and Delivery since 1999. With the most recent release of its web client and Java Desktop Application the secure file sharing options have been expanded and made more flexible and powerful.

*See the **Developer's Getting Started Guide** for an introduction to My Docs Online Secure File Delivery and how to use it programmatically.*

*See the **REST API Reference** for a detailed description of processing flow and functionality.*

The MDO C# API is distributed in the MdoApi.dll, which should be included as a reference in your project.

The MdoApi consists of three classes, MdoClient, MdoShareEvent and MdoShareDetails.

MdoClient is the main class and handles the processing of server requests.

MdoShareEvent is used to exchange share parameters with the server.

MdoShareDetails is used to return detail information about files in a share.

MdoClient has the following methods.

string GetErrorCode()

Returns the three digit error code from the server

000 is no error

string GetErrorMessage()

Return the error message string from the server

bool Authenticate(string MdoUserId, string password)

Sends and authentication request to the server.

returns true if authenticated, false if failure

bool ShareUpload(string folderName, string filePath, string fileName)

Uploads file for share processing

folderName is the name of the folder to contain the uploaded file

filePath is the full path to the file

filename is the file name to be used in the share

Use multiple ShareUploads for more than one file

returns true if ok, false if failure

bool ShareComplete(MdoShareEvent mdoShare)

Completes the share

Input is a MdoShareEvent class which will contain share options and the resulting URL  
returns true if ok, false if failure

bool ShareCancel(string shareId)

Cancels a share

Input is the ShareId returned in the MdoShareEvent from a successful share  
returns true if ok, false if failure

bool ShareDetails(string shareId)

Obtains detail information about each file in a share.

Input is the ShareId returned in the MdoShareEvent from a successful share  
returns true if ok, false if failure

if return code is ok call the GetShareDetails method to get a list of MdoShareDetails

List<MdoShareDetails> GetShareDetails

returns a list of MdoShareDetails containing information about a specific share  
only call this if ShareDetails returned true.

MdoShareEvent has the following Properties

string ShareUrl { get; }

The Share URL returned on a successful share

string ShareId { get; }

The unique share identifier returned on a successful share

string Comment { set; }

The Optional share comment

string SharePassword { set; }

The Optional share password

string DownloadLimit { set; }

The Optional download limit

string ExpireDays { set; }

The Optional days until share expires

MdoShareDetails has the following Properties

string FileName { get; }

The name of the file

string FileSize { get; }

The size of the file

string ShareTime { get; }

The date/time the share occurred

string ExpireTime { get; }

The date/time the share expires  
string FirstDownload { get; }  
The date/time file first downloaded  
string LatestDownload { get; }  
The date/time file last downloaded  
string Comment { get; }  
Comment associated with share  
string DownloadLimit { get; }  
Limit on number of times a file can be downloaded  
string DownloadCount { get; }  
Number of times the file was downloaded  
bool SenderCanceled { get; }  
Share was canceled  
bool Expired { get; }  
Share has expired  
bool HitMaxDownload { get; }  
file downloaded more than Downloadlimit  
bool Finished { get; }  
Share is no longer available.  
reason is SenderCanceled, Expired, HitMaxDownload

### Sample program

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MdoApi;

namespace TestApi
{
    class Program
    {
        static void Main(string[] args)
        {

            // New MDO Client
            MdoClient mdoClient = new MdoClient();

            // Do authenticate
            if (mdoClient.Authenticate("testid", "testpw"))
            {
                // Upload the file
                if ( mdoClient.ShareUpload("UploadFolder", "c:\\test\\xymgk.dss", "cindy.dss")
```

```

{

MdoShareEvent mdoShare = new MdoShareEvent();

mdoShare.ExpireDays = "10";
mdoShare.DownloadLimit = "5";
mdoShare.SharePassword = "abcde";
mdoShare.Comment = "Here is a comment for you";

// Complete the Share
if (mdoClient.ShareComplete(mdoShare))
{
    Console.WriteLine(mdoShare.ShareUrl);
    Console.WriteLine(mdoShare.ShareId);
    if (mdoClient.ShareDetails(mdoShare.ShareId))
    {
        List<ShareDetails> sd = mdoClient.GetShareDetails();
        if (sd.Count > 0)
        {
            foreach (ShareDetails runsd in sd)
            {
                Console.WriteLine("FileName = " + runsd.FileName);
            }
        }
    }
}
else
{
    Console.WriteLine("Share failed error code = " + mdoClient.GetErrorCode());
    Console.WriteLine("Share failed error msg = " + mdoClient.GetErrorMessage());
}
}
else
{
    Console.WriteLine("Share upload failed error code = " + mdoClient.GetErrorCode());
    Console.WriteLine("Share upload error msg = " + mdoClient.GetErrorMessage());
}

}
else
{
    Console.WriteLine("Auth failed error code = " + mdoClient.GetErrorCode());
    Console.WriteLine("Auth failed error msg = " + mdoClient.GetErrorMessage());
}
}

```

```
    }  
  }  
}
```