

MANUALE BUILDER

VERIFICHE E APPROVAZIONI

VERSIONE	REDAZIONE		CONTROLLO APPROVAZIONE		AUTORIZZAZIONE EMISSIONE	
	NOME	DATA	NOME	DATA	NOME	DATA
V05	S. Casciano	09.04.2024	A.Deiro	09.04.2024		
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
V05	Tutto il documento	Revisione del documento.
V04	Paragrafo 8 + vari	Aggiunto paragrafo su bug nel json del Data Grid component. Aggiunta sezione "8. Regole e best practice"
V03	Revisione intero documento	Revisione intero documento
V02	Paragrafo 4.1	Inserita la componente Campo Testò.
V01	Tutto il documento	Versione iniziale del documento.

INDICE

1	SCOPO E RIFERIMENTI DEL DOCUMENTO	4
1.1	SCOPO DEL DOCUMENTO	4
2	GLOSSARIO	4
3	FUNZIONALITÀ DEL SISTEMA	4
3.1	CREAZIONE DI UN MODULO	4
3.2	MODIFICA DI UN MODULO	5
3.3	CONFIGURAZIONE DELLA NUOVA CATEGORIA	5
4	STRUTTURA DI UN MODULO	6
4.1	AGGIUNGERE COMPONENTE IN UN MODULO	6
4.2	MODIFICARE COMPONENTE IN UN MODULO	7
4.3	SPOSTARE LA COMPONENTE IN UN MODULO	7
4.4	COPIA / INCOLLA LA COMPONENTE IN UN MODULO	8
4.5	NOME DELLE PROPRIETÀ (PROPERTY NAME)	8
4.6	LE IMPOSTAZIONI DELLE COMPONENTI	9
5	COMPONENTI DI BASE	26
5.1	CAMPO TESTO (TEXT_FIELD)	26
5.2	CAMPO AREA DI TESTO (TEXT AREA)	27
5.3	CAMPO NUMERO (NUMBER)	28
5.4	PASSWORD (NO UTILIZZATO)	29
5.5	CAMPO CONTROLLO (CHECKBOX)	29
5.6	SELEZIONA CASELLA (SELECT BOX)	29
5.7	SELEZIONARE (SELECT)	31
5.8	RADIO (RADIO)	31
5.9	PULSANTE (BUTTON)	31
6	COMPONENTI AVANZATE	31
7	COMPONENTI VISUALIZZAZIONE	32
7.1	PANNELLO (PANNEL)	32
7.2	SELEZIONE	33
7.3	SELECT BOXES COMPONENT	36
7.4	DATA GRID	37
7.5	EDIT GRID	38
8	CONTROLLI DI VALIDAZIONE	40
8.1	CONTROLLI AUTODEFINITI	40
8.2	CONTROLLI BASATI SU CODICE JAVASCRIPT	41
8.3	CONTROLLO SULL'ESTENSIONE E SUL CONTENUTO DELL'ALLEGATO	47

9	SERVIZI DI POPOLAMENTO	49
10	MAPPING DATI PER INIZIALIZZAZIONE.....	51
10.1	INIT DA AUTENTICAZIONE	51
10.2	INIT DA ANPR	51
11	REGOLE E BEST PRACTICE	53
11.1	USER EXPERIENCE.....	53
11.2	DESIGN.....	55
11.3	FUNZIONAMENTO DEI CAMPI E JSON.....	55

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
OP_ADV	Operatore avanzato che agisce su istanze e configurazioni e non ha obbligo di impostare i filtri
OP_SIMP	Operatore semplice che può gestire le istanze ma non agisce sulle configurazioni

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 avanzato. Se si possiede tale profilo nel menù del backoffice sarà visibile la voce “moduli” dove sarà possibile visualizzare l'elenco dei moduli a cui si è abilitati e soprattutto avere a disposizione il tasto “Nuovo modulo” che serve per creare nuovi moduli.

3.1 Creazione di un modulo

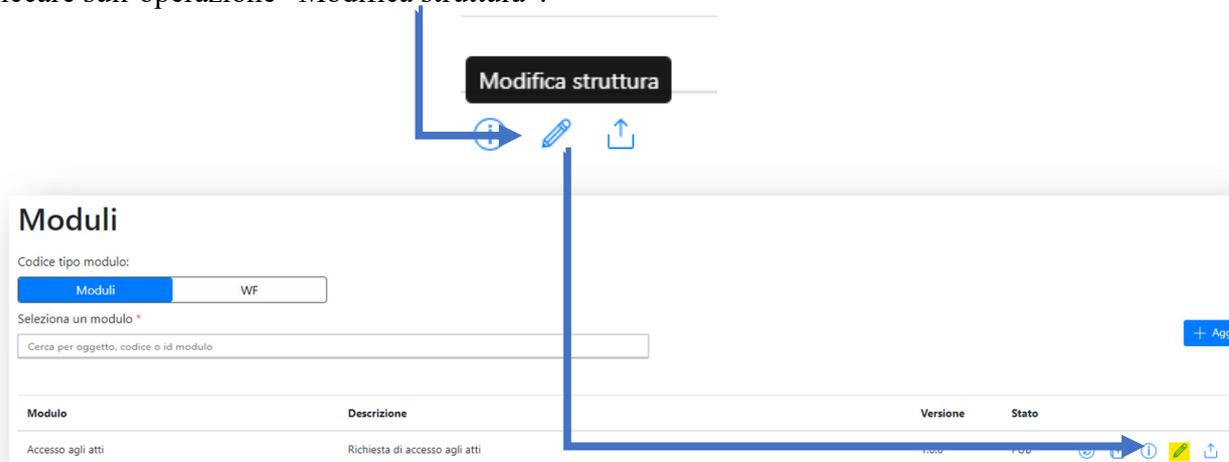
Quando si crea un nuovo modulo vengono richiesti alcuni parametri base obbligatori:

PARAMETRO	DESCRIZIONE
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
DESCRIZIONE	Breve descrizione del modulo
AREA	Rappresenta l'area utente in cui racchiudere tutte le categorie
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 (inviando una richiesta al gruppo di supporto.moon@csi.it)</u>

PARAMETRO	DESCRIZIONE
VERSIONE	Lasciare sempre la versione iniziale 1.0.0
TIPO DI FORM	Sono 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

La modifica di un modulo è possibile solo dopo averlo prima creato e inserite delle componenti. Per procedere alla modifica bisogna entrare nella funzione MODULI e dall'elenco dei moduli cliccare sull'operazione "Modifica struttura":



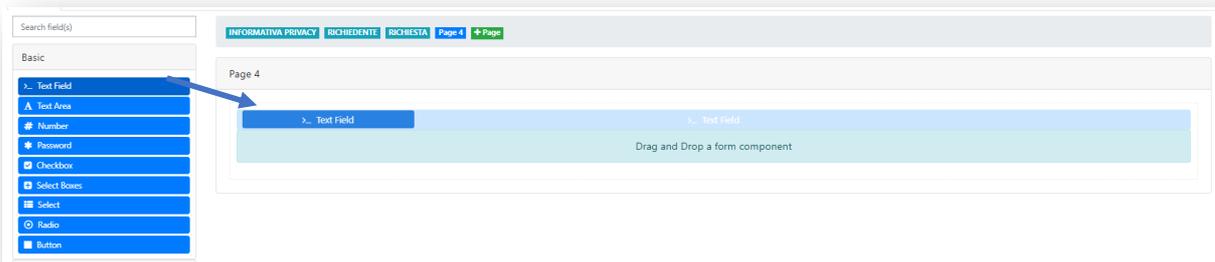
3.3 Configurazione della nuova categoria

Al momento non è possibile creare in autonomia la categoria, ma occorre inviare una mail alla casella supporto.moon@csi.it 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.



Una volta aggiunta la componente nel modulo, si abilitano delle icone:

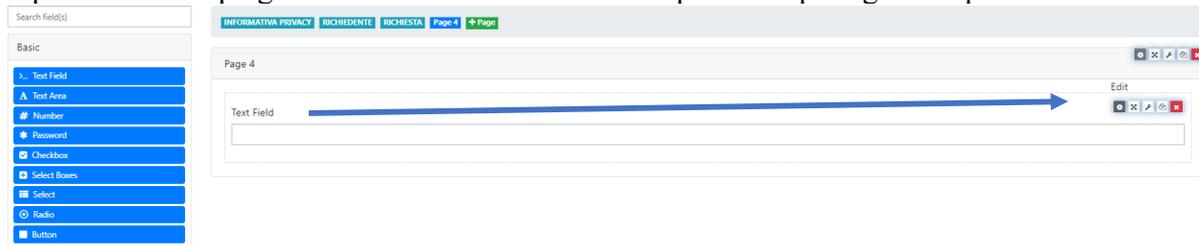


Queste icone consentono di effettuare delle operazioni sulla componente aggiunta sul modulo:

	questa operazione consente di modificare le impostazioni della componente;
	questa operazione consente di spostare la posizione di una componente all'interno del modulo;
	questa operazione consente di gestire la struttura json della componente (questa è consigliabile che sia utilizzata da un operatore che ha competenze avanzate di programmazione);
	questa operazione consente di duplicare la componente copiando e incollando all'interno del modulo;
	questa operazione consente di eliminare la componente.

4.2 Modificare componente in un modulo

Per modificare una componente di un modulo, bisogna 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.

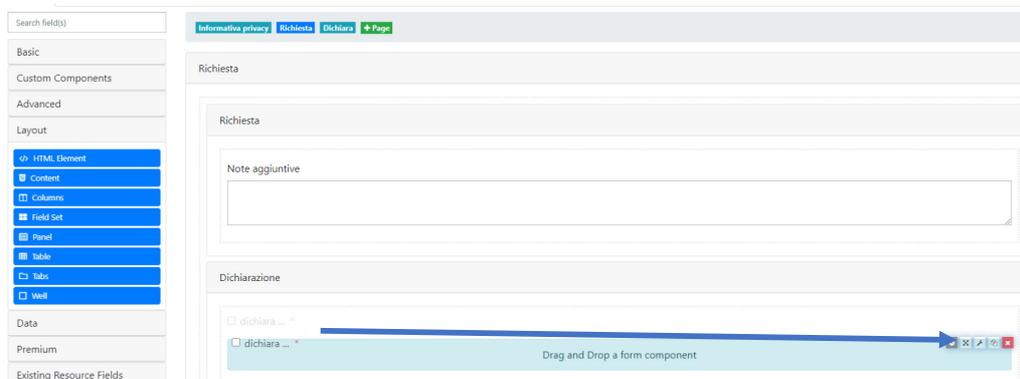


4.3 Spostare la componente in un modulo

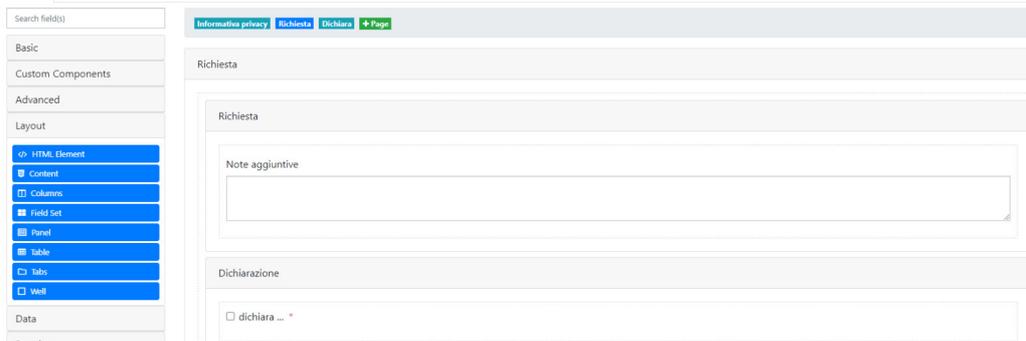
Per spostare una componente di un modulo, bisogna passare sopra alla componente e cliccare sull'icona dell'**incrocio**. Tenendo premuto con il tasto destro del mouse si può spostare la componente, ad esempio, da un pannello all'altro.



Spostando la componente nel pannello “Dichiarazioni”:



Si lascia il tasto destro del mouse nel pannello in cui viene spostata la componente:



4.4 Copia / incolla la componente in un modulo

Per copiare una componente e incollarla nel modulo, bisogna utilizzare l'icona del **copia**. Successivamente si abilita l'icona dell'**incolla** e cliccando si può incollare la componente copiata.

AVVERTENZA: una volta incollata la componente è necessario cambiare la `property_name` della componente copiata.



4.5 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)
Panel Component	Dati richiedente	datiRichiedente	panelDatiRichiedente	NO
Text Field	Cognome	cognome	richiedente.cognome	SI
Text Field	Nome	Nome	richiedente.nome	SI
Text Field (data)	Data di nascita	Data di nascita	richiedente.datINascita	SI
Radio Button	Persona giuridica o fisica?	personaGiuridicaOFisica	richiedente.tipoPersona	SI
Panel Component	Persona Fisica	personaFisica	panelPersonaFisica	NO
Text Field	Cognome	Cognome	richiedente.personaFisica.cognome	SI
Text Field	Nome	nome	richiedente.personaFisica.nome	SI
Panel Component	Persona Giuridica			persona.giuridica
Text Field	Cognome			persona.giuridica.cognome
Text Field	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"

4.6 Le impostazioni delle componenti

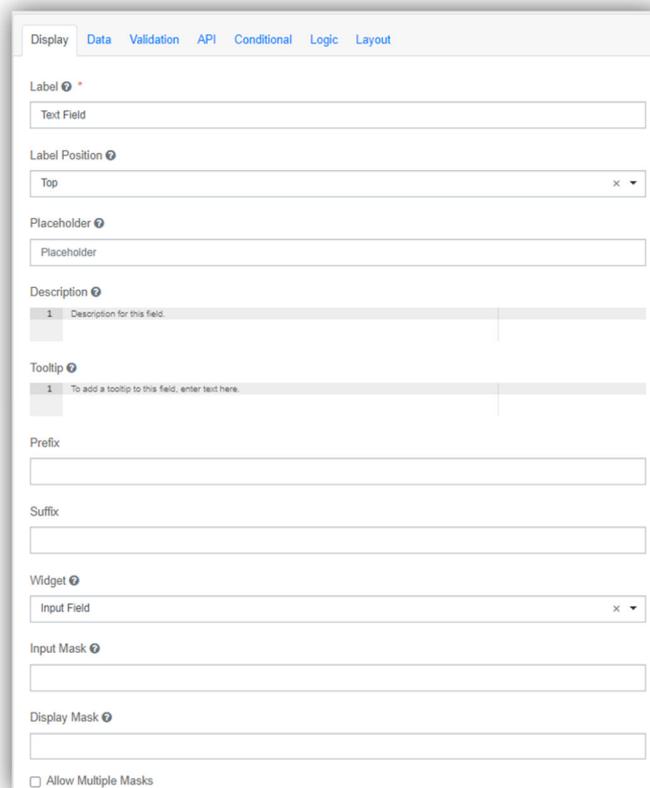
Prima di conoscere i vari componenti è importante comprendere le impostazioni alla base di questi componenti. Sebbene molti componenti offrano impostazioni univoche specifiche per il tipo di componente, esistono molte impostazioni che si estendono alla maggior parte dei componenti.

Queste impostazioni sono denominate **Impostazioni generali**. Di seguito è riportato un elenco di queste impostazioni organizzate per tipo di impostazione.



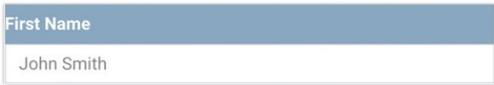
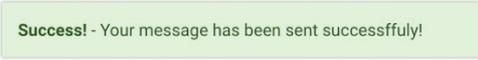
4.6.1 DISPLAY (schermo)

La scheda impostazioni display sarà la prima scheda aperta quando si modificano le impostazioni del componente.



NAME – DIPLAY	NOME - DIPLAY	DESCRIZIONE	NOTE / ESEMPI
Label	Etichetta	Il nome front-end del componente. Per impostazione predefinita, l'etichetta verrà visualizzata nella parte superiore sinistra del campo.	

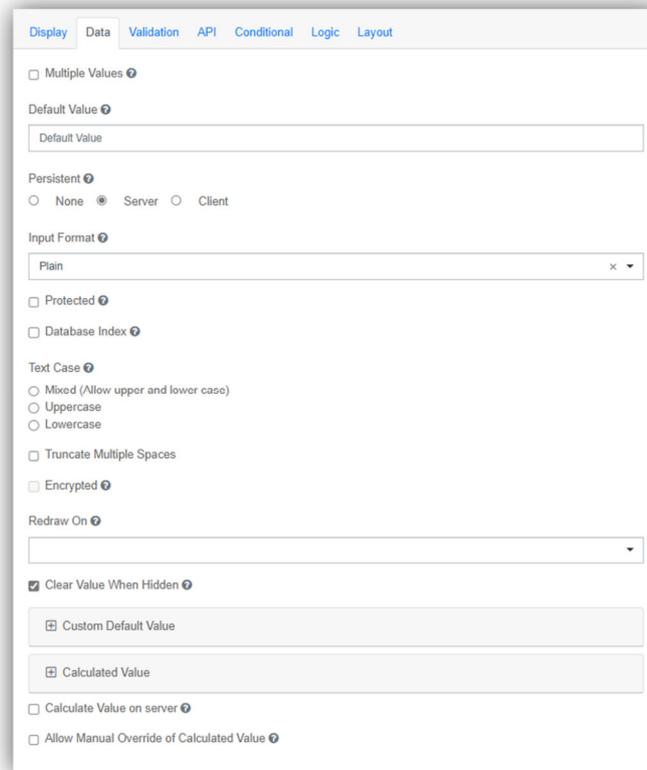
NAME – DIPLAY	NOME - DIPLAY	DESCRIZIONE	NOTE / ESEMPI
Label Position	Posizione dell'etichetta	La posizione dell'etichetta del componente. La Posizione etichetta verrà impostata automaticamente sull'opzione In alto, ma è possibile selezionare diverse altre opzioni:	
		- In alto : l'etichetta verrà visualizzata allineata a sinistra nella parte superiore del componente	
		- Sinistra (allineato a sinistra) : l'etichetta verrà visualizzata a sinistra del componente e allineata a sinistra dell'intervallo di larghezza del componente.	
		- Sinistra (allineata a destra) : l'etichetta verrà visualizzata a sinistra del componente e allineata a destra dell'intervallo di larghezza del componente.	
		- Destra (allineata a sinistra) : l'etichetta verrà visualizzata a destra del componente e allineata a sinistra dell'intervallo di larghezza del componente.	
		- Destra (allineata a destra) : l'etichetta verrà visualizzata a destra del componente e allineata a sinistra dell'intervallo di larghezza del componente.	
		- In basso : l'etichetta verrà visualizzata allineata a sinistra sotto il componente	
Label Width & Margin	Larghezza e margine dell'etichetta	Quando vengono selezionate le opzioni di allineamento. Posizione etichetta sinistra o destra, le impostazioni della larghezza e del margine dell'etichetta diventeranno disponibili. Queste opzioni consentono di aggiungere il riempimento alla rispettiva posizione del componente che creerà spazio tra l'etichetta e il componente. I valori di larghezza e margine si basano sulla percentuale di estensione del campo nel modulo.	
		Utilizza la posizione dell'etichetta quando formatti il modulo in modo simile a una griglia o per risparmiare spazio verticale sul modulo.	
Placeholder	Segnaposto	Un breve testo descrittivo che descrive il valore previsto che l'utente deve inserire nel campo. Il segnaposto verrà visualizzato quando il campo è vuoto e non verrà inviato al server quando il modulo viene inviato. Utilizza il segnaposto come testo di aiuto per gli utenti che compilano il campo.	
Description	Descrizione	Aggiungi una descrizione che verrà visualizzata sotto il campo. Questa impostazione può essere utilizzata in alternativa al segnaposto o in aggiunta se sono necessari ulteriori dettagli per il campo. Utilizza HTML nel campo delle impostazioni per definire lo stile del contenuto della descrizione	

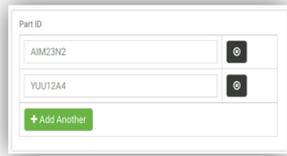
NAME – DIPLAY	NOME - DIPLAY	DESCRIZIONE	NOTE / ESEMPI
Tooltip	Descrizione comando	L'aggiunta di contenuto all'impostazione Descrizione comando aggiungerà un nuovo elemento dell'interfaccia utente al campo rappresentato da un'icona di domanda accanto all'etichetta del campo. Il contenuto all'interno dell'impostazione Descrizione comando verrà visualizzato quando un utente passa il mouse sopra l'elemento dell'interfaccia utente. Le descrizioni comando sono un ottimo modo per fornire istruzioni dettagliate o suggerimenti di aiuto senza occupare spazio nel modulo. Utilizza HTML nel campo di impostazione della descrizione comando per definire lo stile del contenuto all'interno della descrizione comando.	
Prefix	Prefisso	L'impostazione Prefisso aggiunge testo o caratteri prima del campo di input. Questo può essere utilizzato per classificare o etichettare un campo, come utilizzare \$ per la valuta.	
Suffix	Suffisso	L'impostazione Suffisso aggiunge testo o caratteri dopo il campo di input. Questo può essere utilizzato per classificare o etichettare un campo, come utilizzare % per rappresentare una percentuale.	
Custom CSS Class	Classe CSS personalizzata	Applica uno stile e una formattazione unici ai campi aggiungendo classi CSS personalizzate. Form.io utilizza Bootstrap come framework CSS predefinito per il renderer. Di seguito sono riportati degli esempi che mostrano alcune diverse classi CSS personalizzate che potresti trovare utili e il codice della classe.	
	Sfondo del campo, colore del testo	bg-info text-white	
	Avvisi, riempimento	alert-success p-3	
	Riposiziona campi	float-right,float-left	
Tab Index	Indice delle schede	Configura l'attributo Indice schede di questo componente per sovrascrivere l'ordine di tabulazione del modulo. L'impostazione dell'ordine di tabulazione per tutti i componenti del modulo ti dà il controllo sull'intero ordine di tabulazione del modulo.	
Auto Complete	Completamento automatico	Abilitati per impostazione predefinita, i campi di input popoleranno automaticamente i dati dai valori precedentemente immessi dall'utente dettati dalla cache del browser dell'utente. Immettere il valore <i>di disattivazione</i> nel campo di impostazione per disabilitare.	Default è a ON

NAME – DIPLAY	NOME - DIPLAY	DESCRIZIONE	NOTE / ESEMPI
Hidden	Nascosto	I componenti nascosti non vengono visualizzati nel modulo ma fanno comunque parte del JSON del modulo e dell'invio. Questa impostazione può essere utile quando si calcolano dati che non dovrebbero essere visibili all'utente front-end	
Hide Label	Nascondi etichetta	Nasconde l'etichetta di un campo. Risparmia spazio verticale su un modulo nascondendo l'etichetta del campo e utilizzando altre impostazioni di visualizzazione come un segnaposto per identificare il campo.	
Hide Input	Nascondi input	Maschera l'input del campo sostituendo i caratteri con caratteri circolari spesso visualizzati nei campi password. Utilizza questa impostazione quando gestisci dati sensibili come l'ID fiscale o il numero di previdenza sociale.	
Initial Focus	Focus iniziale	Quando viene eseguito il rendering del modulo, il componente con il focus iniziale selezionato sarà l'elemento focalizzato sul modulo. Solo un componente del modulo deve avere l'impostazione Focus iniziale abilitata. Se più di un componente del modulo ha l'impostazione Focus iniziale selezionata, solo l'ultimo componente nella struttura JSON con l'impostazione abilitata verrà focalizzato sul modulo.	
Disabled	Disabilitato	L'impostazione Disabilitato impedirà agli utenti di concentrarsi e inserire dati nel campo. Utilizzare i campi disabilitati insieme ai valori calcolati per impedire a un utente di alterare i calcoli o i valori popolati. I campi disabilitati verranno sempre visualizzati in grigio.	
Table View	Visualizzazione tabella	Quando l'impostazione Visualizzazione tabella è selezionata, il campo e i dati sottostanti verranno visualizzati nella griglia dei dati di invio del modulo. "I dati vengono sempre salvati sul server, indipendentemente dal fatto che l'impostazione sia abilitata o meno.	
Modal Edit	Modifica modale	Quando questa impostazione è abilitata, il campo viene sostituito con un pulsante che, quando cliccato, avvia una finestra modale per inserire e modificare i dati del campo. Abilitando questa impostazione un componente di layout conterrà una moltitudine di campi all'interno di un'unica finestra modale.	

4.6.2 DATA (Dati)

Le impostazioni dei dati riguardano principalmente il modo in cui i dati vengono configurati, interagiscono in relazione ad altri dati di campo o archiviati sul server. Si utilizza l'impostazione dati per eseguire operazioni come l'impostazione di un valore predefinito o la configurazione del calcolo dei dati.



NAME - DATA	NOME - DATI	DESCRIZIONE	NOTE / ESEMPI
Multiple Value	Valore multiplo	L'impostazione Valore multiplo introduce un pulsante <i>Aggiungi un altro</i> sotto il campo e consente all'utente di aggiungere o rimuovere più righe di un componente. I dati vengono salvati come array all'interno dell'invio.	
Default Value	Valore di default	Imposta il valore di input predefinito per il campo che verrà visualizzato nel campo quando viene eseguito il rendering del modulo. Se lasciati invariati, i dati del valore predefinito verranno salvati con l'invio.	
Persistent	Persistente	Questa impostazione determinerà il modo in cui verranno archiviati i dati del campo. Di seguito sono riportate le seguenti opzioni per l'impostazione Persistente. L'impostazione predefinita è Server:	Di default si utilizza il Server
		Server : il front-end invierà i dati al backend e il backend memorizzerà il valore	

NAME - DATA	NOME - DATI	DESCRIZIONE	NOTE / ESEMPI
		Solo client : il front-end NON invierà i dati al backend e il backend NON memorizzerà il valore. Ciò può essere utile quando il server deve intercettare dati per flussi di lavoro come logica o calcoli, ma non è necessario o desiderato che i dati vengano archiviati.	
		Nessuno : il front-end invierà i dati al backend, ma il backend NON memorizzerà il valore. Utilizzare questa impostazione se nel modulo è necessaria la logica lato client senza la necessità di inviare o salvare i dati sul server.	
Protected	Protetto	Questa impostazione impedirà che i dati del campo vengano interrogati da una chiamata API. Controlla questa impostazione se è necessario un ulteriore livello di protezione per i dati sensibili.	
Database Index	Indice della banca dati	Indicizzare il campo all'interno del database per aumentare le prestazioni lato server durante l'esecuzione di query sugli invii.	
Truncate Multiple Spaces	Tronca più spazi	Rimuove eventuali spazi non necessari tra parole o caratteri all'interno del campo.	
Encrypted	Crittografato	Aggiunge un livello di crittografia lato server al campo. Si tratta di una crittografia bidirezionale non adatta alle password.	Non viene utilizzato
Redraw On	Ridisegna attivo	Aggiorna il componente quando si verifica una modifica specifica. Il menu a discesa delle impostazioni di Ridisegna offrirà opzioni per ridisegnare il componente quando si verifica una modifica sul modulo o quando un campo specifico del modulo cambia selezionabile nel menu a discesa delle impostazioni. Utilizzare questa impostazione quando si interpolano i dati per calcoli o logica quando si verificano modifiche dinamiche del modulo.	
Clear Value When Hidden	Cancella valore quando nascosto	Cancella il valore del campo se il campo viene nascosto nel modulo. Questa impostazione è attiva per impostazione predefinita ed è il tipico comportamento del valore che si avrebbe quando si mostrano/nascondono i componenti. Deselezionare questa impostazione se il valore del campo deve essere mantenuto e successivamente visualizzato nuovamente quando si utilizza la logica condizionale	
Custom Default Value	Valore predefinito personalizzato	Espandi la scheda Valore predefinito personalizzato e scrivi Javascript personalizzato o Logica JSON per impostare il valore predefinito del campo.	

NAME - DATA	NOME - DATI	DESCRIZIONE	NOTE / ESEMPI
Calculated Value	Valore calcolato	Espandi la scheda Valore predefinito personalizzato e scrivi una logica Javascript o JSON personalizzata per impostare la logica di calcolo sul campo.	
Calculate Value On Server	Calcola valore sul server	Controlla questa impostazione se desideri eseguire i calcoli sul lato server. Ciò è utile se desideri sovrascrivere i valori inviati sul back-end con i calcoli eseguiti sul server.	
Allow Manual Override of Calculated Value	Consenti la sostituzione manuale del valore calcolato	Consente ai dati di input di sovrascrivere qualsiasi calcolo impostato sul campo. Per impostazione predefinita, il valore del campo non può essere immesso quando nel campo è impostato un calcolo. Controlla questa impostazione se desideri che gli utenti sovrascrivano il calcolo del campo e impostino il proprio valore.	

4.6.3 VALIDATION (Convalida)

Le impostazioni disponibili nella scheda Convalida si riferiscono direttamente alle convalide front-end e back-end configurabili per il campo. La convalida copre impostazioni quali campi obbligatori, dati univoci, requisiti min/max, convalide personalizzate e messaggi di errore personalizzati.

Display
Data
Validation
API
Conditional
Logic
Layout

Validate On ?

Change x ▾

Required ?

Unique ?

Minimum Length ?

Minimum Length

Maximum Length ?

Maximum Length

Minimum Word Length ?

Minimum Word Length

Maximum Word Length ?

Maximum Word Length

Regular Expression Pattern ?

Regular Expression Pattern

Error Label ?

Error Label

Custom Error Message ?

Custom Error Message

+ Custom Validation

+ JSONLogic Validation

+ Custom Errors

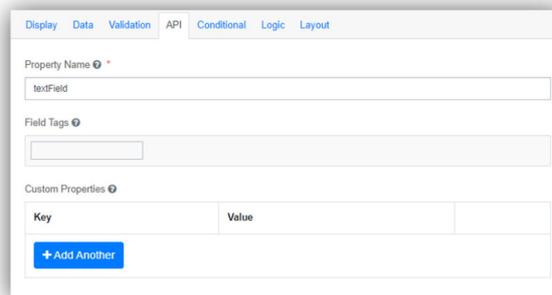
NAME - VALIDATION	NOME - CONVALIDA	DESCRIZIONE	NOTE / ESEMPI
Validate On	Convalida attivata	Determina quando avverrà la convalida front-end del campo in base all'opzione selezionata:	
Change		Modifica : la convalida verrà attivata attivamente ogni volta che si verifica una modifica nel campo. Questa è l'opzione predefinita per ogni campo.	
Blur		Sfocatura : la convalida verrà attivata solo quando il campo non è più a fuoco.	

NAME - VALIDATION	NOME - CONVALIDA	DESCRIZIONE	NOTE / ESEMPI
		Utilizza l'opzione Sfocatura sui campi che richiedono un determinato formato per garantire che la convalida non venga attivata finché l'utente non ha completato l'immissione dei dati. Campo e-mail EG	
Required	Necessario	L'impostazione Obbligatorio è rappresentata da un asterisco rosso accanto all'etichetta del campo e obbliga l'utente a immettere un valore prima che il modulo venga inviato o prima che l'utente passi alla pagina successiva di un modulo della procedura guidata.	
Unique	Unico	Al momento dell'invio, viene eseguita una query sul back-end per garantire che i dati immessi nel campo siano univoci. Applica questa impostazione a un campo E-mail in un modulo di registrazione per garantire che l'indirizzo immesso non sia stato inviato al database.	Non viene utilizzato
Validate When Hidden	Convalida quando nascosto	Quando questa impostazione è abilitata, il componente viene convalidato quando è nascosto o nascosto in modo condizionale. Gli errori di convalida dei componenti nascosti vengono visualizzati nell'avviso di errori del modulo quando si invia il modulo.	
Minimum / Maximum Length	Lunghezza minima/massima	Il requisito di lunghezza minima o massima che il valore del campo deve soddisfare. Questo comportamento cambia a seconda del componente a cui viene applicata l'impostazione.	
		Stringa (EG campo testo, area testo) : determinata dal numero di caratteri immessi nel campo	
		Numero - (Numero EG, Valuta) - Determinato dall'importo numerico immesso nel campo	
		Array (EG Data Grid, Edit Grid) - Determinato dal numero di array o righe aggiunte alla griglia	
		Utilizza la lunghezza minima/massima sulle griglie di dati o sulle griglie di modifica per determinare il numero di righe che un utente può aggiungere o rimuovere. Il valore minimo farà sì che la griglia riproduca quel numero specifico di righe quando viene visualizzato il modulo.	
Regular Expression Pattern	Modello di espressione regolare	Spesso definita Regex, questa impostazione consente di applicare un modello (o filtro) che descrive in dettaglio un insieme di stringhe che devono corrispondere al modello. Con un set di modelli Regex, la stringa deve corrispondere al modello prima che il modulo possa essere inviato.	

NAME - VALIDATION	NOME - CONVALIDA	DESCRIZIONE	NOTE / ESEMPI
Error Label	Etichetta di errore	Modificare il nome del campo visualizzato nel messaggio di convalida quando viene attivata la convalida. Utilizza l'etichetta di errore quando un campo ha un'etichetta o un nome lungo per abbreviare i messaggi di convalida.	
Custom Error Message	Messaggio di errore personalizzato	Scrivi un messaggio di errore personalizzato che sovrascriva il messaggio di convalida front-end predefinito visualizzato nel modulo quando viene attivata la convalida.	
Custom Validation	Convalida personalizzata	Espandi la scheda Convalida personalizzata e scrivi la logica Javascript o JSON per impostare le tue regole di convalida per il campo.	
Custom Errors	Errori personalizzati	Imposta diversi messaggi di errore personalizzati per più errori. Ad esempio, potresti impostare più messaggi di errore per comunicare un campo obbligatorio e una quantità minima/massima di caratteri (a differenza del "Messaggio di errore personalizzato", che ti consente di impostare solo un messaggio di errore per tutti gli errori).	

4.6.4 API

Le impostazioni dei componenti disponibili nella scheda API contengono impostazioni back-end, ad esempio il nome della proprietà del campo, nonché proprietà personalizzate che possono essere impostate, ad esempio HTML e tag.



NAME - VALIDATION	NOME - CONVALIDA	DESCRIZIONE	NOTE / ESEMPI
Property Name	Nome della proprietà	L'identificatore del campo back-end per l'endpoint API. Questo campo viene compilato automaticamente in camelCase in base all'etichetta iniziale fornita al campo. Il nome della proprietà di ogni campo deve essere univoco nel modulo prima che il modulo possa essere salvato. Utilizza il nome della proprietà quando fai riferimento al campo tramite API o logica condizionale.	
Field Tag	Etichetta di campo	Il tag di campo intrinsecamente non fornisce alcun tipo di funzionalità al campo, tuttavia, il tag può essere utilizzato o referenziato all'interno del livello dell'applicazione per facilitare diversi flussi di lavoro o utilizzato per attivare la logica del flusso di lavoro nel modulo.	Non viene utilizzato.
Custom Properties	Proprietà personalizzate	Applica proprietà personalizzate al campo.	Non viene utilizzato.

4.6.5 **CONDITIONAL (Logica e condizioni)**

Crea flussi di lavoro intelligenti utilizzando il sistema Condizioni e Logica di Form.io. Mostra o nascondi i campi, unisci lo schema JSON dei componenti, imposta le impostazioni dei campi in modo condizionale e altro ancora, il tutto in base all'interazione dell'utente nel modulo.

Esiste un'interfaccia utente condizionale Mostra/Nascondi senza codice utilizzando semplici campi a discesa. Questo elimina la necessità di scrivere codice JavaScript personalizzato quando il flusso di lavoro Mostra/Nascondi dipende da più campi o valori

Le condizioni vengono impostate in base al campo. Per configurare una condizione di visualizzazione/nascondimento, aprire le **impostazioni** del campo del componente a cui si desidera applicare la condizione e fare clic sulla **scheda Condizionale**

Esistono due metodi diversi per scrivere le condizioni, **Semplice** e **Avanzato**

Quella **semplice** consente all'utente di mostrare o nascondere i campi in base all'input dell'utente o alla selezione del valore di altri campi nel modulo. Queste condizioni vengono configurate utilizzando un'interfaccia utente senza codice. Esaminiamo ciascuna impostazione e cosa fanno prima di entrare nelle opzioni del flusso di lavoro.

Display Data Validation API Conditional Logic Layout

Simple

This component should Display:

When the form component:

Has the value:

Advanced Conditions

- Mostra o nascondi questa componente: seleziona se desideri **mostrare** il campo o **nascondendolo** quando le condizioni sono soddisfatte
- Quando: seleziona se desideri mostrare o nascondere la componente quando vengono soddisfatte una o tutte le condizioni
- Valore: l'immissione dei dati del campo per eseguire la condizione. Quando il campo condizionale ha valori statici, come i componenti Radio o Seleziona, tali opzioni saranno disponibili all'interno di un campo a discesa da selezionare come Valore

La combinazione di tutte queste impostazioni determinerà quando la condizione verrà eseguita sul modulo. Questa interfaccia logica è completamente priva di codice e offre all'utente la possibilità di creare istruzioni if/then avanzate per mostrare o nascondere un componente in base all'interazione o alla risposta di uno o più componenti.

Quella **avanzato** sfrutta la potenza di JavaScript o JSON scrivendo codice personalizzato per mostrare o nascondere in modo condizionale i campi del modulo. Si utilizza questa opzione quando nel flusso di lavoro sono richieste condizioni più complesse o specifiche, come quando il condizionale si basa su una sottofunzione all'interno del codice Javascript che confronta valori diversi. Ciò offre all'utente la libertà e la flessibilità di scrivere condizioni complesse come meglio crede.

Il JSON Logic è utilizzabile come alternativa a Javascript durante la scrittura delle Condizioni Avanzate. Restano validi gli stessi concetti dettagliati nella sezione Condizioni Avanzate. Sostituisci le variabili e gli operatori Javascript con JSON Logic.

☰ Advanced Conditions

The following variables are available in all scripts.

form	The complete form JSON object
submission	The complete submission object.
data	The complete submission data object.
row	Contextual "row" data, used within DataGrid, EditGrid, and Container components
component	The current component JSON
instance	The current component instance.
value	The current value of the component.
moment	The moment.js library for date manipulation.
_	An instance of Lodash .
utils	An instance of the FormioUtils object.
util	An alias for "utils".

☰ JavaScript

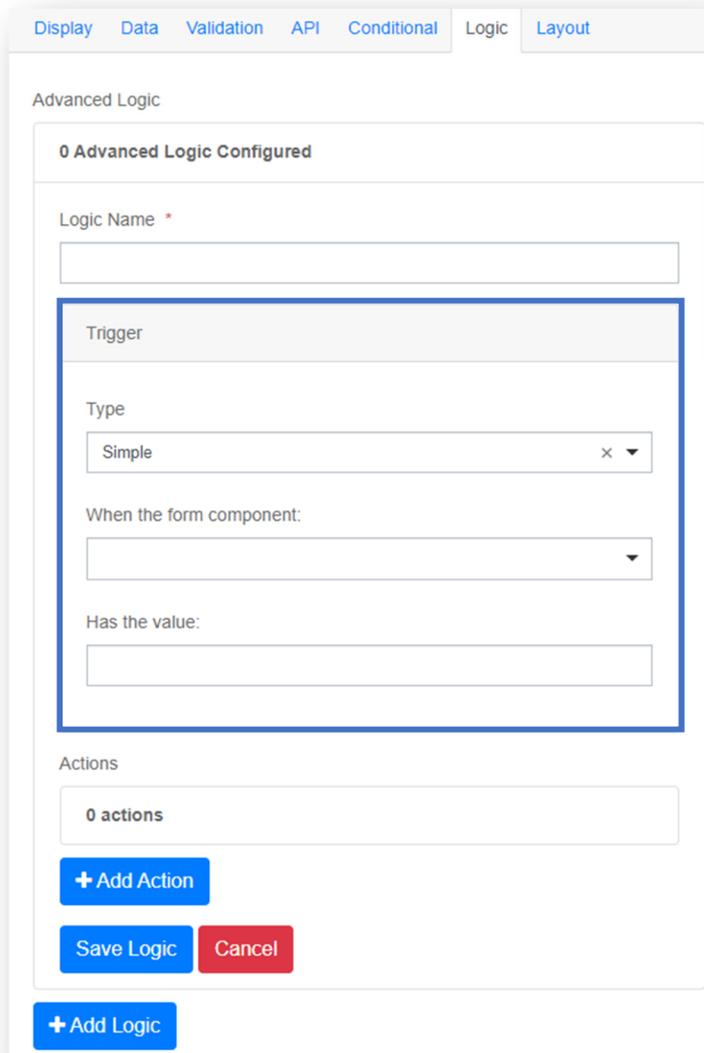
☰ JSONLogic

4.6.6 LOGIC (Logica)

La funzionalità Logica fornisce un motore di regole più affidabile per controllare e manipolare in modo condizionale le impostazioni dei campi, lo schema JSON, i valori dei dati e gli eventi personalizzati. La funzionalità Logic viene applicata in base al campo con la possibilità di aggiungere più istanze Logic al campo, se necessario.

Il sistema Logico consente inoltre all'utente di impostare più trigger e diversi rami di azioni condizionali su un singolo campo. Che tu stia impostando semplici condizioni uno a uno o creando flussi di lavoro complessi, la funzionalità Logic è disponibile per soddisfare tali esigenze.

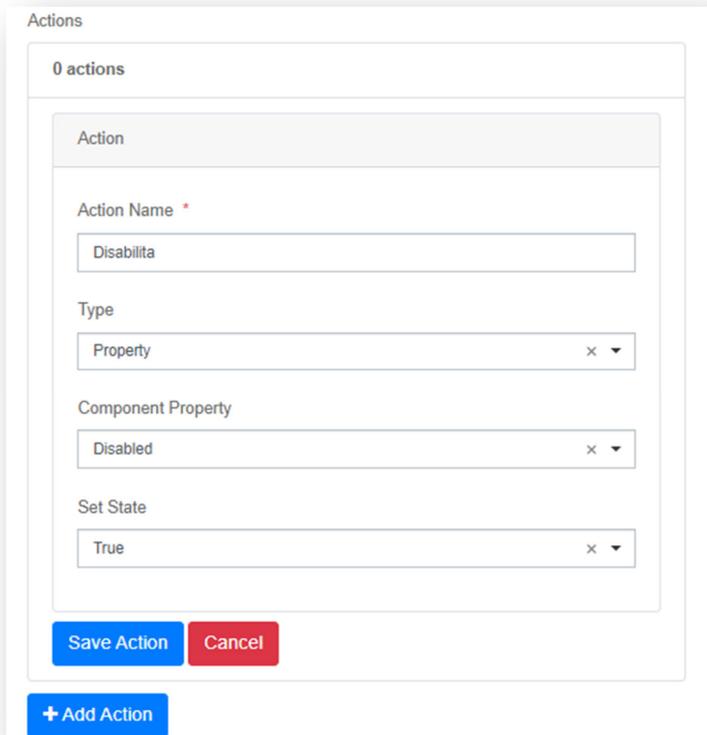
Per accedere alla funzionalità Logica, aprire le impostazioni dei componenti del campo condizionale e fare clic sulla scheda Logica. Fare clic sul pulsante **Aggiungi logica** per creare una nuova istanza logica. Assegna all'istanza Logic un nome che sia facilmente identificabile in futuro



La configurazione del trigger dipenderà dal tipo di trigger selezionato. I tipi di trigger sono elencati di seguito:

- **Semplice:** il tipo di trigger semplice utilizza la stessa interfaccia utente e i metodi di configurazione presenti nella scheda Condizionale.
- **Javascript:** scrivere i condizionali personalizzati utilizzando il codice Javascript. Il tipo di trigger Javascript utilizza la stessa interfaccia utente e i metodi di configurazione utilizzati per i condizionali avanzati.
- **Logica JSON:** utilizzare la logica JSON come alternativa a Javascript durante la scrittura delle condizioni logiche.
- **Evento:** utilizzare l'opzione Evento quando crei i tuoi eventi personalizzati sul modulo.

Ora che il nostro Trigger è impostato, si aggiunge un'azione che verrà eseguita quando la condizione viene attivata. Fai clic sul pulsante **Aggiungi azione**, assegna un nome all'azione e seleziona un **tipo** di azione. Ciò determinerà il tipo di azione che verrà eseguita quando la condizione viene attivata



È possibile scegliere tra quattro **tipi di azione**:

- **Proprietà:** modifica in modo condizionale le impostazioni del campo come Obbligatorio, Etichetta, Disabilitato e altro. A seconda della proprietà del componente selezionata, configurare lo **stato di impostazione** su **True** o **False** (ad esempio impostando un flag richiesto) o inserire la proprietà condizionale **Text** (ad esempio modificando la proprietà Label).
- **Valore:** modifica o compila i valori dei dati nel campo condizionale utilizzando Javascript. Sono disponibili le seguenti variabili da utilizzare nel codice Javascript: "riga", "dati", "componente" e "risultato".

- **Unisci schema componente:**

non tutte le impostazioni e configurazioni dei campi sono offerte con il tipo di azione Proprietà. Aggiorna direttamente il campo Definizione schema per configurare impostazioni aggiuntive utilizzando le condizioni logiche, modificare dinamicamente le etichette o le selezioni dei valori e popolare i dati interpolati da altri campi.

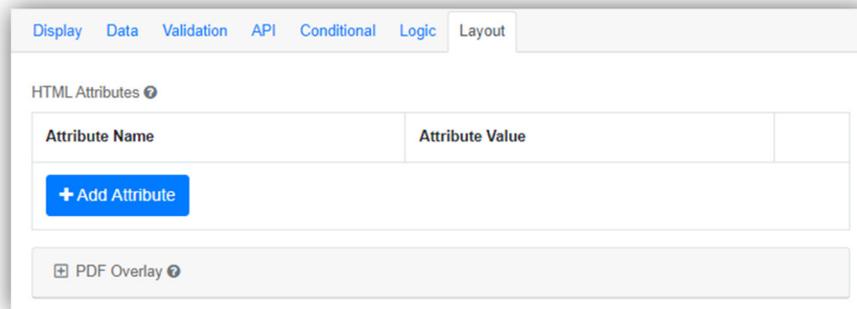
- **Azione personalizzata** Esegue un flusso di lavoro di eventi personalizzati. In genere coincide con altri campi come i componenti Campo nascosto, Origine dati e Pulsante.

Si possono inserire in una singola istanza di Logica una o più azioni oppure si possono inserire più istanze di Logica con una o più azioni.

N.B: l'azione che viene utilizzata è la Proprietà e il Valore. Mentre le altre azioni non state mai utilizzate.

4.6.7 LAYOUT (Impaginazione)

Applica attributi / proprietà HTML e riposiziona i campi PDF a livello di codice.



NAME - VALIDATION	NOME - CONVALIDA	DESCRIZIONE	NOTE / ESEMPI
HTML Attributes	Attributi HTML	Fornire una mappa di attributi HTML per l'elemento di input del componente. Gli attributi provenienti dalle impostazioni di altri componenti o altri attributi generati da form.io hanno la precedenza sugli attributi HTML configurati.	
PDF Overlay	Sovrapposizione PDF	Queste impostazioni consentono all'utente di controllare a livello di codice la posizione del campo sui primi moduli PDF nonché gli stili personalizzati. Utilizza i margini per ottimizzare le posizioni dei componenti di sovrapposizione sui moduli PDF.	Non viene utilizzato
		Stile: stili personalizzati applicati al componente durante il rendering del PDF.	
		Pagina: la pagina PDF su cui verrà visualizzato il componente.	
		Sinistra: il margine sinistro della posizione del componente all'interno del modulo PDF definito in pixel.	
		Superiore: il margine superiore della posizione del componente all'interno del modulo PDF definito in pixel.	
		Larghezza: la larghezza del componente in pixel. Altezza: l'altezza del componente in pixel.	
		Applica misurazioni di altezza identiche a tutti i componenti di sovrapposizione PDF per garantire un'altezza coerente su tutti i componenti di sovrapposizione sul modulo PDF.	

5 Componenti di base

I componenti di base si trovano nella parte superiore dei raggruppamenti di componenti all'interno del Generatore di moduli. Questi componenti sono più comunemente utilizzati nei moduli Web tradizionali e dovrebbero essere componenti familiari se in passato hai utilizzato un generatore di moduli. Puoi trovare informazioni per ciascuno dei componenti di base come impostazioni univoche, codice JSON ed esempi di campi.

Basic	
 Text Field	CAMPO TESTO: utilizzato per l'immissione di testo breve e generale
 Text Area	AREA DI TESTO: è un campo di input su più righe che consente testo più lungo
 Number	NUMERO: è un campo di input limitato a valori di tipo numerico
 Password	PASSWORD: è un campo di input con tipo HTML impostato su password
 Checkbox	CASELLA DI CONTROLLO: è una casella di controllo booleana
 Select Boxes	SELEZIONA CASELLA: è una casella di controllo con più opzioni
 Select	SELEZIONARE: sono valori selezionabili presentati in un menù a discesa
 Radio	RADIO: seleziona una singola opzione da un elenco di opzioni
 Button	PULSANTE: eseguono varie azioni all'interno del modulo

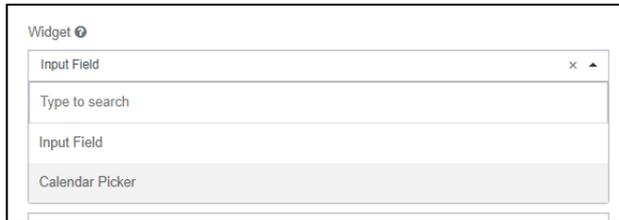
5.1 Campo Testo (Text_Field)

La componente di testo può essere utilizzata, 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.

Quando utilizzare il componente campo di testo?

- Se non è possibile prevedere la risposta di un utente a una richiesta ed esserci un'ampia variabilità nelle risposte degli utenti.
- Se gli utenti desiderano poter incollare una risposta

Impostazioni principali



Il **widget** è l'interfaccia utente di visualizzazione utilizzata per inserire il valore del campo. Per impostazione predefinita, il widget sarà impostato su un campo di input tradizionale, tuttavia, puoi modificare il widget per trasformare il campo di testo in un selettore di calendari



Maschera di input: utilizzare una maschera di input quando è richiesto un formato di input predefinito. Ciò è utile per forzare formati di input come numeri di telefono,

codici postali, SSN, ecc.

9: numerico
A: In ordine alfabetico
*: alfanumerico

Esempio di maschera Numero di telefono: (999) 999-9999.

5.2 Campo Area di Testo (Text Area)

Un'area di testo è un campo di input su più righe che consente testo più lungo. L'area di testo può anche essere convertita in un editor di contenuti WYSIWYG

Quando utilizzare il componente campo di testo?

- Utilizzare l'area di testo quando è richiesto l'inserimento di testo più grande
- Cambia l'area di testo in WYSIWIG per dare all'utente la possibilità di formattare e personalizzare l'input di testo.

Impostazioni principali



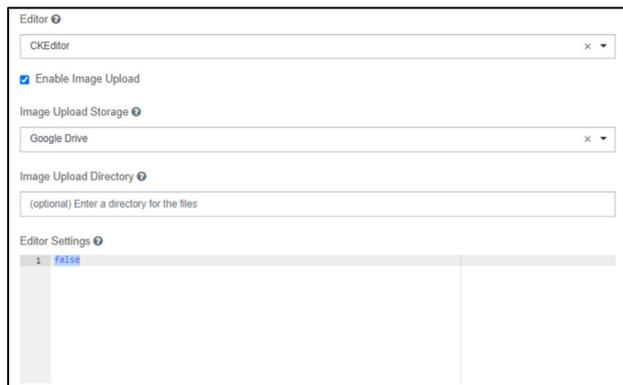
Righe: controlla quante righe predefinite sono impostate nell'area di testo



Espansione automatica: l'area di testo si espanderà automaticamente verticalmente mentre l'utente digita nel campo.



Editor: questa opzione convertirà l'area di testo in uno dei seguenti editor WYSIWYG



Abilita caricamento immagine: consente all'utente di caricare un'immagine nell'editor WYSIWYG

Impostazioni dell'editor: qui puoi modificare la base di codice dell'editor WYSIWYG per personalizzare l'interfaccia utente specifica in base alle tue esigenze.

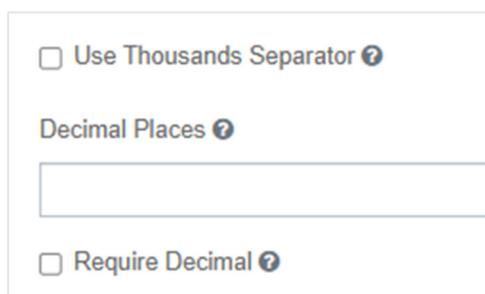
5.3 Campo Numero (Number)

Un campo numero può essere utilizzato per l'immissione di numeri e limitato a un tipo di numero. Sono disponibili opzioni per impostare i separatori delle migliaia, le posizioni decimali e i requisiti decimali.

Quando utilizzare il componente campo numero?

- Utilizzare il componente Numero quando si gestiscono tipi di input specifici per l'input numerico
- Utilizzare le impostazioni delle posizioni decimali per i requisiti numerici, ad esempio una percentuale che potrebbe richiedere l'inclusione di posizioni decimali specifiche nell'invio

Impostazioni principali



Usa il separatore delle migliaia: selezionare questa impostazione se si desidera che il valore di questo componente separi le migliaia in base a un delimitatore locale.

Decimali: Numero massimo di cifre decimali per i valori in questo campo.

Richiedi decimali: Mostra sempre i decimali per questo campo, anche se sono zeri finali.

Simbolo decimale: consente di modificare il simbolo decimale predefinito per il componente numero. **N.B:** Si tratta di un'impostazione nascosta che può essere configurata modificando il codice JSON del componente Numero. Aggiungere quanto segue allo schema: Copiare "decimalSymbol": "," . Dopo aver salvato le modifiche, il componente Numero cambierà il decimale predefinito nel simbolo designato. Utilizzatelo in combinazione con il delimitatore locale per ottenere diversi tipi di sistemi di numerazione.)

5.4 Password (NO UTILIZZATO)

Il campo password ha le stesse opzioni di un componente campo di testo. Si differenzia da un campo di testo in quanto il suo <input>tipo HTML sarà password anziché testo. Ciò farà sì che il campo visualizzi simboli di input nascosti invece del valore immesso.

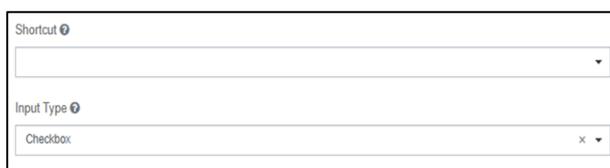
5.5 Campo Controllo (Checkbox)

È possibile utilizzare una casella di controllo per il campo di immissione del valore booleano. Può essere selezionato (vero) o deselezionato (falso). Sono disponibili opzioni per impostare la scelta rapida e il tipo di input.

Quando utilizzare il componente campo di controllo?

Non ci sono particolari indicazioni per utilizzare questa componente.

Impostazioni principali



Scorciatoia: è una scorciatoia da tastiera per selezionare/deselezionare il componente.

Tipo di ingresso: può essere modificato da una casella di controllo con un singolo valore di ingresso o da un componente radio multivalore.

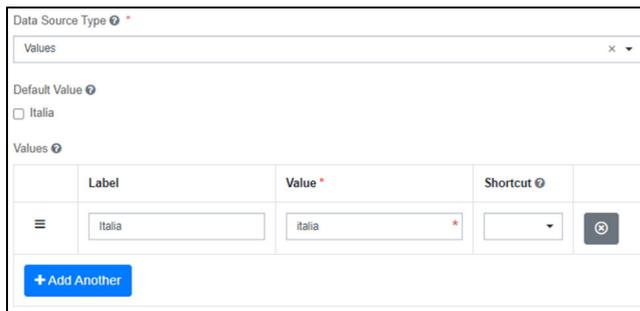
5.6 Seleziona casella (Select Box)

Questo componente multivalore consente agli utenti di selezionare una o più opzioni nel formato stile casella di controllo. Si possono impostare valori nella scheda Dati delle impostazioni e applicare impostazioni uniche come scorciatoie di valore e convalida del valore minimo/massimo.

Quando utilizzare il componente campo di controllo?

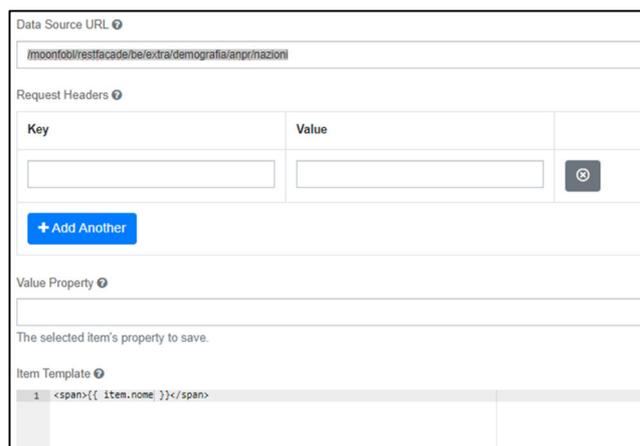
- Usa l'icona **dell'hamburger** per riorganizzare le opzioni.

Impostazioni principali



Tipo di origine dati: L'origine da utilizzare per le opzioni della casella di selezione:

- **Valori:** Creare un elenco personalizzato di valori statici per le opzioni Seleziona casella. L'etichetta è l'identificatore front-end per l'utente finale. Il valore è l'identificatore back-end per l'opzione ed è ciò che viene salvato nel database. Quando si imposta il **valore** dell'opzione, utilizzare solo Stringhe o Valori numerici



- **URL dell'origine dati:** designa un URL che fornisca i valori delle opzioni per l'elenco a discesa. Utilizzare il modello di elemento per determinare quale parte dell'oggetto dell'URL verrà utilizzata come valori dell'opzione.

Si supponga, ad esempio, di voler popolare la casella di selezione con un elenco di tutti gli Stati Uniti. È possibile utilizzare un URL JSON esterno come il seguente:

moonfobl/restfacade/be/extra/demografia/anpr/nazioni

Si inserisce la URL all'interno **dell'URL dell'origine dati**. Specificando una proprietà Value per rappresentare l'opzione che è ciò che verrà salvato all'interno dei metadati. Quando il valore viene lasciato vuoto, l'intero oggetto verrà salvato per l'opzione.



Numero min/max controllato: convalida che determina la quantità minima e massima di opzioni da controllare. Le opzioni selezionate al di sotto del minimo attiveranno la convalida front-end. Quando viene selezionato il numero massimo di opzioni, le opzioni rimanenti verranno disabilitate.

Messaggio di errore min/max controllato: Determina il messaggio di convalida visualizzato all'utente per l'importo minimo o massimo

dell'opzione.

5.7 Selezionare (Select)

Il componente Seleziona visualizza un elenco di valori in un elenco a discesa in cui gli utenti possono selezionare uno dei valori. Questo componente ha flessibilità sulla provenienza dell'origine dati. Esiste anche un'ampia offerta di impostazioni che forniscono diversi modi di filtrare, interrogare e caricare i valori dei dati.

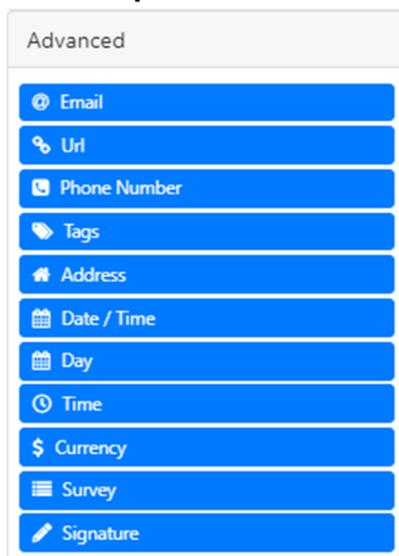
5.8 Radio (Radio)

Il componente radio è un campo che consente agli utenti di selezionare una singola opzione da un elenco di opzioni visualizzate in formato stile radio. Gli utenti possono fare clic su un'opzione selezionata per deselegionarla, riportando il campo al suo stato originale. consente

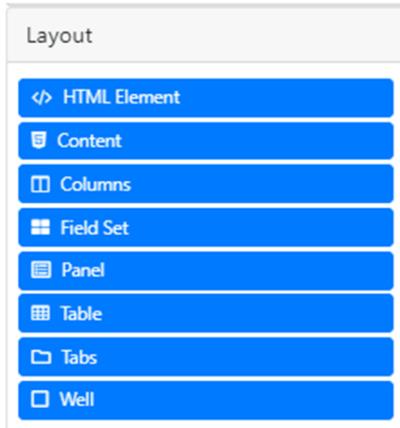
5.9 Pulsante (Button)

È possibile aggiungere pulsanti per eseguire varie azioni all'interno del modulo. La funzione più ovvia del componente Button è l'azione di invio. Tuttavia, puoi anche utilizzare il componente Button per attivare eventi associati alla logica del flusso di lavoro, reimpostare i dati dei campi, eseguire l'autenticazione presso un provider OAuth e altro ancora.

6 Componenti avanzate

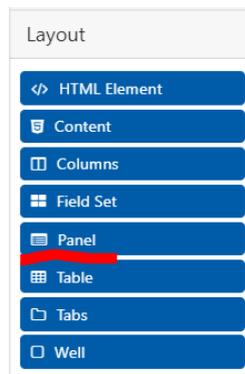


7 Componenti Visualizzazione



7.1 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:

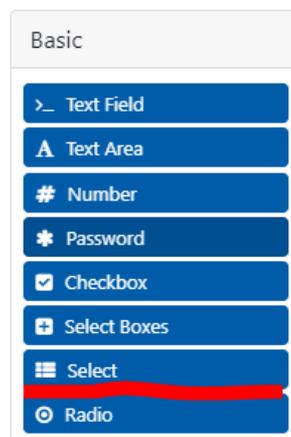
- Tema default: per il primo pannello (creato automaticamente con la page e non visualizzato nel front-end) e per i pannelli di secondo livello (cioè contenuti dentro altri panel);
- Tema info (azzurro): per il pannello subito sotto il primo (cioè il primo pannello mostrato nel front-end) e tutti gli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

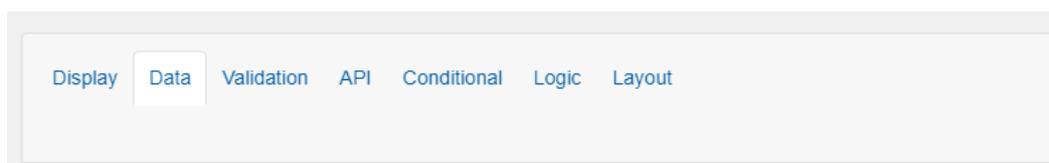
Assegnare al panel un property name di tipo “panel{{nomePagina}}”, ad esempio “panelDichiarante”.

7.2 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

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

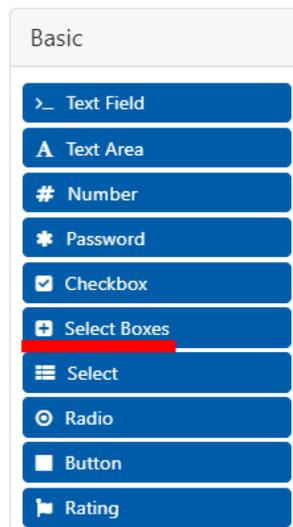
Quando le opzioni di scelta sono molte, ad esempio più di 10, si raccomanda di utilizzare il widget di tipo **“ChoicesJS”**, che consente di effettuare una ricerca testuale fra le opzioni disponibile, per rendere l'esperienza utente agevole.

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

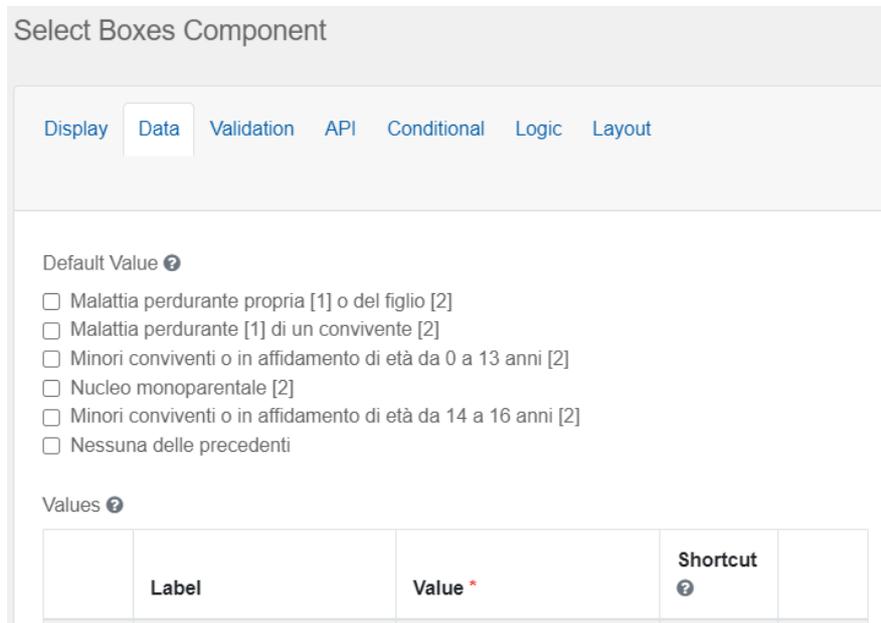
```
{"searchPlaceholderValue": "Scrivi il testo da cercare",  
"searchResultLimit": 10}
```

7.3 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**.



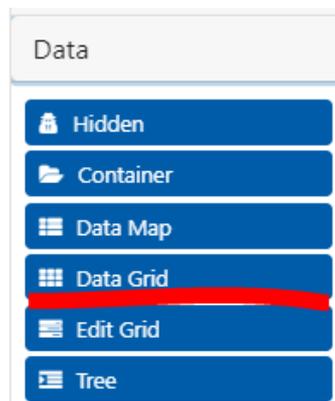
È possibile inserire gli elementi attraverso la seguente funzionalità:



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

7.4 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:

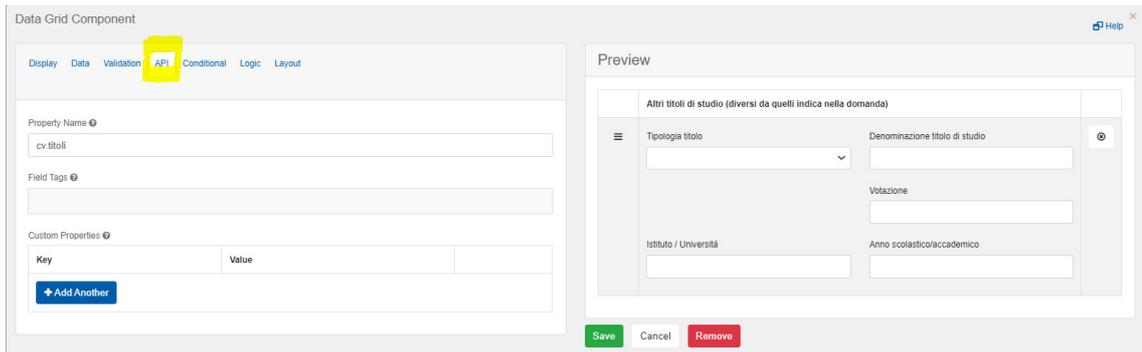
- la gestione di informazioni multiple, aggiungendo le stesse informazioni più volte;
- di riordinare le informazioni, per permettere una visione ordinata delle informazioni multiple richieste.

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:

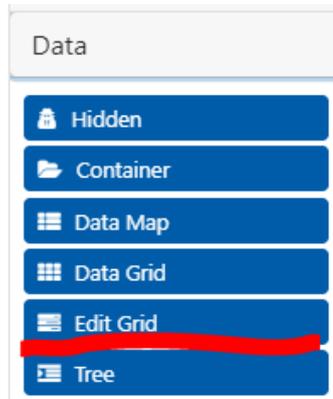
- Tipologia titolo → attributo semplice “tipologia”; se fosse composto e quindi fuori dal data grid sarebbe “cv.titolo.tipologia”;
- Denominazione titolo di studio → attributo semplice “denominazione”; se fosse composto e quindi fuori dal Data Grid sarebbe “cv.titolo.denominazione”.

! 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.

7.5 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>  
    {% 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>
```

8 Controlli di validazione

8.1 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”

8.2 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

8.2.1 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 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        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 = "Formato non corretto";
        }
        else {
            result = true;
        }
    }
}
valid = result;
```

8.2.2 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,");
            codice_cognome += cognome.replace(/^[AEIOU]/gi,");
            codice_cognome+='XXX';
            codice_cognome=codice_cognome.substr(0,3);
            var strCognome = codice_cognome.toUpperCase();

            if (strCognome != cf.substring(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.substring(1,2);
tavola_mesi=['A','B','C','D','E','H','L','M','P','R','S','T'];
cf.substring(9,11)-40 : cf.substring(9,11);
cf_mese="+cf.substring(8,9) + " letteraMese="+tavola_mesi[parseInt(indMese)];
? "F" : "M";
fiscale non è compatibile con il sesso indicato";
fiscale occorre indicare il sesso";
}
else {
    mese = dataNascita.substring(3,5);
    var numMese = (mese > 9 )? mese :
    var indMese = numMese -1;
    anno = dataNascita.substring(8,10);
    if (anno!=cf.substring(6,8)) {
        result = 'errore anno' + msg;
    }
    var
    if (tavola_mesi[indMese]!=cf.substring(8,9)) {
        result = 'errore mese' + msg;
    }
    var giornoCF = (cf.substring(9,11)>31) ?
    if (giornoCF != giorno) {
        result ='errore giorno' + msg;
    }
    // alert(anno+"-"+indMese+"-"+giorno+ "
    if (result === 'true') {
        if (sesso !== 'undefined') {
            var sessoCF = (cf.substring(9,11)>31)
            if (sessoCF !== sesso){
                result = "Attenzione: il codice
            }
        }
        else {
            result = "Prima di inserire il codice
        }
    }
}
}
else {
    result = "Prima di inserire il codice fiscale occorre indicare il
```

```
nome";
    }
    }
    else {
        result = "Prima di inserire il codice fiscale occorre indicare il cognome";
    }
}
if (result === 'true') {
    var validi, i, s, set1, set2, setpari, setdisp;
    var msgErrFormat = "Il codice fiscale indicato non è valido";
    validi = "ABCDEFGHJKLMNOPQRSTUVWXYZ0123456789";
    for( i = 0; i < 16; i++){
        if( validi.indexOf( cf.charAt(i) ) == -1 )
            result = msgErrFormat;
    }
    set1 = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    set2 = "ABCDEFGHIJABCDEFGHIJABCDEFGHIJABCDEFGHIJ";
    setpari = "ABCDEFGHIJABCDEFGHIJABCDEFGHIJABCDEFGHIJ";
    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;
```

8.2.3 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 = "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKL";
    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 = "Formato non corretto";
    }
    else {
        result = true;
    }
}
}
else {
    if( cf.length < 11 || cf.length > 11 ) {
        result = "Formato non corretto";
    }
    else {
        if ( /^[0-9]{11}/.test(cf) ) {
            result = "Formato non corretto";
        }
        else {
```

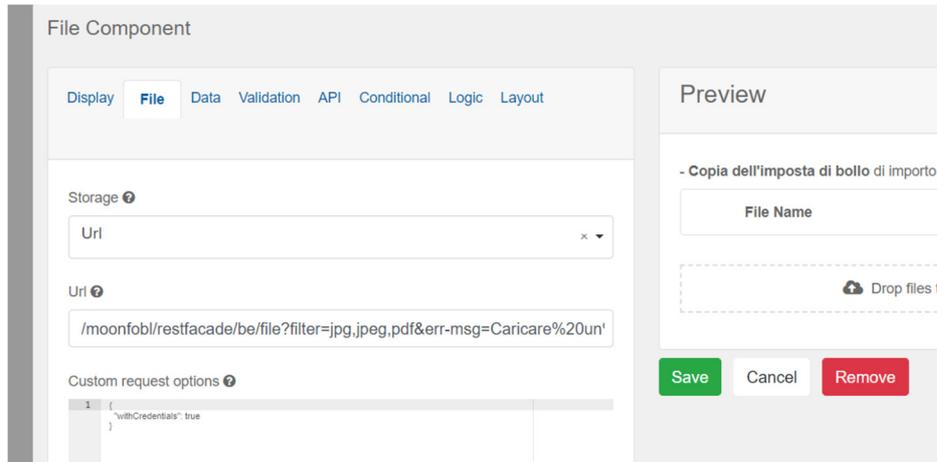
```
        result = true;
    }
}
}
}
valid = result;
```

8.2.4 Controllo su numero di telefono

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

8.3 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&err-msg=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`

9 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 “{{ item.NomeParametro }}” dove NomeParametro è indicato nella relativa colonna. Il valore per esempio potrà essere “{{ item.nome }}”. 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	{{ item.nome}} {{ item.sigla}}
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	{{ item }}
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 }}`/ .

10 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.

10.1 Init da autenticazione

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

10.2 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

11 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.

11.1 User Experience

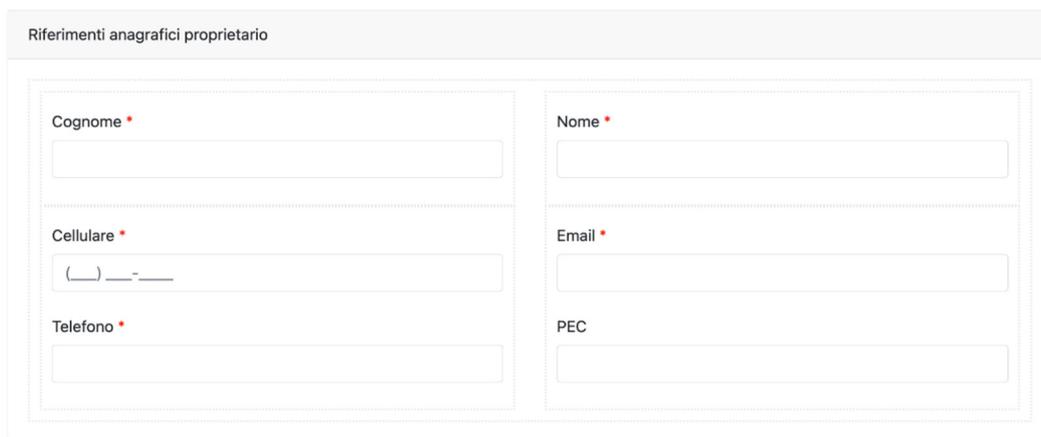
Nel progettare un modulo, anche molto semplice, tenere sempre a mente le necessità di chi dovrà presentare l'istanza, che potrebbe incontrare vari ostacoli nel compilare il modulo (ad esempio scarsa conoscenza della lingua italiana, poca dimestichezza con gli strumenti informatici, conoscenza limitata

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:

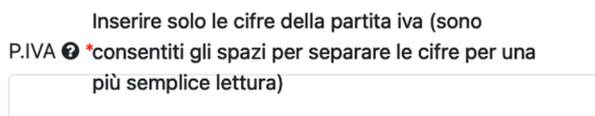
1. Chiedere sempre al cliente una descrizione degli attori coinvolti (se possibile);
2. Assegnare etichette comprensibili e inequivocabili ai vari campi (ad esempio, "Provincia" anziché "Prov");
3. Nei campi dove bisogna inserire una **data**, nelle impostazioni del widget calendar del campo sostituire 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 consentiti gli spazi per separare le cifre per una più semplice lettura)

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...?".

11.2 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.

11.3 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 

 x ▼

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