Globo srl Soluzioni avanzate per il territorio
Viale Europa, 17/B 24048 Treviolo (BG)
☏ +39 035 596911 🖷 +39 035 596999
globo@pec.it info@globogis.it
Reg.Impr. di BG - Cod.Fisc. - P.IVA 02598580161
Capitale sociale € 300.000 interamente versato

Integrazione Validator con WebService esterni

Comunicazione con validazione e/o aggiunta di contenuti tramite webservice nella modulistica. "BP-OR-AP-02 - Integrazione Validator-WebService" del kit di riuso del progetto PON-GOV "SPRINT"

Per conto di: Comune di Trento

Mattia Rossi *Redatto*	Marco Deligios *Verificato e autorizzato*	6/3/2020 *Data d'emissione*	4 *Revisione*
I_Prodotti *Commessa*	Protocollo *Protocollo*	Pagina 1 di 13	

SOMMARIO

0 STORIA DELLE REVISIONI

Rev.	Data	Redatto	Descrizione
0	02/05/2016	Mattia Rossi	Prima emissione
1	4/5/2016	Mattia Rossi	Rimosso la possibilità di inviare valori multipli in un unico parametro del WebService. Modificato la tipologia di dato in entrata dal WebService.
2	10/6/2016	Mattia Rossi	Aggiunta dettagli sui parametri
3	4/8/2016	Mattia Rossi	Aggiunta specifica mappatura su campi multipli
4	6/3/2020	Mattia Rossi	Revisione del documento

1 SCOPO E CAMPO D'APPLICAZIONE

1.1 Scopo

In questo documento si descriverà la modalità di integrazione tra un Web service ed un Modulo HTML che utilizza La libreria JavaScript *Validator* per la verifica dell'inserimento dati.

L'integrazione permette di inserire all'interno di un *Modulo HTML* particolari campi il cui contenuto può essere valorizzato o validato sulla base dei valori inseriti in altri campi del modulo al fine di migliorare la qualità dei dati acquisiti e ridurre la possibilità che siano commessi errori in fase di compilazione.

Il calcolo del il valore atteso per i campi viene eseguito dal Web service, che definisce completamente la logica di business: Validator invoca il Web service passando il valore di alcuni campi di origine e ottiene il valore atteso per i campi di destinazione.

2 TERMINI E DEFINIZIONI

WS WebService su cui operare l'integrazione.
STU Sportello Telematico™, prodotto GLOBO per la compilazione e presentazione delle istanze da parte del cittadino ad un ente.
Modulo HTML la parte documento esposta all'utente che viene compilata.
Validator libreria Javascript che assiste il processo di compilazione dell'utente nel modulo.

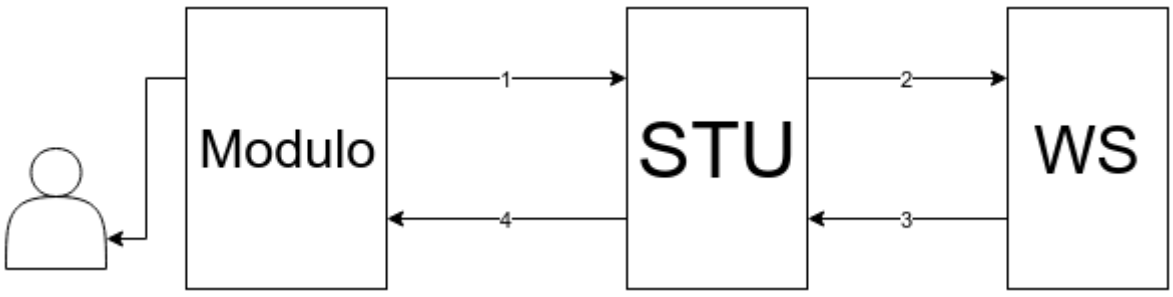
3 CONTENUTI

3.1 Flusso Applicativo

Il processo coinvolge tre componenti fondamentali che concorrono alla realizzazione dell'integrazione tra la parte pubblica esposta come Modulo HTML e la parte di validazione fornita dal WS, in particolare:

- Il Modulo HTML, che include Validator
- Le funzionalità di integrazione di STU
- Il WS da integrare

Queste interagiscono attraverso chiamate HTTP secondo lo schema sotto riportato:



3.1.1 Modulo HTML

Il Modulo HTML si occupa di definire il tipo di WS da richiamare, quali dati devono essere inviati, come trattare i dati di risposta ottenuti e a quale evento scatenare la verifica degli stessi. La definizione di un servizio viene effettuata sfruttando i seguenti parametri inseriti come attributi all'interno del tag HTML appropriato:

- **ws-name**: la chiave identificativa il nome WS da invocare, se questo parametro non è presente la funzionalità di invocazione del WS non verrà applicata nonostante la presenza dei campi associati.
- **ws-param-in**: elenco dei parametri da includere nella richiesta separati da ";" (punto e virgola), risulta essere così costituito:

  <parametro1>=<sorgente1>;<parametro2>=<sorgente2>

  Il nome dei parametri è quello che verrà passato al WS, mentre la sorgente è il campo di riferimento nel Modulo HTML.
- **ws-param-out**: elenco dei campi da popolare con la risposta del web-service separati da ";" (punto e virgola), la struttura della definizione risulta così composta:

  `<destinazione1>=<parametro1>;<destinazione2>=<parametro2>`

  Per ogni parametro si riporta l'identificatore del campo a cui verrà associato il parametro di risposta del WS.

  Talvolta è possibile determinare una destinazione sulla base di un pattern, quindi definendo un campo con all'interno un asterisco sarà possibile mappare una lista non su una `select` ma su dei campi che presentano lo stesso pattern, l'unico pattern valido è numerico e progressivo e parte il conteggio da 1.
- **ws-on**: rappresenta il selettore dell'elemento a cui associare la chiamata al web-service con il nome dell'evento a cui rispondere. Se l'evento non è presente in maniera predefinita è associato l'evento *click*. La sua rappresentazione risulta quindi essere così composta:

  <selettore>:<evento>
- **ws-allow-edit**: questa istruzione permette di inibire o meno all'utente compilatore la possibilità di modificare i valori inseriti nei campi; è possibile indicare un semplice "true/false" per un valore statico, oppure l'id di un campo di input che contenga il valore "true/false" calcolato secondo le necessità di redazione

  "true/false"

  #Field_Id

3.1.2 Sportello Telematico

Lo STU è un attore passivo ed agisce da *proxy* permettendo ai soli utenti autenticati presso di esso di poter invocare il WS originale; è inoltre in grado di alterare la richiesta del modulo aggiungendo informazioni quali l'identificativo dell'utente autenticato sul portale trasferendo queste informazioni al WS.

Nella versione standard è previsto un unico end point per i web service, al suo interno può prevedere meccanismi di *dispatching* (non previsti nella versione standard) che attraverso moduli aggiuntivi, *con l'uso di appositi hook*, potranno definire variazioni sulla chiamata quali:

- Variazione dell'end-point del web-service da invocare

- Alterazione delle informazioni che devono essere veicolate al web-service
- Aggiunta di meccanismi di autenticazione

Queste aggiunte devono essere concordate con Globo SRL.

3.1.3 Webservice

I WS utilizzati devono poter dialogare con protocollo REST-JSON; non sono previsti di default meccanismi di caching ed autenticazione delle chiamate che vengono effettuate al WS. Le chiamate avverranno tramite l'utilizzo del verbo `POST`.

Il WS ha la responsabilità di gestire le richieste che provengono dallo STU, implementare la logica applicativa e ritornare i dati risultato dell'elaborazione. È sua responsabilità ritornare le informazioni necessarie alla visualizzazione di informazioni, gestire casi in cui l'utente non abbia permessi sufficienti per l'accesso alle informazioni o i dati in ingresso non siano validi.

Le informazioni che tale WS riceverà saranno così composte:

```
{
  "name": "NOME_WS",
  "parameters": {},
  "account": "CODICE_FISCALE_UTENTE"
}
```

Tali campi contengono le informazioni definite all'interno del modulo HTML, in particolare:

- **`name`**: contiene il nome del WS definito nell'attributo ws-name del modulo HTML.
- **`parameters`**: contiene l'insieme delle informazioni derivanti dalla configurazione presente all'interno dell'attributo ws-param-in del Modulo HTML.
- **`account`**: contiene il codice fiscale dell'utente autenticato nello STU che sta effettuando la chiamata.

Il WS, utilizzando tali informazioni, si occuperà di restituire una struttura di informazioni così composta:

```
{
  "status" : "OK",
  "results" : {},
  "message": ""
}
```

Dove:

- **`status`**: può acquisire i valori `"OK"` o `"KO"`. Nel caso in cui il campo assume il valore `"OK"` il campo `results` DEVE essere presente, mentre nel caso in cui assume il valore `"KO"` il campo `message` DEVE essere presente;
- **`results`**: elenco di informazioni che possono essere utilizzate all'interno del modulo attraverso il mapping definito nell'attributo `ws-param-out`; Qualora uno dei campi presente è riportato come `JSON Object`, tale valore è utilizzato per popolare le option del campo target, aggiungendo -nel caso si trattasse di una select- il campo per la scelta di un elemento (Scegli un valore), nel caso in cui il campo è un `JSON Object` e il campo target definito contiene un asterisco -*- le chiavi del `JSON Object` saranno mappati su tutti i campi che rispettano tale pattern.

- **`message`**: messaggio di errore da mostrare in caso di errore lato WS (per esempio failure nell'accesso a banche dati).

Note tecniche: si supportano solo strutture dati non complesse nel campo `results`, quando si vuole ritornare un campo che andrà ad aggiungere dei valori in una `select` è necessario che il parametro in questione sia un `JSON Object` e che contenga solo coppie di chiave valore, in questo caso il suo utilizzo è più simile a un dizionario, in cui ogni coppia è una `option` nella `select`, nel campo `value` viene inserita la chiave e come testo viene inserito il suo valore associato a tale chiave, questo rende il campo `select` un ottimo campo per dare all'utente la possibilità di scegliere tra più opzioni tramite le etichette esposte per poi utilizzare in valore come chiave per un altro WS.

Definendo invece come campo target una stringa contenente un asterisco -*- il risultato del web service verrà mappato su tutti i campi che rispettato il tal pattern in ordine numerico, ad esempio definendo che il valore deve essere mappato su `Titolare*_Codice`, cercherà `Titolare1_Codice`, `Titolare2_Codice`, andando a mettere in ordine tutti i campi ricevuti dal web service.

3.2 Gestione degli errori

Validator è in grado di gestire le tipologie di errore riportate di seguito, in tutti i casi se la validazione è stata impostata come obbligatoria dal redattore del modulo non si potrà presentare l'istanza fino a che il WS non validerà i campi, ma si potrà comunque procedere al salvataggio come bozza.

3.2.1 Errore di connessione

Tale errore si riscontra quando vi è l'impossibilità di ottenere una risposta dalla parte di STU; tale tipologia di errore viene mostrato nell'interfaccia comune agli altri errori di compilazione, indicando il tipo di servizio che non può essere raggiunto.

3.2.2 Errore nel WS

Tale errore si riscontra quando lo STU non è in grado di contattare il WS o non è in grado di portare a termine con successo una richiesta che arriva dal Modulo HTML; in questo caso nella risposta dello STU la **status** prenderà il valore di ***KO***.

4 ESEMPIO

Definire un WS che ritorna una lista di istituti presso cui è possibile iscrivere il proprio figlio all'asilo.

4.1 Impostazioni necessarie

È necessario fornire allo sportello telematico l'end-point di accesso ai Web Service configurabile in **[Gestisci/STU configurazione/Configura i Web Servive di Validator]**

esempio: https://webservice.globogis.it/asili/list

4.2 Modulo HTML

Il modulo implementerà un input di tipo *select* che venga popolato alla pressione di un bottone presente nel modulo, sulla base del codice fiscale, dell'indirizzo e del codice comune del richiedente; il modulo conterrà quindi:

```
<!-- ... -->
<input name="Indirizzo" type="text" value="via Roma, 4" />
<input name="Codice_Del_Comune" type="hidden" value="A001" />
<select
  name="Lista Asili"
  id="Lista_Asili"
  ws-name="lista_asili"
  ws-param-in="cf_richiedente=Titolare_Codicefiscale;via=Indirizzo;¬
                                                codice_comune=Codice_Del_Comune"
  ws-param-out="Lista_Asili=lista_asili;Abilita_Mensa=permesso_mensa"
  ws-on="#ws_lista_asili:click"
/>
<input name="Abilita_Mensa" type="hidden" value="NO" />
<button id="ws_lista_asili" name="ws_lista_asili" />
<!-- ... -->
```

Analizziamo brevemente gli attributi custom coinvolti; tramite **`ws-param-in`** vengono identificati i parametri che devono essere inviati al WS per ottenere una risposta, in questo caso invieremo il codice fiscale del richiedente (presente nell'elemento `Titolare_Codicefiscale`) popolando il campo `cf_richiedente`; il suo indirizzo (assegnando al campo `via` il valore contenuto in `Indirizzo`) e il codice del comune (assegnando al campo `codice_comune` il valore contenuto in `Codice_Del_Comune`). Tali informazioni verranno utilizzate dal WS per procedere all'estrazione delle informazioni necessarie.

Alla pressione del bottone (ovvero all'evento `click` dell'oggetto con ID `ws_lista_asili`, come definito nel parametro **`ws-on`**) il modulo scatena un chiamata in POST verso STU, la cui richiesta è così formata:

```
{
  "name" : "lista_asili",
  "parameters" : {
    "via": "via Roma, 4",
    "codice_comune": "A001",
    "cf_richiedente" : "PPPPLT80R10M082K"
  }
}
```

Globo srl Soluzioni avanzate per il territorio
Viale Europa, 17/B 24048 Treviolo (BG)
☏ +39 035 596911 🖷 +39 035 596999
globo@pec.it info@globogis.it
Reg.Impr. di BG - Cod.Fisc. - P.IVA 02598580161
Capitale sociale € 300.000 interamente versato

4.3 Sportello telematico

Lo sportello telematico riceve la richiesta da parte del modulo e:

- Controlla che provenga da un utente autenticato
- Aggiunge la definizione dell'utente corrente (*property* `account`)
- Effettua il *dispatching* verso *WebServices* definito

Nello specifico prende la richiesta e la gira al WS configurato nello sportello telematico (https://webservice.globogis.it/asili/list) aggiungendo il campo `account`, il cui valore conterrà il codice fiscale dell'utente che è attualmente connesso al portale; la richiesta risulta quindi essere così costituita:

```
{
  "name": "lista_asili",
  "parameters": {
    "via": "via Roma, 4",
    "codice_comune": "A001",
    "cf_richiedente": "PPPPLT80R10M082K"
  },
  "account": "PPPPLT80R10M082K"
}
```

Il WebService presa in gestione la richiesta fornirà una risposta così formata:

```
{
  "status" : "OK",
  "results" : {
    "lista_asili": {
      "chiave_asilo1" : "Asilo n1",
      "chiave_asilo2" : "Asilo n2",
      "chiave_asilo3" : "Asilo n3"
    },
    "permesso_mensa": "SI"
  }
}
```

Lo sportello telematico riceve la risposta e la restituisce al Modulo HTML che provvede a inserire i valori di risposta nei rispettivi campi, come definito nell'attributo **`ws-param-out`** generando quindi il markup:

```
<input name="Indirizzo" type="text" value="via Roma, 4" />
<input name="Codice_Del_Comune" type="hidden" value="A001" />
<select
  name="Lista_Asili"
  id="Lista_Asili"
  ws-name="lista_asili"
  ws-param-in="cf_richiedente=Titolare_Codicefiscale;via=Indirizzo;¬
                                                codice_comune=Codice_Del_Comune"
  ws-param-out="Lista_Asili=lista_asili;Abilita_Mensa=permesso_mensa"
  ws-on="#ws_lista_asili:click"
>
  <option value="">Scegli un valore</option>
  <option value="chiave_asilo1">Asilo n1</option>
  <option value="chiave_asilo2">Asilo n2</option>
  <option value="chiave_asilo3">Asilo n3</option>
</select>
<input name="Abilita_Mensa" type="hidden" value="SI" />
<button id="ws_lista_asili" name="ws_lista_asili" />
```

Dove il valore di `lista_asili`, essendo un `object`, viene utilizzato per creare le opzioni di scelta all'interno della select di destinazione (identificata con il name `Lista_Asili`), mentre il valore di `permesso_mensa` viene assegnato al campo `Abilita_Mensa`.

In caso di errore di risposta da parte dell'WebService (HTTP status code 4XX e 5XX) o di time-out lo sportello telematico risponderà al Modulo HTML un JSON con la struttura:

```
{
  "status" : "KO",
  "message" : "Impossibile usufruire del servizio, riprovare in seguito."
}
```

Con messaggio di errore configurabile nella parte amministrativa dello sportello telematico, in tal caso il messaggio sarà mostrato all'utente.

Un ulteriore esempio sulla mappatura di un `JSON Object` in campi multipli tramite i pattern, come già detto l'unico pattern valido è numerico e progressivo e parte il conteggio da 1:

```
<button
  type="button"
  id="test-1"
  ws-name="getListaFamigliaREST"
  ws-param-in="codiceFamiglia=codice_famiglia"
  ws-param-out="Parametri_CFFam*=listaCodiceFiscale"
  ws-on="test-1:click" >

<input id="Parametri_CFFam1" name="Parametri_CFFam1">
<input id="Parametri_CFFam2" name="Parametri_CFFam2">
<input id="Parametri_CFFam3" name="Parametri_CFFam3">
<input id="Parametri_CFFam4" name="Parametri_CFFam4">
<input id="Parametri_CFFam5" name="Parametri_CFFam5">
```

In questo caso la `listaCodiceFiscale` viene mappata elemento per elemento sul `Parametri_CFFam1`, `Parametri_CFFam2` fino a che non termina l'elenco di valori o i campi disponibili, in entrambi i casi non viene riportato nessun errore.

Globo srl Soluzioni avanzate per il territorio
Viale Europa, 17/B 24048 Treviolo (BG)
☏ +39 035 596911 🖷 +39 035 596999
globo@pec.it info@globogis.it
Reg.Impr. di BG - Cod.Fisc. - P.IVA 02598580161
Capitale sociale € 300.000 interamente versato

5 NOTE FINALI

Questo documento definisce le specifiche interne per la redazione di sistemi di integrazione tra lo strumento Modulistica HTML e WS esterni, ma non definisce in maniera esaustiva tutte le casistiche possibili, pertanto è da considerare in uno stato di specifica che può essere ancora soggetta a variazioni. Si cercherà in ogni caso di garantire retro-compatibilità con quanto definito in tale documento anche nelle variazioni che potrebbero avvenire in futuro.