

# MYDICTIONARY: manuale grammatica

MyPay implementa l'integrazione con MyDictionary per quanto riguarda il rendering di form dinamiche per la diversificazione delle tipologie di dovuto spontaneo.

A livello di rendering MyPay supporta tre tipologie di campo:

*render\_mode="TEXT"*

\* Nome e cognome / Ragione sociale:

Nome e cognome / Ragione sociale



```
<xs:element name="nome_cognome_rag_sociale">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:annotation>
      <xs:appinfo>
        <my:render render_mode="TEXT" insertable="true" insertable_order="2" renderable="true"
renderable_order="2" html_label="Nome e cognome / Ragione sociale" detail_link="true" bind_cms="cm:name"/>
        <my:render_ex error_message="Specificare nome e cognome oppure la ragione sociale"
help_message="Specificare il nome e il cognome oppure la ragione sociale dell'intestatario della pratica" />
      </xs:appinfo>
    </xs:annotation>
    <xs:pattern value="^[1,50}$"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
```

Utilizzato per acquisire campi di tipo testuale alfanumerico.

*render\_mode="DATE"*

\* Data del verbale:



```
<xs:element name="data_verbale">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:annotation>
      <xs:appinfo>
        <my:render render_mode="DATE" insertable="true" insertable_order="4" renderable="true"
renderable_order="4" html_label="Data del verbale" detail_link="true" bind_cms="cm:name"/>
        <my:render_ex error_message="Specificare la data del verbale" help_message="Specificare la data
del verbale" validation_type="data"/>
      </xs:appinfo>
    </xs:annotation>
    <xs:pattern value="\d\d\d\d\d\d\d{4}"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
```

Utilizzato per acquisire date.

*render\_mode="SINGLESELECT"*

\* Ufficio competente:



```
<xs:element name="ufficio_competente">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <my:render render_mode="SINGLESELECT" insertable="true" insertable_order="5"
renderable="true" renderable_order="5" html_label="Ufficio competente" detail_link="true" bind_cms="cm:name"/>
          <my:render_ex error_message="Specificare l'ufficio Competente" help_message="Specificare
l'ufficio Competente" />
        </xs:appinfo>
      </xs:annotation>
      <xs:pattern value="^{1,50}$"/>
      <xs:enumeration value="" />
      <xs:enumeration value="Polizia Locale" />
      <xs:enumeration value="Edilizia" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Utilizzato per acquisire una scelta tra un elenco. Ogni elemento della combo si esprime con un tag *<xs:enumeration value="" />*.

*render\_mode="TAB"*

Belluno

Padova

Rovigo

Treviso

Venezia

Verona

Vicenza

```
<xs:element name="tabs_group" type="myp:tabs_province" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:appinfo>
      <my:render render_mode="TAB" insertable="true" insertable_order="7" searchable="false" searchable_order="7"
listable="true"
listable_order="7" renderable="true" renderable_order="7" bind_cms="tabs_group" html_class="center box"/>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

Il contenuto di *myp:tabs\_province* contiene una serie di campi che rappresentano le singole etichette dei tabs.

Utilizzato per creare delle schede.

*render\_mode="CURRENCY"*

Rata

1 €

```
<xs:element name="inner_addizionale_belluno_rata" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <my:render render_mode="CURRENCY" insertable="true" insertable_order="1" renderable="true" renderable_order="1"
html_label="Rata" detail_link="true" bind_cms="inner_addizionale_belluno_rata"/>
          <my:render_ex error_message="Specificare il campo Rata" help_message="Specificare il campo Rata" />
        </xs:appinfo>
      </xs:annotation>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```
</xs:simpleType>
</xs:element>
```

Utilizzato per acquisire un'importo.  
*render\_mode="CURRENCY\_LABEL"*

Totale 11 €

```
<xs:element name="totale">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <myp:render render_mode="CURRENCY_LABEL" insertable="false" insertable_order="20" renderable="true"
renderable_order="10" html_label="Totale" detail_link="true" bind_cms="totale" html_class="center" />
        </xs:appinfo>
      </xs:annotation>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Utilizzato per stampare etichette che rappresentano un'importo.

Per ogni tipologia di campo è possibile specificare:

*html\_label*: etichetta visibile a sinistra del campo

*html\_class*: campo utilizzato per specificare una o più classi css da applicare al layout del campo. (per specificare più classi occorre separarle da uno spazio).

*value\_depends\_on*: funzione javascript che calcola il valore del campo

*value\_depends\_on\_uids*: parametri (es. *value\_depends\_on\_uids*="campo1, campo2") della funzione javascript.

I campi possono essere allo stesso livello del campo da calcolare ed in tal caso è necessario solamente scrivere i nomi degli altri campi.

Se un campo si trova ad un livello "padre" rispetto a quello da calcolare occorre utilizzare il prefisso "parent." altrimenti se un campo si trova ad un livello "figlio" occorre indicare il path relativo per "raggiungere" il campo.

*valid\_depends\_on*: funzione javascript che calcola la validazione di un campo o di un insieme di campi.

*valid\_depends\_on\_uids*: parametri (es. *value\_depends\_on\_uids*="campo1, campo2") della funzione javascript.

I campi possono essere allo stesso livello del campo da calcolare ed in tal caso è necessario solamente scrivere i nomi degli altri campi.

Se un campo si trova ad un livello "padre" rispetto a quello da calcolare occorre utilizzare il prefisso "parent." altrimenti se un campo si trova ad un livello "figlio" occorre indicare il path relativo per "raggiungere" il campo.

`enabled_depends_on`: funzione javascript che calcola l'abilitazione di un campo o di un insieme di campi.

`enabled_depends_on_uids`: parametri (es. `value_depends_on_uids="campo1,campo2"`) della funzione javascript.

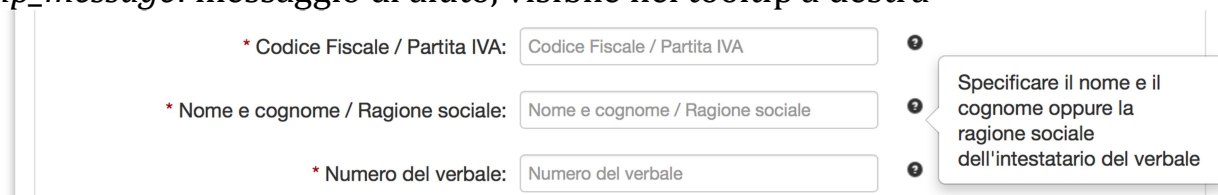
I campi possono essere allo stesso livello del campo da calcolare ed in tal caso è necessario solamente scrivere i nomi degli altri campi.

Se un campo si trova ad un livello "padre" rispetto a quello da calcolare occorre utilizzare il prefisso "parent." altrimenti se un campo si trova ad un livello "figlio" occorre indicare il path relativo per "raggiungere" il campo.

`causale_function`: funzione javascript che viene eseguita nel momento in cui un campo deve essere stampato nella causale (`join_template`). Il valore ritornato verrà utilizzato nel `join_template` al posto del valore del campo stesso. Può essere usata per definire formattazioni particolari nella causale. (es. un importo compare a video così 137,87€ e nella causale lo voglio così 13787)

`causale_function_uids`: il nome o la lista dei nomi dei campi per la funzione javascript di calcolo della causale.

`help_message`: messaggio di aiuto, visibile nel tooltip a destra



The screenshot shows a form with three input fields, each preceded by an asterisk and a label. The first field is labeled '\* Codice Fiscale / Partita IVA:' and contains the text 'Codice Fiscale / Partita IVA'. The second field is labeled '\* Nome e cognome / Ragione sociale:' and contains the text 'Nome e cognome / Ragione sociale'. The third field is labeled '\* Numero del verbale:' and contains the text 'Numero del verbale'. To the right of the fields is a tooltip with a question mark icon. The tooltip contains the text: 'Specificare il nome e il cognome oppure la ragione sociale dell'intestatario del verbale'.

`error_message`: messaggio di errore, visibile sopra la form, nel caso in cui fallisca la validazione del campo.

`pattern`: espressione regolare necessaria per la validazione del campo. E' possibile omettere la regex e valorizzare invece l'attributo `validation_type` nei seguenti casi per indurre validazioni specifiche:

- `id_univoco_FG`: per indurre una validazione specifica di Codici Fiscali oppure P.IVA
- `data`: per indurre una validazione specifica di una data che deve essere compresa in un range predefinito [-100 anni / +100 anni] rispetto alla data odierna.

`optional`: specificare il valore "true" se il campo è opzionale.

È da preferire comunque l'utilizzo del campo `minOccurs="0"` a livello di `<xs:element>` per indicare che il campo è opzionale.

L'elemento con "name=sys\_type", unico per il dovuto, non è renderizzato e mantiene le seguenti informazioni al livello di dovuto:

*join\_template*: template utilizzato da MyPay per la costruzione della causale, generalmente come concatenazione dei valori dei campi di interesse separati dal carattere "#".

*total\_included*: nome del campo che rappresenta l'importo finale per il dovuto