



Descrizione tecnica del Web Service GisMappale

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 tramite dati particellari (foglio, particella e subalterno) che altro non è che una stringa che identifica l'oggetto cartografico interessato.

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 **GisMappale**, 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 AddPointSelection

```
[WebMethod(EnableSession = true)]  
public void AddPointSelection(string Foglio,  
                             string Particella, string Subalterno);
```

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

Foglio: è il valore del foglio catastale.

Particella: è il valore della particella catastale.

Subalterno: è il valore del subalterno catastale.

Altro: è il valore del nome del livello cartografico di proiezione del dato delle particelle. Di default assume sempre il valore "Catasto-Particelle"

Codicecomune: è il valore del codice catastale del comune

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 SaveSelection

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

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

- Metodo GetSelection

```
[WebMethod]  
public IList GetSelection(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.

La struttura array restituita contiene elementi di tipo **Foglio/Particella/Subalterno/Altro/CodiceComune**.

Ecco il dettaglio di questa struttura:

```
string m_foglio;  
string m_particella;  
string m_subalterno;  
string m_altro;  
string m_codicecomune;
```

Come si può notare altro non sono che i valori precedentemente passati nelle chiamate al metodo AddPointSelection 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();  
OikosComuni.GisMappale gis = new OikosComuniPC.GisMappale();  
gis.CookieContainer = cookieGis;  
  
gis.AddPointSelection("", "", "Valore Geocodice");  
string urlCartografia = "http://host/urlcartografia";  
string guid = gis.SaveSelection();  
if (urlCartografia != null && guid != null)  
{  
    finalUrlCartografia = urlCartografia + "&GisGuidMap=" +  
guid;  
    Response.Redirect(finalUrlCartografia);  
}
```



Per richiedere una visualizzazione in mappa di più oggetti contemporaneamente (più particelle contemporaneamente). E' sufficiente accodare più AddPointSelection 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 GetToken**

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 AddPointSelectionWithToken**

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.

Foglio: è il valore del foglio catastale.

Particella: è il valore della particella catastale.

Subalterno: è il valore del subalterno catastale.

Altro: è il valore del nome del livello cartografico di proiezione del dato delle particelle. Di default assume sempre il valore "Catasto-Particelle"

Codicecomune: è il valore del codice catastale del comune

- **Metodo SaveSelectionWithToken**

Vale quanto detto precedentemente per il metodo SaveSelection

- **Metodo GetSelectionString**

Vale quanto detto precedentemente per il metodo GetSelection, con la differenza che tale metodo restituisce i dati in una stringa.

- **Metodo GetPointsSelectionWithTokenOneTime**

Questo metodo consente l'accodamento di più selezioni singole (per eseguire una multiselezione) tramite l'uso del token ricevuto invocando l'apposito metodo, e tramite l'accodamento dei dati di foglio e particella in una stringa come questa:

```
//StrAll = [FOG];[PAR];[SUB];[ALTO];[CODICECOMUNE]$$$.....
```

- **Metodo ClearSelection e ClearSelectionWithToken**

Questi metodi consentono di cancellare le selezioni attualmente in corso