

Karchive Kloudarchive WebServices

Manuale del programmatore
Versione 1.0



GRAMMELOT

Sommario

Elenco web services	3
Inizializzazione ambiente e credenziali	4
1. ArchiveDocByteArray	5
2. AuthenticateUser	8
3. ChangeUserPassword	10
4. DeleteWaitingDocFile	12
5. GetCommercialista	14
6. GetCompanyList	16
7. GetDocFileLink	18
8. GetDocTypeList	20
9. GetDocsByPdA	22
10. GetIndexDescriptions	24
11. GetOriginalDocFileLink	26
12. GetPdAByCompanyDocType	28
14. GetWaitingDocs	32
15. IsAlive	34
16. ResetUserPassword	36
17. SearchDocuments	38
18. UserExist	42

Elenco web services

I webservice sono accessibili all'indirizzo:

`http(s)://<server>/WebSevice/KloudarchiveWebService.asmx`

Di seguito vengono elencati i webservice utili ad interfacciare Karchive/Kloudarchive:

- ArchiveDocByteArray
- AuthenticateUser
- ChangeUserPassword
- DeleteWaitingDocFile
- GetCommercialista
- GetCompanyList
- GetDocFileLink
- GetDocTypeList
- GetDocsByPdA
- GetIndexDescriptions
- GetOriginalDocFileLink
- GetPdAByCompanyDocType
- GetWaitingDocFileLink
- GetWaitingDocs
- IsAlive
- ResetUserPassword
- SearchDocuments
- UserExist

Inizializzazione ambiente e credenziali

Per inizializzare le chiamate ai webservice è necessario predisporre le credenziali di accesso che sono equivalenti a quelle di accesso al sistema in capo all'utente eseguendo le seguenti chiamate:

```
KloudarchiveWebService.KloudarchiveWebService KAWS = new
KloudArchiveWebServiceGUI.KloudarchiveWebService.KloudarchiveWebServ
ice ();

KAWS.SecuredWebServiceHeaderValue = new
KloudarchiveWebService.SecuredWebServiceHeader ();
KAWS.SecuredWebServiceHeaderValue.Username = "<username>";
KAWS.SecuredWebServiceHeaderValue.Password = "<password>";
KAWS.SecuredWebServiceHeaderValue.AuthenticatedToken=
KAWS.AuthenticateUser ();
```

Una volta inizializzate le credenziali si possono utilizzare con successo i webservice; diversamente per qualunque chiamata verrà restituito un messaggio di errore che richiede l'autenticazione.

1. ArchiveDocByteArray

ArchiveDoc serve per eseguire l'upload di un documento a partire dalla sua rappresentazione in byte array.

Definizione:

```
string ArchiveDocByteArray(byte[] FileByteArray, string fileName,
string Company, string DocType, string K_AnnoFiscale, string[]
metadati, string[] Definizioni, string DocId, string Email);
```

Input:

Tipo	Descrizione
<code>byte[] FileByteArray</code>	Byte array del file di cui fare l'upload
<code>string fileName</code>	Nome del file da associare
<code>string Company</code>	Azienda di appartenenza (ATTENZIONE: deve essere quella associata all'utente)
<code>string DocType</code>	Tipo documento (ATTENZIONE: per documenti dello stesso tipo deve essere associato SEMPRE lo stesso tipo)
<code>string K_AnnoFiscale</code>	Anno fiscale di appartenenza (Può anche essere aggiunto il mese come ad esempio: 2016_09)
<code>string[] metadati</code>	Elenco dei VALORI dei metadati specificati nelle descrizioni del parametro che segue
<code>string[] Definizioni</code>	Elenco delle definizioni da associare ai metadati esplicitati nel parametro precedente
<code>string DocId</code>	Identificativo UNIVOCO del documento (associare sempre il nome azienda per garantire l'univocità)
<code>string Email</code>	OPZIONALE: Email a cui inviare in allegato il documento caricato Inserire un campo vuoto "" se non deve essere utilizzato

Output:

Tipo	Descrizione
<code>string</code>	Se l'operazione si è conclusa con successo la stringa riporterà : "OK" Altrimenti verrà riportato l'errore relativo

Esempio di utilizzo ArchiveDocByteArray:

```
string email = "mia.email@gmail.com";
byte[]
FileRead=System.IO.File.ReadAllBytes("c:\\test\\TestArchiveByteArray.pdf");
string[] metadati={"metadato1","metadato2","metadato3"};
string[] definizioni={"descrizione1","descrizione2","descrizione3"};
string result =
KAWS.ArchiveDocByteArray(FileRead,"TestUploadByteArray.pdf","TestCompany",
"UploadDocType","2016",metadati,definizioni,"ID00001","");
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <ArchiveDocByteArray xmlns="http://tempuri.org/">
      <FileByteArray>base64Binary</FileByteArray>
      <fileName>string</fileName>
      <Company>string</Company>
      <DocType>string</DocType>
      <K_AnnoFiscale>string</K_AnnoFiscale>
      <metadati>
        <string>string</string>
        <string>string</string>
      </metadati>
      <Definizioni>
        <string>string</string>
        <string>string</string>
      </Definizioni>
      <DocId>string</DocId>
      <Email>string</Email>
    </ArchiveDocByteArray>
  </soap12:Body>
</soap12:Envelope>
```

Risposta soap 1.2

```
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>
    <ArchiveDocByteArrayResponse xmlns="http://tempuri.org/">
      <ArchiveDocByteArrayResult>string</ArchiveDocByteArrayResult>
    </ArchiveDocByteArrayResponse>
  </soap12:Body>
</soap12:Envelope>
```

2. AuthenticateUser

AuthenticateUser ha lo scopo di convalidare le credenziali inserite in fase di inizializzazione. E' il primo metodo da utilizzare prima di qualunque altra chiamata (a parte IsAlive)

Definizione:

```
string AuthenticateUser();
```

Input:

Tipo	Descrizione

Output:

Tipo	Descrizione
string	Token di autenticazione da utilizzare in accoppiata con username e password per valorizzare il parametro di ambiente: AuthenticatedToken

Esempio di utilizzo AuthenticateUser:

Si veda il paragrafo Inizializzazione.

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <AuthenticateUser xmlns="http://tempuri.org/" />
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <AuthenticateUserResponse xmlns="http://tempuri.org/">
      <AuthenticateUserResult>string</AuthenticateUserResult>
    </AuthenticateUserResponse>
  </soap12:Body>
</soap12:Envelope>
```

3. ChangeUserPassword

ChangeUserPassword ha lo scopo di modificare la password dell'utente specificato in fase di inizializzazione dell'ambiente

Definizione:

```
string ChangeUserPassword(string prevPassword, string NewPassword, );
```

Input:

Tipo	Descrizione
<code>string prevPassword</code>	Password precedente
<code>string NewPassword</code>	Nuova password da associare
<code>string EmailCommercialista</code>	email del commercialista da associare Se non deve essere associata alcuna email valorizzare con una stringa vuota ""

Output:

Tipo	Descrizione
<code>string</code>	Se l'operazione si è conclusa con successo la stringa riporterà : "OK" Altrimenti verrà riportato l'errore relativo

Esempio di utilizzo ChangeUserPassword:

```
KAWS.ChangeUserPassword("OldPassword", "NewPassword", "");
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <ChangeUserPassword xmlns="http://tempuri.org/">
      <prevPassword>string</prevPassword>
      <NewPassword>string</NewPassword>
      <EmailCommercialista>string</EmailCommercialista>
    </ChangeUserPassword>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <ChangeUserPasswordResponse xmlns="http://tempuri.org/">
      <ChangeUserPasswordResult>string</ChangeUserPasswordResult>
    </ChangeUserPasswordResponse>
  </soap12:Body>
</soap12:Envelope>
```

4. DeleteWaitingDocFile

Questo metodo ha lo scopo cancellare un file che è in attesa di essere conservato

Definizione:

```
string DeleteWaitingDocFile(string FileName);
```

Input:

Tipo	Descrizione
string FileName	Nome del file da eliminare

Output:

Tipo	Descrizione
string	Se l'operazione si è conclusa con successo la stringa riporterà : "OK" Altrimenti verrà riportato l'errore relativo

Esempio di utilizzo DeleteWaitingDocFile:

```
KAWS.DeleteWaitingDocFile("FileName.pdf");
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <DeleteWaitingDocFile xmlns="http://tempuri.org/">
      <FileName>string</FileName>
    </DeleteWaitingDocFile>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <DeleteWaitingDocFileResponse xmlns="http://tempuri.org/">
      <DeleteWaitingDocFileResult>string</DeleteWaitingDocFileResult>
    </DeleteWaitingDocFileResponse>
  </soap12:Body>
</soap12:Envelope>
```

5. GetCommercialista

Questo metodo ottiene l'email del commercialista qualora sia impostata

Definizione:

```
string GetCommercialista();
```

Input:

Tipo	Descrizione

Output:

Tipo	Descrizione
<code>string</code>	Se l'operazione si è conclusa con successo la stringa riporterà l'email del commercialista Altrimenti verrà riportato l'errore relativo

Esempio di utilizzo GetCommercialista:

```
KAWS.GetCommercialista();
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <GetCommercialista xmlns="http://tempuri.org/" />
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <GetCommercialistaResponse xmlns="http://tempuri.org/">
      <GetCommercialistaResult>string</GetCommercialistaResult>
    </GetCommercialistaResponse>
  </soap12:Body>
</soap12:Envelope>
```

6. GetCompanyList

Questo metodo restituisce l'elenco delle aziende visibili dall'utente e su cui ha il diritto di visualizzazione

Definizione:

```
string[] GetCompanyList();
```

Input:

Tipo	Descrizione
<code>string FileName</code>	Nome del file da eliminare

Output:

Tipo	Descrizione
<code>String[]</code>	Se l'operazione si è conclusa con successo verrà riportato l'elenco delle aziende. Se l'elenco è vuoto o c'è stato un errore viene restituito null

Esempio di utilizzo GetCompanyList:

```
string[] CompanyList;  
CompanyList = KAWS.GetCompanyList();
```


Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <GetCompanyList xmlns="http://tempuri.org/" />
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <GetCompanyListResponse xmlns="http://tempuri.org/">
      <GetCompanyListResult>
        <string>string</string>
        <string>string</string>
      </GetCompanyListResult>
    </GetCompanyListResponse>
  </soap12:Body>
</soap12:Envelope>
```

7. GetDocFileLink

Questo metodo restituisce il link web al documento richiamato dai parametri di input. Questo Link sarà valido fino ad una nuova richiesta di documento, momento in cui verrà cancellato dal deposito temporaneo e non più disponibile fino ad un suo successivo richiamo. Diversamente dal metodo

GetOriginalDocFileLink che restituisce il documento originale, così come è o era stato caricato, in questo caso se il documento è un pdf viene restituito il pdf, se è un XML PA firmato .p7m verrà restituita la sua rappresentazione in pdf, se è un pdf firmato in p7m verrà restituito il pdf verificato.

Propedeutico all'utilizzo di questa funzione è l'uso precedente del metodo *GetDocsByPdA*.

Definizione:

```
string GetDocFileLink(string PdAName, string Company, string DocType,
string FileName, string PubblicazioneWeb);
```

Input:

Tipo	Descrizione
<code>string PdAName</code>	Nome del PdA di cui fa parte il documento
<code>string Company</code>	Azienda a cui fa capo il documento
<code>string DocType</code>	Tipo del documento
<code>string FileName</code>	Nome del file originale del documento
<code>string PubblicazioneWeb</code>	Area di pubblicazione web

Output:

Tipo	Descrizione
<code>String</code>	Link al documento http(s) o errore relativo

Esempio di utilizzo GetDocFileLink:

```
KAWS.GetDocFileLink("PdAName","KRILL","Fattura_Emessa", "FileName.pdf",
"PubblicazioneWeb");
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <GetDocFileLink xmlns="http://tempuri.org/">
      <PdAName>string</PdAName>
      <Company>string</Company>
      <DocType>string</DocType>
      <FileName>string</FileName>
      <PubblicazioneWeb>string</PubblicazioneWeb>
    </GetDocFileLink>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <GetDocFileLinkResponse xmlns="http://tempuri.org/">
      <GetDocFileLinkResult>string</GetDocFileLinkResult>
    </GetDocFileLinkResponse>
  </soap12:Body>
</soap12:Envelope>
```

8. GetDocTypeList

Questo metodo restituisce l'elenco dei tipi documento associati ad una azienda visibile all'utente

Definizione:

```
string[] GetDocTypeList(string Company);
```

Input:

Tipo	Descrizione
<code>string Company</code>	Nome dell'azienda associata all'utente di cui voglio sapere le tipologie documentali presenti visibili dall'utente

Output:

Tipo	Descrizione
<code>String[]</code>	Se l'operazione si è conclusa con successo verrà riportato l'elenco dei tipi documento appartenenti all'azienda specificata visibili all'utente. Se l'elenco è vuoto o c'è stato un errore viene restituito null

Esempio di utilizzo GetDocTypeList:

```
string[] DocTypeListByCompany;
DocTypeListByCompany = KAWS.GetDocTypeList("Krill");
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <GetDocTypeList xmlns="http://tempuri.org/">
      <Company>string</Company>
    </GetDocTypeList>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <GetDocTypeListResponse xmlns="http://tempuri.org/">
      <GetDocTypeListResult>
        <string>string</string>
        <string>string</string>
      </GetDocTypeListResult>
    </GetDocTypeListResponse>
  </soap12:Body>
</soap12:Envelope>
```

9. GetDocsByPdA

Questo metodo restituisce l'elenco dei documenti appartenenti ad un particolare pacchetto di versamento specificato in input. Propedeutico è l'uso precedente del metodo: *GetPdAByCompanyDocType*

Definizione:

```
string[] GetDocsByPdA(string PdAName);
```

Input:

Tipo	Descrizione
<code>string PdAName</code>	Nome del pacchetto di archiviazione di cui si vuole avere l'elenco dei documenti presenti

Output:

Tipo	Descrizione
<code>String[]</code>	<p>Se l'operazione si è conclusa con successo verrà riportato l'elenco dei documenti presenti. Per ogni stringa presente nell'array saranno indicate le seguenti informazioni in base al prefisso:</p> <ul style="list-style-type: none"> • ID=Identificativo; • FN=FileName; • V=Versione; • PW=PubblicazioneWeb <p>Se l'elenco è vuoto o c'è stato un errore viene restituito null</p>

Esempio di utilizzo GetDocsByPdA:

```
string[] docs =
KAWS.GetDocsByPdA("2013_GASCOM_Fattura_Emessa_LottoR_3");
string toshow = "";
if(alive != null)
foreach (string s in docs) {
    string Identificativo = s.Substring(s.IndexOf("ID=")+3,
s.IndexOf(";FN=") - s.IndexOf("ID=")-3);
    string FileName = s.Substring(s.IndexOf("FN=") + 3,
s.IndexOf(";V=") - s.IndexOf("FN=") - 3);
    string Versione = s.Substring(s.IndexOf("V=") + 2,
s.IndexOf(";PW=") - s.IndexOf("V=") - 2);
    string PubblicazioneWeb = s.Substring(s.IndexOf("PW=") + 3);
    //Uso I parametri acuisiti come reputo utile
}
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <GetDocsByPdA xmlns="http://tempuri.org/">
      <PdAName>string</PdAName>
    </GetDocsByPdA>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <GetDocsByPdAResponse xmlns="http://tempuri.org/">
      <GetDocsByPdAResult>
        <string>string</string>
        <string>string</string>
      </GetDocsByPdAResult>
    </GetDocsByPdAResponse>
  </soap12:Body>
</soap12:Envelope>
```

10. GetIndexDescriptions

Questo metodo

Definizione:

```
string[] GetIndexDescriptions(string Company, string DocType);
```

Input:

Tipo	Descrizione
<code>string Company</code>	Azienda a cui fa capo il documento
<code>string DocType</code>	Tipo del documento

Output:

Tipo	Descrizione
<code>String[]</code>	Elenco di stringhe che contengono in ordine le descrizioni degli indici dei documenti facenti parte della tipologia documentale specificata in input

Esempio di utilizzo GetIndexDescriptions:

```
string[] descriptions;
descriptions = KAWS.GetIndexDescriptions("Krill", "Fattura_Emessa");
string ElencoDescrizioni = "";
if (descriptions != null)
    foreach (string s in descriptions) ElencoDescrizioni += s + "\n";
```


Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <GetIndexDescriptions xmlns="http://tempuri.org/">
      <Company>string</Company>
      <DocType>string</DocType>
    </GetIndexDescriptions>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <GetIndexDescriptionsResponse xmlns="http://tempuri.org/">
      <GetIndexDescriptionsResult>
        <string>string</string>
        <string>string</string>
      </GetIndexDescriptionsResult>
    </GetIndexDescriptionsResponse>
  </soap12:Body>
</soap12:Envelope>
```

11. GetOriginalDocFileLink

Questo metodo restituisce il link web al documento richiamato dai parametri di input. Questo Link sarà valido fino ad una nuova richiesta di documento, momento in cui verrà cancellato dal deposito temporaneo e non più disponibile fino ad un suo successivo richiamo. Restituisce il documento originale, così come è stato caricato.

Definizione:

```
string GetOriginalDocFileLink(string PdAName, string Company, string
DocType, string FileName, string PubblicazioneWeb);
```

Input:

Tipo	Descrizione
<code>string PdAName</code>	Nome del PdA di cui fa parte il documento
<code>string Company</code>	Azienda a cui fa capo il documento
<code>string DocType</code>	Tipo del documento
<code>string FileName</code>	Nome del file originale del documento
<code>string PubblicazioneWeb</code>	Area di pubblicazione web

Output:

Tipo	Descrizione
<code>String</code>	Link al documento http(s) o errore relativo

Esempio di utilizzo GetOriginalDocFileLink:

```
KAWS.GetOriginalDocFileLink("PdAName", "KRILL", "Fattura_Emessa",
"FileName.pdf", "PubblicazioneWeb");
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <GetOriginalDocFileLink xmlns="http://tempuri.org/">
      <PdAName>string</PdAName>
      <Company>string</Company>
      <DocType>string</DocType>
      <FileName>string</FileName>
      <PubblicazioneWeb>string</PubblicazioneWeb>
    </GetOriginalDocFileLink>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <GetOriginalDocFileLinkResponse xmlns="http://tempuri.org/">
      <GetOriginalDocFileLinkResult>string</GetOriginalDocFileLinkResult>
    </GetOriginalDocFileLinkResponse>
  </soap12:Body>
</soap12:Envelope>
```

12. GetPdABByCompanyDocType

Questo metodo restituisce l'elenco dei PdA che fanno capo ai parametri di input passati quali l'azienda di appartenenza e il tipo di documento

Definizione:

```
string[] GetPdABByCompanyDocType(string Company, string DocType);
```

Input:

Tipo	Descrizione
<code>string Company</code>	Azienda a cui fa capo il documento
<code>string DocType</code>	Tipo del documento

Output:

Tipo	Descrizione
<code>string[]</code>	Elenco dei PdA che fanno capo all'azienda e al tipo documento selezionato dai parametri di input

Esempio di utilizzo GetPdABByCompanyDocType:

```
string[] PdA = KAWS.GetPdABByCompanyDocType("KrillSrl", "Fatture");
string elencoPdA = "";
if(PdA != null)
    foreach (string s in PdA) ElencoPdA += s + "\n";
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <GetPdAByCompanyDocType xmlns="http://tempuri.org/">
      <Company>string</Company>
      <DocType>string</DocType>
    </GetPdAByCompanyDocType>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <GetPdAByCompanyDocTypeResponse xmlns="http://tempuri.org/">
      <GetPdAByCompanyDocTypeResult>
        <string>string</string>
        <string>string</string>
      </GetPdAByCompanyDocTypeResult>
    </GetPdAByCompanyDocTypeResponse>
  </soap12:Body>
</soap12:Envelope>
```

13. GetWaitingDocFileLink

Questo metodo restituisce il link al file in attesa di conservazione in base ai parametri di input inseriti. Questo Link sarà valido fino ad una nuova richiesta di documento, momento in cui verrà cancellato dal deposito temporaneo e non più disponibile fino ad un suo successivo richiamo. Propedeutico al suo utilizzo è il metodo *GetWaitingDocs*.

Definizione:

```
string GetWaitingDocFileLink(string FileName);
```

Input:

Tipo	Descrizione
<code>string FileName</code>	Nome del file da richiamare

Output:

Tipo	Descrizione
<code>string</code>	Link al documento http(s) o errore relativo

Esempio di utilizzo GetWaitingDocFileLink:

```
string link = KAWS.GetWaitingDocFileLink("FileName.pdf");
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <GetWaitingDocFileLink xmlns="http://tempuri.org/">
      <FileName>string</FileName>
    </GetWaitingDocFileLink>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <GetWaitingDocFileLinkResponse xmlns="http://tempuri.org/">
      <GetWaitingDocFileLinkResult>string</GetWaitingDocFileLinkResult>
    </GetWaitingDocFileLinkResponse>
  </soap12:Body>
</soap12:Envelope>
```

14. GetWaitingDocs

Questo metodo restituisce l'elenco dei documenti che sono in attesa di conservazione per l'utente autenticato.

Definizione:

```
string[] GetWaitingDocs();
```

Input:

Tipo	Descrizione

Output:

Tipo	Descrizione
<code>String[]</code>	<p>Se l'operazione si è conclusa con successo verrà riportato l'elenco dei documenti in attesa di conservazione.</p> <p>Per ogni stringa presente nell'array saranno indicate le seguenti informazioni in base al prefisso:</p> <ul style="list-style-type: none"> • ID=Identificativo; • FN=FileName; • CP=Company; • DT=DocType <p>Se l'elenco è vuoto o c'è stato un errore viene restituito null</p>

Esempio di utilizzo GetWaitingDocs:

```
string[] Docs = KAWS.GetWaitingDocs();
if (Docs != null)
    foreach (string s in Docs)
    {
        string Identificativo = s.Substring(s.IndexOf("ID=") + 3,
s.IndexOf(";FN=") - s.IndexOf("ID=") - 3);
        string FileName = s.Substring(s.IndexOf("FN=") + 3,
s.IndexOf(";CP=") - s.IndexOf("FN=") - 3);
        string Company = s.Substring(s.IndexOf("CP=") + 3,
s.IndexOf(";DT=") - s.IndexOf("CP=") - 3);
        string DocType = s.Substring(s.IndexOf("DT=") + 3);
        //Ora usa i parametri restituiti in base all'esigenza
    }
```


Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <GetWaitingDocs xmlns="http://tempuri.org/" />
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <GetWaitingDocsResponse xmlns="http://tempuri.org/">
      <GetWaitingDocsResult>
        <string>string</string>
        <string>string</string>
      </GetWaitingDocsResult>
    </GetWaitingDocsResponse>
  </soap12:Body>
</soap12:Envelope>
```

15. IsAlive

Questo metodo serve per sapere se il web service è in ascolto o meno.

Definizione:

```
string IsAlive();
```

Input:

Tipo	Descrizione

Output:

Tipo	Descrizione
String	Stringa con messaggio di benvenuto con il nome utente connesso oppure un messaggio che richiede di autenticarsi.

Esempio di utilizzo IsAlive:

```
string alive = KAWS.IsAlive();
MessageBox.Show(alive);
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <IsAlive xmlns="http://tempuri.org/" />
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <IsAliveResponse xmlns="http://tempuri.org/">
      <IsAliveResult>string</IsAliveResult>
    </IsAliveResponse>
  </soap12:Body>
</soap12:Envelope>
```

16. ResetUserPassword

Questo metodo è utilizzato per eseguire il reset della password dell'utente passato in input. E' inviata una email all'utente con la nuova password di accesso.

Definizione:

```
string ResetUserPassword(string userId);
```

Input:

Tipo	Descrizione
<code>string</code> <code>userId</code>	UserId dell'utente di cui si vuole resettare la password

Output:

Tipo	Descrizione
<code>String</code>	Se l'operazione si è conclusa con successo la stringa riporterà : "OK" Altrimenti verrà riportato l'errore relativo

Esempio di utilizzo ResetUserPassword:

```
string email = "mio.account@gmail.com";  
string alive = KAWS.ResetUserPassword(email);
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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>
    <ResetUserPassword xmlns="http://tempuri.org/">
      <userId>string</userId>
    </ResetUserPassword>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <ResetUserPasswordResponse xmlns="http://tempuri.org/">
      <ResetUserPasswordResult>string</ResetUserPasswordResult>
    </ResetUserPasswordResponse>
  </soap12:Body>
</soap12:Envelope>
```

17. SearchDocuments

Questo metodo restituisce l'elenco dei documenti che corrispondono ai parametri di ricerca che sono stati impostati in input. Il numero di indici ricercabili contemporaneamente sono 3 e la ricerca è in "and".

Definizione:

```
string[] SearchDocuments(string Company, string DocType, string DocId,
string SelectTypeDocId, int IndexNumber1, string IndexValue1, string
SelectType1, int IndexNumber2, string IndexValue2, string SelectType2,
int IndexNumber3, string IndexValue3, string SelectType3, int
DocNumberLimit);
```

Input:

Tipo	Descrizione
<code>string Company</code>	Azienda a cui fanno capo i documenti da ricercare
<code>string DocType</code>	Tipo del documento
<code>string DocId</code>	Identificativo del documento
<code>string SelectTypeDocId</code>	Se pari a "=" saranno cercati tutti i documenti che hanno DocId esattamente uguale a quello specificato, altrimenti verranno cercati i documenti che hanno come DocId una stringa che contiene il DocId specificato
<code>int IndexNumber1</code>	Numero dell'indice da valorizzare per la ricerca
<code>string IndexValue1</code>	Valore dell'indice da cercare specificato
<code>string SelectType1</code>	Se pari a "=" saranno cercati tutti i documenti che hanno IndexValue esattamente uguale a quello specificato, altrimenti verranno cercati i documenti che hanno come IndexValue una stringa che contiene il DocId specificato
<code>int IndexNumber2</code>	Numero dell'indice da valorizzare per la ricerca
<code>string IndexValue2</code>	Valore dell'indice da cercare specificato
<code>string SelectType2</code>	Se pari a "=" saranno cercati tutti i documenti che hanno IndexValue esattamente uguale a quello specificato, altrimenti verranno cercati i documenti che hanno come IndexValue una stringa che contiene il DocId specificato
<code>int IndexNumber3</code>	Numero dell'indice da valorizzare per la ricerca
<code>string IndexValue3</code>	Valore dell'indice da cercare specificato
<code>string SelectType3</code>	Se pari a "=" saranno cercati tutti i documenti che hanno IndexValue esattamente uguale a quello specificato, altrimenti verranno cercati i documenti che hanno come IndexValue una stringa che contiene il DocId specificato
<code>int DocNumberLimit</code>	Limita la ricerca ai primi X documenti specificati da questo parametro

Output:

Tipo	Descrizione
String[]	<p>Se l'operazione si è conclusa con successo verrà riportato l'elenco dei documenti che corrispondono alla ricerca impostata. Per ogni stringa presente nell'array saranno indicate le seguenti informazioni in base al prefisso:</p> <ul style="list-style-type: none"> • ID=Identificativo; • FN=FileName; • V=Versione; • PW=PublicazioneWeb; • PdA=Pacchetto di archiviazione di appartenenza <p>Se l'elenco è vuoto o c'è stato un errore viene restituito null</p>

Esempio di utilizzo SearchDocuments:

```

string[] Docs = KAWS.SearchDocuments("Krill","Fattura_Emessa",
"DocId","=",1,"2013","=",2,"","=",3,"","=",50);
string toshow = "";
if (Docs != null)
    foreach (string s in Docs)
    {
        string Identificativo = s.Substring(s.IndexOf("ID=") + 3,
s.IndexOf(";FN=") - s.IndexOf("ID=") - 3);
        string FileName = s.Substring(s.IndexOf("FN=") + 3,
s.IndexOf(";V=") - s.IndexOf("FN=") - 3);
        string Versione = s.Substring(s.IndexOf("V=") + 2,
s.IndexOf(";PW=") - s.IndexOf("V=") - 2);
        string PubblicazioneWeb = s.Substring(s.IndexOf("PW=") +
3,s.IndexOf(";PdA=") - s.IndexOf("PW=") - 3);
        string PdA = s.Substring(s.IndexOf("PdA=") + 4);
        toshow += s + "\n";
    }
else toshow = "Authenticate first";
MessageBox.Show(toshow);

```

Richiesta Soap 1.2

```

POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <SearchDocuments xmlns="http://tempuri.org/">
      <Company>string</Company>
      <DocType>string</DocType>
      <DocId>string</DocId>
      <SelectTypeDocId>string</SelectTypeDocId>
      <IndexNumber1>int</IndexNumber1>
      <IndexValue1>string</IndexValue1>
      <SelectType1>string</SelectType1>
      <IndexNumber2>int</IndexNumber2>
      <IndexValue2>string</IndexValue2>
      <SelectType2>string</SelectType2>
      <IndexNumber3>int</IndexNumber3>
      <IndexValue3>string</IndexValue3>
      <SelectType3>string</SelectType3>
      <DocNumberLimit>int</DocNumberLimit>
    </SearchDocuments>
  </soap12:Body>
</soap12:Envelope>

```


Risposta Soap 1.2

```
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>
    <SearchDocumentsResponse xmlns="http://tempuri.org/">
      <SearchDocumentsResult>
        <string>string</string>
        <string>string</string>
      </SearchDocumentsResult>
    </SearchDocumentsResponse>
  </soap12:Body>
</soap12:Envelope>
```

18. UserExist

Questo metodo serve per sapere l'utente specificato in input è presente nel sistema oppure no.

Definizione:

```
bool UserExist(string UserName);
```

Input:

Tipo	Descrizione
<code>string</code> UserName	UserId dell'utente di cui si vuole sapere l'esistenza

Output:

Tipo	Descrizione
<code>bool</code>	True se l'utente esiste False se non esiste

Esempio di utilizzo UserExist:

```
bool UserExist = KAWS.UserExist();
```

Richiesta Soap 1.2

```
POST /WebService/KloudarchiveWebService.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:Header>
    <SecuredWebServiceHeader xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
      <AuthenticatedToken>string</AuthenticatedToken>
    </SecuredWebServiceHeader>
  </soap12:Header>
  <soap12:Body>
    <UserExist xmlns="http://tempuri.org/">
      <UserName>string</UserName>
    </UserExist>
  </soap12:Body>
</soap12:Envelope>
```

Risposta Soap 1.2

```
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>
    <UserExistResponse xmlns="http://tempuri.org/">
      <UserExistResult>boolean</UserExistResult>
    </UserExistResponse>
  </soap12:Body>
</soap12:Envelope>
```