

## MANUALE BUILDER

### VERIFICHE E APPROVAZIONI

| VERSIONE | REDAZIONE   |            | CONTROLLO APPROVAZIONE |            | AUTORIZZAZIONE EMISSIONE |      |
|----------|-------------|------------|------------------------|------------|--------------------------|------|
|          | NOME        | DATA       | NOME                   | DATA       | NOME                     | DATA |
| V04      | V.MALACASA  | 22/05/2022 | A.DEIRO                | 22/05/2022 |                          |      |
| V03      | S. CASCIANO | 21/02/2022 | A.DEIRO                | 21/02/2022 |                          |      |
| V02      | S. CASCIANO | 11/11/2021 | A.DEIRO                | 11/11/2021 |                          |      |
| V01      | S. CASCIANO | 13/04/2021 | A.DEIRO                | 13/04/2021 |                          |      |

### STATO DELLE VARIAZIONI

| VERSIONE | PARAGRAFO O PAGINA         | DESCRIZIONE DELLA VARIAZIONE  |
|----------|----------------------------|---|
| V04      | Paragrafo 8 + vari         | Aggiunto paragrafo su bug nel json del Data Grid component.<br>Aggiunta sezione "8. Regole e best practice" |
| V03      | Revisione intero documento | Revisione intero documento  |
| V02      | Paragrafo 4.1              | Inserita la componente Campo Testo.   |
| V01      | Tutto il documento         | Versione iniziale del documento.  |

## INDICE

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>SCOPO E RIFERIMENTI DEL DOCUMENTO .....</b>                | <b>3</b>  |
| 1.1      | SCOPO DEL DOCUMENTO .....                                     | 3         |
| <b>2</b> | <b>GLOSSARIO.....</b>   | <b>3</b>  |
| <b>3</b> | <b>FUNZIONALITÀ DEL SISTEMA.....</b>                          | <b>3</b>  |
| 3.1      | CREAZIONE DI UN MODULO .....                                  | 3         |
| 3.2      | MODIFICA DI UN MODULO .....                                   | 4         |
| 3.3      | CONFIGURAZIONE DELLA NUOVA CATEGORIA .....                    | 4         |
| <b>4</b> | <b>STRUTTURA DI UN MODULO .....</b>                           | <b>4</b>  |
| 4.1      | AGGIUNGERE COMPONENTE IN UN MODULO .....                      | 4         |
| 4.2      | MODIFICARE COMPONENTE IN UN MODULO.....                       | 4         |
| 4.3      | NOME DELLE PROPRIETÀ (PROPERTY NAME).....                     | 5         |
| 4.4      | CAMPO TESTO (TEXT_FIELD).....                                 | 6         |
| 4.5      | CAMPO NUMERO (NUMBER) .....                                   | 12        |
| 4.6      | CAMPO CONTROLLO (CHECKBOX) .....                              | 12        |
| 4.7      | PANNELLO (PANNEL).....  | 12        |
| 4.8      | SELEZIONE .....   | 13        |
| 4.9      | SELECT BOXES COMPONENT .....                                  | 17        |
| 4.10     | DATA GRID .....   | 18        |
| 4.11     | EDIT GRID .....   | 19        |
| <b>5</b> | <b>CONTROLLI DI VALIDAZIONE .....</b>                         | <b>21</b> |
| 5.1      | CONTROLLI AUTODEFINITI .....                                  | 21        |
| 5.2      | CONTROLLI BASATI SU CODICE JAVASCRIPT .....                   | 22        |
| 5.3      | CONTROLLO SULL'ESTENSIONE E SUL CONTENUTO DELL'ALLEGATO ..... | 28        |
| <b>6</b> | <b>SERVIZI DI POPOLAMENTO.....</b>                            | <b>30</b> |
| <b>7</b> | <b>MAPPING DATI PER INIZIALIZZAZIONE.....</b>                 | <b>31</b> |
| 7.1      | INIT DA AUTENTICAZIONE .....                                  | 31        |
| 7.2      | INIT DA ANPR .....  | 32        |
| <b>8</b> | <b>REGOLE E BEST PRACTICE .....</b>                           | <b>34</b> |
| 8.1      | USER EXPERIENCE.....  | 34        |
| 8.2      | DESIGN .....  | 36        |
| 8.3      | FUNZIONAMENTO DEI CAMPI E JSON .....                          | 36        |

## 1 Scopo e riferimenti del documento

### 1.1 Scopo del documento

Lo scopo di questo documento è fornire un manuale per l'utente, per descrivere alcune delle funzionalità e le regole da utilizzare al fine di costruire i moduli utilizzando la componente del sistema MOON denominata builder.

Il documento è strutturato in due sezioni:

- Funzionalità del sistema
- Regole del sistema.

## 2 Glossario

| Acronimo | Definizione                                    |
|----------|--|
| MOOn     | Modulistica Open Online                        |
| API      | Nome attributo di un campo                     |
| ANPR     | Anagrafe Nazionale della Popolazione Residente |

## 3 Funzionalità del sistema

Per visualizzare il builder ed iniziare a creare un nuovo modulo è necessario che la propria utenza abbia il profilo di operatore\_builder. Se si possiede tale profilo nel menù del backoffice sarà visibile la voce "moduli", cliccando sulla quale sarà possibile visualizzare l'elenco dei moduli a cui si è abilitati e soprattutto avere a disposizione il tasto "Nuovo modulo".

### 3.1 Creazione di un modulo

Quando si crea un nuovo modulo vengono richiesti alcuni parametri base obbligatori:

| PARAMETRO    | DESCRIZIONE  |
|--------------|--|
| categoria    | Rappresenta la categoria di moduli all'interno della quale gli utenti visualizzeranno il modulo che si sta creando. Se non esiste ancora la categoria desiderata, assegnare il modulo alla categoria "altro" e quindi procedere alla <u>configurazione della nuova categoria</u> |
| Codice       | Codice da assegnare al modulo: deve essere indicato in maiuscolo, deve essere al massimo di 10 caratteri e non deve contenere spazi e caratteri speciali.  |
| Oggetto      | Oggetto dell'istanza rappresentata dal modulo  |
| Versione     | Lasciare sempre la versione iniziale 1.0.0   |
| Descrizione  | Breve descrizione del modulo   |
| Tipo di form | Sno disponibili 2 valori:  |

FORM = per costruire un modulo costituito da una sola pagina  
WIZARD = per costruire un modulo costituito da più pagine

### 3.2 Modifica di un modulo

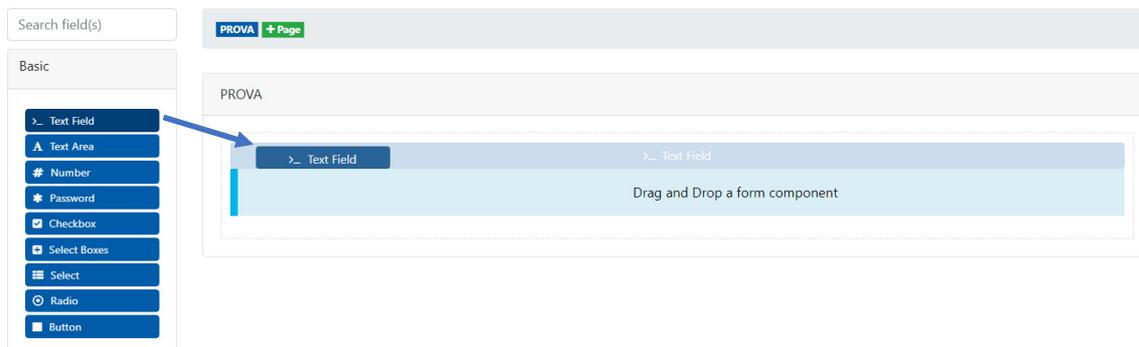
### 3.3 configurazione della nuova categoria

Al momento non è possibile creare in autonomia la categoria, ma occorre inviare una mail alla casella [supporto.moon@csi.it](mailto:supporto.moon@csi.it) con le indicazioni della categoria che si intende creare.

## 4 Struttura di un modulo

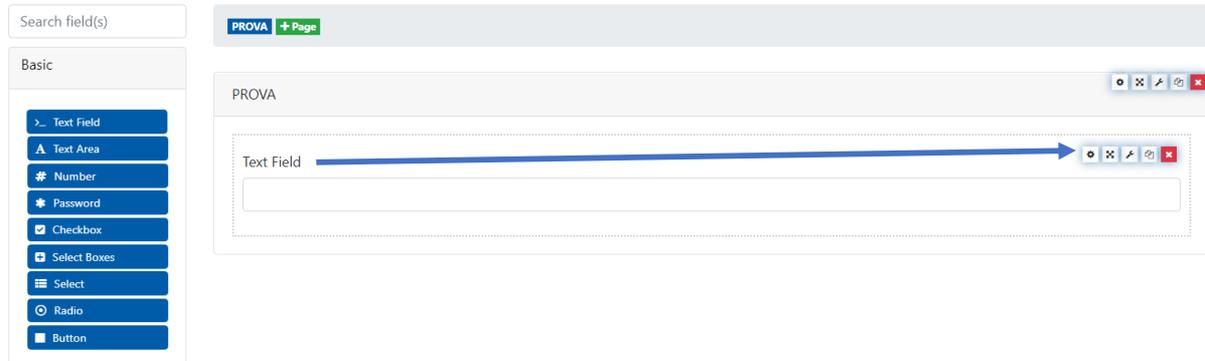
### 4.1 Aggiungere componente in un modulo

Per aggiungere un componente nel modulo, si può trascinare e rilasciare il componente dalla colonna sinistra nella posizione desiderata all'interno del modulo. Ogni nuovo modulo invia con un pulsante di invio (submit) aggiunto automaticamente ad esso.



### 4.2 Modificare componente in un modulo

Per modificare una componente di un modulo, bisogna passare sopra alla componente e cliccare sull'icona dell'**ingranaggio**. Si apriranno le impostazioni della componente da modificare. Esistono per ogni componente delle impostazioni di tipo generale e altre che sono invece specifiche per ogni componente.



### 4.3 Nome delle Proprietà (Property Name)

Durante la creazione delle componenti di un modulo, i titoli del campo (le etichette) vengono tradotte direttamente nel nome della proprietà nell'API, generata nel modulo. Si può modificare il nome della proprietà andando nella funzionalità API.

I Property Name hanno un duplice obiettivo:

- assegnare un nome al campo del modulo all'interno del json dei dati dell'istanza;
- individuazione dei dati di interesse quando si utilizza la funzionalità di esportazione presente nel back office.

Supponiamo di costruire un pannello dei dati del richiedente.

| Componenti               | Etichetta                   | Property Name creato dal builder | Property name consigliato per Moon | Presenti nel Json (dati) dell'istanza (si/no) |
|--------------------------|-----------------------------|----------------------------------|------------------------------------|---|
| <b>Panel Component</b>   | Dati richiedente            | datiRichiedente                  | panelDatiRichiedente               | NO  |
| <b>Text Field</b>        | Cognome                     | cognome                          | richiedente.cognome                | SI  |
| <b>Text Field</b>        | Nome                        | Nome                             | richiedente.nome                   | SI  |
| <b>Text Field (data)</b> | Data di nascita             | Data di nascita                  | richiedente.datINascita            | SI  |
| <b>Radio Button</b>      | Persona giuridica o fisica? | personaGiuridicaO Fisica         | richiedente.tipoPersona            | SI  |
| <b>Panel Component</b>   | Persona Fisica              | personaFisica                    | panelPersonaFisica                 | NO  |
| <b>Text Field</b>        | Cognome                     | Cognome                          | richiedente.personaFisica.cognome  | SI  |
| <b>Text Field</b>        | Nome                        | nome                             | richiedente.personaFisica.nome     | SI  |

|                        |                   |  |  |                           |
|------------------------|-------------------|--|--|---------------------------|
| <b>Panel Component</b> | Persona Giuridica |  |  | persona.giuridica         |
| <b>Text Field</b>      | Cognome           |  |  | persona.giuridica.cognome |
| <b>Text Field</b>      | Nome              |  |  | persona.giuridica.nome    |

È necessario seguire queste regole nella scrittura dei property name:

- Non inserire caratteri speciali;
- Si usa il punto per creare un oggetto all'interno di un nodo nel json finale, ad esempio persona.nome e persona.cognome creeranno la struttura json:  

```
“persona”: {  
    “nome”: “ ”,  
    “cognome”: “ “  
}
```
- Quando si vogliono unire due o più parole, queste vanno scritte in camel case, cioè attaccate e con le prime lettere maiuscole, ad esempio:  
“dichiarante.dataNascita” o “ente.legaleRappresentante.dataNascita

#### 4.4 Campo Testo (Text\_Field)

La componente di testo può essere utilizzato, per l'immissione di testo breve e generale. Sono disponibili opzioni per definire maschere di input e convalide, consentendo agli utenti di modellare le informazioni nei formati desiderati.

Le tabelle che seguono descrivono le principali funzionalità che si possono utilizzare per implementare la componente “Campo testo”:

- Display;
- Data;
- Validation;
- API;
- Conditional;
- Logic;
- Layout

Text Field Component

Display Data Validation API Conditional Logic Layout

Label  \*

Text Field

Label Position 

Top 

Placeholder 

Placeholder

Description 

1

Tooltip 

1

Prefix

Suffix

Widget 

Input Field 

Input Mask 

Allow Multiple Masks

Custom CSS Class 

Custom CSS Class

Tab Index 

0

Hidden 

Hide Label 

Show Word Counter 

Show Character Counter 

Hide Input 

Initial Focus 

Allow Spellcheck

Disabled 

Table View 

Modal Edit 

| DISPLAY   | DESCRIZIONE  | ESEMPI                            |
|---|--|-----------------------------------|
| Label - Etichetta                                 | Il nome o il titolo di questo componente   |                                   |
| Label Position – Posizione etichetta              | Posizione dell'etichetta per questo campo. Si usa sempre quella di default "TOP".  |                                   |
| Placeholder – Segnaposto                          | Il testo segnaposto che apparirà quando questo campo è vuoto   |                                   |
| Description - Descrizione                         | La descrizione è il testo che apparirà sotto il campo di input dell'etichetta.   |                                   |
| Tooltip - Descrizione comando                     | Aggiunge un'icona di suggerimento a lato di questo campo   |                                   |
| Prefix - Prefisso                                 | Il testo da mostrare prima di un campo. Un esempio è "€" per denaro.   |                                   |
| Suffix - Suffisso                                 | Il testo da mostrare dopo un campo. Un esempio potrebbe essere "Kg" per il peso.   |                                   |
| Widget  | Il widget è l'interfaccia utente del display utilizzata per inserire il  |                                   |
| Input Mask - Maschera di input                    | Aiuta a definire   |                                   |
| Allow Multiple masks - Consenti maschere multiple | Questa impostazione ti consentirà di impostare più maschere di input per il campo. La maschera viene selezionata dall'utente tramite un elenco a discesa e cambierà dinamicamente la maschera per il campo quando selezionata. Questa funzione è disponibile solo sul nostro renderer formio.js. | Normalmente non viene utilizzato. |
| Custom CSS Class - Classe CSS personalizzata      | Una classe CSS personalizzata da aggiungere a questo componente. Puoi aggiungere più nomi di classe separati da uno spazio. Si imposta vuota di default  |                                   |
| TabIndex - Indice delle schede                    | Imposta l' <code>tabindex</code> attributo di questo componente per sovrascrivere l'ordine di tabulazione del modulo. Questo campo di default non viene mai utilizzato.  |                                   |
| Hidden – Nascondi                                 | È possibile aggiungere una componente a un modulo e nascondere,  |                                   |
| Hide Label - Nascondi etichetta                   | Nascondi l'etichetta di questo componente. Questa impostazione   |                                   |
| Show Word Counter                                 |  |                                   |
| Show Charater Counter                             |  |                                   |
| Hide input  | Nascondi l'input durante la visualizzazione del modulo dal browser front-end. Questo non crittografa sul server. Non utilizzare per le password.   |                                   |
| Initial Focus                                     | Rendi questo campo l'elemento inizialmente focalizzato su questo modulo durante il rendering. Solo un componente in questo modulo o pagina della procedura guidata può contenere l'impostazione Focus iniziale.  |                                   |
| Allow Spellcheck - Controllo ortografico          | Consente il controllo ortografico  |                                   |
| Disabled – Disabilitato                           | <i>Disabilita questo campo sul modulo</i>  |                                   |
| Table view – Vista tabella                        | Se selezionato, questo valore verrà visualizzato nella vista tabella dell'elenco degli invii   |                                   |
| Modal Edit  | Consente di aprire alla modifica della componente  |                                   |

| DATE                                      | DESCRIZIONE  | ESEMPI                            |
|---|--|-----------------------------------|
| Multiple Value - Valori multipli          | Se selezionato, è possibile aggiungere più valori in questo campo. I valori appariranno come un array nell'API e un pulsante "Aggiungi un altro" sarà visibile sul campo consentendo la creazione di campi aggiuntivi per questo componente. |                                   |
| Default Value – Valore predefinito        | Il valore predefinito sarà il valore per questo campo, prima dell'interazione dell'utente. Avere un valore predefinito sovrascriverà il testo segnalibro.  |                                   |
| Persistent - Persistente                  | Se selezionato, il campo verrà archiviato nel database. Se vuoi che un campo non venga salvato, deseleziona questa casella. Questo è utile per campi come la convalida della password che non devono essere salvati.                         | Normalmente non viene utilizzato. |
| Input Format – Formato di input           | Preimpostato su "Plain".   |                                   |
| Protected -Protetto                       | Non viene mai valorizzato.   |                                   |
| Database index                            | Non viene mai valorizzato.   |                                   |
| Text case – Casi di testo                 | Una volta inseriti i dati è possibile modificare il valore   |                                   |
| Encrypted (Enterprise Only) Criptato      | Cripta questo campo sul server. Questa è una crittografia a due vie che non è adatta per le password. Questa impostazione è disponibile solo nel piano di progetto "Enterprise"  | Normalmente non viene utilizzato. |
| Redraw On                                 | Ridisegna un componente quando cambia un altro componente  |                                   |
| Clear Value When Hidden                   |  |                                   |
| Custom Default Value                      |  |                                   |
| Calculated Values – Valori calcolati      | I valori calcolati consentono di calcolare i valori in base ai valori in altri campi del modulo. I valori calcolati utilizzano javascript semplice e possono restituire qualsiasi valore al campo  |                                   |
| Calculate Value on server                 | Non viene mai valorizzato  |                                   |
| Allow Manual Override of Calculated Value | Non viene mai valorizzato  |                                   |

| VALIDATION                  | DESCRIZIONE   | NOTE / ESEMPI                     |
|-----------------------------|---|-----------------------------------|
| Validate On – Convalida     | Preimpostato su “Change”.   |                                   |
| Required – Necessario       | Se selezionato, il campo sarà richiesto per avere una risposta a ogni domanda. Se è necessario, potrebbe non essere necessario mantenere il valore in quanto si può presumere che venga controllato quando un modulo è stato inviato o non verrà inviato. |                                   |
| Unique – Unico              | Se selezionato, questo campo verrà applicato come univoco per questo modulo. Gli invii verranno controllati per vedere se un valore esistente corrisponde. Questa convalida è attualmente solo lato server.   | Normalmente non viene utilizzato. |
| Minimum Length              | Definire la lunghezza minima che questo campo dovrà soddisfare.   |                                   |
| Maximum Length              | Definire la lunghezza massima che questo campo dovrà soddisfare   |                                   |
| Minimum Word Length         | Definire la quantità minima di parole che possono essere aggiunte a questo campo  |                                   |
| Maximum Word Length         | Definire la quantità massima di parole che possono essere aggiunte a questo campo   |                                   |
| Regular Expression Parttern | Definire una regola per controllare il contenuto formale del campo  |                                   |
| Error Label                 | Definire un’etichetta quando si verifica l’errore   |                                   |
| Custom Error Message        | Definire un messaggio personalizzato quando di verifica l’errore  |                                   |
| Custom Validation           |   |                                   |
| Json Logic Validation       |   |                                   |

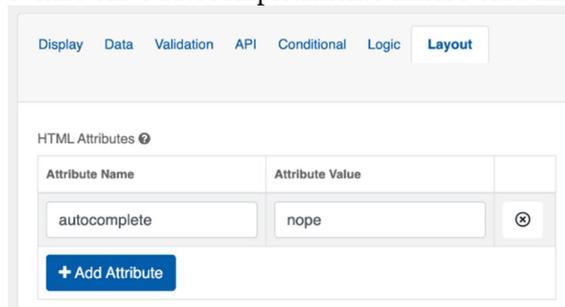
| API  | DESCRIZIONE  | NOTE / ESEMPI                     |
|--|--|-----------------------------------|
| Proprety name - Nome della proprietà         | Si imposta il nome della proprietà del campo.                                    |                                   |
| Field Tags - Tag del campo                   | Contrassegnare il campo per utilizzo nella logica personalizzata                 | Normalmente non viene utilizzato. |
| Custom Properties - Proprietà personalizzata | Configurare il campo in modo personalizzato definendo una chiave e il suo valore | Normalmente non viene utilizzato. |

| CONDITIONAL         | DESCRIZIONE  | NOTE / ESEMPI   |
|---------------------|--|---|
| Simple – Semplice   | Definire la logica condizionale, che si basa sulle seguenti regole:  |   |
|                     | - Ogni campo può essere nascosto o visualizzato.   |   |
|                     | - La visibilità dipende da un altro componente definito all'interno del modulo.  |   |
|                     | - La logica si attiva quando il campo configurato contiene il valore di testo in chiaro definito nelle impostazioni.   |   |
| Advanced - Avanzata | Si può anche utilizzare la logica condizionale avanzata, che utilizza JavaScript. La logica condizionale JavaScript richiede di impostare il valore di show su true o false. Hai accesso al valore corrente di qualsiasi componente del modulo tramite l'oggetto dati e la chiave API dei componenti. La logica condizionale avanzata funzionerà solo se la logica condizionale semplice non è già definita. | Viene utilizzata da esperti di linguaggio JavaScript. |

| LOGIC                            | DESCRIZIONE                  | NOTE / ESEMPI                     |
|----------------------------------|------------------------------|-----------------------------------|
| Advanced Logic – Logica Avanzata | Creare condizioni complesse. | Normalmente non viene utilizzato. |

| LAYOUT                           | DESCRIZIONE              | NOTE / ESEMPI        |
|----------------------------------|--------------------------|----------------------|
| HTML Attributes – Attributi HTML | Inserire degli attributi | autocoplete = 'nope' |

**! ATTENZIONE** – nei campi di testo in cui l'utente deve inserire dati comuni, come Email, nome, cognome ecc, impostare l'attributo “autocomplete” con valore “nope” nel tab “Layout” del campo, per evitare che l'autocompletamento infici l'esattezza dei dati inseriti.



#### 4.5 Campo Numero (Number)

Un campo numero può essere utilizzato per l'immissione di numeri e limitato a un tipo di numero.

L'impostazione della combo sono simili a quelle del campo di testo ma con alcune particolarità che riportiamo di seguito su ciascuna funzione:

| DATE   | DESCRIZIONE  | ESEMPI |
|--|--|--------|
| Use Thousands Separator - Usa il separatore delle migliaia | Se selezionato, il valore in questo componente separa le migliaia in base al delimitatore locale |        |
| Decimal Places - Decimali                                  | Il numero massimo di posizioni decimali per i valori in questo campo.                            |        |
| Require Decimal - Richiedi decimale                        | Mostra sempre i decimali per questo campo, anche se gli zero finali.                             |        |

| VALIDATION                     | DESCRIZIONE  | NOTE / ESEMPI |
|--------------------------------|--|---------------|
| Minimum Value - Valore minimo  | Se viene immesso un numero, il valore immesso nel modulo deve essere maggiore o uguale a questo valore         |               |
| Maximum Value - Valore massimo | Se viene immesso un numero, il valore immesso nel modulo deve essere minore inferiore o uguale a questo valore |               |

#### 4.6 Campo Controllo (Checkbox)

Un campo controllo consente di selezionare una o più opzioni e le caselle possono essere selezionate e deselezionate.

| DISPLAY                           | DESCRIZIONE                          | ESEMPI |
|-----------------------------------|--------------------------------------|--------|
| Label Width - Larghezza etichetta | Definire la larghezza dell'etichetta |        |
| Label Margin - Margine etichetta  | Definire il margine dell'etichetta   |        |

#### 4.7 Pannello (Panel)

L'elemento Panel costituisce il contenitore per tutti gli altri elementi. Non è indispensabile ma è consigliato utilizzarlo in quanto consente di raggruppare campi che definiscono un insieme omogeneo di dati: dati del richiedente, dichiarazioni, ecc.



È possibile modificare in modo minimo l'aspetto di layout del pannello, agendo esclusivamente sul colore di sfondo dell'intestazione. Per modificare il colore è sufficiente accedere all'editing dell'elemento e modificare il valore assegnato alla proprietà "Theme", scegliendo una tra le seguenti voci:

- default
- primary
- info
- success
- danger
- warning

Si consiglia per uniformità di realizzazione dei moduli di utilizzare unicamente i valori default o info, in questo modo:

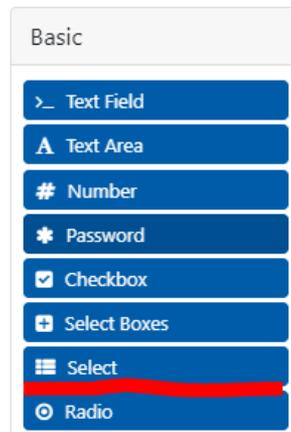
1. Tema default: per il primo pannello (creato automaticamente con la page e non visualizzato nel front-end) e per i pannelli di secondo livello (cioè contenuti dentro altri panel);
2. Tema info (azzurro): per il pannello subito sotto il primo (cioè il primo pannello mostrato nel front-end) e tutti gli altri pannelli di primo livello.

Una volta creato il pannello e inserito un nome nel campo "label" si potrà procedere con l'aggiunta degli altri elementi che costituiscono il modulo.

Assegnare al panel un property name di tipo "panel{{nomePagina}}", ad esempio "panelDichiarante".

#### 4.8 Selezione

La componente da usare per poter impostare delle selezioni è l'elemento **Select** (a tendina)



Per impostare i valori presenti in una select occorre aprire l'editing dell'elemento e selezionare il tab Data.



A questo punto è possibile scegliere le seguenti modalità:

- selezione multipla o singola (Multiple Values);
- tipo di origine dei dati (Data Source Type);
- tipo di archiviazione (Type Storage);

Se si vuole scegliere di avere una selezione multipla di valori allora bisogna selezionare il “Multiple Value”.

Per scegliere la tipologia con cui avere a disposizione i dati ci sono diversi tipi di origine dati:

Multiple Values ?

Data Source Type ?

Raw JSON x ▲

Type to search

Values

URL

Resource

Custom

Raw JSON

IndexedDB

Quelle ad oggi più utilizzate solo le seguenti:

1. Values = si possono inserire il nome e i valori

Data Source Values ?

|   | Label                               | Value                               |   |
|---|-------------------------------------|-------------------------------------|---|
| ☰ | <input type="text" value="prova1"/> | <input type="text" value="prova1"/> | ⊕ |
| ☰ | <input type="text" value="prova2"/> | <input type="text" value="prova2"/> | ⊕ |

2. URL = si inserisce il servizio creato su cui si possono recuperare i dati. Questa modalità consente di recuperare i dati da servizi. Si veda il capitolo “Servizi di popolamento” per vedere quali sono i servizi interrogabili ad oggi.

Data Source URL ?

Lazy Load Data ?

3. Raw JSON = consente di indicare i dati che deve contenere la select utilizzando il formato JSON. Questa modalità consente di definire, per ogni occorrenza di dato mostrato nella select, diversi attributi come ad esempio: codice, descrizione breve, descrizione estesa.

Multiple Values ⓘ

Data Source Type ⓘ

Raw JSON x ▾

Default Value ⓘ

Data Source Raw JSON ⓘ

```
1 [
2 {
3   "codice": "1",
4   "intervento": "Prova 1"
5 },
6 {
7   "codice": "2",
8   "intervento": "Prova 2"
9 }
10 ]
```

Storage Type ⓘ

String x ▾

Value Property ⓘ

The selected item's property to save.

Item Template ⓘ

```
1 <span>{{ item.codice }} - {{ item.intervento }}</span>
```

**! ATTENZIONE:** se si vogliono caricare molti dati, come ad esempio vie/corsi/piazze ecc. Dal servizio di popolamento toponomastica/vie, assicurarsi di spuntare l'opzione **“Disable limiting response”**, nel tab **“Data”** del componente. In questo modo vengono caricati tutti i dati senza un limite.

Quando si utilizza questa struttura è necessario specificare quale valore deve essere utilizzato per costruire l'etichetta mostrata all'interno della tendina. Nel campo **“Item template”** è necessario quindi indicare il nome dell'attributo usato

- item.codice – item.intervento → alla selezione si vedrà il testo concatenato formato dai valori contenuti negli attributi **“codice – intervento”**
- item.codice → alla selezione si vedrà solo il valore assegnato all'attributo codice
- item.intervento → alla selezione si vedrà solo il valore assegnato all'attributo intervento

### Tipo di widget

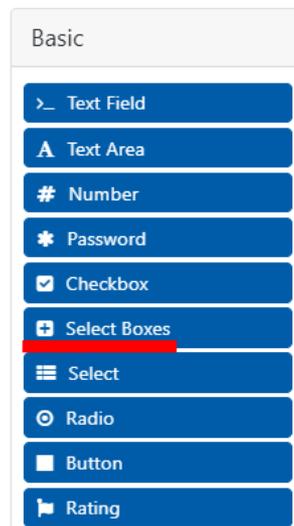
Quando le opzioni di scelta sono molte, ad esempio più di 10, si raccomanda di utilizzare il widget di tipo **“ChoicesJS”**, che consente di effettuare una ricerca testuale fra le opzioni disponibile, per rendere l'esperienza utente agevole.

Per impostare un testo e un limite nei risultati della ricerca testuale, andare in Data --> Chouces.js options e inserire il codice:

```
{"searchPlaceholderValue": "Scrivi il testo da cercare",  
"searchResultLimit": 10}
```

#### 4.9 Select Boxes component

La componente che consente di inserire più elementi contemporaneamente da poter selezionare almeno uno o più di uno è il **Select Boxes Component**.



È possibile inserire gli elementi attraverso la seguente funzionalità:

Select Boxes Component

Display **Data** Validation API Conditional Logic Layout

Default Value ?

- Malattia perdurante propria [1] o del figlio [2]
- Malattia perdurante [1] di un convivente [2]
- Minori conviventi o in affidamento di età da 0 a 13 anni [2]
- Nucleo monoparentale [2]
- Minori conviventi o in affidamento di età da 14 a 16 anni [2]
- Nessuna delle precedenti

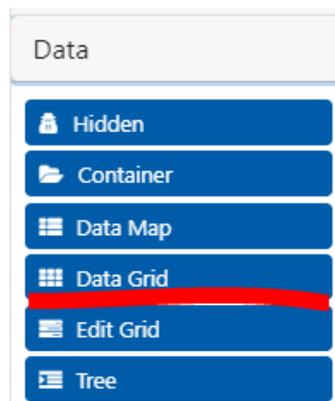
Values ?

|  | Label | Value * | Shortcut ? |
|--|-------|---------|------------|
|  |       |         |            |

Label → si inserisce il nome dell'elemento;  
Value → in automatico riporta l'attributo dell'elemento

#### 4.10 Data Grid

La componente che consente di raggruppare più elementi e aggiungerne più di uno è il **Data Grid**.



La componente **Data Grid** permette di impostare:

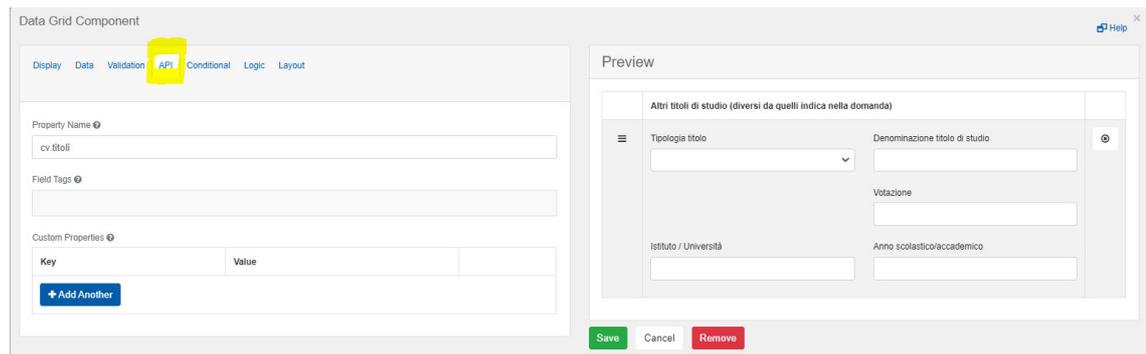
- la gestione di informazioni multiple, aggiungendo le stesse informazioni più volte;
- di riordinare le informazioni, per permettere una visione ordinata delle informazioni multiple richieste.

#### Vincoli

Il Data Grid e gli elementi che lo compongono, devono avere una struttura verticale sull'assegnazione dei nomi degli attributi (API), al fine di garantire la struttura logica del JSON.

Alla luce di questa considerazione è importante assegnare nomi semplici e non composti, come ad esempio nel caso dei titoli di studio:

il data grid si chiama “cv.titoli”: cv perché esiste un pannello chiamato Curriculum Vitae che contiene il data grid che si chiama Titoli.



All'interno del Data Grid abbiamo le componenti che contengono l'attributo semplice e non composto:

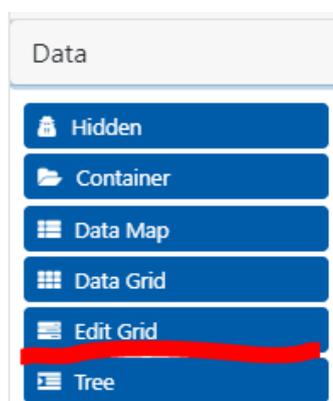
- Tipologia titolo → attributo semplice “tipologia”; se fosse composto e quindi fuori dal data grid sarebbe “cv.titolo.tipologia”;
- Denominazione titolo di studio → attributo semplice “denominazione”; se fosse composto e quindi fuori dal Data Grid sarebbe “cv.titolo.denominazione”.

**! ATTENZIONE:** il componente **Data Grid** mantiene nel suo json tutte le modifiche fatte, ciò significa che se ad esempio viene eliminato un campo al suo interno, questo campo verrà eliminato visivamente ma resterà all'interno della struttura json del Data Grid.

**Quando si apportano modifiche a questo componente, assicurarsi di verificare sempre che il json del modulo e delle istanze siano corretti.**

#### 4.11 Edit Grid

La componente che consente di elencare dati che vengono raggruppati in un'unica componente è l'**Edit Grid**.



La componente **Edit Grid** permette di riassumere i dati, utilizzando la sezione dell'editing denominata "template". Tale sezione è valorizzata di default per mostrare il valore assegnato a ciascuno dei campi inseriti all'interno del componente edit grid. Per modificare tale impostazione e visualizzare solo alcuni dei valori seguire gli esempi riportati:

#### Header Template

```
<div class="row">
  <div class="col-sm-2">Anno Dich.</div>
  <div class="col-sm-2">Sezione</div>
  <div class="col-sm-2">Foglio</div>
  <div class="col-sm-2">Particella</div>
  <div class="col-sm-2">Subalterno</div>
</div>
```

#### Row Template

```
<div class="row">

  {% util.eachComponent(components, function(component) { %}

  {% if(component.key == 'annoDichiarazione' ||
    component.key == 'sezione' ||
    component.key == 'foglio' ||
    component.key == 'particella' ||
    component.key == 'subalterno' ) { %}
  <div class="col-sm-2">
    {{ getView(component, row[component.key]) }}
  </div>
  {% } %}

  {% }) %}

  <div class="col-sm">
    <div class="btn-group pull-right">
      <button class="btn btn-default btn-light btn-sm editRow"><i class="{{ iconClass('edit')
      }}"></i></button>
    </div>
    {% if (!instance.options.readOnly && !instance.originalComponent.disabled) { %}
```

```
{% if (!instance.hasRemoveButtons || instance.hasRemoveButtons()) {%}  
  <button class="btn btn-danger btn-sm removeRow"><i class="{{ iconClass('trash')  
  }}"></i></button>  
  {% } %}  
  {% } %}  
</div>  
</div>  
</div>
```

## 5 Controlli di validazione

### 5.1 Controlli autodefiniti

Vi sono 2 tipi di controlli autodefiniti:

- Obbligatorietà
- Controllo formale (basato su Regular expression)

Per inserire un controllo di obbligatorietà, occorre

- Accedere all'editing dell'elemento
- Selezionare il tab "Validation"
- Selezionare l'opzione "Required"

Per inserire un controllo formale, occorre

- Accedere all'editing dell'elemento
- Selezionare il tab "Validation"
- Indicare la regular expression nel campo "Regular Expression Pattern"

Esempi di regular expression

Per codice fiscale

```
[A-Z]{6}[A-Z0-9]{2}[A-Z][A-Z0-9]{2}[A-Z][A-Z0-9]{3}[A-Z]
```

Per telefono

```
[+]?((?!.*[+\\s-]{2})([\\d][-\\s]*)){7,15}
```

Il significato della regex è il seguente:

1. il [+] iniziale permette l'inserimento (non obbligatorio) dei prefissi internazionali come +39 per Italia o +1264 per Anguilla;
2. la parte ([\\d][-\\s]\*) permette all'utente l'inserimento di sole cifre e di spazi o di - per un inserimento più simile alla scrittura classica di un numero e sua conseguente più facile lettura:

333-555-0123 o 333 555 0123 o 333 55 50 123 etc..

3. la parte `(?!.*[+\s-]{2})` permette di evitare che l'utente abusi della disponibilità di inserire pure spazio o - inserendoli più di una volta successivamente evitando casi come:

333-----5555----0123 o 333 555- - - -0123 etc...

4. la parte `{7,15}` serve a limitare il numero di cifre (*non contano eventuali spazi o -*) ad un massimo di 15 cifre, numero massimo consentito dallo standard internazionale, e un numero minimo di 7 che permette non solo di accettare casi rari di numeri fissi Italiani con poche cifre, solitamente riservati ad aziende o numeri verdi o di emergenza, ma pure casi come per Sant'Elena che il numero nazionale è composto da sole 4 cifre e con aggiunta del prefisso internazionale raggiunge i 7:

+290XXXX)

Si consiglia poi di aggiungere un messaggio esplicativo per il caso di validazione come il seguente: *si prega di inserire il numero correttamente, con i seguenti esempi: "+393333333333 , +39 333 333 3333, 333-333-3333,333 333333, 333333333"*, nel campo "Custom error message"

## 5.2 Controlli basati su codice javascript

Per inserire un controllo di tipo javascript, occorre

- Accedere all'editing dell'elemento
- Selezionare il tab "Validation"
- Aprire la sezione "Custom validation"
- Nell'area di testo, occorre inserire il codice javascript che effettua il controllo.

Si riportano di seguito alcuni esempi di controllo già implementati

### 5.2.1 Controllo su codice fiscale basato sul codice di controllo

```
var cf = input;
var result = true;
if( cf.length > 0 ) {
    var validi, i, s, set1, set2, setpari, setdisp;
    cf = cf.toUpperCase();
    if( cf.length != 16 ) {
        result = "Il codice fiscale deve contenere 16 caratteri";
    }
    else {
        validi = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
        for( i = 0; i < 16; i++ ){
            if( validi.indexOf( cf.charAt(i) ) == -1 )
                result = "Formato non corretto";
        }
        set1 = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        set2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKL";
        setpari = "ABCDEFGHIJKL";
    }
}
```

```
setdisp = "BAKPLCQDREVOSFTGUHMINJWZYX";
s = 0;
for( i = 1; i <= 13; i += 2 )
    s += setpari.indexOf( set2.charAt( set1.indexOf( cf.charAt(i) ) ));
for( i = 0; i <= 14; i += 2 )
    s += setdisp.indexOf( set2.charAt( set1.indexOf( cf.charAt(i) ) ));
if( s%26 != cf.charCodeAt(15)-'A'.charCodeAt(0) ) {
    result = "Formato non corretto";
}
else {
    result = true;
}
}
}
valid = result;
```

#### 5.2.2 Controllo su codice fiscale basato su rigenerazione dei dati che lo costituiscono

```
var cf = data.codiceFiscale;
var nome = data.nome;
var cognome = data.cognome;
var dataNascita = data.dataNascita;
var sesso = data.sesso;
var result = 'true';

if (cf.length!=16) {
    result = "Attenzione! La lunghezza del codice fiscale inserito non è corretta.";
}
else {
    var isCFValido = cf.match("[a-zA-Z]{6}[a-zA-Z0-9]{2}[a-zA-Z][a-zA-Z0-9]{2}[a-zA-Z][a-zA-Z0-9]{3}[a-zA-Z]");
    if(!isCFValido){
        result = "Errore! Formato codice fiscale non valido.";
    }
    else {

        if (cognome!=null) {
            cognome=cognome.trim().toUpperCase();
            var codice_cognome = cognome.replace(/^[BCDFGHJKLMNPQRSTVWXYZ]/gi,"") =
cognome.replace(/^[^BCDFGHJKLMNPQRSTVWXYZ]/gi,"");
            codice_cognome += cognome.replace(/^[^AEIOU]/gi,"");
            codice_cognome+='XXX';
            codice_cognome=codice_cognome.substr(0,3);
            var strCognome = codice_cognome.toUpperCase();

            if (strCognome != cf.substr(0,3)) {
                result = "Attenzione: il codice fiscale non è compatibile con il cognome
dello studente inserito: si prega di verificare usando la tessera sanitaria.";
```

```
    }
    else {
        //controllo nome
        if (nome!==null) {
            nome=nome.trim().toUpperCase();
            var codice_nome =
nome.replace(/^[BCDFGHJKLMNPQRSTVWXYZ]/gi,");
            if(codice_nome.length>=4){
                codice_nome=
                codice_nome.charAt(0)+
                codice_nome.charAt(2)+
                codice_nome.charAt(3);
            }else{
                codice_nome+=nome.replace(/^[AEIOU]/gi,");
                codice_nome+='XXX';
                codice_nome=codice_nome.substr(0,3);
            }
            var strNome = codice_nome.toUpperCase();

            if (strNome!=cf.substring(3,6)) {
                result = "Attenzione: il codice fiscale non è compatibile
con il nome dello studente inserito: si prega di verificare usando la tessera sanitaria.";
            }
            else {
                //controllo data di nascita
                if (dataNascita!==null) {
                    var msg = "Attenzione: il codice fiscale non è
compatibile con la data di nascita dello studente inserito: si prega di verificare usando la tessera sanitaria.
"

                    giorno = dataNascita.substring(0,2);
                    //var numGiorno = (giorno > 9 )? giorno :
giorno.substring(1,2);

                    mese = dataNascita.substring(3,5);
                    var numMese = (mese > 9 )? mese :
mese.substring(1,2);

                    var indMese = numMese -1;
                    anno = dataNascita.substring(8,10);

                    if (anno!=cf.substring(6,8)) {
                        result = 'errore anno' + msg;
                    }
                    var
tavola_mesi=['A','B','C','D','E','H','L','M','P','R','S','T'];
```



```
        result = msgErrFormat;
    }
    set1 = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    set2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMN";
    setpari = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    setdisp = "BAKPLCQDREVOSFTGUHMINJWZYX";
    s = 0;
    for( i = 1; i <= 13; i += 2 )
        s += setpari.indexOf( set2.charAt( set1.indexOf( cf.charAt(i) ) ));
    for( i = 0; i <= 14; i += 2 )
        s += setdisp.indexOf( set2.charAt( set1.indexOf( cf.charAt(i) ) ));
    if( s%26 != cf.charCodeAt(15)-'A'.charCodeAt(0) ) {
        result = msgErrFormat;
    }
}
valid = (result === 'true') ? true : result;
```

### 5.2.3 Controllo su codice fiscale /partita iva

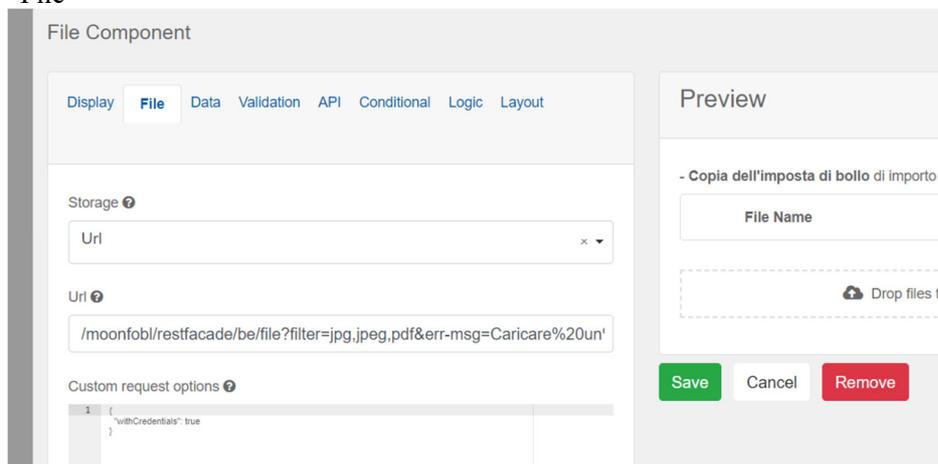
```
var cf = input;
var result = true;
if( cf.length > 0 ) {
    if( cf.length > 12 ) {
        var validi, i, s, set1, set2, setpari, setdisp;
        cf = cf.toUpperCase();
        if( cf.length != 16 ) {
            result = "Il codice fiscale deve contenere 16 caratteri";
        }
    }
    else {
        validi = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
        for( i = 0; i < 16; i++ ){
            if( validi.indexOf( cf.charAt(i) ) == -1 )
                result = "Formato non corretto";
        }
        set1 = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        set2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMN";
        setpari = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        setdisp = "BAKPLCQDREVOSFTGUHMINJWZYX";
    }
}
```



```
if (typeof tel !== 'undefined' && tel !== "") {  
  if(/\\d{7,12}/.test(tel)) {  
    isValidHomePhoneIt = true;  
  }  
  else {  
    isValidHomePhoneIt = false;  
  }  
}  
valid = isValidHomePhoneIt;
```

### 5.3 Controllo sull'estensione e sul contenuto dell'allegato

Per configurare correttamente l'allegato, occorre impostare il parametro della utl nelle "Impostazioni", tab "File"



#### 1. Url che indica, dove vengono salvati gli allegati:

La url utilizzata per salvare gli allegati dell'istanza è la seguente:

**/moonfobl/restfacade/be/file**

oppure nel caso dei un modulo pubblicato su modulistica (senza autenticazione)

**/modulistica/restfacade/be/file**

## **2. Filto relativo all'estensione e al contenuto del file.**

E' possibile impostare filtri lato client che permettono il caricamento dei file con l'estensione desiderata, utilizzando il campo "File pattern" nel tab "File" delle impostazioni.

File Pattern ⓘ

\*.pdf,\*.jpeg,\*.jpg

A volte può capitare che l'utente FO carichi un allegato con il contenuto che non corrisponde all'estensione, es. il file PNG al quale viene cambiata l'estensione in JPG per poter effettuare l'upload nel modulo.

Per impostare la validazione del contenuto dell'allegato lato server si possono utilizzare le seguenti url:

- per controllare che il file d'immagine abbia il contenuto corretto si utilizza la seguente url sempre nel campo url):

**/moonfobl/restfacade/be/file?filter=jpg,jpeg,pdf&errmsg=Caricare%20un%20file%20PDF%20oppure%20immagine%20jpg**

In questo caso la url contiene un filtro nella query string per il controllo del contenuto lato server ed il messaggio di errore da restituire all'utente.

- per la verifica se il file in upload è firmato.

`/file?filter=jpg,jpeg,pdf`

`/file?filter=pdf,pkcs7-signature`

`/file?filter=pdf,pkcs7-signature&err-msg="Caricare un file PDF oppure firmato .pdf.P7M"`

`/file?filter=pdf,pkcs7-signature&err-`

`msg=Caricare%20un%20file%20PDF%20oppure%20firmato%20.pdf.P7M`

## 6 Servizi di popolamento

Nella tabella sottostante, per ogni tipologia di dato acquisibile attraverso l'interrogazione ad un servizio, sono indicati i valori da utilizzare in alcuni parametri specifici

- Data source url
- Item template: in questo caso il valore assegnato a questo parametro deve sempre essere del tipo “<span>{{ item.NomeParametro }}</span>” dove NomeParametro è indicato nella relativa colonna. Il valore per esempio potrà essere “<span>{{ item.nome }}</span>”. In alcuni casi può essere composto da più parametri, come nel caso descritto per le province piemontesi.

Si consiglia inoltre di:

- non selezionare l'opzione “**Lazy load data**”: questo parametro infatti consente se selezionato di caricare il dato solo quando l'utente seleziona il campo; questo comportamento non è auspicabile perché in generale si preferisce mostrare il dato selezionato appena il modulo viene visualizzato;
- spuntare l'opzione “**Disable limiting response**”, nel tab “Data” del componente, soprattutto se si vogliono caricare molti dati.

| Tipologia dati            | DATA SOURCE URL  | Item template - NomeParametro               |
|---------------------------|--|---|
| Tutte le nazioni          | /moonfobl/restfacade/be/extra/demografia/anpr/nazioni                    | nome  |
| Nazioni EU esclusa italia | /moonfobl/restfacade/be/extra/demografia/anpr/nazioni?uso=CIT&ue=UENOITA | nome  |
| Nazioni non UE            | /moonfobl/restfacade/be/extra/demografia/anpr/nazioni?uso=CIT&ue=NONUE   | nome  |
| Tutte le province         | /moonfobl/restfacade/be/extra/istat/province                             | nome  |
| Province piemontesi       | /moonfobl/restfacade/be/extra/istat/regioni/1/province                   | <span>{{ item.nome}} {{ item.sigla}}</span> |

|  |  |                         |
|--|--|-------------------------|
| Comuni per provincia                           | /moonfobl/restfacade/be/extra/istat/province/{data.provincia.codice}/comuni  | nome                    |
| Elenco vie                                     | /moonfobl/restfacade/be/extra/territorio/toponomastica/vie   | nome                    |
| Numero civico sulla base della via selezionata | /moonfobl/restfacade/be/extra/territorio/toponomastica/vie/{data.nuovoindirizzo.nomeVia.codice}/numeri   | <span>{{ item }}</span> |
| Civico completo con filtro su civici           | /moonfobl/restfacade/be/extra/territorio/toponomastica/vie/{data.nuovoindirizzo.nomeVia.codice}/numeri/{data.nuovoindirizzo.civico}/civici?tipologieCivico=1,3 | nome                    |
| Civico completo senza filtro                   | /moonfobl/restfacade/be/extra/territorio/toponomastica/vie/{data.nuovoindirizzo.nomeVia.codice}/numeri/{data.nuovoindirizzo.civico}/                           | nome                    |
| Piano NUI                                      | /moonfobl/restfacade/be/extra/territorio/toponomastica/vie/{data.nuovoindirizzo.nomeVia.codice}/civici/{data.nuovoindirizzo.civicoCompleto.codice}/pianinui    | nome                    |
| Piani elenco standard                          | /moonfobl/restfacade/be/extra/territorio/toponomastica/piani   | nome                    |

Per le select i cui valori dipendono dalla selezione di una select precedente, occorre che nel parametro url sia indicato nel modo corretto il riferimento al valore del parametro Property Name della select di partenza usato per effettuare l'interrogazione.

Esempio:

prima select mostra l'elenco delle vie ed il suo parametro Property Name si chiama nuovoindirizzo.nomeVia

seconda select con i civici corrispondenti alla via selezionata: nella url deve essere indicato il riferimento compreso tra doppie parentesi graffe /{{ data.nuovoindirizzo.nomeVia.codice }}/ .

## 7 Mapping dati per Inizializzazione

In questo capitolo vengono elencati i nomi da assegnare al parametro Property Name nel tab API se si desidera che i campi relativi siano valorizzati in fase iniziale con dei valori acquisiti dall'autenticazione oppure dall'interrogazione del servizio anagrafico ANPR. Questo secondo caso è subordinato all'approvazione da parte dell'ufficio anagrafe.

### 7.1 Init da autenticazione

|                                   |                           |
|-----------------------------------|---------------------------|
| Nome utente autenticato           | richiedente.nome          |
| Cognome utente autenticato        | richiedente.cognome       |
| Codice fiscale utente autenticato | richiedente.codiceFiscale |

## 7.2 Init da ANPR

I dati esterni non sono indicati in quanto facilmente deducibili.

I dati valorizzati possono essere un sottoinsieme di quelli elencati.

| Dato esterno | Property name   |
|--------------|---|
|              | idFamigliaConvivenzaANPR                                    |
|              | richiedente.codiceFiscale                                   |
|              | richiedente.cognome   |
|              | richiedente.nome  |
|              | richiedente.sesso   |
|              | richiedente.nascita.data                                    |
|              | richiedente.nascita.stato                                   |
|              | richiedente.nascita.provincia                               |
|              | richiedente.nascita.comune                                  |
|              | richiedente.nascita.cittaestera                             |
|              | richiedente.cittadinanza.tipo                               |
|              | richiedente.cittadinanza.stato                              |
|              | richiedente.provenienza.tipo (fisso: Altro comune italiano) |
|              | richiedente.provenienza.stato (fisso: ITALIA)               |
|              | richiedente.provenienza.provincia                           |
|              | richiedente.provenienza.comune                              |
|              | richiedente.provenienza.indirizzo                           |
|              | richiedente.provenienza.civico                              |
|              | richiedente.provenienza.cap                                 |
|              | richiedente.genitori.nomeMadre                              |
|              | richiedente.genitori.cognomeMadre                           |
|              | richiedente.genitori.nomePadre                              |
|              | richiedente.genitori.cognomePadre                           |

|                                    |  |
|------------------------------------|--|
| Decodifica dello stato civile      | richiedente.statocivile.stato  |
|                                    | richiedente.statocivile.coniuge.nome   |
|                                    | richiedente.statocivile.coniuge.cognome  |
|                                    | richiedente.statocivile.coniuge.dataNascita  |
|                                    | richiedente.statocivile.coniuge.luogoNascita   |
|                                    |  |
|                                    | richiedente.statocivile.matrimonio.data  |
|                                    | richiedente.statocivile.matrimonio.luogo   |
|                                    | I seguenti campi devono essere inseriti in un elemento edit.grid o data.grid in quanto possono essere multipli |
| Codice fiscale                     | componente.cf  |
|                                    | componente.cognome   |
|                                    | componente.nome  |
|                                    | componente.sesso   |
|                                    | componente.dataNascita   |
| Calcolato e valorizzato con (S /N) | componente.minorenne   |
|                                    | componente.rapportoDiParentela   |
|                                    | componente.cittadinanza  |
|                                    | componente.statoNascita  |
|                                    | componente.provincia   |
|                                    | componente.comune  |
|                                    | componente.cittaestera   |
|                                    | componente.nomeMadre   |
|                                    | componente.cognomeMadre  |
|                                    | componente.nomePadre   |
|                                    | componente.cognomePadre  |
|                                    | componente.statocivile.stato   |
|                                    | componente.statocivile.coniuge.nome  |
|                                    | componente.statocivile.coniuge.cognome   |
|                                    | componente.statocivile.coniuge.dataNascita   |
|                                    | componente.statocivile.coniuge.luogoNascita  |

|           |   |
|-----------|---|
|           | componente.statocivile.matrimonio.luogo |
|           | componente.statocivile.matrimonio.data  |
| calcolato | richiedente.sonoPresentiMinori          |

## 8 Regole e best practice

Questo capitolo vuole essere una guida con buone pratiche da seguire per ottimizzare l'efficacia dei moduli, mantenere coerenza fra loro e semplificare l'operatività degli attori coinvolti.

Queste best practice riguardano aspetti di user experience, design, struttura dei campi e json.

### 8.1 User Experience

Nel progettare un modulo, anche molto semplice, tenere sempre a mente le necessità di chi dovrà presentare l'istanza, che potrebbe incontrare vari ostacoli nel compilare il modulo (ad esempio scarsa conoscenza della lingua italiana, poca dimestichezza con gli strumenti informatici, conoscenza limitata dell'ambito di riferimento, ecc...).

Bisogna assicurarsi che i campi siano ben organizzati e che le informazioni siano richieste in modo chiaro, per garantire una buona esperienza d'uso all'utente finale.

Alcune regole di User Experience da tenere a mente:

1. Chiedere sempre al cliente una descrizione degli attori coinvolti (se possibile);
2. Assegnare etichette comprensibili e inequivocabili ai vari campi (ad esempio, "Provincia" anziché "Prov");
3. Nei campi dove bisogna inserire una **data**, nelle impostazioni del widget calendar del campo sostituire il formato della data "yyyy-MM-dd" con "dd-MM-yyyy" e nel placeholder scrivere "gg-mm-aaaa";
4. Usare i componenti panel e page per raggruppare e organizzare le informazioni (ad esempio usare un panel "Sede in" o "Sede aziendale" per raggruppare tutti i campi che si riferiscono alla sede di un'azienda);

Riferimenti anagrafici proprietario

|                                      |                      |
|--------------------------------------|----------------------|
| Cognome *                            | Nome *               |
| <input type="text"/>                 | <input type="text"/> |
| Cellulare *                          | Email *              |
| <input type="text" value="( ) - -"/> | <input type="text"/> |
| Telefono *                           | PEC                  |
| <input type="text"/>                 | <input type="text"/> |

5. Usare descrizioni e tooltip per specificare come usare un campo (ad esempio come scrivere una certa informazione, quali estensioni di file si possono caricare, quanti caratteri si possono scrivere in una textarea);
6. Usare descrizioni e tooltip per specificare cosa viene richiesto, qualora l'etichetta non basti;
7. Fare in modo che l'utente sappia quali sono i criteri di validazione di un campo;

Inserire solo le cifre della partita iva (sono  
P.IVA ⓘ \*consentiti gli spazi per separare le cifre per una  
più semplice lettura)

8. usare maschere, prefissi e suffissi per guidare l'utente nella scrittura di un valore;
9. Nelle select che hanno più di 10 opzioni utilizzare il widget ChoicesJS;
10. Se le opzioni fra cui scegliere sono poche (da 2 a 4), usare radio button o select boxes anziché le select (in questo modo l'utente dovrà fare un click in meno);
11. Usare il componente Content per aggiungere testi lunghi o immagini utili;
12. Se possibile, fare eseguire dei test dei moduli a una persona terza che non ha lavorato al modulo;
13. Per accettazioni di testi lunghi (ad esempio le informative sulla privacy), scrivere il testo da accettare in un componente di tipo Content e poi usare un componente di tipo Checkbox per la presa visione e/o accettazione del testo;



**ZTL Area Romana\***

compresa nel perimetro: piazza Emanuele Filiberto (compresa) - risvolto sud/ovest di piazza della Repubblica; via Milano (esclusa) - via Corte d'Appello (esclusa) - risvolto nord/est di piazza Savoia - via della Consolata (esclusa) - via Santa Chiara (esclusa) - via delle Orfane (esclusa)

Presa visione dei campi soprastanti \*

14. Evitare di usare negazioni e frasi ambigue, ad esempio la frase che segue:

**E' prevista l'assenza di barriere architettoniche per la partecipazione all'iniziativa di persone con disabilità**

SI

NO

Meglio sostituire con: "Saranno previste barriere architettoniche...?".

## 8.2 Design

Sebbene il design di questi moduli sia minimale, ci sono alcune regole da seguire per garantire uniformità visiva fra i moduli:

1. Il bottone SALVA BOZZA deve essere giallo, posizionato in basso a destra su ogni pagina del modulo;
2. In ogni pagina del modulo bisogna sempre aggiungere un panel iniziale, oltre a quello di default (che non viene mostrato nel front-end). Il primo panel è quello creato automaticamente quando si aggiunge la pagina, e questo avrà il tema default (grigio). Il panel subito al suo interno viene aggiunto manualmente e deve avere il tema info (azzurro);
- 3.

## 8.3 Funzionamento dei campi e json

Di seguito vengono riassunte le regole per far funzionare correttamente i campi e organizzare i json generati:

1. Il property name deve essere UNICO, nessun campo può avere lo stesso property name di un altro campo;
2. I componenti di layout, come i panel, avranno un property name di questo tipo: {{tipoElemento}} {{NomePagina}}, ad esempio "panelDichiarazioni" o "pagePrivacy". Questo property name NON farà parte del json finale dell'istanza;

3. Il bottone SALVA BOZZA deve avere un property name di questo tipo: salvaBozza.{{nomePagina}}, ad esempio “salvaBozza.dichiarazioni”. In questo modo i bottoni finiranno tutti in un unico oggetto nel json dell’istanza;
4. Il componente **Data Grid** spesso ha un json “sporco” perché si porta dietro le modifiche fatte ai campi al suo interno, senza aggiornare il codice json. Fare sempre dei test e accertarsi che il json di questo componente sia pulito;
5. Quando si assegnano dei property name con concatenazione, es. Dichiarante.nucleoFamiliare.nome, gli elementi della concatenazione che precedono l’ultimo elemento non devono essere assegnati a nessun campo (in questo caso, ad esempio, nessun campo potrà avere il property name “dichiarante” o “nucleoFamiliare”);
6. Quando si usa il componente Data Grid, i campi al suo interno dovranno avere un property name non concatenato: “nome” va bene, “famiglia.nome” non va bene;
7. se si vogliono caricare molti dati, come ad esempio vie/corsi/piazze ecc. dal servizio di popolamento toponomastica/vie, assicurarsi di spuntare l’opzione **“Disable limiting response”**, nel tab **“Data”** del componente. In questo modo vengono caricati tutti i dati senza un limite;
8. I campi i cui dati vengono caricati in base all’elemento selezionato in altri campi (es. Il campo “comune” che mostra i comuni della provincia selezionata in un altro campo) devono sempre avere l’opzione **“Refresh Options On”** attiva e impostata sul campo di partenza.

Refresh Options On 

 × 

9. Nei campi di Email e nei campi select di Stato, Provincia, Comune e Indirizzo, impostare sempre l’attributo “autocomplete” con il valore “nope”, per evitare autocompletamenti inesatti;
10. Non utilizzare il componente Table, in quanto le api dei campi non vengono inclusi nel csv;
11. I campi “file” dove l’utente può allegare dei documenti devono avere le seguenti limitazioni: Ogni file può pesare massimo 2MB ed essere di estensione .pdf, .jpg o .jpeg, salvo diversamente richiesto dal cliente