



indra

intellera  
consulting



Comune di Salerno

Servizi di realizzazione e gestione di Portali e Servizi on-line

## Piattaforma Integrata dei Servizi al Cittadino (PICS)

*Gestore Pratiche*

*Manuale utente*

## STORICO DELLE REVISIONI

Ver.	Data emissione	Autore	Descrizione delle modifiche
1.1	27/01/2022	Luca Santaniello	Prima versione
1.2	29/07/2022	Luca Santaniello	Revisione
1.3	31/10/2022	Luca Santaniello	Refactor relazione utente-ruolo (par.5)
1.4	10/02/2023	Luca Santaniello	Aggiunti ulteriori dettagli sui vincoli relativi alla generazione dei report statici. Aggiunto sequence diagram che mostra la generazione report da sottomissione di una nuova istanza di pratica e da SCRIPT task. Aggiunto codice di esempio per lo script task per generare il PDF dal workflow.
1.5	13/02/2023	Luca Santaniello	Allegato documento che descrive i flussi di autenticazione e i flussi di pagamento. Aggiunto paragrafo che descrive la possibilità di invio e-mail da un workflow.
1.6	28/04/2023	Massimiliano Avagliano	Aggiunto paragrafo che descrive la logica sulla generazione dei report. Aggiunto paragrafo che descrive l'associazione protocollo-documenti.
1.7	05/05/2023	Massimiliano Avagliano	Aggiunto paragrafo per i log di importazione avanzata
1.8	31/05/2023	Massimiliano Avagliano	Aggiornato paragrafo di associazione protocollo-documenti Aggiunto capitolo 45 'Configurazione timeout verifica task'



## SOMMARIO

<b>1</b>	<b>INTRODUZIONE.....</b>	<b>6</b>
1.1	Riferimenti .....	6
1.2	Glossario .....	6
1.3	Acronimi .....	6
<b>2</b>	<b>ACCESSO AL SISTEMA .....</b>	<b>7</b>
<b>3</b>	<b>GESTIONE UTENTI .....</b>	<b>8</b>
<b>4</b>	<b>GESTIONE RUOLI .....</b>	<b>9</b>
<b>5</b>	<b>AUTORIZZAZIONE.....</b>	<b>10</b>
5.1	Gestione utenti .....	11
5.2	Gestione domini .....	12
5.3	Gestione ruoli .....	12
5.4	Lavorazione task .....	14
5.5	Storico istanze .....	14
5.6	Gestione procedure .....	15
5.7	Gestione importazione procedure .....	16
5.8	Gestione procedure interne .....	17
5.9	Gestione lookup.....	17
5.10	Ricerca istanze .....	17
5.11	Visualizzazione report.....	18
5.12	Lavorazione bozze .....	19
5.13	Firma Documenti .....	19
<b>6</b>	<b>GESTIONE DOMINI .....</b>	<b>20</b>
<b>7</b>	<b>GESTIONE FORM.....</b>	<b>21</b>
<b>8</b>	<b>GESTIONE WORKFLOW.....</b>	<b>24</b>
<b>9</b>	<b>GESTIONE SUBWORKFLOW.....</b>	<b>28</b>
<b>10</b>	<b>GESTIONE PROCEDURE.....</b>	<b>31</b>
<b>11</b>	<b>CONVERTITORE DI DOCUMENTI .....</b>	<b>36</b>
<b>12</b>	<b>REPORT .....</b>	<b>38</b>
<b>13</b>	<b>RICERCA GLOBALE.....</b>	<b>39</b>
<b>14</b>	<b>VISTA ATTIVITÀ .....</b>	<b>40</b>

15	VISTA STORICO PRATICHE .....	43
16	REPORT DI PROCEDURE (PDF) .....	47
16.1	Generatore PDF .....	49
16.2	Firma del documento generato .....	52
17	GESTIONE DELEGHE .....	53
18	GESTIONE STATI PROCEDURA .....	54
19	GESTIONE IMPORTAZIONE AVANZATA.....	55
20	PROTOCOLLAZIONE INTERNA.....	57
21	CARICAMENTO DINAMICO DI UNA LISTA .....	58
22	CONFIGURAZIONE DI UN NUOVO CONVERTITORE PDF/XML.....	61
22.1	Creazione di un PDF compilabile .....	63
22.2	Creazione di un XSL.....	63
22.3	Esecuzione del test .....	66
22.4	Configurazione del convertitore nel flusso .....	67
22.5	Configurazione Input Parameter .....	67
22.6	Configurazione Output Parameter.....	70
23	CHIUSURA DI UNA PRATICA TRAMITE CHIAMATA REST .....	74
24	VALIDAZIONE SOSTANZIALE TRAMITE SERVIZIO ESTERNO .....	75
25	CREAZIONE DI VARIABILI DI PROCESSO .....	78
26	SINCRONIZZAZIONE DI VARIABILI DI PROCESSO .....	80
27	SETTING DEI CAMPI SPID E RELATIVA VISUALIZZAZIONE.....	81
28	GESTIONE DEL SERVIZIO DI NOTIFICA .....	84
29	INVIO DEL SEGNALE .....	87
29.1	Caso d'uso .....	88
29.2	Sequence diagram .....	89
29.3	Gestione degli errori .....	89
29.4	Set dello stato di una istanza di procedura.....	90
30	GESTIONE AVANZATA LOOKUP.....	95
30.1	Database .....	97
31	RICERCA AVANZATA .....	99
31.1	Ricerca avanzata tramite query builder .....	99
31.2	Definizione del formato di esportazione .....	101
31.3	Ricerca avanzata tramite API .....	102
31.4	Definizione della query .....	105

32	GESTIONE DI UN TEMPLATE DI DOCUMENTO.....	108
33	CREAZIONE DI UN DELEGATO.....	109
34	PROTOCOLLAZIONE CON SERVICE TASK .....	110
35	GENERAZIONE REPORT DI PROCEDURA DA SERVICE TASK .....	111
35.1	Generazione del report di procedura .....	111
35.2	Generazione numero di protocollo.....	112
36	CREAZIONE DI METADATA .....	114
37	CREAZIONE DI LOOKUP.....	115
38	COMANDO PER ATTIVAZIONE/DISATTIVAZIONE PROXY E VISUALIZZAZIONE LOG .....	116
39	CONFIGURAZIONE DI WSO2.....	117
40	AUTENTICAZIONE, AUTORIZZAZIONE E PAGAMENTI .....	119
41	INVIO E-MAIL DA UN WORKFLOW .....	120
41.1	Invio tramite Script Task .....	120
41.2	Invio Tramite Installazione e Configurazione di un Connector Open-Source .....	121
42	GENERAZIONE DINAMICA DEI REPORT (MULTI-PDF) .....	123
42.1	Descrizione API .....	124
43	ASSOCIAZIONE PROTOCOLLO-DOCUMENTI.....	126
43.1	Caso d'uso .....	128
43.2	Visualizzazione documenti.....	132
44	VISUALIZZAZIONE LOG DI IMPORTAZIONE AVANZATA PROCEDURE.....	135
45	CONFIGURAZIONE TIMEOUT VERIFICA TASK .....	136
45.1	Configurazione timeout verifica task .....	136

# 1 INTRODUZIONE

Questo documento delinea un manuale utente per l'uso dell'interfaccia operatore del backoffice di Gestore Pratiche.

## 1.1 Riferimenti

Identificativo	Titolo/Descrizione

## 1.2 Glossario

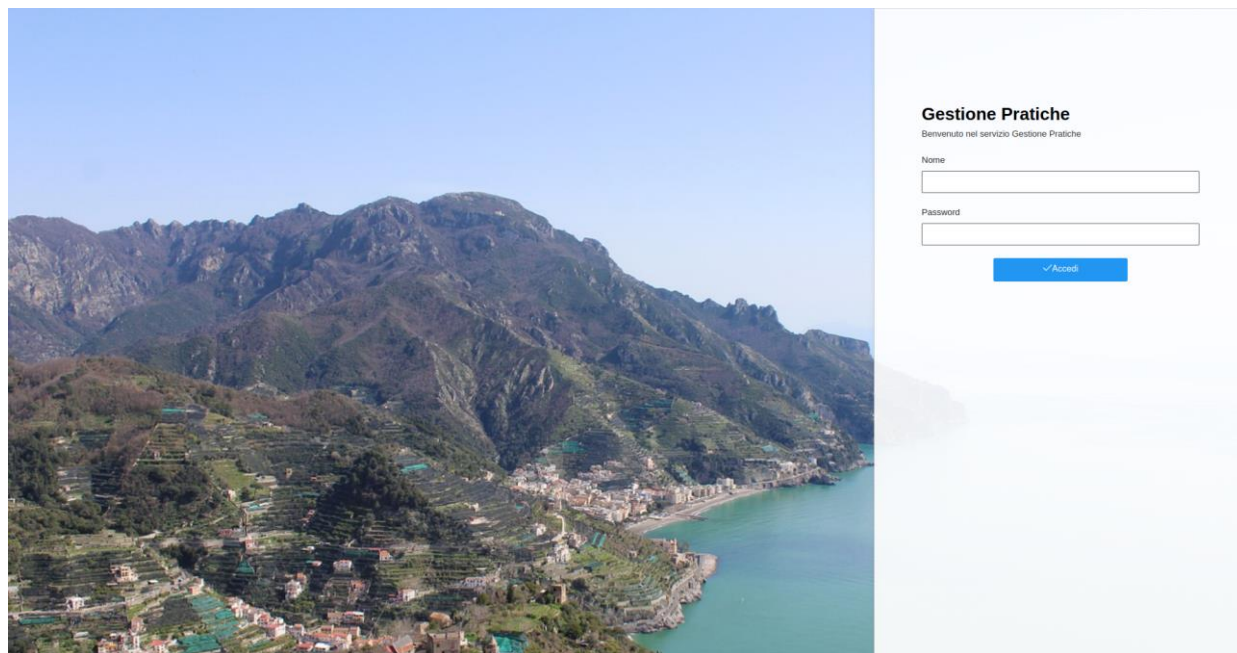
Definizione / Acronimo	Descrizione
Processo o Workflow	Flusso che descrive un processo di lavoro.
Template di Workflow	Sono modelli di workflow riutilizzabili in diversi ambiti funzionali.
Pratica	È la tipologia di flusso che si intende pubblicare e mettere a disposizione del cittadino.
Istanza	Rappresenta un elemento che un utente (es. cittadino) ha creato a partire da una specifica pratica. Ogni istanza avrà i suoi attributi e sarà differente da tutte le altre istanze.
Modulo	Rappresenta la form con i campi che devono essere riempiti dagli utenti. I moduli sono associati ai passi del workflow.
Dominio	Rappresenta il dominio di competenza (Edilizia, amministrativo...)

## 1.3 Acronimi

Definizione / Acronimo	Descrizione

## 2 ACCESSO AL SISTEMA

Gestione Pratiche è dotato di due diversi sistemi di autenticazione, il primo si basa su autenticazione **locale** mentre il secondo si basa su identity server SAML2 per fornire un metodo per permettere l'autenticazione di utenti non direttamente gestiti dall'applicativo.



## 3 GESTIONE UTENTI

Vi sono tre tipologie di utente:

- 1) Interni:** sono gli utenti che vengono creati all'interno di Gestione Pratiche. Per gli utenti interni è possibile modificare gli attributi principali quali password, ruoli e domini
- 2) Esterni:** sono gli utenti che si accedono a Gestione Pratiche tramite autenticazione esterna (identity provider). Quando l'utente accede per la prima volta tramite il suo identity provider, Gestione pratiche registra un nuovo utente di tipo esterno. Per gli utenti esterni è possibile modificare gli attributi relativi ai ruoli e ai domini.
- 3) Servizio:** sono una categoria di utenze usata esclusivamente dagli applicativi che si integrano a Gestione Pratiche. Un utente di servizio è un'utenza speciale che serve a un'applicazione integrata come mezzo di impersonificazione di un utente, in modo che Gestione Pratiche possa applicare correttamente i criteri d'accesso.

**Gestione Pratiche** amministratore Comune di Salerno

Salva Annulla

Dettaglio ruolo

**Dati di base**

Nome \*

Ruolo di default

Tipo \*

Standard

Domini

Permessi \*

Gestione impostazioni pratiche Visualizzazione report Gestione ruoli e utenti

☒ È il ruolo di default per l'utente esterno



## 4 GESTIONE RUOLI

Gestione Pratiche permette di definire i ruoli di un utente. Il dettaglio di un ruolo è rappresentato dalle seguenti proprietà:

- 1) **Nome:** il nome che identifica il ruolo;
- 2) **Tipo:** la tipologia di un ruolo è in sola lettura, di default è standard;
- 3) **Domini:** i domini di un ruolo;
- 4) **Permessi:** i permessi di un ruolo definiscono le visibilità di Gestione Pratiche da parte di un utente (quali sezioni può vedere dell'applicativo).

The screenshot displays the 'Gestione Pratiche' application interface. At the top, a blue header bar contains the title 'Gestione Pratiche' on the left and a user profile icon with the text 'amministratore Comune di Salerno' on the right. Below the header, a sidebar on the left shows a navigation menu with icons for home, list, search, table, settings, and user. The main content area is titled 'Dettaglio ruolo' and features a 'Save' button and an 'Annulla' button in the top right corner. The form is divided into sections: 'Dettaglio ruolo' (with a sub-section 'Dettaglio ruolo'), 'Dettaglio ruolo', and 'Dettaglio ruolo'. The 'Dettaglio ruolo' section includes fields for 'Nome \*', 'Ruolo di default', 'Tipo \*' (with a dropdown menu showing 'Standard'), 'Domini' (with a dropdown menu), and 'Permessi \*' (with a dropdown menu showing 'Gestione impostazioni pratiche', 'Visualizzazione report', and 'Gestione ruoli e utenti'). At the bottom, there is a checkbox labeled 'E il ruolo di default per l'utente esterno' which is checked.

## 5 AUTORIZZAZIONE

Questo capitolo descrive le funzionalità di autorizzazione previste nel backoffice.

Il GP realizza le funzionalità di autorizzazione, ovvero profilazione, tramite la configurazione di ruoli. Ogni ruolo può contenere uno o più permessi/funzioni. Quando un utente viene associato ad un ruolo, erediterà tutti i permessi associati a quel ruolo.

Di seguito un'immagine che descrive la relazione tra utente, ruolo, permessi e domini:

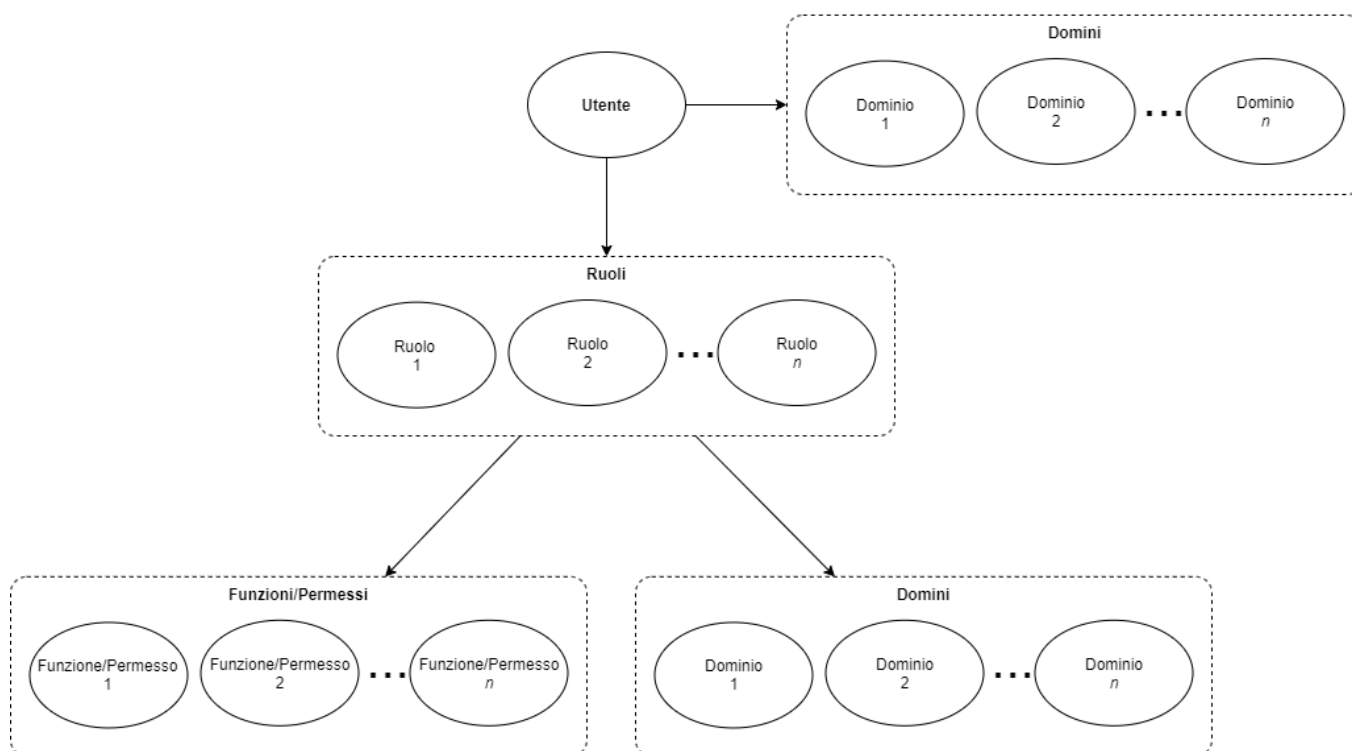


Figura 1 – Relazione tra Utente, Ruolo, Permessi e Domini

Il Gestore Pratiche gestisce le seguenti funzioni/permessi:

- Gestione utenti
- Gestione ruoli
- Visualizzazione report
- Lavorazione task
- Gestione procedure
- Ricerca istanze
- Lavorazione bozze
- Gestione lookup
- Storico istanze
- Gestione procedure interne
- Gestione domini
- Gestione importazione procedure

## ■ Firma Documenti

Una installazione di base, conterrà già alcuni ruoli pre-configurati, secondo la tabella di seguito:

		Funzione												
		Gestione utenti	Gestione ruoli	Visualizzazione report	Lavorazione task	Gestione procedure	Ricerca istanze	Lavorazione bozze	Gestione lookup	Storico istanze	Gestione procedure interne	Gestione domini	Gestione importazione	Firma Documenti
Ruoli disponibili	Amministratore	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Gestione pratiche					✓								
	Operatore				✓		✓			✓				

Ovviamente sarà possibile sempre aggiungere, modificare o eliminare ruoli alla piattaforma.

Nei paragrafi che seguono, descriviamo le funzioni/permessi disponibili sulla piattaforma.

### 5.1 Gestione utenti

L'utente che possiede questo permesso può avere accesso alla funzione di gestione degli utenti. In particolare avrà accesso al menù che gli consente di:

- Creare un nuovo utente (questa funzione non è utilizzabile nel caso di integrazione con una piattaforma di Identity Server)
- Esportare l'elenco degli utenti in CSV
- Modificare i ruoli di un utente
- Gestire i domini di appartenenza di un utente

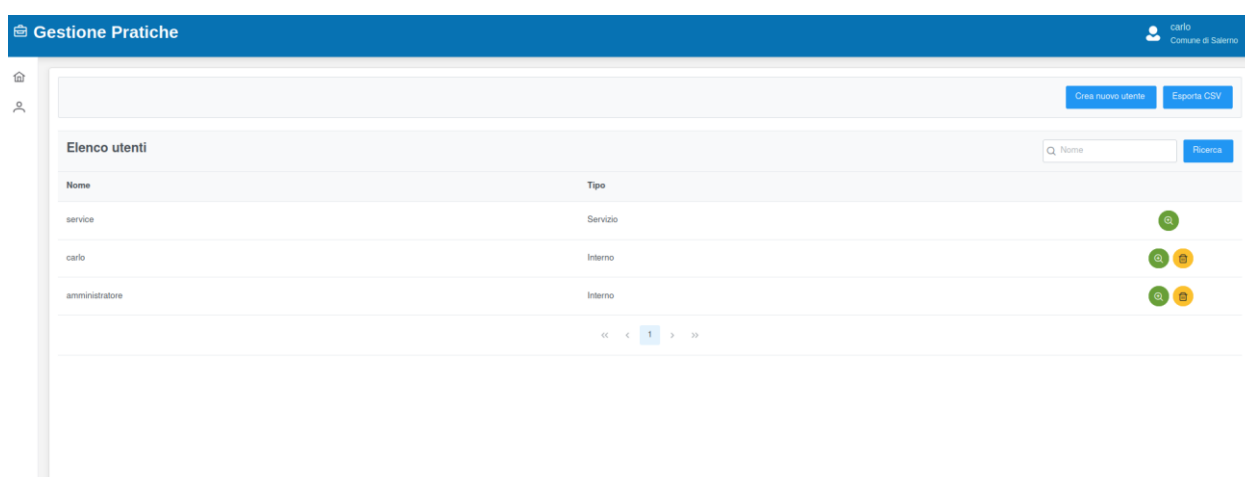


Figura 2 - UI di gestione utenti

## 5.2 Gestione domini

L'utente che possiede questo permesso può avere accesso alla funzione di gestione dei domini. In particolare avrà accesso al menù che gli consente di:

- Creare un nuovo dominio
- Cancellare un dominio
- Esportare la lista dei domini in CSV

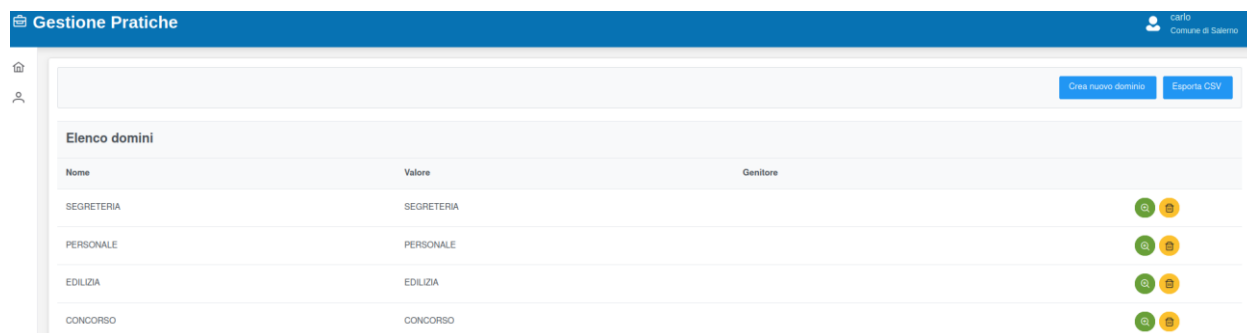


Figura 3 - UI di gestione domini

## 5.3 Gestione ruoli

L'utente che possiede questo permesso può avere accesso alla funzione di gestione dei ruoli. In particolare avrà accesso al menù che gli consente di:

- Creare un nuovo ruolo
- Cancellare un ruolo
- Esportare la lista dei ruoli in CSV

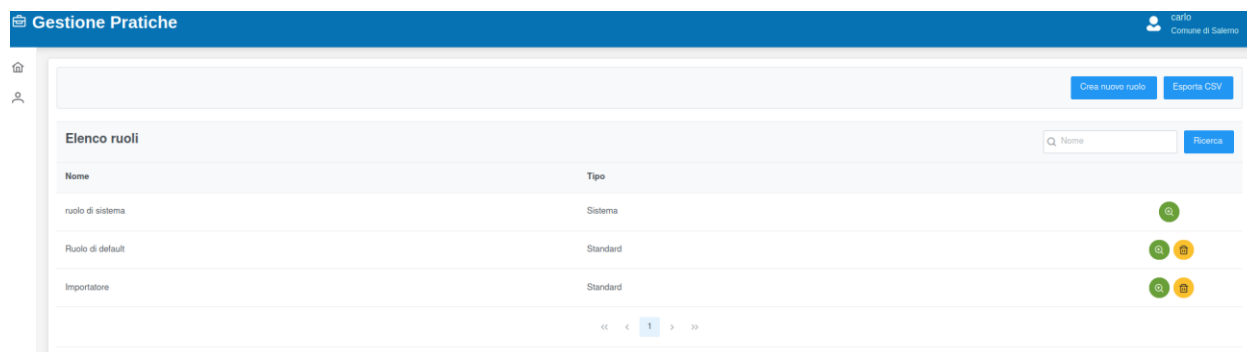


Figura 4 - UI di gestione ruoli

Quando viene creato un ruolo è possibile associare ad esso le funzioni e i domini che si desidera.

La selezione dei domini è mostrata nella figura seguente:

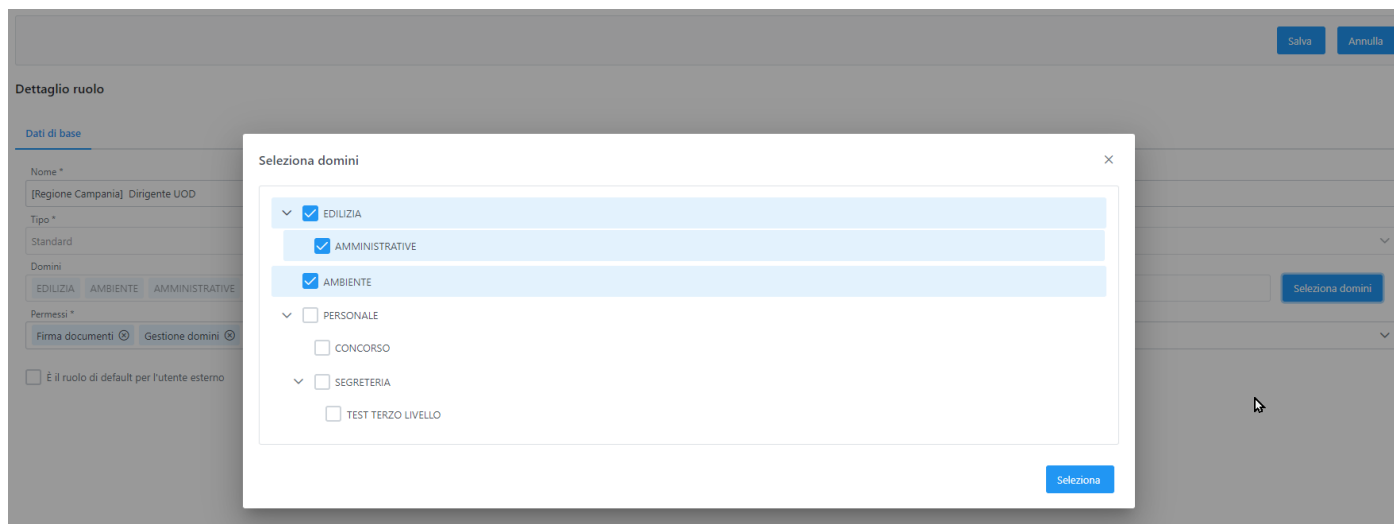


Figura 5 - Associazione dei domini al ruolo

La selezione dei ruoli è mostrata nella figura che segue:

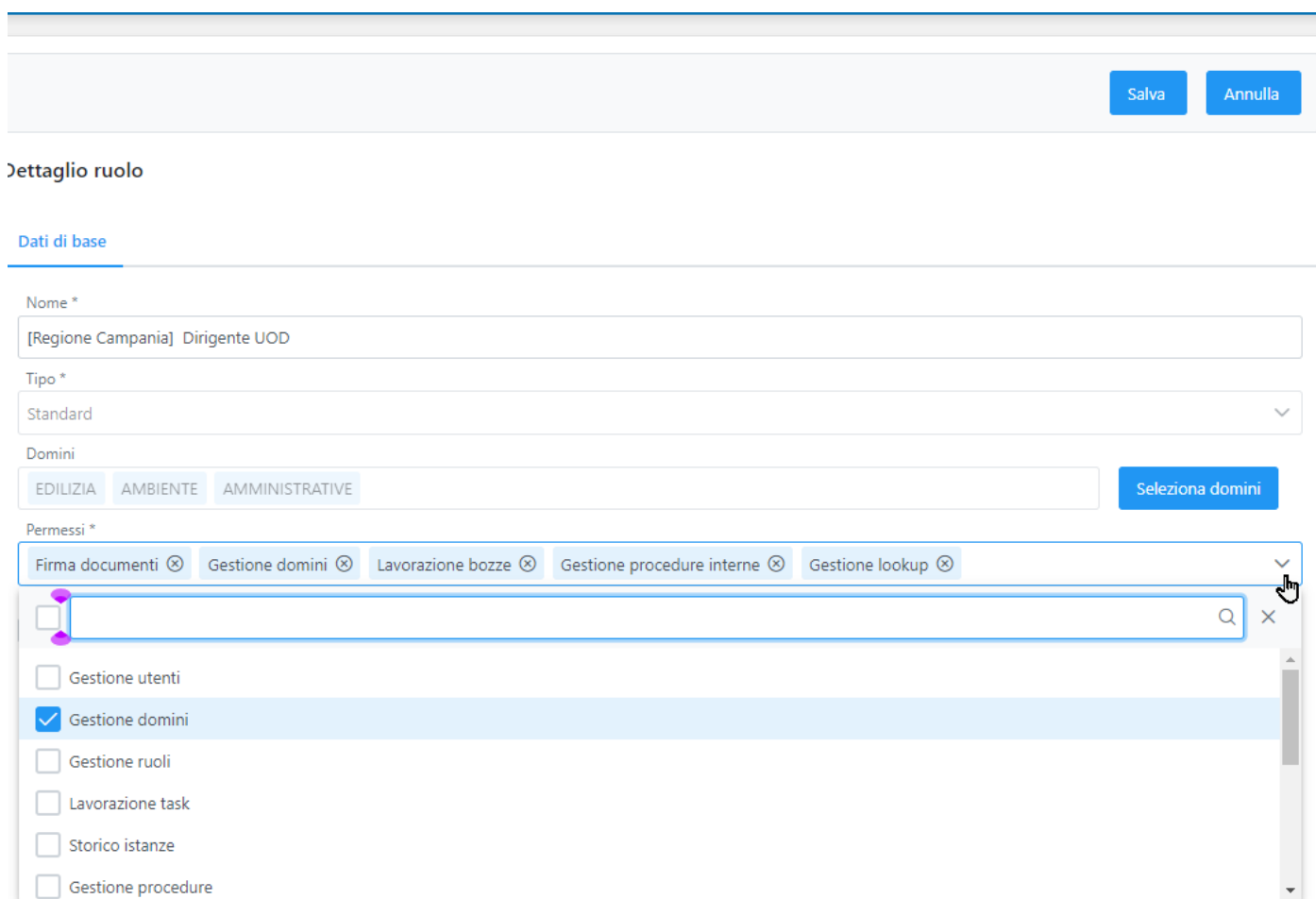


Figura 6 - Associazione dei permessi al ruolo

## 5.4 Lavorazione task

L'utente che possiede questo permesso può prendere in carico le attività assegnate al ruolo che ha associato.

Le funzioni abilitate riguardano:

- Elenco delle attività prese in carico
- Elenco delle attività disponibili
- Possibilità di visualizzare il dettaglio di una attività, tra cui:
  - o Scheda di lavorazione (form)
  - o Variabili
  - o Documenti
  - o Workflow con indicazione dello task corrente
  - o Storico delle sottomissioni

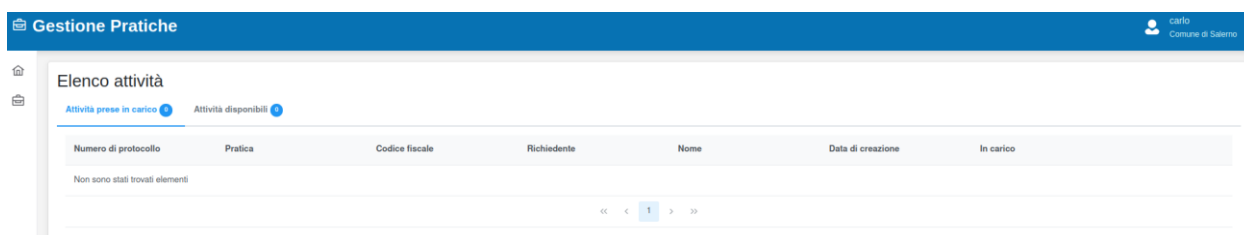


Figura 7 – UI associata al permesso Lavorazione task

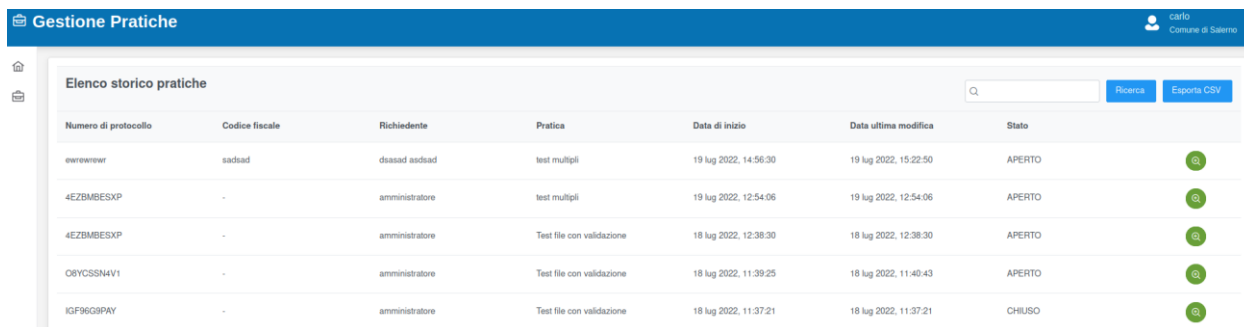
Dal dettaglio di una attività, è possibile prenderla in carico per poterla lavorare:

Figura 8 - Presa in carico dell'attività

## 5.5 Storico istanze

L'utente che possiede questo permesso potrà avere accesso in sola visualizzazione alle istanze di procedura completate. Potrà quindi aprirle in dettaglio e visualizzarne i contenuti.

Da questa schermata sarà possibile esportare un elenco in CSV.



The screenshot shows the 'Gestione Pratiche' (Practice Management) interface. At the top, there's a header bar with the title 'Gestione Pratiche' and a user profile icon for 'staffo Comune di Salerno'. Below the header, there's a section titled 'Elenco storico pratiche' (Historical Practices List). This section contains a table with columns: 'Numero di protocollo' (Protocol Number), 'Codice fiscale' (Tax Code), 'Richiedente' (Requester), 'Pratica' (Practice), 'Data di inizio' (Start Date), 'Data ultima modifica' (Last Modification Date), and 'Stato' (Status). To the right of the table, there are search and export buttons labeled 'Ricerca' and 'Esporta CSV'. The table lists five practices, all with the status 'APERTO' (Open), except for the last one which is 'CHIUSO' (Closed). Each row has a green circular icon with a white 'S' in the rightmost column.

Numero di protocollo	Codice fiscale	Richiedente	Pratica	Data di inizio	Data ultima modifica	Stato
ewewewew	sadsad	dsasad asdsad	test multipli	19 lug 2022, 14:56:30	19 lug 2022, 15:22:50	APERTO
4EZBMBSXP	-	amministratore	test multipli	19 lug 2022, 12:54:06	19 lug 2022, 12:54:06	APERTO
4EZBMBSXP	-	amministratore	Test file con validazione	18 lug 2022, 12:38:30	18 lug 2022, 12:38:30	APERTO
Q8YC5SN4V1	-	amministratore	Test file con validazione	18 lug 2022, 11:39:25	18 lug 2022, 11:40:43	APERTO
KGF96G8PAY	-	amministratore	Test file con validazione	18 lug 2022, 11:37:21	18 lug 2022, 11:37:21	CHIUSO

Figura 9 - UI di storico istanze

## 5.6 Gestione procedure

L'utente che possiede questo permesso può gestire le procedure e le funzionalità base per creare una procedura.

In particolare sarà possibile:

- Creare una nuova procedura
- Modificare una procedura
- Eliminare una procedura (qualora non sia in stato di pubblicazione)
- Pubblicare una procedura (vedi immagine Figura 10 - Pubblicazione di una procedura) e scegliere la visibilità:
  - o Pubblicata (visibile ai cittadini sul portale pubblico)
  - o Pubblicata internamente (visibile solo nel backoffice)
  - o Pubblicata in test
- Annullare la pubblicazione
- Esportare/Importare una procedura
- Esportare un elenco CSV delle procedure

Dettaglio procedura

Dati di base Configurazioni Workflow Permessi Pubblicazioni Metadata

Tipo di procedura \*

Procedura con workflow

Nome \*

Invio segnale di sblocco

Data di fine disponibilità

Domini

CONCORSO

Stati di procedura

aperto chiuso

Descrizione

Normal Sans Serif B I U A [icon] [icon] [icon] [icon] [icon] [icon]

Invio sblocco del segnale

Imposta visibilità della procedura

Tipo \*

Pubblicata

Pubblicata internamente

Test

Seleziona domini

Figura 10 - Pubblicazione di una procedura

Gestione Pratiche

Importa procedura Crea nuova procedura Esporta CSV

Elenco procedure

Nome	Creato il	Data ultima modifica	Data di fine disponibilità	Stato	Visibilità
Test esportazione modulo	7 lug 2022, 15:34:25	19 lug 2022, 17:24:51	4 lug 2022	pubblicato	Pubblicata
Test dominio procedura	7 lug 2022, 12:40:18	19 lug 2022, 17:21:01	29 lug 2022	pubblicato	Pubblicata
Concorso wizard	8 lug 2022, 09:21:09	19 lug 2022, 09:59:33	-	pubblicato	Pubblicata internamente
Test file con validazione	15 lug 2022, 12:57:53	19 lug 2022, 09:58:02	-	pubblicato	Pubblicata internamente

Figura 11 - UI di gestione procedure

## 5.7 Gestione importazione procedure

L'utente che possiede questo permesso ha accesso alla funzione di import delle procedure, ovvero può:

- creare, modificare ed eliminare un nuovo formato di importazione istanze di procedura
- importare le istanze di procedura secondo i formati definiti



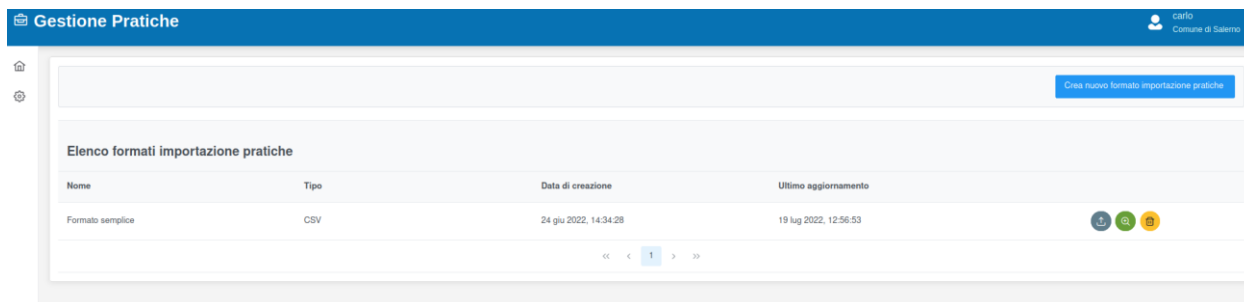


Figura 12 - Gestione importazione procedure

## 5.8 Gestione procedure interne

L'utente che possiede questo permesso ha la possibilità di creare istanze di procedura che sono state pubblicate ad uso interno. La sezione "Le mie pratiche" consente di:

- Creare nuove istanze
- Visualizzare lo stato di avanzamento delle istanze create

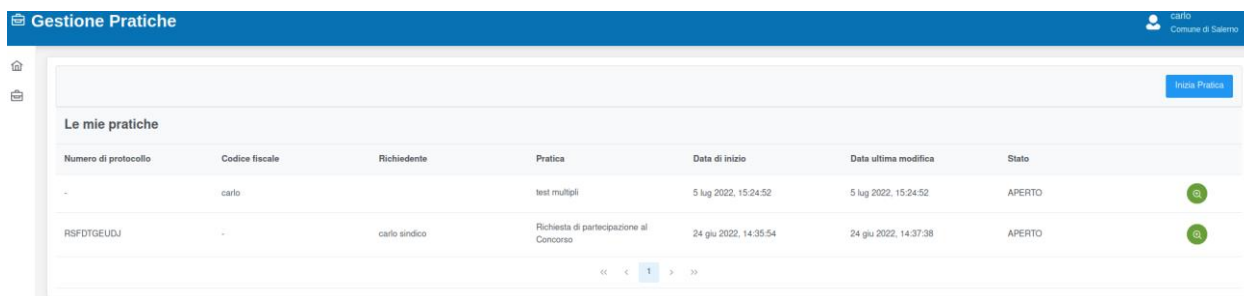


Figura 13 - UI di gestione delle istanze interne - "Le mie pratiche"

## 5.9 Gestione lookup

L'utente che possiede questo permesso può creare delle liste di lookup. Tali liste di valori potranno essere successivamente utilizzate in fase di definizione di una form.

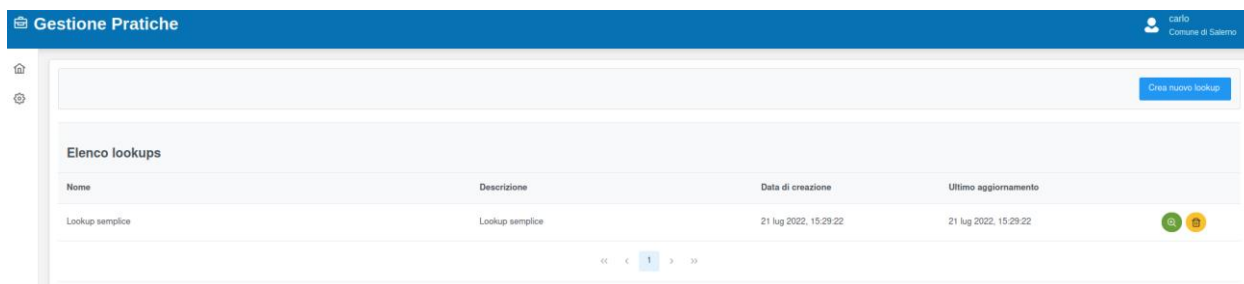


Figura 14 - UI di gestione lookup

## 5.10 Ricerca istanze

L'utente che possiede questo permesso può:

- 1) cercare le pratiche tramite ricerca semplice
- 2) cercare le pratiche tramite ricerca avanzata

The screenshot shows the 'Gestione Pratiche' (Practice Management) interface. On the left, there is a sidebar titled 'Filtri di ricerca' (Search Filters) with a search bar and several filter categories: 'Ulteriori filtri di ricerca' (Further search filters), 'Tipo di pratica' (Practice type), 'Numero di protocollo' (Protocol number), 'Stati di procedura' (Procedure states), 'Codice Fiscale del richiedente o di uno dei suoi delegati' (Tax code of the applicant or one of his delegates), 'Nome del richiedente o di uno dei suoi delegati' (Name of the applicant or one of his delegates), 'Cognome del richiedente o di uno dei suoi delegati' (Surname of the applicant or one of his delegates), 'Data di inizio' (Start date), and 'Data di fine disponibilità' (Availability end date). The main area on the right displays the message 'Per favore, eseguire la ricerca.' (Please perform the search.).

Figura 15 - UI di ricerca semplice

The screenshot shows the 'Ricerca avanzata' (Advanced Search) interface. On the left, there is a sidebar titled 'Ricerca avanzata' with sections: 'Elenco procedure' (Procedure list) with a dropdown menu, 'Filtri di interrogazione' (Interrogation filters) with buttons for 'E', 'O', 'Aggiungi regola' (Add rule), and 'Aggiungi gruppo' (Add group), and 'Ordinamenti' (Ordering) with dropdowns for 'Ordinamento per' (Ordering by) and 'Ordinamento' (Ordering). The main area on the right displays a table with columns: 'Numero di protocollo' (Protocol number), 'Codice fiscale' (Tax code), 'Richiedente' (Applicant), 'Nome' (Name), 'Data di creazione' (Creation date), 'Ultimo aggiornamento' (Last update), and 'Stato' (Status). The table currently shows 'Non sono state trovate pratiche' (No practices were found) and a pagination bar with '1'.

Figura 16 - UI di ricerca avanzata

## 5.11 Visualizzazione report

L'utente che possiede questo permesso può visualizzare i report messi a disposizione per la sua utenza.

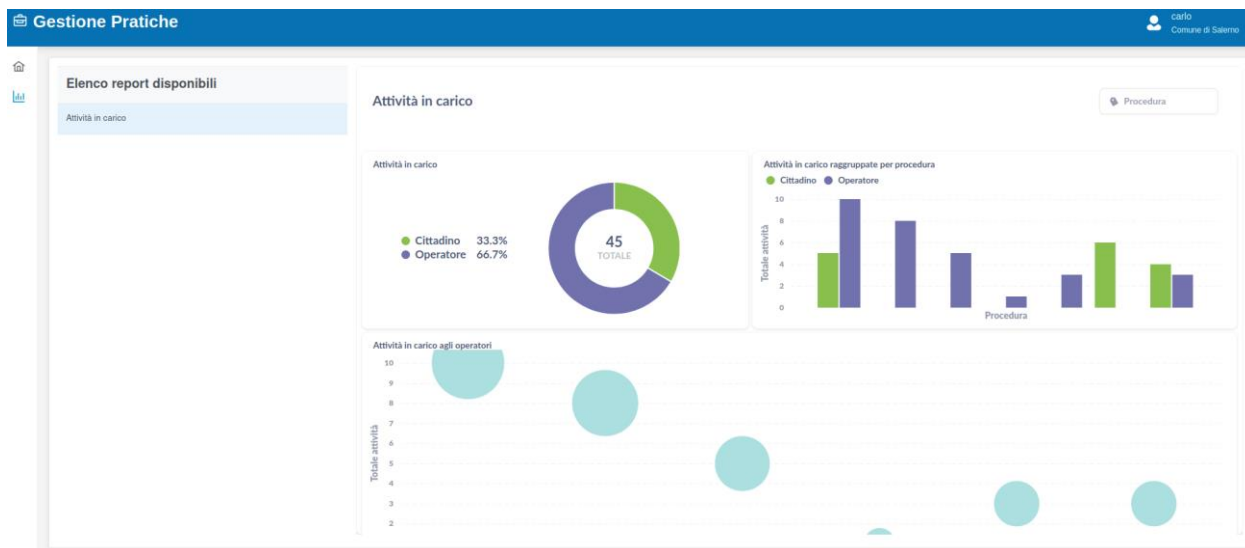


Figura 17 - Ui di visualizzazione report

## 5.12 Lavorazione bozze

L'utente che possiede questo permesso può gestire le bozze delle istanze di pratica, ovvero tutte quelle istanze di pratica che non sono state ancora sottomesse dal cittadino, ma solamente salvate in bozza.

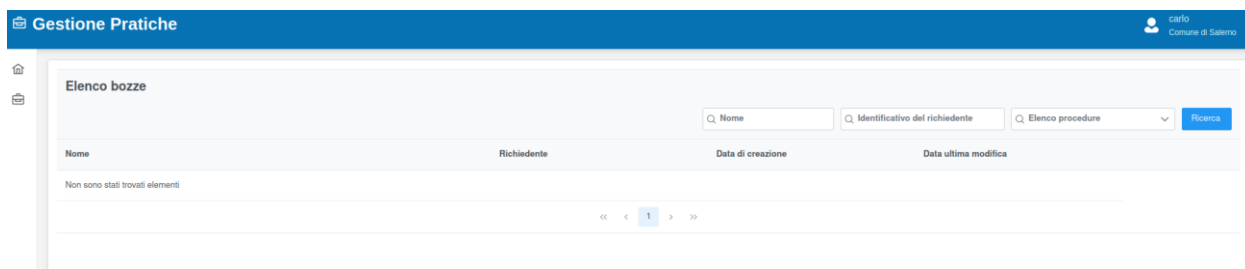


Figura 18 - UI di lavorazione bozza

## 5.13 Firma Documenti

Questo permesso sarà attivo una volta completata la funzionalità di firma documentazione.

## 6 GESTIONE DOMINI

I domini applicativi rappresentano il contesto sul quale si baseranno le procedure ed i relativi contenuti creati dall'utente. I domini servono per etichettare una procedura (una procedura può avere da zero a infiniti domini). I domini possono essere usati anche per filtrare ciò che gli utenti possono vedere, associandoli ai ruoli o direttamente agli utenti stessi: se un utente è abilitato per un dominio o possiede un ruolo abilitato per un dominio, potrà vedere esclusivamente le pratiche di tale dominio.

Un utente non associato a un dominio può vedere i dati di tutte le pratiche senza particolari limitazioni.

I domini sono applicati unicamente alla lavorazione delle pratiche. Ciò significa che i domini non rappresentano un criterio di visibilità per il richiedente della pratica. Ciò è vero sia per gli applicativi terzi (es. Portale istituzionale) che si interfacciano al Gestore pratiche ma anche alle procedure interne che è possibile creare direttamente dal backoffice di Gestore pratiche. Un utente di backoffice vedrà sempre tutte le pratiche nella funzionalità che gli permette di iniziare una nuova pratica interna, mentre le pratiche che egli potrà lavorare saranno visibili in base ai suoi domini.

The screenshot shows the 'Gestione Pratiche' interface. At the top, there is a blue header bar with the title 'Gestione Pratiche' on the left and a user profile 'amministratore Comune di Salerno' on the right. Below the header, there is a sidebar with icons for home, list, search, table, settings, and user. The main content area is titled 'Dettaglio dominio' and contains a 'Dati di base' section. This section has two input fields: 'Nome \*' and 'Edilizia'. Below these, there is a 'Valore \*' field with the text 'EDILIZIA' entered. At the top right of the form, there are 'Salva' and 'Annulla' buttons.

## 7 GESTIONE FORM

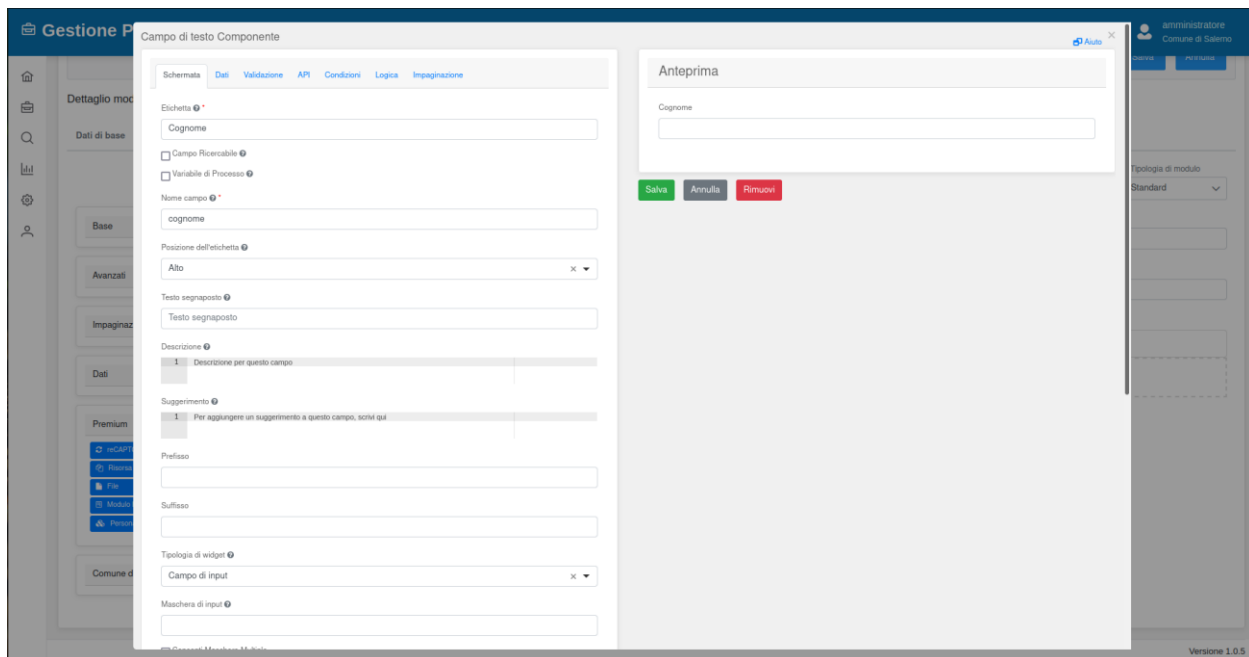
I form, che in Gestione Pratiche si identificano con il termine moduli, permettono di definire le maschere web di una procedura associate a tutte le attività che devono essere prese in carico e concluse dagli utenti.

Alla creazione di un modulo è possibile scegliere i campi che andranno a popolare il form da una palette di campi disponibili.

E' inoltre possibile definire e implementare una palette personalizzata di campi per poter incrementare l'insieme di componenti disponibili per la creazione di nuovi moduli.

Per ogni campo sono selezionabili diversi parametri tra i quali:

- 1) **Etichetta:** il nome del campo visibile sopra al campo stesso;
- 2) **Nome variabile:** identificativo univoco per quel campo all'interno del form;
- 3) **Variabile di processo:** flag che specifica se il campo verrà processato dal workflow engine che esegue la procedura;
- 4) **Campo ricercabile:** indica se il campo è ricercabile o meno. Se un campo è ricercabile, questo verrà usato come contenuto ricercabile dalla sezione di ricerca di Gestore Pratiche.



Gestione Pratiche permette di definire un modulo come la composizione di altri moduli grazie alla funzionalità di importazione ed esportazione. In questo modo è più facile costruire moduli riutilizzabili in futuro.

Nella pagina di creazione di un modulo è possibile selezionare il modulo che si vuole importare:

- 1) il modulo da importare può essere aggiunto in fondo a quello che si sta creando.
- 2) il modulo da importare può essere aggiunto in uno specifico container del modulo che si sta creando.

**N.B per il punto 2) sono supportati i componenti di tipo contenitore e pannello.**

Gestione Pratiche

admin  
Comune di Salerno

Elenco moduli

Nome	Data di creazione	Data ultima modifica
[Salerno] Concorso - Wizard	7 giu 2022, 10:12:27	26 lug 2022, 15:50:43
[Salerno] Toponomastica Wizard	25 mag 2022, 11:42:39	26 lug 2022, 11:42:36
Validazione formale	22 lug 2022, 11:30:45	22 lug 2022, 11:36:49
Test Report	21 lug 2022, 10:18:54	21 lug 2022, 10:18:54
File con validazione	18 lug 2022, 12:20:13	18 lug 2022, 12:20:13
[Salerno] Tari - Riduzione 2022	12 mag 2022, 16:38:59	11 lug 2022, 11:14:06
Test dominio in user task	6 lug 2022, 16:52:37	6 lug 2022, 16:52:37
esempio	5 lug 2022, 09:56:53	5 lug 2022, 09:56:53
Modulo d'esempio	14 giu 2022, 10:38:11	14 giu 2022, 10:38:11
[Salerno] SCIA - VICINATO	7 giu 2022, 09:51:40	10 giu 2022, 11:12:00

<<

<

1

2

3

4

5

>

>>

Importa in un componente

Importa in fondo al modulo

## 8 GESTIONE WORKFLOW

La creazione di un workflow consiste nel comporre un diagramma di flusso attraverso uno strumento visuale che permette di definire le attività, le correlazioni, le condizioni e le dipendenze tra di esse.

Il modello di notazione usato per il disegno dei workflow si basa sullo standard BPMN 2.0, l'acronimo BPMN sta per Business Process Model and Notation, (<https://www.bpmn.org/>).

Le componenti principali dell'editor sono:

- 1) **Palette di sinistra:** permette di aggiungere i componenti che poi faranno parte de diagramma, quali ad esempio *start event*, *user task*, *script task*, *end event*.

Ciascun componente è caratterizzato da un particolare forma, ciò permette una migliore comprensione dell'intero diagramma;



- 2) **Palette di destra:** permette di assegnare le proprietà all'elemento selezionato nel workflow.

I task di tipo utente (user task) sono contraddistinti da una particolare icona.





Questi task sono ricercabili dagli utenti e da essi lavorabili. A ogni task di tipo utente è assegnabile un modulo\form definito precedentemente nella gestione moduli. È altresì possibile creare un modulo personalizzato per ogni singolo task.

L'assegnazione di un form ad un task richiede le seguenti proprietà, impostate nella palette di destra quando il task è selezionato nell'editor:

- 1) **Name:** il nome del task.
- 2) **Assigned to:** il task può essere assegnato al **cittadino** oppure **all'operatore**.

Il form può essere selezionato da uno di quelli precedentemente creati, oppure si può creare un form ad hoc.

The screenshot shows the 'General' tab of a task editor. It has a tab bar at the top with 'General', 'Forms', 'Listeners', and 'Input/Output'. The 'General' section contains an 'Id' field with the value 'Activity\_0yc1cog' and a close button. Below it is a note: 'This maps to the task definition key.' The 'Name' field is empty. The 'Details' section has an 'Assigned to' dropdown menu.

The screenshot shows the 'Forms' tab of the task editor. It has a tab bar at the top with 'General', 'Forms', 'Listeners', and 'Input/Output'. The 'Forms' section has a 'Form Key' label and two buttons: 'Seleziona modulo da catalogo' and 'Nuovo modulo'.

**Gestione Pratiche** admin  
Comune di Salerno

**Elenco ruoli**

Nome	Tipo
[Regione Campania] Dirigente UOD	Standard
[Regione Campania] Direttore Generale	Standard
Utente esterno	Standard
Ruolo di prova	Standard
Responsabile ufficio edilizia	Standard
Responsabile ufficio amministrativo	Standard
Responsabile ufficio ambiente	Standard
Operatore ufficio edilizia	Standard
Operatore ufficio amministrativo	Standard
Operatore ufficio ambiente	Standard

<< < 1 2 > >>

Annulla Ok

Se il task viene assegnato all'operatore, è possibile selezionare i ruoli per cui quel task sarà visibile. Ogni utente ha un ruolo, i domini associati e i permessi di visibilità.

**Gestione Pratiche** amministratore  
Comune di Salerno

Dettaglio workflow

Dati workflow Editor

Importa da template

Process\_02d913k

General Variables Listeners Extensor

General

Id  
Process\_02d913k

Name  
This maps to the process definition key.

Version Tag

External Task Configuration

Task Priority

Job Configuration

Job Priority

Candidate Starter Configuration

Candidate Starter Groups

Specify more than one group as a comma separated list.

Candidate Starter Users

Specify more than one user as a comma separated list.

R.T.I. Almaviva S.p.A/ Almawave S.p.a./ Indra Italia S.p.A/ Intellera Consulting S.r.l.	Sistema Pubblico di Connettività LOTTO 4
Documentazione Tecnica	SPCL4_COMUNESALERNO_PICS_ManualeUtente_GestorePratiche

Il tasto “Importa da template” dà la possibilità di importare direttamente un workflow precedentemente definito e creato: il grande vantaggio della definizione dei template è quella di creare dei modelli di workflow che possono essere riutilizzati in più procedure.

## 9 GESTIONE SUBWORKFLOW

Gestione Pratiche permette di definire un workflow che richiama un altro workflow. Gestione pratiche estende una funzionalità che già viene esposta dal workflow engine,

e si compone delle seguenti proprietà ( il workflow deve definire un elemento di tipo call Activity):

- 1) selezionare (per ogni called activity) il sotto processo.
- 2) selezionare la versione del sotto processo

**Gestione Pratiche**admin  
Comune di Salerno

Salva

Annulla

Dettaglio workflow

Dati workflowEditorRiferimenti

Importa da template

```
graph LR; Start(( )) --> Operatore1[Operatore1]; Operatore1 --> Split{+}; Split --> WF3-1[WF3-1]; Split --> WF3-2[WF3-2]; Split --> WF2[WF2]; WF3-1 --> Join{+}; WF3-2 --> Join; WF2 --> Join; Join --> Operatore1-2[Operatore1-2]; Operatore1-2 --> End(( ))
```

Activity\_1gt7yk9

GeneralVariablesListenersInput/Output

General

Id  
Activity\_1gt7yk9

Name  
WF3-1

Details

CallActivity Type  
BPMN

Called Element  
wf3\_a7a87b19-7359-4e7d-b568-a8f8  
Seleziona sotto processo

Binding  
latest

Tenant Id

☐ Business Key

Delegate Variable Mapping

**Gestione Pratiche**

admin  
Comune di Salerno

**Elenco procedure**

Nome	Data di creazione	Ultimo aggiornamento
Valutazioni Commissario	20 giu 2022, 12:19:54	15 lug 2022, 11:17:26
Test commissari	11 lug 2022, 09:52:58	15 lug 2022, 11:16:14
WF2	8 giu 2022, 12:22:34	14 giu 2022, 09:56:17
Istruttoria	7 giu 2022, 10:36:31	7 giu 2022, 11:10:12
wfl test2	13 giu 2022, 14:00:20	13 giu 2022, 14:21:21
WF3	8 giu 2022, 12:22:34	14 giu 2022, 09:56:17
wf test3	13 giu 2022, 14:00:20	13 giu 2022, 14:21:21
Build Part sub WFL	7 giu 2022, 10:36:31	7 giu 2022, 11:10:12
Procedura con workflow figlio	7 giu 2022, 09:14:20	7 giu 2022, 09:14:48

**Attenzione**

Vuoi selezionare una specifica versione, oppure utilizzare l'ultima?

Nel dettaglio di una pratica è possibile visualizzare le seguenti informazioni:

- 1) verificare lo stato di avanzamento visualizzando sia il workflow di partenza sia tutti i processi figli (subworkflow).
- 2) visualizzare l'origine delle variabili (se appartengono al workflow di partenza oppure ad un subworkflow).

**N.B i servizi che permettono di sincronizzare le variabili tra Gestione Pratiche e il workflow engine richiedono un nuovo parametro "origine" per specificare l'origine delle variabili che verranno create.**

Gestione Pratiche

amministratore  
Comune di Salerno

Dettaglio pratica - GUIE4W5LE4 APERTO

Dati pratica

Richiedenti

Variabili

Documenti

Attività

**Workflow**

Sottomissioni

**Workflow padre**

test multipli

```
graph LR; Start(( )) --> TaskOperator[task operator]; TaskOperator --> ChildProcess[chiamata a processo figlio]; ChildProcess --> End(( ))
```

## 10 GESTIONE PROCEDURE

Gestione Pratiche permette di creare una procedura o di importarne una precedentemente definita.

Una procedura si compone delle seguenti proprietà:

- 1) **Tipo di procedura**, che può essere:
  - di tipo **semplice** quando è composta unicamente da un form che il cittadino deve compilare e inviare al sistema [procedura semplice];
  - di tipo **workflow** quando è associata ad un workflow definito nella gestione workflow.
- 2) **Nome**: nome della procedura, che verrà visualizzato nel dettaglio e in forma tabellare;
- 3) **Domini**: i domini della procedura, selezionati tra i domini creati nell'apposita sezione di gestione. Tali domini servono per etichettare la procedura e per creare particolari filtri di visualizzazione agli utenti (vedere sezione utenti);

The screenshot shows the 'Gestione Pratiche' application interface. The main header is 'Gestione Pratiche' with a user profile 'amministratore Comune di Salerno'. The left sidebar contains navigation icons. The main content area is titled 'Dettaglio procedura' and has tabs for 'Dati di base', 'Configurazioni', 'Workflow', 'Permessi', and 'Pubblicazioni'. The 'Dati di base' tab is active, showing a form with the following fields: 'Tipo di procedura' (dropdown menu with 'Procedura con workflow' selected), 'Nome' (text field with 'Richiesta concessione edilizia'), 'Domini' (dropdown menu with 'Edilizia' selected), and 'Descrizione' (rich text editor with 'Richiesta concessione edilizia'). At the bottom, there is a 'Stato' field with a 'pubblicato' button. At the top right of the form area, there are two buttons: 'Esporta procedura' and 'Annulla la pubblicazione'.

- 4) **Descrizione**: contiene la descrizione<sup>1</sup> della procedura.

Nella tab **Configurazioni** sono presenti i seguenti campi:

<sup>1</sup> La descrizione è in formato HTML.

- 1) **Cartella di salvataggio dei documenti:** il form di un workflow di una procedura può contenere dei campi dove è richiesto l'upload di documenti. È necessario quindi scegliere la cartella di salvataggio di questi documenti. I documenti verranno salvati nel sistema documentale configurato sotto questo percorso.
- 2) **Percorso di salvataggio:** è possibile scegliere un percorso di salvataggio personalizzato, secondo alcune regole definite da un template potendo personalizzare i nomi delle cartelle nelle quali verranno salvati i documenti associati alla procedura. I template sono stringhe contenenti dei particolari segnaposto che saranno calcolati dal sistema in fase di creazione della cartella nel sistema documentale configurato. Il percorso può essere suddiviso in sottocartelle usando il carattere /.

Percorso di salvataggio personalizzato

/\${year}/\${month}/\${requesterIdentifier}



Tali segnaposto sono:

- **year:** anno di creazione della domanda\procedura (es. 2020)
- **month:** mese di creazione della domanda\procedura (es. 06)
- **day:** giorno di creazione della domanda\procedura (es. 16)
- **requesterIdentifier:** codice fiscale del richiedente
- **protocolNumber:** numero di protocollo assegnato dal sistema in fase di protocollazione della domanda\procedura

- 3) **Generatore di protocollo:** il generatore usato per protocollare una procedura. La protocollazione avverrà alla apertura della procedura da parte del cittadino richiedente
- 4) **Report:** il report PDF che sarà prodotto e salvato nel sistema documentale alla apertura della procedura da parte del cittadino richiedente.

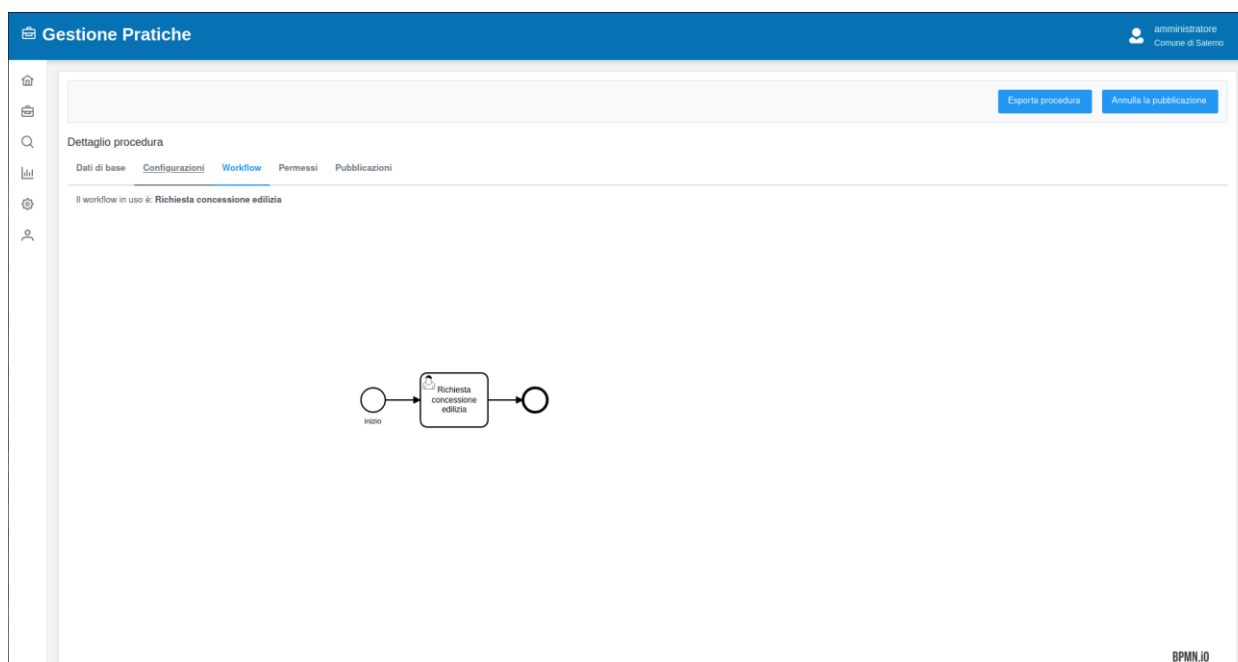


The screenshot shows the 'Gestione Pratiche' interface with the 'Configurazioni' tab selected. The 'Dettagli procedura' section contains the following fields:

- Cartella di salvataggio dei documenti \***: /Root/test/2021/182. A 'Seleziona cartella' button is next to it.
- Percorso di salvataggio personalizzato**: /test/\$/year/\$/day. A help icon (?) is next to it.
- Generatore di numero di protocollo**: protocol number generator.
- Report di procedura**: (empty dropdown menu).

Buttons at the top right: 'Esporta procedura' and 'Annulla la pubblicazione'.

Nel tab **Workflow** si associa uno dei workflow precedentemente definiti nella sezione apposita.



Nel tab **Permessi** si possono impostare i permessi di tutti i campi dei form del workflow e dei documenti della procedura. Tali permessi permettono di definire se campi e documenti sono o meno visibili al cittadino che consulta la pratica\domanda.

- Se il permesso è impostato su **operatore**, le variabili saranno visibili solo lato operatore.
- Se il permesso è impostato su **operatore e cittadino**, le variabili saranno visibili anche lato cittadino.

**Gestione Pratiche**

amministratore  
Comune di Salerno

Esporta procedura Annulla la pubblicazione

Dettaglio procedura

Dati di base Configurazioni Workflow **Permessi** Pubblicazioni

Permessi variabili Permessi documenti

Imposta visibilità per tutte le variabili: Seleziona... Applica

Nome variabile	Visibilità
nome	Cittadino e operatore

Nel tab

**Pubblicazioni**, si possono visualizzare e editare le pubblicazioni di una procedura.

**Gestione Pratiche**

amministratore  
Comune di Salerno

Esporta procedura Annulla la pubblicazione

Dettaglio procedura

Dati di base Configurazioni Workflow Permessi **Pubblicazioni**

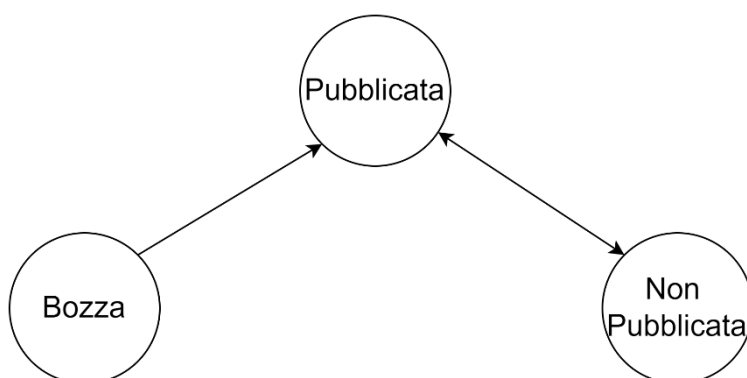
Nome	Versione	Data di pubblicazione
Richiesta concessione edilizia	3	1 lug 2021, 17:33:19
Richiesta concessione edilizia	2	1 lug 2021, 17:14:55
Richiesta concessione edilizia	1	1 lug 2021, 17:07:47

<< < 1 > >>

Una procedura può assumere diversi stati nel corso della sua vita.

- **Bozza**: procedura in fase di definizione da parte dell'operatore;

- **Pubblicata:** procedura disponibile per la richiesta\compilazione da parte dei cittadini. Una procedura in tale stato non è modificabile dall'operatore;
- **Non pubblicata:** procedura rimossa dallo stato di pubblicazione e nuovamente modificabile da parte dell'operatore.



Ogni qualvolta un operatore pubblica una procedura, viene generata una nuova pubblicazione della stessa (una nuova "versione"). Tali pubblicazioni sono disponibili nell'apposita sezione nella pagina di dettaglio della procedura. Le pubblicazioni sono modificabili da parte dell'operatore. Una modifica a una procedura pubblicata deve essere considerata un'operazione ad alto rischio, poichè può introdurre errori nelle procedure già aperte.

Una procedura può essere esportata e importata tramite gli appositi pulsanti presenti nella barra superiore. Queste funzionalità permettono facilmente di creare e collaudare le procedure su un sistema (es. collaudo) e trasferirle integralmente su un altro sistema (es. produzione).

## 11 CONVERTITORE DI DOCUMENTI

I convertitori di documenti sono trasformazioni che vengono applicate ai documenti caricati da cittadino od operatori. Queste trasformazioni producono un ulteriore documento il cui risultato viene salvato nella cartella della procedura nel sistema documentale interfacciato a Gestione Pratiche.

Vi è la possibilità di creare un convertitore per convertire i documenti da un formato di origine a un formato di destinazione. Sono obbligatorie le seguenti proprietà di un convertitore:

1. **Nome:** il nome del convertitore
2. **Tipo:** il tipo di convertitore
3. **Template:** il file che rappresenta il foglio di stile in formato .xsl

Una volta definito il convertitore, è possibile testarlo, caricando un file e verificando il risultato della trasformazione. Questa procedura può essere usata per provare il risultato della trasformazione senza aver la necessità di applicarlo a una pratica.

Per utilizzare il convertitore bisogna selezionarlo dalle proprietà del campo di tipo file del form di un workflow.

Archiviazione ?

Gestione pratiche storage provider



Nome da assegnare al file ?

documento

Convertitore documenti ?



### Convertitore di PDF\XSL

Allo stato attuale l'unico tipo di convertitore disponibile è il convertitore PDF\XML che permette, attraverso l'applicazione di un foglio di trasformazione XSL, di produrre un file XML a partire da un PDF.

Nome

Tipo di convertitore

PDF/XML

Selezione un file

Xsl charset: 

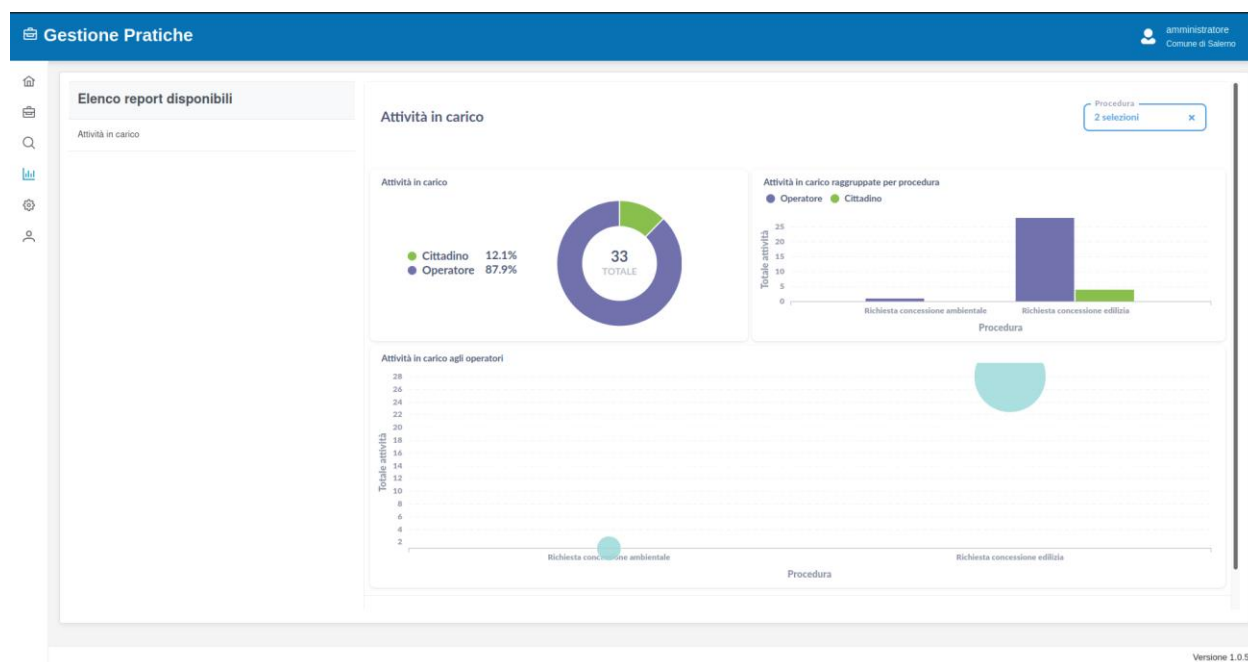
Select a charset

Xsl template

Questo convertitore richiede il caricamento di un foglio di trasformazione in formato XSL e della scelta del charset usato da tale foglio.

## 12REPORT

Nella sezione Report è possibile visualizzare i report definiti per l'organizzazione (ad esempio le attività in carico).



## 13 RICERCA GLOBALE

Gestione Pratiche integra una ricerca globale. Attraverso diversi filtri la ricerca viene estesa su procedure, attività e documenti. Sono disponibili i seguenti filtri:

- 1) **Ricerca nel contenuto:** ricerca per nome, descrizione di una procedura di un'attività, e ricerca nel contenuto di un documento.
- 2) **Numero di protocollo:** ricerca per numero di protocollo.
- 3) **Codice Fiscale:** ricerca per codice fiscale del richiedente.
- 4) **Nome:** ricerca per nome del richiedente.
- 5) **Cognome:** ricerca per cognome del richiedente.
- 6) **Data d'inizio:** ricerca per data d'inizio per procedure, attività o data di salvataggio di un documento.
- 7) **Data di fine:** ricerca per data di fine per procedure, attività o data di salvataggio di un documento.

Gestione Pratiche

amministratore  
Comune di Salerno

Filter di ricerca

Cerca nei contenuti delle pratiche e dei documenti

Ulteriori filtri di ricerca

Tipo di pratica

Richiesta concessione edilizia

Numero di protocollo

Codice Fiscale del richiedente

Nome del richiedente

Cognome del richiedente

Data di inizio

01-07-2021

Data di fine

08-07-2021

Ricerca

Pulisci

Risultati nelle pratiche 1

Risultati nelle attività 0

Risultati nei documenti 0

Numero di protocollo	Codice fiscale	Richiedente	Nome	Data di creazione	Ultimo aggiornamento	Stato
LX1UWGYDT	RSSPLA80A01H501T	Paolo Rossi	Richiesta concessione edilizia	1 lug 2021, 17:08:40	1 lug 2021, 17:08:40	aperto
73P3CE1GCQ	RSSPLA80A01H501T	Paolo Rossi	Richiesta concessione edilizia	1 lug 2021, 17:11:52	1 lug 2021, 17:11:52	aperto
D6M4UPKXT	RSSPLA80A01H501T	Paolo Rossi	Richiesta concessione edilizia	1 lug 2021, 17:14:59	1 lug 2021, 17:14:59	aperto
3M6HXCK9SR	RSSPLA80A01H501T	Paolo Rossi	Richiesta concessione edilizia	1 lug 2021, 17:17:43	1 lug 2021, 17:17:43	aperto
1QKRJMF85	RSSPLA80A01H501T	Paolo Rossi	Richiesta concessione edilizia	1 lug 2021, 17:31:42	1 lug 2021, 17:31:42	aperto

Versione 1.0.5


## 14 VISTA ATTIVITÀ

Questa vista permette di visualizzare tutte le attività\task di tipo utente disponibili e lavorabili in Gestione Pratiche.

La vista attività si compone di due tab:

- 1) **Attività prese in carico:** tutte le attività che l'operatore ha preso in carico;
- 2) **Attività disponibili:** tutte le attività "libere" (non prese in carico da nessun altro operatore) per tutte le pratiche che possono essere visualizzate dall'operatore.

Elenco attività							
Attività prese in carico 0   Attività disponibili 22							
Numero di protocollo	Pratica	Codice fiscale	Richiedente	Nome	Data di creazione	In carico	
39-2021 (16.1)	[Demo] Richiesta di attribuzione/verifica del numero civico	RSSPLA80A01H501T	Paolo Rossi	Validazione Manuale	9 lug 2021, 16:02:07	No	
38-2021 (16.1)	[Demo] Richiesta di attribuzione/verifica del numero civico	RSSPLA80A01H501T	Paolo Rossi	Validazione Manuale	9 lug 2021, 15:58:34	No	
37-2021 (16.1)	[Demo] Richiesta di attribuzione/verifica del numero civico	RSSPLA80A01H501T	Paolo Rossi	Validazione Manuale	5 lug 2021, 17:03:21	No	
36-2021 (16.1)	[Demo] Richiesta di attribuzione/verifica del numero civico	RSSPLA80A01H501T	Paolo Rossi	Validazione Manuale	1 lug 2021, 11:02:12	No	
-	-	-	-	Validazione Manuale	29 giu 2021, 15:20:53	No	
-	-	-	-	Validazione Manuale	29 giu 2021, 15:18:09	No	
-	-	-	-	Validazione Manuale	29 giu 2021, 15:17:55	No	
-	-	-	-	Validazione Manuale	29 giu 2021, 15:01:15	No	

Dalla vista "Elenco attività", andando a premere il tasto  è possibile visualizzare il dettaglio di una singola attività.

In particolare, il dettaglio di un'attività utente è organizzato in diverse sezioni (tab), tra cui:

- 1) **Nome dell'attività:** è il contenuto del form\modulo associato all'attività durante la fase di disegno del workflow. I campi del form non sono editabili finché l'attività non viene presa in carico.

### Dettaglio Attività

[Modifica Dati](#)[Variabili](#)[Documenti](#)[Workflow](#)[Sottomissioni](#)



#### Dettaglio Attività

Modifica Dati **Variabili** Documenti Workflow Sottomissioni

Nome	Tipo	Valore
------	------	--------

- 3) **Documenti:** i documenti che sono stati inviati fino a quel momento dal cittadino o dall'operatore. Qui è specificato il nome, la dimensione, la data di caricamento e la versione

#### Dettaglio Attività

Modifica Dati Variabili **Documenti** Workflow Sottomissioni

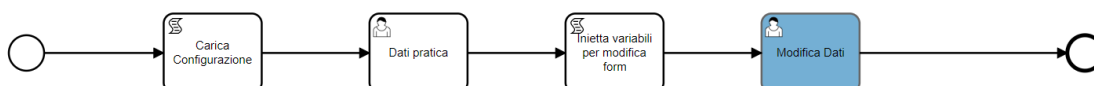
Nome	Dimensione (MB)	Data di caricamento	Versione
------	-----------------	---------------------	----------

- 4) **Workflow:** il diagramma del workflow con evidenziata l'attività che si sta visualizzando. In figura è mostrato un workflow d'esempio:



#### Dettaglio Attività

Modifica Dati Variabili Documenti **Workflow** Sottomissioni



- 5) **Sottomissioni:** le sottomissioni della procedura, quindi tutte le attività che sono state prese in carico e concluse fino a quel momento. Rappresenta, di fatto, il logbook della procedura. Ogni sottomissione permette di visualizzare i dati dell'utente (cittadino od operatore) che l'ha operata, i valori dei campi del form/modulo compilato, i documenti allegati e il modulo così come è stato riempito dall'utente

Dettaglio sottomissione

Dati sottomissione

Variabili

Documenti

Modulo

Nome

slavik

Cognome

prova

Codice Fiscale

Dati di nascita

Provincia di Nascita

SA - Salerno

Comune

Salerno

Data di nascita

06/05/1986

Data di nascita

6

Maggio

1986

Salva

Dati di Residenza

Provincia di Residenza

SA - Salerno

Comune









Valore Predefinito

Indirizzo

via antonio amato 0912

## 15 VISTA STORICO PRATICHE

Lo storico di una pratica permette di elencare e consultare tutte le pratiche presenti nel sistema in base ai filtri di visualizzazione applicati all'utente.

Elenco storico pratiche							
Numero di protocollo	Codice fiscale	Richiedente	Pratica	Data di inizio	Data ultima modifica	Stato	
39-2021 (16.1)	RSSPLA80A01H501T	Paolo Rossi	[Demo] Richiesta di attribuzione/verifica del numero civico	9 lug 2021, 16:02:07	9 lug 2021, 16:02:07	aperto	
38-2021 (16.1)	RSSPLA80A01H501T	Paolo Rossi	[Demo] Richiesta di attribuzione/verifica del numero civico	9 lug 2021, 15:58:34	9 lug 2021, 15:58:34	aperto	
37-2021 (16.1)	RSSPLA80A01H501T	Paolo Rossi	[Demo] Richiesta di attribuzione/verifica del numero civico	5 lug 2021, 17:03:21	5 lug 2021, 17:03:21	aperto	
36-2021 (16.1)	RSSPLA80A01H501T	Paolo Rossi	[Demo] Richiesta di attribuzione/verifica del numero civico	1 lug 2021, 11:02:12	1 lug 2021, 11:02:12	aperto	
35-2021 (16.1)	RSSPLA80A01H501T	Paolo Rossi	[Demo] Richiesta di attribuzione/verifica del numero civico	29 giu 2021, 17:35:06	29 giu 2021, 17:35:06	chiuso	
34-2021 (16.1)	RSSPLA80A01H501T	Paolo Rossi	[Demo] Richiesta di attribuzione/verifica del numero civico	29 giu 2021, 14:50:38	29 giu 2021, 14:50:38	aperto	
33-2021 (16.1)	RSSPLA80A01H501T	Paolo Rossi	[Demo] Richiesta di attribuzione/verifica del numero civico	29 giu 2021, 14:47:46	29 giu 2021, 14:47:46	aperto	
[Demo] Richiesta di							

Dalla lista pratiche è possibile visualizzare il dettaglio di ogni singola pratica, il quale si compone delle seguenti viste (tab):

- 1) **Dati pratica:** il sommario dei dati di una pratica.
- 2) **Variabili:** le variabili della pratica. Rappresentano i campi dei moduli\form compilati dal cittadino o dall'operatore
- 3) **Documenti:** i documenti della pratica inviati dal cittadino o dall'operatore.
- 4) **Attività:** le attività disponibili e le attività che sono state prese in carico dagli operatori nel momento in cui si visualizzano i dettagli della pratica.
- 5) **Workflow:** il diagramma del workflow con evidenziate le attività sulle quali il processo\workflow è attualmente in attesa.

- 6) Sottomissioni: le sottomissioni della procedura, quindi tutte le attività che sono state lavorate fino a quel momento. Rappresenta il logbook della pratica.

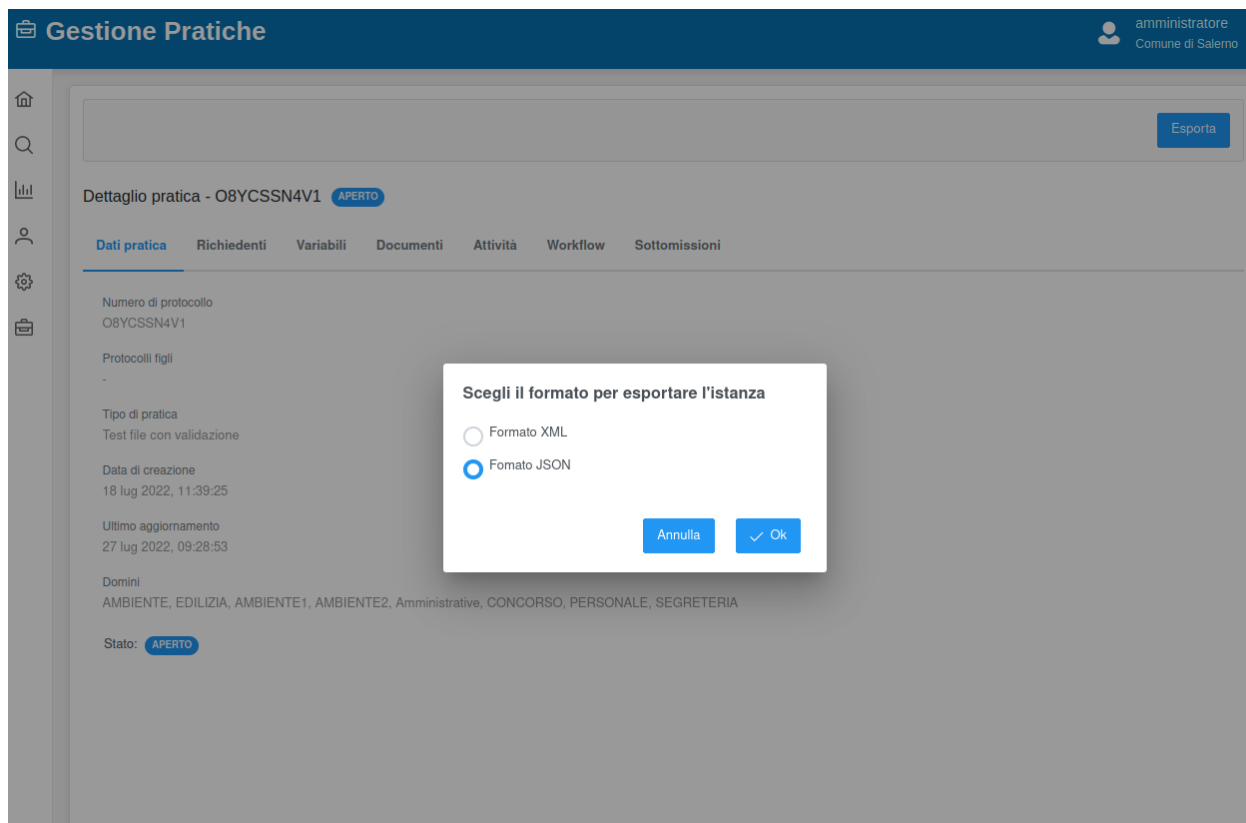
The screenshot displays the 'Gestione Pratiche' interface. The top navigation bar is blue with the title 'Gestione Pratiche' and a user profile 'amministratore Comune di Salerno'. A sidebar on the left contains icons for home, search, charts, users, settings, and a briefcase. The main content area shows the 'Dettaglio pratica - O8YCSSN4V1' with a status badge 'APERTO'. Below this is a tabbed interface with 'Dati pratica' selected. The 'Dati pratica' tab displays the following information:

- Numero di protocollo: O8YCSSN4V1
- Protocolli figli: -
- Tipo di pratica: Test file con validazione
- Data di creazione: 18 lug 2022, 11:39:25
- Ultimo aggiornamento: 27 lug 2022, 09:28:53
- Domini: AMBIENTE, EDILIZIA, AMBIENTE1, AMBIENTE2, Amministrative, CONCORSO, PERSONALE, SEGRETERIA
- Stato: APERTO

An 'Esporta' button is located in the top right corner of the main content area.

E' possibile esportare il dettaglio della pratica, quindi contiene una descrizione completa della pratica, i suoi domini, e le variabili di istanza. I formati disponibili sono:







- 1) formato JSON
- 2) formato XML



Gestione Pratiche permette di impersonificare le azioni del cittadino. Quando la procedura viene pubblicata, all'utente viene chiesto se pubblicare la procedura:

- 1) **internamente/test**: in questo stato la procedura è visibile solo in Gestione Pratiche e non nel Portale Istituzionale.
- 2) **pubblicata**: in questo stato la procedura è visibile anche nel Portale Istituzionale

Le pratiche pubblicate internamente sono visibili attraverso una sezione apposita che si chiama **Le mie pratiche**:








**Gestione Pratiche**







amministratore  
Comune di Salerno

Inizia Pratica

**Le mie pratiche**

Numero di protocollo	Codice fiscale	Richiedente	Pratica	Data di inizio	Data ultima modifica	Stato	
1VNUQVQSTD	-	amministratore	test multipli	28 lug 2022, 11:30:01	28 lug 2022, 11:30:12	APERTO	
O8YCSSN4V1	-	amministratore	Test file con validazione	18 lug 2022, 11:39:25	27 lug 2022, 09:28:53	APERTO	
VUBLN5KZG2	-	amministratore	test multipli	26 lug 2022, 17:39:55	26 lug 2022, 17:40:40	APERTO	
4EZBMBESXP	-	amministratore	Test file con validazione	18 lug 2022, 12:38:30	26 lug 2022, 16:51:22	APERTO	
4EZBMBESXP	-	amministratore	test multipli	19 lug 2022, 12:54:06	26 lug 2022, 16:47:53	APERTO	

E' possibile iniziare nuove pratiche impersonificando il cittadino.



**Gestione Pratiche**





amministratore  
Comune di Salerno

Inizia Pratica

**Le mie pratiche**

**Elenco procedure**

Tipo  
Pubblicata internamente

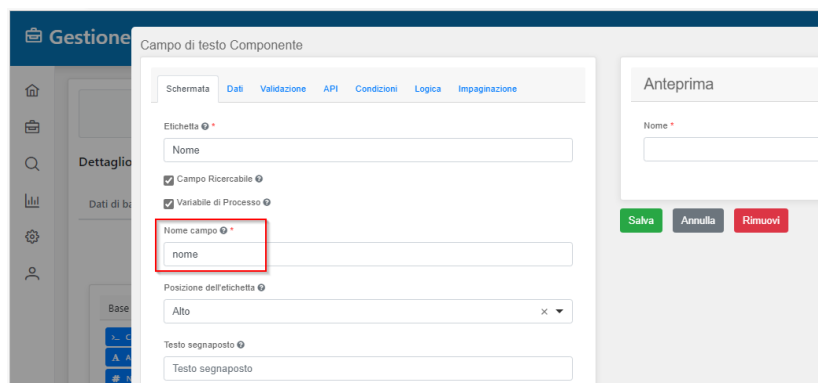
Nome	Data di creazione	Data ultima modifica	
Richiesta di partecipazione al Concorso	22 lug 2022, 09:47:55	22 lug 2022, 09:59:47	
Valutazioni Commissario	22 lug 2022, 09:49:49	22 lug 2022, 09:49:59	
Test file con validazione	15 lug 2022, 12:57:53	19 lug 2022, 09:58:02	
test multipli	27 giu 2022, 17:10:30	5 lug 2022, 15:17:38	

<< < 1 > >>

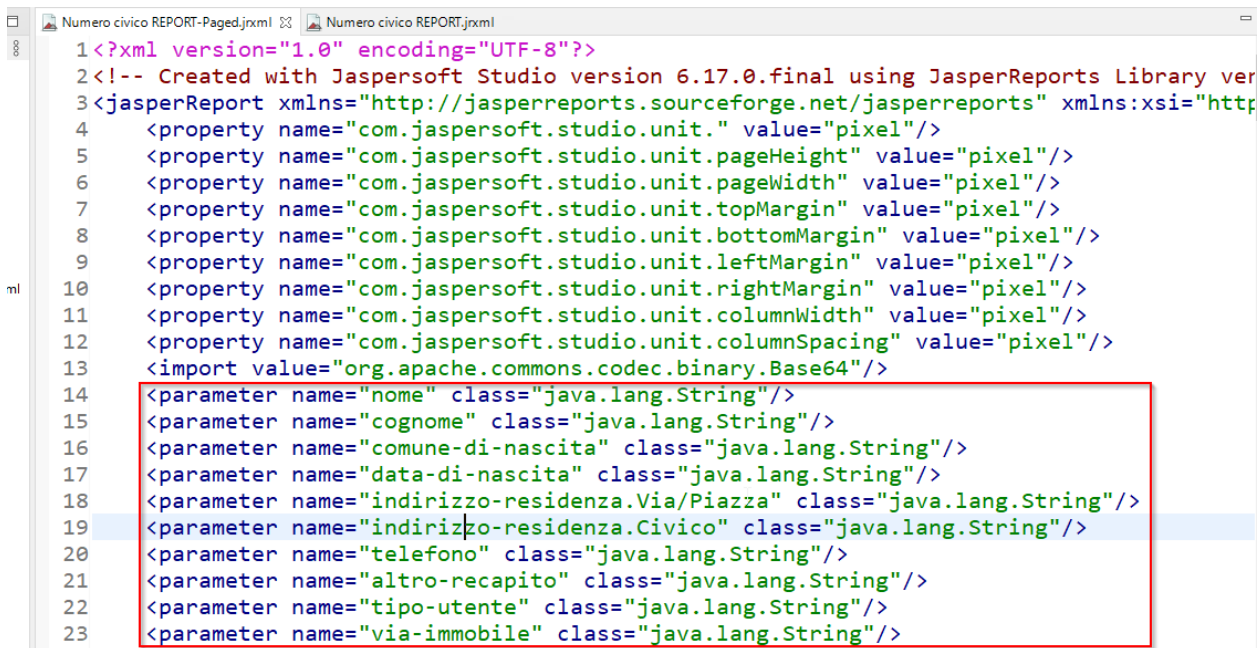
## 16REPORT DI PROCEDURE (PDF)

Gestione Pratiche permette di creare un report di procedura, permette cioè la creazione automatizzata di documenti PDF che vengono allegati alla procedura al suo avvio e\o in fasi successive (definite dal workflow).

Per creare un report abbiamo la necessità di conoscere gli id dei campi che vogliamo inserire nel report. A questo proposito, quando creiamo il form, prendiamo nota dei nomi dei campi che vogliamo utilizzare per il report.



Una volta aperto Jaspersoft Studio, va creato un file di report e vanno specificati quali sono i parametri di input che deve gestire, ovvero i nomi dei campi che vogliamo esportare.



```
1<?xml version="1.0" encoding="UTF-8"?>
2<!-- Created with Jaspersoft Studio version 6.17.0.final using JasperReports Library ver
3<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:xsi="http
4  <property name="com.jaspersoft.studio.unit." value="pixel"/>
5  <property name="com.jaspersoft.studio.unit.pageHeight" value="pixel"/>
6  <property name="com.jaspersoft.studio.unit.pageWidth" value="pixel"/>
7  <property name="com.jaspersoft.studio.unit.topMargin" value="pixel"/>
8  <property name="com.jaspersoft.studio.unit.bottomMargin" value="pixel"/>
9  <property name="com.jaspersoft.studio.unit.leftMargin" value="pixel"/>
10 <property name="com.jaspersoft.studio.unit.rightMargin" value="pixel"/>
11 <property name="com.jaspersoft.studio.unit.columnWidth" value="pixel"/>
12 <property name="com.jaspersoft.studio.unit.columnSpacing" value="pixel"/>
13 <import value="org.apache.commons.codec.binary.Base64"/>
14 <parameter name="nome" class="java.lang.String"/>
15 <parameter name="cognome" class="java.lang.String"/>
16 <parameter name="comune-di-nascita" class="java.lang.String"/>
17 <parameter name="data-di-nascita" class="java.lang.String"/>
18 <parameter name="indirizzo-residenza.Via/Piazza" class="java.lang.String"/>
19 <parameter name="indirizzo-residenza.Civico" class="java.lang.String"/>
20 <parameter name="telefono" class="java.lang.String"/>
21 <parameter name="altro-recapito" class="java.lang.String"/>
22 <parameter name="tipo-utente" class="java.lang.String"/>
23 <parameter name="via-immobile" class="java.lang.String"/>
```

È possibile definire i parametri in base alla tipologia di campo, in particolare per i campi di test o di selezione abbiamo il tipo String, mentre per i valori tipo Checkbox, abbiamo il tipo Boolean, ad esempio:

```
<parameter name="nome" class="java.lang.String"/>
<parameter name="cognome" class="java.lang.String"/>

<parameter name="richiesta-copia" class="java.lang.Boolean"/>
<parameter name="allegato-incluso" class="java.lang.Boolean"/>
```

Nel caso dell'estensione Form.IO per il campo Indirizzo per il Comune di Salerno, ogni sottocampo è un parametro a sé del report.

Ad esempio nel caso di un campo Indirizzo con name = "indirizzo-residenza" si procede come segue

```
<parameter name="indirizzo-residenza.Via/Piazza" class="java.lang.String"/>
<parameter name="indirizzo-residenza.Civico" class="java.lang.String"/>
<parameter name="indirizzo-residenza.Città" class="java.lang.String"/>
<parameter name="indirizzo-residenza.Provincia" class="java.lang.String"/>
```

Per utilizzare i campi è sufficiente a questo punto utilizzare la sintassi seguente:

Il/La sottoscritto/a (cognome – nome) `$P{cognome}+" "+$P{nome}`

È possibile utilizzare qualsiasi tipo di espressione:

```
<textFieldExpression><![CDATA[$P{cognome}+" "+$P{nome}]]></textFieldExpression>
```

Un volta creato il report in configurazione "Report di Procedure", sarà sufficiente selezionarlo alla procedura nel tab "Configurazioni".

Il report di procedura si compone della seguente vista:

- 1) **Nome:** nome del report.
- 2) **File:** il file template in formato .jrxml (Jasper Reports) che verrà utilizzato per generare il report di una procedura.



Versione 1.0.5

Nel tab **Configurazioni** della procedura è necessario selezionare il template di report da utilizzare ed anche il nome da assegnare al file report che verrà generato.

Report di procedura

Report

Nome del report \*

Documento

## 16.1 Generatore PDF

Il generatore del PDF è un elemento chiave del sistema e può essere invocato anche **direttamente dal workflow**. La sua funzione principale è quella di generare un documento PDF che rispecchi le informazioni inserite nell'istanza della pratica.

Il generatore ha la capacità di accedere a tutte le variabili presenti nell'istanza, incluse quelle che provengono dal form web e quelle che sono associate ai PDF editabili. Questo significa che il generatore ha una vista completa delle informazioni raccolte e può utilizzare queste informazioni per creare un PDF che soddisfi le esigenze dell'utente.

Il workflow viene avviato al momento in cui l'istanza viene salvata, ovvero al submit della maschera iniziale della pratica. Tuttavia, a causa di questa configurazione, non è possibile visualizzare il PDF generato direttamente nella maschera iniziale. Sarà necessario accedere a un'altra sezione del sistema per visualizzare il PDF completato, ad esempio dal backoffice nella sezione documenti, oppure dal portale utilizzando le API del gestore pratiche per recuperare l'elenco dei documenti.

Il PDF viene costruito a partire da un file Jasper Report, che è uno strumento open source progettato appositamente per la creazione di report. Questo file viene utilizzato come base per la generazione del PDF e viene personalizzato in base alle informazioni raccolte dall'istanza. Tuttavia, è importante notare che il sistema descritto non permette di iniettare direttamente i valori all'interno di un PDF editabile. Pertanto il PDF generato è statico e non modificabile. Ciò significa che i PDF generati dal sistema descritto non potranno essere modificati direttamente dall'utente. Il flusso di creazione del report PDF è mostrato nel sequence di seguito:

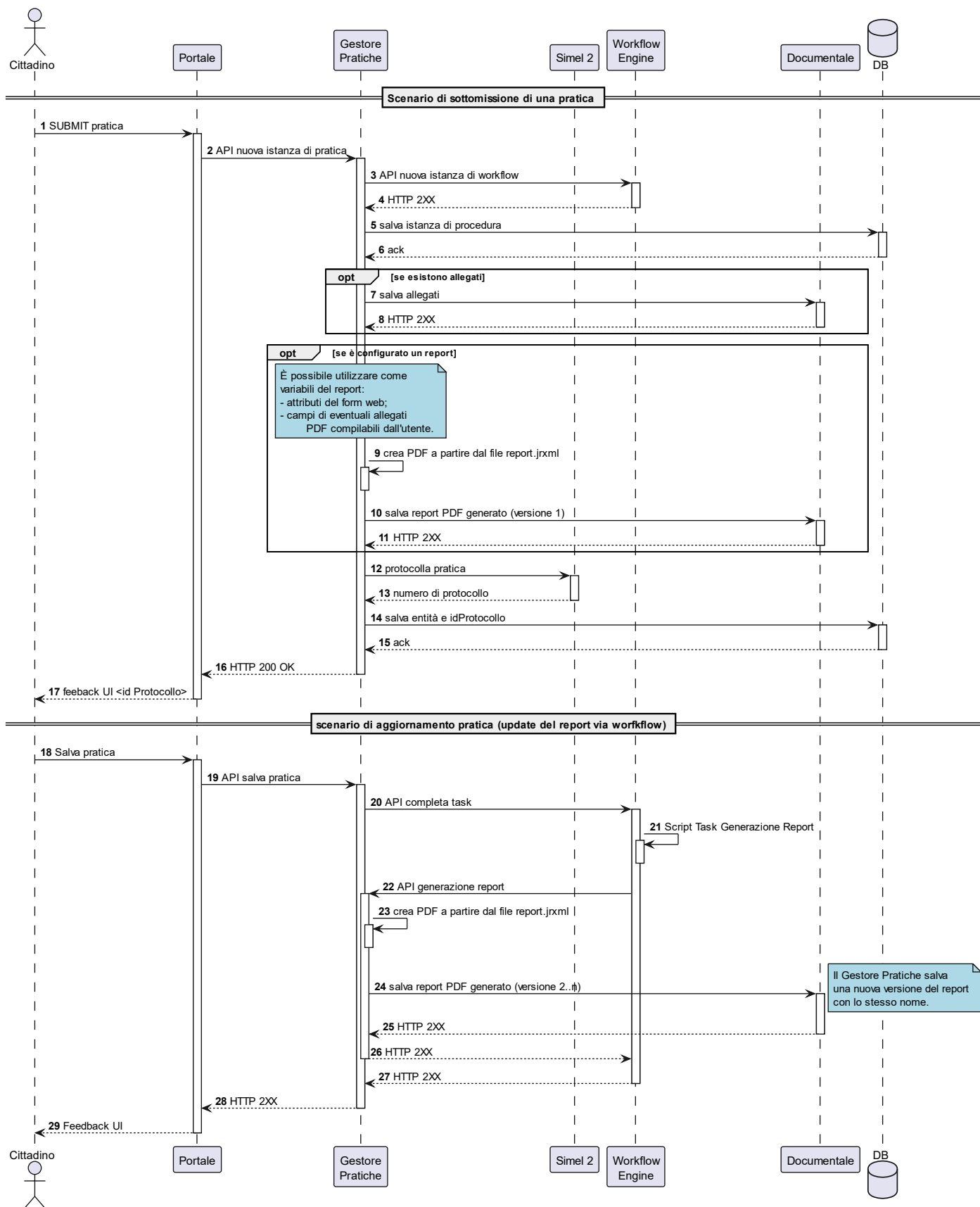


Figura 19 - Scenario generazione report

Nel primo scenario di sottomissione di una pratica, il cittadino invia una richiesta di submit della pratica attraverso il portale. Il portale invoca una nuova istanza di pratica tramite un'API per il gestore pratiche, che a sua volta invoca una nuova istanza di workflow tramite un'altra API. Se esistono allegati, il gestore pratiche li salva nel documentale e successivamente salva l'istanza di procedura nel database.

Se è configurato un report, il gestore pratiche crea il PDF a partire dal file report.jrxml utilizzando i campi del form web, eventuali allegati e i PDF compilabili dall'utente. Il PDF generato viene successivamente salvato nel documentale. La pratica viene poi protocollata da Simel 2 e il numero di protocollo viene salvato nel database. Il portale invia un feedback all'utente con l'ID del protocollo.

Nel secondo scenario di aggiornamento pratica, il cittadino salva la pratica attraverso il portale. Il portale invoca un'API di salvataggio per il gestore pratiche, che completa il task attuale tramite un'API per l'engine del workflow e genera un nuovo report tramite un'altra API per il gestore pratiche. Il gestore pratiche crea una nuova versione del PDF a partire dal file report.jrxml e la salva nel documentale. Il feedback viene successivamente inviato all'utente tramite il portale.

Nel secondo scenario, quindi, se abbiamo configurato una nuova generazione del report PDF utilizzando le API invocate tramite un Service Task, possiamo pilotare a nostro piacimento la generazione di una nuova versione del report. Vedi ad esempio un estratto del workflow di seguito, dove il cittadino può modificare delle informazioni e il sistema, tramite SCRIPT, genera un nuovo report:

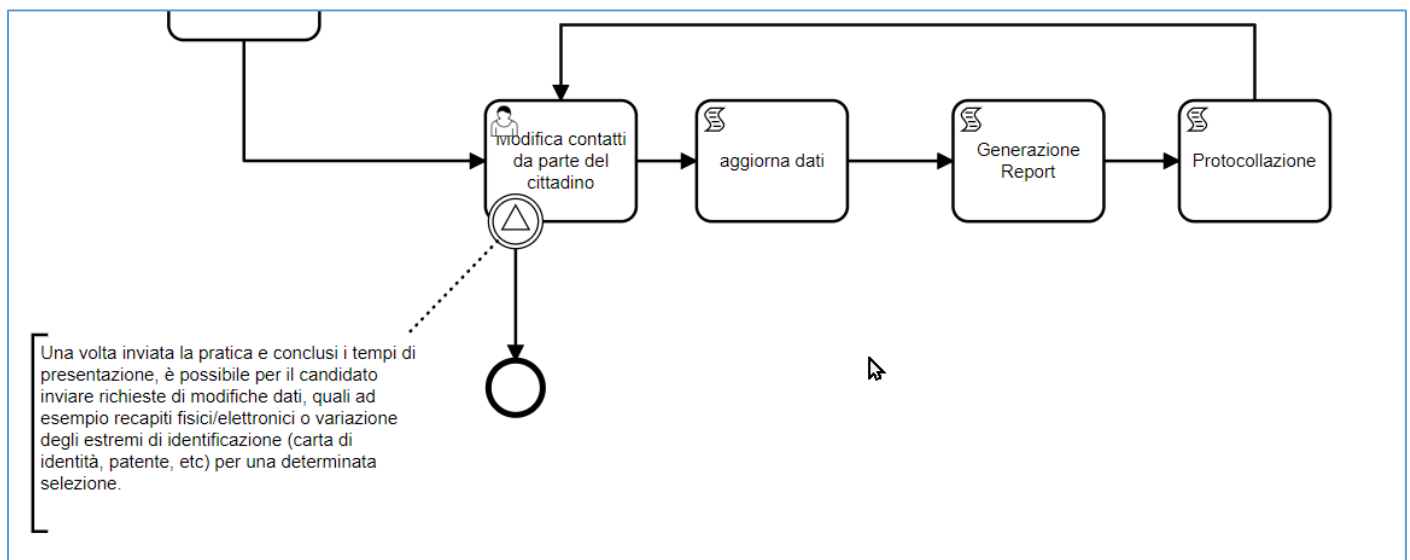


Figura 20 - Esempio di generazione report utilizzando uno SCRIPT task e le API messe a disposizione dal Gestore Pratiche.

Il codice di seguito presente nel Service Task di Generazione Report, permette di generare una nuova versione del report:

generateReport.groovy

```

import groovy.json.*
import org.camunda.bpm.engine.impl.persistence.entity.ExecutionEntity
import org.camunda.bpm.engine.impl.persistence.entity.TaskEntity

```

```
def endpointURL = 'http://10.0.254.7/gestionepratiche/api/bpm/procedure/report/generate'
def httpConnector = org.camunda.connect.Connectors.http()

def payload = JsonOutput.toJson([ 'externalProcedureInstanceId': execution.getProcessInstanceId(),
                                   //'resolveNameConflicts': false
                                   ])

println('Payload          : ' + payload)
println('Endpoint        : ' + endpointURL)

try {
    def resp = httpConnector.createRequest()
    .post()
    .header('gptenantid', '1')
    .header('Accept', 'application/json')
    .header('origin', 'http://camunda')
    .header("Access-Control-Expose-Headers", "gpaauthorization")
    .url(endpointURL)
    .contentType('application/json')
    .payload(payload)
    .execute()
    //def result = resp.getResponse()
    println('Response Status Code : ' + resp.getStatusCode())
    println('Response Headers      : ' + resp.getHeaders())
    println('Response Body           : ' + resp.getResponse())
    resp.close()
} catch (Exception ex) {
    println("ERROR: La chiamata REST ha restituito un errore. Il BPMN continua, ma l'errore è loggato di seguito: " +
ex.getMessage())
}
```

Questo codice Groovy è un esempio di Script Task per la generazione di un report all'interno di un processo di business definito nel Gestore Pratiche. Il codice utilizza la libreria HTTP Connector di Camunda per inviare una richiesta HTTP POST a un endpoint che genera il report in formato PDF. Il payload della richiesta include l'ID dell'istanza di procedura corrente. Il codice gestisce anche le eccezioni che possono verificarsi durante la richiesta HTTP e registra eventuali errori che possono verificarsi. In generale, questo codice è flessibile perché può essere modificato per adattarsi a diverse esigenze ed è semplice da modificare a seconda delle esigenze.

## 16.2 Firma del documento generato

Se sussiste la necessità di gestire la firma di questo documento generato dal sistema, una soluzione percorribile è la seguente:

Inserire nel Workflow, come primo task, la generazione del pdf e come secondo task, un task umano assegnato al cittadino che contiene un form con la possibilità di allegare un file. Il cittadino scarica il file, lo firma digitalmente, allega il doc firmato nella maschera e salva.

## 17 GESTIONE DELEGHE

Gestione Pratiche permette di definire i delegati di una pratica. I delegati hanno la stessa visibilità del richiedente (per le pratiche che appartengono al richiedente). Nel dettaglio di una pratica è possibile verificare chi sono i richiedenti (compresi i delegati) per quella pratica:

Gestione Pratiche

amministratore  
Comune di Salerno

Dettaglio pratica - 4EZBMESXP

APERTO

Dati pratica

Richiedenti

Variabili

Documenti

Attività

Workflow

Sottomissioni

Identificativo	Nome	Cognome	Tipo
amministratore	amministratore	amministratore	richiedente

## 18 GESTIONE STATI PROCEDURA

Gestione Pratiche permette di definire gli stati di una procedura. Una istanza di procedura infatti può assumere diversi stati, tra cui alcuni di default.

Uno stato di procedura si compone delle seguenti proprietà:

- 1) **etichetta**: rappresenta il nome dello stato
- 2) **chiave**: rappresenta la chiave univoca per identificare lo stato

Vi sono gli stati di sistema (APERTO, CHIUSO) che non possono essere rimossi e che vengono impostati rispettivamente all'avvio di una pratica e alla chiusura di essa (quando il workflow entra nello stato CLOSED).

Gli stati di una pratica possono essere impostati dinamicamente come descritto nel documento **GestionePratiche Invio segnale verso Workflow Engine/Set stato istanza di procedura.docx**

Gestione Pratiche

admin  
Comune di Salerno

Crea nuovo stato di procedura

Esporta CSV

Lista stati di procedura

Nome

Ricerca

Etichetta	Chiave	Data di creazione	Data ultima modifica	
In_lavorazione	IN_LAVORAZIONE	25 mar 2022, 15:03:27	25 mar 2022, 15:03:27	<div></div> <div></div>
Idle	IDLE	23 mar 2022, 10:32:55	23 mar 2022, 10:32:55	<div></div> <div></div>
Generazione Graduatoria	GENERAZIONE_GRADUATORIA	18 mar 2022, 12:12:09	18 mar 2022, 12:12:09	<div></div> <div></div>
chiuso	CLOSED	17 mar 2022, 17:39:33	17 mar 2022, 17:39:33	<div></div>
aperto	OPEN	17 mar 2022, 17:39:20	17 mar 2022, 17:39:20	<div></div>

## 19 GESTIONE IMPORTAZIONE AVANZATA

Gestione Pratiche permette di creare e/o aggiornare le pratiche in base ad un formato di importazione. Un formato di importazione si compone delle seguenti proprietà:

- 1) **nome**: rappresenta il nome del formato di importazione
- 2) **tipo di importazione**: indica se si vuole creare nuove istanze e aggiornare quelle esistenti in fase di importazione oppure aggiornare solo quelle esistenti.
- 3) **procedura**: rappresente la procedura per la quale si vogliono creare/aggiornare le istanze
- 4) **tipo**: rappresenta il tipo di file da importare, attualmente è supportato il formato CSV
- 5) **separatore**: indica il separatore di riga nel file da importare
- 6) **righe da saltare**: indica quante righe si vogliono saltare durante la fase di importazione.

Il formato di importazione richiede la definizione dei campi che dovranno essere mappati.

Possiamo definire due tipi di campo:

- 1) **proprietà**: sono le proprietà base di una procedura (stato, identificativo/nome/cognome richiedente, numero di protocollo). Le proprietà che identificano la pratica sono il numero di protocollo e l'identificativo del richiedente.
- 2) **variabile**: sono le variabili della procedura. Serve per mappare la variabile definita nel formato con la variabile della procedura.

The screenshot displays the 'Gestione Pratiche' application interface. At the top, there is a dark blue header with the title 'Gestione Pratiche' and a user profile icon labeled 'amministratore Comune di Salerno'. Below the header, a sidebar on the left contains navigation icons. The main content area shows a 'Dettaglio formato importatore pratica' section with tabs for 'Dati di base' and 'Campi'. The 'Campi' tab is active, showing a table with columns 'Colonna' and 'Tipo'. A modal dialog titled 'Aggiungi nuovo campo' is open in the center. This dialog has two dropdown menus: 'Tipo campo' (set to 'Proprietà') and 'Nome campo' (set to 'Identificativo del richiedente'). There is also a checkbox labeled 'Identifica la pratica' which is currently unchecked. At the bottom right of the dialog is a blue button labeled 'Salva'. In the background, there are buttons for 'Salva' and 'Annulla' at the top right, and a '+ Aggiungi nuovo campo' button on the right side of the main area.

Sono ammessi i seguenti tipi di variabile:

**Stringa:** il valore della variabile è una stringa

**Data:** il valore della variabile è una data e ha il seguente formato YYYY-mm-ddTHH:MM:SSSZ

**Si/No:** il valore della variabile è un boolean e assume il seguente valore true/false

**Double:** il valore della variabile è numero a virgola mobile e assume il seguente formato #.#

**Intero:** il valore della variabile è un numero intero

**Long:** il valore della variabile è un numero long

**Short:** il valore della variabile è un numero short

La fase di importazione è un wizard, si compone quindi di vari step:

- 1) selezione pratica: si seleziona la sorgente dati da importare.
- 2) simulazione importazione: viene eseguita una simulazione e vengono mostrati i record che verranno importati e quali invece hanno dato errore.
- 3) importazione: l'ultimo step dove viene eseguita l'importazione.

**Gestione Pratiche** amministratore Comune di Salerno

Crea nuovo formato importazione pratiche

### Wizard di importazione pratica

1 Selezione pratica 2 **Simulazione importazione** 3 Importazione

N° di record validi 0  
N° di record non validi 1

N° di record	Stato	Azione di importazione	protocolNumber
1	Il record non può essere mappato perchè la procedura non è valida	Nessuna azione. Precondizioni non rispettate.	O8YCASN4V1

Avanti



## 20PROTOCOLLAZIONE INTERNA

Gestione Pratiche permette di visualizzare i protocolli “figli” (protocollo interno) per ciascuna pratica.

Nel dettaglio di una pratica, nel tab “Dati pratica” è possibile visualizzare i protocolli figli generati dal servizio descritto in **GestionePratiche protocolloazione con service task.docx**.

Gestione Pratiche

amministratore  
Comune di Salerno

Dettaglio pratica - BUD8OREH6U APERTO

Dati pratica

Richiedenti

Variabili

Documenti

Attività

Workflow

Sottomissioni

Numero di protocollo  
BUD8OREH6U

Protocolli figli  
81585A28JC

Tipo di pratica  
Test multiprotocolloazione

Data di creazione  
28 lug 2022, 16:28:30

Ultimo aggiornamento  
28 lug 2022, 16:28:59

Domini  
AMBIENTE, EDILIZIA, Amministrative

Stato: APERTO

Esporta

## 21 CARICAMENTO DINAMICO DI UNA LISTA

In questa guida descriviamo i passi per poter disegnare, mediante il componente Form.io, una maschera dinamica che recupera le informazioni da un servizio remoto.

Implementiamo il classico modulo "Provincia/Comune". I dati delle Province e dei Comuni vengono caricati via REST API.

Supponiamo di avere a disposizione il servizio REST per il caricamento dell'elenco delle Province:

```
curl https://mgdlab.it/mock/province.php
```

```
[  
  {  
    "nome": "Agrigento",  
    "sigla": "AG",  
    "regione": "Sicilia"  
  },  
  {  
    "nome": "Alessandria",  
    "sigla": "AL",  
    "regione": "Piemonte"  
  },  
  ...  
]
```

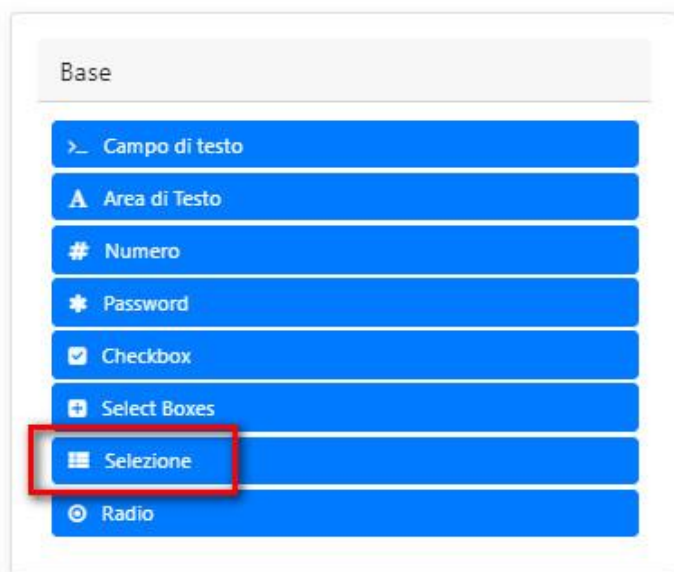
Ed esiste inoltre un servizio per il caricamento dei comuni:

```
curl https://mgdlab.it/mock/comuni.php?p=AG
```

```
["Agrigento", "Alessandria della  
Rocca", "Aragona", "Bivona", "Burgio", "Calamonaci", "Caltabellotta", "Camastra", "Cammarata", ...]
```

Nella schermata di Form.io effettuare i seguenti passi:

1. Inserimento del campo di selezione per la Provincia



2. Inserire Etichetta e Nome campo

Schermata **Dati** Validazione AF

Etichetta <sup>?</sup>\*

Provincia di Nascita

☐ Variabile di Processo <sup>?</sup>

Nome campo <sup>?</sup>\*

provinciaNascita

- Impostare l'ID che deve assumere il valore del campo di Selezione. In questo caso impostiamo "sigla" perché il servizio restituisce tale campo come sigla della provincia.

ID Path <sup>?</sup>

sigla

Value Property <sup>?</sup>

sigla

The selected item's property to save.

- Impostare l'url del servizio:

Selezione Componente

Schermata **Dati** <sup>1</sup> Validazione API Condizioni Logica Impaginazione

☐ Valori Multipli <sup>?</sup>

Tipo di Dati Sorgente <sup>?</sup> <sup>2</sup>

URL

Valore Predefinito <sup>?</sup>

Valore Predefinito

Data Source URL <sup>?</sup> <sup>3</sup>

http://mgdlab.it/mock/province.php

- Definire il layout di visualizzazione per il frontend

Modello oggetto ?

```
i 1 <span>{{ item.sigla }} - {{ item.nome }}</span>
```

6. Creare un nuovo campo di selezione per i Comuni e impostare il relativo URL per filtrare i valori a seconda della provincia selezionata:

Schermata   Dati   Validazione   API   Condizioni   Logica   Impaginazione

☐ Valori Multipli ?

Tipo di Dati Sorgente ?

URL

Valore Predefinito ?

Valore Predefinito

Data Source URL ?

http://mgdlab.it/mock/comuni.php?p={{ row.provincia }}

☒ Lazy Load Data ?

7. Impostare il layout

Modello oggetto ?

```
i 1 <span>{{ item.label }}</span>
```

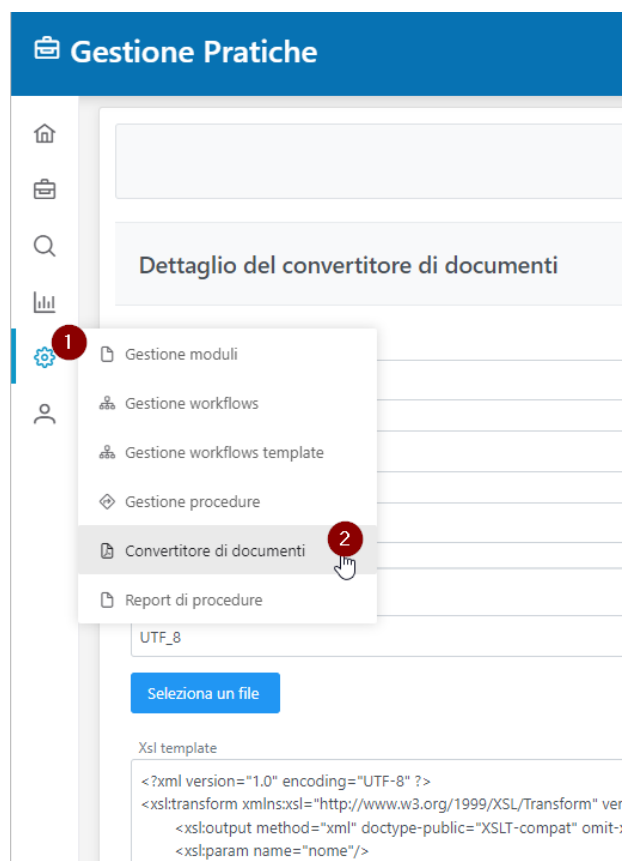
8. Impostare l'aggiornamento dei valori del campo Comune affinché vengano riletti quando si aggiorna il campo Provincia

Aggiorna Opzioni su ?

Provincia preferita per l'impiego

## 22 CONFIGURAZIONE DI UN NUOVO CONVERTITORE PDF/XML

Per configurare un nuovo convertitore PDF/XML utilizzare il seguente menù:



Una volta cliccato il tasto per la creazione di un nuovo convertitore, dovremo riempire i campi mostrati, come nella schermata di esempio:

## Gestione Pratiche

admin  
Comune di Salerno



Test Salva Annulla

### Dettaglio del convertitore di documenti

Nome

Nome Convertitore

Identificativo

convertitore-v01

Tipo di convertitore

PDF/XML

Charset utilizzato dall'Xsl

UTF\_8

Seleziona un file

Xsl template

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:output method="xml" doctype-public="XSLT-compat" omit-xml-declaration="yes" encoding="UTF-8" indent="yes"/>
  <xsl:param name="nome"/>
  <xsl:param name="cognome"/>
  <xsl:param name="data"/>
  <xsl:template match="/">
    <xsl:apply-templates select="rootDocument"/>
  </xsl:template>
  <xsl:template match="rootDocument">
    <contratto>
      <xsl:attribute name="nome">
        <xsl:value-of select="$nome"/>
      </xsl:attribute>
      <xsl:attribute name="cognome">
```

## 22.1 Creazione di un PDF compilabile

Il PDF editabile nasce per semplificare la vita all'amministratore che deve disegnare la pratica e alleggerire la maschera WEB che, in alcuni casi, potrebbe diventare molto complessa da gestire.

Per eseguire questa funzionalità, prepariamo un PDF che abbia dei campi compilabili dall'utente, ad esempio:

**Comune di Salerno - Test Moduli Piattaforma**

Nome:  Cognome:

Codice Fiscale:  Data di Nascita:

Luogo di Nascita:  e-Mail:

Servizi Scelti:

- ☒ s1 Installazione
- ☒ s2 Configurazione
- ☒ s3 Verifica di congruità
- ☒ s4 Integrazione sistemi esterni
- ☒ s5 Test di curvatura 9.99

Vuoi ricevere delle notifiche periodiche?

☒ Si ☒ No ☒ Ricordamelo più tardi

**mc** Seleziona le modalità:

- Abilità Transfascia:
- Fotonica tradizionale:
- Phaser:

Seleziona Anticipo in Euro:

**ALLINEA** **CENTRA**

**DIMENSIONI UGUALI** **DISTRIBUISCI**

**CAMPI**

- ☒ nome
- ☒ cognome
- ☒ codiceFiscale
- ☒ dataDiNascita
- ☒ luogo
- ☒ email
- ☒ s1
- ☒ s2
- ☒ s3
- ☒ s4
- ☒ s5
- ☒ notifiche
  - ☒ true
  - ☐ false
  - ☐ remind
- ☒ mode

[Distribuisci...](#)

[Individua...](#)

## 22.2 Creazione di un XSL

Nell'XSL di esempio, prendiamo in esame alcuni attributi del PDF compilato dall'utente.

È possibile utilizzare diversi PDF in input associando la corrispondente variabile impostata nel modulo FORM.IO.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:output method="xml" doctype-public="XSLT-compat" omit-xml-declaration="yes"
encoding="UTF-8" indent="yes"/>
  <xsl:param name="nome"/>
  <xsl:param name="cognome"/>
  <xsl:param name="data"/>
  <xsl:template match="/">
    <xsl:apply-templates select="rootDocument"/>
  </xsl:template>
```

```

<xsl:template match="rootDocument">
  <contratto>
    <xsl:attribute name="nome">
      <xsl:value-of select="$nome"/>
    </xsl:attribute>
    <xsl:attribute name="cognome">
      <xsl:value-of select="$cognome"/>
    </xsl:attribute>
    <xsl:attribute name="data">
      <xsl:value-of select="$data"/>
    </xsl:attribute>
    <form>
      <nome>
        <xsl:value-of select="$nome"/>
      </nome>
      <cognome>
        <xsl:value-of select="$cognome"/>
      </cognome>
      <data>
        <xsl:value-of select="$data"/>
      </data>
    </form>
    <dati-intestatario>
      <nome>
        <xsl:value-of select="//documentContent[@id='modulo-
compilato']/nome"/>
      </nome>
      <cognome>
        <xsl:value-of select="//documentContent[@id='modulo-
compilato']/cognome"/>
      </cognome>
      <nome-referente>
        <xsl:value-of select="//documentContent[@id='modulo-
compilato2']/nome"/>
      </nome-referente>
      <cognome-referente>
        <xsl:value-of select="//documentContent[@id='modulo-
compilato2']/cognome"/>
      </cognome-referente>
      <cf>
        <xsl:value-of select="//documentContent[@id='modulo-
compilato']/codiceFiscale"/>
      </cf>
      <data>
        <xsl:value-of select="//documentContent[@id='modulo-
compilato']/dataDiNascita"/>
      </data>
      <luogo>
        <xsl:value-of select="//documentContent[@id='modulo-
compilato']/luogo"/>
      </luogo>
      <email>
        <xsl:value-of select="//documentContent[@id='modulo-
compilato']/email"/>

```



```

        </email>
    </dati-intestatario>
    <servizi-scelti>
        <installazione>
            <xsl:value-of select="//documentContent[@id='modulo-
compilato']/s1"/>
        </installazione>
        <configurazione>
            <xsl:value-of select="//documentContent[@id='modulo-
compilato']/s2"/>
        </configurazione>
        <congruita>
            <xsl:value-of select="//documentContent[@id='modulo-
compilato']/s3"/>
        </congruita>
        <esterni>
            <xsl:value-of select="//documentContent[@id='modulo-
compilato']/s4"/>
        </esterni>
        <curvatura>
            <xsl:value-of select="//documentContent[@id='modulo-
compilato']/s5"/>
        </curvatura>
    </servizi-scelti>
    <notifiche>
        <xsl:value-of select="//documentContent[@id='modulo-
compilato']/notifiche"/>
    </notifiche>
    <modalita>
        <xsl:attribute name="mode">
            <xsl:value-of select="//documentContent[@id='modulo-
compilato']/mode"/>
        </xsl:attribute>
        <transfascia>
            <xsl:value-of select="//documentContent[@id='modulo-
compilato']/transfascia"/>
        </transfascia>
        <fotonica>
            <xsl:value-of select="//documentContent[@id='modulo-
compilato']/fotonica"/>
        </fotonica>
        <phaser>
            <xsl:value-of select="//documentContent[@id='modulo-
compilato']/phaser"/>
        </phaser>
    </modalita>
    <anticipo>
        <xsl:value-of select="//documentContent[@id='modulo-
compilato']/anticipo"/>
    </anticipo>
    </contratto>
</xsl:template>
</xsl:transform>

```

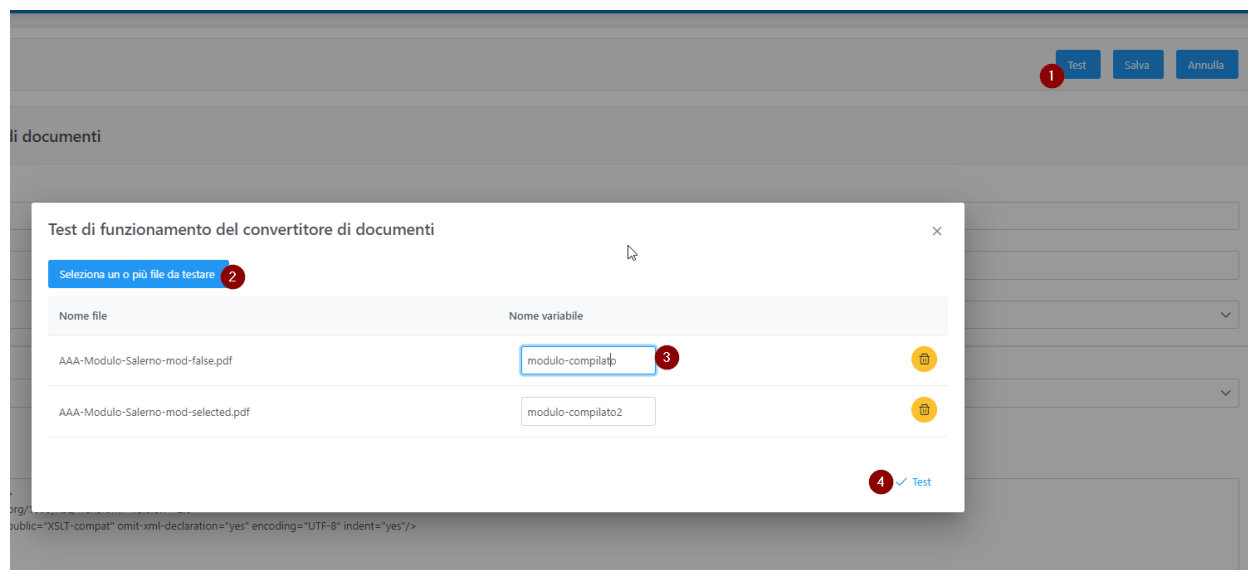
## 22.3 Esecuzione del test

Nell'esempio che segue produciamo un XML a partire da due diversi file PDF. Tali file PDF sono identificati con i seguenti valori di variabile nell'XML:

- `modulo-compilato` corrispondente al file [AAA-Modulo-Salerno-mod-false.pdf](#)
- `modulo-compilato2` corrispondente al file [AAA-Modulo-Salerno-mod-selected.pdf](#)

Si ricorda che le variabili referenziate nell'XSL devono corrispondere agli ID dei campi dei file impostati in FORM.IO.

Di seguito mostriamo come impostare il test:



output XML a partire dai file allegati

```
<contratto nome="" cognome="" data="">
  <form>
    <nome/>
    <cognome/>
    <data/>
  </form>
  <dati-intestatario>
    <nome>Marco</nome>
    <cognome>Riberico</cognome>
    <nome-referente>Marco</nome-referente>
    <cognome-referente>Riberico</cognome-referente>
    <cf>mrtkuj03h77h704f</cf>
    <data>11/09/1963</data>
    <luogo>Cava (SA)</luogo>
    <email>m.ribery@torre.gl</email>
  </dati-intestatario>
  <servizi-scelti>
    <installazione/>
    <configurazione/>
    <congruita/>
    <esterni/>
    <curvatura>Sì</curvatura>
  </servizi-scelti>
</contratto>
```

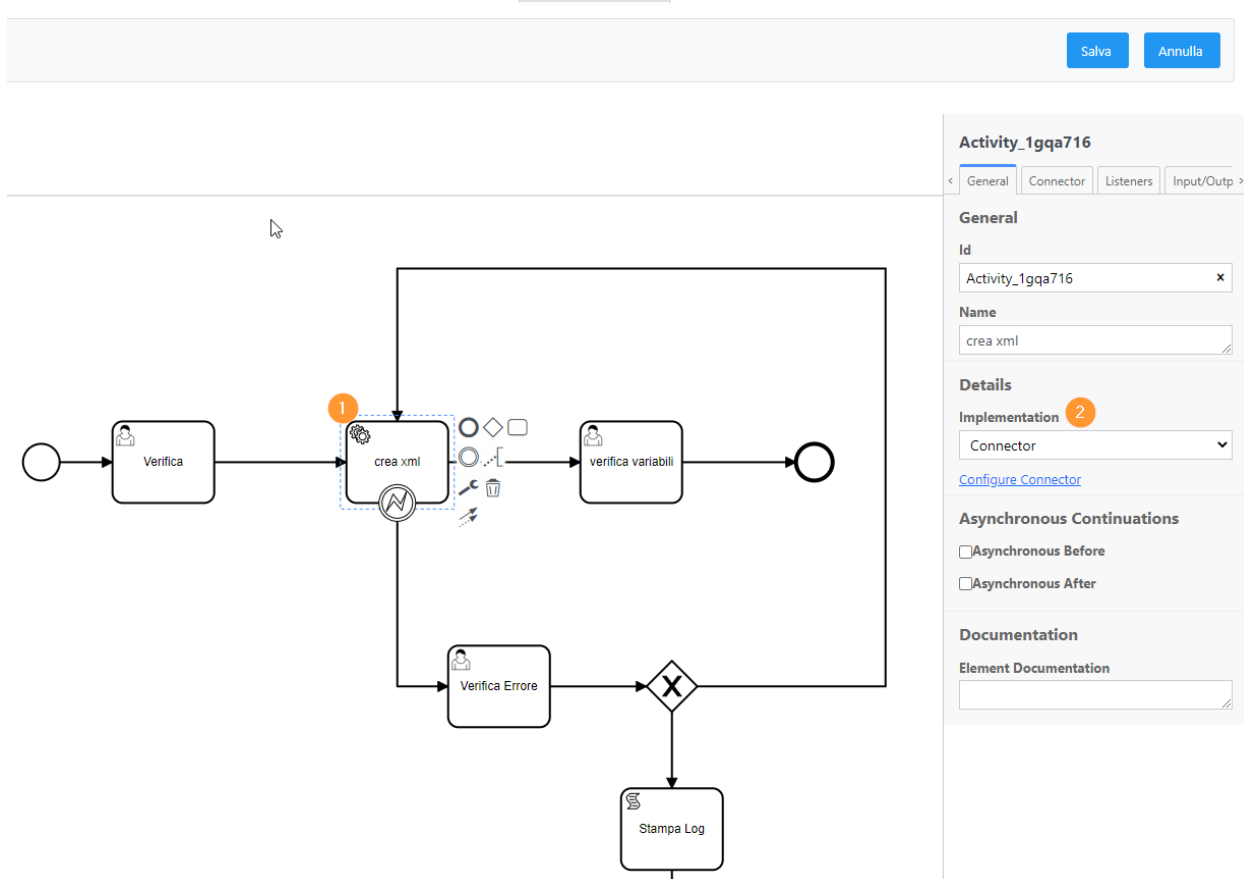
```
<notifiche>true</notifiche>
<modalita mode="">
  <trasfasica>base</trasfasica>
  <fotonica>singolo</fotonica>
  <phaser>sovraccarico</phaser>
</modalita>
<anticipo>700</anticipo>
</contratto>
```

## 22.4 Configurazione del convertitore nel flusso

Una volta effettuato il test, si può predisporre nel workflow la conversione del documento invocando una API REST del Gestore Pratiche. Di seguito mostriamo come utilizzare l'oggetto **Connector** per effettuare la chiamata e quindi la conversione del documento.

Nell'esempio che stiamo considerando, assumiamo di voler utilizzare due documenti di input PDF per produrre un XML.

- Configuriamo un service task
- Scegliamo come implementazione **Connector**



## 22.5 Configurazione Input Parameter

<http://10.0.254.7/gestionepratiche/api/bpm/document/convert>

**Activity\_1gqa716**

< General **Connector** Listeners Input/Outp >

**Details**

**Connector Id**

http-connector x

**Input Parameters** +

url x

**Local Variable Name**

url x

**Variable Assignment Type**

String or Expression v

**Variable Assignment Value**

http://10.0.254.7/gestionepratiche/api  
/bpm/document/convert

Start typing "\$fj" to create an expression.

> **method** ← POST

> **headers** ← Map

> **p.** ← { "externalProcedureInstanceId" ...

**Output Parameters** +

> **headers** ← \${headers}

> **statusCode** ← \${statusCode}

> **r.** ← import groovy.json.\* import org...

Impostare method POST ed utilizzare i seguenti header:

```
accept application/json
content-type application/json
origin http://camunda
gptenantid 1
```

**Activity\_1gqa716**

< General **Connector** Listeners Input/Outp >

**Details**

**Connector Id**  
http-connector x

**Input Parameters** +

> u... http://10.0.254.7/gestioneprati...

> method POST

▼ headers x

**Local Variable Name**  
headers x

**Variable Assignment Type**  
Map ▼

**Add Entry** +

Key	Value	
accept	application/json	x
content-type	application/json	x
origin	http://camunda	x
gptenantid	1	x

> p. { "externalProcedureInstanceId" ...

**Output Parameters** +

> headers {headers}

> statusCode {statusCode}

> r. import groovy.json.\* imp **BPMN.io**

Utilizziamo il seguente Payload:

```
{
  "externalProcedureInstanceId": "${execution.getProcessInstanceId()}",
  "data": {
    "documentConverterKey": "convertitore-v01",
    "documents": [
      {
        "name": "modulo-compilato",
        "mimeType": "application/pdf"
      },
      {
        "name": "modulo-compilato2",
        "mimeType": "application/pdf"
      }
    ]
  }
}
```

**Input Parameters** +

> **u..** ← http://10.0.254.7/gestionepr...

> **method** ← POST

> **headers** ← Map

▼ **payload** x

**Local Variable Name**

payload x

**Variable Assignment Type**

String or Expression ▼

**Variable Assignment Value**

```
{  
  
  "externalProcedureInstanceId": "${e  
xecution.getProcessInstanceId()}",  
  "data": {  
  
    "documentConverterKey": "convertit  
ore-v01",  
    "documents": [  
      {  
        "-----"-----d...l-  
      }  
    ]  
  }  
}
```

## 22.6 Configurazione Output Parameter

**Activity\_1gqa716**

< General **Connector** Listeners Input/Output >

**Details**

**Connector Id**

http-connector x

**Input Parameters** +

> **u..** ← http://10.0.254.7/gestioneprati...

> **method** ← POST

> **headers** ← Map

> **p.** ← { "externalProcedureInstanceId"...

**Output Parameters** +

▼ **headers** x

**Process Variable Name**

headers x

**Variable Assignment Type**

String or Expression ▼

**Variable Assignment Value**

`${headers}`

Start typing "\$()" to create an expression.

> **statusCode** ← `$(statusCode)`

> **r.** ← `import groovy.json.* import org...`

Activity\_1gqa716

< General Connector Listeners Input/Outp >

**Details**

Connector Id  
http-connector x

**Input Parameters** +

> u. http://10.0.254.7/gestioneprati...  
> method POST  
> headers Map  
> p. { "externalProcedureInstanceId"...

**Output Parameters** +

> headers \${headers} x  
▼ statusCode x  
Process Variable Name  
statusCode x  
Variable Assignment Type  
String or Expression  
Variable Assignment Value  
\${statusCode}  
Start typing "\${}" to create an expression.  
> r. import groovy.json.\* import org...

Script per la gestione dell'output:

```
import groovy.json.*
import org.camunda.bpm.engine.delegate.BpmnError

def httpResponse = connector.getVariable("response");
def httpStatusCode = connector.getVariable("statusCode");
def httpHeaders = connector.getVariable("headers");

JsonSlurper sl = new JsonSlurper();
Map parsedJson = sl.parseText(response);

def status = parsedJson.get("status");
def item = parsedJson.get("item");

println "httpResponse: " + httpResponse;
println "httpStatusCode: " + httpStatusCode;
println "httpHeaders: " + httpHeaders;
println "status: " + status;
println "item: " + item;
println "-----";

// if 500 throw an error
if ((httpStatusCode / 100) == 5) {
    println "HTTP Error: " + httpStatusCode;
    throw new BpmnError("connector-5xx-error");
}
```

Connector Id  
http-connector x

Input Parameters +

> u.. http://10.0.254.7/gestionepr...

> method POST

> headers Map

> p. { "externalProcedureInstance..."

Output Parameters +

> headers \${headers}

> statusCode \${statusCode}

response x

Process Variable Name  
response x

Variable Assignment Type  
Script v

Script Format  
groovy x

Script Type  
Inline Script v

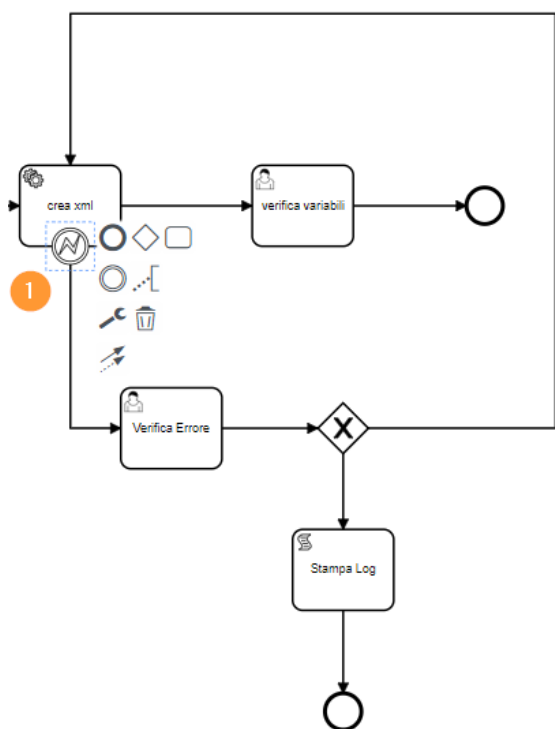
Script

```
import groovy.json.*
import
org.camunda.bpm.engine.delega
te.BpmnError

def httpResponse =
connector.getVariable("res
");
```

Gestione dell'eccezione connector-5xx-error lanciata dallo script:





### Event\_1hl5p6y

General Listeners Extensions

#### General

Id

Event\_1hl5p6y x

Name

#### Details

Error

connector-5xx-error (id=Error\_0b7l) v

Error Name

connector-5xx-error x

Error Code

connector-5xx-error x

Error Message

Error Code Variable

Error Message Variable

#### Asynchronous Continuations

☐ Asynchronous Before

☐ Asynchronous After

Documentation

[BPMN.io](https://bpmn.io)

## 23 CHIUSURA DI UNA PRATICA TRAMITE CHIAMATA REST

Quando si disegna un flusso con il Gestore Pratiche, l'istanza della pratica viene impostata su uno stato tecnico di "OPEN" ovvero aperta. Quando l'istanza del flusso termina, è necessario impostare lo stato a "CLOSED" invocando un apposito servizio del backend del gestore pratiche.

Bisogna quindi creare uno script che invoca tale servizio e imposta lo stato di chiusura ogni volta che nel BPMN si ha una situazione di terminazione dell'istanza.

Di seguito un esempio di script che realizza la chiamata:

```
import groovy.json.*;
import org.camunda.bpm.engine.impl.persistence.entity.ExecutionEntity;
import org.camunda.bpm.engine.impl.persistence.entity.TaskEntity;

def endpointURL = execution.getVariable("gpHost")
def httpConnector = org.camunda.connect.Connectors.http();

def payload = JsonOutput.toJson([ "externalProcedureInstanceId":
execution.getProcessInstanceId(),
    "status": "CLOSED"
]);

println("Payload          : " + payload);
println("Endpoint        : " + endpointURL);
def endpointCloseInstance = endpointURL + "/bpm/status";
println("EndpointClose     : " + endpointCloseInstance);

try {
    def resp = httpConnector.createRequest()
        .put()
        .header("gptenantid", "1")
        .header("Accept", "application/json")
        .header("origin", "http://camunda")
        .url(endpointCloseInstance)
        .contentType("application/json")
        .payload(payload)
        .execute();
    def result = resp.getResponse();
    println("Response Status Code : " + resp.getStatusCode());
    println("Response Headers     : " + resp.getHeaders());
    println("Response Body        : " + resp.getResponse());
    resp.close();
} catch (Exception ex) {
    println("ERROR: La chiusura della procedura ha restituito un errore. Il BPMN
continua, ma l'errore è loggato di seguito: " + ex.getMessage());
}
```

## 24 VALIDAZIONE SOSTANZIALE TRAMITE SERVIZIO ESTERNO

Dato un modulo (form) del GP, si vuole effettuare una validazione di uno o più dati tramite invocazione di un servizio esterno.

La soluzione per implementare tale meccanismo è riassunta nei passi che seguono:

1. Impostare i campi di interesse del modulo come "variabili di Processo", cosicché siano disponibili nell'istanza del workflow.
2. Creare uno script che recupera le variabili e invoca un servizio esterno per la validazione.
3. Pilotare il flusso utilizzando una variabile di output.

### Step 1. Configurare le variabili di istanza

Campo di testo Componente

Schermata **Dati** Validazione API Condizioni Logica Impaginazione

Etichetta ⓘ \*

Il/la sottoscritto/a

☐ Campo Ricercabile ⓘ

☒ Variabile di Processo ⓘ

Nome campo ⓘ \*

nomeCognome

Posizione dell'etichetta ⓘ

Alto

Testo segnaposto ⓘ

nome e cognome

### Step 2. Definire lo script per l'invocazione del servizio e l'impostazione della variabile per pilotare il flusso

```
import groovy.json.*;
import org.camunda.bpm.engine.impl.persistence.entity.ExecutionEntity;
import org.camunda.bpm.engine.impl.persistence.entity.TaskEntity;

def substantialValidationURL = "https://mgdlab.it/fake/ws-validation-mock.php"
def httpConnector = org.camunda.connect.Connectors.http();

def payload = JsonOutput.toJson([ "nomeCognome": execution.getVariable("nomeCognome"),
                                   "comuneResidenza": execution.getVariable("comuneResidenza"),
```

```
        "viaResidenza": execution.getVariable("viaResidenza"),
        "numeroResidenza": execution.getVariable("numeroResidenza"),
        "provincia": execution.getVariable("provinciaResidenza"),
        "codiceFiscale": execution.getVariable("codiceFiscale"),
        "comuneDiNascita": execution.getVariable("comuneDiNascita"),
        "dataDiNascita": execution.getVariable("dataDiNascita"),
        "telefono": execution.getVariable("telefono"),
        "email": execution.getVariable("email")
    ]);

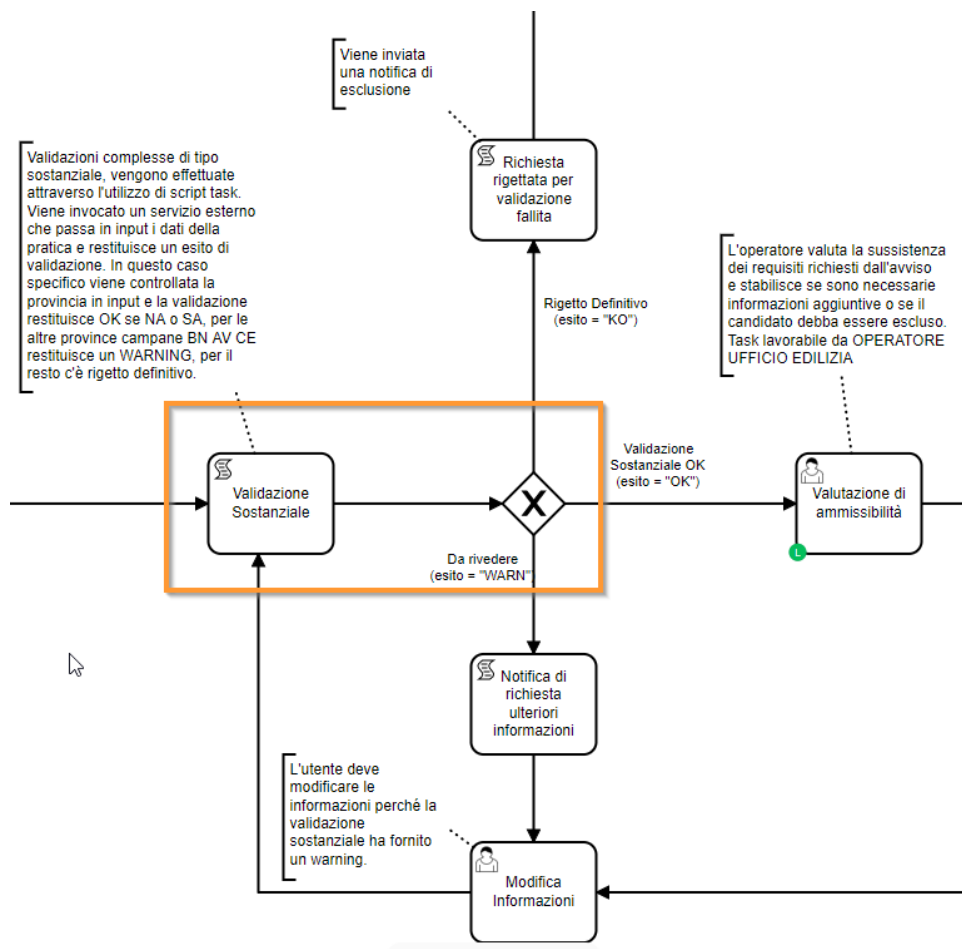
println("Payload          : " + payload);

try {
    def resp = httpConnector.createRequest()
        .post()
        .url(substantialValidationURL)
        .contentType("application/json")
        .payload(payload)
        .execute();

    def result = resp.getResponse();
    println("Response Status Code :" + resp.getStatusCode());
    println("Response Headers      :" + resp.getHeaders());
    println("Response Body           :" + resp.getResponse());

    def jsonSlurper = new JsonSlurper();
    def object = jsonSlurper.parseText(resp.getResponse());
    def valid = object.valid;
    // impostazione della variabile per pilotare il flusso, in base all'esito del
servizio
    def validationResult = "";
    if (valid == "OK") {
        validationResult = "OK";
    } else if (valid == "KO") {
        validationResult = "KO";
    } else if (valid == "WARN") {
        validationResult = "INTEGRAZIONE";
        // viene impostata una descrizione che compare sulla maschera dell'utente
finale.
        execution.setVariable("noteDiSistema", "Errore di validazione sostanziale sul
campo Provincia, si prega di selezionare SA o NA.");
    }
    execution.setVariable("esito", validationResult);
    resp.close();
} catch (Exception ex) {
    println("ERROR: La validazione sostanziale ha restituito un errore. Il BPMN
continua, ma l'errore è loggato di seguito: " + ex.getMessage());
    execution.setVariable("esito", "KO");
    execution.setVariable("noteDiSistema", "Il controllo di validazione ha restituito un
errore: " + ex.getMessage());
}
```

Step 3. Pilotare il flusso utilizzando la variabile impostata



In questo caso i 3 rami sono impostati utilizzando un Condition Type = Expression con i seguenti valori:

```

${esito == "KO"}
${esito == "OK"}
${esito == "INTEGRAZIONE"}
  
```

## 25 CREAZIONE DI VARIABILI DI PROCESSO

Gestione Pratiche permette di sincronizzare una o più variabili di processo all'interno di un workflow.

API:

POST {{baseUrl}}/api/bpm/procedure/variable

Payload:

```
{
  "externalProcedureInstanceId": "f2e1b974-7aa3-11ec-95e8-0242ac140006",
  "procedureVariableSetModel": {
    "procedureVariableVisibilityType": "CITIZEN_VISIBILITY",
    "variableList": [
      {
        "name": "nome",
        "type": "string",
        "value": "paolo"
      },
      {
        "name": "cognome",
        "type": "string",
        "value": "rossi"
      },
      {
        "name": "indirizzo",
        "type": "string",
        "value": "via anfossi 3"
      }
    ]
  }
}
```

- **externalProcedureInstanceId**: rappresenta l'identificativo dell'istanza di processo del workflow engine.

- **procedureVariableVisibilityType**: rappresenta la visibilità delle variabili "CITIZEN\_VISIBILITY" o "OPERATOR\_VISIBILITY".

- **variableList**: le variabili di processo definite nel workflow engine

R.T.I. Almaviva S.p.A/ Almawave S.p.a./ Indra Italia S.p.A/ Intellera Consulting S.r.l.	Sistema Pubblico di Connettività LOTTO 4
Documentazione Tecnica	SPCL4_COMUNESALERNO_PICS_ManualeUtente_GestorePratiche

I tipi supportati sono:

[boolean, bytes, date, double, file, integer, json, long, null, short, storedFileReference, string] (for POJO property 'variableList')

## 26 SINCRONIZZAZIONE DI VARIABILI DI PROCESSO

Gestione Pratiche permette di sincronizzare le variabili di processo all'interno di un workflow secondo una strategia.

API:

```
POST {{baseUrl}}/api/bpm/procedure/variable/sync
```

Payload:

```
{
  "externalProcedureInstanceId": "ad6c9e1b-7aa2-11ec-95e8-0242ac140006",
  "procedureVariableSyncModel": {
    "procedureVariableVisibilityType": null,
    "procedureVariableSyncStrategyType": "SYNC_ONLY_EXISTING"
  }
}
```

- **externalProcedureInstanceId**: rappresenta l'identificativo dell'istanza di processo del workflow engine.

- **procedureVariableVisibilityType**: rappresenta la visibilità delle variabili "CITIZEN\_VISIBILITY" o "OPERATOR\_VISIBILITY".

- **procedureVariableSyncStrategyType**: rappresenta la strategia di sincronizzazione "SYNC\_ONLY\_EXISTING", sincronizza solo le variabili già presenti in GestionePratiche, mentre "SYNC\_ALL" sincronizza tutte le variabili dal workflow engine verso GestionePratiche (**in questo caso è richiesta la visibilità per quelle variabili di processo che devono essere create in GestionePratiche**).

**N.B:** non è possibile sincronizzare le variabili di processo di tipo file.



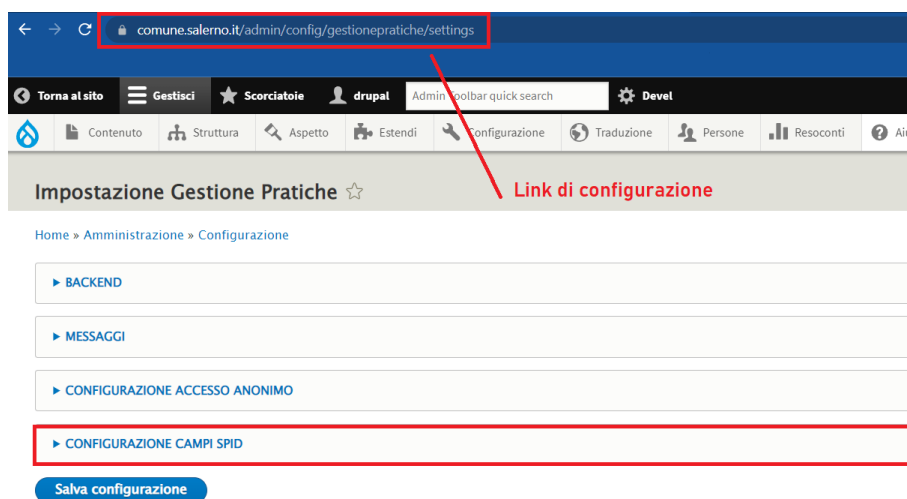
## 27 SETTING DEI CAMPI SPID E RELATIVA VISUALIZZAZIONE

Per poter eseguire il setting dei campi SPID occorre effettuare l'accesso come Amministratore (username: drupal).

Innanzitutto, bisogna configurare il nome dei campi SPID che si vuole pre-popolare, la configurazione è possibile accedendo alla pagina seguente:

<https://www.comune.salerno.it/admin/config/gestionepratiche/settings>.

Spostarsi, quindi, sulla scheda "CONFIGURAZIONE CAMPI SPID" e fare click per espandere.



Una volta aperta la scheda configurare il nome dei campi SPID da pre-polare.

A screenshot of the 'CONFIGURAZIONE CAMPI SPID' form. The form is titled '▼ CONFIGURAZIONE CAMPI SPID' and contains three input fields: 'Campo Nome SPID \*' with the value 'nome\_spid', 'Campo Cognome SPID \*' with the value 'cognome\_spid', and 'Campo Codice Fiscale SPID \*' with the value 'codiceFiscale\_spid'. A red box highlights these three fields, and a red arrow points from the text 'Setting campi spid' to the box. At the bottom of the form is a blue button labeled 'Salva configurazione'.

e infine salvare la configurazione.

▼ CONFIGURAZIONE CAMPI SPID

**Campo Nome SPID \***

**Campo Cognome SPID \***

**Campo Codice Fiscale SPID \***

**Salva configurazione**

Salvataggio configurazione

In seguito, tutte le pratiche con nome dei campi SPID che corrispondono ai nomi dei campi SPID impostati nella pagina di configurazione saranno pre-popolati.

Comune di Salerno  
Hippocratica Civitas

Amministrazione Aree Tematiche Documenti e dati Luoghi

Home / Tutte Le Pratiche / Tutte Le Pratiche / Nuova Pratica

Test spid

Nome

Codice Fiscale

Corrispondenza campi SPID


▼ CONFIGURAZIONE CAMPI SPID

**Campo Nome SPID \***

```

<div class="container mt-5 my-4" id="myform" style="display: block;">
  <div class="row"></div>
  <div class="col-12" id="formio">
    <div ref="component" class="formio-component formio-component-form formio-component-label-hidden" id="egh7pc">
      <div novalidate ref="webform" class="formio-form">
        <div ref="component" class="row formio-component formio-component-columns formio-component-columns formio-component-label-hidden" id="elvm1">
          <div ref="column-elvm1" class="col-md-6" col-md-offset-0 col-md-push-0 col-md-pull-0">
            <div ref="component" class="form-group has-feedback formio-component formio-component-textfield formio-component-no-e_spid" id="e0y2hb">
              <label for="e0y2hb-nome_spid" class="col-form-label active" style="width: auto;"> Nome </label>
              <div ref="element">
                <input id="e0y2hb-nome_spid" value="test.cittadino" spellcheck="true" lang="it" class="form-control" type="text" name="data[nome_spid]" ref="input"> -- $0
              </div>
              <div class="formio-errors invalid-feedback" ref="messageContainer"></div>
            </div>
          <div ref="column-elvm1" class="col-md-6" col-md-offset-0 col-md-push-0 col-md-pull-0">
            <div class="formio-errors invalid-feedback" ref="messageContainer"></div>
          </div>
        </div>
      </div>
    </div>
  </div>
</pre>

```

**Comune di Salerno**  
Hippocratica Civitas

[Amministrazione](#) [Aree Tematiche](#) [Documenti e dati](#) [Luoghi](#) [Novità](#) [Turismo](#) [Tutti gli argomenti](#)

[Home](#) / [Tutte Le Pratiche](#) / [Tutte Le Pratiche](#) / [Nuova Pratica](#)

Test spid

Nome

test.cittadino

Cognome

Villaggio

Codice Fiscale

test.cittadino

Chiudi

Salva Bozza

Invia

## AMMINISTRAZIONE TRASPARENTE

Disposizioni generali  
Organizzazione  
Personale  
Bandi di concorso

Sovvenzioni, contributi, sussidi, vantaggi economici  
Bilanci  
Beni immobile e gestione patrimonio  
Controlli e rilievi sull'amministrazione

N.B: I campi SPID configurabili sono soltanto tre: nome, cognome, codice fiscale. I nomi dei campi SPID delle pratiche devono corrispondere a quelli impostati nella pagina di configurazione.

## 28 GESTIONE DEL SERVIZIO DI NOTIFICA

Il servizio è esposto sull'endpoint <https://www.comune.salerno.it/gestionepratiche/notification> e viene abilitato tramite una chiamata HTTP con metodo POST, in cui va definito il seguente body:

```
{
  "user" : " ",
  "mess" : " ",
  "idPratica" : " ",
  "codiceFiscale" : " "
}
```

di seguito si riportano le descrizioni dei singoli parametri<sup>2</sup>:

user: rappresenta l'user dell'utente che effettua la pratica.

mess: rappresenta il messaggio che si vuole visualizzare

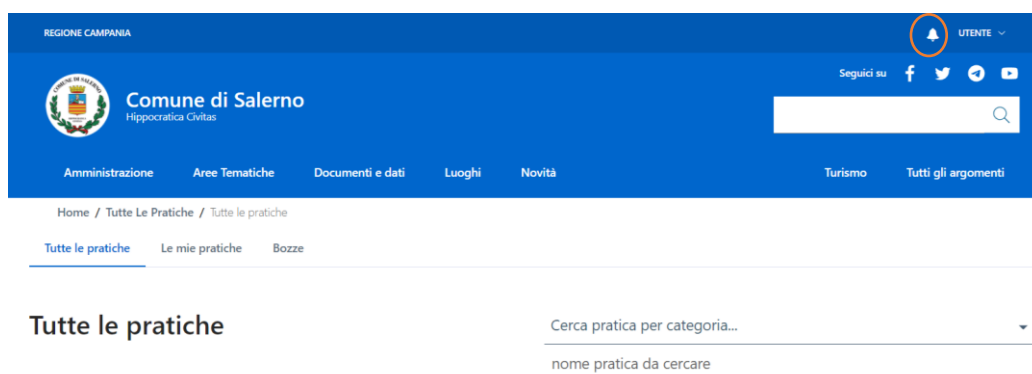
idPratica: rappresenta l'id della pratica istanziata

codiceFiscale: rappresenta il codice fiscale dell'utente che ha istanziato la pratica

La risposta al servizio fornirà un output del tipo:

```
{
  "status": true | false,
  "message": "user not found"
}
```

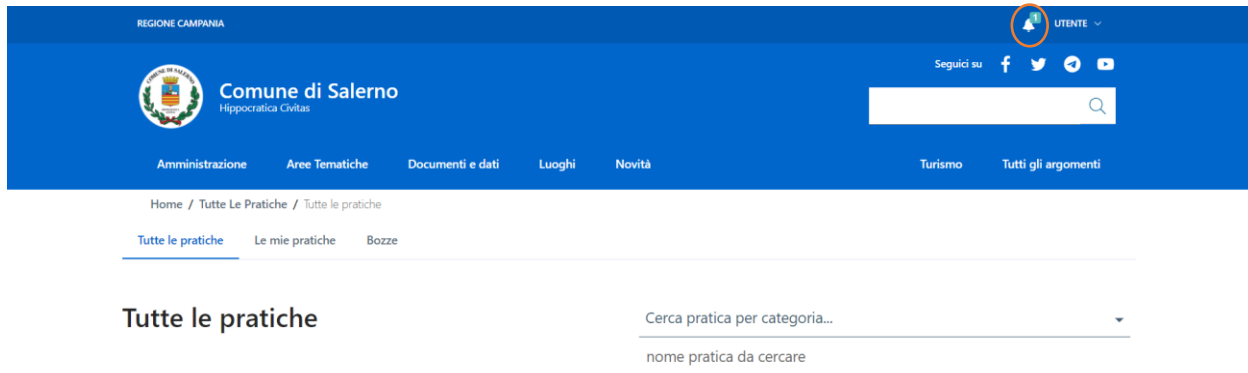
A livello visivo, la funzione di notifica è rappresentata tramite l'icona a campanella, presente in alto a destra dell'header del portale, come mostrato di seguito.



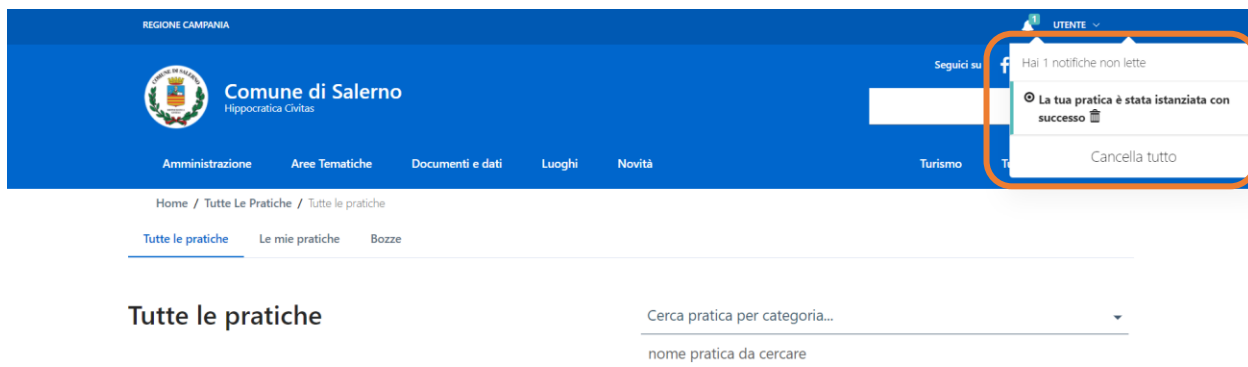
La presenza di eventuali notifiche è segnalata tramite un indicatore numerico posto in apice sul simbolo mostrato precedentemente.

<sup>2</sup> Questi parametri sono **obbligatori**. Se non viene rispettato questo vincolo, il portale mostrerà una schermata di errore.

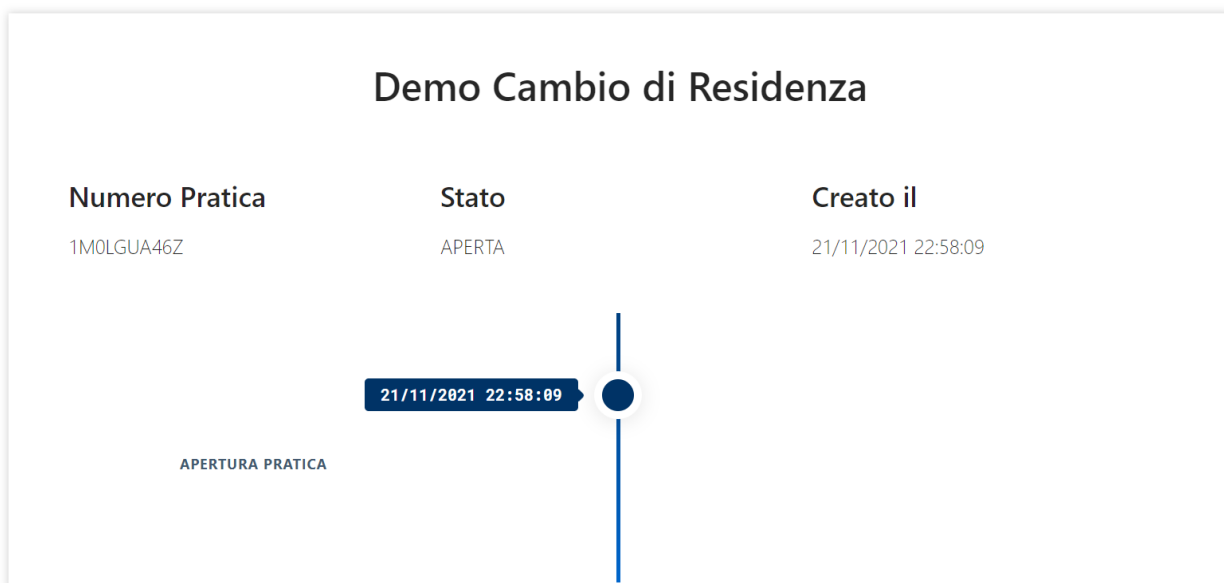
<sup>3</sup> Questo parametro viene valorizzato solo se l'esito del servizio ha dato ko, ovvero status: false.



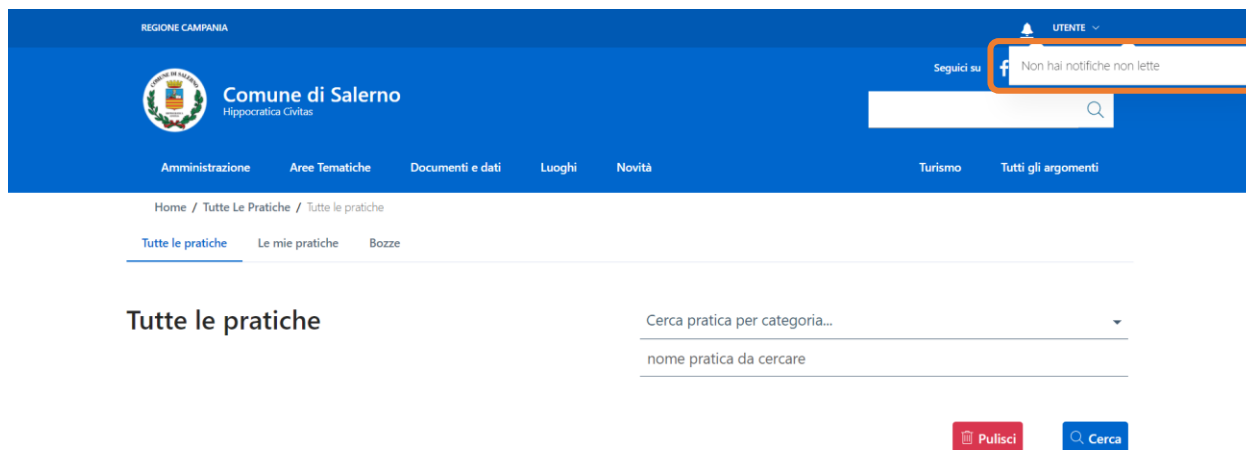
Cliccando sulla notifica, apparirà un pop-up in cui verrà visualizzato l'elenco delle notifiche con relativo messaggio.



Cliccando, ancora, sulla singola notifica, verrà mostrato il dettaglio e lo stato della pratica, come mostrato successivamente.



Infine, se non sono presenti notifiche, si avrà la situazione descritta dalla successiva immagine.



## 29 INVIO DEL SEGNALE

Gestione Pratiche permette di inviare un segnale verso il workflow engine.

API:

POST `{{baseUrl}}/api/bpmexternal/throwsignal`

Payload:

```
{
  "name": "SBLOCCA-CONCORSO-XYZ",
  "procedureId": 130,
  "procedureInstanceIds": [4265],
  "variables": [
    {
      "name": "nome",
      "type": "string",
      "value": "utente"
    }
  ]
}
```

Il segnale può essere inviato a:

- 1) tutte le istanze della procedura, in questo caso bisogna popolare il campo **procedureId**, e non passare il campo **procedureInstanceId**. Il segnale verrà propagato per tutte le istanze che hanno un workflow e che non si trovano nello stato **CLOSED**.
- 2) una singola istanza della procedura, in questo caso bisogna popolare il campo **procedureInstanceId**, e non passare il campo **procedureId**.
- 3) broadcast, quindi a tutte le procedure avviate.

-**name**: rappresenta il nome univoco del segnale da inviare, è un campo obbligatorio.

- **variables**: lista di variabili legate all'istanza di processo. Di seguito i tipi di variabili supportati.

```
@JsonSubTypes.Type(value = BooleanProcessVariable.class, name = "boolean"),
@JsonSubTypes.Type(value = DoubleProcessVariable.class, name = "double"),
@JsonSubTypes.Type(value = IntegerProcessVariable.class, name = "integer"),
@JsonSubTypes.Type(value = LongProcessVariable.class, name = "long"),
@JsonSubTypes.Type(value = NullProcessVariable.class, name = "null"),
@JsonSubTypes.Type(value = ShortProcessVariable.class, name = "short"),
@JsonSubTypes.Type(value = StringProcessVariable.class, name = "string"),
@JsonSubTypes.Type(value = DateProcessVariable.class, name = "date"),
@JsonSubTypes.Type(value = BytesProcessVariable.class, name = "bytes"),
@JsonSubTypes.Type(value = JsonObjectProcessVariable.class, name = "json")
```

## 29.1 Caso d'uso



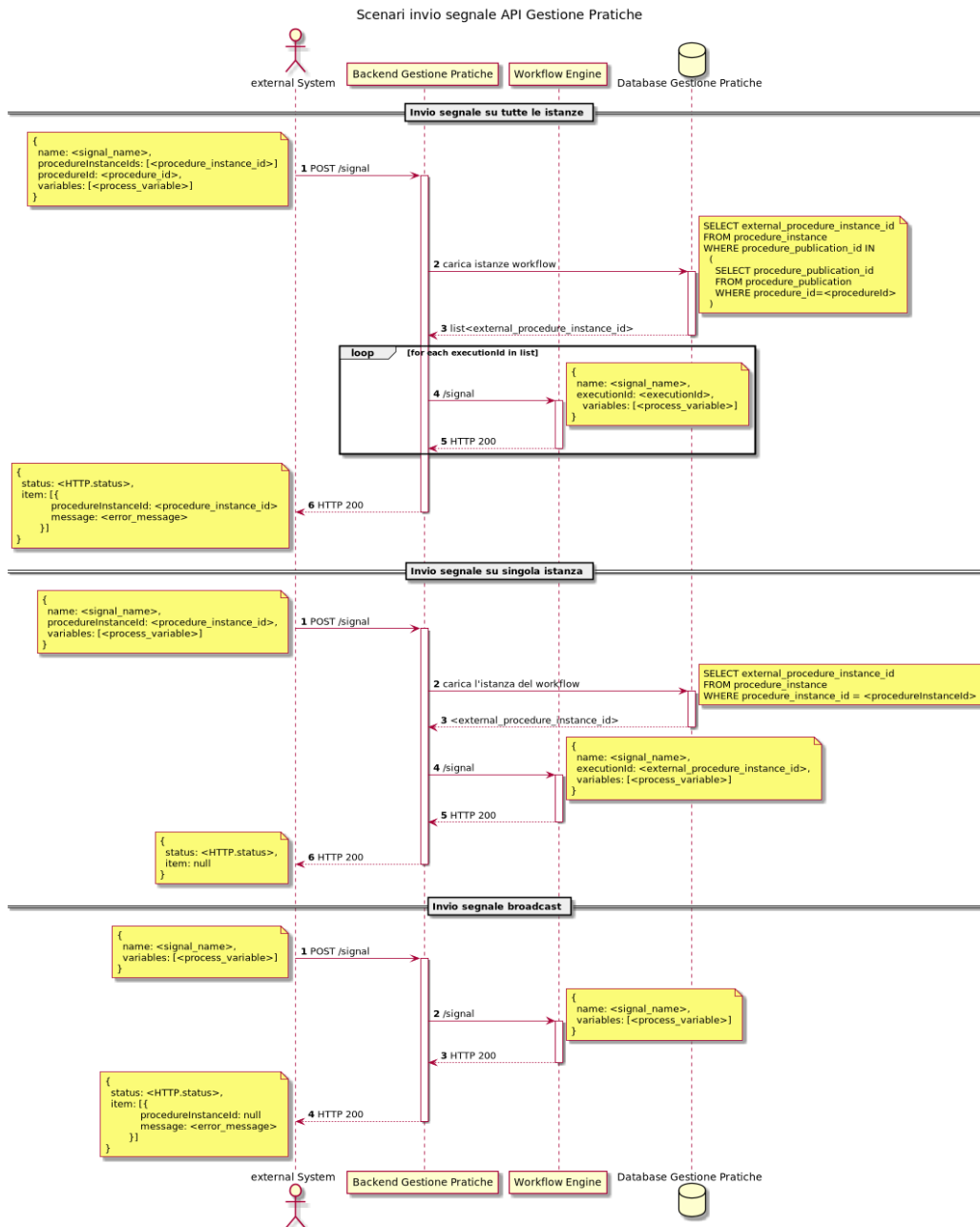
In questo caso d'uso abbiamo due task assegnati rispettivamente ad un utente e al cittadino. Prima di passare al task del cittadino, il workflow sospende il flusso e rimane in attesa del segnale con identificativo **signalTest**.

Per sbloccare il flusso è quindi necessario inviare un segnale per questa istanza di procedura.

```
{
  "name": "signalTest",
  "procedureInstanceIds": [4265],
  "variables": []
}
```



## 29.2 Sequence diagram



## 29.3 Gestione degli errori

Il servizio per l'invio di un segnale ritorna la lista di istanze di procedure il cui invio del segnale è fallito:

```

{
  "status": "ERROR",
  "item": [
    {
      "procedureInstanceId": 122,
      "message": "Execution not found"
    }
  ]
}

```

```
}  
]  
}
```

## 29.4 Set dello stato di una istanza di procedura

Gestione Pratiche permette di impostare lo stato di una istanza di procedura.

### API:

PUT {{baseUrl}}/api/bpmexternal/status

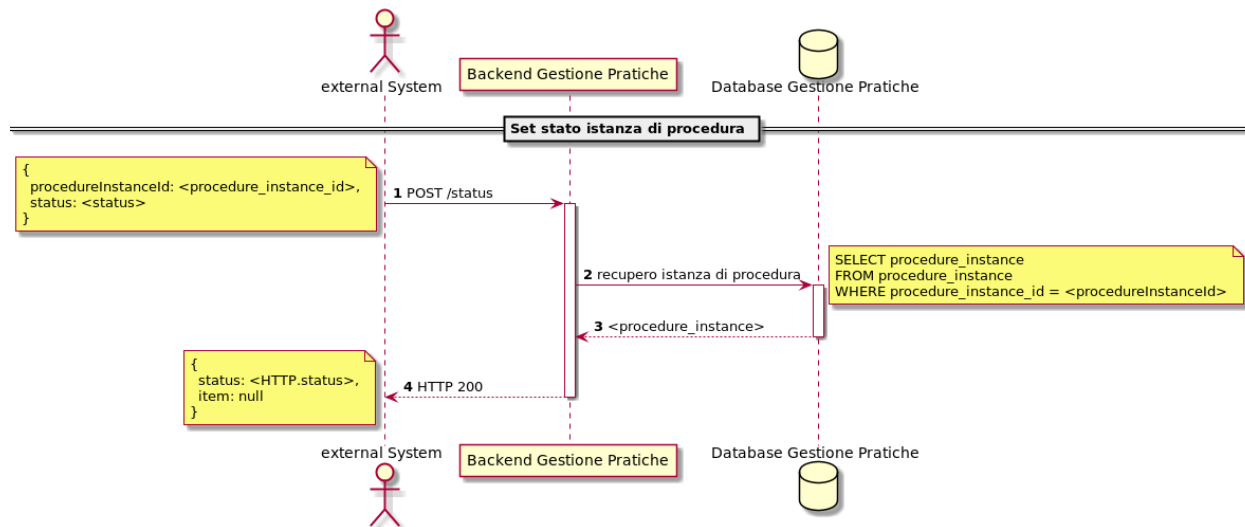
### Payload:

```
{  
  "procedureInstanceId": 202,  
  "status": "CLOSED"  
}
```

- 1) **procedureInstanceId**: identificativo della istanza di procedura
- 2) **status**: identificativo (key) dello stato di procedura.

E' possibile impostare solamente gli stati che appartengono a quella procedura.

## Sequence diagram



## Gestione avanzata lookup

Gestione Pratiche permette di definire i valori di un lookup e anche i suoi tag (per ciascun valore).

Gestione Pratiche

amministratore  
Comune di Salerno

Salva

Annulla

Dettaglio lookup

Dati di base

Dettaglio lookup

Importa valori

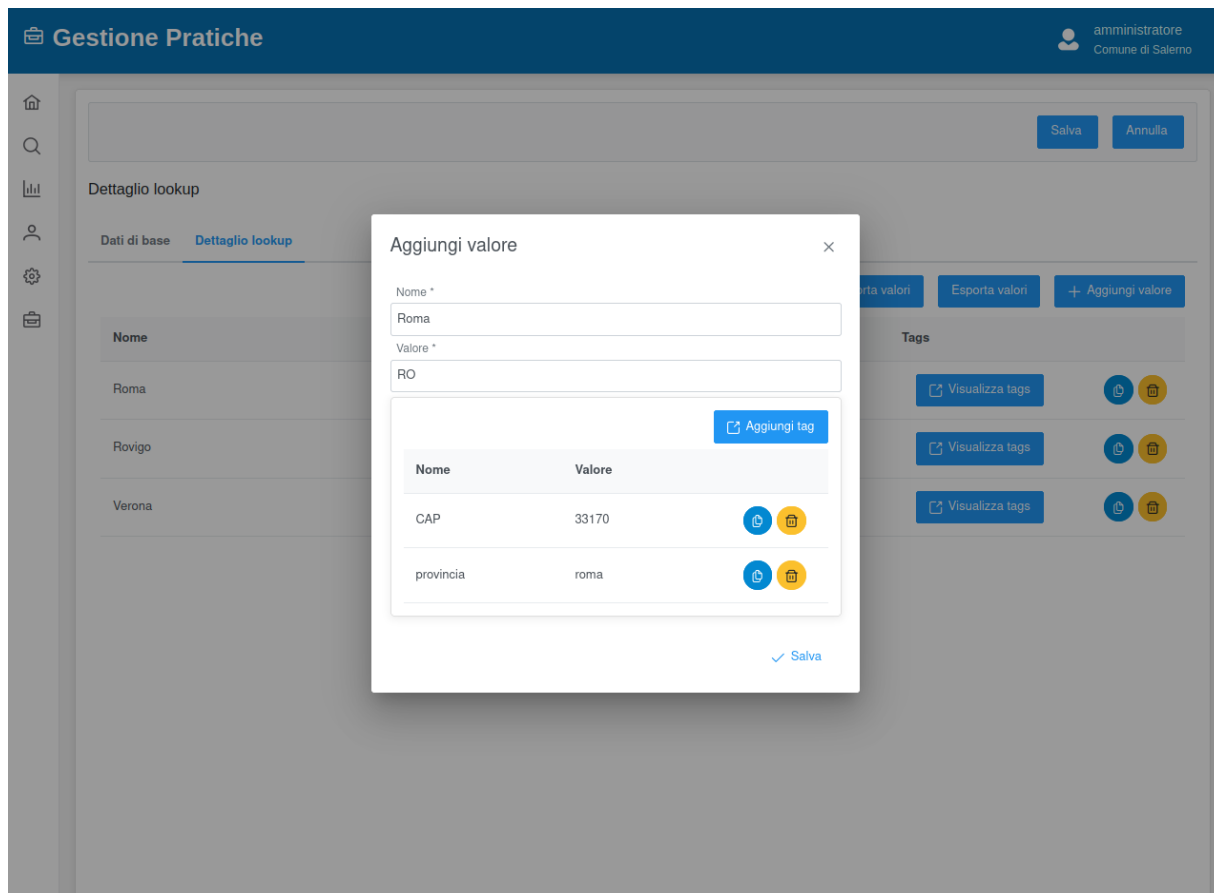
Esporta valori

+ Aggiungi valore

Nome	Valore	Tags
Roma	RO	<div>Visualizza tags</div> <div></div>
Rovigo	RV	<div>Visualizza tags</div> <div></div>
Verona	VR	<div>Visualizza tags</div> <div></div>

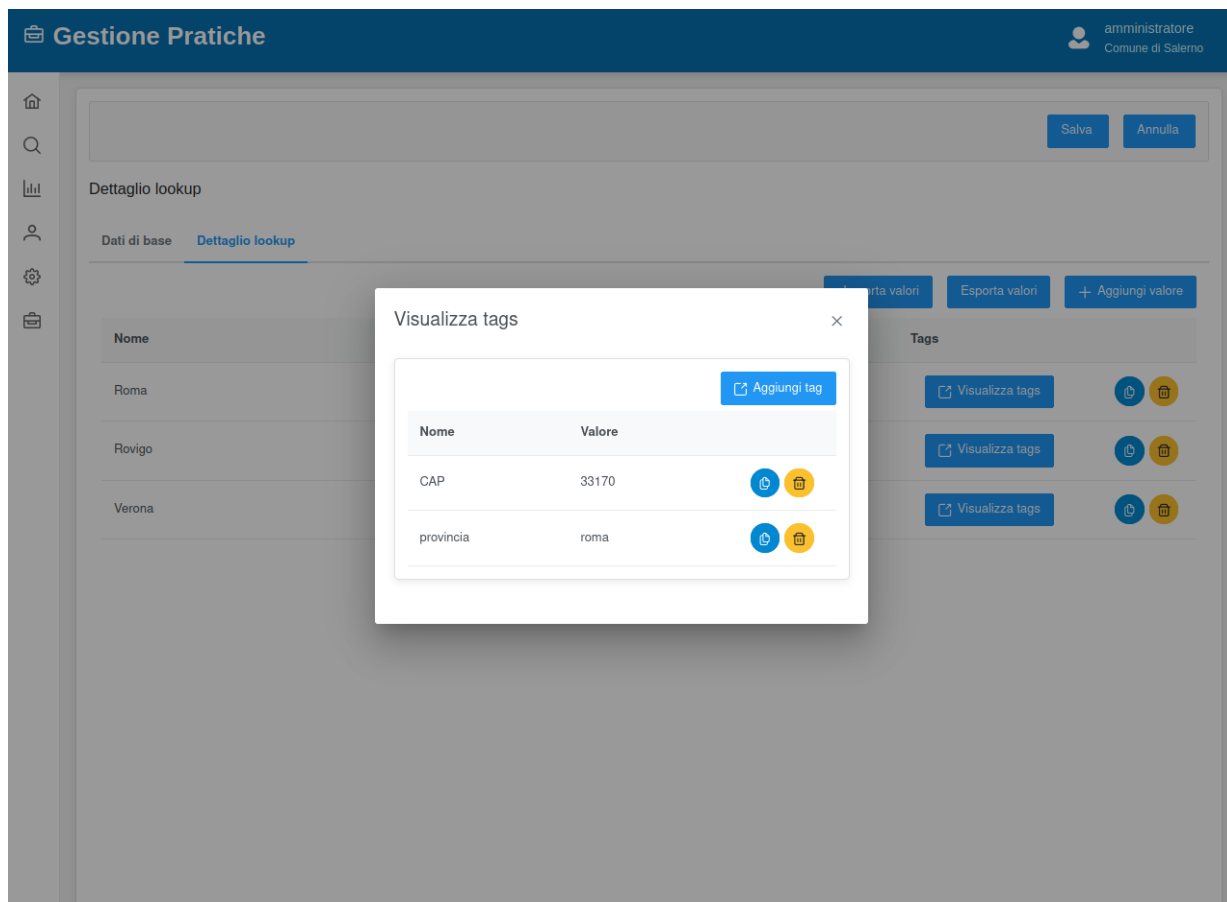
In particolare è possibile:

- 1) importare i valori da formato csv secondo la regola “name;value;tagName=tagValue|”.  
**N.B** il foglio csv deve avere le intestazioni “Name,Value,Tags”
- 2) esportare i valori in formato csv secondo le regole definite precedentemente.
- 3) aggiungere un valore e i suoi tag manualmente.



Per ogni  
valore è  
possibile  
modificarlo,  
ed

visualizzare/modificare i suoi tags.



**N.B** Il nome di ogni valore è univoco per ciascun lookup, mentre il nome del tag è univoco per ciascun valore.

## 30 GESTIONE AVANZATA LOOKUP

Gestione Pratiche fornisce due servizi per il recupero dei valori del lookup

### API:

GET {{baseUrl}}/api/lookupexternal

### Esempio:

{{baseUrl}}/api/lookupexternal?id=68

### Parameters:

- **id** (required) identificativo del lookup

Il servizio ritorna la lista dei valori compresi dei tags dato come parametro l'id del lookup.

### Response:

```
"status": "OK",
"item": [
  {
    "id": 278,
    "name": "Roma",
    "value": "R0",
    "lookupTagValueModels": [
      {
        "id": 217,
        "name": "CAP",
        "value": "33170"
      },
      {
        "id": 218,
        "name": "provincia",
        "value": "roma"
      }
    ]
  },
  {
    "id": 279,
    "name": "Rovigo",
    "value": "RV",
    "lookupTagValueModels": [
      {
        "id": 219,
        "name": "CAP",
        "value": "33120"
      },
      {
        "id": 220,
        "name": "provincia",
        "value": "rovigo"
      }
    ]
  }
]
```

```
{
  "id": 280,
  "name": "Verona",
  "value": "VR",
  "lookupTagValueModels": [
    {
      "id": 221,
      "name": "CAP",
      "value": "1220"
    },
    {
      "id": 222,
      "name": "provincia",
      "value": "verona"
    }
  ]
}
```

#### API:

GET `{{baseUrl}}/api/lookupexternal/search`

#### Esempio:

`{{baseUrl}}/api/lookupexternal/search?name=ProvincedelVeneto&comune=padova&codice=M1`



#### Parameters:

- **id** (optional) rappresenta l'id del lookup
- **name** (required) rappresenta il nome del lookup

Nome \*

Provincie del Veneto

- **value** (optional) rappresenta il value di un valore di lookup.

Nome	Valore	Tags
Roma	RO	<a href="#">Visualizza tags</a>  

**N.B per filtrare per 1 o più tag inserire i parametri (query param) come in esempio:**

`?name=ProvincedelVeneto&<tag_key>=<key>&<tag_value>=<value>&....`

**N.B almeno uno dei seguenti parametri è richiesto id, name**

Il servizio ritorna la lista dei valori di un lookup ed utilizza diversi filtri.



```
{
  "status": "OK",
  "item": [
    {
      "id": 278,
      "name": "Roma",
      "value": "R0",
      "lookupTagValueModels": [
        {
          "id": 217,
          "name": "CAP",
          "value": "33170"
        },
        {
          "id": 218,
          "name": "provincia",
          "value": "roma"
        }
      ]
    },
    {
      "id": 279,
      "name": "Rovigo",
      "value": "RV",
      "lookupTagValueModels": [
        {
          "id": 219,
          "name": "CAP",
          "value": "33120"
        },
        {
          "id": 220,
          "name": "provincia",
          "value": "rovigo"
        }
      ]
    },
    {
      "id": 280,
      "name": "Verona",
      "value": "VR",
      "lookupTagValueModels": [
        {
          "id": 221,
          "name": "CAP",
          "value": "1220"
        },
        {
          "id": 222,
          "name": "provincia",
          "value": "verona"
        }
      ]
    }
  ]
}
```

## 30.1 Database

La gestione avanzata dei lookup coinvolge le seguenti tabelle del database:

- 1) lookup
- 2) lookup\_value

R.T.I. Almaviva S.p.A/ Almawave S.p.a./ Indra Italia S.p.A/ Intellera Consulting S.r.l.	Sistema Pubblico di Connettività LOTTO 4
Documentazione Tecnica	SPCL4_COMUNESALERNO_PICS_ManualeUtente_GestorePratiche

### 3) lookup\_tag\_value

## 31 RICERCA AVANZATA

La ricerca avanzata si compone delle seguenti funzionalità che descriveremo in dettaglio nei paragrafi successivi:

- 1) Ricerca avanzata tramite query builder
- 2) Definizione del formato di esportazione
- 3) Ricerca avanzata tramite API
- 4) Definizione della query

### 31.1 Ricerca avanzata tramite query builder

Gestione Pratiche permette di costruire query dinamiche per cercare le istanze di procedura.

Una query è un insieme di filtri che vengono composti tramite il componente QueryBuilder.

I filtri sono le nostre variabili che abbiamo definito durante l'avanzamento della istanza di procedura.

La seguente pagina si suddivide in tre pannelli:

- 1) Il pannello laterale a sinistra (che descriveremo in seguito).
- 2) Il pannello laterale a destra che mostra i risultati della query.

- 3) il pannello in alto con i tre pulsanti rispettivamente di **esportazione risultati**, **caricamento di una query**, **salvataggio di una query**.

Il pannello laterale a sinistra permette di:

- 1) Selezionare una o più istanze di procedura. Questo ci permette di scegliere i filtri che vogliamo utilizzare per comporre la nostra query.
- 2) Definire la query utilizzando i filtri precedentemente selezionati (i filtri vengono precaricati automaticamente).
- 3) Definire uno o più ordinamenti utilizzando per ciascuno il filtro ed il tipo di ordinamento.

**N.B Anche le variabili di tipo json possono essere usate per comporre la query con alcuni limiti:**

- 1) Gestione Pratiche per ogni variabile di tipo json esegue un flat e recupera solamente il primo livello (nested) del json.
- 2) Per ogni variabile di tipo json che è un array viene preso solo il primo elemento.

## 31.2 Definizione del formato di esportazione

Gestione Pratiche permette di definire il formato di esportazione della query. Indica quindi quali saranno i campi che andranno a comporre le colonne del file csv esportato.

The screenshot shows the 'Gestione Pratiche' interface. At the top, there is a blue header bar with the title 'Gestione Pratiche' and a user profile icon labeled 'amministratore Comune di Salerno'. Below the header, there is a sidebar with icons for home, search, list, user, settings, and a menu. The main content area is titled 'Dettaglio formato di esportazione' and contains a form with the following fields:

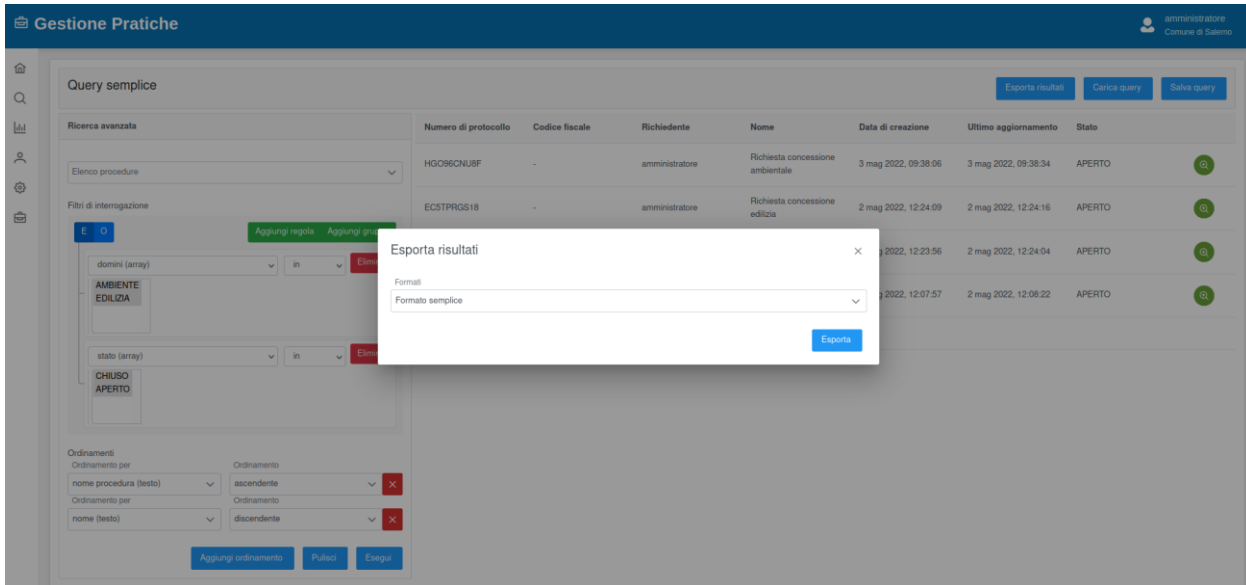
- Nome \***: A text input field with the value 'Formato semplice'.
- Descrizione \***: A text input field with the value 'Formato semplice'.
- Elenco procedure**: A dropdown menu with the selected value 'Richiesta concessione edilizia' and another option 'Richiesta concessione ambientale'.
- Valori di formato di esportazione \***: A dropdown menu with a list of variables: 'identificativo istanza di procedura', 'nome', 'numero di protocollo', 'nome del richiedente', 'identificativo del richiedente', 'cognome del richiedente', 'stato', 'data cambio stato in chiuso', 'data cambio stato in aperto', and 'data ultima modifica'.

At the top right of the form, there are two buttons: 'Salva' and 'Annulla'.

Il formato di esportazione si compone delle seguenti proprietà:

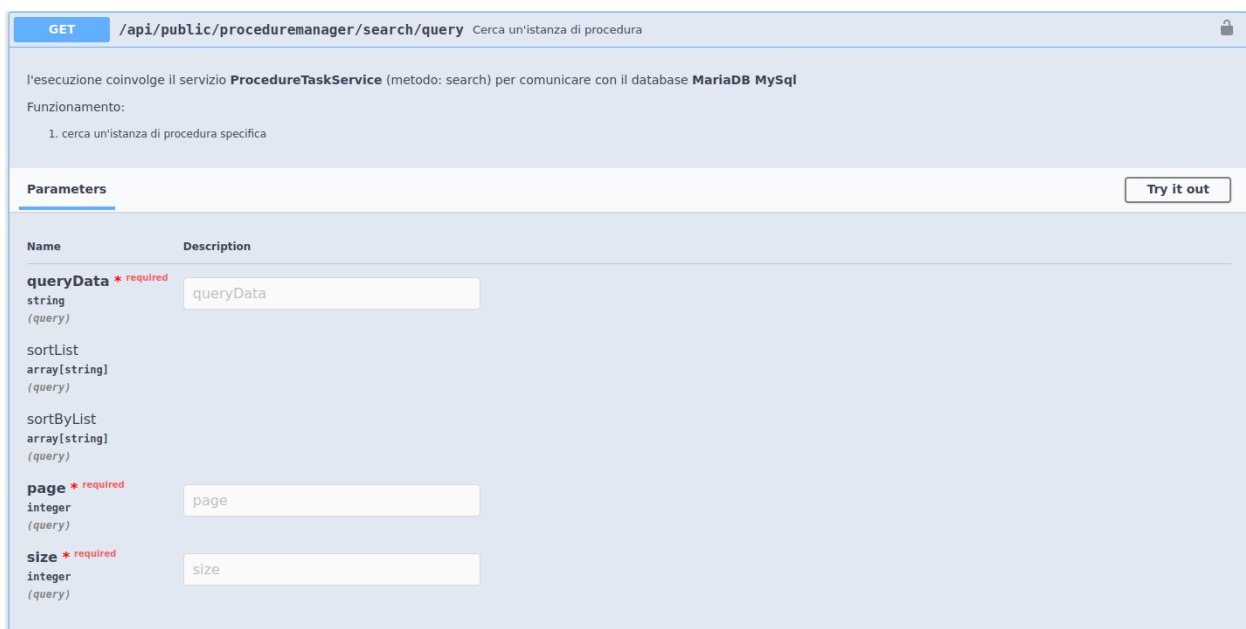
- 1) **nome** (campo obbligatorio) nome univoco del formato di esportazione.
- 2) **descrizione** (campo obbligatorio) descrizione del formato di esportazione.
- 3) **elenco procedure** (campo opzionale) permette di selezionare le procedure delle quali recuperare le variabili come filtri.
- 4) **valori formato di esportazione** (campo obbligatorio) i valori (filtri) che compariranno nel csv.

Il formato di esportazione viene utilizzato nella pagina dove si effettua la ricerca tramite query builder



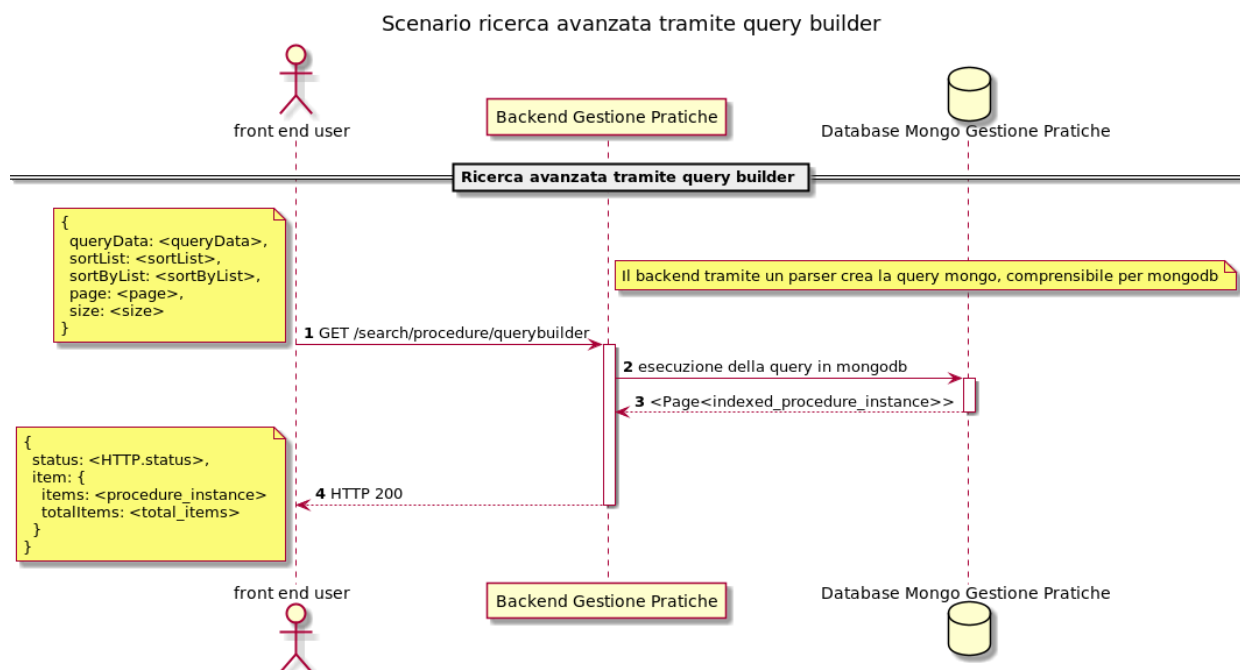
### 31.3 Ricerca avanzata tramite API

GestionePratiche espone una API pubblica ed i servizi (che è possibile chiamare solo da workflow) per effettuare la ricerca avanzata tramite query builder:



Il servizio pubblico richiede i seguenti parametri:

- 1) **queryData** (campo obbligatorio) rappresenta la query. (Successivamente descriveremo come comporre una query).
- 2) **sortList** (campo opzionale) rappresenta la lista di tipi di ordinamento (ASC, DESC).
- 3) **sortByList** (campo opzionale) rappresenta la lista di campi di ordinamento. **N.B** sortList e sortByList se utilizzati, vanno usati assieme.
- 4) **page** (campo obbligatorio) il servizio è paginato quindi richiede il campo page.
- 5) **size** (campo obbligatorio) il servizio è paginato quindi richiede il campo size.



Di seguito elencati i servizi che è possibile chiamare solo da workflow (creando un service task o uno script task):

Il servizio paginato ha gli stessi parametri del servizio precedente come body. (In questo caso è una POST, per gestire meglio il payload)

**POST** /api/bpm/procedureinstance/page/search ^

**Parameters** Try it out

No parameters

**Request body** required application/json v

**Example Value** | Schema

```
{
  "queryData": "string",
  "sortList": [
    "string"
  ],
  "sortByList": [
    "string"
  ],
  "page": 0,
  "size": 0
}
```

Il servizio paginato richiede il seguente body:

```
{
  "queryData":
  "{ \"condition\": \"AND\", \"rules\": [{ \"id\": \"domains\", \"field\": \"domains\", \"type\": \"string\", \"input\": \"select\", \"operator\": \"in\", \"value\": [ \"AMBIENTE\", \"EDILIZIA\" ] }, { \"condition\": \"AND\", \"rules\": [{ \"id\": \"status\", \"field\": \"status\", \"type\": \"string\", \"input\": \"select\", \"operator\": \"in\", \"value\": [ \"CLOSED\", \"OPEN\" ] } ] }, \"valid\": true\",
  \"page\": 0,
  \"size\": 10
}
```

Sostanzialmente sono gli stessi parametri dei servizi precedenti ma qui vengono passati nel body

Due parametri opzionali sono **sortList** e **sortByList**, descritti precedentemente.



**POST** /api/bpm/procedureinstance/search ^

**Parameters** Try it out

No parameters

**Request body** required application/json v

**Example Value** | Schema

```
{
  "queryData": "string"
}
```

Il servizio non paginato richiede il seguente body:

```
{
  "queryData":
  "{ \"condition\": \"AND\", \"rules\": [{ \"id\": \"domains\", \"field\": \"domains\", \"type\": \"string\", \"input\": \"select\", \"operator\": \"in\", \"value\": [ \"AMBIENTE\", \"EDILIZIA\" ] }, { \"condition\": \"AND\", \"rules\": [{ \"id\": \"status\", \"field\": \"status\", \"type\": \"string\", \"input\": \"select\", \"operator\": \"in\", \"value\": [ \"CLOSED\", \"OPEN\" ] } ] }, \"valid\": true\"
}
```

### 31.4 Definizione della query

Per la ricerca avanzata tramite API è richiesto come parametro la query. Partiamo da un esempio semplice di query:

```
{
  "condition": "AND",
  "rules": [
    {
      "id": "domains",
      "field": "domains",
      "type": "string",
      "input": "select",
      "operator": "in",
      "value": [
        "AMBIENTE",
        "EDILIZIA"
      ]
    },
    {
      "id": "status",
      "field": "status",
      "type": "string",
      "input": "select",
      "operator": "in",
```

```
    "value": [
      "CLOSED",
      "OPEN"
    ]
  },
],
"valid": true
}
```

la query è un oggetto json che contiene le seguenti proprietà:

1) **condition:** assume il valore **AND** oppure **OR**.

1) **rules:** è l'insieme delle regole.

2) **valid:** è un parametro che si può omettere.

Ogni rule è composta dalle seguenti proprietà:

1) **id:** è l'identificativo della rule

2) **field:** è il nome del campo della procedura, se è un campo *base* assume il valore del campo, se è un campo *variabile* ha il seguente formato **VARIABLE | nomeVariabile**.

3) **type:** assume uno dei seguenti valori (string, integer, double, date, time, datetime, boolean).

4) **input:** assume uno dei seguenti valori (text, number, textarea, radio, checkbox, select).

5) **operator:** assume uno dei seguenti valori (equal, not\_equal, in, not\_in, less, less\_or\_equal, greater, greater\_or\_equal, between, not\_between, begins\_with, not\_begins\_with, contains, not\_contains, ends\_with, not\_ends\_with, is\_empty, is\_not\_empty, is\_null, is\_not\_null).

6) **value:** rappresenta il valore che si vuole cercare.

La query può avere più rules innestate (possiamo pensarla come una query innestata), ad esempio:

```
{
  "condition": "AND",
  "rules": [
    {
      "id": "domains",
      "field": "domains",

```

```
    "type": "string",
    "input": "select",
    "operator": "in",
    "value": [
      "AMBIENTE",
      "EDILIZIA"
    ]
  },
  {
    "condition": "AND",
    "rules": [
      {
        "id": "status",
        "field": "status",
        "type": "string",
        "input": "select",
        "operator": "in",
        "value": [
          "CLOSED",
          "OPEN"
        ]
      }
    ]
  }
],
"valid": true
}
```

La query può avere delle rules sulle variabili, di seguito un esempio:

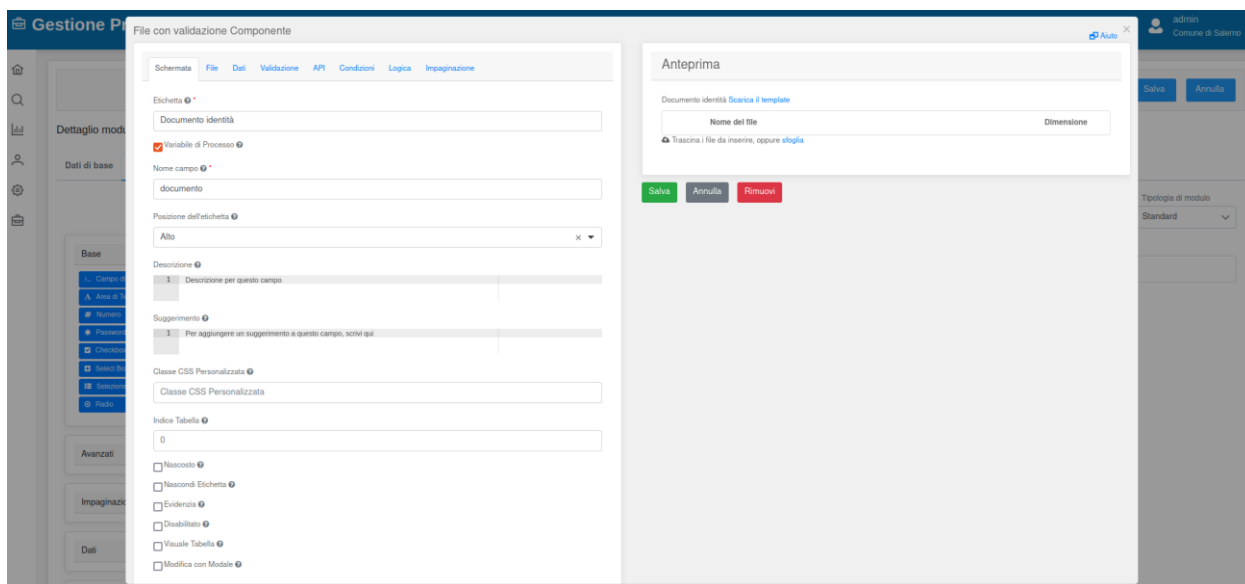
```
{
  "condition": "AND",
  "rules": [
    {
      "id": "VARIABLE|nome|string",
      "field": "VARIABLE|nome",
      "type": "string",
      "input": "text",
      "operator": "equal",
      "value": "carlo"
    }
  ],
  "valid": true
}
```

## 32 GESTIONE DI UN TEMPLATE DI DOCUMENTO

Gestione Pratiche definisce un nuovo tipo custom di componente “File con validazione”, che permette di validare un file pdf rispetto ad un template.

File con validazione estende il componente File di Formio, con in più due proprietà:

- 1) **Template documento:** serve a selezionare il template da utilizzare per la validazione
- 2) **Nome link download template:** serve a definire la label da mostrare nel componente per indicare il link di download del template.



## 33 CREAZIONE DI UN DELEGATO

Gestione Pratiche, tramite il servizio pubblico per avviare una nuova istanza di procedura, permette di definire i delegati per quella procedura.

POST `{{baseApiUrl}}/api/public/proceduremanager/start/{id}`

Il servizio richiede il seguente body come da specifica swagger. E' stato aggiunto il parametro **delegateRequesters** che rappresenta la lista dei delegati.

**POST** `/api/public/proceduremanager/start/{id}` Avvia una nuova istanza della procedura

l'esecuzione coinvolge il servizio **ProcedureInstanceService** (metodo: `startInstance`) per comunicare con **Camunda BPM**

Funzionamento:

1. cerca la procedura richiesta tramite l'id
2. trova l'ultima versione della pubblicazione della pratica richiesta
3. crea e popola istanza
4. salva istanza nel database
5. crea submission specificando:
  - oggetto istanza - richiedente - riferimento al primo form del istanza del processo
6. salva la submission sul database

**Parameters** [Try it out](#)

Name	Description
<b>id</b> <small>required</small> integer (path)	<input type="text" value="id"/>

**Request body** required application/json

**Example Value** [Schema](#)

```
{
  "requester": {
    "identifier": "string",
    "firstName": "string",
    "lastName": "string"
  },
  "delegateRequesters": [
    {
      "identifier": "string",
      "firstName": "string",
      "lastName": "string"
    }
  ],
  "variables": [
    {
      "type": "unknown",
      "name": "string",
      "value": true
    },
    {
      "type": "unknown",
      "name": "string",
      "value": [
        "string"
      ]
    }
  ]
}
```

## 34 PROTOCOLLAZIONE CON SERVICE TASK

Gestione Pratiche definisce un servizio che permette di generare il numero di protocollo di una pratica, il servizio richiede il seguente body:

- 1) **externalProcedureInstanceId**: rappresenta l'identificativo della istanza di procedura
- 2) **servicePluginProtocolDelegateUniqueIdentifier**: rappresenta l'identificativo del servizio di protocollazione(*services.plugins.sime12.protocolNumberGenerator*, *services.plugins.development.protocolNumberGenerator*).
- 3) **generateSubProtocols**: indica se si vuole generare anche i protocollo "figli".

**PUT** **/api/bpm/protocolnumber/generate** Genera numero di protocollo

l'esecuzione coinvolge il servizio **BpmServiceTaskDelegateService** (metodo: generateProtocolNumber) per comunicare con **workflow engine bpm**

Funzionamento:

1. genera numero di protocollo

**Parameters** Try it out

No parameters

**Request body** required application/json

**Example Value** | Schema

```
{
  "externalProcedureInstanceId": "string",
  "generateProtocolNumberModel": {
    "servicePluginProtocolDelegateUniqueIdentifier": "string",
    "generateSubProtocols": true
  }
}
```

## 35 GENERAZIONE REPORT DI PROCEDURA DA SERVICE TASK

In questo capitolo descriviamo la generazione del report di una istanza di procedura e del numero di protocollo tramite service task. Descriveremo in dettaglio i seguenti servizi:

Generazione del report di procedura tramite service task

Generazione del numero di protocollo tramite service task

### 35.1 Generazione del report di procedura

Gestione Pratiche permette di generare il report di una procedura utilizzando il formato jrxml.

**POST** **/api/bpm/procedure/report/generate** Genera il report di una istanza di procedura ^

l'esecuzione coinvolge il servizio **BpmServiceTaskDelegateService** (metodo: generateReport) per comunicare con il **documnetale**

Funzionamento:

- genera il report di una istanza di procedura

Parameters

No parameters

Try it out

Request body **required**

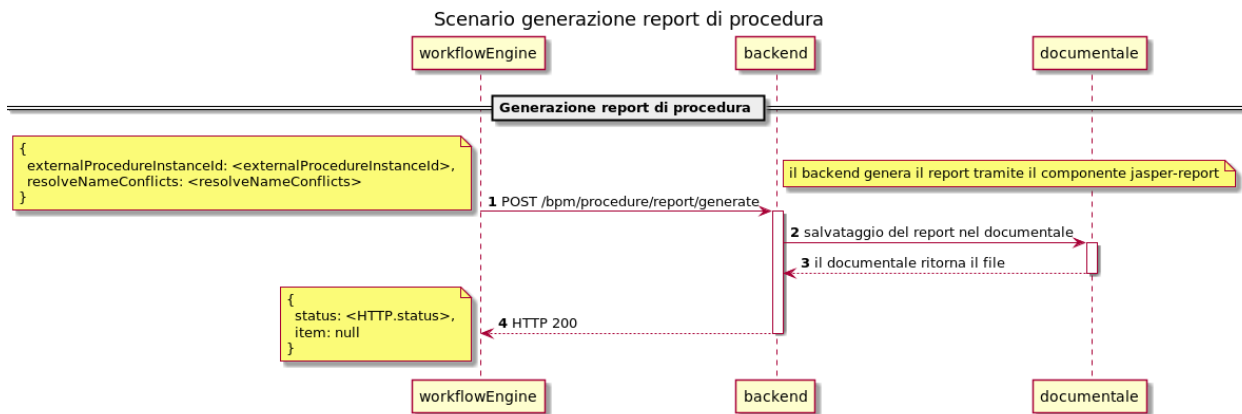
application/json

Example Value | Schema

```
{
  "externalProcedureInstanceId": "string",
  "resolveNameConflicts": true
}
```

Il servizio richiede il seguente body:

- 1) **externalProcedureInstanceId** (obbligatorio) rappresenta l'identificativo dell'istanza di processo (workflow).
- 2) **resolveNameConflicts** (obbligatorio) indica se nel salvataggio nel documentale del report si vuole sovrascrivere il file (**false**) oppure creare una nuova versione (**true**).



## 35.2 Generazione numero di protocollo

Gestione Pratiche permette di generare il numero di protocollo per una istanza di procedura.

**PUT**
**/api/bpm/protocolnumber/generate**
Genera numero di protocollo

l'esecuzione coinvolge il servizio **BpmServiceTaskDelegateService** (metodo: generateProtocolNumber) per comunicare con **workflow engine bpm**

Funzionamento:

- genera numero di protocollo

**Parameters**
Try it out

No parameters

**Request body** required
application/json

**Example Value** | Schema

```

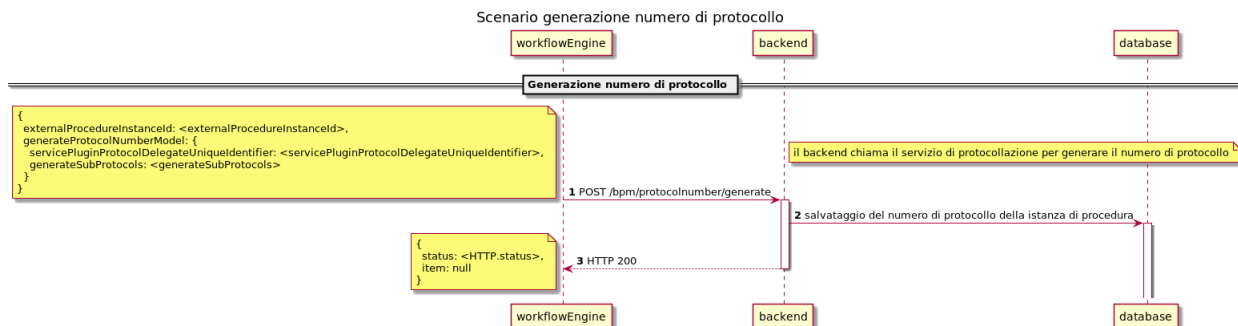
{
  "externalProcedureInstanceId": "string",
  "generateProtocolNumberModel": {
    "servicePluginProtocolDelegateUniqueIdentifier": "string",
    "generateSubProtocols": true
  }
}
  
```

Il servizio richiede il seguente body:

- externalProcedureInstanceId** (obbligatorio) rappresenta l'identificativo dell'istanza di processo (workflow).

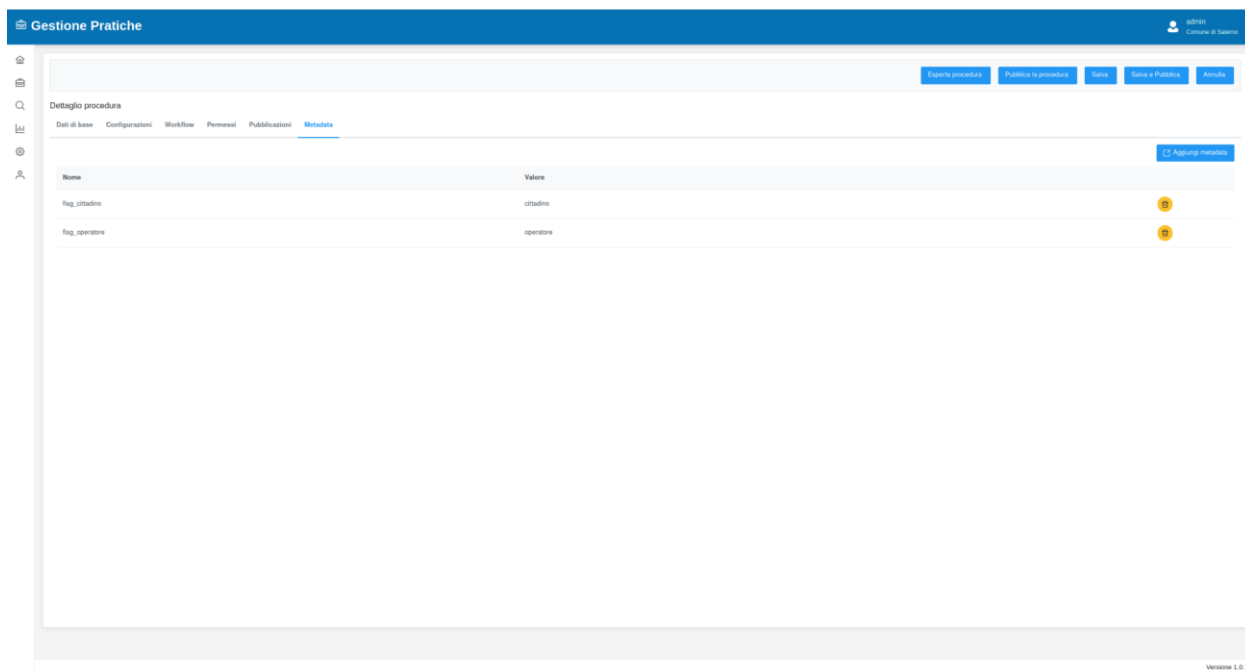


- 2) **generateProtocolNumberModel** (obbligatorio) rappresenta il modello che identifica quale plugin utilizzare per la generazione del numero di protocollo. Il campo **generateSubProtocols** indica se si vuole generare anche i protocollo figli a partire dal protocollo generato all'avvio dell'istanza di procedura.



## 36 CREAZIONE DI METADATA

Gestione Pratiche permette definire i metadata di una procedura come ad esempio flag richiesti per la procedura.



I metadata sono disponibili anche lato cittadino quando si recuperano le procedure pubblicate.

## 37 CREAZIONE DI LOOKUP

Gestione Pratiche permette la creazione di valori di lookup che potranno essere utilizzati all'interno dei moduli.

**Gestione Pratiche** admin Contatore di Sistema

Dettaglio lookup Salva Annulla

**Dati di base** Dettaglio lookup

Nome \*

Richiedi

Description \*

Richiedi

Versione 1.0.7

**Gestione Pratiche** admin Contatore di Sistema

Dettaglio lookup Salva Annulla

**Dati di base** **Dettaglio lookup** Apertura dettaglio lookup

Nome	Valore
Esterno	Pacchetto
Interno	Pacchetto

Versione 1.0.7

## 38COMANDO PER ATTIVAZIONE/DISATTIVAZIONE PROXY E VISUALIZZAZIONE LOG

Per poter attivare/disattivare il proxy bisogna, dapprima, collegarsi in SSH alla sessione dedicata al proxy (per esempio per il proxy 1 è [centos@156.54.45.16](#) → centos@haproxy1-centos ) e in seguito digitare il seguente comando:

```
sudo systemctl command haproxy
```

dove ***command*** sarà:

- start: per attivare il proxy;
- stop: per disattivare il proxy;
- status: per vedere lo stato del proxy.

Infine, per poter visualizzare i log (cartella /var/log/haproxy.log) occorre digitare il comando:

```
!tail
```


I log saranno mostrati come di seguito:


```
centos@haproxy1-centos7 ~$ !tail
tail -f /var/log/haproxy.log
Sep  7 08:50:45 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39041 10.0.254.25:34648 GET /client/scheda_news.aspx?news=40325&prov=766&stile=7 HTTP/1.1 404\"
Sep  7 08:50:45 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39087 10.0.254.25:34652 GET /themes/custom/italiagov/assets/icons/sprite.svg HTTP/1.1 200\"
Sep  7 08:50:45 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39089 10.0.254.25:34654 GET /sites/default/files/Logo+Salerno+HD_0_0.png HTTP/1.1 404\"
Sep  7 08:50:45 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39097 10.0.254.25:34660 POST /core/modules/statistics/statistics.php HTTP/1.1 200\"
Sep  7 08:50:46 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39095 10.0.254.25:34656 POST /contextual/render HTTP/1.1 200\"
Sep  7 08:50:46 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39093 10.0.254.25:34658 GET /allegati/30790.pdf HTTP/1.1 404\"
Sep  7 08:50:47 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39105 10.0.254.25:34668 GET /sites/default/files/2022-08/AVVISI%20PROGRESSIONE%20A%20B%20A%20C1.pdf HTTP/1.1 304\"
Sep  7 08:50:48 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39099 10.0.254.25:34672 GET /sites/default/files/Logo+Salerno+HD_0_0.png HTTP/1.1 404\"
Sep  7 08:50:49 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39041 10.0.254.25:34676 POST /contextual/render HTTP/1.1 200\"
Sep  7 08:50:50 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39107 10.0.254.25:34680 GET /sites/default/files/Logo%20Salerno%20HD_0_0.png HTTP/1.1 200\"
Sep  7 08:50:51 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39113 10.0.254.25:34686 GET /allegati/30797.pdf HTTP/1.1 404\"
Sep  7 08:50:56 haproxy1-centos7 haproxy[2079]: "10.0.254.10:20311 -:- GET /client/scheda.aspx?scheda=4845&stile=26&ti=4 HTTP/1.1 302\"
Sep  7 08:50:57 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39127 10.0.254.25:34704 GET /node/19941 HTTP/1.1 200\"
Sep  7 08:50:57 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39129 10.0.254.25:34708 GET /sites/default/files/css/css_10_PC0EYAbCbMvOHCEVn9QYUM_APzUCKXScPj52kgxM.css HTTP/1.1 200\"
Sep  7 08:50:57 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39127 10.0.254.25:34710 GET /sites/default/files/js/js_mM5Mh6abFCUUMDIv9dVYHbGhZofQ2xoNFRYVxbII0jI.js HTTP/1.1 200\"
Sep  7 08:50:57 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39129 10.0.254.25:34716 POST /core/modules/statistics/statistics.php HTTP/1.1 200\"
Sep  7 08:50:57 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39123 10.0.254.25:34712 POST /contextual/render HTTP/1.1 200\"
Sep  7 08:50:57 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39125 10.0.254.25:34718 POST /core/modules/statistics/statistics.php HTTP/1.1 200\"
Sep  7 08:50:57 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39127 10.0.254.25:34714 POST /contextual/render HTTP/1.1 200\"
Sep  7 08:50:57 haproxy1-centos7 haproxy[2079]: "10.0.254.10:20317 -:- GET /amministrazioneTrasparente/organizzazione/sanzioni-mancata-comunicazione-dei-dati HTTP/1.1 302\"
Sep  7 08:50:58 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39139 -:- GET /client/scheda.aspx?scheda=4845&stile=26&ti=4 HTTP/1.1 503\"
Sep  7 08:50:59 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39141 -:- GET /amministrazioneTrasparente/organizzazione/sanzioni-mancata-comunicazione-dei-dati HTTP/1.1 503\"
Sep  7 08:50:59 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39143 10.0.254.25:34728 GET /client/menu.aspx?menu=3444&stile=4&ti=45 HTTP/1.1 404\"
Sep  7 08:51:00 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39145 10.0.254.25:34726 GET /admin/appearance/settings/italiagov HTTP/1.1 200\"
Sep  7 08:51:00 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39145 10.0.254.25:34740 GET /sites/default/files/css/css_Fq_eP8vQ_YzUwts-FEjVJa8BBih-DNLtn3N78U1kAE.css HTTP/1.1 200\"
Sep  7 08:51:00 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39147 10.0.254.25:34730 GET /admin/appearance/settings/bootstrap-italia HTTP/1.1 200\"
Sep  7 08:51:01 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39153 10.0.254.25:34738 GET /admin/appearance/settings/seven HTTP/1.1 200\"
Sep  7 08:51:01 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39145 10.0.254.25:34746 GET /core/misc/icons/bebebe/hamburger.svg HTTP/1.1 200\"
Sep  7 08:51:01 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39159 10.0.254.25:34748 GET /core/misc/icons/bebebe/chevron-disc-left.svg HTTP/1.1 200\"
Sep  7 08:51:01 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39147 10.0.254.25:34750 GET /modules/contrib/admin_toolbar/admin_toolbar_tools/misc/icons/ffffff/drupal-9-logo.svg HTTP/1.1 200\"
Sep  7 08:51:01 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39155 10.0.254.25:34752 GET /core/misc/icons/787878/file.svg HTTP/1.1 200\"
Sep  7 08:51:01 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39157 10.0.254.25:34754 GET /core/misc/icons/787878/puzzlepiece.svg HTTP/1.1 200\"
Sep  7 08:51:01 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39153 10.0.254.25:34756 GET /sites/default/files/js/js_600jd2oRlj0Hhw_ozMhs7_g18BtoJ6w5Wjk3R5jh0U.js HTTP/1.1 200\"
Sep  7 08:51:01 haproxy1-centos7 haproxy[2079]: "10.0.254.10:39153 10.0.254.25:34758 GET /admin/appearance/settings HTTP/1.1 200\"
```


## 39 CONFIGURAZIONE DI WSO2


Per accedere all'identity server utilizzare le seguenti credenziali:


- 1) username: admin
- 2) password: admin

Management Console  
[Sign-in](#) | [Docs](#) | [About](#)

[User Guide](#)  
WSO2 Carbon user guide.

[Forum](#)  
The interactive message board for sharing information, questions and comments about WSO2 products.

[Issue Tracker](#)  
Users are encouraged to report issues & suggest improvements using the JIRA issue tracker. In addition, users can observe the status of the reported issues in progress.

[Mailing Lists](#)  
Report issues, provide feedback & get help from our mailing lists.

Sign-in

Username

admin

Password

\*\*\*\*\*

☐ Remember Me

Sign-in

[Sign-in Help](#)

Identifichiamo i service provider come applicazioni. Alcune applicazioni da questo elenco non espongono API, in quanto servono solo per configurare l'accesso tramite protocollo OIDC.

L'ambiente di produzione espone i seguenti service provider:

**admin\_GestionePratiche Application\_SANDBOX:** Service Provider per l'applicazione Gestione Pratiche

**admin\_PortaleIstituzionale Applicazione\_SANDBOX:** Service Provider per l'applicazione Portale Istituzionale

**admin\_AppMobile\_SANDBOX:** Service Provider per l'applicazione App Mobile

**admin\_Moodle Applicazione\_PRODUCTION:** Service Provider per l'applicazione Moodle Applicazione

**admin\_Alfresco\_SANDBOX:** Service Provider per l'applicazione Alfresco

**admin\_PortaleMoodle\_SANDBOX:** Service Provider per l'applicazione Portale Moodle

**admin\_LaPiazza\_SANDBOX:** Service Provider per l'applicazione LaPiazza

L'ambiente di collaudo espone i seguenti service provider:

**admin\_Gestionepratiche Applicazione\_SANDBOX:** Service Provider per l'applicazione Gestione Pratiche

**admin\_AppMobile\_SANDBOX:** Service Provider per l'applicazione Applicazione Mobile

**admin\_PortaleIstituzionale Applicazione\_SANDBOX:** Service Provider per l'applicazione Portale Istituzionale

**admin\_Alfresco\_SANDBOX:** Service Provider per l'applicazione Alfresco

R.T.I. Al maviva S.p.A/ Al mawave S.p.a./ Indra Italia S.p.A/ Intellera Consulting S.r.l.	Sistema Pubblico di Connettività LOTTO 4
Documentazione Tecnica	SPCL4_COMUNESALERNO_PICS_ManualeUtente_GestorePratiche

**admin\_Moodle Applicazione\_SANDBOX:** Service Provider per l'applicazione Moodle Applicazione

**admin\_PortaleMoodle\_SANDBOX:** Service Provider per l'applicazione Portale Moodle

**admin\_LaPiazza\_SANDBOX:** Service Provider per l'applicazione LaPiazza

## 40AUTENTICAZIONE, AUTORIZZAZIONE E PAGAMENTI

La gestione dell'autenticazione, autorizzazione e pagamento sono argomenti trattati separatamente.

## 41 INVIO E-MAIL DA UN WORKFLOW

Per inviare una email da un workflow, è possibile farlo nelle seguenti modalità:

### 41.1 Invio tramite Script Task

Per inviare una e-mail tramite script task, è possibile utilizzare uno script come nell'esempio di seguito in linguaggio Groovy:

```
groovy-script-sendmail.groovy

import org.camunda.bpm.engine.variable.value.TypedValue
import javax.mail.*
import javax.mail.internet.*

def to = "destinatario@mailserver-example.com"
def subject = "Soggetto"
def body = "Corpo email"
def from = "from-address@mailrelay.example"
def password = "PASSWORD-SERVER"
def host = "HOST-SERVER-ADDRESS"

// Configura la sessione SMTP/SMTPS
Properties props = new Properties()
props.put("mail.smtp.auth", "true")
props.put("mail.smtp.starttls.enable", "true")
props.put("mail.smtp.host", host)
props.put("mail.smtp.port", "587")
props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory") //in caso di SMTPS
Session session = Session.getInstance(props,
    new javax.mail.Authenticator() {
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(from, password);
        }
    });

// Crea il messaggio di posta elettronica
Message message = new MimeMessage(session);
message.setFrom(new InternetAddress(from));
message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(to));
message.setSubject(subject);
message.setText(body);

// Invia il messaggio di posta elettronica
Transport.send(message);
```

Per utilizzare questo script, ci si deve assicurare che nel server che ospita Camunda siano disponibili le librerie richieste, ovvero:

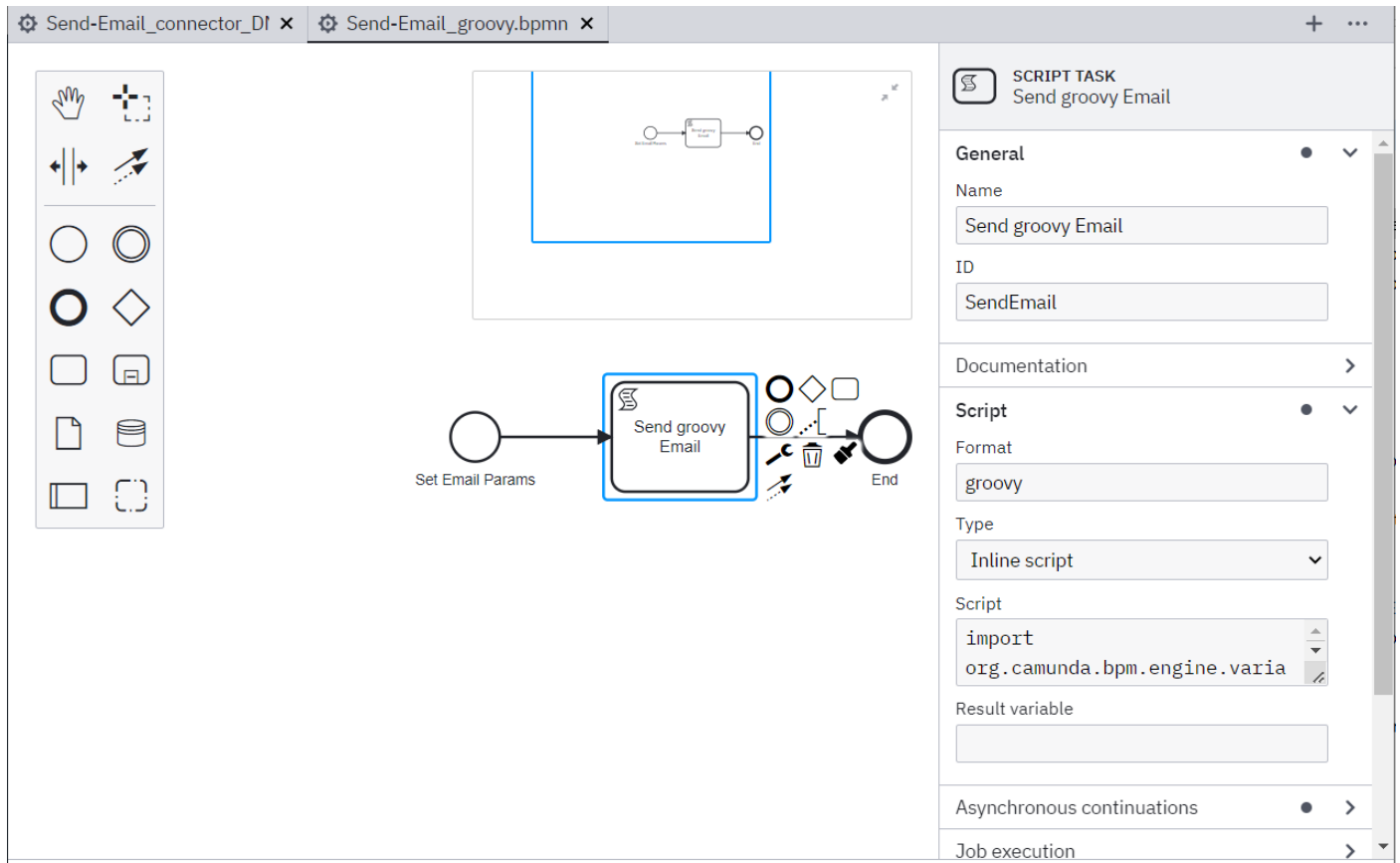
- javax.mail.jar
- activation-1.1.1.jar

Nel caso non fossero presenti, va richiesto all'amministratore di scaricarle, ad esempio da:

1. Maven Central Repository:  
<https://mvnrepository.com/artifact/javax.activation/activation/1.1.1>
2. JavaMail Project:  
<https://java.net/projects/javamail/pages/Home>

Di seguito una configurazione di esempio dello script:



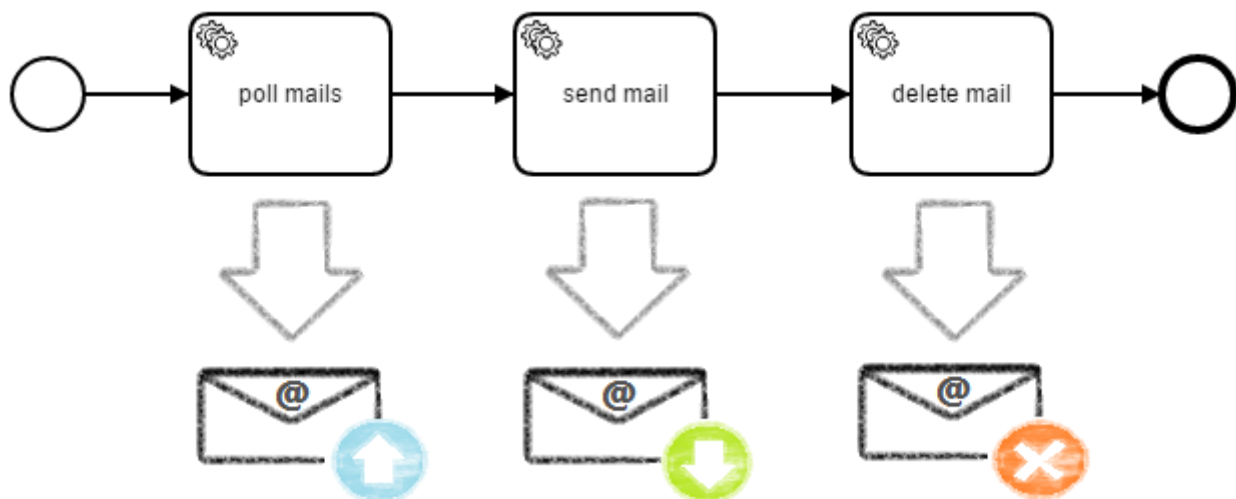


## 41.2 Invio Tramite Installazione e Configurazione di un Connector Open-Source

In alternativa allo script, è possibile installare un componente open-source sull'engine di Camunda, creato dalla community appositamente per inviare email:

[GitHub - camunda-community-hub/camunda-platform-7-mail: Mail connectors for Camunda Platform 7](https://github.com/camunda-community-hub/camunda-platform-7-mail)

Questo componente permette di effettuare diverse operazioni, rappresentate di seguito e riportare sulla pagina web di riferimento:



R.T.I. Almaviva S.p.A/ Almawave S.p.a./ Indra Italia S.p.A/ Intellera Consulting S.r.l.	Sistema Pubblico di Connettività LOTTO 4
Documentazione Tecnica	SPCL4_COMUNESALERNO_PICS_ManualeUtente_GestorePratiche

Per le caratteristiche complete, fare riferimento alla documentazione ufficiale.

## 42 GENERAZIONE DINAMICA DEI REPORT (MULTI-PDF)

GestionePratiche permette di generare report “dinamici” durante l’esecuzione del processo di workflow engine. Sono necessari alcuni passaggi per la configurazione del service task.

Prima della pubblicazione della procedura, bisogna definire i report aggiuntivi che si vogliono utilizzare durante il processo di workflow engine.

The screenshot shows the 'Gestione Pratiche' interface with the 'Configurazioni' tab selected. The 'Report di procedura' section is expanded, showing a list of reports. The 'report dinamico' report is selected. The interface includes a sidebar with navigation icons and a top bar with the user's name and role.

È necessario aggiungere ai permessi della procedura i permessi per i report generati. Se si vogliono generare 3 report diversi, bisogna aggiungere 3 permessi.

The screenshot shows the 'Gestione Pratiche' interface with the 'Permessi' tab selected. The 'Permessi documenti' section is expanded, showing a table of permissions for variables. The table has columns for 'Nome variabile', 'Visibilità', and 'Personalizzata'. The 'report\_dinamico' variable is highlighted with a yellow circle.

Nome variabile	Visibilità	Personalizzata
documento	Cittadino e operatore	no
report_dinamico	Cittadino e operatore	si
documento_apertura_domanda	Solo operatore	no

Ogni report ha un identificativo che deve essere definito alla creazione del report di procedura.

Versione 2.1.2

## 42.1 Descrizione API

GestionePratiche espone un service task per la generazione dinamica dei report.

API:  
GET `{{baseUrl}}/api/bpm/procedure/report/dynamic/generate`

Questa API permette di generare il report di una istanza di procedura.

**POST** `/api/bpm/procedure/report/dynamic/generate` Genera il report di una istanza di procedura

L'esecuzione coinvolge il servizio **BpmServiceTaskDelegateService** (metodo: generateReport) per comunicare con il **documentale**

Funzionamento:

- genera il report di una istanza di procedura

**Parameters** [Try it out](#)

No parameters

**Request body** *required* application/json

**Example Value** | [Schema](#)

```
{
  "externalProcedureInstanceId": "string",
  "dynamicReportModel": {
    "reportIdentifier": "string",
    "fileName": "string",
    "variableName": "string",
    "useDataSource": true,
    "resolveNameConflicts": true
  }
}
```

**Responses**

Code	Description	Links
200	OK	No links

Media type: application/json

Content Accept Header:

**Example Value** | [Schema](#)

```
{
  "status": "OK",
  "item": {}
}
```

Sono richiesti i seguenti parametri nel body:

- **reportIdentifier**: rappresenta l'identificativo del report che è stato descritto nella sezione precedente.
- **fileName**: rappresenta il nome che si vuole dare al report
- **variableName**: rappresenta il nome della variabile associata al report. (Se si vogliono creare report diversi, bisogna definire per ogni report una variabile)
- **useDataSource**: rappresenta la stessa funzionalità che si ha a disposizione durante la creazione di un report.
- **resolveNameConflicts**: se a true viene gestita la risoluzione dei nomi per i file nel documentale, se a false viene creata una nuova versione del file.

Esempio di request body:

```
{
  "externalProcedureInstanceId": "6a98cdbc-f86f-11ec-b79f-0242ac190004",
  "dynamicReportModel": {
    "reportIdentifier": "report_dinamico",
    "fileName": "report_name",
    "variableName": "report_variable",
    "useDataSource": false,
    "resolveNameConflicts": false
  }
}
```

## 43 ASSOCIAZIONE PROTOCOLLO-DOCUMENTI

GestionePratiche permette di assegnare ai documenti un numero di protocollo e di salvare la configurazione del protocollo. Oltre al numero di protocollo è possibile, quindi, memorizzare anche proprietà aggiuntive tramite i service task per la protocollazione dei documenti. Vengono forniti due service task, una versione sincrona e una versione asincrona.

API:  
POST {{baseUrl}}/api/bpm/document/register

**POST** /api/bpm/document/register Protocolla uno o più documenti dell'istanza di procedura

L'esecuzione coinvolge il servizio **BpmServiceTaskDelegateService** (metodo: registerDocuments) per comunicare con **workflow engine bpm**

Funzionamento:

1. protocolla i documenti dell'istanza di procedura

**Parameters**

No parameters

**Request body** required application/json

**Example Value** | Schema

```
{
  "externalProcedureInstanceId": "string",
  "registerDocumentModel": {
    "protocolNumber": "string",
    "protocolConfiguration": {
      "additionalProp1": "string",
      "additionalProp2": "string",
      "additionalProp3": "string"
    },
    "documentNames": [
      "string"
    ]
  }
}
```

- **externalProcedureInstanceId**: rappresenta l'id della istanza di processo. Utilizzando il workflow engine può essere recuperata con *execution.getProcessInstanceId()*
- **registerDocumentModel**: rappresenta il modello di dati richiesti così composto:
  - o **protocolNumber**: campo obbligatorio, rappresenta il numero di protocollo
  - o **documentNames**: campo opzionale, rappresenta la lista dei nomi dei documenti che è possibile fornire in input se non si vogliono protocollare tutti i documenti dell'istanza.

- **protocolConfiguration**: è una `Map<String,String>` che rappresenta la configurazione aggiuntiva. I campi “additionalPropx” possono essere sostituiti da una stringa qualsiasi.

API:  
POST `{{baseUrl}}/api/bpm/document/register/async`

La versione asincrona utilizza lo stesso payload della versione sincrona.

**N.B Le API appena descritte sono accessibili solo da workflow engine context**

### 43.1 Caso d'uso

Di seguito si presenta il caso d'uso in cui:

- A. Vengono protocollati tutti i documenti inseriti alla valle della submit;
- B. Vengono protocollati esclusivamente i documenti inseriti nello user task interno al workflow.

Il bpmn costruito per il caso d'uso è il seguente:



1. Form Iniziale: il cittadino inserisce due allegati;
2. Generazione report;
3. Associazione protocollo-documenti (tutti quelli inseriti nel punto 1.): nel caso d'uso i documenti sono protocollati con numero 'PROT-01'.
4. User task in cui si possono allegare ulteriori due documenti.
5. Associazione protocollo-documenti (solo quelli inseriti al punto 4.): nel caso d'uso questi documenti saranno protocollati con numero 'PROT-02'.

Il cittadino inserisce i due allegati:

Invio Completato

Elenco dichiarazioni [Clicca qui per scaricare il file](#)

	Nome del file	Dimensione
✕	SPCL4_COMUNESALERNO_PICS_DICHIARAZIONI.pdf	130.09 kB

Asseverazione [Clicca qui per scaricare il file](#)

	Nome del file	Dimensione
✕	SPCL4_COMUNESALERNO_PICS_RELAZIONE_TECNICA_ASSEVERAZIONE.pdf	260.30 kB

A valle della submit parte il workflow e vengono eseguiti gli script task dei punti 2 e 3; in particolare lo script task di associazione documento-protocollo viene riportato di seguito:

```
import groovy.json.*;
import org.camunda.bpm.engine.impl.persistence.entity.ExecutionEntity;
import org.camunda.bpm.engine.impl.persistence.entity.TaskEntity;

def endpointURL = "http://10.0.254.7/gestionepratiche/api/bpm/document/register/async"4
def httpConnector = org.camunda.connect.Connectors.http();

def payload = JsonOutput.toJson([ "externalProcedureInstanceId": execution.getProcessInstanceId(),
```

<sup>4</sup> Viene utilizzata la versione asincrona in quanto va eseguita a seguito della submit.



```

        "registerDocumentModel": [
            {
                "protocolNumber": "PROT-01",
                "protocolConfiguration": [
                    {
                        "oggettoProtocollazione": "OGGETTO PROT TEST",
                        "dataProtocollazione": "29/05/2023"
                    }
                ]
            }
        ]
    });










println("Payload          : " + payload);
println("Endpoint        : " + endpointURL);

try {
    def resp = httpConnector.createRequest()
        .post()
        .header("gptenantid", "1")
        .header("Accept", "application/json")
        .header("origin", "http://camunda")
        .header("Access-Control-Expose-Headers", "gpauthorization")
        .url(endpointURL)
        .contentType("application/json")
        .payload(payload)
        .execute();
    def result = resp.getResponse();
    println("Response Status Code : " + resp.getStatusCode());
    println("Response Headers   : " + resp.getHeaders());
    println("Response Body       : " + resp.getResponse());
    resp.close();
} catch (Exception ex) {
    println("ERROR: La chiusura della procedura ha restituito un errore. Il BPMN continua, ma l'errore è loggato di seguito: " + ex.getMessage());
}

```

Questo script permette di associare a tutti i documenti inseriti nella form iniziale (compreso il report) il protocollo relativo ('PROT-01' in questo caso).

Questa relazione può essere verificata nel dettaglio della pratica, nel tab Documenti:

Nome	Dimensione (MB)	Data di creazione	Data ultima modifica	Versione	
report_globale.pdf	0.19	18 apr 2023, 16:27:28	18 apr 2023, 16:27:29	1.1	  
asseverazione_tecnica.pdf	0.27		2023, 16:27:27	1.0	  
elenco_dichiarazioni.pdf	0.13		2023, 16:27:27	1.0	  

Protecollazioni del documento

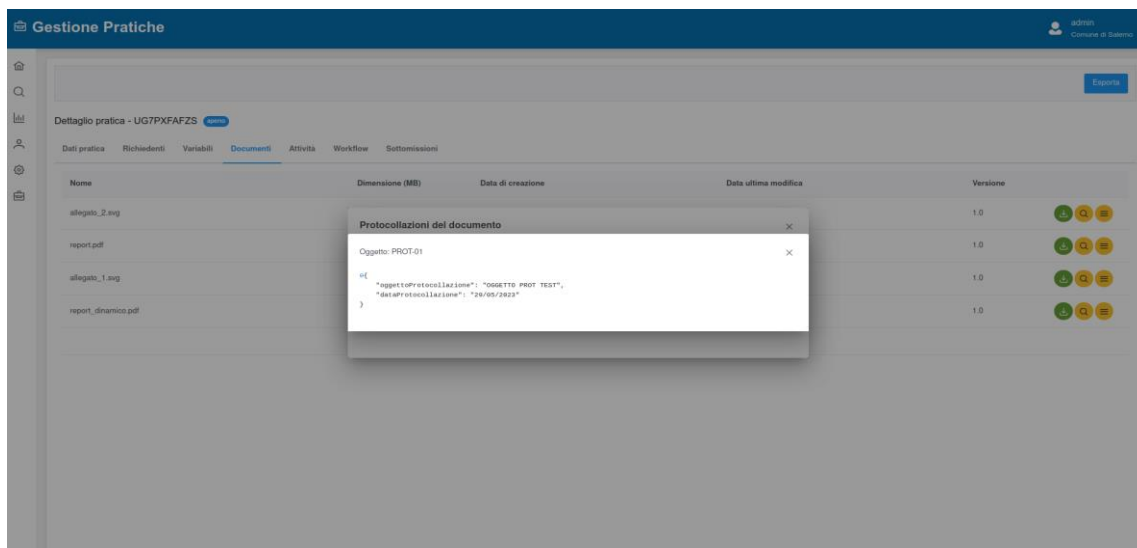
Nome

JKAEGUAQV5

PROT-01

<< < 1 > >>

Per visualizzare la configurazione si accede al dettaglio dell'istanza di procedura, e per ogni documento, viene mostrata la lista di protocollo e la relativa configurazione se esistente.



Avanzando la pratica, il cittadino inserisce ulteriori due allegati (punto 4.): quando viene eseguito lo script task del punto 5, i due documenti vengono correlati al protocollo specificato nello script.

Nome	Dimensione (MB)	Data di creazione	Data ultima modifica	Versione
report_globale.pdf	0.19	18 apr 2023, 16:27:28	18 apr 2023, 16:40:03	1.2
allegato_1.pdf	0.11	18 apr 2023, 16:40:01	18 apr 2023, 16:40:01	1.0
allegato_2.pdf	0.12	18 apr 2023, 16:40:01	18 apr 2023, 16:40:01	1.0
asseverazione_tecnica.pdf	0.27	18 apr 2023, 16:27:27	18 apr 2023, 16:27:27	1.0
elenco_dichiarazioni.pdf	0.13	18 apr 2023, 16:27:27	18 apr 2023, 16:27:27	1.0

Protocollo del documento

Nome

PROT-02

Per completezza si riporta lo script task che esegue l'associazione del protocollo 'PROT-02' ai documenti di interesse.

```
import groovy.json.*;
import org.camunda.bpm.engine.impl.persistence.entity.ExecutionEntity;
import org.camunda.bpm.engine.impl.persistence.entity.TaskEntity;

def endpointURL = "http://10.0.254.7/gestionepratiche/api/bpm/document/register"
//def endpointURL = "http://10.0.254.7/gestionepratiche/api/bpm/document/register/async"
def httpConnector = org.camunda.connect.Connectors.http();

def payload = JsonOutput.toJson([ "externalProcedureInstanceId": execution.getProcessInstanceId(),
                                  "registerDocumentModel": [

    "protocolNumber": "PROT-02",

    "documentNames": [

        "allegato_1.pdf", "allegato_2.pdf"

    ]

  ])
```

5 Rappresenta la lista dei nomi dei documenti che è possibile fornire in input se non si vogliono protocollare tutti i documenti dell'istanza. La nomenclatura di output del documento deve essere scelta in fase di configurazione del file nella maschera di Form.io ('Nome da assegnare al file' nel tab File)

```
    ]
  });

println("Payload          : " + payload);
println("Endpoint        : " + endpointURL);

try {
    def resp = httpConnector.createRequest()
        .post()
        .header("gptenantid", "1")
        .header("Accept", "application/json")
        .header("origin", "http://camunda")
        .header("Access-Control-Expose-Headers", "gpauthorization")
        .url(endpointURL)
        .contentType("application/json")
        .payload(payload)
        .execute();
    def result = resp.getResponse();
    println("Response Status Code : " + resp.getStatusCode());
    println("Response Headers    : " + resp.getHeaders());
    println("Response Body       : " + resp.getResponse());
    resp.close();
} catch (Exception ex) {
    println("ERROR: La chiusura della procedura ha restituito un errore. Il BPMN continua, ma l'errore è loggato  
di seguito: " + ex.getMessage());
}
```

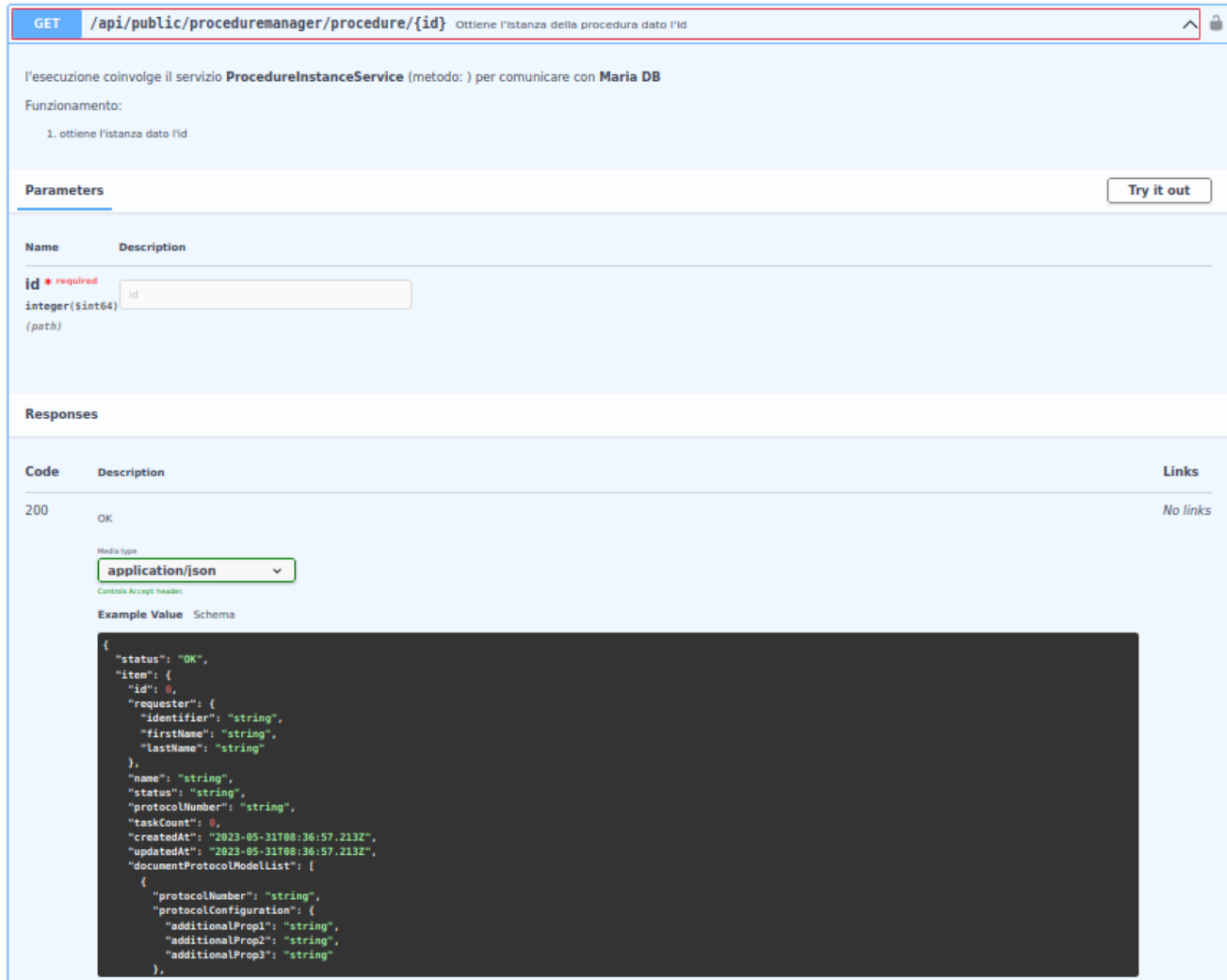
**NOTA:** il protocolNumber sarà valorizzato con il valore recuperato dalla response dell'invocazione a simel2.

## 43.2 Visualizzazione documenti

GestionePratiche permette la visualizzazione dei documenti di una pratica.

API:  
GET `{{baseUrl}}/api/bpm/proceduremanager/procedure/{{id}}`

Questa API fornisce il dettaglio di una istanza di procedura, in particolare anche la lista dei suoi documenti con il relativo numero di protocollo e la configurazione relativa.



Esempio di risposta:

```
{
  "status": "OK",
  "item": {
    "id": 557,
    "requester": {
      "identifier": "amministratore",
      "firstName": "amministratore",
      "lastName": "amministratore"
    },
    "name": "protocollazione documenti",
    "status": "OPEN",
    "protocolNumber": null,
    "taskCount": 0,
    "createdAt": "2023-04-14T13:07:36.000+00:00",
    "updatedAt": "2023-04-14T13:10:42.000+00:00",
    "documentProtocolModelList": [
```

```
{
  "protocolNumber": "PROT-01",
  "protocolConfiguration": {
    "oggettoProtocollazione": "OGGETTO PROT TEST",
    "dataProtocollazione": "29/05/2023"
  },
  "documentIdList": [
    "007c2762-fea8-4e62-86f8-18f12a1ab0d0;1.0",
    "273106a9-0c74-422e-8d76-bc252a7e5fc4;1.0",
    "de259e5d-ecc2-4e55-97ed-76a7d9ec026b;1.0",
    "e5b20b97-8702-4eb6-9a68-a39bfdc974ac;1.0"
  ]
},
{
  "procedureId": 665
}
}
```

API:  
GET {{baseUrl}}/api/bpm/proceduremanager/procedure/document/list

Questa API permette di visualizzare la lista dei documenti di una istanza di procedura

- `proceduralInstanceId`: rappresenta l'id della istanza di procedura

**GET** `/api/public/proceduremanager/procedure/document/list` Ottiene la lista di documenti di una istanza di procedura

L'esecuzione coinvolge il servizio **ProcedureInstanceService** per comunicare con **Camunda BPM**

Funzionamento:

- ottiene la lista di documenti di una istanza di procedura

**Parameters** Try it out

Name	Description
<b>proceduralInstanceId</b> * required integer(\$int64) (query)	<input type="text" value="proceduralInstanceId"/>

**Responses**

Code	Description	Links
200	OK	No links

Media type:

Controls Accept header.

[Example Value](#) | [Schema](#)

```
{
  "status": "OK",
  "item": [
    {
      "id": "string",
      "name": "string",
      "size": 0,
      "minetype": "string",
      "version": "string",
      "createdAt": "2023-04-14T13:41:27.734Z",
      "updatedAt": "2023-04-14T13:41:27.734Z"
    }
  ]
}
```

Esempio di response:

```
{
  "status": "OK",
  "item": [
    {
      "id": "7cc26439-2d0d-428f-9d38-9968e0c86965;1.0",
      "name": "Gestore Pratiche - Requisiti v. 0.4.docx",
      "size": 1054731,
      "mimetype": "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
      "version": "1.0",
      "createdAt": "2023-04-14T13:07:40.051+00:00",
      "updatedAt": "2023-04-14T13:07:42.645+00:00"
    },
    {
      "id": "cedf02eb-e78c-489b-a4d4-031b2248a402;1.0",
      "name": "Gestore Pratiche - Requisiti v. 0.1.docx",
      "size": 336174,
      "mimetype": "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
      "version": "1.0",
      "createdAt": "2023-04-14T13:10:40.867+00:00",
      "updatedAt": "2023-04-14T13:10:42.043+00:00"
    }
  ]
}
```

Di seguito si mostra un esempio di visualizzazione lato portale:

## Elenco documenti

### Numero Protocollo

540

### Del

30/05/2023

### Oggetto

Richiesta di attribuzione /  
verifica del numero civico

[- Dettaglio protocollo](#)

### Nome

### Creato il

stralcio\_aerofotogrammetrico.pdf

30/05/2023 14:53:28

[Scarica](#)

documento\_di\_riconoscimento.pdf

30/05/2023 14:53:29

[Scarica](#)

visura\_catastale.pdf

30/05/2023 14:53:28

[Scarica](#)

report\_toponomastica.pdf

30/05/2023 14:53:31

[Scarica](#)

## 44 VISUALIZZAZIONE LOG DI IMPORTAZIONE AVANZATA PROCEDURE

GestionePratiche permette di visualizzare i log dopo che è stata eseguita una importazione avanzata di procedura, che ne ha creato o aggiornato una istanza.

Il log mostra gli step in dettaglio che hanno preceduto la creazione o l'aggiornamento di una procedura.

The screenshot shows the 'Gestione Pratiche' interface. At the top, there's a blue header with the title 'Gestione Pratiche' and a user profile 'amministratore Comune di Salerno'. Below the header, there's a sidebar with navigation icons. The main content area displays a table titled 'Elenco formati importazione pratiche'. The table has four columns: 'Nome', 'Tipo', 'Data di creazione', and 'Ultimo aggiornamento'. There is one row with the name 'Importazione semplice', type 'CSV', creation date '21 set 2022, 16:11:51', and last update '24 apr 2023, 10:49:53'. To the right of the table, there are three circular icons: a blue one with a download arrow, a yellow one with a list icon, and a green one with a plus icon. At the bottom right of the interface, the version 'Versione 2.1.4' is indicated.

Nome	Tipo	Data di creazione	Ultimo aggiornamento
Importazione semplice	CSV	21 set 2022, 16:11:51	24 apr 2023, 10:49:53

The screenshot shows the 'Gestione Pratiche' interface with a modal window titled 'Lista log di importazione' open. The modal contains a table with three columns: 'Nome', 'Utente', and 'Data di creazione'. There are five rows of log entries, each with a green circular icon containing a download arrow to its right. The log entries describe the generation of reports for document collation. At the bottom right of the interface, the version 'Versione 2.1.4' is indicated.

Nome	Utente	Data di creazione
RUN-protocollo documenti con generazione report-27fed8c-9f49-427a-a631-d28591978825.log	amministratore	27 apr 2023, 14:27:16
RUN-protocollo documenti con generazione report-3f26b2f-8809-4c64-82f-604666561ace.log	amministratore	27 apr 2023, 12:58:43
RUN-protocollo documenti con generazione report-f9a61d18-c045-48ed-ba2e-9d9178beb67.log	amministratore	26 apr 2023, 18:15:33
RUN-protocollo documenti con generazione report-9471c1de-2943-404e-9102-73043865c7b5.log	amministratore	26 apr 2023, 14:29:27
RUN-protocollo documenti con generazione report-088ac68b-21e6-4f74-a06f-a8ca166ca1ea.log	amministratore	26 apr 2023, 12:52:35

L'import viene tracciato anche nel tab 'Sottomissioni' della procedura che ha subito l'import.

## 45 CONFIGURAZIONE TIMEOUT VERIFICA TASK

### 45.1 Configurazione timeout verifica task

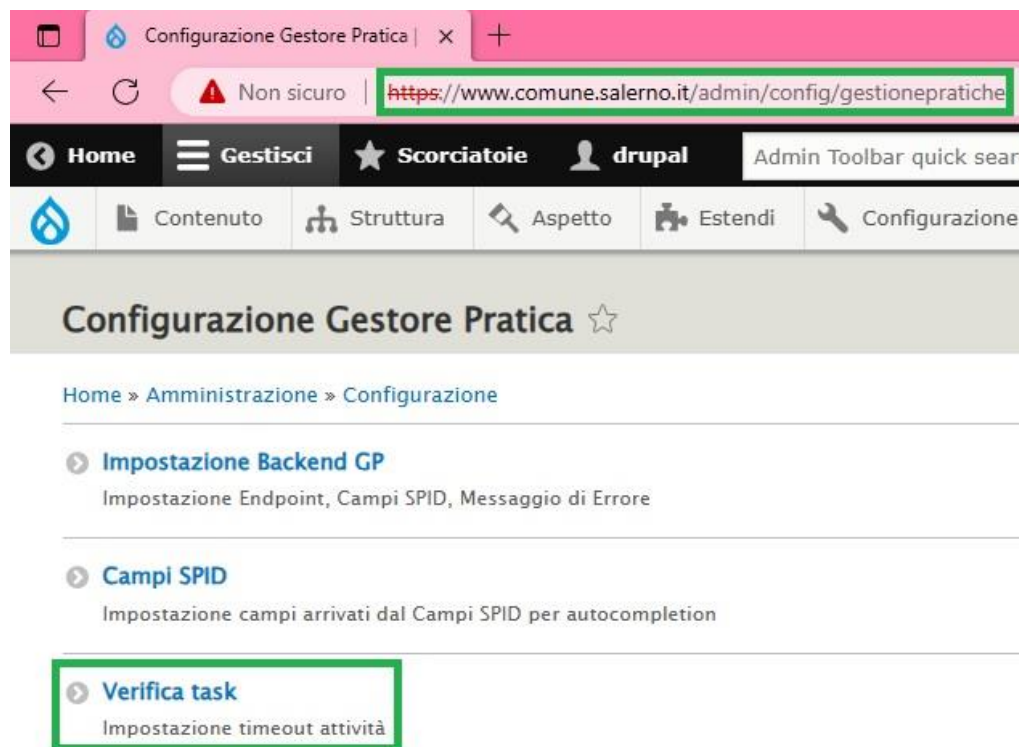
**NB: Per visualizzare il pannello effettuare l'accesso come amministratore.**

Accedere al pannello di configurazione del gestore pratiche (back-end di Drupal), presente al link:

<https://www.comune.salerno.it/admin/config/gestorepratiche>

In questo pannello è presente una sezione chiamata Verifica task, cliccare su tale sezione per accedere al pannello di configurazione della verifica dei task per il cittadino. Tale meccanismo è illustrato a figura 1.

In figura è possibile visualizzare una anteprima della configurazione per la verifica dei task, tale configurazione permette di esprimere un tempo (indicato in secondi) durante il quale effettuare la verifica della presenza di task, allo scadere di tale tempo se la verifica risponde sempre "false", l'utente proseguirà con il flusso normale, considerando che non sono presenti task (attività) a cui deve rispondere/compilare. Viceversa, se è presente un task, l'utente (alla prima risposta "true" della verifica) sarà automaticamente reindirizzato alla compilazione di tale task.





## Check Task Configuration ☆

[Home](#) » [Amministrazione](#) » [Configurazione](#) » [Configurazione Gestore Pratica](#) » [Impostazione Gestione Pratiche](#)

### CONFIGURAZIONE VERIFICA TASK

Verifica timeout (in secondi) \*

30 ▼

Seleziona il timeout (in secondi) della verifica.

Salva configurazione

## Check Task Configuration ☆

[Home](#) » [Amministrazione](#) » [Configurazione](#) » [Config](#)

### CONFIGURAZIONE VERIFICA TASK

Verifica timeout (in secondi) \*



30 ▼

10  
30  
60  
90  
180  
240  
360



il timeout (in secondi) della verifica.

Configurazione

Lato portale, a valle della submit di una procedura, il cittadino vedrà la seguente schermata:




**Comune di Salerno**  
Hippocratica Civitas

[Seguici su](#)  

[Amministrazione](#) [Aree Tematiche](#) [Documenti e dati](#) [Luoghi](#) [Novità](#) [Elezioni politiche 2022](#) [Tutti gli argomenti](#)

[Home](#) / [Tutte Le Pratiche](#) / [Tutte Le Pratiche](#) / [Nuova Pratica](#)

❗ **Verifica!**

 Verifica della presenza di attività, attendere prego...