



Descrizione tecnica del Web Service GisNumeroCivicoService

L'interfacciamento con la Cartografia è ottenuto tramite dei Web Services che, in base al tipo di dato che si desidera utilizzare per la comunicazione, svolgono tale scopo. In questo documento focalizziamo l'attenzione sul WS che permette l'interfacciamento per i dati relativi ai numeri civici.

Il meccanismo di scambio avviene in questo modo:

- L'applicazione da cui parte la fase di interazione invoca un metodo del WS per ogni punto cartografico che desidera prendere in considerazione.
- Invoca un ulteriore metodo che salva questa sequenza di punti restituendo un GUID, un codice globale univoco, che dovrà essere utilizzato sull'URL o su altro canale di comunicazione per la chiamata vera e propria verso l'applicazione di destinazione.
- L'applicazione di destinazione recepisce questa chiamata e, tramite un terzo metodo del WS utilizzando il GUID fornito, ottiene la sequenza di punti cartografici che sono stati richiesti dall'applicazione di partenza.

Il Web Service, chiamato **GisNumeroCivicoService**, è composto da 3 metodi principali.

La sintassi qui esposta è quella C# (il linguaggio in cui è stato scritto il WS), comunque di facile comprensione indipendentemente dal linguaggio utilizzato per accedere al WS stesso.

- Metodo AddPointSelectionExt

```
[WebMethod(EnableSession = true)]  
public void AddPointSelectionExt(string CodiceVia,  
                                string Numero, string Sub, string  
                                Scala);
```

Questo è il primo metodo che viene invocato dall'applicazione da cui parte l'interazione.

CodiceVia: è un intero che identifica il codice via della toponomastica relativa al civico che si vuole vedere selezionato in cartografia. Tale codice è presente nelle banche dati dell'anagrafe relativa all'applicazione da cui parte l'interazione.

Numero: è una stringa che identifica il numero civico vero e proprio.

Sub: è una stringa che identifica il subalterno o interno, come ulteriore dato oltre il numero civico.

Scala: è una stringa che identifica per l'appunto la scala, come ulteriore dato che identifica il numero civico.

E' importante sottolineare che, come si può notare dalla dichiarazione del metodo, si tratta di un metodo che prevede l'abilitazione della sessione (normalmente i WS sono invece sessionless), trattandosi di un metodo che raccoglie le informazioni tra una chiamata e l'altra, mantenendone la persistenza fino alla chiamata del metodo SaveSelection.

- Metodo SaveSelectionExt

```
[WebMethod(EnableSession = true)]  
public string SaveSelectionExt()
```

Salva in modo persistente i dati precedentemente inseriti tramite la chiamata al metodo AddPointSelectionExt, restituendo una stringa che indica il GUID che dovrà successivamente essere utilizzando nella chiamata per invocare l'applicazione di destinazione.

- Metodo GetSelectionExt

```
[WebMethod]  
public IList GetSelectionExt(string Code)
```

Restituisce un array di elementi di selezione in base al **Code** fornito al metodo ovvero il GUID ottenuto da una precedente chiamata al metodo SaveSelection e passato all'applicazione di destinazione, applicazione che è stata chiamata tramite URL o altro meccanismo di attivazione che permetta il passaggio di un parametro.

Ecco il dettaglio di questa struttura:

```
string m_CodiceVia;  
string m_Numero;  
string m_Sub;  
string m_Scala;
```

Come si può notare altro non sono che i valori precedentemente passati nelle chiamate al metodo AddPointSelectionExt effettuate dall'applicazione che inizia la fase di interazione con l'applicazione di destinazione.

Per ulteriore chiarezza riportiamo un esempio di chiamata effettuata dall'applicazione iniziale, l'esempio è in linguaggio C# su tecnologia .NET framework.

```
/// Attiva la gestione della sessione per consentire la  
persistenza fra una chiamata e l'altra  
/// all'interno del medesimo contesto di sessione HTTP  
System.Net.CookieContainer cookieGis = new  
System.Net.CookieContainer();  
OikosComuniNumeroCivico.GisNumeroCivicoService gis = new  
OikosComuniNumeroCivico.GisNumeroCivicoService();  
gis.CookieContainer = cookieGis;  
  
gis.AddPointSelectionExt("", "", "Valore Geocodice");  
string urlCartografia = "http://host/urlcartografia";  
string guid = gis.SaveSelectionExt();  
if (urlCartografia != null && guid != null)  
{  
    finalUrlCartografia = urlCartografia + "&GisGuidNumCiv=" +  
guid;  
    Response.Redirect(finalUrlCartografia);  
}
```



Per richiedere una visualizzazione in mappa di più oggetti contemporaneamente (più numeri civici contemporaneamente) è sufficiente accodare più AddPointSelectionExt successivi.

Per consentire un utilizzo agevole di tale Web Service, anche da piattaforme e linguaggi che prevedono l'abilitazione della sessione, sono state implementate una serie di estensioni ai metodi principali qui sopra descritti.

- **Metodo GetTokenExt**

Metodo che consente la generazione di un token da utilizzare nel resto delle invocazioni ai metodi, per arrivare a passare il parametro GUID all'applicazione cartografica

- **Metodo AddPointSelectionWithTokenExt**

Questo è il primo metodo da invocare dopo aver generato il token tramite il precedente metodo:

Token: è il valore del token ottenuto tramite la chiamata del precedente metodo.

CodiceVia: è un intero che identifica il codice via della toponomastica relativa al civico che si vuole vedere selezionato in cartografia. Tale codice è presente nelle banche dati dell'anagrafe relativa all'applicazione da cui parte l'interazione.

Numero: è una stringa che identifica il numero civico vero e proprio.

Sub: è una stringa che identifica il subalterno o interno, come ulteriore dato oltre il numero civico.

Scala: è una stringa che identifica per l'appunto la scala, come ulteriore dato che identifica il numero civico.

E' importante sottolineare che, come si può notare dalla dichiarazione del metodo, si tratta di un metodo che prevede l'abilitazione della sessione (normalmente i WS sono invece sessionless), trattandosi di un metodo che raccoglie le informazioni tra una chiamata e l'altra, mantenendone la persistenza fino alla chiamata del metodo SaveSelection.

- **Metodo AddMultipleSelectionWithTokenExt**

Metodo simile a quello precedente. In più consente di effettuare la selezione contemporanea di più numeri civici contemporaneamente in mappa

- **Metodo SaveSelectionWithTokenExt**

Metodo simile al metodo SaveSelectionExt descritto sopra

- **Metodo GetSelectionWithTokenExt**

Metodo simile al metodo GetSelectionExt descritto sopra