Globlacom CAP web service

Manuale Tecnico

Revisione: 4.8
Data ultima modifica: 07/06/2019

Sommario

Sommario

1 Manuale tecnico GBCCap
- 1.1 Scopi e campi d'azione
- 1.2 Login e accountistica di Globalcom
  - Esempio 1.2.1
- 1.3 Url web service di Globalcom web

2 Lista dei metodi
- 2.1 Metodo Login
  - Esempio 2.1.1
- 2.2 Metodo SessionCheck
  - Esempio 2.2.1
- 2.3 Metodo CittaDaCap
  - Esempi 2.3.1
- 2.4 Metodo DUG
  - Esempi 2.4.1
- 2.5 Metodo DUGDaCAP
  - Esempio 2.5.1
- 2.6 Metodo DUGDaCitta
  - Esempio 2.6.1
- 2.7 Metodo ListaCAP
  - Esempio 2.7.1
- 2.8 Metodo ListaCAPDaCitta
  - Esempio 2.8.1
- 2.9 Metodo ListaCAPDaCittaEstesa
  - Esempio 2.9.1
- 2.10 Metodo ListaCAPDaVia
  - Esempio 2.10.1
- 2.11 Metodo ListaCAPDaViaEstesa
  - Esempio 2.11.1
- 2.12 Metodo ListaCitta
  - Esempio 2.12.1
- 2.13 Metodo ListaComuniDaProvincia

Esempio 2.13.1
2.14 Metodo ListaFrazioniDaCAP
Esempio 2.14.1
2.15 Metodo ListaFrazioniDaCittaCAP
Esempio 2.15.1
2.16 Metodo ListaProvince
Esempio 2.16.1
2.17 Metodo ListaProvinceDaRegione
Esempio 2.17.1
2.18 Metodo ListaProvinceDaRegioneEstesa
Esempio 2.18.1
2.19 Metodo ListaProvinceEstesa
Esempio 2.19.1
2.20 Metodo ListaRegioni
Esempio 2.20.1
2.21 Metodo ListaRegioniEstesa
Esempio 2.21.1
2.22 Metodo ListaVieDaCAP
Esempio 2.22.1
2.23 Metodo ListaVieDaCitta
Esempi 2.23.1
2.24 Metodo VerificaPartitaIVA
Esempio 2.24.1

3 Classi di dati
3.1 contexttype

4 Altri esempi
4.1 Session ID
Esempio C#
Esempio Android

5 Modifiche al documento rispetto alla versione precedente
5.1 Versione 4.8 del 07/06/2019
Dettaglio cambiamenti

1 Manuale tecnico GBCCap

1.1 Scopi e campi d'azione

Scopo di questo documento è illustrare le funzionalità remote esposte dall'interfaccia web di Globalcom, sotto forma di web service di tipo SOAP, dedicata alle informazioni geografiche volte alla spedizione postale.
Le funzionalità esposte sono legate all'invio di prodotti postali tramite Poste Italiane, e servono a consultare, verificare ed esporre vie, cap, città, province e regioni Italiane, ed eventualmente agli stati esteri supportati dall'invio postale.

1.2 Login e accountistica di Globalcom

Per le funzioni che richiedono un login, è necessario comprendere il funzionamento delle utenze di Globalcom.
Quando si definisce uno USER del sistema Globalcom si deve tenere presente che lo stesso può essere associato o meno ad un "gruppo" specifico. Se non viene associato ad un gruppo specifico, il sistema associa automaticamente lo USER al gruppo predefinito <DEFAULT>, che equivale a NESSUN GRUPPO.
Lo stesso USER può essere associato anche ad un altro GRUPPO specifico (volendo anche più di uno).

Quando lo USER vuole effettuare l'accesso a Globalcom può farlo o come utente spare (cioè GRUPPO DEFAULT) oppure come utente di un gruppo specifico.

A seconda del tipo di accesso effettuato lo stesso USER interagirà diversamente con il sistema in funzione dei relativi permessi associati.
Ad esempio, accedendo come utente spare (cioè gruppo DEFAULT) potrà visionare solo la messaggistica generata da lui stesso, accedendo come associato ad un gruppo specifico, del quale è stato magari definito "amministratore di gruppo", potrebbe invece visionare il traffico generato dai tutti gli USER di quel gruppo, ecc…

In definitiva quindi lo stesso USER può agire con profili diversi a seconda se si connette come utente spare (senza gruppo), oppure associate ad un gruppo specifico.
Ogni USER risulta quindi riconosciuto al login al sistema Globalcom mediante una combinazione di USERID, PASSWORD e GRUPPO.

Nel sistema Globalcom vengono inoltre configurati uno o più account per i servizi di corrispondenza online di POSTE ITALIANE e, più genericamente, un certo numero di account verso gestori dei vari servizi di comunicazione gestiti da Globalcom stesso (ad es. SMS, FAX, PEC, ecc.).

Ogni USER Globalcom può quindi essere associato ad uno degli account configurati nel sistema per l'accesso ai servizi offerti dai vari gestori. Anche in questo caso, l'associazione

può variare in funzione del GRUPPO. Si intende cioè che a seconda del tipo di accesso effettuato dallo USER (cioè come utente spare o associato ad un gruppo) lo stesso USER potrebbe risultare associato anche a gestori dei servizi differenti o allo stesso gestore ma con differente account.
A titolo di esempio è possibile definire uno USER, appartenente alla segreteria comune di uno studio professionale ove operano 3 professionisti, associandolo a 3 gruppi, ciascuno riferito al rispettivo professionista GRUUPI A, B e C). Quando lo USER effettuerà l'accesso con il gruppo del professionista A (gruppo A) la corrispondenza generata sarà inviata con l'account del gestore associato al GRUPPO A, e così via per gli altri gruppi. In questo modo il traffico sarà fatturato automaticamente dal gestore direttamente ai 3 professionisti senza richiedere una reciproca rifatturazione delle spese.

Vista in un'ottica più allargata, il medesimo meccanismo viene utilizzato quando il sistema Globalcom viene condiviso da più società appartenenti al uno stesso gruppo consentendo quindi una riduzione della complessità informatica e delle spese di gestione (centralizzate per l'intero gruppo) mantenendo però del tutto separate le fatturazioni da parte dei gestori alle singole società del gruppo stesso.

Superadmin
E' possibile definire uno o più USER con i permessi di "Super Admin", come lo USER pre-definito SYSDBA. Uno USER "Super Admin" non è associabile ad un gruppo e può operare qualsiasi modifica a tutti gli altri account, nonché interrogare supervisionare tutto il traffico transitato nel sistema ed gestire tutti i servizi offerti dai gestori ai quali è associato.

Esempio 1.2.1
Prendendo in esame un ipotetico utente "DEMO", possiamo inserire questo come utente spare (OSSIA GRUPPO default) ad esempio con i permessi per inviare solo Raccomandate e Lettere, utilizzando l'account POSTE1 verso i servizi di corrispondenza online di Poste Italiane.
Allo stesso modo possiamo poi associare l'utente DEMO al gruppo "AMMINISTRAZIONE PERSONALE" (al quale è associato l'account POSTE2 per i servizi di corrispondenza online).
Quando l'user DEMO effettuerà l'accesso specificando anche il Gruppo, la corrispondenza generata nella sessione sarà spedita a POSTE con l'account POSTE2, ossia con un account differente da POSTE1, al quale è invece stato associato come utente "spare" .
Analogamente potranno essere abilitati o meno anche gli altri servizi (FAX, SMS, ecc) che risulteranno pertanto attivi in base alla configurazione operata.
L'utente potrà avere password diverse per l'accesso come spare e come gruppo, quindi non è detto che DEMO spare debba necessariamente essere utilizzato dalla stessa persona che utilizza l'utente DEMO/AMMINISTRAZIONE PERSONALE.
Altresì, in un altro gruppo, lo stesso utente DEMO potrebbe risultare capogruppo, e controllare anche gli invii degli altri membri del gruppo, e così via.

1.3 Url web service di Globalcom web

Le webservice contenute nel servizio GBCWeb sono presenti in ogni installazione di Globalcom. Le url per raggiungerle sono costruite in questo modo:

- Servizio CAP: **http://%INDIRIZZO-VOSTRO-SERVER-GLOBALCOM%/gbcweb/gbccap.asmx?wsdl**
- Informazioni generali sulle web service Globalcom: https://servizipostaliglobalcom.it/gbcweb/gbcwebservice.asmx

Nota bene: in questi indirizzi, **%INDIRIZZO-VOSTRO-SERVER-GLOBALCOM%** si riferisce all'indirizzo del server **Globalcom** a cui si sta facendo riferimento.
Quindi, se il vostro server ha come indirizzo IP 192.168.1.1, la url della web service principale sarà:

http://192.168.1.1/gbcweb/gbccap.asmx?wsdl

Questo dipende quindi dall'installazione Globalcom presente nella **vostra** rete aziendale.

2 Lista dei metodi

La maggior parte dei metodi di questa web service hanno un parametro *prefixText* e possono avere uno o più parametri *contextKey*, che indicano parametri di ricerca.
Nota 1: Va specificato che per tutti i metodi della GBCCap in cui non sia un valore numerico, sia *prefixText* che *contextKey (qualsiasi numero ce ne sia)* vanno considerati **case insensitive**, quindi è indifferente indicare "ROMA" o "roma": restituirà comunque gli stessi risultati.
Nota 2: Questa web service non è sempre disponibile nell'installazione di Globalcom a cui si fa riferimento, quindi non risponderà sempre per qualsiasi installazione. Questo perchè si tratta di un pacchetto applicativo che non è sempre presente nell'installazione, ma viene distribuito come estensione al sistema.

2.1 Metodo Login

Parametri in

- user (string)
- password (string)
- gruppo (string)

Parametri Out

- message (string)

Risultato

bool

Descrizione
Metodo utilizzato per avviare la sessione iniziale con la web service. I parametri in ingresso devono corrispondere ad un'utenza di globalcom. In caso di utilizzo di un'utenza "spare", ovvero non legata ad un gruppo specifico, come gruppo va inserito "<DEFAULT>".
Per poter usufruire della sessione nei metodi successivi, è necessario abilitare un cookie, o recuperare l'id di sessione da *message*, per poi riproporlo nell'header http delle successive chiamate con il tag "ASP.NET_SessionId".
La funzione restituisce true se il login ha avuto successo, false in caso di errori o se user, password e gruppo non corrispondono ad un utenza valida.

Esempio 2.1.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   result = true;
}
```

2.2 Metodo SessionCheck

Parametri in
Parametri out
- errormessage (string)

Risultato
- bool

Descrizione
Questo metodo permette unicamente di controllare se la sessione di riferimento (passata tramite sessionid o tramite cookies) è ancora attiva, senza richiamare un metodo che restituisca risultati più corposi.
Se la funzione restituisce true, si può richiamare altri metodi direttamente, altrimenti va effettuato un nuovo login.

Esempio 2.2.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (!localgbc.SessionCheck(out message))
{
   //Altrimenti riavvio l'interfaccia proxy e rifaccio il login.
   logline("[" + funzione + "] Sessione scaduta, reinizializzazione...");
   if (localgbc.Login(user, password, group, out message))
   {
       result = true;
   }
}
```

2.3 Metodo CittaDaCap

Parametri in
- prefixText (string)
- contextKey (string)

Risultato
- string[]

Descrizione
Restituisce una lista di città che corrispondono al CAP passato in *contextKey*. Se *prefixText* viene valorizzato, la lista sarà ulteriormente filtrata in base alle città che iniziano per *prefixText.*
La lista viene restituita in uscita come vettore di stringhe. Se nessuna città nel cappario di Globalcom corrisponde alla combinazione di CAP e prefisso, il metodo restituirà una lista vuota.

Esempi 2.3.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string[] result = localgbc.CittaDaCap("", "00169");
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.4 Metodo DUG

Parametri in

- prefixText (string)

Risultato

- string[]

Descrizione

Restituisce una lista dei DUG contenuti nel viario di Globalcom. I DUG sono le tipologie di nome che possono avere i vari indirizzi.
Alcuni esempi (il DUG è quello in grassetto):

- **viale** mazzoni 10.
- **corso** vittorio emanuele II
- **piazza** delle gardenie

Se *prefixText* viene valorizzato, la lista viene filtrata per i DUG che iniziano con quelle lettere.

Esempi 2.4.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string[] result = localgbc.DUG("VI");
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.5 Metodo DUGDaCAP

Parametri in

- prefixText (string)
- contextKey (string)

Risultato

- string[]

Descrizione

Il funzionamento di questo metodo è molto simile a quello di DUG (2.3.4), con la differenza che qui si può specificare anche una *contextKey*, contenente un CAP.
La lista restituirà dunque solo i DUG che vengono utilizzati dagli indirizzi compresi in quel CAP.

Esempio 2.5.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
```

```
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string[] result = localgbc.DUGDaCAP("VI", "00166");
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.6 Metodo DUGDaCitta

Parametri in

- prefixText (string)
- contextKey (string)

Risultato

- string[]

Descrizione

Terzo e ultimo metodo relativo ai DUG, funziona in maniera molto simile a DUGDaCAP (2.3.6), con la sola differenza che il parametro *contextKey* non indica un CAP in base al quale filtrare i DUG, ma una città.
Quindi restituirà tutti i DUG utilizzati entro quella città, con un ulteriore filtro in base al *prefixText*.

Esempio 2.6.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
```

```
{
   string[] result = localgbc.DUGDaCitta("VI", "genova");
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.7 Metodo ListaCAP

Parametri in

- prefixText (string)

Risultato

- string[]

Descrizione

Questo metodo restituisce una lista dei CAP contenuti nel cappario di Globalcom, in un singolo vettore di stringhe.
La lista viene ulteriormente filtrata se *prefixText* viene valorizzato, riportando solo i CAP che iniziano con le cifre contenute nel parametro.
I CAP sono tutti composti da un numero di cinque cifre.

Esempio 2.7.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string[] result = localgbc.ListaCAP("001");
   if (result != null)
   {
```

```
        StringBuilder totallines = new StringBuilder();
        totallines.AppendLine("------------------------");
        foreach (string currline in result)
        {
            totallines.AppendLine(currline);
        }
        totallines.AppendLine("------------------------");

        e.Result = totallines.ToString();
    }
}
```

2.8 Metodo ListaCAPDaCitta

Parametri in

- prefixText (string)
- contextKey (string)
- tipoRicerca (contexttype)

Risultato

- string[]

Descrizione

Anche questo metodo restituisce una lista di CAP contenuti nel cappario di Globalcom, ma in questo caso solo quelli pertinenti alla città indicata, in base ai parametri *contextKey* e *tipoRicerca*.
contextKey conterrà quindi un nome di città, o parte del nome di una città, dipendentemente da come viene passato *tipoRicerca*.
Quindi se viene passato come *contextKey* "ROMA", e come tipoRicerca "ctPrefix", la lista risultante conterrà i CAP di tutte le località che cominciano con "ROMA", quindi "ROMA", "ROMANO CANAVESE", "ROMAGNANO SESIA" e così via.
La lista è comunque filtrata anche per il prefisso sull'inizio dei CAP, attraverso il parametro *prefixText*.

Esempio 2.8.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
```

```
if (localgbc.Login(user, password, group, out message))
{
   string[] result = localgbc.ListaCAPDaCitta("", "Ven", contexttype.ctPrefix);
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.9 Metodo ListaCAPDaCittaEstesa

Parametri in

- prefixText (string)
- contextKey (string)
- tipoRicerca (contexttype)

Risultato

- string[][]

Descrizione

Questo metodo funziona esattamente come ListaCAPDaCitta, con la sola differenza nel tipo di dati ritornati: Invece di una lista di soli CAP, restituisce una matrice bidimensionale con quattro campi:

- CAP
- CITTA
- PROVINCIA
- PROVINCIAESTESA

Dove PROVINCIAESTESA rappresenta la denominazione allargata della PROVINCIA, quindi dove PROVINCIA è “TO”, PROVINCIAESTESA sarà “TORINO”, dove PROVINCIA è “NO”, PROVINCIAESTESA sarà “NOVARA” e così via.
I parametri in entrata vengono trattati esattamente come nel metodo precedente (vedere 2.3.8). La lista in uscita non comprende i nomi dei campi, quindi nella linea 0 ci saranno già dei dati effettivi.

Esempio 2.9.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string[][] result = localgbc.ListaCAPDaCittaEstesa("", "Ven",
contexttype.ctPrefix);
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string[] currline in result)
       {
           foreach (string currcol in currline)
           {
               totallines.Append(currcol + ", ");
           }

           totallines.Remove(totallines.Length - 2, 2);//rimuovo la virgola e lo
spazio finali
           totallines.Append(Environment.NewLine);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.10 Metodo ListaCAPDaVia

Parametri in

- prefixText (string)
- contextKey (string)
- contextKey2 (string)
- contextKey3 (string)
- tipoRicerca (contexttype)

Risultato

- string[]

Descrizione

Anche questo metodo restituisce una lista di cap (mono-colonna), ma mentre il prefisso si comporta sempre allo stesso modo, le *contextKey* permettono un livello di dettaglio nel filtro più complesso.
Il campo *contextKey* è sempre la città per la quale cercare i CAP, ma in questo caso è sempre trattata come ctExact. Comunque se lasciato vuoto non sarà considerato, e i CAP verranno estratti indipendentemente dalla città di pertinenza. Il campo *contextKey2* invece indica la via per la quale trovare dei CAP, mentre *contextKey3* è il civico (molte vie maggiori passano da diverse zone e CAP, quindi il civico cambia i CAP di pertinenza).
Anche *contextKey3* è opzionale e, se valorizzata, deve contenere un numero civico: i CAP restituiti in quel caso verranno filtrati in base ai gruppi di civici pertinenti ogni via recuperata, e quindi solo quelli del gruppo di civici a cui appartiene quello indicato verranno estratti.
Anche *contextKey2* è opzionale, ma nel caso venga valorizzata ha effetto il parametro *tipoRicerca* sul come questo valore verrà trattato, sempre in ottemperanza alle specifiche della classe **contexttype** (vedere 3.10).
Se *tipoRicerca* viene passato come ctExact, il DUG deve essere compreso nella via, quindi "via giuseppe mazzini" troverà dei valori sia come ctPartial, che come ctPrefix, che come ctExact, mentre "giuseppe mazzini" solo come ctPartial.

Esempio 2.10.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
 string[] result = localgbc.ListaCAPDaVia("", "genova", "san ", "6",
contexttype.ctPrefix);
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");
```

```
        e.Result = totallines.ToString();
    }
}
```

2.11 Metodo ListaCAPDaViaEstesa

Parametri in

- prefixText (string)
- contextKey (string)
- contextKey2 (string)
- contextKey3 (string)
- tipoRicerca (contexttype)

Risultato

- string[][]

Descrizione

In maniera analoga a **ListaCAPDaCittaEstesa** (vedere 2.3.9), questo metodo funziona come **ListaCAPDaVia**, con l'unica differenza che il ritorno non è solo una lista di CAP ma una matrice bidimensionale con questi campi:

- VIA
- CIVICODA
- CIVICOA
- CAP

Anche in questo caso i campi non sono presenti nella lista, quindi la prima riga è di dati effettivi.

Esempio 2.11.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string[][] result = localgbc.ListaCAPDaViaEstesa("", "genova", "san ", "6",
contexttype.ctPrefix);
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
```

```
        foreach (string[] currline in result)
        {
            foreach (string currcol in currline)
            {
                totallines.Append(currcol + ", ");
            }

            totallines.Remove(totallines.Length - 2, 2);//rimuovo la virgola e lo
spazio finali
            totallines.Append(Environment.NewLine);
        }
        totallines.AppendLine("------------------------");

        e.Result = totallines.ToString();
    }
}
```

2.12 Metodo ListaCitta

Parametri in

- prefixText (string)

Risultato

- string[]

Descrizione

Restituisce una lista di tutte le città (ma non le frazioni) presenti nel viario di Globalcom. Come al solito *prefixText* indica il prefisso della città da cercare, quindi l'inizio del nome. Il risultato è una singola colonna delle città trovate.

Esempio 2.12.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string result = localgbc.ListaCitta("POL");
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
```

```
        {
            totallines.AppendLine(currline);
        }
        totallines.AppendLine("------------------------");

        e.Result = totallines.ToString();
    }
}
```

2.13 Metodo ListaComuniDaProvincia

Parametri in

- prefixText (string)
- contextKey (string)

Risultato

- string[]

Descrizione

Restituisce una lista di città o frazioni dal viario, in base alla provincia in *contextKey*. Se la provincia viene passata vuota, il metodo restituirà tutte le città e frazioni nel viario. Se viene valorizzata, è trattata come ricerca esatta, e corrisponde al campo della provincia breve.
Quindi per ottenere tutte le città e frazioni della provincia di Roma, va passato “RM” come *contextKey*.

Esempio 2.13.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string result = localgbc.ListaComuniDaProvincia("ROCCA ", "RM");
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
```

```
            totallines.AppendLine(currline);
        }
        totallines.AppendLine("------------------------");

        e.Result = totallines.ToString();
    }
}
```

2.14 Metodo ListaFrazioniDaCAP

Parametri in

- prefixText (string)
- contextKey (string)

Risultato

- string[]

Descrizione

Questo metodo restituisce tutte le frazioni (non le città principali quindi) contenute nel viario di Globalcom, relative ad un determinato CAP. Se *contextKey* viene lasciato vuoto il metodo restituirà tutte le frazioni contenute nel viario.
Come al solito il primo campo indica un ulteriore filtro sull'inizio del nome delle frazioni.

Esempio 2.14.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string result = localgbc.ListaFrazioniDaCAP("V", "02016");
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
```

```
        }
        totallines.AppendLine("------------------------");

        e.Result = totallines.ToString();
    }
}
```

2.15 Metodo ListaFrazioniDaCittaCAP

Parametri in

- prefixText (string)
- contextKey (string)
- contextKey2 (string)

Risultato

- string[]

Descrizione

Questo metodo rappresenta un ulteriore filtro rispetto al precedente, permettendo di ottenere le frazioni appartenenti ad un determinato CAP e città.
Il secondo campo *contextKey2* indica la città di appartenenza delle frazioni che si vuole ottenere, è trattato sempre come ricerca esatta, e come sempre è case insensitive.

Esempio 2.15.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string result = localgbc.ListaFrazioniDaCittaCAP("SA", "", "Cassino");
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
```

```
        }
        totallines.AppendLine("------------------------");

        e.Result = totallines.ToString();
    }
}
```

2.16 Metodo ListaProvince

Risultato

- string[]

Descrizione

Questo metodo restituisce una lista di tutte le provincie contenute nel viario, senza limitazioni. Non ha filtri quindi viene richiamata solo quando serve velocemente la lista completa, come per esempio per riempire una combobox.

Esempio 2.16.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string result = localgbc.ListaProvince();
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.17 Metodo ListaProvinceDaRegione

Parametri in

- contextKey (string)
- tipoRicerca (contexttype)

Risultato
- string[]

Descrizione
Questo metodo restituisce una lista di province, in maniera similare a quello precedente, ma filtrate in base alla regione d'appartenenza. La regione può essere cercata in maniera esatta, come prefisso o parziale, in accordo col campo *tipoRicerca*; Non è presente invece un campo prefisso per il nome provincia, visto che queste sono sempre da due lettere quindi sarebbe superfluo.

Esempio 2.17.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string result = localgbc.ListaProvinceDaRegione("BASILICATA", contexttype.ctExact);
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.18 Metodo ListaProvinceDaRegioneEstesa

Parametri in
- contextKey (string)
- tipoRicerca (contexttype)

Risultato

- string[][]

Descrizione

Anche qui come in altri metodi con "estesi", il comportamento è identico alla versione normale, con l'unica differenza che il risultato è in forma tabellare, ovvero una matrice di stringhe a due dimensioni. Le colonne in questo caso sono due:

- PROVINCIA
- PROVINCIAESTESA

Dove PROVINCIAESTESA è il nome esteso, letterale, della provincia di riferimento. Ad esempio per "TO" sarà "TORINO".

Esempio 2.18.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string[][] result = localgbc.ListaProvinceDaRegioneEstesa("BASILICATA",
contexttype.ctExact);
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string[] currline in result)
       {
           foreach (string currcol in currline)
           {
               totallines.Append(currcol + ", ");
           }

           totallines.Remove(totallines.Length - 2, 2);//rimuovo la virgola e lo
spazio finali
           totallines.Append(Environment.NewLine);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
```

```
}
```

2.19 Metodo ListaProvinceEstesa

Risultato

- string[][]

Descrizione

Questo metodo restituisce gli stessi due campi di **ListaProvinceDaRegioneEstesa**, ma non ha nessun parametro, quindi si limita a restituire tutte le provincie contenute nel viario.

Esempio 2.19.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string[][] result = localgbc.ListaProvinceEstesa();
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string[] currline in result)
       {
           foreach (string currcol in currline)
           {
               totallines.Append(currcol + ", ");
           }

           totallines.Remove(totallines.Length - 2, 2);//rimuovo la virgola e lo
spazio finali
           totallines.Append(Environment.NewLine);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.20 Metodo ListaRegioni

Risultato

- string[]

Descrizione

Questo metodo restituisce invece tutte le regioni contenute nel viario di Globalcom, senza alcuna restrizione, in una singola lista.

Esempio 2.20.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string result = localgbc.ListaRegioni();
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.21 Metodo ListaRegioniEstesa

Risultato

- string[][]

Descrizione

Come **ListaRegioni**, anche questo metodo restituisce tutte le regioni, ma in forma tabellare, con un primo campo riempito semplicemente con un id progressivo (che parte da 1).

Esempio 2.21.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string[][] result = localgbc.ListaRegioniEstesa();
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string[] currline in result)
       {
           foreach (string currcol in currline)
           {
               totallines.Append(currcol + ", ");
           }

           totallines.Remove(totallines.Length - 2, 2);//rimuovo la virgola e lo
spazio finali
           totallines.Append(Environment.NewLine);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.22 Metodo ListaVieDaCAP

Parametri in

- prefixText (string)
- contextKey (string)
- tipoRicerca (contexttype)

Risultato

- string[]

Descrizione

Questo metodo restituisce una lista di tutte le vie contenute nel viario che corrispondono ai criteri di ricerca passati nei parametri.
prefixText si riferisce all'inizio del nome della via (DUG compreso), mentre nel parametro

contextKey si può passare un CAP di riferimento o parte di esso, in base al settaggio di *tipoRicerca*.

Esempio 2.22.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string result = localgbc.ListaVieDaCAP("PIAZZA", "7412", contexttype.ctPartial);
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.23 Metodo ListaVieDaCitta

Parametri in

- prefixText (string)
- contextKey (string)
- tipoRicerca (contexttype)

Risultato

- string[]

Descrizione

Anche in questo caso viene restituita una lista di vie, con l'unica differenza che invece di un CAP (o parte), viene filtrata la lista in base ad un nome di città (o parte di esso).
Allo stato attuale questo filtro si basa unicamente sulle città, e non sulle frazioni. In ogni caso le vie di una frazione saranno contenute anche nella città di riferimento per quella frazione.

Esempi 2.23.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string result = localgbc.ListaVieDaCitta("PIAZZA", "Napoli",
contexttype.ctPartial);
   if (result != null)
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       foreach (string currline in result)
       {
           totallines.AppendLine(currline);
       }
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

2.24 Metodo VerificaPartitaIVA

Parametri in

- partitaIVA (string)
- paese (string)

Parametri out

- isvalid (bool)
- nome (string)
- indirizzo (string)
- messaggio (string)

Risultato

- bool

Descrizione

Questo metodo differisce nello scopo e nel funzionamento dagli altri della GBCCap, ma viene considerato parte dello stesso pacchetto applicativo.
Serve a verficare la validità di una partita iva, in base ad una richiesta ad una web service di riferimento.
In entrata va passata la partita iva in questione e il paese di riferimento; il metodo

restituirà un risultato positivo se l'operazione è andata a buon fine, e riepirà i parametri isvalid, nome e indirizzo con le informazioni estratte per quella partita iva, se trovata.
Se invece l'operazione fallisce per problemi di rete o errori di sistema, restituirà un risultato negativo, e riempirù *messaggio* con eventuali dettagli sull'errore.
Nota: Un risultato positivo non indica che la partita iva indicata è valida, ma solo che si è ottenuto un risultato dalla web service di riferimento. Va controllato il parametro *isvalid* per sapere se effettivamente la partita iva è valida. Allo stesso modo, se il risultato è negativo non vuol dire che la partita iva non è stata trovata, ma che non è stato possibile ottenere l'informazione per problemi di rete o malfunzionamento nella web service di riferimento.

Esempio 2.24.1

```
GBCCAPInt.GBCCap localgbc
localgbc = new GBCCAPInt.GBCCap();
localgbc.Url = this.currenturl.ToString();
localgbc.Timeout = 120000;
localgbc.CookieContainer = new CookieContainer();
if (localgbc.Login(user, password, group, out message))
{
   string valida;
   string nome;
   string indirizzo;
   string errore;
   if (localgbc.VerificaPartitaIVA("0030339329320", "ITALIA", out valida, out nome,
out indirizzo, out errore))
   {
       StringBuilder totallines = new StringBuilder();
       totallines.AppendLine("------------------------");
       totallines.AppendLine("Validità partita IVA: " + valida.ToString());
       totallines.AppendLine("Intestata a : " + nome);
       totallines.AppendLine("Sede: " + indirizzo);
       totallines.AppendLine("------------------------");

       e.Result = totallines.ToString();
   }
}
```

3 Classi di dati

3.1 contexttype

Questa classe viene utilizzata in alcuni metodi di ricerca (Soprattutto per vie e cap), per

indicare in che modo deve essere interpretata una contextKey, ovvero una chiave di ricerca.
I valori possibili sono:

- ctPartial - *La chiave di ricerca indica un testo ovunque all'interno delle parole da cercare. Questo vuol dire che se la contextKey è "ROSA", i risultati comprenderanno, tra l'altro:*
  *- MASSA**ROSA***
  *- **ROSA**TE*
  *- MONTECCHIA DI C**ROSA**RA*
- ctPrefix - *La chiave di ricerca è un "prefisso", ovvero un testo con cui le parole da cercare iniziano. Ad esempio, cercando "ROSA", otterremo, tra l'altro:*
  *- **ROSA**TE*
  *- **ROSA**RNO*
  *- **ROSA**ZZA*
  Ma non otteremo:
  *- MASSA**ROSA***
  *- MONTECCHIA DI C**ROSA**RA*
- ctExact - *Le parole restituite devono corrispondere esattamente a quella cercata; quindi per "ROMA" otterremo solo i risultati relativi alla parola "ROMA", non otterremo invece ROMANO CANAVESE, o MONTORIO ROMANO, o CASTEL SAN PIETRO ROMANO.*

4 Altri esempi

4.1 Session ID

L'id di sessione restituito dal metodo login della web service può essere utilizzato, in assenza di gestione cookies, riportandolo nelle successive chiamate nell'header della busta SOAP, come spiegato in questo articolo:
http://stackoverflow.com/questions/6381491/how-to-save-and-return-cookies-to-the-web-service

La chiave dell'header è "ASP.NET_SessionId"

Esempio C#

```
///Attraverso i cookies, basta lanciare il metodo CookieContainer della classe proxy
quando viene inizializzata:
gbcweb.CookieContainer();
```

```
///Metodo session id
///Prima di tutto va creata una partial class della classe proxy che aggiunge la
proprietà
///sessionid e la aggiunge agli httpheader
namespace test_ws.GBCWebInt
{
   public partial class GBCWebservice :
System.Web.Services.Protocols.SoapHttpClientProtocol
   {
       public string sessionid = string.Empty;

       protected override System.Net.WebRequest GetWebRequest(Uri uri)
       {
           HttpWebRequest request = (HttpWebRequest)base.GetWebRequest(uri);

           if ((!string.IsNullOrEmpty(sessionid)) && (request.Headers != null))
           {
               //Disabilitata perché al momento non serve.
               request.Headers.Add("Cookie", "ASP.NET_SessionId=" + sessionid);
           }

           return request;
       }
   }
}

///Dopodiché si salva il sessionid nella proprietà all'atto del login:
if (gbcweb.Login(parametri["USERID"], parametri["PASSWORD"], parametri["GRUPPO"], out
message))
{
   gbcweb.sessionid = message;//Questo fà l'equivalente del CookieContainer
}
```

Esempio Android

```
//Salvataggio sessionid:
List headerList = androidHttpTransport.call(SOAP_ACTION, envelope, null);

for (Object header : headerList) {
   HeaderProperty headerProperty = (HeaderProperty) header;
   String headerKey = headerProperty.getKey();
   String headerValue = headerProperty.getValue();

   System.out.println(headerKey +" : " + headerValue);
```

```
   sharedPreferenceEditor.putString(headerKey, headerValue);

}

sharedPreferenceEditor.commit();

//Invio sessionid:
HeaderProperty headerPropertyObj = new HeaderProperty("cookie",
preferences.getString("set-cookie", ""));
headerList.add(headerPropertyObj);
androidHttpTransport.call(SOAP_ACTION, envelope, headerList);
```

5 Modifiche al documento rispetto alla versione precedente

5.1 Versione 4.8 del 07/06/2019

Versione GBCWeb 4.1.0.42. Questa è la prima versione del manuale come documento a se stante, separato dal manuale delle web service principali.
Sono stati eliminati i metodi, classi ed esempi non pertinenti a questa web service.

Dettaglio cambiamenti

Paragrafo	Modifiche
Tutti i metodi del capitolo 2	Modificata la numerazione per riflettere l'attuale condizione di manuale separato.