
Guida alla Console di Gestione

Release 3.3.15

Link.it

29 lug 2024

Contents

1	Introduzione	1
1.1	I Profili di Interoperabilità	1
1.2	Le entità di configurazione dei servizi	2
1.3	Il processo di configurazione dei servizi	3
1.4	Profilo Utente	4
1.5	Informazioni confidenziali	6
2	Profilo “API Gateway”	11
2.1	Definizione delle API	11
2.2	Registrazione dell’erogazione	14
2.3	Registrazione della fruizione	20
2.4	Versionamento delle API	22
2.5	Configurazione dell’API	28
2.6	Sospensione API	31
2.7	Connettore	33
2.8	Gestione CORS	58
2.9	Differenziare le configurazioni specifiche per risorsa/azione	59
2.10	Controllo degli Accessi	60
2.11	Rate Limiting	100
2.12	Validazione dei messaggi	127
2.13	Caching Risposta	130
2.14	Sicurezza a livello del messaggio	130
2.15	Trasformazioni	132
2.16	Tracciamento	144
2.17	Correlazione Applicativa	145
2.18	MTOM	148
2.19	Registrazione Messaggi	149
2.20	Proprietà	152
3	Profilo “ModI”	155
3.1	Concetti Preliminari	155
3.2	Sicurezza Canale	157
3.3	Sicurezza Messaggio	160
3.4	Pattern di Interazione	267
4	Profilo “eDelivery”	287
4.1	Passi preliminari di configurazione	287

4.2	Erogazione di servizi in modalità eDelivery	288
4.3	Fruizione di servizi in modalità eDelivery	291
4.4	Generazione del PMODE Domibus	292
5	Profilo “SPCoop”	293
5.1	Configurazione di un servizio SPCoop	293
5.2	Profili Asincroni	296
5.3	Interfacce WSDL (concettuale, logico ed implementativo)	304
5.4	Profili di gestione della busta eGov	304
6	Profilo “Fatturazione Elettronica”	307
6.1	Fatturazione Passiva	308
6.2	Fatturazione Attiva	311
7	Strumenti	315
7.1	Runtime	315
7.2	Auditing	316
8	Configurazione	321
8.1	Generale	321
8.2	Cache	342
8.3	Cache PDND	346
8.4	Controllo del Traffico	346
8.5	Tracciamento	352
8.6	Registrazione Messaggi	389
8.7	Rate Limiting	390
8.8	Token Policy	403
8.9	Attribute Authority	425
8.10	Tags	433
8.11	Utenti	435
8.12	Importa	440
8.13	Esporta	441
8.14	Auditing	443
9	Errori generati da GovWay	447
9.1	Classificazione degli Errori	449
9.2	REST Problem Details - RFC 7807	464
9.3	SOAP Fault	465
9.4	Attivazione di Codici di Errore Specifici	467
10	Funzionalità Avanzate	477
10.1	Modalità Avanzata	477
10.2	Configurazione manuale delle interfacce	478
10.3	Versionamento delle API e delle Erogazioni/Fruizioni	481
10.4	Modalità di identificazione dell’azione	481
10.5	Multi-Tenant	484
10.6	Header di Integrazione	485
10.7	Connettori	494
10.8	Device PKCS11	509
10.9	Online Certificate Status Protocol (OCSP)	510
10.10	Correlazione tra transazioni differenti	510
10.11	Opzioni Avanzate per Erogazioni/Fruizioni	511
10.12	Gestione Proxy	511
10.13	Autenticazione e Autorizzazione Principal (Security Constraint)	517
10.14	Aggiunta di Claims nei Token	519

10.15 Accesso alle proprietà delle entità del Registro	520
10.16 Espressioni XPath su messaggi JSON	522
10.17 Validazione dei messaggi con OpenAPI 3.x	524
10.18 Cifratura delle Password	525
10.19 Visualizzazione delle Informazioni Confidenziali Cifrate	526
10.20 Cifratura delle Informazioni Confidenziali	527
10.21 Health Check	530

CHAPTER 1

Introduzione

Questo manuale documenta le funzionalità e le modalità d'uso della *Console di Gestione* del prodotto *GovWay* (<http://govway.org>).

Nota: Oltre alla console di Gestione, GovWay mette a disposizione dei gestori una seconda console utilizzata per il monitoraggio delle richieste applicative gestite dal gateway. Per informazioni sulle modalità di utilizzo della Console di Monitoraggio si rimanda alla relativa manualistica distribuita con il prodotto.

Nel prosieguo si assume che il prodotto GovWay sia già correttamente installato e la console di gestione sia accessibile via browser dai Gestori del Sistema.

L'indirizzo standard della Console di Gestione è *http://ip:porta/govwayConsole*, che dovrà essere correttamente perfezionato con ip e porta del proprio ambiente di installazione. Per informazioni sulle modalità di installazione si rimanda alla relativa manualistica distribuita con il prodotto.

Nota: L'accesso alle diverse funzionalità della console è sempre mediato da un sistema di autorizzazione che verifica che l'utente sia in possesso dei dovuti permessi. Le istruzioni operative sulla gestione degli utenti e la configurazione dei permessi sono descritte nella sezione *Utenti*.

1.1 I Profili di Interoperabilità

GovWay si differenzia dagli API Gateway tradizionali per essere progettato in conformità con i principali profili di interoperabilità in uso nella Pubblica Amministrazione italiana ed europea. Per tale motivo, le modalità di configurazione del prodotto si differenziano in funzione dello specifico profilo a cui le API debbano conformarsi. I profili di interoperabilità supportati dalla distribuzione standard del prodotto sono i seguenti:

- *API Gateway*: è il profilo di interoperabilità di base che consente di supportare qualunque generica API basata su scambio di messaggi SOAP e REST.

- *ModI*: è il profilo che consente di supportare gli scenari di comunicazione basati sul Modello di Interoperabilità rilasciato da AGID, che fornisce i requisiti per l'integrazione tra il sistema informativo complessivo della Pubblica Amministrazione, Cittadini e Imprese.
- *eDelivery*: è il profilo standard adottato a livello europeo nell'ambito del progetto *CEF*, e basato sul protocollo AS4.
- *SPCoop*: il profilo *SPCoop* è il profilo basato sull'uso della busta eGov e sulla Porta di Dominio, recentemente deprecato da AGID, ma ancora in uso per la quasi totalità dei servizi centrali erogati dalla Pubblica Amministrazione italiana.
- *Fatturazione Elettronica*: questo profilo supporta le modalità di scambio delle fatture elettroniche, nel formato FatturaPA, veicolate tramite il Sistema di Interscambio.

In fase di installazione possono essere scelti i profili di proprio interesse (per default viene proposto il solo profilo di API Gateway).

Durante l'utilizzo della Console di Gestione è preferibile selezionare il profilo di interoperabilità adeguato in base al tipo di configurazioni sui quali si lavora. La selezione del profilo di interoperabilità, tramite il menù presente in testata (Fig. 1.1), comporta la visualizzazione dei soli elementi dell'interfaccia, e relativi dati, attinenti con tale profilo.



Figure 1.1: Selezione del profilo di interoperabilità

Nota: La selezione del profilo tramite il menù presente in testata non è persistente e al successivo login verrà nuovamente presentato il profilo di interoperabilità di default associato al profilo utente. Per modificarlo si rimanda alla sezione [Profilo Utente](#).

Esiste la possibilità (non consigliata) di operare sulla console selezionando il profilo *Tutti*. In tal caso non saranno applicati filtri sui contenuti e le maschere di visualizzazione e di configurazione potranno apparire più complesse di quanto avviene selezionando lo specifico profilo su cui si sta lavorando.

Nota: Ulteriori profili sono programmabili in GovWay ed alcuni di questi sono in uso in importanti progetti della pubblica amministrazione, come la Porta di Comunicazione del Sistema di Interscambio del Mercato dell'Energia.

1.2 Le entità di configurazione dei servizi

Prima di descrivere le entità di configurazione presenti nel registro è importante chiarire il concetto di *Dominio* cui alcuni elementi di configurazione fanno riferimento. Il dominio rappresenta il confine logico (tipicamente un ente amministrativo) entro il quale sono racchiuse le risorse applicative da condividere con l'esterno. Nel seguito si fa distinzione tra i seguenti:

- *Dominio Gestito*: l'insieme delle risorse applicative i cui flussi di comunicazione sono sotto il controllo del GovWay di propria gestione.
- *Dominio Esterno*: Insieme di risorse applicative esterne al dominio gestito.

Le principali entità di configurazione del Registro sono:

- *API*

Descrizione formale dei flussi di comunicazione previsti da un dato servizio, erogato o fruito nel proprio dominio. Ad ogni API è assegnata una singola modalità operativa e, in base ad essa, sarà fornita una descrizione formale delle interfacce di dialogo supportate. Ad esempio saranno forniti WSDL/XSD per le interfacce Soap o un file YAML in formato Swagger per quelle Rest.

- *Erogazione*

Registrazione di una specifica istanza di servizio che un soggetto del dominio interno eroga in accordo alle interfacce applicative descritte da un set di API censito nel registro.

- *Fruizione*

Registrazione di una specifica istanza di servizio che un soggetto del dominio interno fruisce in accordo alle interfacce applicative descritte da un set di API censito nel registro.

- *Soggetto*

Entità che rappresenta la singola organizzazione, o ente amministrativo, coinvolto nei flussi di comunicazione. Ciascun soggetto censito nel registro può appartenere al dominio interno o esterno e può avere associata un'unica modalità operativa.

- *Applicativo*

Entità per censire i client, riferiti ad uno specifico soggetto (e quindi modalità), che fruiscono di servizi. Censire un applicativo è indispensabile nei casi in cui l'identificazione è necessaria per poter superare i criteri di autenticazione autorizzazione specificati nella configurazione del *Controllo degli Accessi* per ciascun servizio fruito.

- *Ruolo*

Entità per censire i ruoli che possono essere utilizzati nell'ambito del controllo degli accessi per costruire specifici criteri di autorizzazione. I ruoli possono avere origine interna al registro oppure essere passati da un sistema esterno, sia in contesti fruizione che di erogazione.

- *Scope*

Entità per censire gli scope che possono essere utilizzati nell'ambito del controllo degli accessi per costruire specifici criteri di autorizzazione basato sui token.

1.3 Il processo di configurazione dei servizi

Le sezioni successive del documento illustrano i passi necessari per realizzare le configurazioni necessarie per rendere operativi i flussi di erogazione/fruizione dei servizi nei diversi profili di interoperabilità supportati.

Per semplificare il processo di configurazione, nel caso di configurazioni per l'interoperabilità con le note piattaforme di erogazione di servizi centralizzate, GovWay mette a disposizione specifici package, denominati *Govlet*. Il Govlet, attraverso un modello di tipo wizard, consente all'utente di fornire i dati necessari a produrre le entità di configurazione per uno specifico servizio. I Govlet disponibili possono essere acquisiti dal sito di Govway al seguente indirizzo <http://www.govway.org/govlets>. Alcuni esempi di Govlet:

- *FatturaPA - Fatturazione Attiva*: configurazione del servizio per l'invio di fatture elettroniche al Sistema d'Interscambio (SDI).
- *FatturaPA - Fatturazione Passiva*: configurazione del servizio per la ricezione di fatture elettroniche dal Sistema d'Interscambio (SDI).
- *SIOPE+*: configurazione del servizio per l'invio degli ordinativi di pagamento alla piattaforma SIOPE+ e ricezione delle relative notifiche e giornale di cassa.

- *pagoPA*: configurazione del servizio per l’accesso alla piattaforma dei pagamenti elettronici pagoPA.

Una volta entrati in possesso del Govlet è necessario eseguirlo sulla govwayConsole tramite la funzione *Importa* descritta nella sezione *Importa*.

Per procedere manualmente alla produzione delle configurazioni per i servizi, si utilizzano le funzionalità presenti nella sezione *Registro* della GovWayConsole. Il processo manuale di configurazione può essere schematizzato nei passi seguenti:

1. *Definizione delle API*. Il primo passo prevede la definizione delle API relative ai servizi che si vogliono utilizzare. In questa fase tipicamente si provvede al caricamento del descrittore formale delle interfacce (WSDL, WADL, ...).
2. *Registrazione dell’erogazione o fruizione*. Il secondo passo, dopo aver registrato l’API del servizio, prevede la creazione di una Erogazione, o di una Fruizione, a seconda del ruolo previsto nell’interazione col servizio.
3. *Configurazione Specifica*. Le interfacce della GovWayConsole sono state progettate in modo che, il completamento dei primi due passi di configurazione, sia sufficiente a disporre di una configurazione funzionante del servizio. Il terzo, e quindi opzionale passo, consiste nella produzione di tutti i dettagli aggiuntivi di configurazione che sono necessari alla particolare situazione.

In questo passo si forniscono i dettagli delle funzionalità aggiuntive, che riguardano:

- *Controllo degli Accessi*: indicazione dei criteri di autenticazione e autorizzazione necessari per l’accesso al servizio.
- *Validazione*: processo di validazione dei messaggi in transito sul gateway.
- *Sicurezza Messaggio*: misure di sicurezza al livello del messaggio richieste.
- *Tracciamento*: personalizzazione delle tracce prodotte nel corso dell’elaborazione delle richieste di servizio.
- *Registrazione Messaggi*: indicazione dei criteri di salvataggio degli elementi che compongono le richieste di servizio (payload, header, allegati, ...).

La Fig. 1.2 descrive lo scenario generale in cui opera GovWay.

La sezioni successive descrivono in dettaglio il processo di configurazione di cui sopra, fornendo i dettagli specifici per ciascun profilo di interoperabilità.

1.4 Profilo Utente

Durante l’utilizzo della Console di Gestione è preferibile configurare il profilo dell’utenza più opportuno all’utilizzo che se ne intende fare rispetto ai parametri descritti di seguito.

- **Modalità Interfaccia**: consente di decidere quale modalità, tra standard e avanzata, è quella di default per l’utenza. Le due modalità si differenziano come segue (per maggiori dettagli si rimanda alla sezione *Modalità Avanzata*):
 - la modalità standard prevede varie semplificazioni, sulle opzioni visualizzate nelle schermate, mirate al compimento delle operazioni di uso comune;
 - operando in modalità avanzata, in ciascuno dei contesti di configurazione descritti in questo manuale, compariranno opzioni aggiuntive per le quali sono previsti valori di default nel caso della modalità standard.
- **Profilo Interoperabilità**: consente di impostare un profilo di interoperabilità di default associato all’utente tra quelli descritti nella sezione *I Profili di Interoperabilità*;
- **Soggetto Operativo**: voce presente solamente se è stato selezionato un profilo di interoperabilità, consente di associare un soggetto operativo di default all’utente.

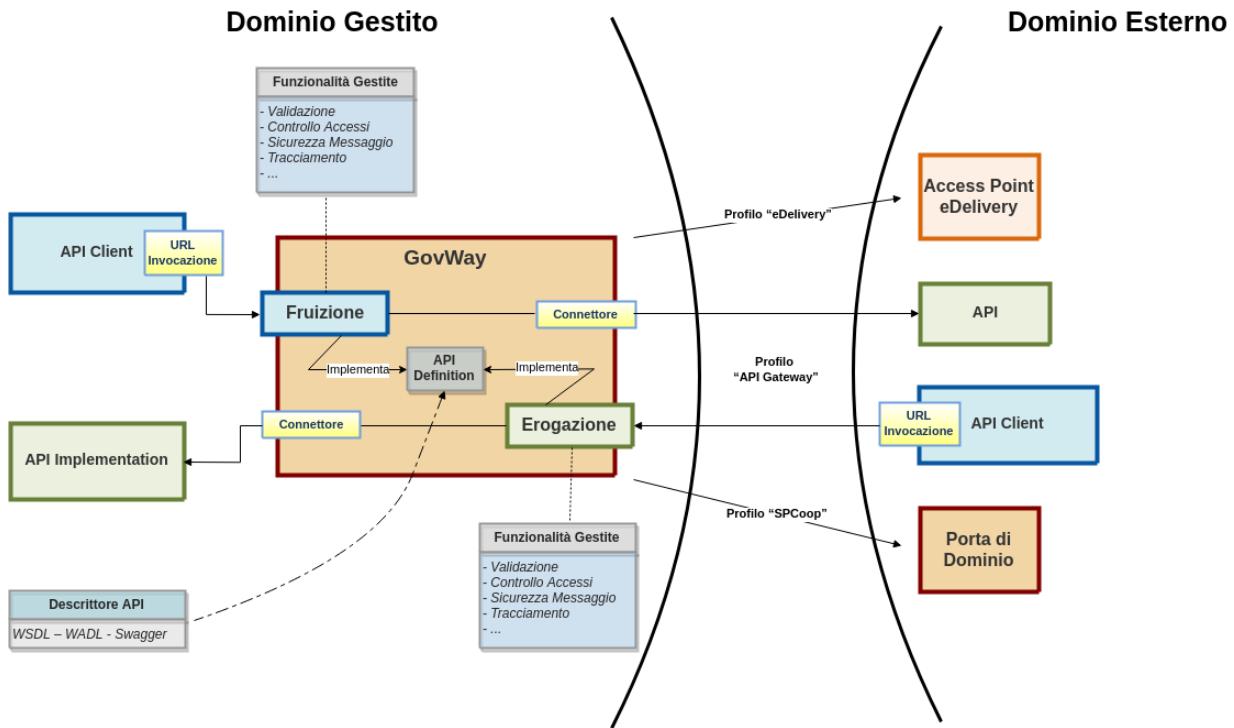


Figure1.2: Scenario Generale

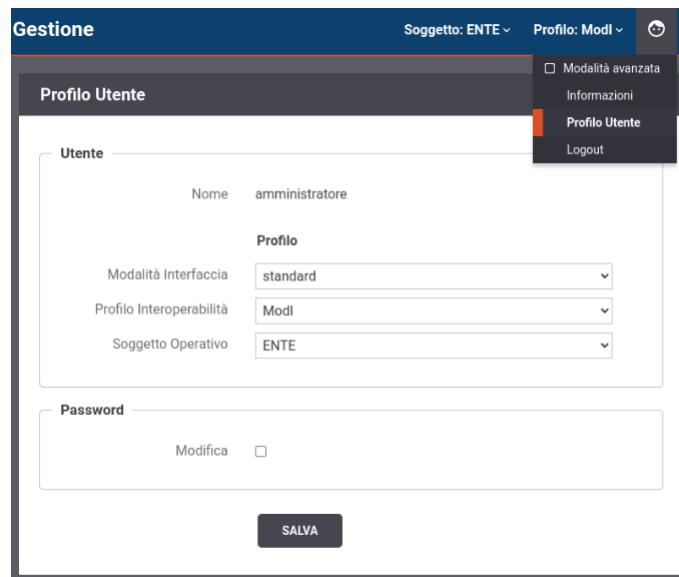


Figure1.3: Profilo Utente della Console di Gestione

Nota: La modalità di utilizzo dell’interfaccia, il profilo di interoperabilità e il soggetto operativo sono modificabili anche una volta effettuato il login sulla console, agendo nelle voci presenti nel menù in alto a destra. Le modifiche attuate in questa modalità non sono persistenti e al successivo login verranno nuovamente presentate le scelte impostate come default nel profilo utente.

È infine possibile modificare la password associata all’utente.

1.5 Informazioni confidenziali

Nella base dati di GovWay sono presenti diverse informazioni confidenziali, descritte nel seguito di questa sezione suddividendole per categorie di appartenenza.

- *Credenziali di entità censite nel registro di GovWay* che verranno utilizzate dalle applicazioni client per invocare le API esposte su GovWay o dagli utenti per collegarsi alle console di gestione e monitoraggio;
- *Credenziali richieste per l’accesso ad entità terze*; password per accedere a keystore e chiavi private utilizzate per instaurare connessioni HTTPS o per attuare sicurezza di messaggio, credenziali HTTP Basic o API key inviate a un backend, o password per l’accesso a un proxy HTTP;
- *Proprietà censite nel registro di GovWay* che possono contenere informazioni confidenziali e quindi necessitano a loro volta di una cifratura;
- *Keystore su filesystem* riferiti dalle configurazioni utilizzate per instaurare connessioni HTTPS o per attuare sicurezza di messaggio.

1.5.1 Credenziali di entità censite nel registro di GovWay

Si tratta di credenziali utilizzate:

- dai soggetti (*Creazione di un soggetto*) e dagli applicativi (*Creazione di un applicativo*), registrati con *Credenziali “api-key”* o *Credenziali “http-basic”*, per invocare le API esposte da GovWay;
- dagli utenti per accedere alle console di gestione e monitoraggio descritte nella sezione *Utenti*.

Le credenziali sono disponibili solamente nell’avviso visualizzato in seguito alla creazione dell’entità (es. Fig. 2.86) e successivamente non sono più consultabili; in caso di smarrimento è necessario procedere con un aggiornamento della credenziale.

Le credenziali vengono cifrate per default con un algoritmo di cifratura “SHA-512-based Unix crypt (\$6\$)”; maggiori dettagli vengono forniti nella sezione *Cifratura delle Password*.

1.5.2 Credenziali richieste per l’accesso ad entità terze

Rientrano in questa casistica:

- le password per l’accesso ai keystore e alle chiavi private utilizzate nei connettori https (*Autenticazione Https*), nei pattern di sicurezza ModI *Sicurezza Messaggio* e nelle funzionalità di sicurezza messaggio (*Sicurezza a livello del messaggio*);
- le credenziali utilizzate per la connessione verso un backend in una *Autenticazione Http* o una *Autenticazione API Key*;
- la password per l’accesso ad un HTTP *Proxy*.

La gestione delle informazioni sopra descritte varia a seconda se viene attivata la modalità di cifratura delle informazioni confidenziali descritta nella sezione “byokInstallSecurityGovWay” della Guida di Installazione.

Senza una byokInstallSecurityGovWay

La console visualizzerà l’informazione mascherandola come mostrato nella figura [Fig. 1.4](#) e fornendo la possibilità di visualizzarla in chiaro come mostrato nella figura [Fig. 1.5](#).

Un’informazione confidenziale, una volta salvata nella base dati, sarà sempre possibile modificarla cliccando sulla matita presente nella maschera di modifica come mostrato nella figura [Fig. 1.6](#).

Nota: Le informazioni confidenziali, senza una cifratura attiva, verranno memorizzate in chiaro sulla base dati.

Figure1.4: Cifratura delle informazioni confidenziali non abilitata: mascheramento attivo

Figure1.5: Cifratura delle informazioni confidenziali non abilitata: mascheramento non attivo

Figure1.6: Cifratura delle informazioni confidenziali non abilitata: modifica di una informazione precedentemente salvata

Abilitando la byokInstallSecurityGovWay

In fase di creazione di un’entità, le maschere di gestione sono simili a quelle mostrate nelle figure [Fig. 1.4](#) e [Fig. 1.5](#), con la differenza che comparirà un lucchetto aperto a indicare che l’informazione verrà salvata cifrata sulla base dati, come mostrato nella figura [Fig. 1.7](#) e [Fig. 1.8](#).

Cliccando sul lucchetto o procedendo a salvare l’entità che si sta registrando, l’informazione confidenziale verrà cifrata sulla base dati e non sarà più possibile visualizzarla in chiaro tramite la console, come mostrato nella figura [Fig. 1.9](#), dove viene mostrato un lucchetto chiuso.

Cliccando sulla matita sarà invece possibile impostare un nuovo valore.

Nota: Le informazioni confidenziali, con la cifratura attiva, verranno memorizzate cifrate nella base dati e non sarà più possibile visualizzarle in chiaro a meno che non venga abilitata tale possibilità agendo sulla configurazione avanzata della console, descritta nella sezione [Visualizzazione delle Informazioni Confidenziali Cifrate](#).



The screenshot shows a login form titled 'Autenticazione Http'. It has two fields: 'Utente' (User) with the value 'test' and 'Password' (Password) with the value '.....'. To the right of the password field are two icons: a magnifying glass and a lock. The password field is partially obscured by dots, indicating it is masked.

Figure1.7: Cifratura delle informazioni confidenziali abilitata: mascheramento attivo durante la fase di registrazione



The screenshot shows a login form titled 'Autenticazione Http'. It has two fields: 'Utente' (User) with the value 'test' and 'Password' (Password) with the value '123456'. To the right of the password field are two icons: a magnifying glass and a lock. The password field is not masked, showing the actual characters.

Figure1.8: Cifratura delle informazioni confidenziali abilitata: mascheramento non attivo durante la fase di registrazione

1.5.3 Proprietà censite nel registro di GovWay

Nel registro di GovWay è possibile registrare proprietà nome-valore per le seguenti entità di registro:

- API erogata o fruita (*Proprietà*)
- un soggetto
- un applicativo
- configurazione globale (*Proprietà*)

Abilitando la `byokInstallSecurityGovWay`, anche la registrazione delle proprietà può prevedere il salvataggio di alcune proprietà contenenti informazioni confidenziali cifrandole.

Durante la fase di registrazione di una proprietà, nel campo indicante il valore, è presente un lucchetto aperto che consente, cliccandolo, di cifrare il contenuto inserito (Fig. 1.10).

Cliccando sul lucchetto, l'informazione confidenziale verrà cifrata nella base dati e non sarà più possibile visualizzarla in chiaro tramite la console, come mostrato nella figura Fig. 1.11, dove viene visualizzato un lucchetto chiuso.

Cliccando sulla matita sarà invece possibile impostare un nuovo valore.

Nota: La visualizzazione di un valore cifrato non è consentita, a meno che non venga abilitata tale possibilità nella configurazione avanzata della console descritta nella sezione *Visualizzazione delle Informazioni Confidenziali Cifrate*.



The screenshot shows a login form titled 'Autenticazione Http'. It has two fields: 'Utente' (User) with the value 'test' and 'Password' (Password) with the value '.....'. To the right of the password field are two icons: a magnifying glass and a lock. The password field is partially obscured by dots, indicating it is masked.

Figure1.9: Cifratura delle informazioni confidenziali abilitata: informazione salvata nella base dati non più visualizzabile in chiaro

Proprietà

Nome * test

Valore * 123456

Figure1.10: Cifratura delle informazioni confidenziali abilitata: salvataggio di una proprietà

Proprietà

Nome * test

Valore *

Figure1.11: Cifratura delle informazioni confidenziali abilitata: proprietà cifrata

1.5.4 Keystore su filesystem

Rientrano in questa casistica i keystore riferiti tramite path su file-system indicati:

- nei connettori https (*Autenticazione Https*);
- nei pattern di sicurezza ModI (*Sicurezza Messaggio*);
- nelle funzionalità di sicurezza messaggio (*Sicurezza a livello del messaggio*).

Se il keystore riferito è cifrato, deve essere indicato in fase di configurazione uno dei KMS di “unwrap” disponibili per la sua decodifica, scegliendolo tra quelli registrati nella configurazione (*Decodifica di un keystore cifrato tramite una BYOK Policy*).

La figura Fig. 1.12 mostra un esempio di utilizzo di una policy BYOK necessaria per decodificare il keystore cifrato riferito, contenente la chiave e il certificato client da utilizzare in un connettore HTTPS.

Autenticazione Client

Abilitato

Dati Accesso al KeyStore

Tipo

Path * /tmp/testClient.p12

Password *

Password Chiave Privata *

Alias Chiave Privata

Algoritmo * SunX509

BYOK Policy

Figure1.12: Decodifica di un keystore cifrato tramite una BYOK Policy

CHAPTER 2

Profilo “API Gateway”

In questa sezione descriviamo le fasi di configurazione di GovWay al fine di attivare l’erogazione o la fruizione di servizi che rispettano lo standard Soap o Rest. Per semplificare l’utilizzo della console grafica govwayConsole, è consigliabile effettuare la selezione del profilo *API Gateway* tramite l’apposito selettori posto nell’intestazione della pagina.

2.1 Definizione delle API

Indipendentemente che si voglia erogare o fruire un servizio, è necessario iniziare il processo di configurazione con il censimento delle relative API. Questa operazione si effettua sulla govwayConsole posizionandosi nella sezione *Registro > API*.

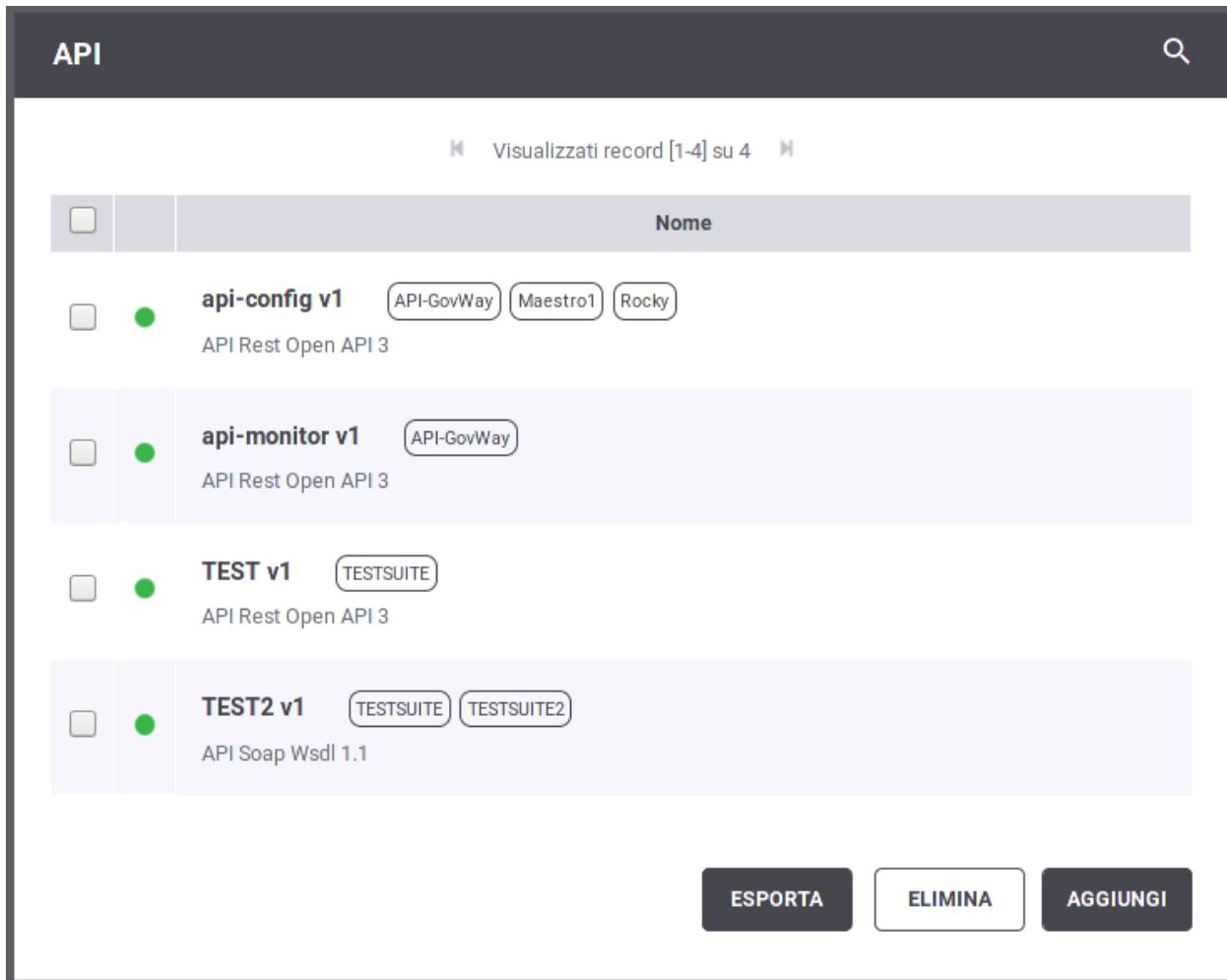
La pagina di ingresso mostra l’elenco delle API eventualmente già presenti in configurazione. Ciascun elemento dell’elenco riporta l’identificativo, il tipo SOAP o REST e il formato del descrittore fornito in configurazione (Fig. 2.1).

Gli elementi dell’elenco possono essere selezionati per l’eliminazione, con il pulsante Elimina, e per l’esportazione, con il pulsante Esporta. La funzione di esportazione è descritta nella sezione *Esporta*.

Si crea una nuova API premendo il pulsante *Aggiungi*.

Compilare il form (Fig. 2.2) inserendo i seguenti dati:

- *Tipo*: Selezionare il tipo delle API a scelta tra «Soap» e «Rest».
- *Nome*: Assegnare un nome che identifichi le API.
- *Descrizione*: un testo opzionale di descrizione.
- *Tags*: un elenco di tag da associare all’API per classificarla. Iniziando a scrivere, vengono proposti i tag già esistenti compatibili.
- *Versione*: progressivo numerico che identifica l’indice di revisione.



The screenshot shows a list of APIs in the 'API' section of the management console. The list is titled 'Visualizzati record [1-4] su 4'. Each API entry includes a checkbox, a green status dot, the API name, its version, supported protocols (e.g., API-GovWay, Maestro1, Rocky, TESTSUITE, TESTSUITE2), and the API type (e.g., API Rest Open API 3, API Soap Wsdl 1.1). The bottom right of the list area features three buttons: 'ESPORTA', 'ELIMINA', and 'AGGIUNGI'.

		Nome	
<input type="checkbox"/>	●	api-config v1	API-GovWay, Maestro1, Rocky
		API Rest Open API 3	
<input type="checkbox"/>	●	api-monitor v1	API-GovWay
		API Rest Open API 3	
<input type="checkbox"/>	●	TEST v1	TESTSUITE
		API Rest Open API 3	
<input type="checkbox"/>	●	TEST2 v1	TESTSUITE, TESTSUITE2
		API Soap Wsdl 1.1	

ESPORTA ELIMINA AGGIUNGI

Figure2.1: Elenco delle API

API > Aggiungi

Note: (*) Campi obbligatori

API

Tipo	Rest
Nome *	HelloAPI
Descrizione	
Tags	tagTest x tag2Test x
Versione	1

Specifiche delle interfacce

Formato Specifica	Open API 3.0
Open API 3.0	<input type="button" value="Browse..."/> No file selected.

SALVA

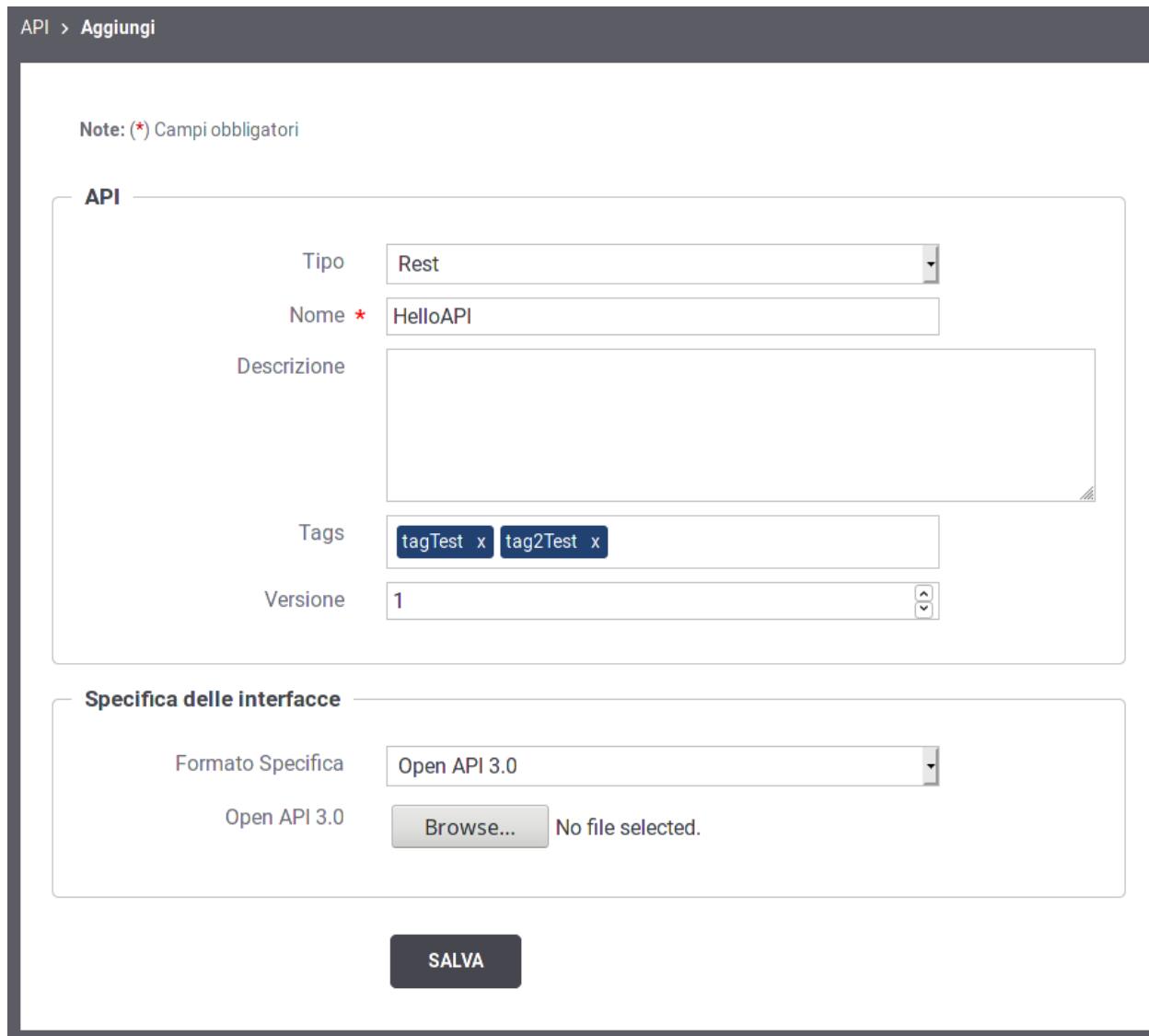


Figure2.2: Definizione di una API

- *Specifica delle Interfacce:* In questa sezione è possibile caricare il descrittore formale dell’interfaccia, analizzando il quale, il gateway produce la corrispondente configurazione. Nel caso di interfacce Soap si potrà caricare il relativo WSDL. Nel caso di interfacce Rest si potrà scegliere tra i formati: WADL, Swagger 2.x e OpenAPI 3.3.

Nel caso non si disponga del descrittore dell’interfaccia è sempre possibile inserire manualmente la relativa configurazione seguendo le modalità descritte alla sezione *Configurazione manuale delle interfacce*.

Effettuato il salvataggio, l’API sarà consultabile all’interno dell’elenco delle API registrate. Accedendo al dettaglio si potranno visionare, a seconda del tipo di API SOAP o REST, rispettivamente i servizi o le risorse che tale API dispone. Nella figura Fig. 2.3 viene riporta l’elenco delle risorse di una API REST.

	Method	Path	Descrizione
<input type="checkbox"/>	POST	/pet	
<input type="checkbox"/>	PUT	/pet	
<input type="checkbox"/>	GET	/pet/findByStatus	
<input type="checkbox"/>	GET	/pet/findByTags	
<input type="checkbox"/>	DELETE	/pet/{petId}	
<input type="checkbox"/>	GET	/pet/{petId}	
<input type="checkbox"/>	POST	/pet/{petId}	
<input type="checkbox"/>	POST	/pet/{petId}/uploadImage	

Figure2.3: Risorse di una API REST

2.2 Registrazione dell’erogazione

Una volta disponibile la definizione delle API, si passa alla registrazione dell’erogazione fornendo i dati di base per l’esposizione del servizio erogato tramite GovWay. In Fig. 2.4 è illustrato graficamente il caso dell’erogazione.

Per registrare l’erogazione del servizio ci si posiziona nella sezione *Registro > Erogazioni* e si preme il pulsante *Aggiungi*.

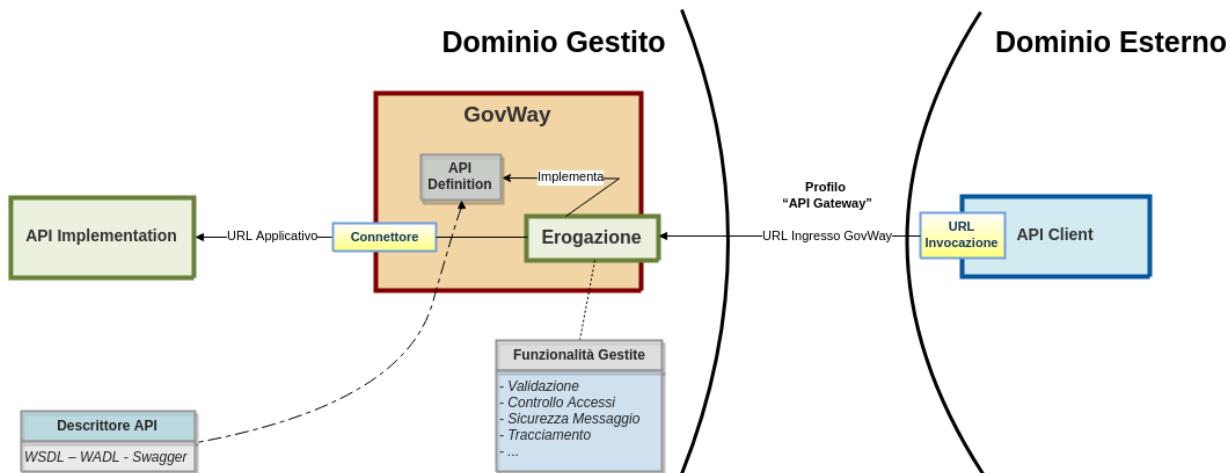


Figure2.4: Scenario di riferimento per l'erogazione

Compilare il form (Fig. 2.5) inserendo i seguenti dati:

- **API - Nome:** Selezionare dall'elenco il nome e la versione relativa alla API cui l'erogazione fa riferimento. Se la API selezionata è di tipo Soap, sarà necessario selezionare anche il Servizio che si vuole erogare.
- **Controllo degli Accessi:** In questa sezione è possibile stabilire l'eventuale controllo degli accessi all'erogazione:
 - **Pubblico:** non sono richieste credenziali per l'accesso.
 - **Autenticato:** l'accesso è ammesso solo previa verifica dei criteri di autenticazione e autorizzazione previsti in configurazione.
 Selezionando l'opzione **Autenticato**, dopo la creazione dell'erogazione, sarà necessario completare la configurazione del controllo degli accessi come descritto nella sezione *Autenticazione Trasporto*.
- **Connettore:** In questa sezione devono essere specificati i riferimenti al servizio, al fine di rendere possibile il corretto instradamento delle richieste inviate dai soggetti fruitori. Questo connettore riferisce il servizio del dominio interno che si sta erogando.

Le informazioni da fornire sono:

- **Utilizza Applicativo Server:** flag che consente di selezionare un applicativo di tipo «Server» invece di fornire tutte le informazioni relative al connettore. Per i dettagli consultare la sezione *Connettore*.
- **Endpoint:** la url per la consegna delle richieste al servizio.
- **Autenticazione Http:** credenziali da fornire nel caso in cui il servizio richieda autenticazione di tipo HTTP-BASIC.
- **Autenticazione Https:** credenziali da fornire nel caso in cui il servizio richieda autenticazione di tipo HTTPS.
- **Proxy:** nel caso in cui l'endpoint del servizio sia raggiungibile solo attraverso un proxy, possono essere indicati qui i relativi riferimenti.
- **Ridefinisci Tempi Risposta:** permette di ridefinire i tempi di risposta che sono stati configurati a livello generale, nell'ambito del controllo del traffico (vedi sezione *Tempi Risposta*)

Nota: Se l'API riferita dall'erogazione possiede un descrittore (WSDL, OpenAPI, ecc.) l'interfaccia propone come valore di default per il connettore l'endpoint del servizio.

Erogazioni > Aggiungi

Note: (*) Campi obbligatori

Informazioni Generali

API

Nome: TEST2 v2

Tipo: Soap

Servizio (Soap) *: Esitoldentificazione

Controllo degli Accessi

Accesso API: autenticato

Connettore

Utilizza Applicativo Server:

Endpoint *: http://10.114.87.21:8180/openspcoop/PD/SPCCentroAnagrafico /SPCComune/SPCEsitoldentificazione/Risultato

Autenticazione Http:

Autenticazione Token:

AutenticazioneHttps:

Proxy:

Ridefinisci Tempi Risposta:

SALVA

Figure2.5: Registrazione di una Erogazione

2.2.1 Completamento configurazione e indirizzamento del servizio

Dopo aver definito le API e registrato la relativa erogazione, come descritto nelle sezioni precedenti, si dispone della configurazione di un servizio erogato i cui riferimenti possono essere comunicati ai fruitori.

Per aggiungere ulteriori dettagli di configurazione, o semplicemente per conoscere il giusto endpoint cui il fruitore deve indirizzare le richieste, si procede dalla pagina di dettaglio dell'erogazione già creata. Il dettaglio dell'erogazione si raggiunge andando alla sezione del menu *Registro > Erogazioni*, cliccando sull'elemento visualizzato nell'elenco delle erogazioni presenti nel registro (Fig. 2.6).

Figure2.6: Elenco Erogazioni presenti nel registro

Per la ricerca dell'elemento nell'elenco delle erogazioni è possibile filtrare i dati visualizzati tramite la maschera di filtro che compare cliccando sulla voce *Erogazioni* nell'intestazione dell'elenco (Fig. 2.7).

Il dettaglio dell'erogazione mostra i dati principali e con le icone «matita» è possibile entrare sulle maschere di editing per effettuare delle modifiche. In corrispondenza del connettore è disponibile anche un pulsante che consente di verificare la raggiungibilità dell'indirizzo impostato. In corrispondenza della API riferita, è possibile accedere al relativo dettaglio aprendo un nuovo tab del browser (Fig. 2.8).

La pagina di dettaglio dell'erogazione visualizza i principali elementi di configurazione, che sono:

Erogazioni

Tipo API

Tag

API / Soggetto Erogatore

FILTRA **RIPULISCI**

Figure2.7: Filtro delle Erogazioni presenti nel registro

Esitoidentificazione v2

Nome	● Esitoidentificazione v2	
API	TEST2 v2 (Soap) <input type="button" value="TESTSUITE"/> <input type="button" value="TESTSUITE2"/>	
URL Invocazione	http://localhost:8080/govway/ENTE/Esitoidentificazione/v2	
Connettore	http://127.0.0.1:8080/TestService/echo	
Gestione CORS	<input checked="" type="checkbox"/> Abilitato	

CONFIGURA

Figure2.8: Dettaglio dell'erogazione

- **Nome:** nome dell'erogazione. Accanto al valore è presente l'icona a matita che consente di modificare tale valore. In assenza di configurazioni specifiche per risorsa/azione (sezione *Differenziare le configurazioni specifiche per risorsa/azione*) è presente anche un'icona che permette di disattivare/riattivare l'erogazione. Lo stato di attivazione dell'erogazione è segnalato tramite l'icona colorata presente accanto al nome.
- **API:** API cui fa riferimento l'erogazione con evidenza degli eventuali tags. È presente un'icona che apre in una nuova finestra l'interfaccia per la gestione della configurazione della specifica API.
- **URL Invocazione:** URL che deve utilizzare il mittente per accedere al servizio erogato tramite il gateway. Questo dato rappresenta la *URL* del servizio nel caso Soap o la *Base URL* nel caso Rest. Per la selezione dell'operazione da invocare si distinguono i seguenti casi:
 - *REST*: Indipendentemente che l'API sia stata configurata fornendo il relativo descrittore, WADL o OpenAPI, l'identificazione dell'operation sarà sempre effettuata in automatico dal contesto di invocazione. Non è quindi necessario fornire ulteriori indicazioni.
 - *SOAP*
 - * *API con WSDL*: l'operation viene automaticamente identificata dal contesto di invocazione grazie alle informazioni presenti nel descrittore.
 - * *API senza WSDL*: l'operation viene identificata inserendo il relativo identificativo nella URL di invocazione, <URL_Invocazione>/<Azione>

Sono disponibili ulteriori metodi per l'identificazione dell'operation nel caso SOAP, per i cui dettagli si rimanda alla sezione *Modalità di identificazione dell'azione*.

- **Connettore:** Endpoint del servizio acceduto dal gateway, cui verranno consegnate le richieste pervenute. È presente l'icona a matita per aggiornare il valore del connettore. È inoltre presente un'icona che consente di testare la raggiungibilità del servizio tramite il connettore fornito. Maggiori dettagli vengono forniti nella sezione *Connettore*.
- **Gestione CORS:** stato abilitazione della funzione CORS. L'icona a matita consente di modificare l'impostazione corrente come descritto nella sezione *Gestione CORS*.

Ulteriori elementi possono essere indicati per specificare il funzionamento dell'erogazione. Si tratta degli elementi di configurazione specifica, per i cui dettagli si rimanda alla sezione *Configurazione dell'API*.

2.2.2 Condivisione dei dati di integrazione

Le richieste di erogazione, pervenute a GovWay, vengono elaborate e, nel corso dell'operazione, vengono creati i riferimenti alle entità di configurazione presenti nel registro.

GovWay comunica i dati di contesto ricavati, ai sistemi interlocutori, ed in particolare:

- Tutti i dati dell'header di integrazione, relativi al messaggio di richiesta, vengono inviati all'applicativo destinatario (erogatore). I dati che compongono l'header di integrazione sono quelli descritti nelle tabelle presenti alla sezione *Header di Integrazione*.
- Un sottoinsieme dell'header di integrazione, relativo al messaggio di risposta, viene inviato al soggetto mittente (fruitore). I dati inviati (sempre in riferimento alle tabelle della *Header di Integrazione*) sono:
 - *GovWay-Message-ID*
 - *GovWay-Relates-To*
 - *GovWay-Conversation-ID*
 - *GovWay-Transaction-ID*

2.2.3 Errori Generati dal Gateway

La gestione dei casi di errore, nelle comunicazioni mediate da un Gateway, deve tener conto di ulteriori casi di errore che possono presentarsi rispetto al dialogo diretto tra gli applicativi. Oltre agli errori già previsti nelle interfacce dell'API, gli applicativi client possono pertanto ricevere due tipi di errori generati direttamente da GovWay:

- *Errori Client*: identificabili da un codice http 4xx su API REST o da un fault code “Client” su API SOAP. Indicano che GovWay ha rilevato problemi nella richiesta effettuata dal client (es. errore autenticazione, autorizzazione, validazione contenuti...).
- *Errori Server*: identificabili dai codici http 502, 503 e 504 per le API REST o da un fault code “Server” generato dal Gateway e restituito con codice http 500 per le API SOAP.

Per ciascun errore GovWay riporta le seguenti informazioni:

- Un codice http su API REST o un fault code su API SOAP come descritto in precedenza.
- Un codice di errore, indicato nell'header http “GovWay-Transaction-ErrorType”, che riporta l'errore rilevato dal gateway (es. AuthenticationRequired, TokenExpired, InvalidRequestContent...).
- Un identificativo di transazione, indicato nell'header http “GovWay-Transaction-ID”, che identifica la transazione in errore, utile principalmente per indagini diagnostiche.
- Un payload http, contenente maggiori dettagli sull'errore, opportunamente codificato per API REST ([REST Problem Details - RFC 7807](#)) o SOAP ([SOAP Fault](#)).

Maggiori dettagli, sulla gestione degli errori, sono disponibili nella sezione [Errori generati da GovWay](#).

2.3 Registrazione della fruizione

Nel processo di fruizione sono coinvolti i client (o applicativi) interni al dominio che richiedono, tramite accesso sul gateway, un servizio erogato da un soggetto di un dominio esterno.

In Fig. 2.9 è illustrato graficamente il caso della fruizione.

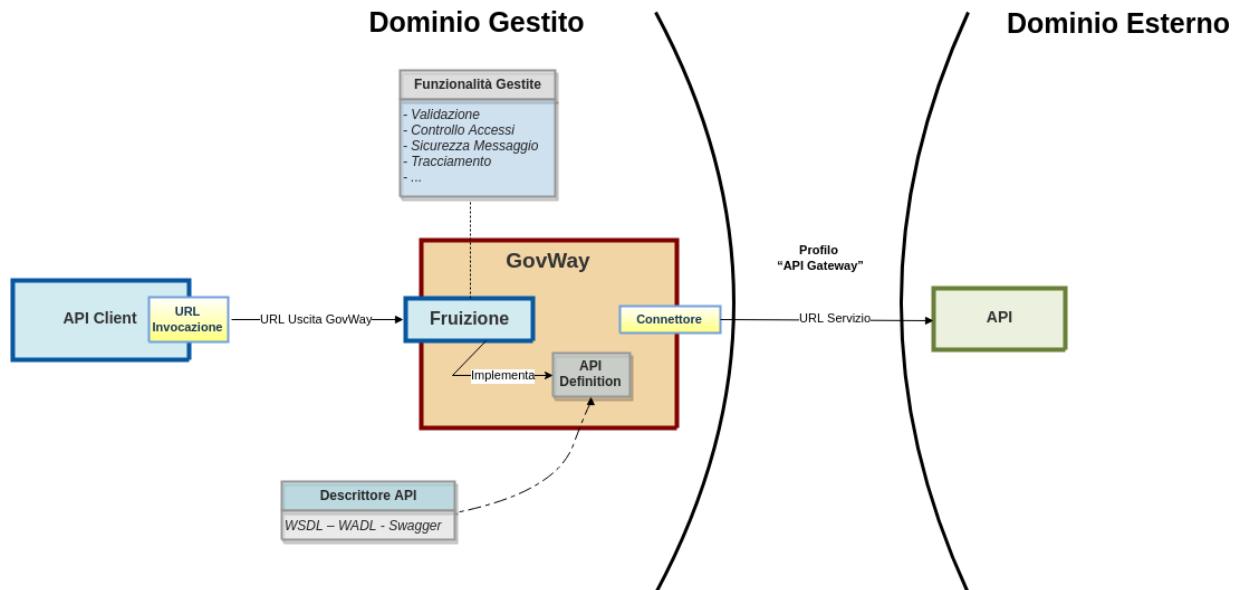


Figure2.9: Scenario di riferimento per la fruizione

Analogamente a quanto descritto per le erogazioni, è possibile procedere con la configurazione delle fruizioni accedendo alla sezione di menu *Registro > Fruizioni*.

La configurazione delle fruizioni presenta maschere della GovWayConsole del tutto analoghe al caso dell'erogazione. È quindi possibile seguire il processo di configurazione attuando i medesimi passi, illustrati per le erogazioni, calandole sul contesto delle fruizioni.

L'unica differenza, rispetto al processo di configurazione delle erogazioni, è rappresentata dalla presenza del campo *Soggetto Erogatore*, da selezionare come soggetto che eroga il servizio (Fig. 2.10).

Note: (*) Campi obbligatori

Informazioni Generali

API

Nome: API_SOAP_1 v1

Tipo: Soap

Servizio: EsitoIdentificazione

Soggetto Erogatore

Nome: Piffero

Controllo degli Accessi

Accesso API: autenticato

Connettore

Endpoint: http://10.114.87.21:8180/openscoop/PD/SPCCentroAnagrafico /SPCComune/SPCEsitoIdentificazione/Risultato

Autenticazione Http:

Autenticazione Https:

Proxy:

Ridefinisci Tempi Risposta:

SALVA

Figure2.10: Registrazione di una Fruizione

Nota: Benché non vi siano differenze nelle modalità di configurazione del *Connettore*, nel caso della fruizione questi consiste nei dati di puntamento al servizio erogato sul dominio esterno.

2.3.1 Condivisione dei dati di integrazione

Le richieste di fruizione, pervenute a GovWay, vengono elaborate e, nel corso dell'operazione, vengono creati i riferimenti alle entità di configurazione presenti nel registro.

GovWay comunica i dati di contesto ricavati, ai sistemi interlocutori:

- *GovWay-Message-ID*
- *GovWay-Relates-To*
- *GovWay-Conversation-ID*
- *GovWay-Transaction-ID*

Per ulteriori dettagli si consiglia di consultare la sezione *Header di Integrazione*.

2.3.2 Errori Generati dal Gateway

Analogamente a quanto descritto per le erogazioni, la gestione dei casi di errore nelle comunicazioni mediate da un Gateway devono tenere conto di ulteriori situazioni che possono presentarsi rispetto alla situazione di dialogo diretto tra gli applicativi.

La gestione degli errori viene descritta approfonditamente nella sezione *Errori generati da GovWay*.

2.4 Versionamento delle API

Come descritto nelle precedenti sezioni, ogni API possiede una versione. È possibile registrare una nuova versione dell'API cliccando sul pulsante “Nuova Versione” presente nel dettaglio di una API (Fig. 2.11).

La maschera di creazione della nuova versione non permette né di modificare il nome dell'API né di scendere di versione. Vengono ereditati dall'API precedente le altre caratteristiche quali il tipo di API tra SOAP e REST, i tags, la descrizione, il soggetto referente etc.

Se l'opzione “Ridefinisci Interfaccia” è abilitata, viene richiesta una nuova specifica dell'interfaccia dell'API. Terminando la creazione della nuova API verranno creati automaticamente i servizi e le azioni su SOAP o le risorse su REST definiti nella nuova interfaccia (Fig. 2.12).

In alternativa, se l'opzione “Ridefinisci Interfaccia” viene disabilitata, non viene richiesta una nuova specifica di interfaccia e la nuova versione dell'API conterrà la medesima specifica della precedente versione con i medesimi servizi e azioni su SOAP o risorse su REST (Fig. 2.13).

Una volta creata una nuova versione dell'API, è possibile effettuare l'upgrade verso la nuova versione direttamente nell'erogazione e/o nella fruizione che implementa l'API. Infatti se esiste più di una versione di una medesima API è possibile modificarne la versione implementata nell'erogazione o nella fruizione tramite il bottone “modifica” evidenziato nella figura Fig. 2.14.

Accedendo alla modifica è possibile scegliere la versione implementata dell'API, tra le versioni disponibili, come mostrato nella figura Fig. 2.15.

La modifica della versione dell'API implementata dall'erogazione, comporta automaticamente anche la modifica della versione dell'erogazione stessa. Questo si riflette nell'url di invocazione che viene automaticamente aggiornata rispetto alla nuova versione come evidenziato nella figura Fig. 2.16.

Nota: Se viene scelta una versione dell'API per la quale esiste già una medesima versione dell'erogazione, il cambio di versione dell'API non si rifletterà sulla versione dell'erogazione e sulla url di invocazione ma solamente sui messaggi

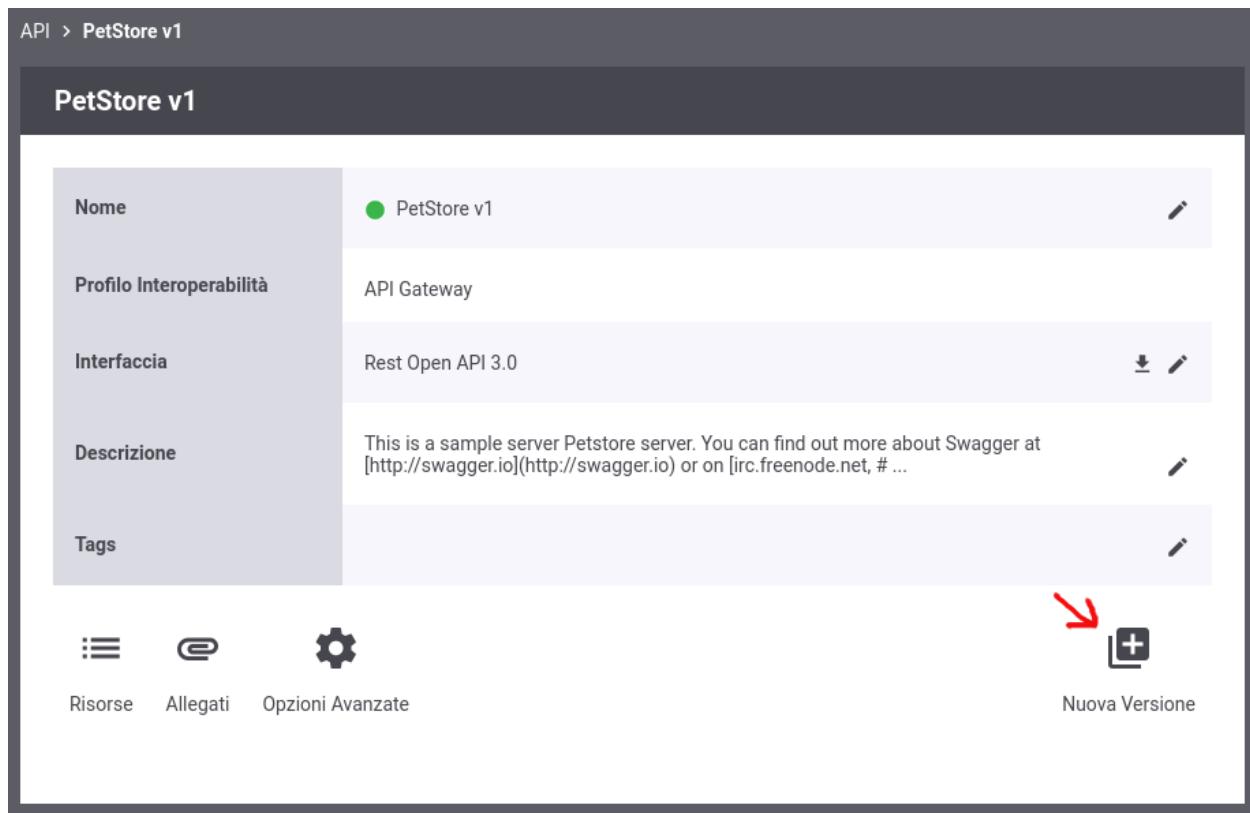


Figure2.11: Nuova Versione di una API

The screenshot shows the 'API > Aggiungi' (Add API) screen. The 'API' section contains the following fields:

- Nome:** PetStore
- Descrizione:** This is a sample server Petstore server.
- Tags:** (empty input field)
- Versione:** 2

The 'Specifiche delle interfacce' (Interface Specifications) section contains:

- Ridefinisci Interfaccia:**
- Formato Specifica:** Open API 3.0
- Open API 3.0:** Choose File | No file chosen

A large 'SALVA' (Save) button is located at the bottom of the form.

Figure2.12: Nuova Versione di una API tramite ridefinizione dell'intervaccia

API > Aggiungi

API

Nome	PetStore
Descrizione	This is a sample server Petstore server.
Tags	
Versione	2

Specifiche delle interfacce

Ridefinisci Interfaccia

SALVA

Figure2.13: Nuova API che eredita la specifica di interfaccia dalla versione precedente

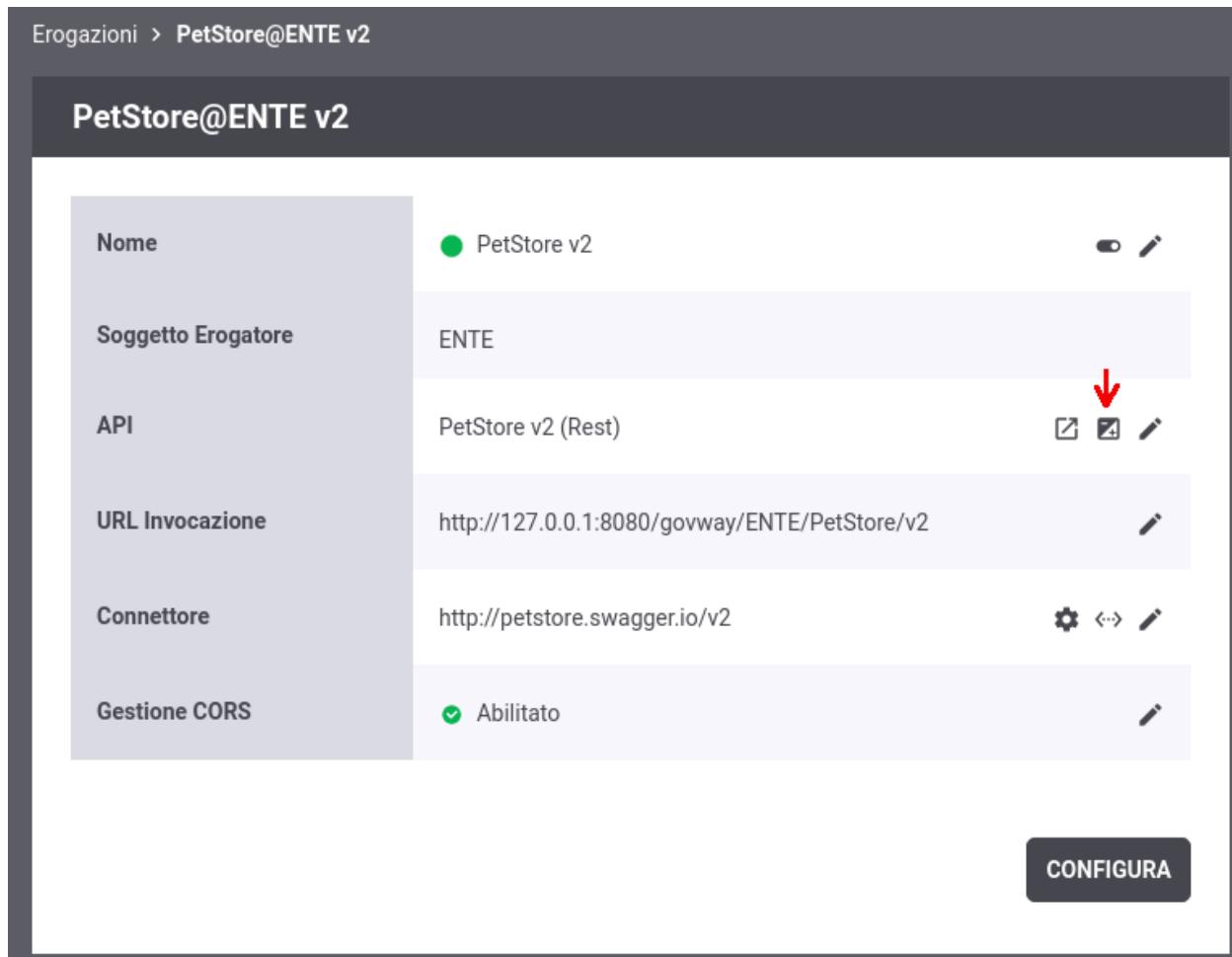


Figure2.14: Upgrade di versione dell'API implementata in una erogazione

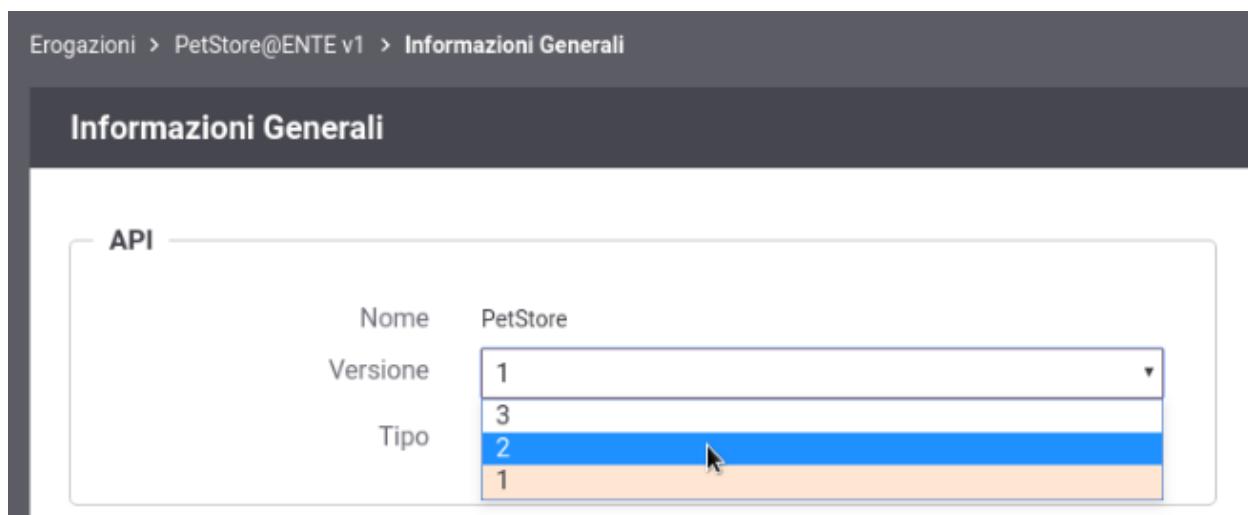


Figure2.15: Scelta della versione dell'API implementata in una erogazione

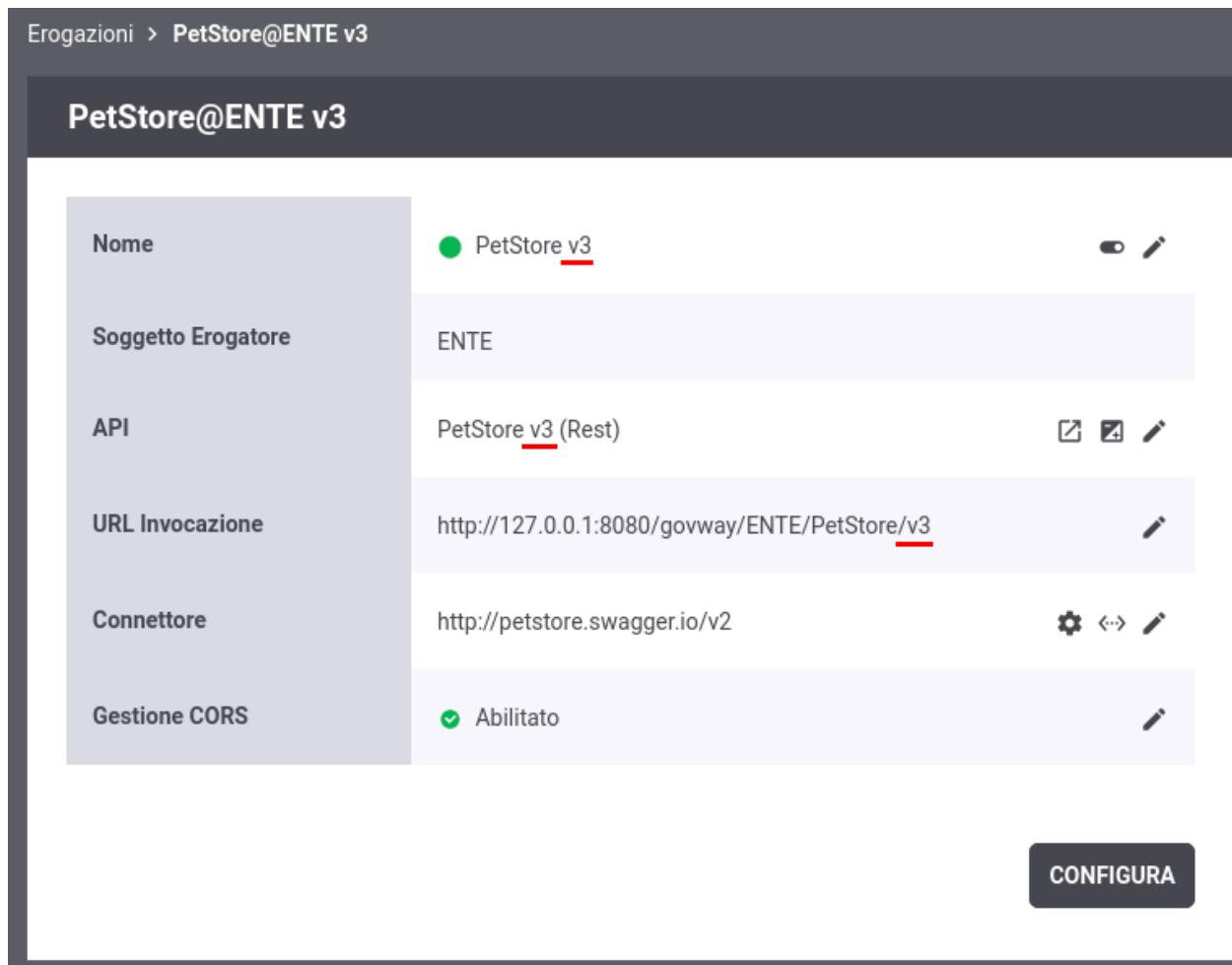


Figure2.16: Modifica della versione si riflette sia sull'erogazione che sulla url di invocazione

scambiati e sulle azioni (soap) o risorse (rest) che l'erogazione espone.

Per maggiori dettagli sul versionamento differente tra erogazione/fruizione ed API e di conseguenza su come questo si riflette nella url di invocazione si rimanda alla sezione [Versionamento delle API e delle Erogazioni/Fruizioni](#)

2.5 Configurazione dell'API

I passi di configurazione fin qui descritti, per la registrazione di erogazioni e fruizioni, consentono di ottenere uno stato delle entità del registro pronto all'utilizzo in numerose situazioni.

Cliccando sulla voce *Erogazioni* o *Fruizioni* nell'intestazione dell'elenco è possibile consultarne i dettagli selezionando l'API attivata di interesse.

La pagina di dettaglio consente di accedere ai principali elementi di configurazione (Fig. 2.17):

- *Nome*: in assenza di configurazioni specifiche per risorsa/azione (sezione [Differenziare le configurazioni specifiche per risorsa/azione](#)), accanto al nome dell'erogazione o della fruizione è presente un'icona che permette di disattivare/riattivare l'API come descritto nella sezione [Sospensione API](#).
- *URL Invocazione*: se la console viene utilizzata in modalità avanzata (sezione [Modalità Avanzata](#)), accedendo alla modifica della URL di Invocazione è possibile configurare la modalità di identificazione dell'azione come descritto nella sezione [Modalità di identificazione dell'azione](#).
- *Connettore*: endpoint del servizio acceduto dal gateway, cui verranno consegnate le richieste pervenute. In questa è presente sia l'icona a matita per aggiornare il valore del connettore che un'icona che consente di testare la raggiungibilità del servizio tramite il connettore fornito. Maggiori dettagli vengono forniti nella sezione [Connettore](#).
- *Gestione CORS*: stato abilitazione della funzione CORS. L'icona a matita consente di modificare l'impostazione corrente come descritto nella sezione [Gestione CORS](#).

Tramite il pulsante *Configura* è inoltre possibile aggiungere ulteriori elementi di configurazione attraverso le ulteriori funzionalità messe a disposizione da GovWay (Fig. 2.18).

Le voci di configurazione che possono essere accedute sono:

- [Controllo degli Accessi](#)
- [Rate Limiting](#)
- [Validazione dei messaggi](#)
- [Caching Risposta](#)
- [Sicurezza a livello del messaggio](#)
- [MTOM](#)
- [Trasformazioni](#)
- [Tracciamento](#)
- [Registrazione Messaggi](#)
- [Proprietà](#)
- [Opzioni Avanzate per Erogazioni/Fruizioni](#)

Accanto a ciascuna delle voci in elenco è presente un'icona che in base al colore assume i seguenti significati:

- **Grigio**: funzionalità non attiva
- **Rosso**: funzionalità attivata ma configurata in maniera incompleta o errata, quindi non funzionante

Erogazioni > **api-config@ENTE v1**

api-config@ENTE v1

Nome	api-config v1	
Soggetto Erogatore	ENTE	
API	api-config v1 (Rest) API-GovWay	
URL Invocazione	http://127.0.0.1:8080/govway/ENTE/api-config/v1	
Connettore	http://127.0.0.1:8080/govwayAPIConfig/	
Gestione CORS	Abilitato	

CONFIGURA

Figure2.17: Dettaglio di una erogazione

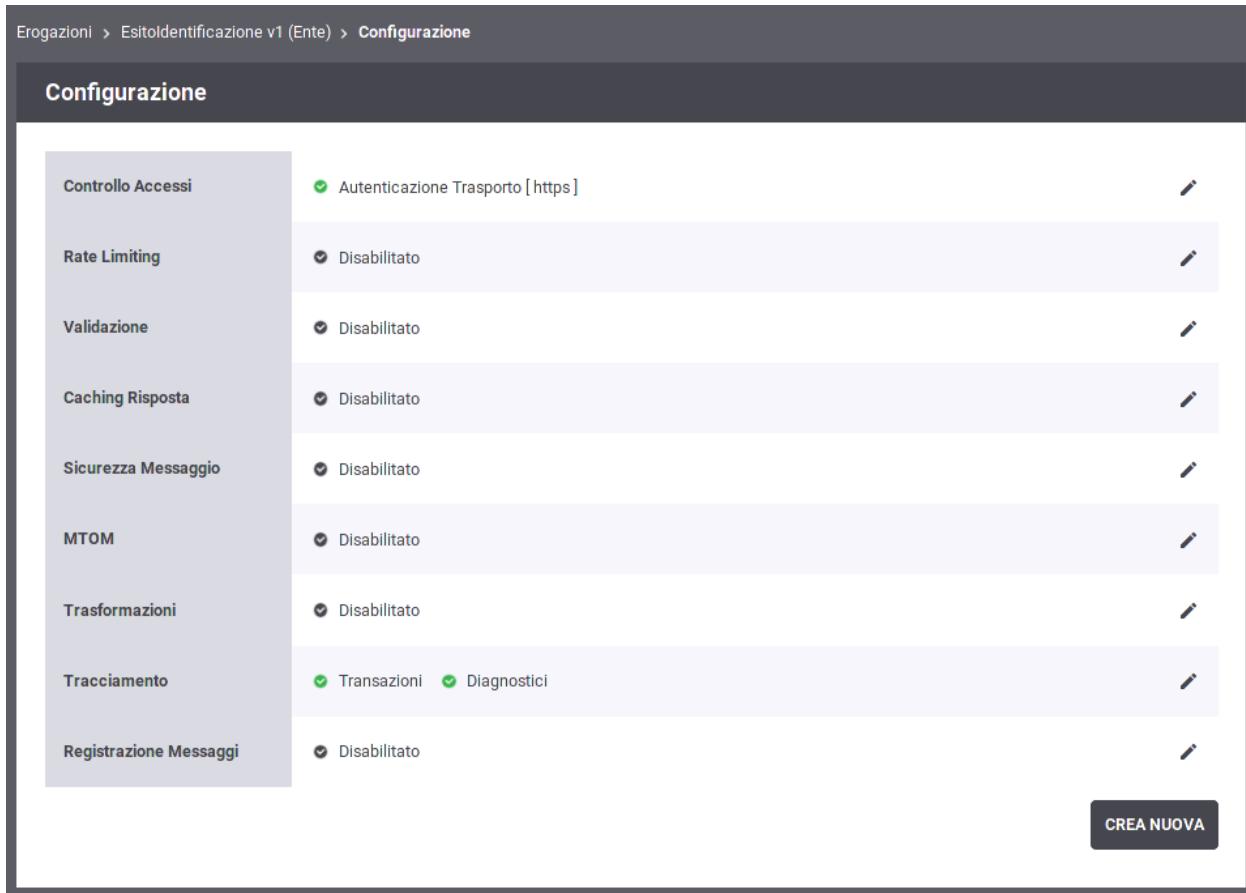


Figure2.18: Configurazione di una erogazione

- **Giallo:** funzionalità attivata in modalità opzionale o «non bloccante» e quindi in sola notifica
- **Verde:** funzionalità attiva

Le funzionalità specifiche possono essere configurate in maniera differenziata per gruppi di risorse/azioni relative alla API erogata/fruita. Una nuova configurazione specifica può essere creata tramite il pulsante *Crea Nuova*. Il passaggio tra una configurazione e l'altra sarà possibile tramite i tab che risulteranno visibili nell'interfaccia. Questa funzionalità è descritta in dettaglio nella sezione *Differenziare le configurazioni specifiche per risorsa/azione*.

Le sezioni successive descrivono in dettaglio le configurazioni sopraelencate e i relativi contesti di utilizzo. Tranne dove esplicitamente dichiarato, gli schemi di configurazione descritti in seguito possono essere attuati sia sulle erogazioni che sulle fruizioni.

2.6 Sospensione API

La console consente di disabilitare temporaneamente una API attiva sul gateway. Successive invocazioni destinate all'API verranno rifiutate generando un codice di errore *APISuspended*.

È possibile sospendere una API cliccando sull'icona toggle presente nella riga *Nome* del dettaglio di una erogazione o fruizione (Fig. 2.19).

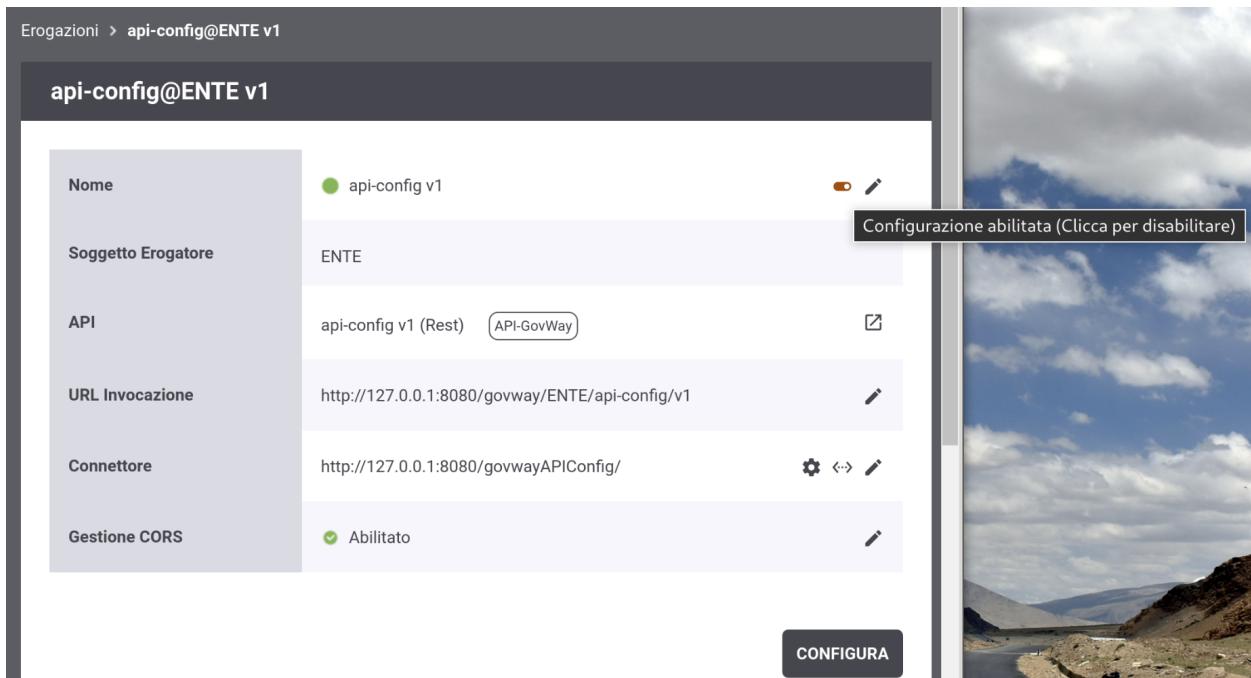


Figure 2.19: Sospensione di una API

Per poter procedere con la sospensione dell'API l'utente deve confermare l'operazione richiesta come evidenziato nella figura Fig. 2.20.

Quando una API viene sospesa, il suo nome viene affiancato da uno stato rosso che ne evidenzia l'inutilizzo temporaneo. Per abilitarla nuovamente si deve procedere con gli stessi passi effettuati per sospenderla (Fig. 2.21).

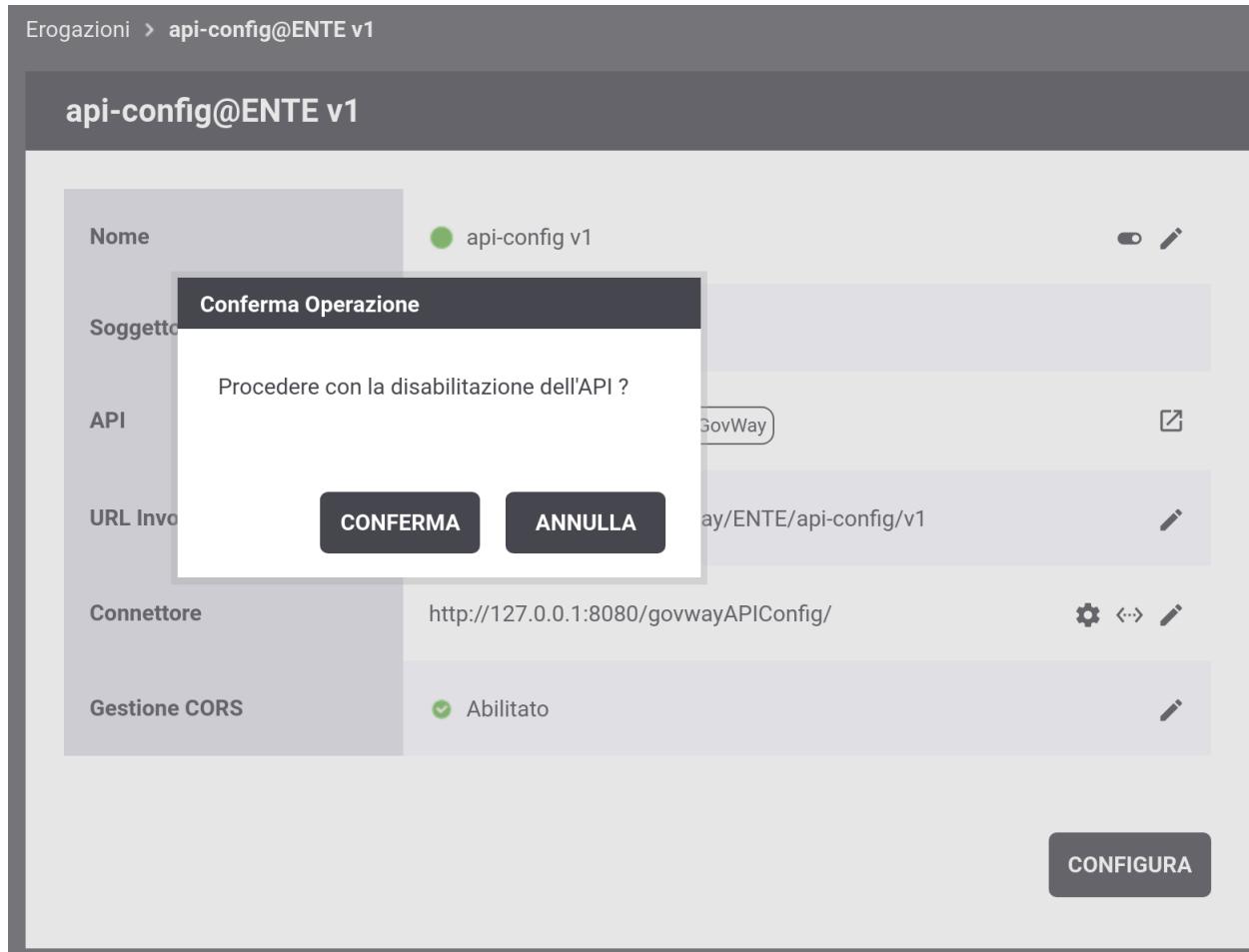


Figure2.20: Conferma dell'operazione di sospensione di una API

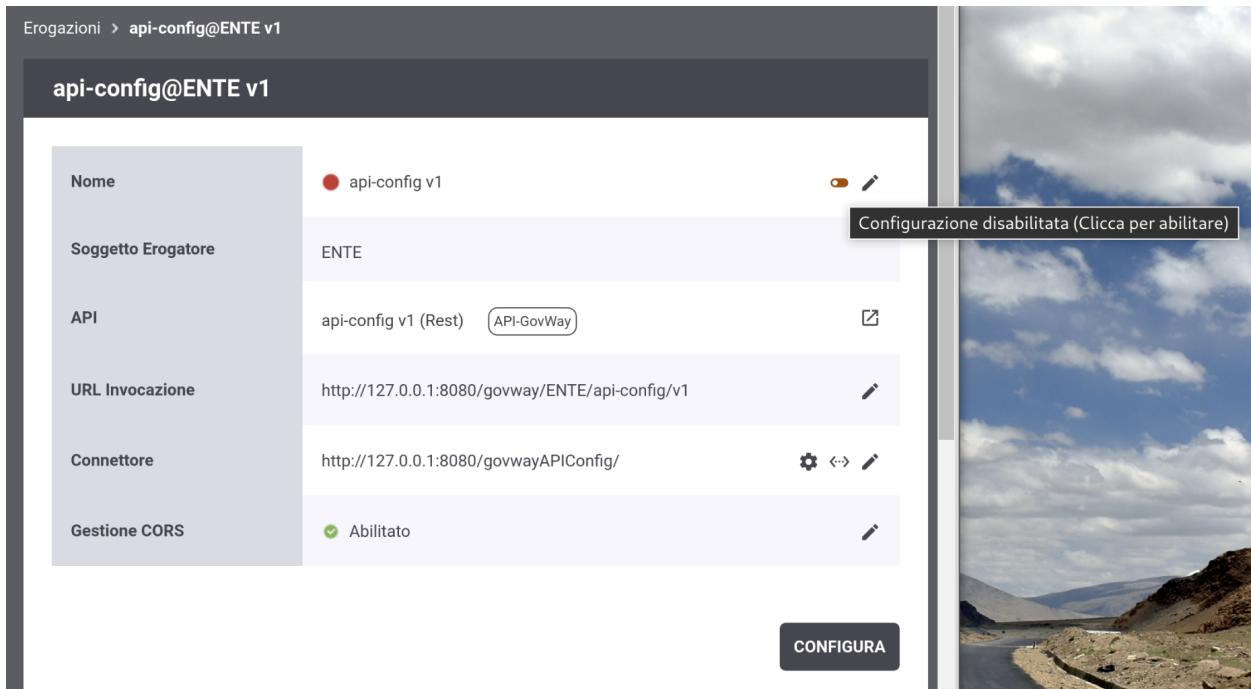


Figure 2.21: Attivazione di una API

2.7 Connettore

È possibile modificare le impostazioni del connettore (ad esempio per modificare l'endpoint o aggiungere il proxy) seguendo il collegamento presente nella riga *Connettore* del dettaglio di una erogazione o fruizione. I campi del form sono uguali a quelli già descritti per la fase di creazione dell'erogazione (sezione [Registrazione dell'erogazione](#)). Ulteriori dettagli di configurazione e tipi di connettore diversi da HTTP e HTTPS sono descritti nella sezione [Connettori](#).

La sezione [Verifica Connettività Connnettore](#) descrive uno strumento per verificare la raggiungibilità dell'indirizzo impostato.

La sezione [Applicativi Server](#) descrive invece come censire un'applicazione di backend in modo da poterla riferire su diversi connettori relativi ad erogazioni di API.

Le sezioni successive descrivono le funzionalità inerenti l'utilizzo di endpoint multipli allo scopo di bilanciare il carico o differenziarlo rispetto a variabili presenti nella richiesta, sempre relativamente ad erogazioni di API.

Infine la sezione [ProxyPassReverse per Header HTTP Location e Set-Cookie](#) descrive la funzionalità di riscrittura delle url negli header HTTP della risposta.

Nota: Le funzionalità relative ad un applicativo “Server” (sezione [Applicativi Server](#)) e ai connettori multipli ([Load Balancer](#) e [Consegna Condizionale](#)) sono applicabili solamente per le erogazioni di API.

2.7.1 Verifica Connattività Connettore

I contesti in cui l’interfaccia visualizza il valore di un connettore comprendono anche uno strumento per verificare la raggiungibilità dell’indirizzo impostato (Fig. 2.22).

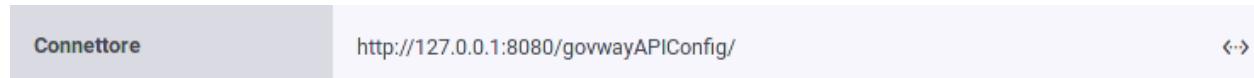


Figure2.22: Pulsante per la verifica del connettore

Dopo aver premuto il pulsante si accede ad una schermata che riepiloga le proprietà del connettore e comprende il pulsante *Verifica* per procedere con la verifica (Fig. 2.23).

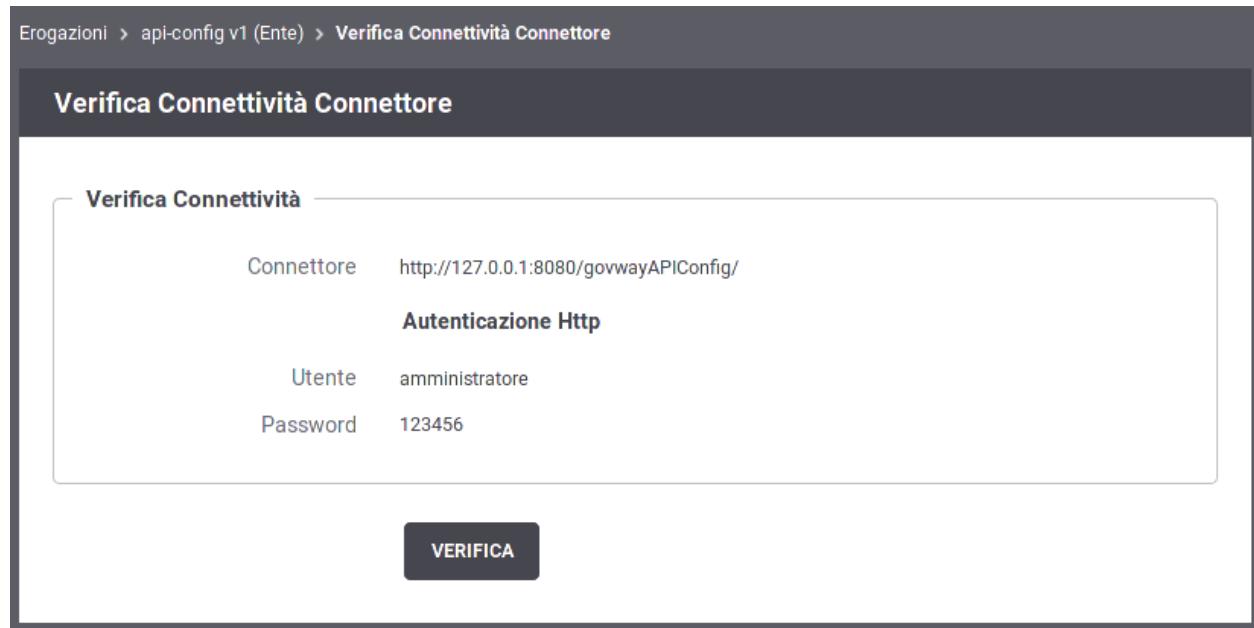


Figure2.23: Verifica del connettore

Dopo aver premuto il pulsante *Verifica* viene presentato l’esito della verifica di raggiungibilità (Fig. 2.24).

In presenza di un endpoint https, è possibile effettuare il download dei certificati ritornati dal server cliccando sul link “Download Certificati Server”. Il formato del file scaricato è un PEM contenente tutti i certificati ritornati dal server.

2.7.2 Applicativi Server

Un applicativo di tipo “Server” consente di censire un’applicazione di backend alla quale associare quelle informazioni tipicamente indicate finora nella sezione “Connettore” dell’erogazione della API (endpoint, credenziali, …). In una erogazione è così possibile riferire un applicativo server già registrato come modalità alternativa a quella di indicare esplicitamente tutte le informazioni richieste.

Per registrare l’applicativo posizionarsi nella sezione *Registro > Applicativi*, quindi premere il pulsante *Aggiungi*. Compilare il form come segue (Fig. 2.26):

- *Profilo Interoperabilità*: Opzione visibile solo nel caso in cui non sia stata effettuata la relativa scelta sul menu della testata.

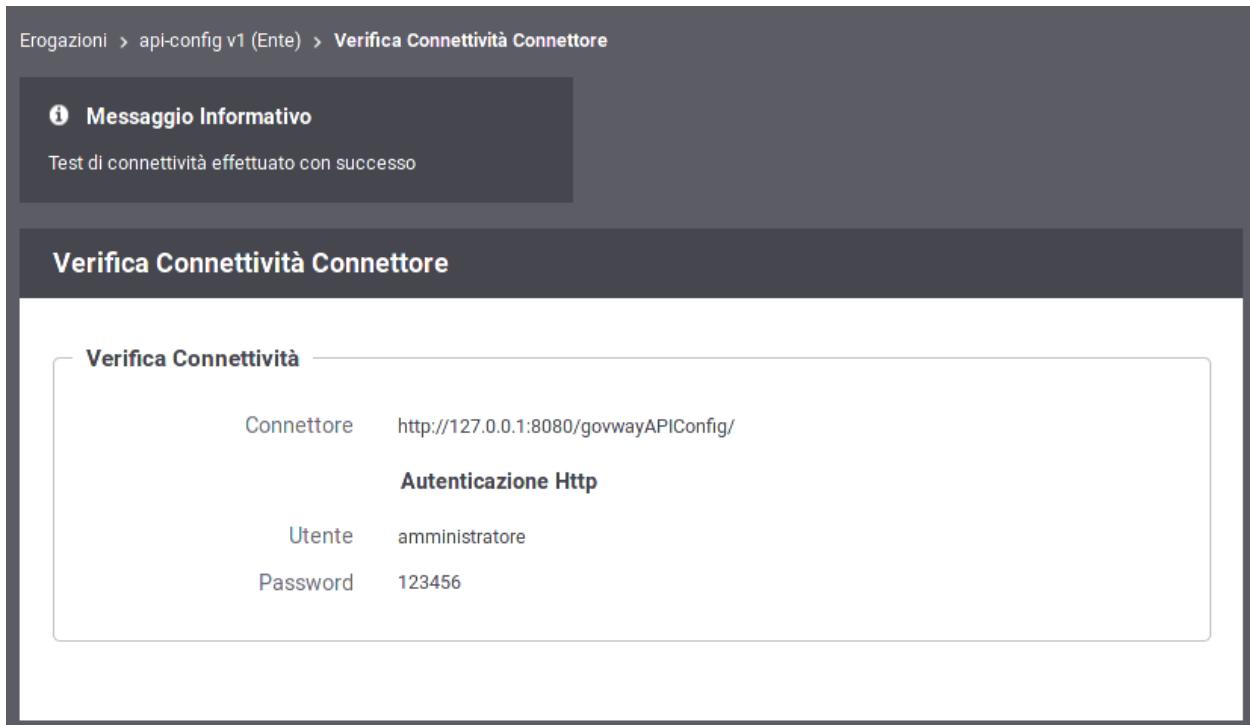


Figure2.24: Esito Verifica del connettore

- *Nome*: Assegnare un nome all'applicativo. È necessario che il nome indicato risulti univoco rispetto ai nomi già presenti per la modalità operativa selezionata (in questo caso API Gateway).
- *Tipo*: Utilizzare il tipo “Server” per censire un'applicativo di backend.
- *Connettore*: Tramite la sezione *Connettore* è possibile fornire quelle informazioni tipicamente indicate sinora nella sezione “Connettore” dell'erogazione di una API.

Dopo averlo creato, l'applicativo è associabile ad una Erogazione accedendo alla sezione «Connettore» come evidenziato nella figura Fig. 2.27.

2.7.3 Load Balancer

Per le erogazioni di API è possibile definire connettori multipli con la finalità di attuare su di essi il bilanciamento delle richieste pervenute.

I contesti in cui l'interfaccia visualizza il valore di un connettore comprendono anche uno strumento per abilitare la gestione dei connettori multipli (Fig. 2.28).

Dopo aver premuto tale pulsante si accede ad una schermata che consente di abilitare e configurare tale funzionalità. In questa sezione, in particolare, viene descritta la funzionalità *Load Balancer* che rappresenta la voce di default una volta abilitato lo stato relativo alla configurazione dei connettori multipli (Fig. 2.29). Per maggiori dettagli si rimanda alla sezione *Configurazione del Bilanciamento del Carico*.

Una volta attivata la funzione di Load Balancer, nei contesti in cui l'interfaccia visualizza l'endpoint di un connettore, viene adesso invece evidenziata la presenza di tale funzionalità. In tale contesto è possibile definire i nuovi connettori accedendo all'elenco dei connettori registrati tramite il nuovo pulsante dedicato (Fig. 2.30) come descritto nella sezione *Elenco dei Connettori Bilanciati*.

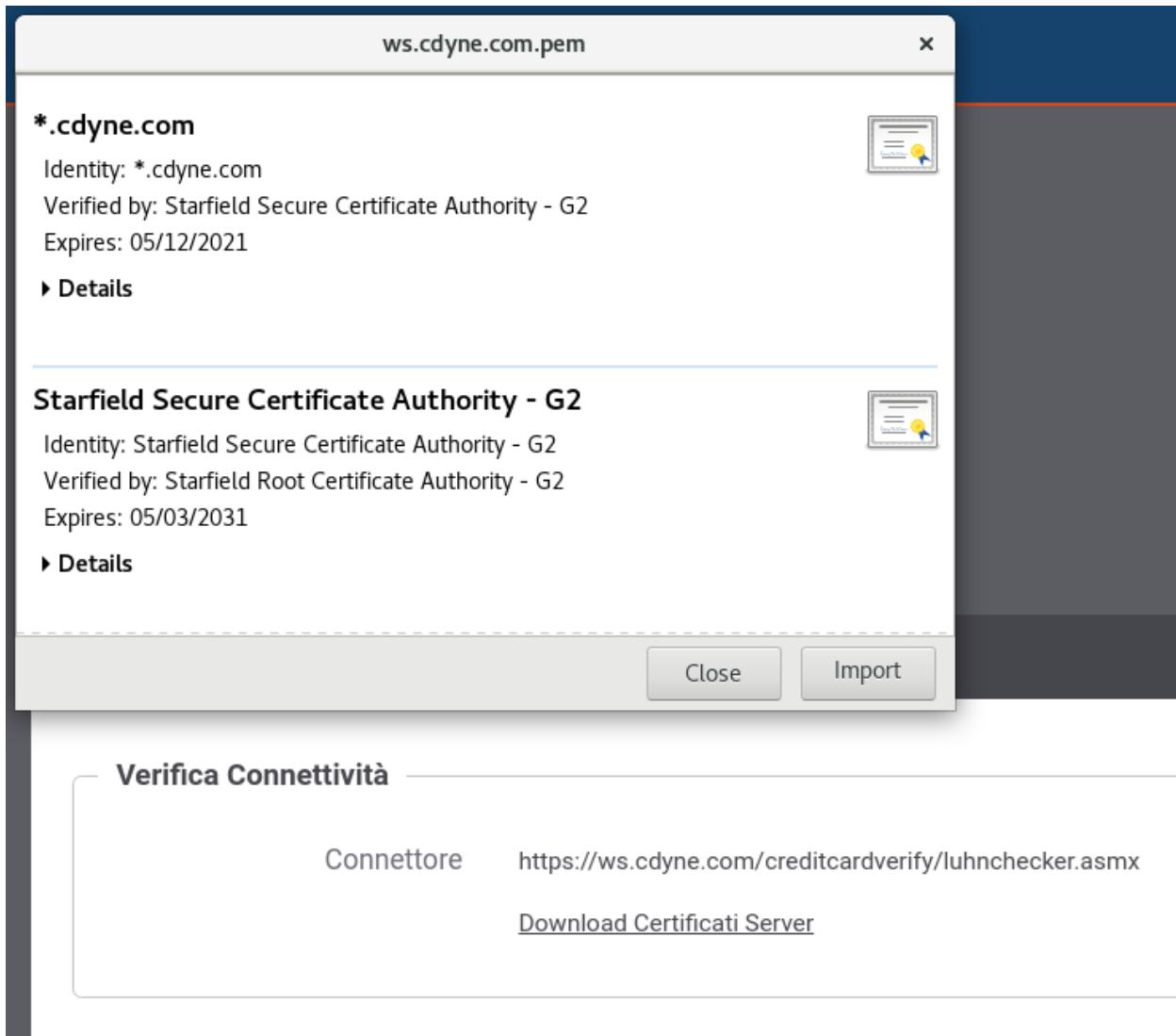


Figure2.25: Download Certificati Server

Applicativi > Aggiungi

Note: (*) Campi obbligatori

Applicativo

Nome *

Tipo

Connettore

Endpoint *

Autenticazione Http

Autenticazione Token

AutenticazioneHttps

Proxy

Ridefinisci Tempi Risposta

SALVA

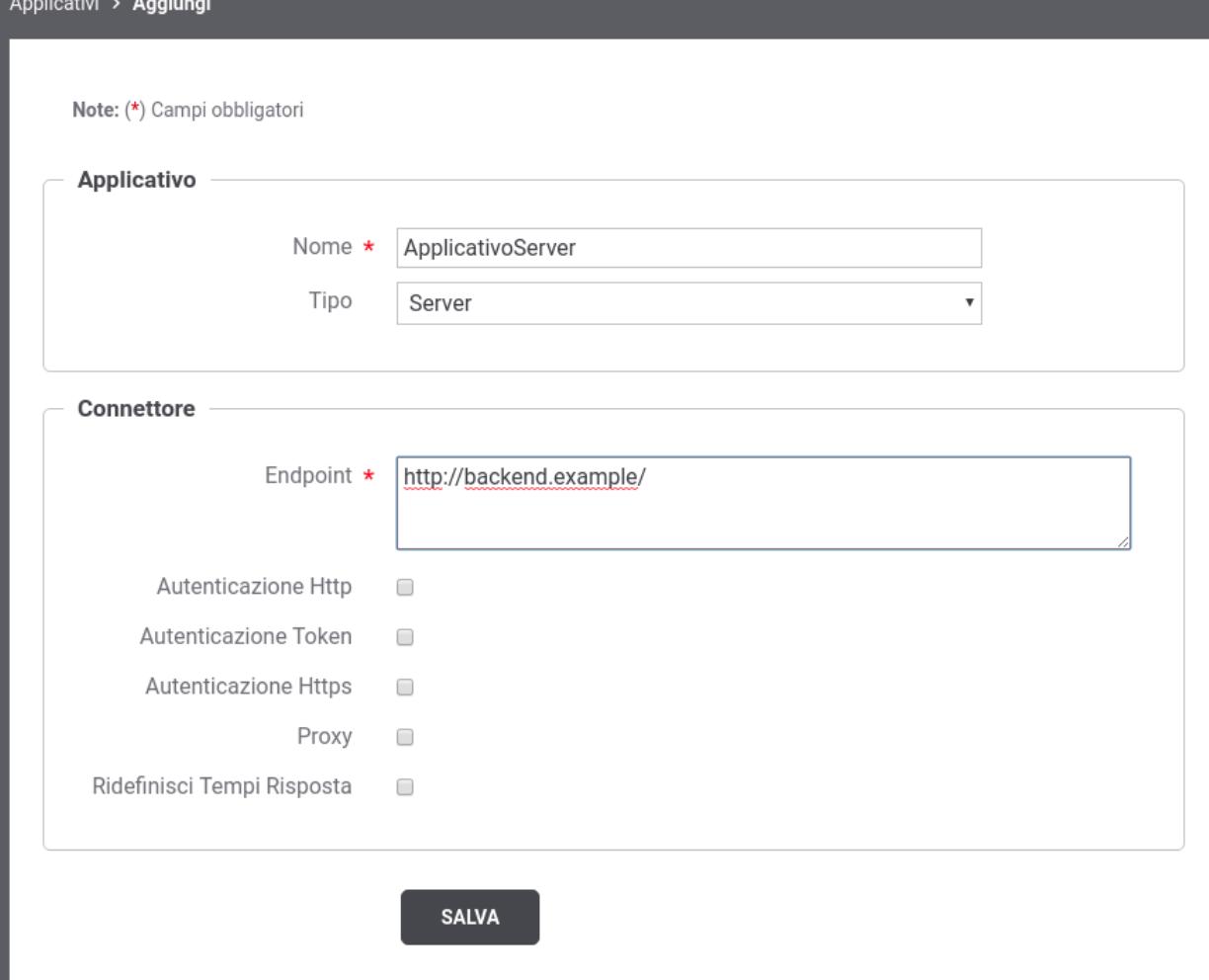


Figure2.26: Creazione di un Applicativo Server

Erogazioni > TEST v1 (ENTE) > **Connettore**

Connettore

Connettore

Utilizza Applicativo Server

Applicativo

SALVA

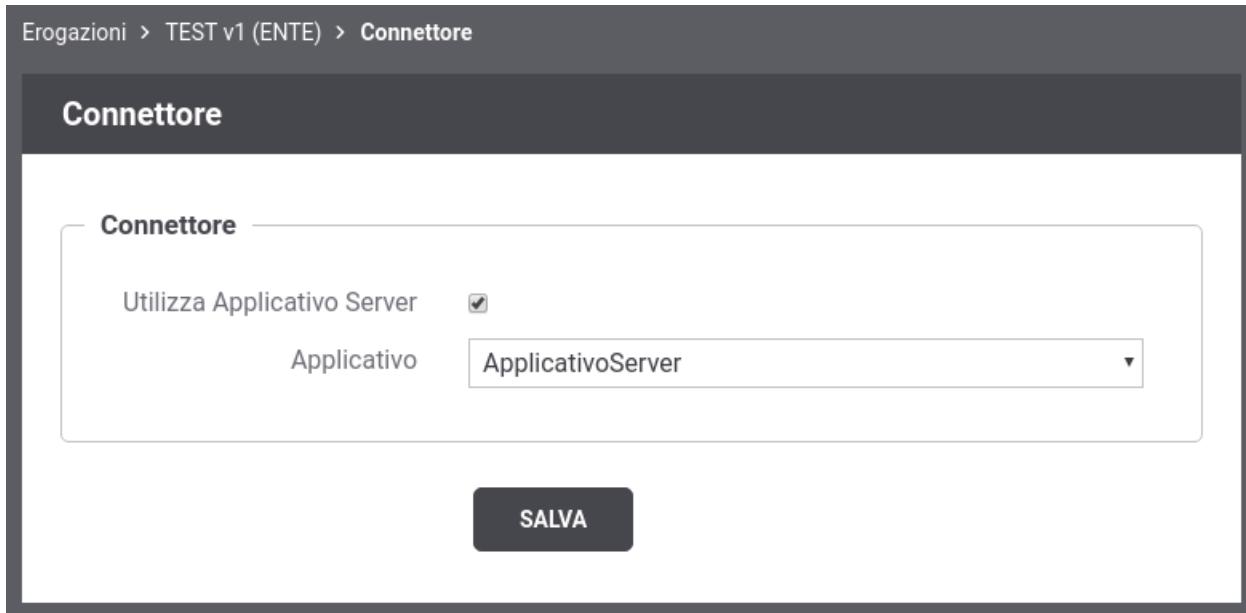


Figure2.27: Associazione di un Applicativo Server ad una Erogazione

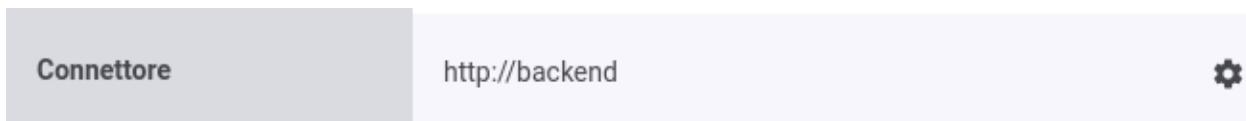


Figure2.28: Pulsante per la configurazione dei connettori multipli

Erogazioni > TEST v1 (ENTE) > **Configurazione Connettori Multipli**

Configurazione Connettori Multipli

Configurazione Connettori Multipli

Stato

Modalità Consegna

Strategia ⓘ

Sessione Sticky Tutte le richieste che presentano lo stesso id di sessione vengono servite tramite lo stesso connettore ⓘ

Health Check Un utilizzo di un connettore che provoca un errore di connettività comporta la sua esclusione dal pool ⓘ

Consegna Condizionale Solo i connettori che corrispondono alla condizione indicata concorrono per la consegna

SALVA

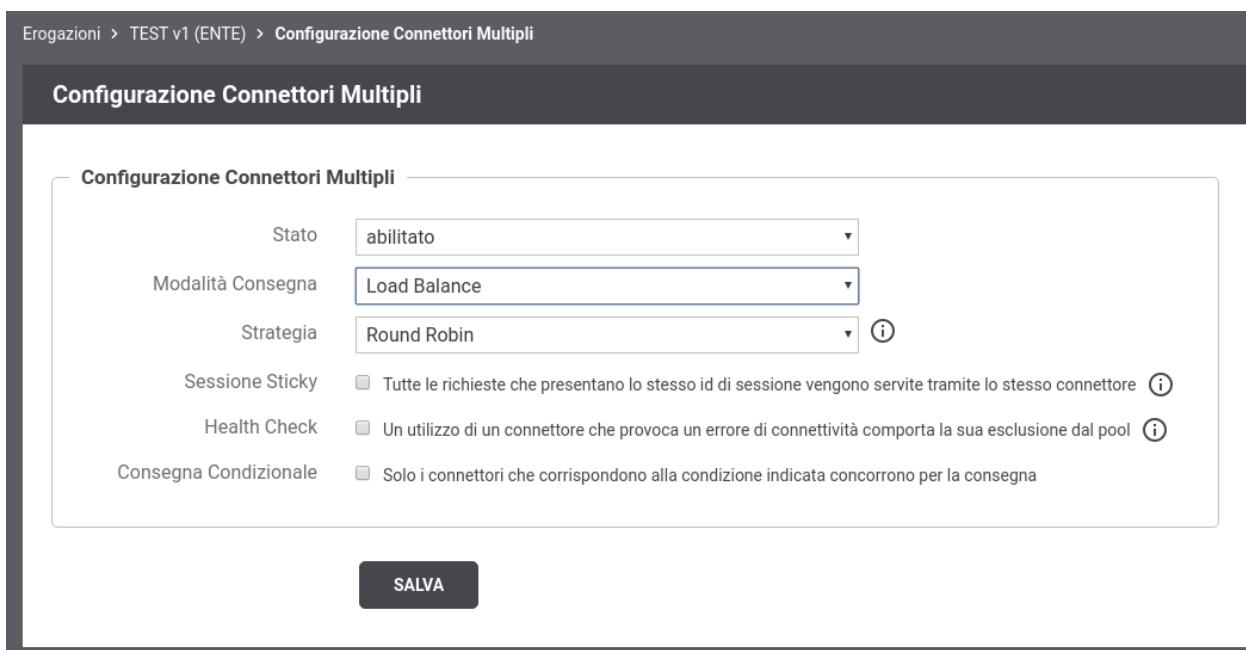


Figure2.29: Load Balancer



Figure2.30: Pulsante per accedere all'elenco dei connettori

Configurazione del Bilanciamento del Carico

Per abilitare la funzionalità di Load Balancer accedere alla sezione di dettaglio di una erogazione di API e cliccare sul pulsante di configurazione dei connettori multipli (Fig. 2.31).

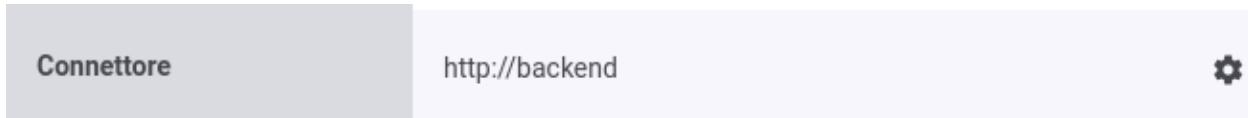


Figure2.31: Pulsante per la configurazione dei connettori multipli

Dopo aver premuto il pulsante si accede ad una schermata che consente di abilitare una funzionalità relativa ai connettori multipli. In questa sezione, in particolare, viene descritta la funzionalità *Load Balance* che rappresenta la voce di default una volta abilitato lo stato relativo alla configurazione dei connettori multipli (Fig. 2.32).

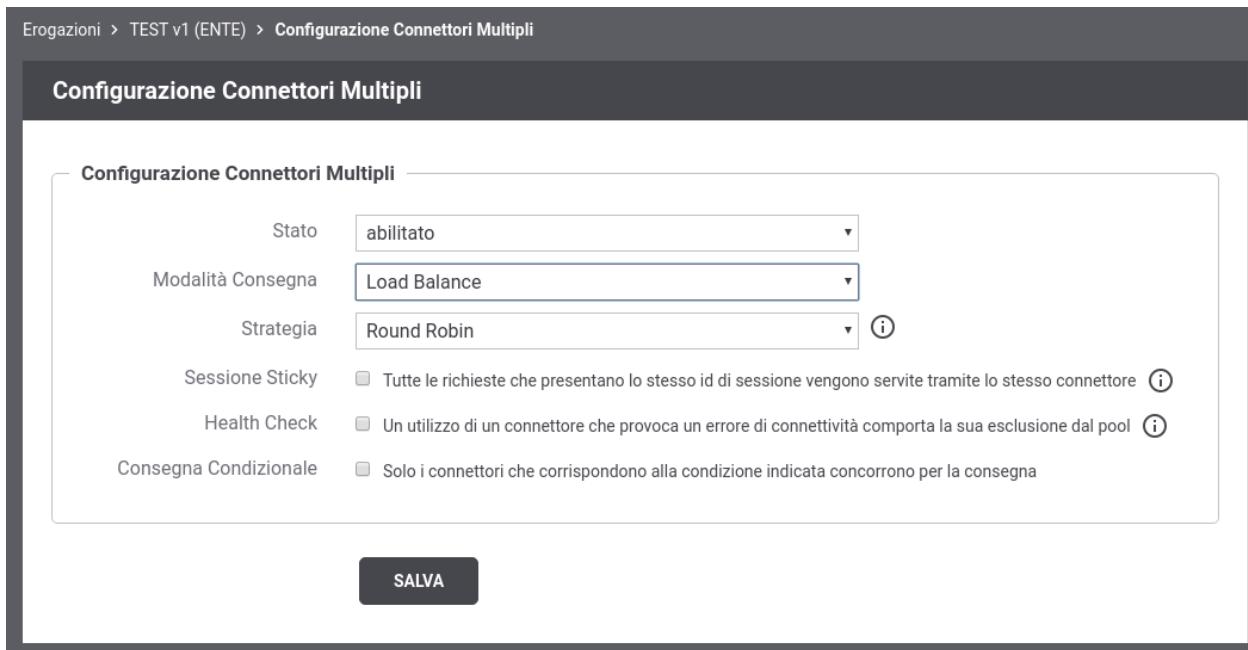


Figure2.32: Load Balancer

Vengono forniti differenti tipi di bilanciamento del carico:

- *Round Robin*: le richieste vengono distribuite in ordine tra i connettori registrati;
- *Weight Round Robin*: rispetto al Round Robin consente di riequilibrare eventuali server eterogenei tramite una distribuzione bilanciata rispetto al peso associato ad ogni connettore;
- *Random*: le richieste vengono distribuite casualmente tra i connettori registrati;
- *Weight Random*: rispetto al Random si ha una distribuzione casuale che considera però il peso associato ad ogni connettore;

- *Source IP hash*: combina l'indirizzo IP del client e l'eventuale indirizzo IP portato in un header appartenente alla classe «Forwarded-For» o «Client-IP» per generare una chiave hash che viene designata per un connettore specifico;
- *Least Connections*: la richiesta viene indirizzata verso il connettore che ha il numero minimo di connessioni attive.

La configurazione permette anche di abilitare una sessione sticky in modo che tutte le richieste che presentano lo stesso id di sessione vengano servite tramite lo stesso connettore. Se l'identificativo di sessione si riferisce ad una nuova sessione, viene selezionato un connettore rispetto alla strategia indicata.

Erogazioni > TEST v1 (ENTE) > Configurazione Connettori Multipli

Configurazione Connettori Multipli

Note: (*) Campi obbligatori

Configurazione Connettori Multipli

Stato: abilitato

Modalità Consegn: Load Balance

Strategia: Round Robin

Sessione Sticky: Tutte le richieste che presentano lo stesso id di sessione vengono servite tramite lo stesso connettore (i)

Health Check: Un utilizzo di un connettore che provoca un errore di connettività comporta la sua esclusione dal pool (i)

Consegn Condizionale: Solo i connettori che corrispondono alla condizione indicata concorrono per la consegna

Sessione Sticky

Identificativo Sessione: Cookie

Nome *: JSESSIONID

Max Age:

È possibile indicare la durata della sessione in secondi

SALVA

Figure2.33: Load Balancer con Sessione Sticky

L'identificativo di sessione utilizzato è individuabile tramite una delle seguenti modalità (Fig. 2.33):

- *Cookie*: nome di un cookie;
- *Header HTTP*: nome di un header http;
- *Url di Invocazione*: espressione regolare applicata sulla url di invocazione (l'espressione deve avere un match con l'intera url);
- *Parametro della Url*: nome del parametro presente nella url di invocazione;
- *Contenuto*: espressione (xPath o jsonPath) utilizzata per estrarre un identificativo dal body della richiesta;

- *Client IP*: indirizzo IP del client;
- *X-Forwarded-For*: header http appartenente alla classe «Forwarded-For» o «Client-IP»;
- *Template*: l'identificativo di sessione è il risultato dell'istanziazione del template fornito rispetto ai dati della richiesta;
- *Freemarker Template*: l'identificativo di sessione è ottenuto tramite il processamento di un Freemarker Template;
- *Velocity Template*: l'identificativo di sessione è ottenuto tramite il processamento di un Velocity Template.

È anche possibile attivare un “Passive Health Check” che verifica la connettività verso i connettori configurati. Un utilizzo di un connettore che provoca un errore di connettività comporta la sua esclusione dal pool dei connettori utilizzabili per un intervallo di tempo configurabile (Fig. 2.34).

Erogazioni > TEST v1 (ENTE) > Configurazione Connettori Multipli

Configurazione Connettori Multipli

Configurazione Connettori Multipli

Stato	abilitato
Modalità Consegna	Load Balance
Strategia	Round Robin
Sessione Sticky	<input type="checkbox"/> Tutte le richieste che presentano lo stesso id di sessione vengono servite tramite lo stesso connettore (i)
Health Check	<input checked="" type="checkbox"/> Un utilizzo di un connettore che provoca un errore di connettività comporta la sua esclusione dal pool (i)
Consegna Condizionale	<input type="checkbox"/> Solo i connettori che corrispondono alla condizione indicata concorrono per la consegna

Passive Health Check

Intervallo Esclusione	60
Indicare in secondi la durata dell'esclusione del connettore dal pool	

SALVA

Figure2.34: Load Balancer con Passive Health Check

È infine possibile attivare una funzione di selezione dei connettori che partecipano al bilanciamento delle richieste in funzione di parametri della richiesta stessa (Fig. 2.35). Per ulteriori dettagli sulle modalità di selezione condizionale dei connettori si rimanda alla sezione *Consegna Condizionale* poichè gli aspetti di questa configurazione sono identici a quelli descritti per la funzionalità di consegna condizionale.

Erogazioni > TEST v1 (ENTE) > Configurazione Connettori Multipli

Configurazione Connettori Multipli

Note: (*) Campi obbligatori

Configurazione Connettori Multipli

Stato	abilitato
Modalità Consegnna	Load Balance
Strategia	Round Robin
Sessione Sticky	<input type="checkbox"/> Tutte le richieste che presentano lo stesso id di sessione vengono servite tramite lo stesso connettore
Health Check	<input type="checkbox"/> Un utilizzo di un connettore che provoca un errore di connettività comporta la sua esclusione dal pool
Consegna Condizionale	<input checked="" type="checkbox"/> Solo i connettori che corrispondono alla condizione indicata concorrono per la consegna

Configurazione Condizionalità

Identificazione Condizione	Header HTTP
Nome *	X-FiltroCustom
Prefisso	
Suffisso	

Identificazione Condizione Fallita

Termina con Errore

Nessun Connnettore Utilizzabile

Termina con Errore

SALVA

Figure2.35: Selezione condizionale dei connettori che partecipano al bilanciamento

Elenco dei Connettori Bilanciati

Per le erogazioni di API è possibile definire connettori multipli con finalità di bilanciamento delle richieste in arrivo.

Dopo aver attivato la funzione di Load Balancer, nei contesti in cui l’interfaccia visualizzava l’endpoint di un connettore, viene adesso invece evidenziata la presenza della funzionalità di Load Balancer. In tale contesto è possibile definire i nuovi connettori accedendo all’elenco dei connettori registrati tramite il nuovo pulsante dedicato (Fig. 2.36).



Figure2.36: Pulsante per accedere all’elenco dei connettori

Accedendo all’elenco la prima volta si troverà il solo connettore di default definito al momento della registrazione dell’API erogata (Fig. 2.37).

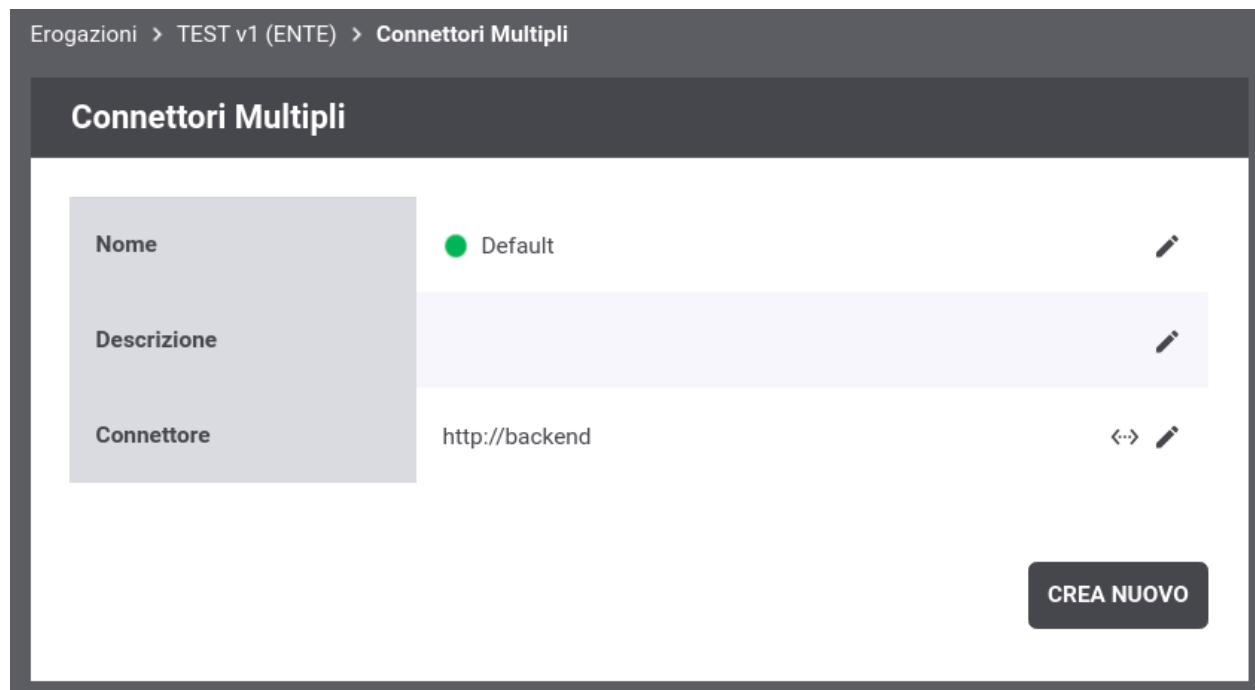


Figure2.37: Elenco dei connettori bilanciati con presenza del solo connettore di default

Tramite il pulsante *Crea Nuovo* è possibile registrare un nuovo connettore. Compilare il form come segue (Fig. 2.38):

- *Nome*: Assegnare un nome al connettore. È necessario che il nome indicato risulti univoco all’interno del pool dei connettori definiti per l’API.
- *Stato*: Indica lo stato del connettore. È possibile abilitare o disabilitare il singolo connettore anche dopo che è stato definito.
- *Descrizione*: Permette di fornire una descrizione generica.
- *Connettore*: Tramite la sezione *Connettore* è possibile fornire quelle informazioni tipicamente indicate sinora nella sezione “Connettore” dell’erogazione di una API.

All’interno della definizione dei dati di un connettore, è anche possibile riferire un Applicativo di tipo “server” precedentemente registrato come descritto nella sezione *Applicativi Server* (Fig. 2.39).

Erogazioni > TEST v1 (ENTE) > Connettori Multipli > Aggiungi

Note: (*) Campi obbligatori

Dati Generali

Nome *

Stato

Descrizione

Connettore

Utilizza Applicativo Server

Endpoint *

Autenticazione Http

Autenticazione Token

AutenticazioneHttps

Proxy

Ridefinisci Tempi Risposta

SALVA

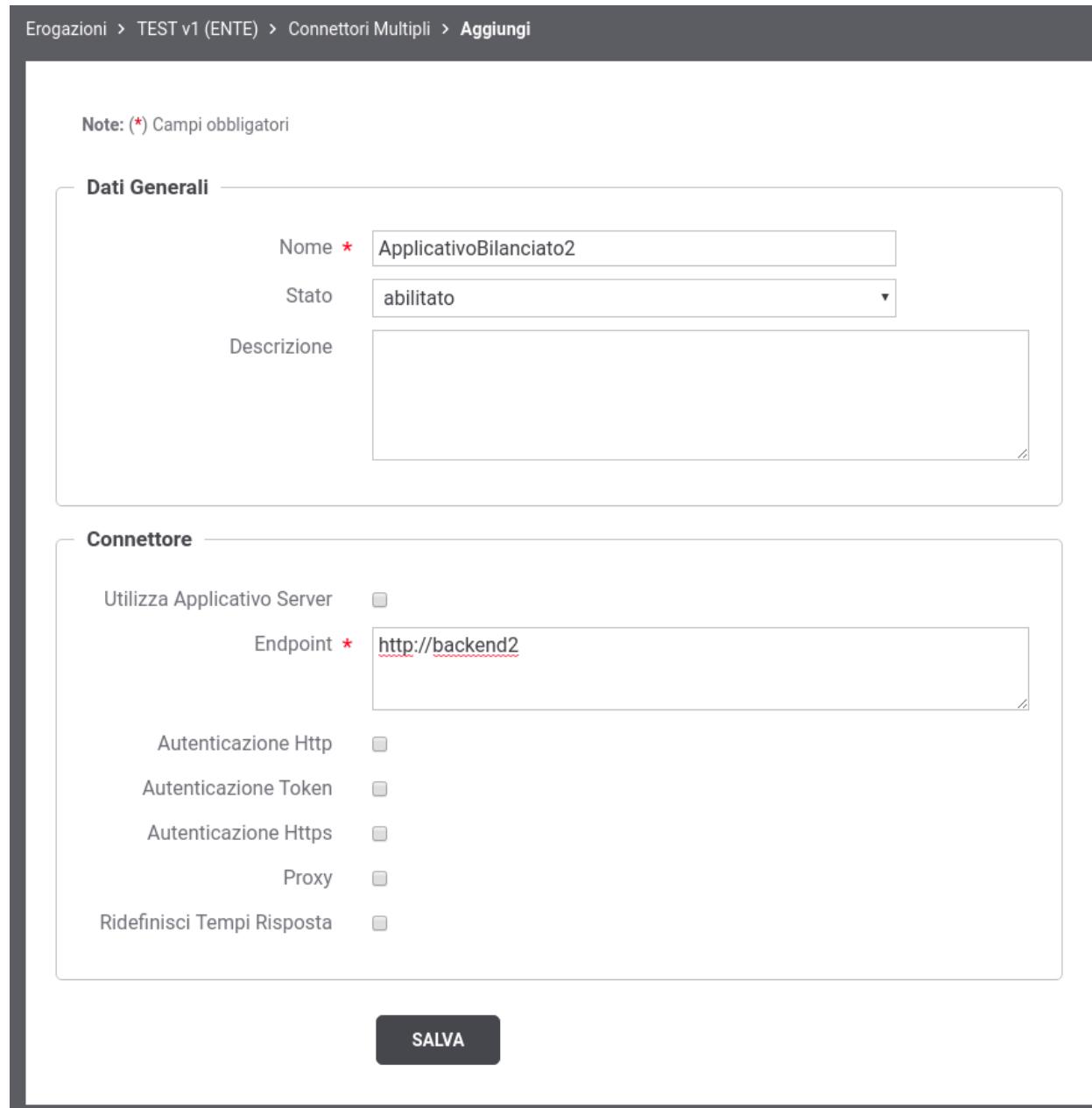


Figure2.38: Registrazione di un nuovo connettore per il bilanciamento del carico

Erogazioni > TEST v1 (ENTE) > Connettori Multipli > Aggiungi

Note: (*) Campi obbligatori

Dati Generali

Nome *

Stato

Descrizione

Connettore

Utilizza Applicativo Server

Applicativo

SALVA

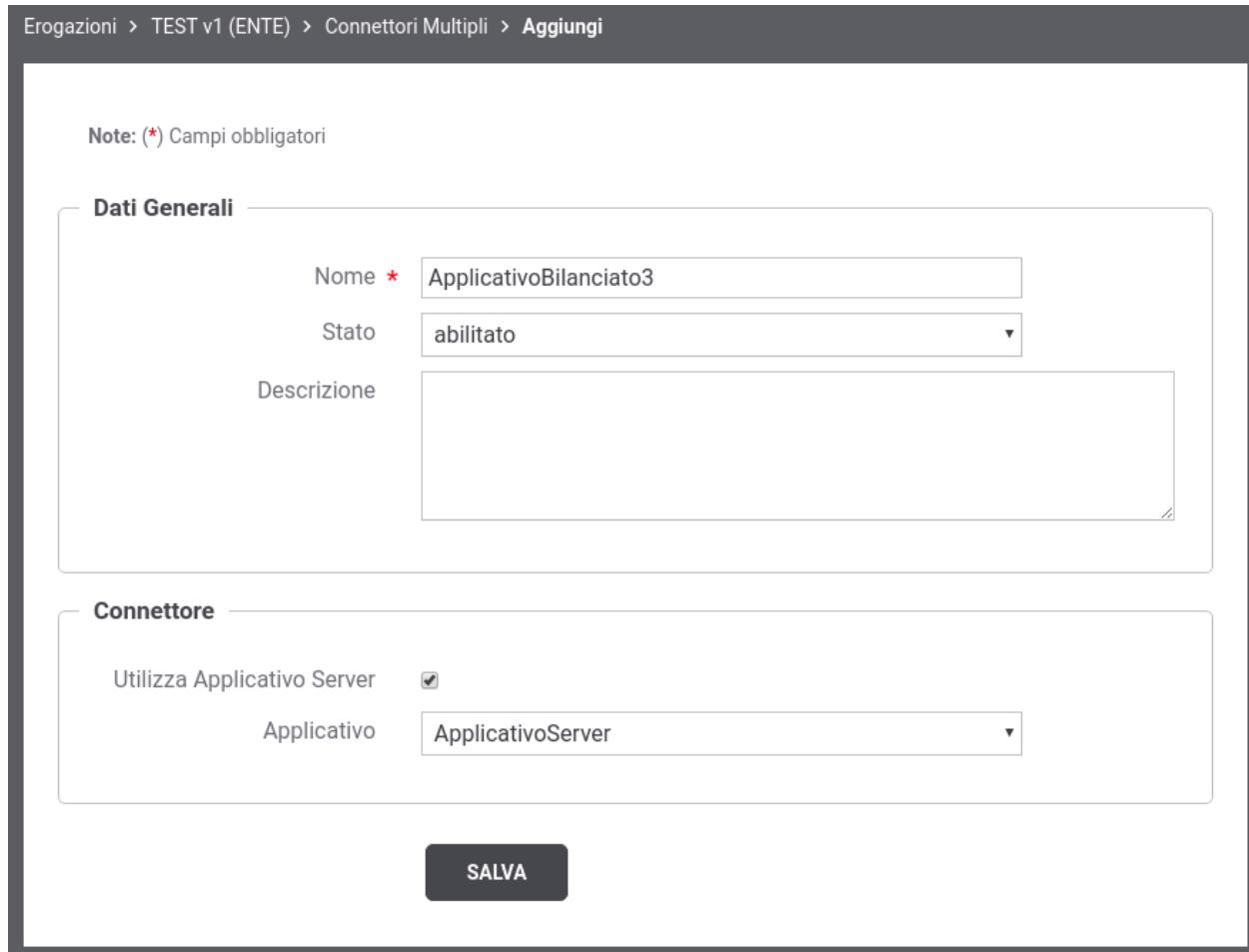
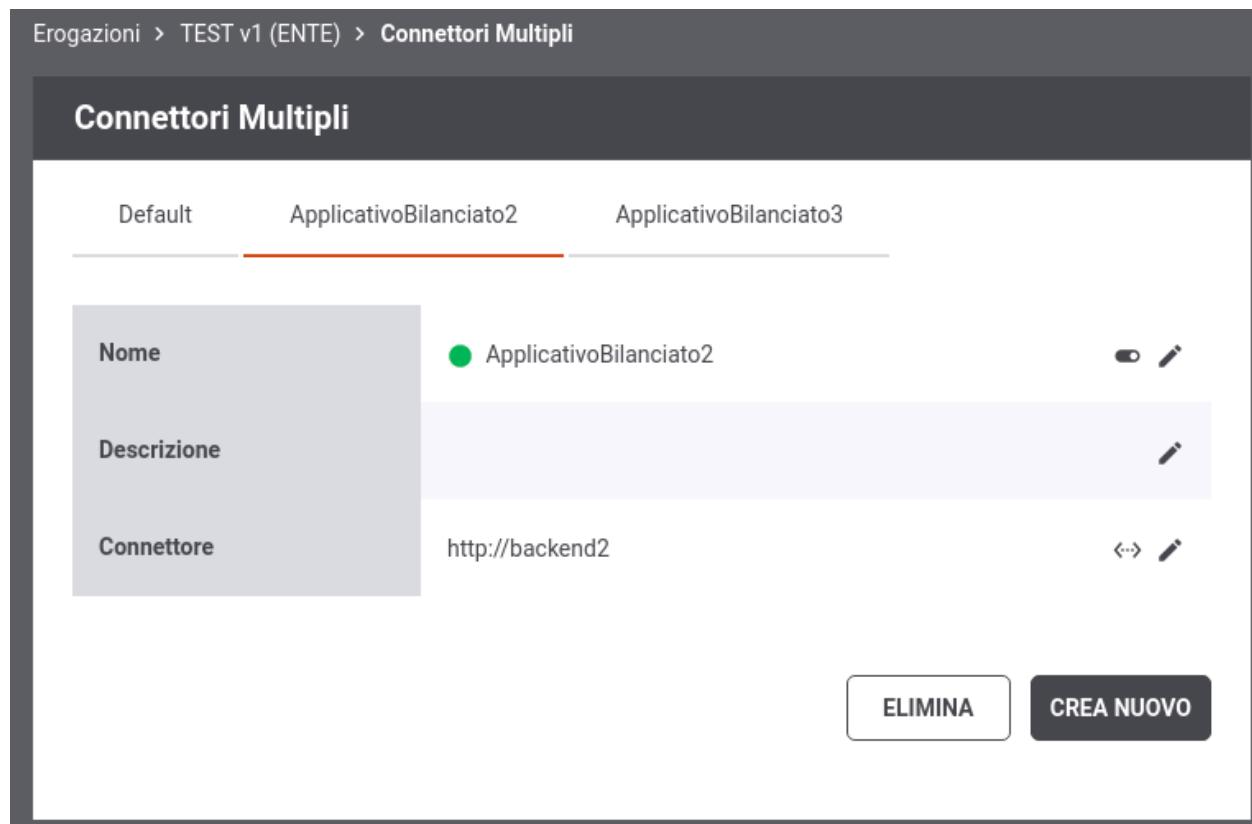


Figure2.39: Registrazione di un nuovo connettore, per il bilanciamento del carico, che riferisce un Applicativo Server

I nuovi connettori creati sono accessibili nell'elenco dei connettori (Fig. 2.40). I tab presenti nell'elenco riportano i nomi dei connettori configurati, e selezionando quello di interesse è possibile visualizzare e/o modificare i dati del connettore oltre ad eliminarlo tramite il pulsante *Elimina*.



Erogazioni > TEST v1 (ENTE) > **Connettori Multipli**

Connettori Multipli

Default ApplicativoBilanciato2 ApplicativoBilanciato3

Nome	ApplicativoBilanciato2	•	Elimina
Descrizione			Modifica
Connettore	http://backend2		Modifica

ELIMINA **CREA NUOVO**

Figure2.40: Elenco dei connettori bilanciati

Nel caso sia stato selezionato un tipo di Load Balancer “Weight” nell'elenco dei connettori sarà possibile anche associare un peso maggiore al singolo connettore (Fig. 2.41).

2.7.4 Consegnna Condizionale

Per le erogazioni di API è possibile definire connettori diversi, selezionati dinamicamente al verificarsi di specifiche condizioni.

Per abilitare una consegna condizionale è necessario accedere al dettaglio di una erogazione, dove viene visualizzato il valore del connettore, e successivamente cliccare sul pulsante che consente di gestire la configurazione dei connettori multipli (Fig. 2.42).

Dopo aver premuto il pulsante si accede ad una schermata che consente di abilitare una funzionalità relativa ai connettori multipli. In questa sezione, in particolare, viene descritta la funzionalità *Consegna Condizionale* che dovrà essere selezionata una volta abilitato lo stato relativo alla configurazione dei connettori multipli (Fig. 2.43). Per maggiori dettagli si rimanda alla sezione *Configurazione della Consegnna Condizionale*.

Una volta attivata la funzione di Consegnna Condizionale, nei contesti in cui l'interfaccia visualizzava l'endpoint di un connettore, viene adesso invece evidenziata la presenza di tale funzionalità. In tale contesto è possibile definire i nuovi connettori accedendo all'elenco dei connettori registrati tramite il nuovo pulsante dedicato (Fig. 2.44) come descritto nella sezione *Elenco dei Connettori*.

Erogazioni > api-config v1 (ENTE) > **Connettori Multipli**

Connettori Multipli

Default ApplicativoBilanciato2 ApplicativoBilanciato3

Nome	ApplicativoBilanciato2	•	
Descrizione			
Connettore	http://backend.applicativo2		
Load Balance	Weight: 1		

ELIMINA **CREA NUOVO**

Figure2.41: Elenco dei connettori bilanciati con opzione “Weight”

Connettore http://backend

Figure2.42: Pulsante per la configurazione dei connettori multipli

Erogazioni > api-config v1 (ENTE) > **Configurazione Connettori Multipli**

Configurazione Connettori Multipli

Note: (*) Campi obbligatori

Configurazione Connettori Multipli

Stato	abilitato
Modalità Consegnna	Consegna Condizionale

La consegna avviene sul connettore che corrisponde alla condizione indicata

Configurazione Condizionalità

Selezione Connettore By	Filtro
Identificazione Condizione	Header HTTP
Nome *	X-FiltroCustom
Prefisso	
Suffisso	

Identificazione Condizione Fallita

Termina con Errore

Nessun Connettore Utilizzabile

Termina con Errore

SALVA

Figure2.43: Consegnna Condizionale

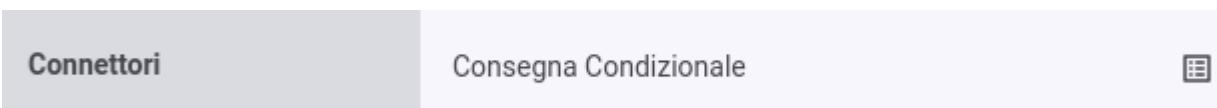


Figure2.44: Pulsante per accedere all'elenco dei connettori

Configurazione della Consegna Condizionale

Per abilitare la funzionalità di Consegna Condizionale accedere alla sezione di dettaglio di una erogazione di API e cliccare sul pulsante di configurazione dei connettori multipli (Fig. 2.45).

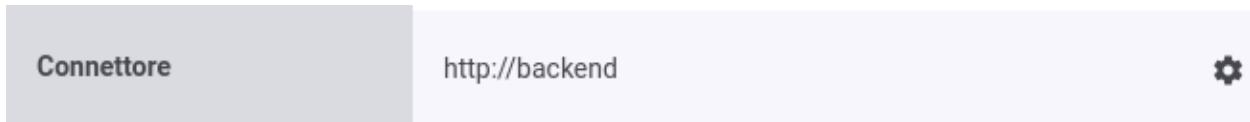


Figure2.45: Pulsante per la configurazione dei connettori multipli

Dopo aver premuto il pulsante si accede ad una schermata che consente di abilitare una funzionalità relativa ai connettori multipli. In questa sezione, in particolare, viene descritta la funzionalità *Consegna Condizionale* che dovrà essere selezionata una volta abilitato lo stato relativo alla configurazione dei connettori multipli (Fig. 2.46).

Il connettore su cui verrà inoltrata la richiesta pervenuta sul Gateway, viene selezionato in base al suo nome o a un filtro associato al connettore stesso. La modalità di selezione desiderata deve essere indicata tramite la voce “Selezione Connettore By”. Il valore del filtro (utilizzato per identificare il connettore di consegna) o il nome del connettore stesso, viene individuato all’interno della richiesta attraverso una delle seguenti modalità:

- *Header HTTP*: nome di un header http;
- *Url di Invocazione*: espressione regolare applicata sulla url di invocazione (l’espressione deve avere un match con l’intera url);
- *Parametro della Url*: nome del parametro presente nella url di invocazione;
- *SOAPAction*: individua una operazione SOAP;
- *Contenuto*: espressione (xPath o jsonPath) utilizzata per estrarre un identificativo dal body della richiesta;
- *Client IP*: indirizzo IP del client;
- *X-Forwarded-For*: header http appartenente alla classe «Forwarded-For» o «Client-IP»;
- *Template*: l’identificativo di sessione è il risultato dell’istanziazione del template fornito rispetto ai dati della richiesta;
- *Freemarker Template*: l’identificativo di sessione è ottenuto tramite il processamento di un Freemarker Template;
- *Velocity Template*: l’identificativo di sessione è ottenuto tramite il processamento di un Velocity Template;

I campi “Prefisso” e “Suffisso” permettono di anteporre al valore estratto dalla richiesta un prefisso e/o un suffisso prima di utilizzare tale valore per l’identificazione del connettore (sia tramite nome che tramite filtro).

Tramite le checkbox “Termina con Errore” è infine possibile configurare l’erogazione per utilizzare uno specifico connettore di default, invece di terminare la transazione con errore, nel caso la condizione non sia presente nella richiesta o non permetta di identificare alcun connettore all’interno del pool. Nel caso non venga terminata la transazione con errore, è anche possibile impostare l’emissione o meno di un diagnostico che segnali la condizione fallita (esempio riportato nella figura Fig. 2.47).

Le regole per la selezione del connettore sopra descritte possono essere ridefinite per singole o gruppi di operazioni attraverso la definizione di regole specifiche, accedendo al link “regole” presente nella maschera di configurazione.

La creazione di una regola specifica deve innanzitutto identificare le operazioni dell’API a cui la regola è riferita tramite il campo “Risorsa” o “Azione” attraverso una delle seguenti modalità:

- *Nome Azione o Risorsa*: il nome esatto dell’azione o della risorsa su cui verrà applicativa la regola; può in alternativa essere utilizzata un’espressione regolare (es. ^(?:POST.operazione1|GET.operazione2)\$)

Erogazioni > api-config v1 (ENTE) > **Configurazione Connettori Multipli**

Configurazione Connettori Multipli

Note: (*) Campi obbligatori

Configurazione Connettori Multipli

Stato	abilitato
Modalità Consegna	Consegna Condizionale

La consegna avviene sul connettore che corrisponde alla condizione indicata

Configurazione Condizionalità

Selezione Connettore By	Filtro
Identificazione Condizione	Header HTTP
Nome *	X-FiltroCustom
Prefisso	
Suffisso	

Identificazione Condizione Fallita

Termina con Errore

Nessun Connettore Utilizzabile

Termina con Errore

SALVA

Figure2.46: Consegna Condizionale

Configurazione Condizionalità

Selezione Connettore By	Filtro
Identificazione Condizione	Header HTTP
Nome *	X-FiltroCustom
Prefisso	
Suffisso	

[Regole\(0\)](#)

Identificazione Condizione Fallita

<input type="checkbox"/> Termina con Errore	
Emissione Diagnostico	con livello di severità 'error'
Utilizza Connuttore	Default

Nessun Connuttore Utilizzabile

<input type="checkbox"/> Termina con Errore	
Emissione Diagnostico	con livello di severità 'error'
Utilizza Connuttore	Default

Figure2.47: Consegnare Condizionale con Connuttore di Default

- *HttpMethod e Path* (utilizzabile solo su API REST): metodo http e path di una risorsa dell'API; è possibile indicare qualsiasi metodo o qualsiasi path con il carattere speciale “*”. È inoltre possibile definire solamente la parte iniziale di un path attraverso lo “*”. Alcuni esempi:
 - “POST /resource”
 - “* /resource”
 - “POST *”
 - “* /resource/*”

Nella figura Fig. 2.48 viene visualizzata la maschera di creazione di una regola specifica. Le modalità di identificazione del nome del connettore o del valore del filtro sono le medesime descritte in precedenza.

Note: (*) Campi obbligatori

Regola

Nome Regola *

Risorsa * ⓘ

Identificazione Condizione

Nome *

Prefisso

Suffisso

SALVA

Figure2.48: Regola di Consegnna Condizionale per Operazione

Elenco dei Connettori

Per le erogazioni di API è possibile definire connettori multipli con finalità di selezione condizionale del connettore a cui inoltrare le richieste in arrivo.

Dopo aver attivato la funzione di Consegnna Condizionale, nei contesti in cui l'interfaccia visualizzava l'endpoint di un connettore, viene adesso invece evidenziata la presenza della funzionalità attivata. In tale contesto è possibile definire i nuovi connettori accedendo all'elenco dei connettori registrati tramite il nuovo pulsante dedicato (Fig. 2.49).

Accedendo all'elenco la prima volta si troverà il solo connettore di default definito al momento della registrazione dell'API erogata (Fig. 2.50).

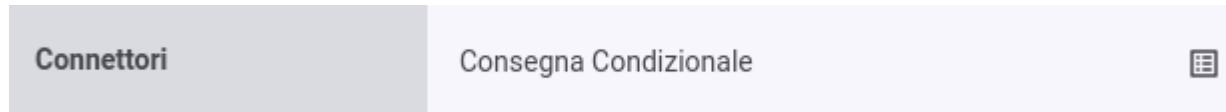


Figure2.49: Pulsante per accedere all'elenco dei connettori

A screenshot of a web-based application interface. The top navigation bar shows 'Erogazioni > api-config v1 (ENTE) > Connettori Multipli'. Below this, a dark header bar displays 'Connettori Multipli'. The main content area is a table with four rows. The first row has a green circular icon with a dot, followed by the text 'Default' and an edit icon. The second row has a blank 'Descrizione' field and an edit icon. The third row has a 'Connettore' field containing 'http://127.0.0.1:8080/govwayAPIConfig/' with a link and edit icons. The fourth row has a 'Filtri' field and an edit icon. At the bottom right of the table is a dark button labeled 'CREA NUOVO'.

Figure2.50: Elenco dei connettori con presenza del solo connettore di default

Tramite il pulsante *Crea Nuovo* è possibile registrare un nuovo connettore. Compilare il form come segue (Fig. 2.51):

- *Nome*: Assegnare un nome al connettore. È necessario che il nome indicato risulti univoco all'interno del pool dei connettori definiti per l'API.
- *Stato*: Indica lo stato del connettore. È possibile abilitare o disabilitare il singolo connettore anche dopo che è stato definito.
- *Descrizione*: Permette di fornire una descrizione generica.
- *Filtri*: Nel caso sia stata configurata una selezione del connettore basata sui filtri, questo campo permette di assegnare al connettore i valori con cui verrà selezionato dal Gateway.
- *Connettore*: Tramite la sezione *Connettore* è possibile fornire quelle informazioni tipicamente indicate sinora nella sezione “Connettore” dell'erogazione di una API.

All'interno della definizione dei dati di un connettore, è anche possibile riferire un Applicativo di tipo “server” precedentemente registrato come descritto nella sezione *Applicativi Server* (Fig. 2.52).

I nuovi connettori creati sono accessibili nell'elenco dei connettori (Fig. 2.53). I tab presenti nell'elenco riportano i nomi dei connettori configurati, e selezionando quello di interesse è possibile visualizzare e/o modificare i dati del connettore oltre ad eliminarlo tramite il pulsante *Elimina*.

Nel caso sia stata configurata una selezione del connettore basata sui filtri, si deve procedere ad assegnare anche al connettore di default uno o più valori nei filtri in modo che sia selezionabile dal Gateway. Tale operazione non è necessaria solamente se si desidera utilizzare il connettore di default solamente nei casi in cui la condizione non è identificata nella richiesta o non abbia consentito ad identificare un connettore.

2.7.5 ProxyPassReverse per Header HTTP Location e Set-Cookie

La funzionalità “ProxyPassReverse” può essere attivata per gestire due tipologie di header HTTP presenti nelle risposte:

- modificare la URL presente negli header HTTP “Location” e “Content-Location” sulle risposte di reindirizzamento HTTP sostituendo il backend server (se presente come url assoluta) e il context path con l'indirizzo di esposizione dell'API Su GovWay;
- modificare gli attributi “Path” e “Domain”, presenti negli header HTTP “Set-Cookie” restituiti dal backend server, sostituendo i valori rispettivamente con il context path e con il dominio di esposizione dell'API Su GovWay.

Nota: L'indirizzo di esposizione di una API su GovWay è configurabile e personalizzabile come descritto nella sezione *URL di Invocazione API*.

La configurazione di default di GovWay effettua la traduzione solamente delle URL presenti negli header HTTP “Location” e “Content-Location” su API di tipo REST.

È possibile personalizzare la configurazione registrando le seguenti *Proprietà* sulla singola erogazione o fruizione:

- *connettori.proxyPassReverse.enabled*: (default: true su API REST, false su API SOAP) consente di abilitare o disabilitare la funzionalità di proxy pass reverse sulle risposte di reindirizzamento HTTP (Location). I valori associabili alle proprietà sono “true” o “false”.
- *connettori.proxyPassReverse.headers*: (default: Location,Content-Location) consente di indicare i nomi degli header HTTP della risposta su cui verrà attuata la trasformazione della url.
- *connettori.proxyPassReverse.setCookie.enabled*: (default: false) consente di abilitare la funzionalità di proxy pass reverse sugli header HTTP “Set-Cookie”. I valori associabili alle proprietà sono “true” o “false”. La proprietà abilita la traduzione di entrambi gli attributi “Path” e “Domain”. In alternativa è possibile utilizzare le proprietà seguenti per abilitare puntualmente la traduzione solo su uno dei due attributi:

Erogazioni > api-config v1 (ENTE) > Connettori Multipli > Aggiungi

Note: (*) Campi obbligatori

Dati Generali

Nome *

Stato

Descrizione

Filtri

Connettore

Utilizza Applicativo Server

Endpoint *

Autenticazione Http

Autenticazione Token

AutenticazioneHttps

Proxy

Ridefinisci Tempi Risposta

SALVA

Figure2.51: Registrazione di un nuovo connettore per la consegna condizionale

Erogazioni > api-config v1 (ENTE) > Connettori Multipli > Aggiungi

Note: (*) Campi obbligatori

Dati Generali

Nome *

Stato

Descrizione

Filtri

Connettore

Utilizza Applicativo Server

Applicativo

SALVA

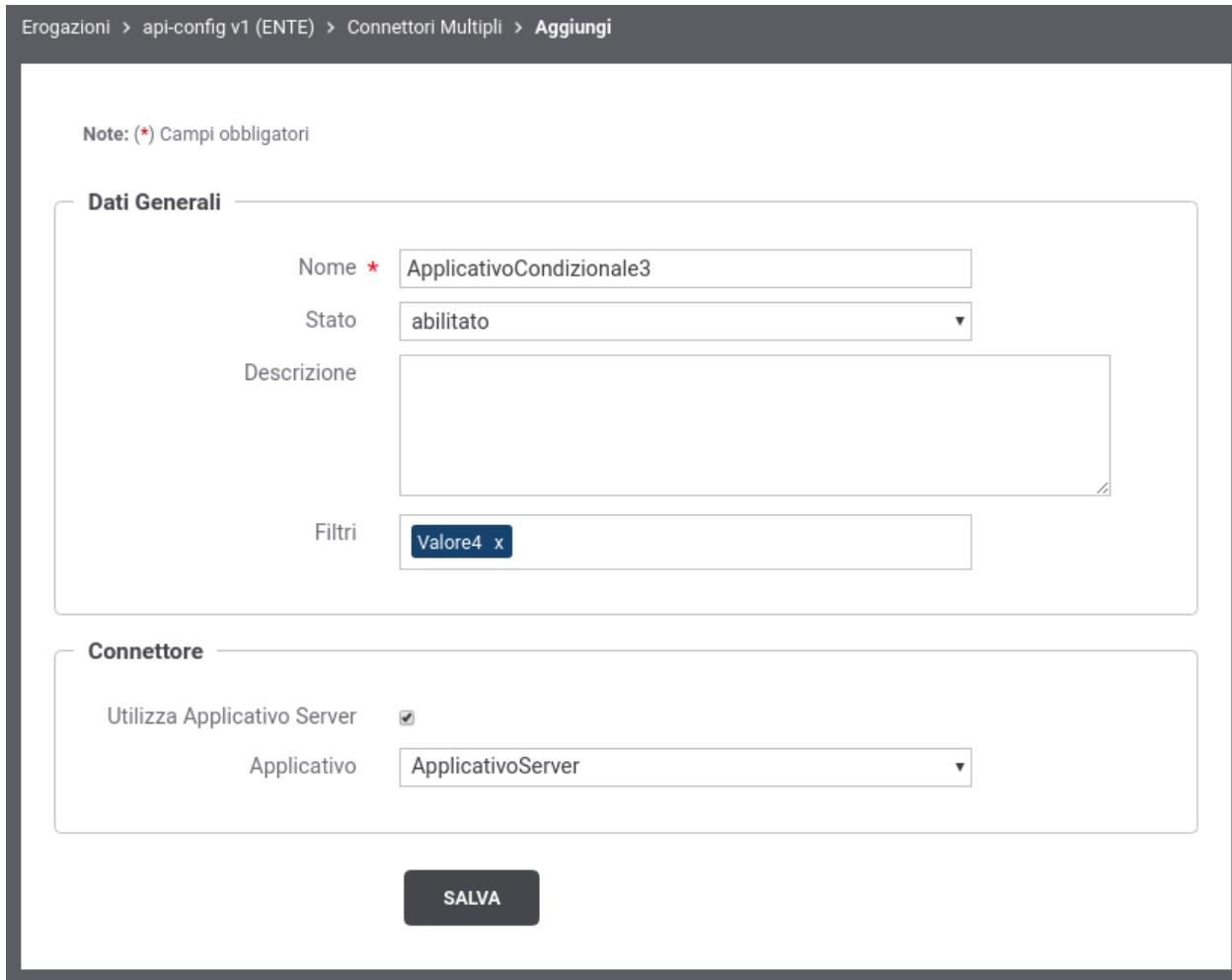


Figure2.52: Registrazione di un nuovo connettore, per la consegna condizionale, che riferisce un Applicativo Server

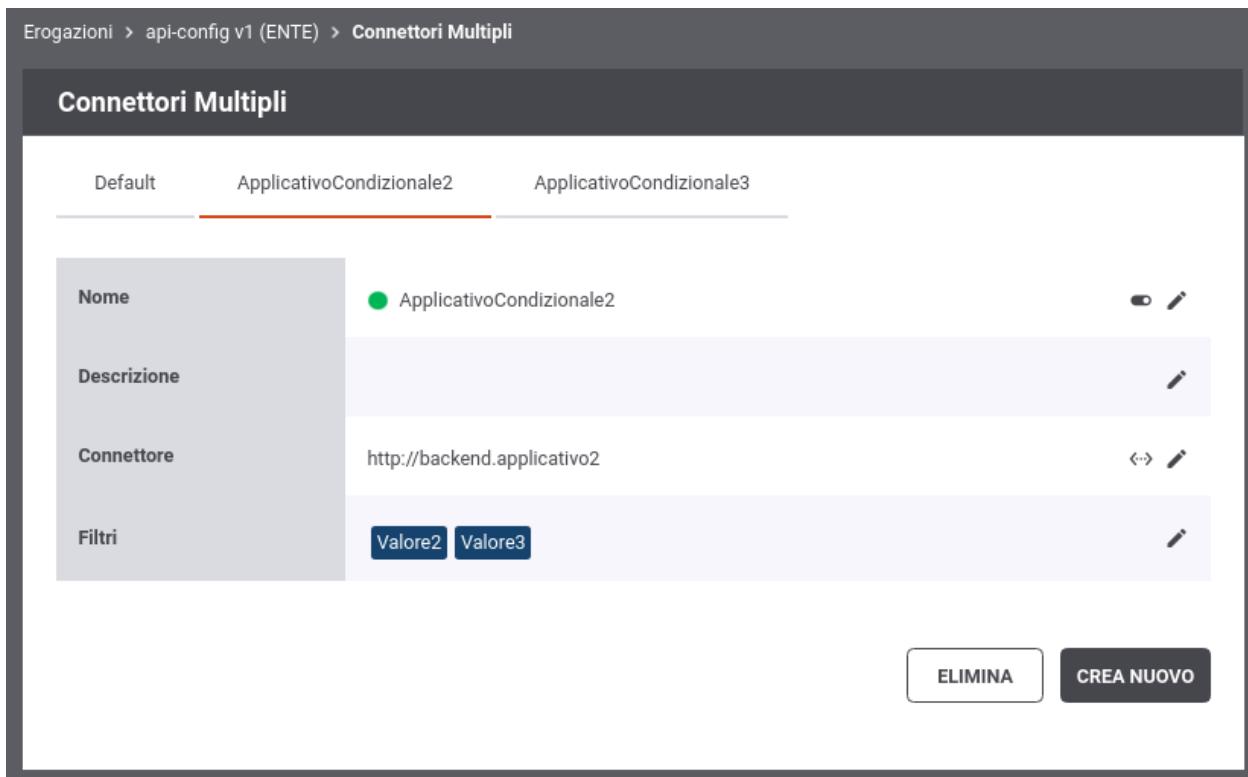


Figure2.53: Elenco dei connettori selezionabili per la consegna

- *connettori.proxyPassReverse.setCookie.path.enabled*
- *connettori.proxyPassReverse.setCookie.domain.enabled*
- *connettori.proxyPassReverse.setCookie.headers*: (default: Set-Cookie) consente di indicare i nomi degli header HTTP della risposta su cui viene atteso un cookie in cui verrà attuata la trasformazione del path e/o del domain.

Configurazione su API SOAP

Poichè la funzionalità di proxy pass reverse è completamente disabilitata per default su API di tipo SOAP, per attivarla è necessario attuare la registrazione delle *Proprietà* “*connettori.proxyPassReverse.enabled*” e/o “*connettori.proxyPassReverse.setCookie.enabled*” (o proprietà specifiche per path/domain) con il valore “true”.

La sola registrazione delle proprietà non è sufficiente su API SOAP poichè per default gli unici header HTTP della risposta che vengono inoltrati dal backend verso il client sono quelli relativi alle funzionalità CORS (Access-Control-*). È possibile configurare GovWay per far inoltrare gli header “Location”, “Content-Location” o “Set-Cookie” al client in una delle seguenti due modalità:

- *puntuale sull'erogazione/fruizione di API*: creare una regola di trasformazione sugli header http di risposta che consenta l'inoltro verso il client degli header “Location”, “Content-Location” o “Set-Cookie” (per ulteriori dettagli sulle trasformazioni far riferimento alla sezione *Regole di Trasformazione della Risposta*). Nella figura (Fig. 2.54) viene fornito un esempio di configurazione.

Table2.1: Trasformazione per attuare proxy pass reverse su API SOAP

Nome	Valore	Operazione
Location	\${headerResponse:Location}	update
Content-Location	\${headerResponse:Content-Location}	update

continues on next page

Table 2.1 – continua dalla pagina precedente

Nome	Valore	Operazione
Set-Cookie	\${headerResponse:Set-Cookie}	update

Nome	Operazione	Valore
Content-Location	update	\${headerResponse:Content-Location}
Location	update	\${headerResponse:Location}
Set-Cookie	update	\${headerResponse:Set-Cookie}

Figure 2.54: Forward header http di risposta su API SOAP

- *globale per tutte le API SOAP*: editare il file <directory-lavoro>/govway_local.properties aggiungendo la seguente riga:

```
# Header su cui attuare il cookie proxy pass reverse per API SOAP
org.openscoop2.pdd.soap.headers.whiteList.response=Access-Control-*,  

    ↴Location,Content-Location,Set-Cookie
```

2.8 Gestione CORS

Quando un'applicazione client in esecuzione su un browser (es. codice javascript) richiede l'accesso ad una risorsa di un differente dominio, protocollo o porta tale richiesta viene gestita dal browser tramite una politica di *cross-origin HTTP request (CORS)*. Il CORS definisce un modo nel quale un browser ed un server (o il gateway) possono interagire per abilitare interazioni attraverso differenti domini.

In GovWay è possibile abilitare la gestione del CORS sia globalmente, in modo che sia valida per tutte le APIs, che singolarmente sulla singola erogazione o fruizione.

È possibile modificare le impostazioni CORS seguendo il collegamento presente nella riga *Gestione CORS* del dettaglio di una erogazione o fruizione. L'impostazione permette di ridefinire la configurazione globale; i campi del form sono i medesimi descritti nella configurazione globale (sezione *Gestione CORS*).

2.9 Differenziare le configurazioni specifiche per risorsa/azione

Le configurazioni specifiche che andiamo a descrivere si possono differenziare per sottoinsiemi delle azioni/risorse presenti nel servizio erogato/fruito. Il sistema crea automaticamente una configurazione unica, valida per tutte le azioni/risorse del servizio. Per intervenire su tale configurazione, o crearne di nuove, sia accede al collegamento presente nella colonna *Configurazione*, in corrispondenza della voce di erogazione/fruitazione in elenco. Le funzionalità di configurazione disponibili per ciascun sottoinsieme di azioni/risorse sono raggruppabili in:

- *Controllo Accessi*: per configurare i criteri di autenticazione, autorizzazione e gestione token delle richieste.
- *Rate Limiting*: per configurare i meccanismi di controllo del traffico a salvaguardia delle prestazioni.
- *Validazione*: per configurare i criteri di validazione dei messaggi in transito sul gateway.
- *Caching Risposta*: per configurare l'utilizzo della cache per i messaggi di risposta.
- *Sicurezza Messaggio*: per configurare le misure di sicurezza applicate a livello del messaggio.
- *Tracciamento*: per configurare specifiche modalità di estrazione dati, dalle comunicazioni in transito, per l'arricchimento della traccia prodotta.
- *Trasformazioni*: per configurare le operazioni di trasformazione attivabili sui flussi in entrata ed uscita.
- *MTOM*: per configurare l'utilizzo del protocollo ottimizzato per l'invio di attachment tra nodi SOAP.
- *Registrazione Messaggi*: consente di ridefinire le politiche di archiviazione dei payload rispetto a quanto previsto dalla configurazione di default (vedi sezione *Tracciamento*).

Per creare un nuovo gruppo di configurazione, dopo aver seguito il collegamento *visualizza* relativo all'erogazione/fruitazione selezionata, si preme il pulsante *Aggiungi*

Note: (*) Campi obbligatori

Configurazione

Nome Gruppo *	Gruppo2
Risorse *	POST /store/pdf
Mode	Eredita Da
Gruppo	'Predefinito'

SALVA

Figure2.55: Aggiunta di un gruppo di configurazioni

Compilare il form di creazione della nuova configurazione (Fig. 2.55):

- *Azioni*: selezionare dall'elenco le azioni sulle quali si vuole abbia effetto la nuova configurazione.

- *Mode*: effettuare la scelta tra *Eredita Da* e *Nuova*. Scegliendo la prima opzione, verrà creata una configurazione clone di quella selezionata nell'elemento del form subito successivo (Configurazione). Scegliendo la seconda opzione, si procederà alla creazione di una nuova configurazione, specificando subito le informazioni di Controllo degli Accessi e Connettore.

Nota: Nota Dopo aver creato ulteriori configurazioni, si tenga presente che la configurazione di default verrà applicata alle sole azioni per le quali non è presente una regola di configurazione specifica.

Nota: Nota È possibile disabilitare un'intera configurazione, senza la necessità di eliminarla, utilizzando il collegamento presente nella colonna «Abilitato» in corrispondenza dell'elemento di configurazione. Un successivo clic farà tornare la configurazione nello stato abilitato.

2.10 Controllo degli Accessi

Tramite questa funzionalità è possibile configurare i criteri di gestione token, autenticazione e autorizzazione delle richieste in ingresso sul gateway. Per aggiungere questa funzionalità si procede selezionando prima il collegamento, presente nella colonna «Configurazione», relativo all'erogazione/fruizione presente nell'elenco. Successivamente si utilizza il collegamento, presente nella colonna «Controllo Accessi», relativamente alla configurazione che si vuole modificare (Fig. 2.56).

Le sezioni seguenti descrivono le modalità per configurare gli aspetti che compongono il controllo degli accessi.

2.10.1 Autenticazione Token

Questa sezione consente di configurare il controllo degli accessi basato su Bearer Token OAuth2. Facendo transitare lo stato su «abilitato» compare l'elemento *Policy* (obbligatorio) per la selezione della policy di autenticazione token che si vuole applicare. In questa lista a discesa saranno visualizzate tutte le *Token Policy* di tipo *Validazione* che sono state registrate in precedenza. Per le istruzioni sulla registrazione delle Token Policy si faccia riferimento alla sezione *Token Policy*.

Una volta selezionata la policy compariranno sotto gli elementi per stabilire le specifiche azioni da abilitare rispetto al totale di quelle previste nella policy stessa (Fig. 2.57).

Supponendo che la policy copra tutti gli aspetti disponibili, le opzioni configurabili sono le seguenti:

- *Token Opzionale*: consente di non forzare i richiedenti al passaggio del token, che rimane quindi un'operazione opzionale.
- *Introspection*: consente di abilitare/disabilitare l'operazione di Token Introspection, al fine di validare il token ricevuto ed ottenere le metainformazioni associate (ad esempio scope e riferimento al possessore del token). Selezionando l'opzione *WarningOnly* è possibile non rendere bloccante l'evento di fallimento della validazione, ottenendo come unico effetto l'emissione di un messaggio diagnostico di segnalazione.
- *User Info*: consente di abilitare/disabilitare l'operazione UserInfo al fine di ottenere le informazioni di dettaglio dell'utente possessore del token. Selezionando l'opzione *WarningOnly* è possibile non rendere bloccante l'evento di fallimento della validazione, ottenendo come unico effetto l'emissione di un messaggio diagnostico di segnalazione.

Erogazioni > api-config v1 (ENTE) > Configurazione > **Controllo Accessi**

Controllo Accessi

^ Autenticazione Token

Stato

^ Autenticazione Trasporto

Stato

^ Identificazione Attributi

Stato

^ Autorizzazione

Stato

^ Autorizzazione Contenuti

Stato

Figure2.56: Controllo degli Accessi

Autenticazione Token

Stato	abilitato
Policy *	Google
Token Opzionale	<input type="checkbox"/>
Validazione JWT	disabilitato
Introspection	abilitato
User Info	abilitato
Token Forward	abilitato

Required Claims

Issuer	<input type="checkbox"/>
ClientId	<input checked="" type="checkbox"/>
Subject	<input type="checkbox"/>
Username	<input type="checkbox"/>
eMail	<input type="checkbox"/>

Figure2.57: Configurazione della gestione token

- *Token Forward*: consente di abilitare/disabilitare l'operazione di inoltro, al servizio destinatario, del token ricevuto dal mittente.

Le azioni che sono state abilitate saranno effettuate in accordo a quanto configurato nella relativa Token Policy selezionata.

Nota: È disponibile la Token Policy *Google* preconfigurata in modo da utilizzare i servizi di elaborazione token esposti pubblicamente da Google e quindi:

- La Validazione JWT basata su *Google - ID Token* (<https://www.googleapis.com/oauth2/v3/certs>)
- Il servizio di token introspection basato su *Google - TokenInfo* (<https://www.googleapis.com/oauth2/v3/tokeninfo>)
- Il servizio di User Info basato su *Google - UserInfo* (<https://www.googleapis.com/oauth2/v3/userinfo>)

È possibile inoltre far verificare la presenza obbligatoria delle seguenti metainformazioni all'interno del token:

- Issuer
- ClientId
- Subject
- Username
- Email

2.10.2 Autenticazione Trasporto

In questa sezione è possibile configurare il meccanismo di autenticazione richiesto per l'accesso al servizio.

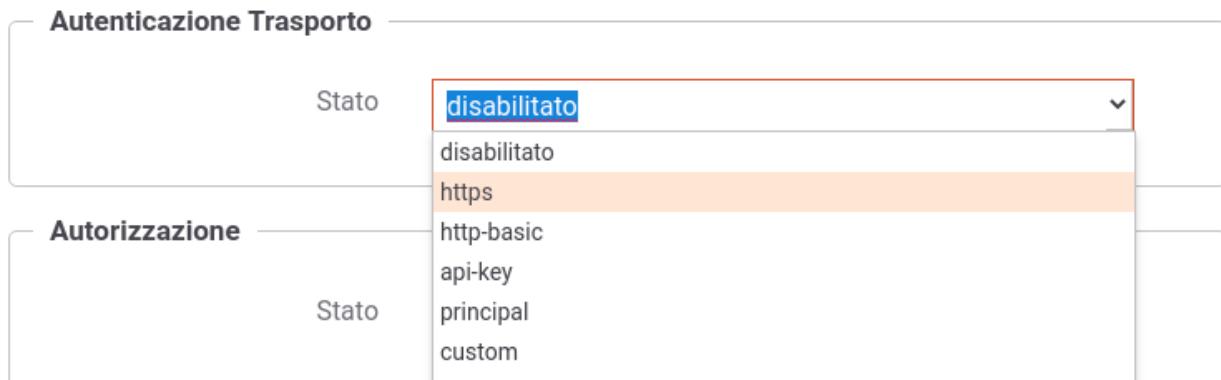


Figure2.58: Configurazione dell'autenticazione del servizio

Come mostrato in Fig. 2.58 la configurazione dell'autenticazione deve essere effettuata attraverso la selezione di un tipo di autenticazione tra quelli disponibili:

- disabilitato

Nessuna autenticazione.

- https

(Fig. 2.59) La richiesta deve possedere un certificato client X509. La presenza del certificato client nella richiesta è obbligatoria a meno che non sia abilitato il flag *Opzionale*. Per maggiori informazioni sulla configurazione necessaria

affinchè il certificato client sia ricevuto dal gateway si faccia riferimento alla sezione `install_ssl_server` della “Guida di Installazione”.

Se è presente un certificato client, il gateway cercherà di identificare un applicativo o un soggetto a cui è stato associato il certificato come credenziale di accesso (per ulteriori dettagli si rimanda alle sezioni *Creazione di un soggetto* e *Creazione di un applicativo*); l’identificazione non è obbligatoria ma nel caso avvenga con successo l’applicativo o il soggetto verrà registrato nei log e potrà essere utilizzato anche ai fini di autorizzazione puntuale e per ruoli (*Autorizzazione*).

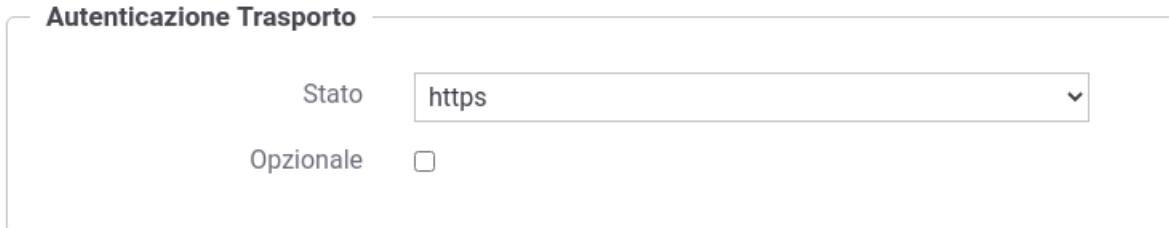


Figure2.59: Configurazione Autenticazione “https”

Durante il processo di autenticazione https, sull’eventuale certificato client presente viene effettuata una validazione della scadenza. È possibile disabilitare tale controllo o abilitarne altri registrando una delle seguenti *Proprietà* sull’erogazione o sulla fruizione:

- `authentication.https.validityCheck` : (true/false, default:true) consente di disabilitare il controllo sulla scadenza dei certificati ricevuti;
- `authentication.https.truststore.enabled` : (true/false, default:false) consente di abilitare l’utilizzo di un truststore per verificare i certificati ricevuti;
- `authentication.https.truststore`, `authentication.https.truststore.password` e `authentication.https.truststore.type` : parametri che definiscono un truststore da utilizzare per la verifica dei certificati ricevuti;
- `authentication.https.truststore.crls`: in presenza di un truststore, è possibile indicare delle CRLs per verificare se un certificato risultato revocato.
- `authentication.https.truststore.ocspPolicy`: in alternativa alla validazione tramite CRL è possibile associare una policy OCSP indicando uno dei tipi registrati nel file `<directory-lavoro>/ocsp.properties` come proprietà “`ocsp.<idPolicy>.type`”; per ulteriori dettagli si rimanda alle sezioni `ocspInstall` e `ocspConfig`.

- http-basic

(Fig. 2.60) La richiesta deve possedere un header http «Authorization» che veicola credenziali Basic (username e password) come indicato nel rfc2617#section-2 (<https://tools.ietf.org/html/rfc2617#section-2>). La presenza dell’header «Authorization Basic» nella richiesta è obbligatoria a meno che non sia abilitato il flag *Opzionale*.

Abilitando l’ulteriore opzione *Forward Authorization* è possibile propagare all’endpoint di destinazione l’header http «Authorization» che altrimenti verrà consumata.

Le credenziali devono corrispondere ad un applicativo o un soggetto registrato nel gateway (per ulteriori dettagli si rimanda alle sezioni *Creazione di un soggetto* e *Creazione di un applicativo*).

- api-key

(Fig. 2.61) La richiesta deve possedere una chiave di identificazione “Api Key” veicolata in un header http, un parametro della url o un cookie come indicato nella specifica “OAS3 API Keys” (<https://swagger.io/docs/specification/authentication/api-keys/>). È possibile abilitare anche la modalità “App ID” che prevede oltre all’`ApiKey` un identificatore dell’applicazione; modalità denominata “Multiple API Keys” nella specifica “OAS3 API Keys”. La presenza di una “Api Key”, e se attivata di una “App ID”, nella richiesta è obbligatoria a meno che non sia abilitato il flag *Opzionale*.

Autenticazione Trasporto

Stato	<input type="text" value="http-basic"/>	<input type="button" value="▼"/>
Forward Authorization	<input type="checkbox"/>	
Opzionale	<input type="checkbox"/>	

Figure2.60: Configurazione Autenticazione “http-basic”

Abilitando le ulteriori opzioni *Forward* è possibile propagare all’endpoint di destinazione la chiave di identificazione ricevuta che altrimenti verrà consumata.

Le credenziali devono corrispondere ad un applicativo o un soggetto registrato nel gateway (per ulteriori dettagli si rimanda alle sezioni *Creazione di un soggetto* e *Creazione di un applicativo*).

La configurazione consente anche di indicare dove il gateway debba ricercare la chiave di accesso tra header http, parametro della url e cookie, permettendone anche di personalizzare i nomi che per default sono quelli indicati nella specifica OAS3 (Fig. 2.62).

Autenticazione Trasporto

Stato	<input type="text" value="api-key"/>	<input type="button" value="▼"/>
App ID	<input type="checkbox"/>	
Opzionale	<input type="checkbox"/>	
Posizione	<input type="button" value="Parametro della Url
Header HTTP
Cookie"/>	
Nomi Standard OAS3	<input checked="" type="checkbox"/>	
Forward Api Key	<input type="checkbox"/>	

Figure2.61: Configurazione Autenticazione “api-key”

- principal

(Fig. 2.64) La richiesta deve possedere il «principal» che identifica il chiamante. La modalità con cui il gateway può ottenere il principale deve essere scelta tra le seguenti opzioni:

- *Container*: il principal viene fornito direttamente dal container sul quale è in esecuzione il gateway (per maggiori dettagli si faccia riferimento alla sezione *Autenticazione e Autorizzazione Principal (Security Constraint)*).
- *Header HTTP*: il principal viene estratto dallo specifico header http che viene indicato successivamente. È inoltre possibile attivare l’opzione *Forward Header* per far sì che il gateway propaghi il dato di autenticazione.
- *Parametro della Url*: il principal viene estratto da un parametro della query string il cui nome viene indicato successivamente. È inoltre possibile attivare l’opzione *Forward Parametro Url* per far sì che il gateway propaghi

Autenticazione Trasporto

Stato	<input type="text" value="api-key"/>
App ID	<input checked="" type="checkbox"/>
Opzionale	<input type="checkbox"/>
Posizione	<input type="text" value="Parametro della Url
Header HTTP
Cookie"/>
Nomi Standard OAS3	<input type="checkbox"/>

Api Key

Forward	<input type="checkbox"/>
Parametro della Url *	<input type="text" value="api_key"/>
Header HTTP *	<input type="text" value="X-API-KEY"/>

App ID

Forward	<input type="checkbox"/>
Parametro della Url *	<input type="text" value="app_id"/>
Header HTTP *	<input type="text" value="X-APP-ID"/>

Figure2.62: Configurazione Autenticazione “api-key” con personalizzazione della posizione e dei nomi

The screenshot shows a configuration interface for 'Autenticazione Trasporto'. The 'Tipo' field is set to 'Header HTTP'. A dropdown menu is open, showing several options: 'Container', 'Header HTTP' (which is highlighted in orange), 'Parametro della Url', 'Url di Invocazione', 'Client IP', 'X-Forwarded-For', and 'Token'. The 'Header HTTP' option is currently selected.

Figure2.63: Configurazione Tipo di Autenticazione “principal”

il dato di autenticazione.

- *Url di Invocazione*: il principal viene estratto direttamente dalla URL di invocazione tramite l'espressione regolare che viene fornita successivamente (l'espressione deve avere un match con l'intera url).
- *Client IP*: il principal utilizzato è l'indirizzo IP di provenienza.
- *X-Forwarded-For*: il principal viene estratto dall'header http utilizzato per il mantenimento dell'IP di origine nel caso di nodi intermedi (es. X-Forwarded-For).
- *Token*: opzione presente solamente se è stata attivata, al passo precedente, l'autenticazione del token. Il principal viene letto da uno dei claim presenti nel token.

Il flag *Opzionale* consente di non rendere bloccante il superamento dell'autenticazione nel caso la richiesta non possiede il principal atteso.

Se è presente un principal, il gateway cercherà di identificare un applicativo o un soggetto a cui è stato associato il principal come credenziale di accesso (per ulteriori dettagli si rimanda alle sezioni *Creazione di un soggetto* e *Creazione di un applicativo*); l'identificazione non è obbligatoria ma nel caso avvenga con successo l'applicativo o il soggetto verrà registrato nei log e potrà essere utilizzato anche ai fini di autorizzazione puntuale e per ruoli (*Autorizzazione*).

- plugin

Metodo di autenticazione fornito tramite personalizzazioni di GovWay.

2.10.3 Identificazione Attributi

Questa sezione consente di abilitare l'interrogazione di una o più *Attribute Authority* al fine di recuperare gli attributi qualificati del soggetto identificato su GovWay tramite i meccanismi di autenticazione precedentemente descritti nelle sezioni “*Autenticazione Token*” e “*Autenticazione Trasporto*”.

Nota: La sezione viene visualizzata solamente se è stata registrata almeno una *Attribute Authority*.

Gli attributi recuperati saranno inseriti nel contesto della richiesta e potranno essere utilizzati per definire politiche di controllo degli accessi basate sugli attributi tramite i meccanismi di autorizzazione descritti nelle successive sezioni (“*XACML-Policy*”, “*Autorizzazione per Token Claims*” e “*Autorizzazione Contenuti*”).

Autenticazione Trasporto

Stato	principal
Tipo	Header HTTP
Nome *	X-Principal
Forward Header	<input type="checkbox"/>
Opzionale	<input type="checkbox"/>

Figure2.64: Configurazione Autenticazione “principal”

Il form di configurazione appare come quello illustrato in Fig. 2.65.

Identificazione Attributi

Stato	abilitato
Attribute Authority *	<div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> AA1 AA2 AA3 AA4 AA5 </div>
Attributi Richiesti	profilo,sesso
Elencare gli attributi da richiedere, separandoli con la virgola	

Figure2.65: Configurazione Identificazione Attributi tramite una singola AA

Una volta abilitata la funzionalità, se viene selezionata un'unica *Attribute Authority* sarà possibile indicare quali attributi qualificati debbano essere recuperati indicandoli nel campo “Attributi Richiesti”, separandoli con la virgola, come riportato nell'esempio in Fig. 2.65.

Se invece vengono selezionate molteplici AA, gli attributi da richiedere devono essere indicati tramite una riga per ogni A.A. (nomeAA=listaAttributi) come in Fig. 2.66.

^ **Identificazione Attributi**

Stato	abilitato
Attribute Authority *	AA1 AA2 AA3 AA4 AA5
Attributi Richiesti	AA2=denominazione,profilo,indirizzo AA3=sesso
Elencare gli attributi da richiedere, separandoli con la virgola, utilizzando una riga per ogni A.A. (nomeAA=listaAttributi)	

Figure2.66: Configurazione Identificazione Attributi tramite multiple AA

2.10.4 Autorizzazione

L'autorizzazione è un ulteriore meccanismo per il controllo degli accessi tramite il quale è possibile specificare con maggior dettaglio le richieste che sono in grado di essere accettate per l'accesso al servizio.

I meccanismi supportati, per specificare i criteri di autorizzazione, sono i seguenti:

- *Autorizzazione Trasporto per Richiedente*: (Fig. 2.67) superato il processo di autenticazione trasporto, saranno accettate le sole richieste provenienti dai mittenti indicati singolarmente nella lista fornita con il criterio. Dopo aver abilitato questa opzione, ed aver confermato tramite il pulsante Invia, sarà possibile fornire la lista dei mittenti autorizzati ad accedere al servizio.

I mittenti che possono essere indicati sono Soggetti (solo nel caso delle erogazioni) e Applicativi. Tali entità dovranno essere precedentemente registrate sulla govwayConsole seguendo le indicazioni fornite in sezione *Creazione di un soggetto* e *Creazione di un applicativo* e dovranno possedere credenziali di accesso compatibili con l'autenticazione di trasporto configurata.

Nota: L'opzione di autorizzazione trasporto per richiedente è disponibile solo se è stata attivata l'autenticazione trasporto.

- *Autorizzazione Trasporto per Ruoli*: (Fig. 2.68) consente di concedere l'autorizzazione per il servizio solo ai richiedenti in possesso di determinati ruoli nel proprio profilo. Dopo aver barrato questa opzione, ed aver confermato tramite il pulsante Invia, sarà possibile fornire una lista dei ruoli che devono essere posseduti dal chiamante per poter accedere al servizio. In particolare si dovrà anche specificare la *fonte* di provenienza dei ruoli, che può essere *esterna*, cioè proveniente dal sistema che ha autenticato il chiamante, oppure *registro*, cioè i ruoli che sono stati censiti nel registro di GovWay e assegnati all'applicativo o al soggetto chiamante identificato tramite l'autenticazione trasporto. Inoltre si deve scegliere l'opzione *Ruoli Richiesti* per indicare se, in presenza di più di un ruolo come criterio, il chiamante deve possedere «tutti» i ruoli indicati o «almeno uno».

Per le indicazioni sul censimento dei ruoli fare riferimento alla sezione *Creazione di un ruolo*.

- *Autorizzazione Trasporto per Richiedente e per Ruoli*: abilitando entrambi i criteri autorizzativi descritti precedentemente, il chiamante verrà autorizzato se soddisfa almeno uno dei due criteri.
- *Autorizzazione Token per Richiedente*: (Fig. 2.69) superato il processo di autenticazione token, saranno accettate

^ Autorizzazione

Stato	abilitato	▼
-------	-----------	---

Autorizzazione Trasporto

per Richiedente

Soggetti (0)

Applicativi (2)

per Ruoli

Figure2.67: Configurazione Autorizzazione Trasporto per Richiedente

^ Autorizzazione

Stato	abilitato	▼
-------	-----------	---

Autorizzazione Trasporto

per Richiedente

per Ruoli

Fonte

Ruoli Richiesti

Ruoli (2)

Figure2.68: Configurazione Autorizzazione Trasporto per Ruoli

le sole richieste provenienti dai mittenti indicati singolarmente nella lista fornita con il criterio. Dopo aver abilitato questa opzione, ed aver confermato tramite il pulsante Invia, sarà possibile fornire la lista dei mittenti autorizzati ad accedere al servizio.

I mittenti che possono essere indicati sono Applicativi. Tali entità dovranno essere precedentemente registrate sulla govwayConsole seguendo le indicazioni fornite in [Creazione di un applicativo](#) e dovranno possedere credenziali di accesso di tipo “token”.

Nota: L'opzione di autorizzazione basata sui token è disponibile solo se è stata preventivamente attivata la Gestione Token e selezionata la relativa policy.

^ Autorizzazione

Stato: abilitato

Autorizzazione Trasporto

per Richiedente

per Ruoli

Autorizzazione Token

per Richiedente

Applicativi (1)

per Ruoli

Figure2.69: Configurazione Autorizzazione Token per Richiedente

- **Autorizzazione Token per Ruoli:** (Fig. 2.70) simile all'autorizzazione trasporto per ruoli, si differenzia nel fatto che i ruoli devono essere posseduti dagli applicativi censiti nel registro di GovWay tramite credenziali di tipo *token* (*fonte* di provenienza dei ruoli impostata a *registro*) o devono essere presenti all'interno del token ricevuto (*fonte* di provenienza dei ruoli impostata a *esterna*). Come per l'autorizzazione per trasporto è possibile scegliere l'opzione *Ruoli Richiesti* per indicare se, in presenza di più di un ruolo come criterio, il chiamante deve possedere «tutti» i ruoli indicati o «almeno uno».

Per le indicazioni sul censimento dei ruoli fare riferimento alla sezione [Creazione di un ruolo](#).

Nota: L'opzione di autorizzazione basata sui token è disponibile solo se è stata preventivamente attivata la Gestione Token e selezionata la relativa policy.

- **Autorizzazione Token per Richiedente e per Ruoli:** abilitando entrambi i criteri autorizzativi descritti precedentemente, il chiamante verrà autorizzato se soddisfa almeno uno dei due criteri.
- **Autorizzazione per Token Claims:** (Fig. 2.71) Se è stata abilitata la gestione del token si ha la possibilità di autorizzare le richieste inserendo i valori ammessi per i claims contenuti nel token. La configurazione viene effettuata inserendo nel campo di testo ciascun claim in una riga, facendo seguire dopo l'uguale i valori ammessi separati da virgola.

^ Autorizzazione

Stato	abilitato	<input type="button" value="▼"/>
-------	-----------	----------------------------------

Autorizzazione Trasporto

per Richiedente	<input type="checkbox"/>
per Ruoli	<input type="checkbox"/>

Autorizzazione Token

per Richiedente	<input type="checkbox"/>	
per Ruoli	<input checked="" type="checkbox"/>	
Fonte	Qualsiasi	<input type="button" value="▼"/>
Ruoli Richiesti	tutti	<input type="button" value="▼"/>

Ruoli (1)

Figure2.70: Configurazione Autorizzazione Token per Ruoli

Per le indicazioni di dettaglio sui possibili controlli effettuabili su ogni claim si faccia riferimento alla sezione [Autorizzazione per Token Claims](#).

Nota: L'opzione di autorizzazione basata sui token è disponibile solo se è stata preventivamente attivata la Gestione Token e selezionata la relativa policy.

- *Autorizzazione per Token Scope:* (Fig. 2.72) criterio di autorizzazione che verifica la corrispondenza tra gli scope indicati e quelli estratti dal token presente nella richiesta ricevuta. Una volta attivata l'opzione si deve effettuare una scelta per l'elemento *Scope Richiesti*, tra i valori «tutti» (tutti gli scope indicati devono essere presenti nel token per superare l'autorizzazione) e «almeno uno» (è richiesta la presenza di almeno uno scope tra quelli indicati nella policy di autorizzazione). Dopo aver confermato la scelta con il pulsante «Invia» verrà richiesto di inserire gli scope tra quelli già censiti ed abilitati per l'uso nei contesti di erogazione (o qualsiasi contesto).

Per le indicazioni sul censimento degli scope fare riferimento alla sezione [Scope](#).

Nota: L'opzione di autorizzazione basata sugli scope è disponibile solo se è stata preventivamente attivata la Gestione Token e selezionata la relativa policy.

- *XACML-Policy:* (Fig. 2.73) È possibile basare il meccanismo di autorizzazione sulla valutazione di una policy xacml selezionando la relativa opzione sulla lista «Stato».

Per le indicazioni di dettaglio sulla configurazione delle xacml-Policy si faccia riferimento alla sezione [XACML-Policy](#).

- *Custom:* Sulla lista «Stato», è possibile selezionare questo metodo di autorizzazione eventualmente fornito

^ Autorizzazione

Stato	<input type="text" value="abilitato"/> ▼
-------	--

Autorizzazione Trasporto

per Richiedente	<input type="checkbox"/>
per Ruoli	<input type="checkbox"/>

Autorizzazione Token

per Richiedente	<input type="checkbox"/>
per Ruoli	<input type="checkbox"/>

Autorizzazione per Token Claims

Abilitato	<input checked="" type="checkbox"/>
Claims	<input type="text" value="aud=AppTest
sub=user1,user2,user3"/> i

Indicare per riga i claims richiesti (nome=valore); visualizzare 'info' per maggiori dettagli

Autorizzazione per Token Scope

Abilitato	<input type="checkbox"/>
-----------	--------------------------

Figure2.71: Configurazione Autorizzazione per Token Claims

Autorizzazione

Stato

Autorizzazione Trasporto

per Richiedente

per Ruoli

Autorizzazione Token

per Richiedente

per Ruoli

Autorizzazione per Token Claims

Abilitato

Autorizzazione per Token Scope

Abilitato

Scope Richiesti

[Scope \(1\)](#)

Figure2.72: Configurazione Autorizzazione per Token Scope

Autorizzazione

Stato

Fonte Ruoli

Policy No file chosen
SAMLPolicy.xml

Figure2.73: Configurazione Autorizzazione XACML-Policy

tramite estensione di GovWay.

2.10.5 Autorizzazione Contenuti

L'autorizzazione dei contenuti è un ulteriore meccanismo per il controllo degli accessi tramite il quale è possibile specificare regole di autorizzazione che verificano aspetti della richiesta quali ad esempio gli header http, l'url di invocazione, parti del messaggio etc.

Una volta abilitata l'autorizzazione per contenuto si possono configuare una serie di controlli di autorizzazione nella forma (risorsa=valore).

Una risorsa identifica un header, una parte dell'url o del messaggio, un claim del token o un principal etc. Per identificare una risorsa sono utilizzabili le seguenti espressioni dinamiche:

- \${header:NAME}: valore presente nell'header http che possiede il nome “NAME”
- \${query:NAME}: valore associato al parametro della url con nome “NAME”
- \${urlRegExp:EXPR}: espressione regolare applicata sulla url di invocazione (l'espressione deve avere un match con l'intera url)
- \${xpath:EXPR}: espressione XPath applicata su un messaggio XML
- \${jsonPath:EXPR}: espressione JSONPath applicata su un messaggio JSON
- \${tokenInfo:FIELD}: permette di accedere ai claim di un token; il valore “FIELD” fornito deve rappresentare un field valido all'interno della classe “org.openscoop2.pdd.core.token.InformazioniToken” (es. per ottenere il valore del claim “sub” usare \${tokenInfo:sub})
- \${tokenClient:FIELD}: identità dell'applicativo client identificato tramite il clientId presente nel token; il valore “FIELD” fornito deve rappresentare un field valido all'interno della classe “org.openscoop2.core.id.IDServizioApplicativo” (es. per ottenere il nome dell'applicativo usare \${tokenClient:nome})
- \${aa:FIELD} : permette di accedere agli attributi recuperati tramite Attribute Authority; il valore “FIELD” fornito deve rappresentare un field valido all'interno della classe “org.openscoop2.pdd.core.token.attribute_authority.InformazioniAttributi” (es. per ottenere il valore dell'attributo “attr1” usare \${aa:attributes[attr1]}, se configurata solamente 1 A.A., altrimenti usare \${aa:attributes[nomeAttributeAuthority][attr1]})
- \${transportContext:FIELD}: permette di accedere ai dati della richiesta http; il valore “FIELD” fornito deve rappresentare un field valido all'interno della classe “org.openscoop2.utils.transport.http.HttpServletTransportRequestContext” (es. per il principal usare \${transportContext:credential.principal})
- \${busta:FIELD}: permette di utilizzare informazioni generiche del profilo; il valore “FIELD” fornito deve rappresentare un field valido all'interno della classe “org.openscoop2.protocol.sdk.Busta” (es. per il mittente usare \${busta:mittente})
- \${property:NAME}: utilizzabile solamente su erogazioni, permette di riferire informazioni specifiche del profilo presenti nella traccia (es. identificativo SDI). Il valore “NAME” indica il nome della proprietà da utilizzare
- \${securityToken:FIELD}: permette di accedere ai certificati e ai security token presenti nella richiesta; il valore “FIELD” fornito deve rappresentare un field valido all'interno della classe “org.openscoop2.protocol.sdk.SecurityToken” (es. per accedere al CN del certificato presente nel token “Authorization” usare \${securityToken:authorization.certificate.subject.info(CN)})
- \${config:NAME}: valore della proprietà configurata sull'API che possiede il nome “NAME”
- \${clientApplicationConfig:NAME}: valore della proprietà configurata nell'applicativo fruitore che possiede il nome “NAME”

- \${clientOrganizationConfig:NAME}: valore della proprietà configurata nel soggetto fruitore che possiede il nome “NAME”
- \${providerOrganizationConfig:NAME}: valore della proprietà configurata nel soggetto erogatore che possiede il nome “NAME”
- \${tokenClientApplicationConfig:NAME}: permette di accedere alla proprietà, configurata nell’applicativo client identificato tramite il clientId presente nel token, con nome “NAME”
- \${tokenClientOrganizationConfig:NAME}: permette di accedere alla proprietà, configurata nel soggetto proprietario dell’applicativo client identificato tramite il clientId presente nel token, con nome “NAME”
- \${dynamicConfig:FIELD}: permette di accedere alle proprietà degli attori coinvolti nella richiesta (api, applicativi, soggetti); il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.pdd.core.dynamic.DynamicConfig”; maggiori informazioni sulla funzionalità sono disponibili nella sezione “[Accesso alle proprietà delle entità del Registro](#)”.
- \${system:NAME}: valore associato alla proprietà di sistema, indicata nella configurazione generale, con nome “NAME”
- \${env:NAME}: valore associato alla variabile di sistema con nome “NAME”
- \${java:NAME}: valore associato alla variabile java con nome “NAME”
- \${envj:NAME}: valore associato alla variabile di sistema o java con nome “NAME”; la variabile viene cercata prima come variabile di sistema e se non presente come variabile della jvm

Ogni valore atteso per una risorsa può essere fornito in una delle seguenti modalità:

- \${anyValue} : indica qualsiasi valore non nullo.
- \${undefined} : la risorsa indicata non deve esistere o non deve essere valorizzata.
- \${regExpMatch:EXPR} : la regola è soddisfatta se il valore della risorsa ha un match completo rispetto all’espressione regolare EXPR indicata. È possibile utilizzare anche la versione \${regExpNotMatch:EXPR} che consente di attuare una negazione della condizione.
- \${regExpFind:EXPR} : simile alla precedente regola, il match dell’espressione regolare può avvenire anche su una sottostringa del valore della risorsa. Come per la precedente esiste anche la versione \${regExpNotFind:EXPR}.
- valore : indica esattamente il valore (case sensitive) che deve possedere la risorsa; il valore può essere definito come costante o contenere parti dinamiche risolte a runtime dal Gateway nella forma descritta precedentemente per le risorse.
- valore1,...,valoreN : è possibile elencare differenti valori ammissibili; come per la precedente opzione il valore può contenere parti dinamiche.
- \${ignoreCase:valore} o \${ignoreCase:valore1,...,valoreN} : simile alle precedenti regole consente di attuare una verifica case insensitive.
- \${not:valore} o \${not:valore1,...,valoreN} : simile alle precedenti regole consente di indicare esattamente i valori (case sensitive) che non deve possedere la risorsa. È possibile utilizzarla anche in combinazione con il controllo case-insensitive: \${not:\${ignoreCase:valore}} o \${not:\${ignoreCase:valore1,...,valoreN}}

Di seguito alcuni esempi:

- \${header:X-Prova}=test : viene verificato che l’header “X-Prova” possiede il valore “test”
- \${header:X-Prova}=test,test2,test3 : viene verificato che l’header “X-Prova” possiede il valore “test” o “test2” o “test3”
- \${transportContext:credential.principal}=\${header:X-SSO} : viene verificato che l’identità principal del chiamante corrisponda al valore fornito nell’header “X-SSO”

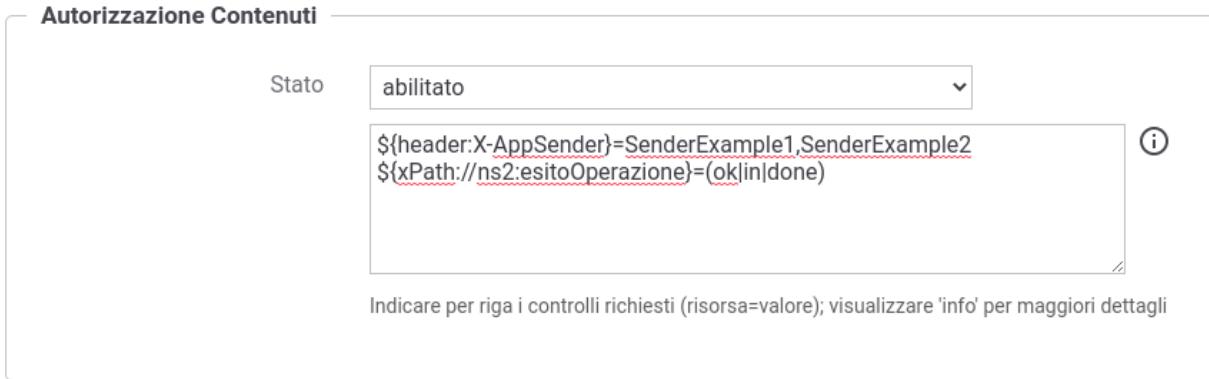


Figure2.74: Configurazione Autorizzazione Contenuti

- \${transportContext:credential.principal}=prefix\${header:X-SSO}suffix : simile alla precedente regola, dove l'identità principal viene confrontata con il valore presente nell'header concatenato da un prefisso e da un suffisso statico.
- \${xPath:EXPR}=\${regExpMatch:[0-9]} : viene estratto il contenuto dalla richiesta xml tramite l'espressione XPath EXPR e verificato che sia corrispondente ad una cifra decimale attraverso l'espressione regolare “[0-9]”
- \${jsonPath:EXPR}=\${transportContext:credential.principal} : viene estratto il contenuto dalla richiesta json tramite l'espressione jsonPath EXPR e verificato che sia uguale all'identità principal del chiamante
- \${context:CLIENT_IP_REMOTE_ADDRESS}=10.114.44.3,10.114.44.4,10.114.44.5 : viene verificato che l'indirizzo ip del client sia tra gli indirizzi ip elencati.
- \${context:CLIENT_IP_TRANSPORT_ADDRESS}=\${regExpMatch:10.114.44..*|10.114.43..*} : viene verificato che l'indirizzo ip del client sia nella sottorete 10.114.44.0/255 o 10.114.43.0/255; l'indirizzo ip viene estratto dagli header http utilizzati per il mantenimento dell'IP di origine nel caso di nodi intermedi (es. X-Forwarded-For).
- \${transportContext:credential.certificateChain.certificate.subject.info(CN)}=EsempioEnte1,EsempioEnte2 : viene verificato che il CN del certificato TLS client corrisponda a uno dei due valori tra EsempioEnte1 e EsempioEnte2.
- \${securityToken:integrity.certificate.subject.info(ORGANIZATION_IDENTIFIER)}=\${regExpMatch:CF:IT-.+} : il valore del campo “Organization Identifier” (2.5.4.97) del DN del certificato presente nel token di sicurezza ModI “Agid-JWT-Signature” deve iniziare onc “CF:IT-“.
- \${securityToken:channel.certificate.hasExtendedKeyUsage(CLIENT_AUTH)}=true : il certificato TLS client deve possedere il purpose (ExtendedKeyUsage) “client auth” (1.3.6.1.5.5.7.3.2).
- \${securityToken:authorization.certificate.hasKeyUsage(DIGITAL_SIGNATURE)}=true : il certificato presente nel token di sicurezza ModI “Authorization” deve possedere la key usage per la firma digitale.

2.10.6 Creazione di un soggetto

Affinché possano essere utilizzate le funzionalità di autenticazione ed autorizzazione, associate alle erogazioni, è necessario che vengano censiti i soggetti fruitori che inviano le richieste di servizio. La registrazione di un soggetto consente di assegnargli delle credenziali che lo identificano ed eventuali ruoli provenienti dalla fonte «Registro».

Soggetto

Dominio	Esterno
Profilo Interoperabilità	API Gateway
Nome *	MinisteroEsempio
Tipologia	Fruitore
Descrizione	

Modalità di Accesso

Tipo	https
Configurazione	
Modalità	Upload Archivio
Formato	PKCS12
Password *	123456
Archivio *	Choose File pa.p12

Figure2.75: Creazione di un soggetto

Per creare il soggetto posizionarsi nella sezione *Registro > Soggetti*, quindi premere il pulsante *Aggiungi*. Compilare il form come segue (Fig. 2.75):

- **Profilo Interoperabilità:** La scelta del profilo di interoperabilità sarà richiesta solo nel caso in cui non sia stata effettuata la relativa scelta dal menu in testata.
- **Nome:** Il nome del soggetto. È necessario che il nome indicato risulti univoco rispetto ai nomi già presenti per la modalità operativa selezionata (in questo caso API Gateway).

- *Tipologia*: Indicare se si tratta di un soggetto esclusivamente erogatore, esclusivamente fruitore o con entrambi i ruoli.
- *Descrizione*: Un testo di descrizione per il soggetto.
- *Modalità di Accesso*: Sezione presente solo nel caso in cui il soggetto ricopra il ruolo di fruitore. Tramite il campo *Tipo* si seleziona il tipo di credenziali richieste per l'autenticazione del soggetto. In base alla scelta effettuata saranno mostrati i campi per consentire l'inserimento delle credenziali richieste. Per i dettagli sulla configurazione della modalità di accesso si faccia riferimento alla sezione *Modalità di Accesso*.

Dopo aver creato il soggetto è opzionalmente possibile assegnargli dei ruoli, tra quelli che sono presenti nel registro e contrassegnati come *fonte registro*. Per associare ruoli ad un soggetto seguire il collegamento presente nella colonna *Ruoli*, in corrispondenza del soggetto scelto. Premere quindi il pulsante *Aggiungi*. Nel form che si apre (Fig. 2.76) è presente una lista dalla quale è possibile selezionare un ruolo alla volta, che viene aggiunto confermando con il tasto *Invia*.

Ruolo	
Nome	Ticket

Invia Cancella

Figure2.76: Assegnazione di ruoli ad un soggetto

2.10.7 Creazione di un applicativo

Affinché possano essere utilizzate le funzionalità di autenticazione ed autorizzazione, associate alle fruizioni, è necessario che vengano censiti gli applicativi client, interni al dominio, che inviano le richieste di servizio. La registrazione di un applicativo, di tipo client, consente di assegnargli delle credenziali che lo identificano ed eventuali ruoli provenienti dalla fonte «Registro».

Per registrare l'applicativo posizionarsi nella sezione *Registro > Applicativi*, quindi premere il pulsante *Aggiungi*. Compilare il form come segue (Fig. 2.77):

- *Profilo Interoperabilità*: Opzione visibile solo nel caso in cui non sia stata effettuata la relativa scelta sul menu della testata.
- *Nome*: Assegnare un nome all'applicativo. È necessario che il nome indicato risulti univoco rispetto ai nomi già presenti per la modalità operativa selezionata (in questo caso API Gateway).
- *Tipo*: Utilizzare il tipo “Client” per censire un'applicativo allo scopo di identificarlo ed autorizzarlo durante l'invocazione di erogazioni o fruizioni di API.
- *Modalità di Accesso*: Tramite il campo *Tipo* si seleziona il tipo di credenziali richieste per l'autenticazione dell'applicativo. In base alla scelta effettuata saranno mostrati i campi per consentire l'inserimento delle credenziali richieste. Per i dettagli sulla configurazione della modalità di accesso si faccia riferimento alla sezione *Modalità di Accesso*.

Applicativi > Aggiungi

Note: (*) Campi obbligatori

Applicativo

Nome *

Tipo

Modalità di Accesso

Tipo

Utente *

Password *

SALVA

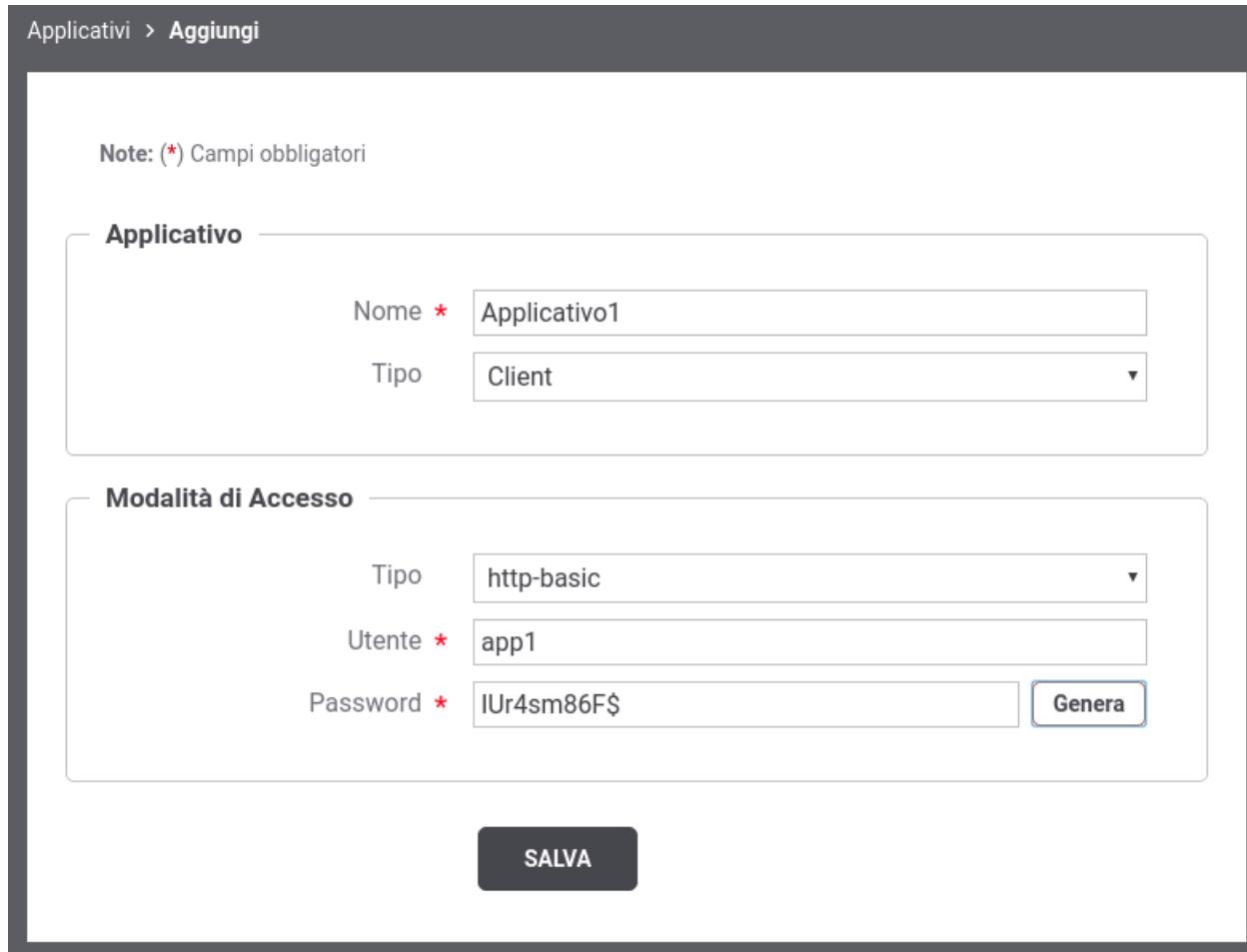


Figure2.77: Creazione di un applicativo

Dopo aver creato l'applicativo è opzionalmente possibile assegnargli dei ruoli, tra quelli che sono presenti nel registro e contrassegnati come *fonte registro*. Per associare ruoli ad un applicativo seguire il collegamento presente nella colonna *Ruoli*, in corrispondenza dell'applicativo scelto. Premere quindi il pulsante *Aggiungi*. Nel form che si apre (Fig. 2.78) è presente una lista dalla quale è possibile selezionare un ruolo alla volta, che viene aggiunto confermando con il tasto *Invia*.

The screenshot shows a user interface for adding a role to an application. The path in the top navigation bar is: Applicativi > AnagraficaResidentiANPR > Ruoli > Aggiungi. The main form is titled 'Ruolo' and contains a single input field labeled 'Nome' with the value 'RuoloDemo'. Below the input field is a 'SALVA' (Save) button.

Figure 2.78: Assegnazione di ruoli ad un applicativo

2.10.8 Modalità di Accesso

Agli applicativi ed ai soggetti registrati nel gateway (come indicato nelle sezioni *Creazione di un soggetto* e *Creazione di un applicativo*) devono essere assegnate delle credenziali in modo che il gateway possa effettuare:

- *autenticazione trasporto*: nel caso in cui il controllo degli accessi sia stato configurato con *Autenticazione Trasporto* “http-basic” o “api-key”
- *identificazione*: l'identificazione non è obbligatoria per le autenticazioni trasporto differenti da “http-basic” e “api-key” o per l'*Autenticazione Token*, ma nel caso avvenga con successo l'applicativo o il soggetto verrà registrato nei log e potrà essere ricercato tramite gli strumenti di monitoraggio
- *autorizzazione*: se un applicativo o un soggetto viene identificato, può essere autorizzato puntualmente nel controllo degli accessi tramite l'autorizzazione (*Autorizzazione*).

Come mostrato in Fig. 2.79 l'assegnazione delle credenziali deve essere effettuata attraverso la selezione di un tipo di credenziali tra quelli disponibili:

- *https*: richiede la registrazione di un certificato client X509
- *http-basic*: deve essere definito un username univoco e deve essere generata una password
- *apikey*: richiede la generazione di una chiave di identificazione univoca
- *principal*: deve essere assegnato un identificatore univoco
- *token*: deve essere selezionata una *Token Policy Validazione* e deve essere assegnato un identificatore univoco

Nota: Il tipo di credenziale *token* è associabile solamente agli applicativi e consente durante il processo di *Autenticazione Token* di identificare un applicativo tramite il claim “clientId” presente all'interno del token.

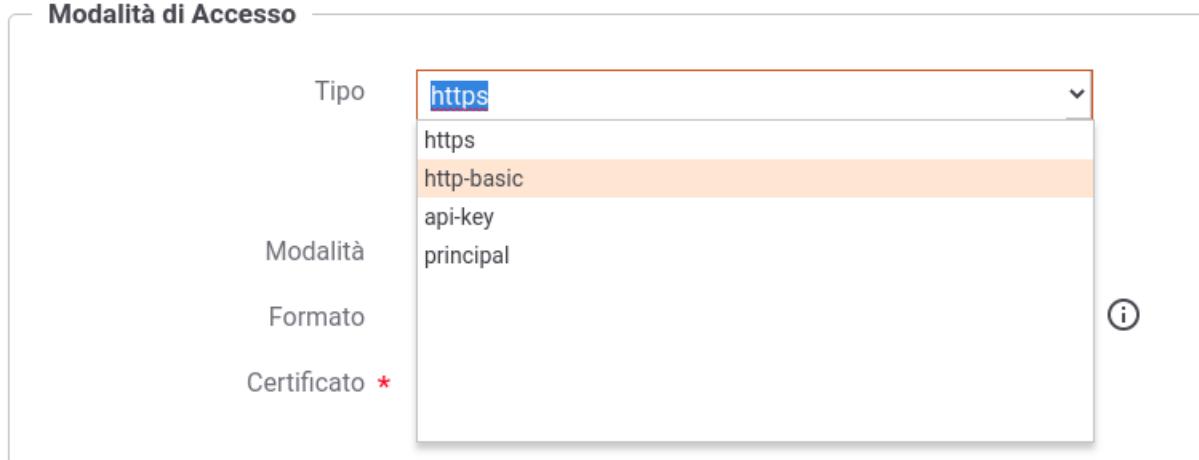


Figure2.79: Configurazione della Modalità di Accesso

Credenziali “https”

Agli applicativi ed ai soggetti registrati nel gateway, identificabili con credenziali “https”, deve essere associato un certificato client X509.

Per la configurazione si procede selezionando dall’elemento *Modalità* una tra le seguenti opzioni:

- **Upload Archivio** (Fig. 2.80): con questa modalità di configurazione si procede con il caricamento del certificato che sarà utilizzato per l’autenticazione. È necessario indicare:
 - *Formato*: il formato del certificato fornito specificando tra le seguenti opzioni supportate:
 - * *CER*: il certificato da caricare è in formato *DER* o *PEM*.
 - * *JKS*: il certificato da caricare è contenuto in un keystore *JKS*.
 - * *PKCS12*: il certificato da caricare è contenuto in un keystore *PKCS12*.
 - *Password*: campo visibile nel caso in cui il certificato da caricare è contenuto in un keystore *JKS* o *PKCS12*. Rappresenta la password per l’accesso al keystore.
 - *Archivio*: selezionare dal proprio filesystem il file che contiene il certificato.
 - *Alias*: nel caso in cui il keystore contenga più di un certificato (frequente in formati *JKS*), questa lista consente di selezionare l’alias che riferisce l’elemento corretto.

Una volta caricato l’archivio verranno mostrati a video i dettagli del certificato selezionato (Fig. 2.81), al fine di poterli verificare prima di confermare l’inserimento. Il certificato caricato verrà confrontato con il certificato fornito durante l’autenticazione se la voce *Verifica* è abilitata, altrimenti verranno controllati solamente che i DN del Subject e dell’Issuer siano identici. Un confronto fallito causeranno il fallimento dell’autenticazione.

Dopo aver creato un applicativo o un soggetto con associato un certificato, visualizzandone i dati è possibile effettuare il download del certificato o aggiungerne di ulteriori.

La funzionalità di aggiunta di un certificato può essere utilizzata per gestire preventivamente la scadenza di un certificato caricando anche la versione aggiornata in modo da poter essere in grado di autenticare l’applicativo non appena inizia ad utilizzare il nuovo certificato. Sia in fase di aggiunta che successivamente sarà possibile promuovere a “principale” la versione aggiornata del certificato ed eliminare successivamente la versione scaduta.

Modalità di Accesso

Tipo	<input type="text" value="https"/>
Configurazione	
Modalità	<input type="text" value="Upload Archivio"/>
Formato	<input type="text" value="PKCS12"/>
Password *	<input type="text" value="123456"/>
Archivio *	<input type="button" value="Choose File"/> ExampleClient2.p12

CARICA CERTIFICATO

Figure2.80: Credenziali di tipo HTTPS (upload archivio 1/2)

- **Configurazione Manuale** (Fig. 2.84): con questa modalità di configurazione si procede con l'inserimento dei seguenti dati:
 - *Self Signed*: opzione per indicare se il certificato è self-signed oppure rilasciato da una CA.
 - *Subject*: il subject del certificato.
 - *Issuer*: l'issuer del certificato, nel caso in cui non sia self-signed.

Credenziali “http-basic”

Agli applicativi ed ai soggetti registrati nel gateway, identificabili con credenziali “http-basic”, deve essere associato un identificativo utente univoco e una password (Fig. 2.85). La password può essere generata tramite l'apposito pulsante.

Nota: La password generata e assegnata all'applicativo o al soggetto viene visualizzata solamente nell'avviso visualizzato in seguito alla creazione (Fig. 2.86) e successivamente non è più consultabile.

Modalità di Accesso

Tipo	<input type="text" value="https"/> 
Configurazione	
<i>ExampleClient1.crt</i>	
Cambia Certificato	
Certificato X.509 v3	
Verifica	<input checked="" type="checkbox"/>
Subject	<input type="text" value="/l=Pisa/st=Italy/o=Example/c=IT/cn=ExampleClient1/"/>
Issuer	<input type="text" value="/l=Pisa/st=Italy/o=Example/c=IT/cn=ExampleCA/"/>
Serial Number	2
Self Signed	No
Not Before	09/07/2019 12:26:00
Not After	30/07/2040 12:26:00

Figure2.81: Credenziali di tipo HTTPS (upload archivio 2/2)

Modalità di Accesso

Tipo	<input type="text" value="https"/>
------	------------------------------------

Configurazione

[Elenco Certificati \(2\)](#)

Certificato X.509 v1

[Download](#)

Verifica	<input checked="" type="checkbox"/>
Subject	/l=Pisa/st=Italy/ou=test/emailaddress=apoli@link.it/o=openspcoop.org/c=IT/cn=sil1/
Issuer	/l=Pisa/st=Italy/ou=test/emailaddress=apoli@link.it/o=openspcoop.org/c=IT/cn=sil1/
Serial Number	1318427594
Self Signed	Si
Not Before	12/10/2011 15:53:00
Not After	09/10/2021 15:53:00

Figure2.82: Credenziali di tipo HTTPS (consultazione)

Certificati							
	Principale	Subject	Issuer	Verifica	Not Before	Not After	
<input type="checkbox"/>	Si	/l=Pisa/st=Italy/ou=test/emailaddress=apoli@link.it/o=openspcoop.org/c=IT/cn=sil1/	/l=Pisa/st=Italy/ou=test/emailaddress=apoli@link.it/o=openspcoop.org/c=IT/cn=sil1/	Certificato	12/10/2011 15:53:00	09/10/2021 15:53:00	
<input type="checkbox"/>	No	/l=Pisa/st=Italy/o=Example/c=IT/cn=ExampleClientScaduto/	/l=Pisa/st=Italy/o=Example/c=IT/cn=ExampleCA/	Certificato	09/07/2019 12:29:00	09/07/2019 12:30:00	

Figure2.83: Credenziali di tipo HTTPS (certificati aggiuntivi)

Modalità di Accesso

Tipo	<input type="text" value="https"/>
Configurazione	
Modalità	<input type="text" value="Configurazione Manuale"/>
Self Signed	<input type="checkbox"/>
Subject *	<input type="text" value="cn=ExampleClient2, l=Pisa, st=Italy, o=Example, c=IT"/>
Issuer	<input type="text"/>

Figure2.84: Credenziali di tipo HTTPS (configurazione manuale)

Modalità di Accesso

Tipo	<input type="text" value="http-basic"/>
Utente *	<input type="text" value="utenteTest"/>
Password *	<input type="text" value="7\$94EYrdnleM3Exd6mq8"/> <input type="button" value="Genera"/>

Figure2.85: Credenziali di tipo HTTP-Basic

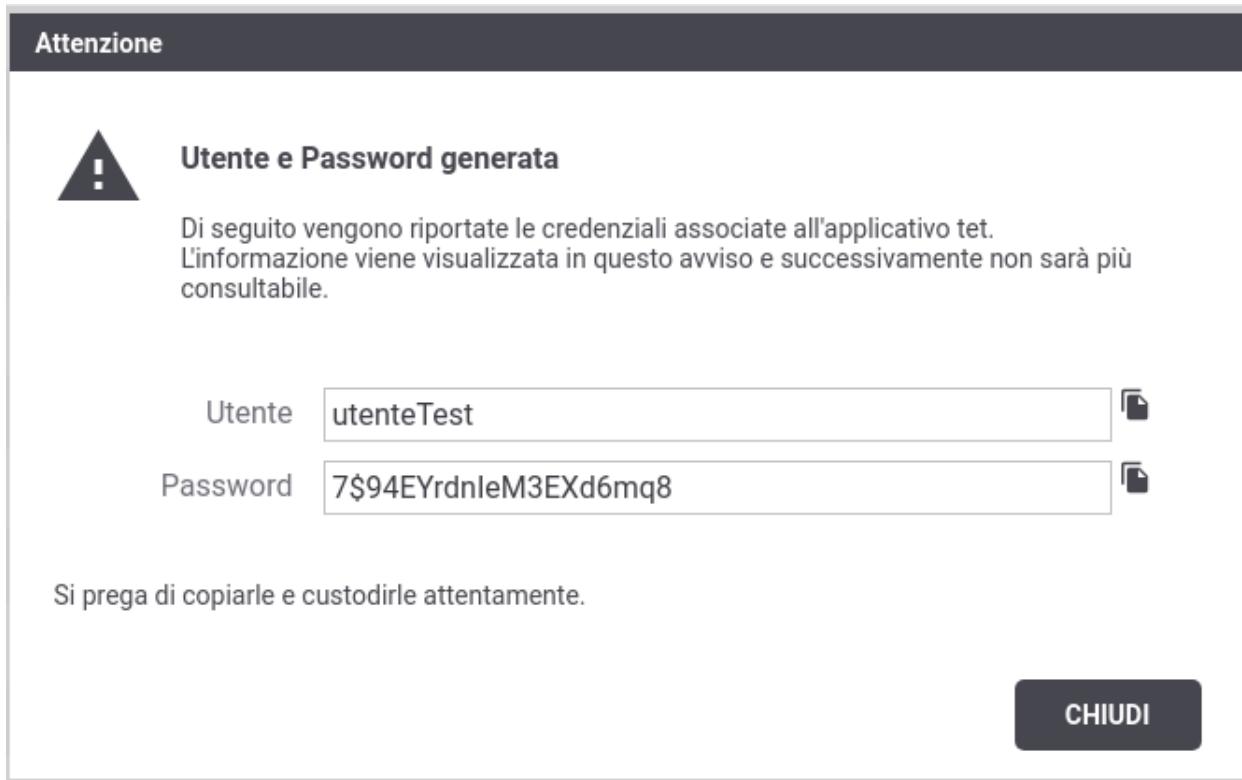


Figure2.86: Avviso di copia delle credenziali HTTP-Basic

Nel caso di smarrimento della password è necessario procedere con la generazione di una nuova password (Fig. 2.87).

Modalità di Accesso

Tipo

Utente *

Modifica Password

Nuova Password * **Genera**

Figure2.87: Aggiornamento delle credenziali HTTP-Basic

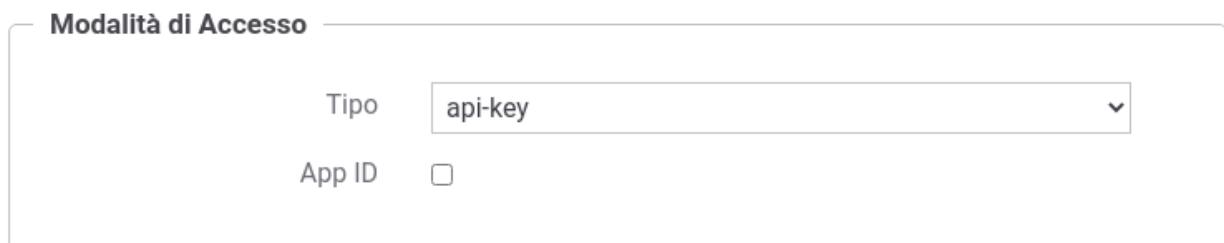
Credenziali “api-key”

Agli applicativi ed ai soggetti registrati nel gateway, identificabili con credenziali “api-key” deve essere associato una chiave di identificazione univoca “Api Key” come descritto nella specifica “OAS3 API Keys” (<https://swagger.io/docs/specification/authentication/api-keys/>).

La credenziale può inoltre essere composta da un’ulteriore informazione riguardante l’identificatore dell’applicativo “App ID”; modalità denominata “Multiple API Keys” nella specifica “OAS3 API Keys”.

L’associazione di credenziali “api-key” ad un applicativo o soggetto comporta solamente l’indicazione se deve essere generato anche un “App ID” o meno (Fig. 2.88).

La generazione della “Api Key” e dell’eventuale “App ID” è automatica e viene visualizzata non appena si completa la registrazione dell’applicativo o del soggetto. Nella figura Fig. 2.89 viene riportato un avviso di generazione di una credenziale senza “App ID”, mentre nella figura Fig. 2.90 è stato generato anche l’identificatore dell’applicativo.



Modalità di Accesso

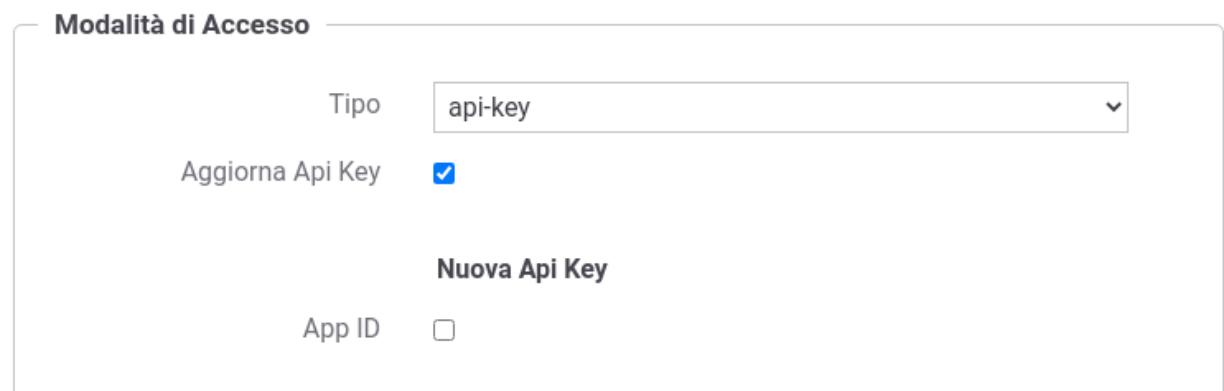
Typo: api-key

App ID:

Figure2.88: Credenziali “api-key”

Nota: La chiave di identificazione generata e assegnata all’applicativo o al soggetto viene visualizzata solamente nell’avviso visualizzato in seguito alla creazione (Fig. 2.89) e successivamente non è più consultabile.

Nel caso di smarrimento della chiave è necessario procedere con la generazione di una nuova chiave (Fig. 2.91).



Modalità di Accesso

Typo: api-key

Aggiorna Api Key:

Nuova Api Key

App ID:

Figure2.91: Aggiornamento delle credenziali “api-key”

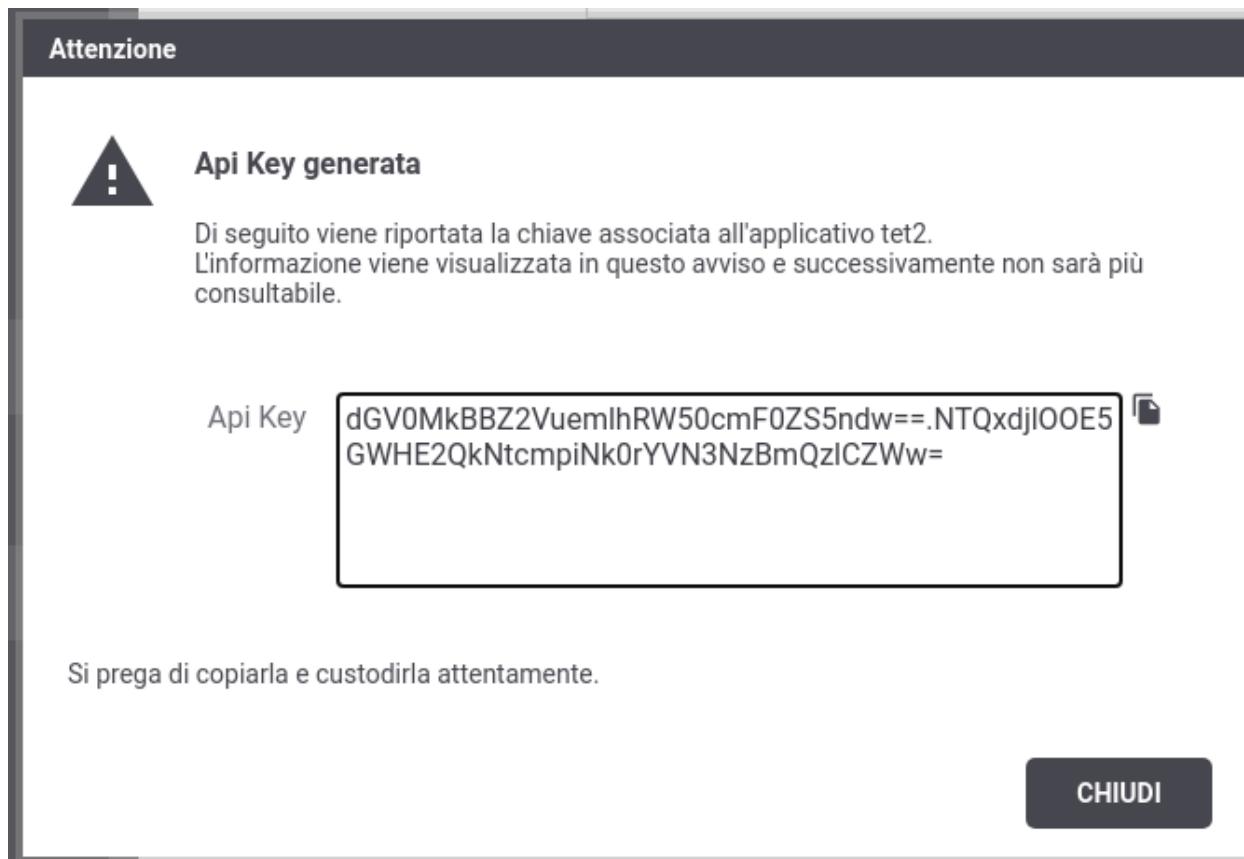


Figure2.89: Avviso di copia delle credenziali “api-key”



Figure2.90: Avviso di copia delle credenziali “api-key” (con App ID)

Credenziali “principal”

Agli applicativi ed ai soggetti registrati nel gateway, identificabili con credenziali “principal”, deve essere associato un identificativo univoco (Fig. 2.92).

Modalità di Accesso

Tipo: principal

UserId *

Figure2.92: Credenziali Principal

Credenziali “token”

Il tipo di credenziale *token* è associabile solamente agli applicativi.

Consente durante il processo di *Autenticazione Token* di identificare un applicativo registrato con la medesima *Token Policy Validazione* utilizzata nell'autenticazione e contenente come identificatore univoco il valore presente nel claim “clientId” all'interno del token (Fig. 2.93).

Modalità di Accesso

Tipo: token

Token Policy *: Google

Identificativo *: clientIdXATest

Figure2.93: Credenziali Token

2.10.9 Creazione di un ruolo

È possibile censire i ruoli che potranno essere utilizzati come criterio di autorizzazione. Quelli contrassegnati come *fonte registro* potranno essere associandoli ai soggetti. Quelli invece contrassegnati come *fonte esterna* verranno assegnati dinamicamente ai soggetti che si autenticano, sulla base di quanto comunicato dal container dopo che l'utente ha effettuato l'autenticazione esternamente.

Per creare un nuovo ruolo ci si posiziona nella sezione *Registro > Ruoli* e si preme il pulsante *Aggiungi*.

Compilare il form (Fig. 2.94) nel seguente modo:

- *Nome*: identifica univocamente il ruolo.

Note: (*) Campi obbligatori

Ruolo

Nome *	Sanzioni
Descrizione	descrizione del ruolo
Fonte	Esterna
Identificativo Esterno	Multe
Contesto	Erogazione

Invia **Cancella**

Figure2.94: Registrazione di un ruolo

- *Descrizione*: rappresenta una descrizione generica del ruolo.
- *Fonte*: la gestione del ruolo può essere effettuata direttamente su GovWay (fonte: registro) dove può essere assegnato ad un soggetto o applicativo. In alternativa (fonte: esterna) la gestione può essere delegata all'Application Server o a qualunque altra modalità che permetta al gateway di accedere ai ruoli tramite la api *HttpServletRequest.isUserInRole()*. In questo caso il nome del ruolo deve corrispondere allo stesso identificativo utilizzato nella configurazione esterna.

Se non viene specificata alcuna fonte il ruolo potrà essere utilizzato per entrambe le modalità.

- *Contesto*: l'utilizzo del ruolo può essere limitato ad un contesto di erogazione o fruizione di servizio attraverso questa opzione.
- *Identificativo Esterno*: Nei casi in cui il ruolo provenga da un sistema esterno, è possibile che il suo identificativo sia differente rispetto a quello indicato nel contesto del Registro. In tal caso inserire in questo campo tale identificativo esterno.

2.10.10 Attribuzione dei Ruoli a Soggetti ed Applicativi

È possibile attribuire un ruolo ad un soggetto cliccando sulla voce “Ruoli” presente sia nell’elenco dei soggetti che nel dettaglio di un singolo soggetto. L’attribuzione consiste nello scegliere uno dei ruoli selezionabili per il soggetto tra quelli compatibili con il contesto di erogazione di servizio e che prevedono una fonte di registrazione interna al registro.

In uguale maniera è possibile attribuire un ruolo ad un applicativo di tipologia *Frutore* cliccando sulla voce “Ruoli” presente nel dettaglio dell’applicativo. L’attribuzione consiste nello scegliere uno dei ruoli selezionabili per il servizio

Soggetti > Elenco > Ruoli di PROXY/ENTE

Ruolo

Nome: ruoloEsempio

Invia **Cancella**

Figure2.95: Attribuzione di un ruolo ad un soggetto

applicativo tra quelli compatibili con il contesto di fruizione di servizio e che prevedono una fonte di registrazione interna al registro.

Servizio Applicativo > Elenco > Ruoli di SA1

Ruolo

Nome: ruoloEsempio

Invia **Cancella**

Figure2.96: Attribuzione di un ruolo ad un applicativo

2.10.11 Autorizzazione per Token Claims

Se è stata abilitata la gestione del token si ha la possibilità di autorizzare le richieste inserendo i valori ammessi per i claims contenuti nel token.

L'autorizzazione per token claims permette di effettuare dei semplici controlli sui valori dei claim presenti nel token, una volta verificato che il token sia valido. La funzionalità è utilizzabile nei contesti in cui il controllo di autorizzazione possiede una logica semplice che si basa sulla verifica del valore di uno o più claim. Dove serve una logica più complessa (ad es. con rami “if-else”) il controllo deve essere effettuato utilizzando una XACMLPolicy ([XACML-Policy](#)).

La configurazione viene effettuata inserendo nel campo di testo un claim da verificare per ogni riga, facendo seguire dopo l'uguale un valore fornito in una delle seguenti modalità:

- \${anyValue} : indica qualsiasi valore non nullo.

- `${undefined}` : la risorsa indicata non deve esistere o non deve essere valorizzata.
- `${regExpMatch:EXPR}` : la regola è soddisfatta se l'intero valore del claim ha un match rispetto all'espressione regolare EXPR indicata. È possibile utilizzare anche la versione `${regExpNotMatch:EXPR}` che consente di attuare una negazione della condizione.
- `${regExpFind:EXPR}` : simile alla precedente regola, il match dell'espressione regolare può avvenire anche su una sottostringa del valore del claim. Come per la precedente esiste anche la versione `${regExpNotFind:EXPR}`.
- valore : indica esattamente il valore (case sensitive) che deve possedere il claim; il valore può essere definito come costante o contenere parti dinamiche risolte a runtime dal Gateway descritte di seguito.
- valore1,...,valoreN : è possibile elencare differenti valori ammissibili; come per la precedente opzione il valore può contenere parti dinamiche.
- `${ignoreCase:valore}` o `${ignoreCase:valore1,...,valoreN}` : simile alle precedenti regole consente di attuare una verifica case insensitive.
- `${not:valore}` o `${not:valore1,...,valoreN}` : simile alle precedenti regole consente di indicare esattamente i valori (case sensitive) che non deve possedere la risorsa. È possibile utilizzarla anche in combinazione con il controllo case-insensitive: `${not:${ignoreCase:valore}}` o `${not:${ignoreCase:valore1,...,valoreN}}`

Le espressioni utilizzabili come parti dinamiche, risolte a runtime dal gateway, sono:

- `${header:NAME}`: valore presente nell'header http che possiede il nome "NAME"
- `${query:NAME}`: valore associato al parametro della url con nome "NAME"
- `${urlRegExp:EXPR}`: espressione regolare applicata sulla url di invocazione (l'espressione deve avere un match con l'intera url)
- `${xPath:EXPR}`: espressione XPath applicata su un messaggio XML
- `${jsonPath:EXPR}`: espressione JSONPath applicata su un messaggio JSON
- `${tokenClient:FIELD}`: identità dell'applicativo client identificato tramite il clientId presente nel token; il valore "FIELD" fornito deve rappresentare un field valido all'interno della classe "org.openscoop2.core.id.IDServizioApplicativo" (es. per ottenere il nome dell'applicativo usare `${tokenClient:nome}`)
- `${transportContext:FIELD}`: permette di accedere ai dati della richiesta http; il valore "FIELD" fornito deve rappresentare un field valido all'interno della classe "org.openscoop2.utils.transport.http.HttpServletTransportRequestContext" (es. per il principal usare `${transportContext:credential.principal}`)
- `${busta:FIELD}`: permette di utilizzare informazioni generiche del profilo; il valore "FIELD" fornito deve rappresentare un field valido all'interno della classe "org.openscoop2.protocol.sdk.Busta" (es. per il mittente usare `${busta:mittente}`)
- `${property:NAME}`: utilizzabile solamente su erogazioni, permette di riferire informazioni specifiche del profilo presenti nella traccia (es. identificativo SDI). Il valore "NAME" indica il nome della proprietà da utilizzare
- `${securityToken:FIELD}`: permette di accedere ai certificati e ai security token presenti nella richiesta; il valore "FIELD" fornito deve rappresentare un field valido all'interno della classe "org.openscoop2.protocol.sdk.SecurityToken" (es. per accedere al CN del certificato presente nel token "Authorization" usare `${securityToken:authorization.certificate.subject.info(CN)}`)
- `${config:NAME}`: valore della proprietà configurata sull'API che possiede il nome "NAME"
- `${clientApplicationConfig:NAME}`: valore della proprietà configurata nell'applicativo fruitore che possiede il nome "NAME"
- `${clientOrganizationConfig:NAME}`: valore della proprietà configurata nel soggetto fruitore che possiede il nome "NAME"

- \${providerOrganizationConfig:NAME}: valore della proprietà configurata nel soggetto erogatore che possiede il nome “NAME”
- \${tokenClientApplicationConfig:NAME}: permette di accedere alla proprietà, configurata nell’applicativo client identificato tramite il clientId presente nel token, con nome “NAME”
- \${tokenClientOrganizationConfig:NAME}: permette di accedere alla proprietà, configurata nel soggetto proprietario dell’applicativo client identificato tramite il clientId presente nel token, con nome “NAME”
- \${dynamicConfig:FIELD}: permette di accedere alle proprietà degli attori coinvolti nella richiesta (api, applicativi, soggetti); il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.pdd.core.dynamic.DynamicConfig”; maggiori informazioni sulla funzionalità sono disponibili nella sezione “[Accesso alle proprietà delle entità del Registro](#)”.
- \${system:NAME}: valore associato alla proprietà di sistema, indicata nella configurazione generale, con nome “NAME”
- \${env:NAME}: valore associato alla variabile di sistema con nome “NAME”
- \${java:NAME}: valore associato alla variabile java con nome “NAME”
- \${envj:NAME}: valore associato alla variabile di sistema o java con nome “NAME”; la variabile viene cercata prima come variabile di sistema e se non presente come variabile della jvm

Di seguito alcuni esempi:

- client_id=3 : viene verificato che il claim “client_id” possieda il valore 3
- client_id=\${not:3} : viene verificato che il claim “client_id” non possieda il valore 3
- client_id=3,5,6 : viene verificato che il claim “client_id” possieda il valore 3 o 5 o 6
- client_id=\${not:3,5,6} : viene verificato che il claim “client_id” non possieda nessuno dei valori indicati: 3, 5 e 6
- username=\${ignoreCase:paolo rossi} : la verifica sul valore del claim “username” avviene con un criterio case insensitive. Nell’esempio i valori “Paolo Rossi”, “paolo rossi”, “PAOLO ROSSI” hanno tutti un match.
- username=\${not\${ignoreCase:paolo rossi,marco verdi}} : viene verificato che il claim “username” non possieda nessuno dei valori indicati. I valori vengono controllati con un criterio case insensitive.
- client_id=\${anyValue} : viene verificato che il claim “client_id” possieda un valore (not null e not empty)
- client_id=\${regExpMatch:[0-9]} : viene verificato che il claim “client_id” possieda esattamente una cifra decimale attraverso la verifica di un match con l’espressione regolare “[0-9]”
- client_id=\${regExpNotMatch:[0-9]} : viene verificato che il claim “client_id” non sia composto da una cifra decimale (l’espressione regolare “[0-9]” non deve essere soddisfatta)
- client_id=\${regExpFind:[0-9]} : viene verificato che il claim “client_id” contenga una cifra decimale attraverso l’espressione regolare “[0-9]”
- client_id=\${regExpNotFind:[0-9]} : viene verificato che il claim “client_id” non contenga una cifra decimale (l’espressione regolare “[0-9]” non deve essere soddisfatta)
- client_id=\${header:X-Prova} : viene verificato che il claim “client_id” possieda lo stesso valore presente nell’header http “X-Prova” presente nella richiesta
- client_id=cl-\${header:X-Prova} : viene verificato che il claim “client_id” possieda il valore presente nell’header http “X-Prova” arricchito del prefisso “cl-”
- client_id=\${query:prova} : viene verificato che il claim “client_id” possieda lo stesso valore presente nel parametro “prova” della url di invocazione

- `client_id=${urlRegExp:EXPR}` : viene verificato che il claim “client_id” possieda lo stesso valore estratto dalla url di invocazione attraverso l’applicazione dell’espressione regolare EXPR
- `client_id=${xpath:EXPR}` : viene verificato che il claim “client_id” possieda lo stesso valore estratto dalla richiesta xml tramite l’espressione XPath EXPR.
- `client_id=${jsonPath:EXPR}` : viene verificato che il claim “client_id” possieda lo stesso valore estratto dalla richiesta json tramite l’espressione jsonPath EXPR.
- `client_id=${transportContext:credential.certificateChain.certificate.subject.info(CN)}`: viene verificato che il claim “client_id” possieda lo stesso valore estratto dal “CN” del certificato TLS client.

Per verificare un attributo indicarlo con il prefisso “attribute.” nella forma “attribute.nome=valore”. Di seguito alcuni esempi

- `attribute.sesso=m` : viene verificato che l’attributo “sesso” possieda il valore m
- `attribute.stato=3,5,6` : viene verificato che l’attributo “stato” possieda il valore 3 o 5 o 6

Nel caso la configurazione relativa all’*Identificazione Attributi* prevede più AA, la verifica di un attributo prelevato da un authority va indicato con i prefissi “aa.” e “attribute.” nella forma “aa.nomeAuthority.attribute.nomeAttributo=valore”.

- `aa.AA2.attribute.sesso=m` : viene verificato che l’attributo “sesso”, prelevato tramite l’Attribute Authority “AA2”, possia il valore m
- `aa.AA2.attribute.stato=3,5,6` : viene verificato che l’attributo “stato”, prelevato tramite l’Attribute Authority “AA2”, possia il valore 3 o 5 o 6

2.10.12 XACML-Policy

Questa tipologia di autorizzazione prevede di limitare l’accesso ai soli applicativi o soggetti fruitori che soddisfino una determinata policy XACML. La policy deve essere caricata nel contesto dell’autorizzazione sul controllo degli accessi, come mostrato in Fig. 2.97.

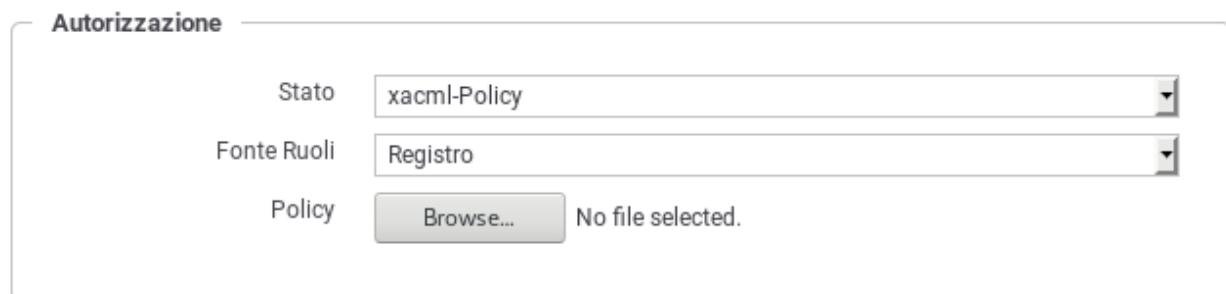


Figure2.97: Registrazione di una XACML-Policy per l’erogazione

In fase di autorizzazione, il gateway costruisce una XACMLRequest contenente tutti i parametri della richiesta, comprese le informazioni relative al chiamante (credenziali ed eventuali ruoli), e la valida rispetto alla XACML-Policy associata all’erogazione. I parametri inseriti nella XACMLRequest, che possono essere utilizzati per effettuare la verifica all’interno di una XACML-Policy, sono i seguenti:

Table2.2: Parametri inseriti in una XACMLRequest

Nome	Descrizione
Sezione "Action"	
org:govway:action:provider	Indica il soggetto erogatore del servizio
org:govway:action:provider:config:<nome>	Proprietà configurate nel soggetto erogatore del servizio
org:govway:action:service	Indica il servizio nel formato tipo/nome
org:govway:action:service:config:<nome>	Proprietà configurate nell'erogazione o nella fruizione
org:govway:action:action	Nome dell'operazione del servizio invocata
org:govway:action:url	Url di invocazione utilizzata dal mittente
org:govway:action:url:parameter:NOME_PARAM	Tutti i parametri presenti nell'url di invocazione saranno inseriti nella XACMLRequest con questo formato
org:govway:action:transport:header:NOME_HDR	Tutti gli header http presenti nell'url di invocazione saranno inseriti nella XACMLRequest con questo formato
org:govway:action:soapAction	Valore della SOAPAction
org:govway:action:gwService	Ruolo della transazione (inbound/outbound)
org:govway:action:protocol	Modalità associata al servizio richiesto (es. spcoop)
org:govway:action:token:audience	Destinatario del token
org:govway:action:token:scope	Lista di scopes
org:govway:action:token:jwt:claim:<nome>	Tutti i claims presenti nel jwt validato
org:govway:action:token:introspection:claim:<nome>	Tutti i claims presenti nella risposta del servizio di introspection
Sezione "Subject"	
org:govway:subject:organization	Indica il soggetto fruitore
org:govway:subject:organization:config:<nome>	Proprietà configurate nel soggetto fruitore
org:govway:subject:client	Identificativo dell'applicativo client
org:govway:subject:client:config:<nome>	Proprietà configurate nell'applicativo client
org:govway:subject:credential	Rappresenta la credenziale di accesso (username, subject o il principal) utilizzata dal client per richiedere il servizio
org:govway:subject:role	Elenco dei ruoli che possiede il client che ha richiesto il servizio
org:govway:subject:token:issuer	Issuer del token
org:govway:subject:token:subject	Subject del token
org:govway:subject:token:username	Username dell'utente cui è associato il token
org:govway:subject:token:clientId	Identificativo del client che ha negoziato il token
org:govway:subject:token:client	Identificativo dell'applicativo client registrato su GovWay e identificato tramite il clientId presente nel token (nel seguito applicativo token)
org:govway:subject:token:client:config:<nome>	Proprietà configurate nell'applicativo token
org:govway:subject:token:client:organization	Identificativo del soggetto proprietario dell'applicativo token
org:govway:subject:token:client:organization:config:<nome>	Proprietà configurate nel soggetto proprietario dell'applicativo token
org:govway:subject:token:client:role	Elenco dei ruoli che possiede l'applicativo token

continues on next page

Table 2.2 – continua dalla pagina precedente

Nome	Descrizione
org:govway:subject:token:userInfo:fullName	Nome completo dell'utente cui è associato il token
org:govway:subject:token:userInfo:firstName	Nome dell'utente cui è associato il token
org:govway:subject:token:userInfo:middleName	Secondo nome (o nomi aggiuntivi) dell'utente cui è associato il token
org:govway:subject:token:userInfo:familyName	Cognome dell'utente cui è associato il token
org:govway:subject:token:userInfo:eMail	Email dell'utente cui è associato il token
org:govway:subject:token:userInfo:claim:<nome>	Tutti i claims presenti nella risposta del servizio di UserInfo
org:govway:subject:attributes	Elenco dei nomi degli attributi recuperati interagendo con gli Attribute Authority configurati
org:govway:subject:attribute:<nome>	In caso sia configurato un unico Attribute Authority, nella configurazione relativa all' <i>Identificazione Attributi</i> , tutti gli attributi recuperati saranno inseriti nella XACMLRequest con questo formato
org:govway:subject:aa:<attributeAuthority>:attribute:<nome>	In caso siano configurate più Attribute Authority, nella configurazione relativa all' <i>Identificazione Attributi</i> , tutti gli attributi recuperati saranno inseriti nella XACMLRequest con questo formato

Di seguito un esempio di XACMLPolicy che autorizza le richieste dei chiamanti che possiedono il ruolo “Amministratore” ed uno tra i due ruoli “Operatore1” e “Operatore2”:

```

<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" PolicyId="Policy" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides" xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-policy-schema-os.xsd">
  <Target />
  <Rule Effect="Permit" RuleId="ok">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <SubjectAttributeDesignator AttributeId="org:govway:subject:role" DataType="http://www.w3.org/2001/XMLSchema#string" />
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              Amministratore
            </AttributeValue>
          </Apply>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <SubjectAttributeDesignator AttributeId="org:govway:subject:role" DataType="http://www.w3.org/2001/XMLSchema#string" />
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              Operatore1
              Operatore2
            </AttributeValue>
          </Apply>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>

```

(continues on next page)

(continua dalla pagina precedente)

```

<Operatore1</AttributeValue>
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
<Operatore2</AttributeValue>
    </Apply>
</Apply>
</Condition>
</Rule>
<Rule Effect="Deny" RuleId="ko" />
</Policy>

```

Un altro esempio di policy che verifica l'uguaglianza tra il valore del claim “sub” presente nel token e quello fornito nel query parameter “sub” è la seguente:

```

<Policy PolicyId="Policy"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-
    overrides"
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os" xmlns:xsi="http://www.w3.org/
    2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os http://docs.oasis-
    open.org/xacml/2.0/access_control-xacml-2.0-policy-schema-os.xsd">
    <Target />
    <Rule Effect="Permit" RuleId="ok">
        <Condition>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                <Apply FunctionId="urn:oasis:names:tc:xacml:1.
                <@:function:any-of-any">
                    <Function FunctionId="urn:oasis:names:tc:xacml:1.
                <@:function:string-equal"/>
                    <ActionAttributeDesignator
                        AttributeId="org:govway:action:url:parameter:sub
                        <">
                        DataType="http://www.w3.org/2001/XMLSchema
                        #string"
                        MustBePresent="false"
                    />
                    <ActionAttributeDesignator
                        AttributeId=
                            "org:govway:action:token:introspection:claim:sub"
                        DataType="http://www.w3.org/2001/XMLSchema
                        #string"
                        MustBePresent="false"
                    />
                </Apply>
            </Condition>
        </Rule>
        <Rule Effect="Deny" RuleId="ko" />
    </Policy>

```

2.10.13 Scope

Nella sezione *Registro > Scope* è possibile gestire il censimento degli scope da utilizzare successivamente per le politiche di autorizzazione nell'ambito del controllo degli accessi.

La maschera di creazione di uno scope è quella mostrata in Fig. 2.98.

Scope > Aggiungi

Note: (*) Campi obbligatori

Scope

Nome *

Descrizione

Identificativo Esterno

Contesto Qualsiasi

Invia Cancella

Figure 2.98: Creazione di uno Scope

I dati da fornire sono:

- *Nome*: nome assegnato internamente allo scope
- *Descrizione*: un testo di descrizione
- *Identificativo Esterno*: nome originale dello scope presente nel token
- *Contesto*: specifica se lo scope si utilizza solo nei contesti di erogazione, fruizione o entrambe le possibilità.

2.11 Rate Limiting

Questa sezione di configurazione consente di attivare policy di Rate Limiting per una specifica erogazione o fruizione (o specifico gruppo di configurazione nell'ambito di un'erogazione/fruizione).

L'attivazione di policy di rate limiting rientra nell'ambito degli strumenti per il controllo del traffico. La descrizione di dettaglio di questi strumenti è presente nella sezione *Controllo del Traffico*, e nella sezione *Policy Globali* per quanto riguarda l'attivazione di policy a valenza globale.

Nota: In presenza di una installazione con più nodi gateway attivi, GowWay per default effettua il conteggio delle

metriche utilizzate dalle policy di rate limiting indipendentemente su ogni singolo nodo del cluster. Questa soluzione è certamente la più efficiente, ma presenta dei limiti evidenti se è necessaria una contabilità precisa del numero di accessi consentiti globalmente da parte dell'intero cluster. In tali situazioni è necessario modificare la configurazioni di default per attivare modalità di conteggio distribuite, come descritto nella sezione [Rate Limiting in presenza di un cluster di nodi](#).

Una policy di rate limiting si compone concettualmente dei seguenti elementi che verranno maggiormente dettagliati nella sezione [Registrazione di una policy](#):

- **Criterio di Misurazione:** elemento che consente di calcolare un valore utile per la valutazione della policy. Il valore calcolato dipende dalla **metrica** scelta. La metrica viene scelta in fase di configurazione tra quelle disponibili, che sono:
 - *Numero Richieste*: consente di limitare il numero totale massimo di richieste consentite.
 - *Numero Richieste Simultanee*: limita il numero totale massimo di richieste simultanee consentite.
 - *Dimensione Massima Messaggi*: limita la dimensione massima accettata di una richiesta e di una risposta.
 - *Occupazione Banda*: limita il numero totale massimo di KB consentiti.
 - *Tempo Medio Risposta*: la policy blocca ogni successiva richiesta se viene rilevato un tempo medio di risposta elevato.
 - *Tempo Complessivo Risposta*: la policy limita il numero totale massimo di secondi consentiti.
 - *Numero Richieste Completate con Successo*: vengono conteggiate solamente il numero di richieste completate con successo; raggiunto il limite, ogni successiva richiesta viene bloccata.
 - *Numero Richieste Fallite*: vengono conteggiate il numero di richieste fallite; raggiunto il limite, ogni successiva richiesta viene bloccata.
 - *Numero Fault Applicativi*: vengono conteggiate il numero di richieste che veicolano un fault applicativo; raggiunto il limite, ogni successiva richiesta viene bloccata.
 - *Numero Richieste Fallite o Fault Applicativi*: vengono conteggiate il numero di richieste fallite o che veicolano un fault applicativo; raggiunto il limite, ogni successiva richiesta viene bloccata.
- **Dimensione della Finestra Temporale:** per ottenere un valore di confronto, alla metrica è necessario associare un intervallo di osservazione che consente di stabilire univocamente il conteggio risultante (fa eccezione la metrica “Numero Richieste Simultanee”). L’intervallo di osservazione può essere espresso scegliendone uno tra i seguenti:
 - *Minuti*
 - *Orario*
 - *Giornaliero*

Nota: L’intervallo indicato definisce una “fixed window”. Ad esempio definendo 1 minuto, anche se la prima richiesta arriva alle 12:00:07 l’intervallo di osservazione sarà [12:00:00.000 - 12:00:59.999], il successivo [12:01:00.000 - 12:01:59.999] e così via...

- **Soglia di Confronto:** elemento della policy che fornisce il valore di soglia da confrontare con il valore ottenuto dalla metrica impostata.
- **Filtro di Applicabilità:** elemento della policy che stabilisce i criteri per i quali è applicabile la policy sui flussi in elaborazione sul Gateway (filtro su mittente, api, applicativo, ecc.).

Criteri di valutazione delle policy

Per ogni singola erogazione o fruizione di API è possibile definire più politiche di Rate Limiting, anche con analoga metrica. Per ogni richiesta viene applicato il seguente algoritmo di valutazione delle policy (una descrizione di maggior dettaglio viene fornita nella sezione [Criteri di valutazione delle policy](#)):

- le policy vengono raggruppate «per metrica» e per ogni metrica vengono valutate nell'ordine di elenco.
- per ogni metrica vengono valutate le policy applicabili, cioè per le quali risultano soddisfatti il filtro e le condizioni di applicabilità.
- se la policy viola i livelli di soglia previsti, la transazione viene bloccata (o segnalata se configurata come «warning only») e la valutazione delle policy viene terminata.
- se la policy non viola invece i livelli di soglia previsti, si prosegue nella valutazione di ulteriori policy per quella metrica, solo se la policy è marcata come «prosegui».

Header HTTP informativi restituiti ai client: quote e finestre temporali

All'applicativo client vengono restituiti header http informativi che consentono di conoscere:

- il numero massimo di richieste effettuabili (quota);
- la finestra temporale in cui si applica la quota (informazione non attiva per default);
- il numero di secondi mancanti alla prossima finestra temporale dove il numero di richieste conteggiate verrà azzerato;
- il numero di richieste ancora effettuabili nella finestra temporale in corso;
- in caso di violazione della policy, il numero di secondi dopo i quali riprovare ad utilizzare il servizio.

Una descrizione di dettaglio degli header http viene fornita nella sezione [Header HTTP informativi restituiti ai client: quote e finestre temporali](#).

2.11.1 Registrazione di una policy

Per attivare una nuova policy dalla sezione di rate limiting si procede utilizzando il pulsante *Aggiungi* che apre il form di [Fig. 2.99](#).

Si compilano i campi seguenti:

- *Policy*: la policy da attivare. Si compone di:
 - *Nome*: Identificativo univoco della policy.
 - *Stato*: Lo stato della policy. Sono disponibili le seguenti opzioni:
 - * *Abilitato*: le violazioni rilevate saranno gestite in maniera restrittiva (negazione del servizio).
 - * *WarningOnly*: la policy è abilitata in modalità WarningOnly. Questo significa che le violazioni rilevate saranno solo segnalate tramite messaggi diagnostici ma non ci saranno ripercussioni sull'elaborazione della richiesta.
 - * *Disabilitato*: La policy è disabilitata.
 - *Elaborazione*: Indica quale azione attuare per la policy, nell'ambito del flusso di elaborazione delle policy di eguale metrica, nel caso in cui venga superato il controllo (maggiori dettagli sull'algoritmo di valutazione delle policy sono disponibili nella sezione [Criteri di valutazione delle policy](#)):
 - * *Interrompi*: non verranno valutate ulteriori policy che seguono nell'ordine tra quelle di eguale metrica.
 - * *Prosegui*: si procede con la valutazione della successiva policy nell'ordine tra quelle di eguale metrica.
 - *Identificazione Policy*: Scelta tra due opzioni:

Erogazioni > api-monitor v1 (Ente) > Configurazione > Rate Limiting > Aggiungi

Note: (*) Campi obbligatori

Policy

Nome *	<input type="text"/>
Stato	abilitato
Elaborazione	interrompi
Identificazione Policy	Scegli criteri

Criteri

Metrica	Numero Richieste
Intervallo Osservazione	Orario

Applicata solo in presenza di Congestione del Traffico (i)

Applicata solo in presenza di Degrado Prestazionale (i)

Valori di Soglia

Ridefinisci Valori di Soglia	<input type="checkbox"/>
Num. Massimo Richieste	100

Raggruppamento

Se abilitato, il calcolo del valore della soglia avviene raggruppando le richieste in funzione dei criteri selezionati

Stato	disabilitato
-------	--------------

Filtro

Stato	disabilitato
-------	--------------

SALVA

Figure2.99: Attivazione di una policy di Rate Limiting

- * *Scegli Criteri*: permette di indicare direttamente i criteri che la politica deve garantire; tra i criteri utilizzabili: la metrica (numero richieste, occupazione banda, tempi medi, ...), l'intervallo temporale (minuto, ora, giorno) e le condizioni di applicabilità (congestione, degrado prestazionale).
- * *Selezione dal Registro*: permette di utilizzare una politica arbitraria, precedentemente definita dall'utente.

Nota: La descrizione che segue assume che venga attuata una identificazione della policy per criteri. Per i dettagli sulla configurazione di policy personalizzate dall'utente si faccia riferimento alla sezione [Rate Limiting](#).

- *Criteri*: devono essere forniti la metrica e l'intervallo di osservazione scelti tra i valori descritti in precedenza ([Rate Limiting](#)). Possono inoltre essere indicate le seguenti opzioni:
 - * *Applicata solo in presenza di Congestione del Traffico*: attivando questa opzione la policy risulta applicabile solo nel caso in cui il gateway sia entrato in modalità «Congestione», sulla base di quanto descritto nella sezione [Controllo del Traffico](#).
 - * *Applicata solo in presenza di Degrado Prestazionale*: attivando questa opzione la policy risulta applicabile solo nel caso in cui il gateway abbia rilevato un degrado prestazionale e cioè un tempo medio di risposta del servizio superiore alla soglia configurata.
- *Valori di Soglia*: Le soglie per la valutazione della policy:
 - *Ridefinisci Valori di Soglia*: Opzione che consente di variare la soglia predefinita.
 - *Soglia*: Questo campo riporta, in base alla metrica selezionata sopra, il valore di riferimento. Tale valore risulta modificabile attivando l'opzione al punto precedente.
 - *Raggruppamento*: In questa sezione è possibile attivare opzionalmente alcuni criteri per il raggruppamento dei dati utilizzati come soglie di confronto. Ad esempio se la policy limita a 20 il numero di richieste su base per minuti, significa che al raggiungimento della ventunesima richiesta, nella stessa finestra temporale, si otterrà una violazione della policy. Aggiungendo un raggruppamento per risorsa, saranno conteggiate separatamente le richieste in base alla specifica risorsa invocata. In questo caso la policy risulterà violata solo al raggiungimento della ventunesima richiesta, nella stessa finestra temporale, relativa alla medesima risorsa. È ammesso anche il raggruppamento su criteri multipli. La logica è del tutto analoga a quella dell'operatore GROUP BY del linguaggio SQL. I criteri di raggruppamento selezionabili sono:
 - * *Risorsa/Azione*: il valore di soglia rappresenta il totale per ciascuna azione/risorsa
 - * *Richiedente*: il valore di soglia rappresenta il totale ripartito per ciascun mittente
 - * *Token*: il valore di soglia rappresenta il totale ripartito tra le richieste in base al token di provenienza. Si possono specificare i «claims» da prendere in considerazione per distinguere i token.
 - * *Chiave*: il valore di soglia rappresenta il totale ripartito tra le richieste raggruppate in base ad una chiave personalizzata il cui valore viene fornito secondo uno dei metodi selezionati tra i seguenti:
 - *Header HTTP*: La chiave è presente nell'header di trasporto indicato nella proprietà «Nome».
 - *Url di Invocazione*: La chiave è presente nella URL ricavabile tramite l'espressione regolare fornita nell'elemento seguente (l'espressione deve avere un match con l'intera url).
 - *Parametro della Url*: La chiave viene fornita in modalità Form Encoded con il parametro indicato nell'elemento «Nome».
 - *SOAPAction*: La chiave corrisponde al valore della SoapAction.
 - *Contenuto*: La chiave è presente nel body del messaggio e viene ricavata tramite una espressione XPath o JsonPath fornito nell'elemento seguente.

- *Client IP*: La chiave corrisponde all'indirizzo IP del client.
- *X-Forwarded-For*: La chiave corrisponde all'indirizzo IP del client presente negli header http utilizzati per il mantenimento dell'IP di origine nel caso di nodi intermedi (es. X-Forwarded-For).
- *Plugin Personalizzato*: La chiave viene restituita tramite l'esecuzione di una classe il cui nome viene fornito con il campo «Tipo». Per maggiori dettagli si rimanda alla sezione *Filtro o Raggruppamento Personalizzato*
- *Filtro*: Abilitando questa sezione è possibile indicare i criteri affinché la policy sia applicabile in base alle caratteristiche di ciascuna richiesta in ingresso. In assenza di filtro, la policy sarà valutata su tutte le richieste in ingresso che riguardano l'erogazione/fruizione che si sta configurando. Per la creazione del filtro sono disponibili i seguenti campi:
 - *Risorsa/Azione*: Opzione per filtrare le richieste in base all'azione/risorsa invocata.
 - *Ruolo Richiedente*: Opzione per filtrare le richieste in base al ruolo posseduto dal richiedente (sia che si tratti di un soggetto che di un applicativo).
 - *Soggetto o Applicativo Fruitore*: In alternativa al filtro per ruolo, è possibile specificare un soggetto fruitore ed eventualmente uno dei suoi applicativi (identificati tramite autenticazione trasporto o token).
 - *Token Claims*: consente di indicare per riga (nome=valore) i «claims» che le richieste devono possedere nel token OAuth2.
 - *Chiave*: Si tratta di un'opzione avanzata che consente di filtrare le richieste in ingresso sul gateway in base ad una chiave che può essere specificata in maniera personalizzata effettuando una delle seguenti scelte per il campo *Tipologia*:
 - * *Header HTTP*: Occorre fornire i dati “Nome” e “Valore”. La policy si applicherà soltanto alle richieste che hanno un header http che corrisponde.
 - * *Url di Invocazione*: Occorre fornire i dati “Espressione Regolare” e “Valore”. La policy si applicherà soltanto alle richieste ove, applicando l'espressione regolare alla URL di invocazione, si ottiene un valore identico a quello fornito.
 - * *Parametro della Url*: Occorre fornire i dati “Nome” e “Valore”. La policy si applicherà soltanto alle richieste che contengono nella url di invocazione un parametro corrispondente ai dati forniti.
 - * *SOAPAction*: Occorre fornire il dato “Valore”. La policy si applicherà soltanto alle richieste che si presentano con una SOAPAction avente il valore fornito.
 - * *Contenuto*: Occorre fornire i dati “Pattern” e “Valore”. La policy si applicherà soltanto alle richieste dove, applicando l'espressione XPath o JsonPath al messaggio di richiesta, si ottiene un valore identico a quello fornito.
 - * *Client IP*: La policy si applicherà soltanto alle richieste che provengono dall'indirizzo IP indicato.
 - * *X-Forwarded-For*: La policy si applicherà soltanto alle richieste che provengono dall'indirizzo IP indicato presente negli header http utilizzati per il mantenimento dell'IP di origine nel caso di nodi intermedi (es. X-Forwarded-For).
 - * *Plugin Personalizzato*: Permette di definire un criterio di filtro personalizzato. Per maggiori dettagli si rimanda alla sezione *Filtro o Raggruppamento Personalizzato*

2.11.2 Criteri di valutazione delle policy

Le policy di rate limiting create, per la data erogazione/fruizione, sono visualizzate in un elenco che filtra automaticamente su una singola metrica (ad esempio «Numero Richieste» o «Occupazione Banda»). L'elenco delle policy visualizzato è analogo a quello riportato in Fig. 2.100.

	Ordine	Stato	Nome	Soglia	Runtime	Elaborazione
■	▼	●	numeroMax	100	Visualizza	▼
■	^ ▼	●	limiteMaxGiornaliero	1000	Visualizza	×
■	^	●	sogliaMinuto	10	Visualizza	×

Figure 2.100: Elenco delle policy di Rate Limiting

Ciascun elemento in elenco riporta le seguenti informazioni:

- *Ordine*: pulsanti per variare la posizione della policy nell'elenco per la data metrica.
- *Stato*: lo stato di abilitazione della policy, sulla base di quanto descritto in precedenza ([Registrazione di una policy](#)).
- *Nome*: il nome della policy.
- *Soglia*: il valore di soglia impostato per la policy.
- *Runtime*: permette di effettuare una verifica in tempo reale della metrica interrogando il runtime del gateway. Maggiori dettagli sono presenti nella sezione [Visualizzazione Statistiche Policy](#).
- *Elaborazione*: flusso di elaborazione (proseguì, interrompi) nel caso di superamento del controllo relativo alla policy.

L'elenco delle policy può essere aggiornato utilizzando il meccanismo di filtro presente nell'intestazione della tabella. Sono disponibili le seguenti opzioni:

- *Metrica*: permette di stabilire le policy da visualizzare in base alla rispettiva metrica.
- *Ricerca*: permette di visualizzare le policy in base alla presenza di un pattern nel nome.

Per ogni richiesta relativa alla specifica erogazione/fruizione viene applicato l'algoritmo di valutazione delle policy che è il seguente:

- le policy vengono raggruppate «per metrica» e per ogni metrica vengono valutate nell'ordine di elenco prima utilizzando le politiche di Rate Limiting definite sull'API e poi, se esistenti, le politiche a valenza globale ([Policy Globali](#)).
- per ogni metrica vengono valutate le policy applicabili, cioè per le quali risultano soddisfatti il filtro e le condizioni di applicabilità.
- se la policy viola i livelli di soglia previsti, la transazione viene bloccata (o segnalata se configurata come «warning only») e la valutazione delle policy viene terminata.

- se la policy non viola invece i livelli di soglia previsti, si prosegue nella valutazione di ulteriori policy per quella metrica, solo se la policy è marcata come «proseguì».

2.11.3 Header HTTP informativi restituiti ai client: quote e finestre temporali

All'applicativo client vengono restituiti header http informativi che consentono di conoscere:

- il numero massimo di richieste effettuabili (quota);
- la finestra temporale in cui si applica la quota (informazione disponibile solamente con opzione “window” abilitata, descritta nella sezione *Personalizzazione degli Header HTTP restituiti al client*);
- il numero di secondi mancanti alla prossima finestra temporale dove il numero di richieste conteggiate verrà azzerato;
- il numero di richieste ancora effettuabili nella finestra temporale in corso;
- in caso di violazione della policy, il numero di secondi dopo i quali riprovare ad utilizzare il servizio.

Il nome dell'header HTTP utilizzato da GovWay varia a seconda della metrica associata ad una policy. Ad esempio nella tabella [Tabella 2.4](#) vengono riportati i nomi degli header HTTP utilizzati per la metrica “Numero Risorse” che contengono le informazioni sopra descritte.

Se verranno configurate policy con metriche differenti verranno restituite al client i corrispettivi header HTTP previsti per ogni metrica verificata. Una descrizione completa dei nomi degli header http utilizzati per ogni metrica è disponibile nella sezione *Header HTTP utilizzati nelle Policy di Rate Limiting*.

In presenza di molteplici policy con la medesima metrica verranno ritornate le informazioni relative alla policy più restrittiva.

L'indicazione sul numero di secondi dopo i quali il client può riprovare ad utilizzare il servizio, in caso di violazione di una policy, viene invece riportato nell'header http descritto nella tabella [Tabella 2.3](#). Il numero indicato nell'header *Retry-After* viene calcolato sommando al numero di secondi mancante alla prossima finestra temporale, un tempo di backoff rappresentato da un numero random di secondi tra 0 e 60. L'aggiunto del tempo di backoff mira ad evitare che al ripristino dell'intervallo tutti i client in attesa concentrino le richieste nel medesimo istante.

È possibile personalizzare gli header http restituiti al client disabilitandone la generazione oppure facendo ritornare anche l'informazione sulla finestra temporale. Le modalità di personalizzazione vengono descritte nella sezione *Personalizzazione degli Header HTTP restituiti al client*.

Header HTTP utilizzati nelle Policy di Rate Limiting

- **Retry After** ([Tabella 2.3](#))

Table2.3: Header HTTP “Retry After”

Nome Header HTTP	Descrizione
Retry-After	Numero di secondi dopo i quali il client può riprovare ad utilizzare il servizio

- **Numero Richieste** ([Tabella 2.4](#))

Table2.4: Header HTTP relativi alla metrica “Numero Richieste”

Nome Header HTTP	Descrizione
X-RateLimit-Limit	Numero massimo di richieste effettuabili nell’intervallo temporale configurato
X-RateLimit-Remaining	Numero di richieste ancora effettuabili nella finestra temporale in corso
X-RateLimit-Reset	Numero di secondi mancante alla prossima finestra temporale

- **Numero Richieste Simultanee** ([Tabella 2.5](#))

Table2.5: Header HTTP relativi alla metrica “Numero Richieste Simultanee”

Nome Header HTTP	Descrizione
GovWay-RateLimit- ConcurrentRequest-Limit	Numero massimo di richieste simultanee effettuabili
GovWay-RateLimit- ConcurrentRequest-Remaining	Numero di richieste ancora effettuabili

- **Occupazione Banda** ([Tabella 2.6](#))

Table2.6: Header HTTP relativi alla metrica “Occupazione Banda”

Nome Header HTTP	Descrizione
GovWay-RateLimit- BandwithQuota-Limit	Numero totale massimo di KB consentiti nell’intervallo temporale configurato
GovWay-RateLimit- BandwithQuota-Remaining	Banda ancora occupabile (in KB) nella finestra temporale in corso
GovWay-RateLimit- BandwithQuota-Reset	Numero di secondi mancante alla prossima finestra temporale

- **Tempo Medio Risposta** ([Tabella 2.7](#))

Table2.7: Header HTTP relativi alla metrica “Tempo Medio Risposta”

Nome Header HTTP	Descrizione
GovWay-RateLimit- AvgTimeResponse-Limit	Tempo medio di risposta atteso
GovWay-RateLimit- AvgTimeResponse-Reset	Numero di secondi mancante alla prossima finestra temporale

- **Tempo Complessivo Risposta** ([Tabella 2.8](#))

Table2.8: Header HTTP relativi alla metrica “Tempo Complessivo Risposta”

Nome Header HTTP	Descrizione
GovWay-RateLimit-TimeResponseQuota-Limit	Numero totale massimo di secondi consentiti nell’intervallo temporale configurato
GovWay-RateLimit-TimeResponseQuota-Remaining	Tempo di risposta (in secondi) ancora occupabile nella finestra temporale in corso
GovWay-RateLimit-TimeResponseQuota-Reset	Numero di secondi mancante alla prossima finestra temporale

- **Numero Richieste Completate con Successo, Fallite o con Fault Applicativi** (Tabella 2.9)

Table2.9: Header HTTP relativi alla metrica “Numero Richieste Completate con Successo, Fallite o con Fault Applicativi”

Header metrica “Completate con Successo”	Header HTTP metrica “Fallite”	Header metrica “Fault Applicativi”	Header metrica “Fallite e Fault Applicativi”	Descrizione
GovWay-RateLimit-RequestSuccessful-Limit	GovWay-RateLimit-RequestFailed-Limit	GovWay-RateLimit-Fault-Limit	GovWay-RateLimit-RequestFailedOrFaultLimit	Numero di richieste consentite nell’intervallo temporale configurato
GovWay-RateLimit-RequestSuccessful-Remaining	GovWay-RateLimit-RequestFailed-Remaining	GovWay-RateLimit-Fault-Remaining	GovWay-RateLimit-RequestFailedOrFaultRemaining	Numero di richieste ancora effettuabili nella finestra temporale in corso
GovWay-RateLimit-RequestSuccessful-Reset	GovWay-RateLimit-RequestFailed-Reset	GovWay-RateLimit-Fault-Reset	GovWay-RateLimit-RequestFailedOrFaultReset	Numero di secondi mancante alla prossima finestra temporale

Personalizzazione degli Header HTTP restituiti al client

Per personalizzare gli header http restituiti al client è richiesto l’accesso alla govwayConsole in modalità *avanzata* (sezione [Modalità Avanzata](#)).

A partire dall’erogazione o fruizione di una API, accedendo alla sezione [Configurazione dell’API](#) in modalità avanzata compare una sezione precedentemente non documentata denominata *Opzioni Avanzate*. All’interno di tale sezione è possibile agire sulla configurazione della voce *HTTP Headers* nella sezione *Rate Limiting* per personalizzare la generazione degli header http (Fig. 2.101).

Le configurazioni attuabili sono:

- *Default*: vengono generati tutti gli header HTTP descritti nella sezione [Header HTTP utilizzati nelle Policy di Rate Limiting](#). Il valore degli header “*-Limit” non contiene l’indicazione della finestra temporale;
- *Disabilitato*: gli header HTTP relativi al RateLimiting non vengono generati;
- *Ridefinito*: consente di personalizzare gli header HTTP restituiti.

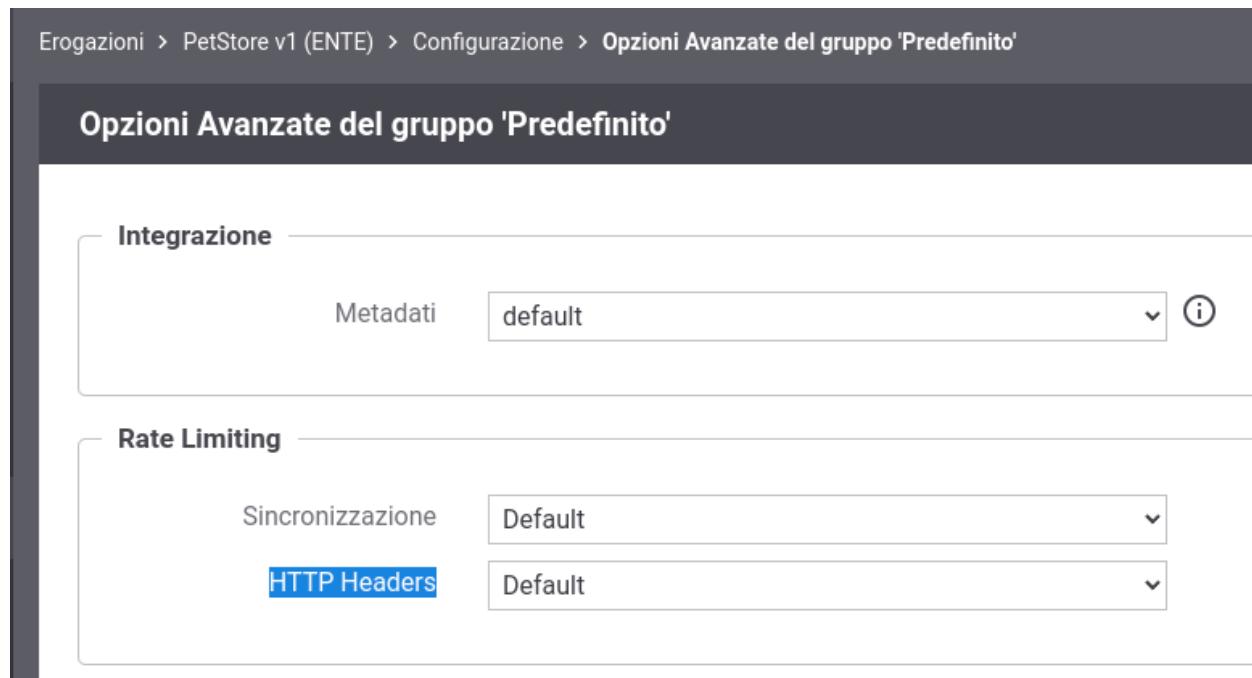


Figure2.101: Configurazione Header HTTP per il Rate Limiting

La maschera di configurazione fornirà le seguenti ulteriori voci nel caso venga selezionata la ridefinizione degli header HTTP (Fig. 2.102):

- *Limiti di Quota*: consente di personalizzare la generazione degli header “*-Limit”:
 - *Abilitato (senza finestra temporale)*: il valore contiene solamente il numero massimo di richieste effettuabili nella finestra temporale;
 - *Abilitato (con finestra temporale)*: oltre al numero massimo di richieste, viene anche riportata la finestra temporale e le ulteriori finestre delle policy attive che possiedono una medesima metrica ma sono meno restrittive (esempio: X-RateLimit-Limit: 10, 10;w=60, 1000;w=3600);
 - *Disabilitato*: gli header “*-Limit” non verranno generati;
 - *Default*: viene utilizzata la modalità indicata nel file di proprietà «govway.properties» che nella configurazione di default del prodotto equivalente alla voce “Abilitato (senza finestra temporale)”.
- *Rimanenza della Quota*:
 - *Abilitato*: il valore contiene il numero di richieste ancora effettuabili nella finestra temporale in corso;
 - *Disabilitato*: gli header “*-Remaining” non verranno generati;
 - *Default*: viene utilizzata la modalità indicata nel file di proprietà «govway.properties» che nella configurazione di default del prodotto equivalente alla voce “Abilitato”;
- *Reset della Quota (secondi)*:
 - *Abilitato*: il valore contiene il numero di secondi mancanti alla prossima finestra temporale;
 - *Disabilitato*: gli header “*-Reset” non verranno generati;
 - *Default*: viene utilizzata la modalità indicata nel file di proprietà «govway.properties» che nella configurazione di default del prodotto equivalente alla voce “Abilitato”;
- *Retry-After*:

- *Abilitato (con backoff)*: l'header viene valorizzato sommando al numero di secondi mancante alla prossima finestra temporale un tempo di backoff rappresentato da un numero random di secondi tra 0 e un intervallo massimo configurabile da console;
- *Abilitato (senza backoff)*: l'header viene valorizzato solamente con il numero di secondi mancante alla prossima finestra temporale;
- *Disabilitato*: l'header non viene generato;
- *Default*: viene utilizzata la modalità indicata nel file di proprietà «govway.properties» che nella configurazione di default del prodotto equivalente alla voce “Abilitato (con backoff)” dove l'intervallo massimo di backoff è 60 secondi.

Rate Limiting

Sincronizzazione	Default
HTTP Headers	Ridefinito
Header HTTP	
Limiti di Quota	Abilitato (con finestra temporale)
Rimanenza della Quota	Default
Reset della Quota (secondi)	Default
Retry-After	Abilitato (con backoff)
Backoff (secondi) *	

Figure2.102: Configurazione ridefinita per gli Header HTTP del Rate Limiting

Oltre a personalizzare la gestione degli header http puntualmente su una erogazione o fruizione è possibile attuare una configurazione, identica a quanto già precedentemente descritto, a livello globale di GovWay agendo nella sezione *Rate Limiting* presente nella maschera di configurazione del *Controllo del Traffico* (sezione *Rate Limiting*).

2.11.4 Rate Limiting in presenza di un cluster di nodi

In presenza di una installazione con più nodi gateway attivi, GowWay per default effettua il conteggio delle metriche utilizzate dalle policy di rate limiting indipendentemente su ogni singolo nodo del cluster. Questa soluzione è certamente la più efficiente, ma presenta dei limiti evidenti se è necessaria una contabilità precisa del numero di accessi consentiti globalmente da parte dell'intero cluster, in quanto si potrebbe riscontrare la violazione di una policy solamente su alcuni nodi e non su altri a seconda di come avvenga la distribuzione delle richieste sui singoli nodi.

In alcuni casi specifici questa soluzione può comunque essere adottata, appoggiandosi alla collaborazione dei load balancer. Ad esempio, nel caso si voglia adottare una politica che limiti il numero di richieste per ogni client fruitore, è possibile utilizzare sui load balancer una politica di sticky session in modo che le richieste dello stesso client siano gestite sempre dalla stessa istanza GovWay, così da ottenere il rispetto puntuale dei limiti impostati. Se invece si volesse adottare una politica che limiti il numero di richieste verso una specifica API, indipendentemente dal client

che le effettui, è possibile utilizzare un bilanciamento del carico di tipo “round robin” per l’endpoint corrispondente a quella API. Le richieste sarebbero così distribuite equamente tra i nodi consentendo di applicare correttamente la policy semplicemente suddividendo la quota per il numero di nodi attivi.

Ma già se si volessero attuare entrambe queste politiche simultaneamente, le politiche di load balancing richieste risulterebbero tra loro incompatibili, e si dovrebbe pertanto passare a modalità di calcolo delle metriche distribuite tra i diversi nodi del cluster. Visto che tali modalità hanno un certo impatto prestazionale, è sempre necessaria un’attenta analisi preliminare per individuare la configurazione più indicata per la propria specifica situazione.

GovWay consente di modificare la modalità di gestione di default delle policy di rate limiting sia globalmente che puntualmente rispetto alla singola erogazione o fruizione.

Per modificare la modalità di default è possibile intervenire nella sezione *Rate Limiting* presente nella maschera di configurazione del *Controllo del Traffico* (sezione *Rate Limiting*). Per modificare la modalità di gestione su una singola erogazione o fruizione è richiesto invece l’accesso alla govwayConsole in modalità *avanzata* (sezione *Modalità Avanzata*). Nel seguito viene documentato come modificare la configurazione per una singola erogazione o fruizione, analoghe modalità potranno essere utilizzate per intervenire a livello globale.

Dopo aver selezionato una specifica erogazione o fruizione, accedendo alla sezione *Configurazione dell’API* in modalità avanzata compare la sezione *Opzioni Avanzate*. All’interno di tale sezione è possibile agire sulla configurazione della voce *Sincronizzazione* nella sezione *Rate Limiting* (Fig. 2.103).

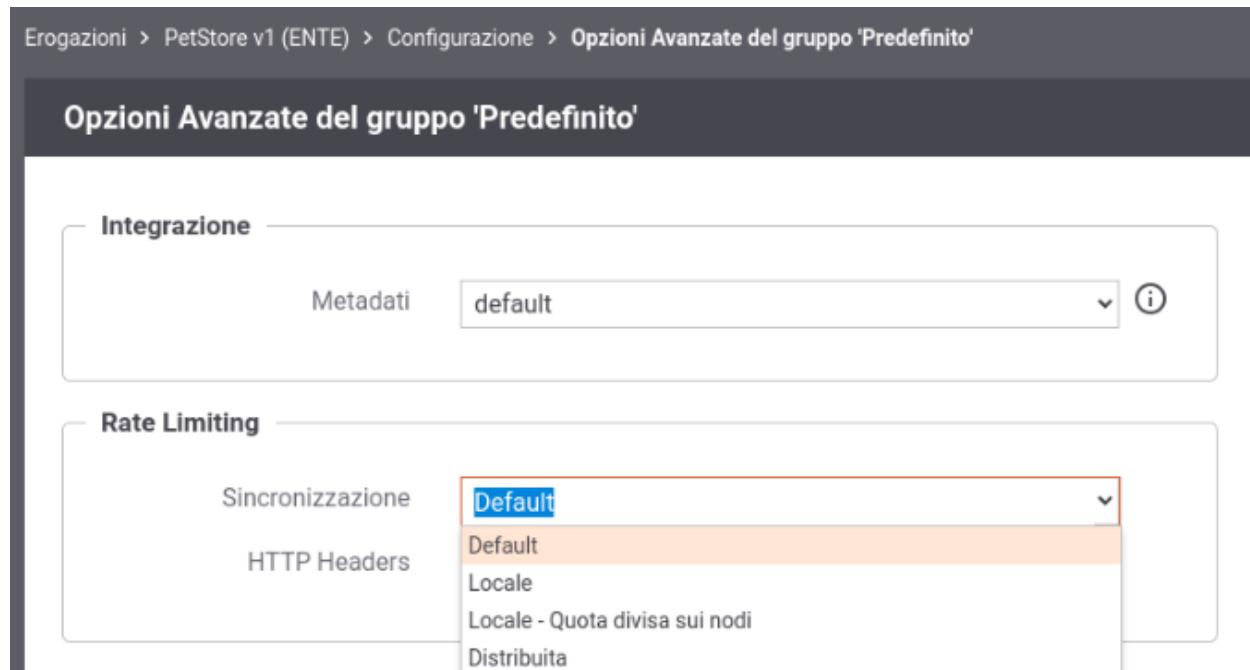


Figure 2.103: Sincronizzazione del Rate Limiting in un cluster di nodi

Le modalità di gestione delle policy attivabili su GovWay sono le seguenti:

- *Locale*: ogni nodo effettua il proprio conteggio;
- *Locale - quota divisa sui nodi*: gestione delle policy tramite bilanciamento del carico, come descritto nella sezione *Sincronizzazione Locale con quota divisa tra i nodi*.
- *Distribuita*: il conteggio viene attuato tramite un archivio dati distribuito, come descritto nella sezione *Sincronizzazione Distribuita*;
- *Default*: viene utilizzata la gestione indicata nel file di proprietà «govway.properties» che nella configurazione di default del prodotto è equivalente alla voce “Locale”.

Sincronizzazione Locale con quota divisa tra i nodi

Questa modalità di gestione delle policy di rate limiting fornisce una possibile soluzione nei casi in cui sia possibile distribuire equamente tra i nodi tutte le richieste relative alla policy che si intende attuare.

Prerequisito: la presenza di un load balancer che distribuisca in maniera uniforme tra i singoli nodi del cluster le richieste relative ad ogni singola policy.

In questo caso è possibile attuare correttamente la policy di rate limiting semplicemente suddividendo la quota configurata per il numero di nodi attivi.

La modalità di sincronizzazione “*Locale - quota divisa sui nodi*” attiva la registrazione automatica dei singoli nodi del cluster al proprio avvio e la loro cancellazione durante lo shutdown, in modo da permettere ad ogni nodo di conoscere l’effettivo numero di nodi attivi ai fini del corretto calcolo della quota da applicare su ogni singolo nodo (Fig. 2.104).

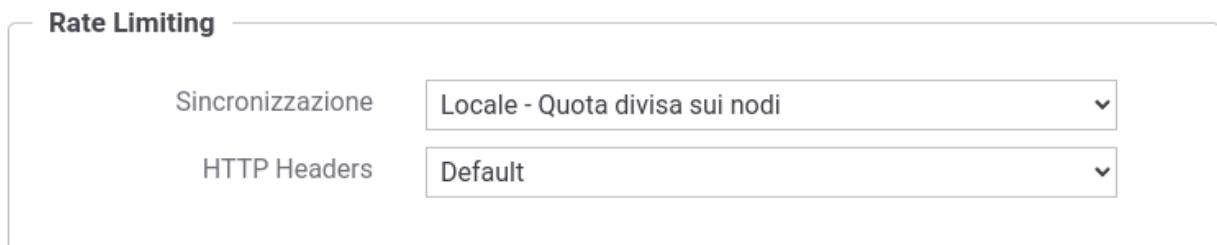


Figure 2.104: Sincronizzazione Locale con quota divisa sui nodi del cluster

Vantaggi

Questa modalità assicura le stesse prestazioni della sincronizzazione *Locale* attiva per default sul prodotto.

Svantaggi

1. Dipendenza dal Load Balancer

La soluzione richiede la collaborazione del load balancer, ad esempio con una politica “round robin” sulle richieste in arrivo sull’endpoint di una specifica API per politiche che richiedono di limitare il numero di richieste per quella API.

2. Applicabilità limitata

La soluzione è banalmente applicabile solo nei casi in cui GovWay gestisca un’unica API e il rate limiting vada applicato esclusivamente al numero di richieste verso quella API, indipendentemente dal client che le stia generando. Nei casi in cui siano gestite più API, la politica di gestione “round robin” sul bilanciatore dovrà essere gestita separatamente per l’endpoint di ogni API interessata, mentre la soluzione non risulta più applicabile se il conteggio debba tener conto del client richiedente (es. 1000 richieste max al minuto per API, 10 richieste max al minuto per ogni client). In tal caso il bilanciamento dovrebbe avvenire per ogni client fruitore, operazione molto complessa da gestire sui load balancer.

Inoltre questa modalità è applicabile solamente con metriche che prevedono di contare il numero di richieste in una finestra temporale. Non è quindi applicabile per le metriche: «Numero Richieste Simultanee», «Occupazione Banda», «Tempo Medio Risposta» e «Tempo Complessivo Risposta».

2. Header «*-Remaining»

La modalità prevede di suddividere la quota per il numero di nodi e ciò rende difficile calcolare il valore esatto del numero di richieste ancora disponibili nella finestra temporale in corso. Nel resto di questa sezione verranno fornite indicazioni sulle tecniche di approssimazione dei valori ritornati negli header «*-Remaining».

Header «-Remaining»*

Il valore del numero di richieste ancora disponibili nella finestra temporale in corso verrà calcolato tramite una tecnica di approssimazione che consiste nel moltiplicare la quota rimasta su un nodo per il numero di nodi attivi.

Di seguito viene mostrato un flusso di richieste di esempio, supponendo di avere un cluster formato da 2 nodi, e una policy di rate limiting impostata con una metrica che prevede 11 richieste al minuto.

```
Quota configurata: 11
Nodi attivi: 2
Quota effettiva = 11/2 = 5

Invocazione 1 (nodo1): X-RateLimit-Remaining (4*2 = 8)
Invocazione 2 (nodo2): X-RateLimit-Remaining (4*2 = 8)
Invocazione 3 (nodo1): X-RateLimit-Remaining (3*2 = 6)
Invocazione 4 (nodo2): X-RateLimit-Remaining (3*2 = 6)
Invocazione 5 (nodo1): X-RateLimit-Remaining (2*2 = 4)
Invocazione 6 (nodo2): X-RateLimit-Remaining (2*2 = 4)
Invocazione 7 (nodo1): X-RateLimit-Remaining (1*2 = 2)
Invocazione 8 (nodo2): X-RateLimit-Remaining (1*2 = 2)
Invocazione 9 (nodo1): X-RateLimit-Remaining (0*2 = ['remaining.zeroValue' ? 0
  ↵: 1])
Invocazione 10(nodo2): X-RateLimit-Remaining (0*2 = ['remaining.zeroValue' ? 0
  ↵: 1])
Invocazione 11(nodo1): 409
Invocazione 12(nodo2): 409
```

Vi sono alcuni aspetti dell'approssimazione che sono configurabili agendo sul file `<directory-lavoro>/govway_local.properties`

- Quota effettiva: come mostrato nell'esempio la quota configurata viene suddivisa sui nodi attivi. L'arrotondamento del valore ottenuto può essere configurato per difetto o per eccesso. Con un arrotondamento per difetto, se la divisione non consente di avere un numero maggiore di 0 viene associato il valore 1 ad ogni nodo. Per default è attivo un arrotondamento per difetto.

```
# Quota effettiva
org.openscoop2.pdd.controlloTrafico.gestorePolicy.inMemory.LOCAL_DIVIDED_
  ↵BY_NODES.limit.roundingDown=true
```

- Header «*-Limit»: nell'header è possibile indicare la quota configurata o la quota effettiva (ottenuta dal calcolo per difetto o eccesso) moltiplicata per il numero di nodi. Nell'esempio sopra riportato nel primo caso verrebbe ritornato il valore 11, mentre nel secondo 10. Per default viene ritornata la quota configurata.

```
# Quota normalizzata
org.openscoop2.pdd.controlloTrafico.gestorePolicy.inMemory.LOCAL_DIVIDED_
  ↵BY_NODES.limit.normalizedQuota=false
```

- Remaining 0 o 1: è possibile configurare quale valore restituire nel caso sia rimasta solamente 1 invocazione a disposizione (per default viene ritornato il valore 1):
 - valore 1, viene correttamente rispettato il contratto con il client; ad esempio, nel caso di 2 nodi, con quota N, alla N-1 esima invocazione il client si vede restituire correttamente un remaining=1; lo svantaggio di questa soluzione è che il client riceverà un errore 409 senza mai aver ricevuto una risposta con remaining=0;
 - restituendo 0 il client ottiene un remaining=0 alla penultima invocazione e potrebbe quindi decidere di non fare ulteriori invocazioni evitando di ricevere un errore 409. Lo svantaggio di

questa soluzione è che il client, se rispetta le indicazioni dell'header, effettuerà N-1 invocazioni rispetto alle N consentite dalla propria policy.

```
# Remaining zeroValue
org.openspcoop2.pdd.controlloTrafico.gestorePolicy.inMemory.LOCAL_DIVIDED_
-BY_NODES.remaining.zeroValue=false
```

Sincronizzazione Distribuita

Questa modalità consente di implementare qualunque politica di rate limiting in maniera effettivamente distribuita tra i nodi del cluster, indipendentemente dalle modalità previste per il bilanciamento del carico.

Il conteggio delle metriche viene effettuato tramite un archivio dati distribuito.

Vantaggi

Non vi è alcuna dipendenza rispetto alla modalità di bilanciamento del carico, né alcuna limitazione sul tipo di policy applicabile.

Svantaggi

Aumento della latenza di gestione delle richieste, dovuta alla concorrenza delle operazioni sui contatori distribuiti. Per ovviare al potenziale degrado prestazionale vengono fornite varie modalità di sincronizzazione che consentono di ottimizzare le prestazioni, rinunciando alla completa precisione dei conteggi.

Differenti tecniche di sincronizzazione

Il principale problema con un archivio dati centralizzato è dovuto alla concorrenza delle operazioni di aggiornamento che devono essere effettuate in maniera atomica per assicurare la consistenza del dato totale conteggiato. In presenza di elevato traffico questo può comportare un apprezzabile degrado prestazionale.

Per mitigare il problema, è possibile utilizzare modalità asincrone di aggiornamento dei dati tra i nodi del cluster. Con questo approccio il dato “master” risulta sempre consistente mentre la copia locale di ogni nodo viene aggiornata con sincronizzazioni periodiche, con la controindicazione che la precisione dei conteggi soffre delle finestre di risincronizzazione.

Soluzioni

GovWay consente di avere un conteggio delle metriche effettuato tramite una delle seguenti tipologie di archivio dati distribuito:

- *Hazelcast* (<https://github.com/hazelcast/>): soluzione fornita built-in con GovWay, che consente di implementare l’archivio dati distribuito tra i nodi del cluster;
- *Redis* (<https://redis.io/>); soluzione alternativa alla precedente, dove GovWay deve essere configurato per accedere ad un database redis;
- *Embedded*; implementazione built-in, in cui il dato “master” viene gestito sul database di GovWay. La soluzione viene fornita per ambienti di test in cui si desidera provare le funzionalità di rate limiting distribuito.

Nota: La soluzione “embedded” non è utilizzabile su ambienti di produzione.

Hazelcast

Il conteggio delle metriche viene effettuato tramite un archivio dati distribuito implementato tramite Hazelcast (<https://github.com/hazelcast/>).

GovWay consente di configurare 2 tecniche di sincronizzazione:

- *Misurazione Esatta*: attivabile impostando la sincronizzazione “*Distribuita*” e scegliendo le voci “*Misurazione esatta*” e “*Algoritmo atomic-long-counters*” (Fig. 2.105). Con questa modalità sia il dato “master” che quelli locali al nodo risultano essere sempre aggiornati.

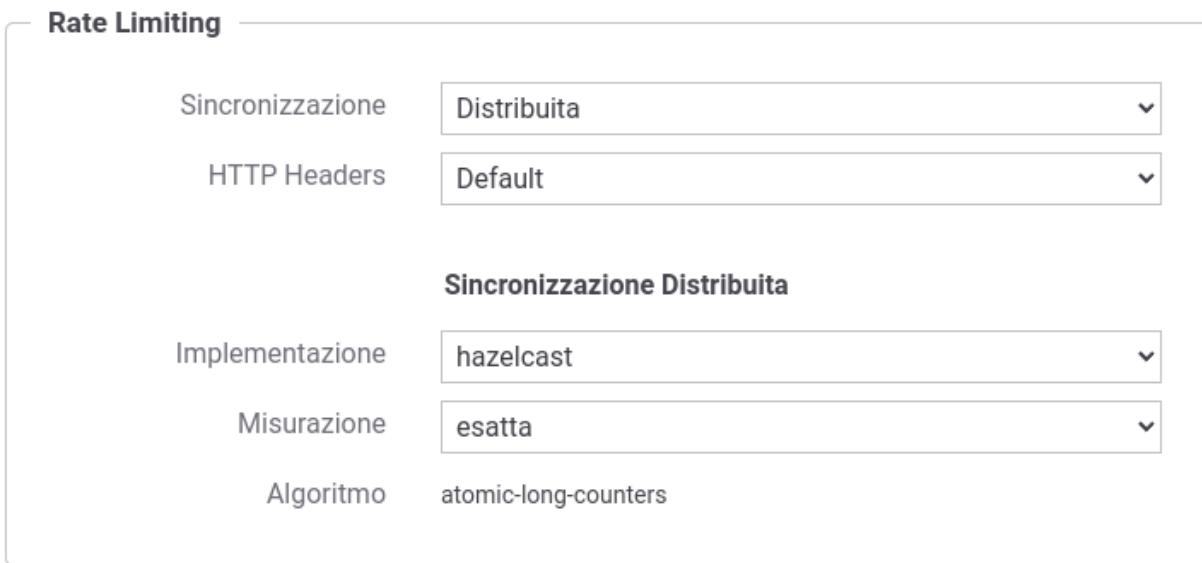


Figure 2.105: Sincronizzazione Distribuita “Hazelcast” con misurazione delle metriche esatta

- *Misurazione Approssimata*: attivabile impostando la sincronizzazione “*Distribuita*” e scegliendo le voci “*Misurazione esatta*” e “*Algoritmo pn-counters*” (Fig. 2.106). Con questa modalità l’incremento del dato “master” viene effettuato sempre in maniera atomica, ma la sua esecuzione avviene in maniera asincrona senza bloccare il nodo che ha effettuato l’operazione che utilizza i «PN Counters» per consultare i dati locali al nodo. I «PN Counters» sono una struttura dati fornita da Hazelcast in cui tutti gli aggiornamenti effettuati su un nodo vengono replicati in modo asincrono sugli altri. Tutti i nodi convergono sullo stesso stato dopo un pò di tempo. Maggiori dettagli vengono forniti nella documentazione del prodotto: <https://docs.hazelcast.com/hazelcast/5.1/data-structures/pn-counter>

Nella sezione *Configurazione di Hazelcast* vengono fornite informazioni sul tipo di configurazione utilizzata su Hazelcast, mentre nella sezione *Log emessi da Hazelcast* viene indicato dove è possibile reperire i log emessi da Hazelcast.

Altre modalità di utilizzo di hazelcast, forniti nelle precedenti versioni di GovWay, vengono descritte nella sezione *Algoritmi Alternativi di Hazelcast* ma ne si sconsiglia l’utilizzo poichè meno performanti rispetto alle due soluzioni sopra indicate.

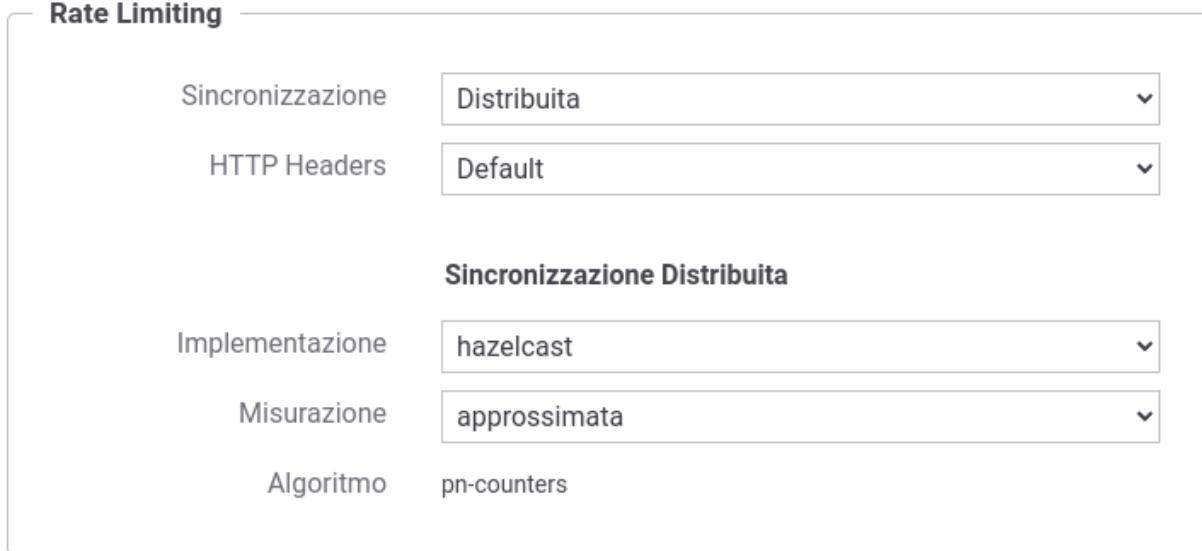


Figure2.106: Sincronizzazione Distribuita “Hazelcast” con misurazione delle metriche approssimata tramite «PNCounters»

Configurazione di Hazelcast

Nelle sezioni successive viene mostrato come modificare le configurazioni Hazelcast di default utilizzate dai gestori delle policy di rate limiting descritti nella sezione *Hazelcast*.

Per ogni algoritmo utilizzato per collezionare i contatori relativi alle polici di RateLimiting viene associata una istanza di Hazelcast dedicata che viene attivata con una configurazione specifica configurabile nei seguenti aspetti:

- “*Nome del Cluster*”
- “*Discovery dei nodi del cluster*”
- “*Porte dedicate ad ogni tipo di misurazione*”

Nome del Cluster

Tutti i nodi che concorrono a formare un cluster hazelcast sono identificati da un nome di cluster identificativo. Se il processo di discovery dei nodi in una rete non utilizza un censimento puntuale dei nodi (es. lista di indirizzi ip) ma tecniche di auto-discovery o multicast, può essere necessario utilizzare un nome di cluster differente per raggruppare nodi della stessa rete in gruppi funzionali differenti (es. produzione, sviluppo, test). Ulteriori dettagli su come configurare il processo di discovery vengono forniti nella successiva sezione “*Discovery dei nodi del cluster*”.

Per default il nome del cluster utilizzato è *govway*, ma è possibile ridefinirlo tramite 2 modalità:

- definire la proprietà *org.openspcoop2.pdd.controlloTraffic.gestorePolicy.inMemory.HAZELCAST.group_id* (default *govway*) agendo sul file *<directory-lavoro>/govway_local.properties*;
- definire l’elemento *cluster-name* all’interno del file *<directory-lavoro>/govway_local.hazelcast.yaml*.

Nota: Nel caso vengano utilizzate entrambe le modalità sopra descritte, la definizione all’interno del file *<directory-lavoro>/govway_local.hazelcast.yaml* prevale.

Discovery dei nodi del cluster

La configurazione di default di hazelcast prova automaticamente a rilevare nodi del cluster presenti nella rete. Se in una medesima rete esistono cluster funzionali diversi, è necessario associare dei cluster name differenti in modo da raggruppare i nodi per gruppi funzionali (es. produzione, sviluppo, test). Maggiori informazioni su come modificare il cluster name vengono fornite nella sezione “[Nome del Cluster](#)”.

È possibile modificare la configurazione di discovery di default agendo sul file `<directory-lavoro>/govway_local.hazelcast.yaml`. Per una documentazione completa sugli elementi configurabili nel file è possibile consultare la documentazione di Hazelcast: “<https://docs.hazelcast.com/hazelcast/5.1/clusters/network-configuration>”.

Nota: Ambiente di Produzione In un ambiente di produzione non è consigliato utilizzare un discovery automatico (<https://docs.hazelcast.com/hazelcast/5.1/clusters/discovery-mechanisms#auto-detection>). Si consiglia di disabilitare l’elemento “auto-detection” e di utilizzare un meccanismo alternativo come ad esempio il censimento puntuale dei nodi tramite l’elemento “tpc-ip.member-list”.

La configurazione di rete definita nel file `<directory-lavoro>/govway_local.hazelcast.yaml` verrà applicata per tutte le istanze attivate agli algoritmi descritti nelle sezioni “[Hazelcast](#)” e “[Algoritmi Alternativi di Hazelcast](#)”.

Se nel file `<directory-lavoro>/govway_local.hazelcast.yaml` viene definito l’elemento “port”, la sua configurazione viene ignorata e non viene riportata sulle istanze attivate poichè, come descritto nella sezione “[Porte dedicate ad ogni tipo di misurazione](#)”, ad ogni algoritmo viene associata una porta dedicata. Per modificare la porta si deve agire sulla configurazione specifica dell’algoritmo.

Per lo stesso motivo i membri definiti nell’elemento “tpc-ip.member-list” non dovrebbero contenere l’indicazione della porta. In alternativa è possibile utilizzare la keyword “`GOVWAY_INSTANCE_PORT`” che verrà sostituita per ogni algoritmo con la porta corretta.

Porte dedicate ad ogni tipo di misurazione

Per ogni algoritmo utilizzato per collezionare i contatori relativi alle polici di RateLimiting (descritti nelle sezioni “[Hazelcast](#)” e “[Algoritmi Alternativi di Hazelcast](#)”) viene associata una istanza di Hazelcast dedicata attivata con una configurazione specifica.

Ad ogni istanza viene associato un cluster name differente formato dal valore configurato su govway (descritto nella sezione “[Nome del Cluster](#)”) e un suffisso che riporta la modalità di conteggio selezionata (es. con un cluster name “govway” le istanze verranno configurate come “govway-atomic-long”, “govway-pncounter”, ...).

Inoltre la configurazione associata ad ogni istanza deve possedere una porta univoca che consente l’attivazione di molteplici istanze Hazelcast.

Per questo motivo la funzionalità “auto-increment” deve rimanere disabilitata in modo che ogni nodo del cluster sappia esattamente la porta associata all’algoritmo configurato.

Nota: Per preservare l’associazione porta-algoritmo, se nel file `<directory-lavoro>/govway_local.hazelcast.yaml` (descritto nella sezione “[Discovery dei nodi del cluster](#)”) viene definito l’elemento “port” la sua configurazione viene ignorata. Per modificare la porta si deve agire sulla configurazione specifica dell’algoritmo riportata di seguito in questa pagina.

Di seguito vengono riportate le configurazioni yaml di default associate agli algoritmi descritti nella sezione “[Hazelcast](#)”.

- *Misurazione Esatta:*

```

hazelcast:
  cluster-name: govway

  network:
    port:
      auto-increment: false
      port: 5701

```

- *Misurazione Approssimata:*

```

hazelcast:
  cluster-name: govway

  pn-counter:
    "pncounter-*-rl":
      # Lasciare abilitata la statistica altrimenti si ha il seguente bug
      # nel govway_hazelcast.log:
      # java.lang.NullPointerException: null
      # at com.hazelcast.internal.crdt.pncounter.PNCounterService.
      merge(PNCounterService.java)
      statistics-enabled: true

  network:
    port:
      auto-increment: false
      port: 5702

```

È possibile utilizzare una configurazione differente da quella di default definendo un file di configurazione yaml nella `<directory-lavoro>/govway.hazelcast-atomic-long-counters.yaml`

- *Misurazione Esatta: `<directory-lavoro>/govway.hazelcast-atomic-long-counters.yaml`*
- *Misurazione Approssimata: `<directory-lavoro>/govway.hazelcast-pn-counters.yaml`*

Log emessi da Hazelcast

I log emessi da Hazelcast, riguardanti lo stato della sincronizzazione dei nodi del cluster sono riversati nel file di log `<directory-log>/govway_hazelcast.log`

Algoritmi Alternativi di Hazelcast

Le modalità in cui vengono collezionati i contatori relativi alle politiche di RateLimiting in GovWay sono due:

- contatori relativi ad ogni informazione (numero di richieste, banda, intervallo temporale, richieste attive) gestiti singolarmente;
- contatori raggruppati in un oggetto che viene salvata in una mappa.

La seconda modalità, utilizzata nelle precedenti versioni di GovWay e descritta in questa sezione, è meno performante rispetto all'utilizzo dei singoli contatori descritti nella sezione “[Hazelcast](#)”. Le modalità vengono comunque mantenute sia per backward compatibility che per supportare scenari avanzati di policy (vedi sezione “[Registro Policy](#)”) che richiedono una gestione raggruppata in un unico oggetto dei vari contatori (es. intervallo con finestra “precedente”).

Differenti tecniche di sincronizzazione su una mappa

Nella sezione “*Hazelcast*” sono già stati descritti due approcci che si differenziano sulla gestione del dato locale al nodo, mentre l’aggiornamento del dato “master” risulta sempre consistente. Il dato locale su un nodo viene letto in un caso sempre in maniera consistente accedendo al data master (misurazione esatta), mentre in un altro vi è una copia locale su ogni nodo che viene aggiornata con sincronizzazioni periodiche, con la controindicazione che la precisione dei conteggi soffre delle finestre di risincronizzazione (misurazione approssimata).

Oltre ai due approcci sopra indicati, nelle tecniche di sincronizzazione su una mappa viene anche utilizzato un terzo approccio «*get and set*», in cui si recupera il valore corrente, lo si incrementa e quindi lo si rispedisce al datastore senza utilizzare le tecniche di incremento atomico fornite dal gestore stesso. In questo modo il dato “master” perderà di precisione poichè potrà capitare che richieste simultanee gestite su nodi differenti prelevino la stessa informazione e la modifichino senza tenere conto delle altre analoghe operazioni in corso, ottenendo così che il conteggio risulti approssimato per difetto. Per migliorare ulteriormente le prestazioni, anche l’operazione di “*set*” può essere resa asincrona.

Di seguito vengono fornite le varie modalità di sincronizzazione distribuita che utilizzano una mappa configurabili su GovWay, presentate in ordine, dalla modalità di “Misurazione Esatta” che garantisce il rispetto puntuale delle politiche previste, fino alla modalità con «*get and set asincrono*», che è quella che garantisce le prestazioni migliori ma con una maggiore perdita di precisione dei conteggi.

Nota: Configurazione degli algoritmi alternativi La configurazione di default disponibile sulla console di gestione (govwayConsole) non consente di selezionare gli algoritmi alternativi descritti in questa sezione. Per abilitarne la configurazione deve essere aggiunta la proprietà seguente al file `<directory-lavoro>/console_local.properties`:

```
# Gestori delle Policy di RateLimiting
controlloTrafico.policyRateLimiting.tipiGestori=LOCAL,LOCAL_DIVIDED_BY_NODES,
  ↳DATABASE,HAZELCAST_ATOMIC_LONG,HAZELCAST_PNCOUNTER,HAZELCAST_MAP,HAZELCAST_
  ↳NEAR_CACHE,HAZELCAST_NEAR_CACHE_UNSAFE_SYNC_MAP,HAZELCAST_NEAR_CACHE_UNSAFE_
  ↳ASYNC_MAP,HAZELCAST_LOCAL_CACHE,REDISSON_ATOMIC_LONG
```

-
- *Misurazione Esatta*: attivabile impostando la sincronizzazione “*Distribuita*” e scegliendo le voci “*Misurazione esatta*” e “*Algoritmo map*” (Fig. 2.107). Con questa modalità sia il dato “master” che quelli locali al nodo risultano essere sempre aggiornati.
 - *Near Cache*: attivabile impostando la sincronizzazione “*Distribuita*” e scegliendo le voci “*Misurazione approssimata*” e “*Algoritmo near-cache*” (Fig. 2.108). Con questa modalità l’incremento del dato “master” viene effettuato sempre in maniera atomica, ma la sua esecuzione avviene in maniera asincrona senza bloccare il nodo che ha effettuato l’operazione che utilizza una «*NearCache*» per consultare i dati locali al nodo. La «*NearCache*» è una struttura dati fornita da Hazelcast che viene risincronizzata rispetto ai dati remoti con sincronizzazioni periodiche. Maggiori dettagli vengono forniti nella documentazione del prodotto: <https://docs.hazelcast.com/imdg/latest/performance/near-cache>
 - *Local Cache*: attivabile impostando la sincronizzazione “*Distribuita*” e scegliendo le voci “*Misurazione approssimata*” e “*Algoritmo local-cache*” (Fig. 2.109). Questa modalità è analoga alla precedente, ma i nodi del gateway, anzichè utilizzare la «*NearCache*» di Hazelcast, utilizzano una propria copia locale del dato che viene risincronizzata ogni 5 secondi. L’intervallo di risincronizzazione può essere modificato agendo sul file `<directory-lavoro>/govway_local.properties` tramite la seguente proprietà:

```
# Intervallo di aggiornamento della cache in secondi
org.openspcoop2.pdd.controlloTrafico.gestorePolicy.inMemory.HAZELCAST_
  ↳LOCAL_CACHE.updateInterval=5
```

- *Misurazione approssimata con «*get and set sincrono*»*: attivabile impostando la sincronizzazione “*Distribuita*” e scegliendo le voci “*Misurazione inconsistente*” e “*Algoritmo remote-sync*” (Fig. 2.110). Con questa modalità l’incremento viene effettuato utilizzando un approccio «*get and set*» senza atomicità in cui la modifica del “dato master” avviene tramite un’operazione sincrona.

Rate Limiting

Sincronizzazione	Distribuita
HTTP Headers	Default

Sincronizzazione Distribuita

Implementazione	hazelcast
Misurazione	esatta
Algoritmo	map

Figure2.107: Sincronizzazione Distribuita con misurazione delle metriche esatta memorizzata su una mappa

Rate Limiting

Sincronizzazione	Distribuita
HTTP Headers	Default

Sincronizzazione Distribuita

Implementazione	hazelcast
Misurazione	approssimata
Algoritmo	near-cache

Figure2.108: Sincronizzazione Distribuita con misurazione delle metriche esatta in remoto ed utilizzo di una «NearCache» in locale

Rate Limiting

Sincronizzazione	Distribuita
HTTP Headers	Default

Sincronizzazione Distribuita

Implementazione	hazelcast
Misurazione	approssimata
Algoritmo	local-cache

Figure2.109: Sincronizzazione Distribuita con misurazione delle metriche esatta in remoto ed utilizzo di una cache locale

Rate Limiting

Sincronizzazione	Distribuita
HTTP Headers	Default

Sincronizzazione Distribuita

Implementazione	hazelcast
Misurazione	inconsistente
Algoritmo	remote-sync

Figure2.110: Sincronizzazione Distribuita con misurazione delle metriche approssimata tramite algoritmo «get and set» con pubblicazione sincrona

- *Misurazione approssimata con «get and set asincrono»:* attivabile impostando la sincronizzazione “*Distribuita*” e scegliendo le voci “*Misurazione inconsistente*” e “*Algoritmo remote-async*” (Fig. 2.111). Come nella precedente modalità l’incremento viene effettuato utilizzando un approccio «get and set» senza atomicità in cui la modifica del “dato master” avviene tramite un’operazione asincrona.

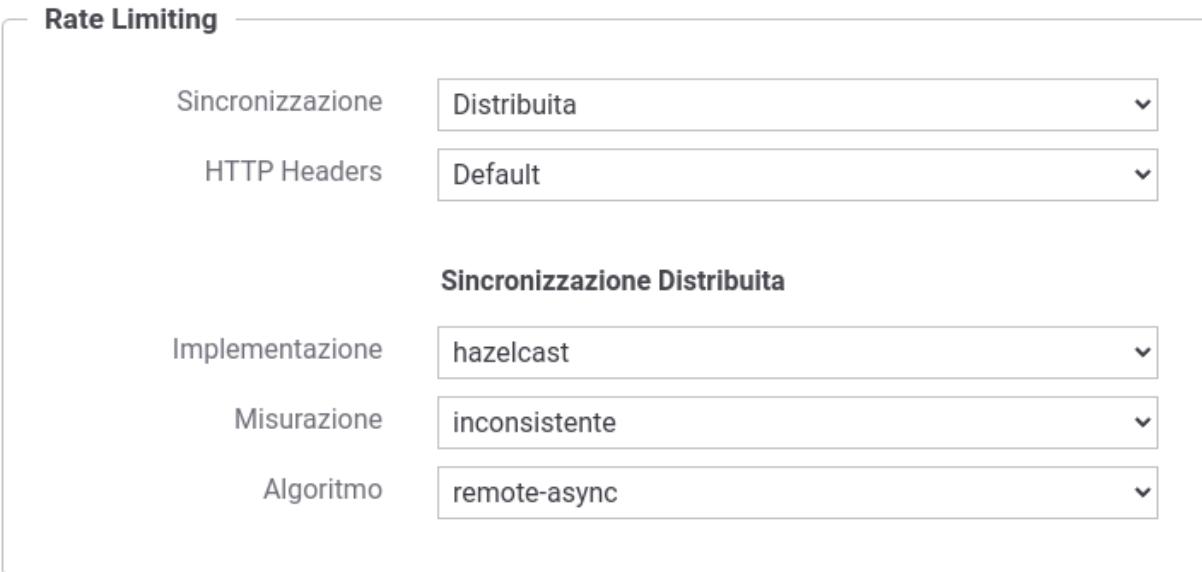


Figure 2.111: Sincronizzazione Distribuita con misurazione delle metriche approssimata tramite algoritmo «get and set» con pubblicazione asincrona

Configurazione di Hazelcast per la gestione della mappa

Di seguito vengono mostrate le configurazioni Hazelcast di default utilizzate dai gestori delle policy di rate limiting descritti in questa sezione.

Ad ogni tipo di gestore viene associata una istanza di Hazelcast dedicata che viene attivata con una configurazione specifica.

Nota: Port. La configurazione associata ad ogni gestore deve possedere una porta univoca che consenta l’attivazione di molteplici istanze Hazelcast. La funzionalità “auto-increment” deve rimanere disabilitata in modo che ogni nodo del cluster sappia esattamente la porta associata al tipo di gestore configurato.

- *Misurazione Esatta:*

```

hazelcast:
  cluster-name: govway
  map:
    "hazelcast-*-rate-limiting":
      in-memory-format: BINARY

  serialization:
    serializers:
      - type-class: org.openspcoop2.core.controllo_traffico.beans.
        IDUnivocoGroupByPolicy
        class-name: org.openspcoop2.pdd.core.controllo_traffico.policy.
        driver.hazelcast.IDUnivocoGroupByPolicyStreamSerializer
  
```

(continues on next page)

(continua dalla pagina precedente)

```
network:
  port:
    auto-increment: false
  port: 5707
```

- *Near Cache:*

```
hazelcast:
  cluster-name: goway
  map:
    "hazelcast-*-rate-limiting":
      in-memory-format: BINARY
      backup-count: 0
      async-backup-count: 1

    near-cache:
      in-memory-format: BINARY

  serialization:
    serializers:
      - type-class: org.openspcoop2.core.controllo_traffico.beans.
        ↵IDUnivocoGroupByPolicy
          class-name: org.openspcoop2.pdd.core.controllo_traffico.policy.
        ↵driver.hazelcast.IDUnivocoGroupByPolicyStreamSerializer

  network:
    port:
      auto-increment: false
    port: 5709
```

- *Local Cache:*

```
::

hazelcast:
  cluster-name: goway
  map:
    "hazelcast-*-rate-limiting":
      in-memory-format: BINARY
      backup-count: 0
      async-backup-count: 1

  serialization:
    serializers:
      - type-class: org.openspcoop2.core.controllo_traffico.beans.
        ↵IDUnivocoGroupByPolicy
          class-name: org.openspcoop2.pdd.core.controllo_traffico.policy.driver.
        ↵hazelcast.IDUnivocoGroupByPolicyStreamSerializer

  network:
    port:
```

(continues on next page)

(continua dalla pagina precedente)

```
auto-increment: false
port: 5703
```

- *Misurazione approssimata con «get and set sincrono»:*

```
hazelcast:
  cluster-name: govway
  map:
    "hazelcast-*-rate-limiting":
      in-memory-format: BINARY
      backup-count: 0
      async-backup-count: 1

    near-cache:
      in-memory-format: BINARY

  serialization:
    serializers:
      - type-class: org.openspcoop2.core.controllo_traffico.beans.
        IDUnivocoGroupByPolicy
        class-name: org.openspcoop2.pdd.core.controllo_traffico.policy.
        driver.hazelcast.IDUnivocoGroupByPolicyStreamSerializer

  network:
    port:
      auto-increment: false
      port: 5704
```

- *Misurazione approssimata con «get and set asincrono»:*

```
hazelcast:
  cluster-name: govway
  map:
    "hazelcast-*-rate-limiting":
      in-memory-format: BINARY
      backup-count: 0
      async-backup-count: 1

    near-cache:
      in-memory-format: BINARY

  serialization:
    serializers:
      - type-class: org.openspcoop2.core.controllo_traffico.beans.
        IDUnivocoGroupByPolicy
        class-name: org.openspcoop2.pdd.core.controllo_traffico.policy.
        driver.hazelcast.IDUnivocoGroupByPolicyStreamSerializer

  network:
    port:
      auto-increment: false
      port: 5705
```

È possibile utilizzare una configurazione differente da quella di default definendo un file di configurazione yaml nella `<directory-lavoro>` di govway specifico per ogni modalità:

- *Misurazione Esatta: <directory-lavoro>/govway.hazelcast-map.yaml*
- *Near Cache: <directory-lavoro>/govway.hazelcast-near-cache.yaml*
- *Local Cache: <directory-lavoro>/govway.hazelcast-local-cache.yaml*
- *Misurazione approssimata con «get and set sincrono: <directory-lavoro>/govway.hazelcast-near-cache-unsafe-sync-map.yaml*
- *Misurazione approssimata con «get and set asincrono: <directory-lavoro>/govway.hazelcast-near-cache-unsafe-async-map.yaml*

Redis

Il conteggio delle metriche viene effettuato tramite un archivio dati distribuito implementato tramite Redis (<https://redis.io/>).

La url di connessione verso il database Redis deve essere configurata sul file `<directory-lavoro>/govway_local.properties` tramite la seguente proprietà:

```
# Connection Url (possono essere fornite più url separate da virgola)
# usare rediss:// per TLS (con due s)
org.openspcoop2.pdd.controlloTrafico.gestorePolicy.inMemory.REDIS.
  ↪connectionUrl=redis://<HOST>:<PORT>
```

Su Redis viene attualmente supportata una tecnica di sincronizzazione con misurazione esatta attivabile impostando la sincronizzazione “*Distribuita*”, implementazione “*Redis*” e scegliendo le voci “*Misurazione esatta*” e “*Algoritmo atomic-long-counters*” (Fig. 2.112). Con questa modalità sia il dato “*master*” che quelli locali al nodo risultano essere sempre aggiornati.

Rate Limiting

Sincronizzazione	Distribuita
HTTP Headers	Default
Sincronizzazione Distribuita	
Implementazione	redis
Misurazione	esatta
Algoritmo	atomic-long-counters

Figure2.112: Sincronizzazione Distribuita “Redis” con misurazione delle metriche esatta

Embedded

Il conteggio delle metriche viene effettuato tramite un archivio dati distribuito implementato sul database di GovWay ed avviene tramite una tecnica di sincronizzazione con misurazione esatta.

La soluzione viene fornita per ambienti di test in cui si desidera provare le funzionalità di rate limiting distribuito.

Nota: La soluzione “embedded” non è utilizzabile su ambienti di produzione.

Per attivarlo impostare una sincronizzazione “*Distribuita*” e scegliere l’implementazione “*Embedded*”.

Rate Limiting

Sincronizzazione	Distribuita
HTTP Headers	Default

Sincronizzazione Distribuita

Implementazione	embedded
-----------------	----------

Figure2.113: Sincronizzazione Distribuita “*Embedded*” con misurazione delle metriche esatta

2.12 Validazione dei messaggi

Per attivare la validazione dei messaggi in transito sul gateway si accede al collegamento presente nella colonna *Validazione* presente tra gli elementi di configurazione della specifica erogazione/fruizione.

Compilare il form di configurazione (Fig. 2.114):

- *Stato*: Consente di abilitare/disabilitare la funzionalità di validazione sulla voce di configurazione scelta. L’opzione *warnignOnly* consente di attivare la funzionalità di validazione evitando però che, se tale fase non viene superata, venga bloccato il messaggio e restituito un errore. In quest’ultimo caso, gli errori di validazione verranno segnalati solo tramite l’emissione di opportuni messaggi diagnostici dal servizio di tracciamento.
- *Tipo*: Nel caso si sia abilitato il servizio di validazione, questo campo consente di selezionare la metodologia che si vuole utilizzare. I valori selezionabili da questo elenco cambiano in base alla tipologia delle API cui fa riferimento l’erogazione/fruizione.

I tipi di validazione previsti sono:

- *Wsdl 1.1*, la validazione si basa sull’interfaccia wsdl fornita con la API. Questo tipo di validazione è più rigorosa in quanto controlla non solo la conformità sintattica ma viene validato il messaggio in transito verificando che sia idoneo al PortType e Operation in uso. Questo tipo di validazione è applicabile solo al caso Soap.



Figure2.114: Validazione dei messaggi

- *Swagger 2.0 o OpenAPI 3.0*, nei casi in cui si è fornito un descrittore formale per una API Rest, la validazione sarà effettuata utilizzando gli strumenti associati allo specifico formato.
- *Schemi XSD*, la validazione si basa sugli schemi xsd allegati alle API. Utilizzato per la validazione sintattica dei messaggi XML sia nel caso Soap che Rest.

Nel caso di servizi Soap, se i messaggi che transitano sulla porta di dominio possiedono il formato MTOM, per poterli validare è necessario attivare l'opzione *Accetta MTOM*. Tale opzione normalizza i messaggi prima di effettuarne la validazione e ripristina il formato originale una volta completato il processo di validazione.

Nota: Per la validazione dei messaggi riguardanti API REST con specifiche di interfaccia OpenAPI 3.x, è possibile attuare una configurazione avanzata del tipo di validazione effettuato. Maggiori dettagli vengono forniti nella sezione *Validazione dei messaggi con OpenAPI 3.x*.

2.12.1 Gestione differente tra Richiesta e Risposta

Per default, il tipo di validazione dei messaggi impostato riguarderà sia le richieste che le risposte.

È possibile differenziare il tipo di validazione registrando le seguenti *Proprietà* sull'erogazione o sulla fruizione:

- *validation.request.enabled* o *validation.response.enabled* : consentono di modificare l'impostazione configurata rispettivamente per la richiesta o la risposta. I valori associabili alle proprietà sono “true”, “false” o “warning”.
- *validation.request.type* o *validation.response.type* : consentono di modificare il tipo di validazione. Per una validazione basata sulla Specifica dell'API utilizzare il valore “interface”, mentre per utilizzare solamente gli schemi indicare il valore “xsd”.
- *validation.request.acceptMtom* o *validation.response.acceptMtom* : consentono di modificare l'impostazione configurata per i messaggi che possiedono il formato MTOM. I valori associabili alle proprietà sono “true” o “false”.

2.12.2 Configurazione per API SOAP

È possibile configurare il tipo di validazione attuata su API SOAP registrando le seguenti *Proprietà* sull'erogazione o sulla fruizione:

- *validation.soapAction.enabled* : consente di disabilitare la verifica della SOAPAction. I valori associabili alle proprietà sono “true” o “false”;
- *validation.rpc.rootElementUnqualified.accept* : consente di indicare se devono essere accettate o meno richieste RPC il cui root-element non appartiene ad alcun namespace. Il comportamento di default del prodotto (configurabile anche a livello generale agendo sulla proprietà “org.openscoop2.pdd.validazioneContenutiApplicativi.rpc.rootElement.unqualified.accept” in govway_local.properties) è di accettare le richieste per essere compatibile con framework soap datati. I valori associabili alle proprietà sono “true” o “false”.

2.12.3 Configurazione per API REST

È possibile configurare il tipo di validazione attuata su API REST registrando le seguenti *Proprietà* sull'erogazione o sulla fruizione:

Nota: Tutte le proprietà configurate vengono verificate in AND tra di loro. Ad esempio è quindi possibile definire sia il codice http che il Content-Type per cui si desidera abilitare una validazione.

- *validation.emptyResponse.enabled* : consente di disabilitare la validazione della risposta in caso di payload http vuoto. I valori associabili alla proprietà sono “true” o “false”. Per default questo controllo è abilitato.
- *validation.problemDetails.enabled* : consente di disabilitare la validazione della risposta nel caso il payload http contenga un oggetto *Problem Details* come definito nella specifica *RFC 7807* (<https://tools.ietf.org/html/rfc7807>). I valori associabili alle proprietà sono “true” o “false”. Per default questo controllo è abilitato.
- *validation.returnValue* : consente di indicare i soli codici http per cui la validazione della risposta verrà effettuata. Possono essere associati differenti valori separati con la virgola, e ogni valore può essere un codice o un intervallo di codice (es. 200-299,404). Per default viene verificato qualsiasi codice http.
- *validation.returnValue.not* : consente di impostare una validazione della risposta solamente per i messaggi che non corrispondono ai codici http definiti nella proprietà “validation.returnValue”.
- *validation.contentType* : consente di indicare i soli Content-Type per cui la validazione della risposta verrà effettuata. Possono essere associati differenti Content-Type separati con la virgola, e possono essere utilizzati anche i tipi speciali “<type>/*” e “*/*” (es. text/xml,application/*). Per default viene verificato qualsiasi Content-Type.
- *validation.contentType.not* : consente di impostare una validazione della risposta solamente per i messaggi che non corrispondono ai Content-Type definiti nella proprietà “validation.contentType”.

Nota: Per la validazione dei messaggi con specifiche di interfaccia OpenAPI 3.x, è possibile attuare una configurazione avanzata del tipo di validazione effettuato. Maggiori dettagli vengono forniti nella sezione *Validazione dei messaggi con OpenAPI 3.x*.

2.12.4 Opzioni Avanzate

È possibile modificare l'engine di validazione registrando le seguenti *Proprietà* sull'erogazione o sulla fruizione:

- *validation.buffer.enabled*: consente di abilitare o disabilitare il buffer che preserva i dati letti dallo stream. Se l'opzione viene disabilitata, il contenuto inoltrato al backend verrà ottenuto serializzando l'oggetto costruito in seguito alla lettura dello stream (es. serializzazione dell'elemento DOM in xml). I valori associabili alle proprietà sono “true” o “false”.

2.13 Caching Risposta

In GovWay è possibile abilitare il salvataggio delle risposte in una cache sia globalmente, in modo che sia attivo per tutte le APIs, che singolarmente sulla singola erogazione o fruizione. Questa funzionalità permette ad un backend server di non dover riprocesare le stesse richieste più volte.

Tramite il collegamento *Caching Risposta*, presente nella sezione di configurazione della specifica erogazione/fruizione, è possibile agire sulla configurazione di tale funzionalità. L'impostazione permette di ridefinire la configurazione globale; i campi del form sono i medesimi descritti nella configurazione globale (sezione *Caching Risposta*).

2.14 Sicurezza a livello del messaggio

Tramite il collegamento *Sicurezza Messaggio*, presente nella sezione di configurazione della specifica erogazione/fruizione, è possibile impostare criteri di elaborazione dei messaggi in transito, attuati dal gateway, al fine di gestire i meccanismi di sicurezza previsti a livello del messaggio.

Il form presenta inizialmente lo *Stato* disabilitato. Per abilitare la sicurezza, impostare il valore dello stato su abilitato e confermare con il pulsante *Invia*. Appariranno gli elementi *Richiesta* e *Risposta*, come nella figura seguente.

Il form consente di selezionare uno schema di sicurezza, tra quelli disponibili, da applicare al messaggio di richiesta ed a quello di risposta. Gli schemi di sicurezza applicabili cambiano in base alla tipologia del messaggio sul quale si applica.

Per la gestione della sicurezza sul messaggio di richiesta, nel caso di una erogazione, il gateway agisce con il ruolo *Receiver* che comporta la seguente casistica:

- *Nel caso del protocollo SOAP*:
 - *WSSec Signature*, in ricezione si attende un messaggio firmato; l'azione è quella di verificare la firma presente
 - *WSSec Decrypt*, il messaggio ricevuto verrà decifrato
 - *WSSec SAML Token*, si attende un messaggio contenente una asserzione SAML; viene effettuata la verifica dell'asserzione presente.
 - *WSSec Username Token*, viene effettuata la validazione del token di autenticazione
 - *WSSec Timestamp*, se è prevista una scadenza all'interno del timestamp presente nel messaggio, se ne verificherà la validità
- *Nel caso del protocollo REST*
 - *JWT Decrypt*: il messaggio JSON ricevuto viene decifrato.
 - *JWT Verifier Signature*: al messaggio JSON ricevuto viene verificata la firma.

Erogazioni > Configurazioni di HelloPortType:1 (EnteInterno) > **Sicurezza Messaggio di Default**

Message-Security

Stato

Richiesta

Schema Sicurezza

Risposta

Schema Sicurezza

Invia **Cancella**

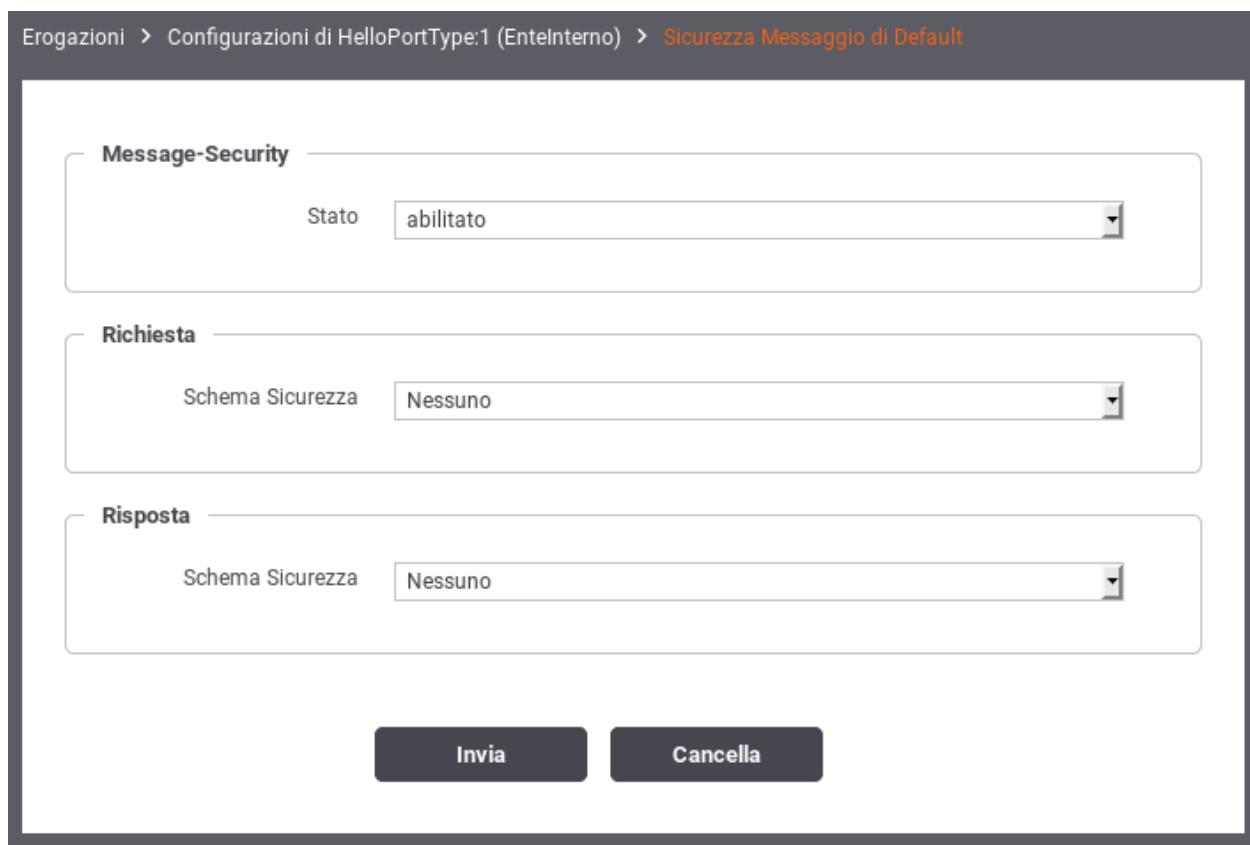


Figure2.115: Abilitazione Sicurezza Messaggio

- *XML Decrypt*: il messaggio XML ricevuto viene decifrato.
- *XML Verifier Signature*: al messaggio XML ricevuto viene verificata la firma.

Per la gestione della sicurezza sul messaggio di risposta, nel caso di una erogazione, il gateway agisce con il ruolo *Sender* che comporta la seguente casistica:

- *Nel caso del protocollo SOAP*:
 - *WSSec Signature*, il messaggio verrà firmato
 - *WSSec Encrypt*, il messaggio verrà cifrato
 - *WSSec SAML Token*, sul messaggio verrà inserita una asserzione SAML
 - *WSSec Username Token*, il messaggio verrà arrichito di un token di autenticazione
 - *WSSec Timestamp*, il messaggio verrà arrichito di una informazione temporale (tipicamente utilizzato insieme alla firma del messaggio)
- *Nel caso del protocollo REST*:
 - *JWT Encrypt*: il messaggio JSON di risposta viene cifrato prima dell'invio.
 - *JWT Signature*: il messaggio JSON di risposta viene firmato prima dell'invio.
 - *XML Encrypt*: il messaggio XML di risposta viene cifrato prima dell'invio.
 - *XML Signature*: il messaggio XML di risposta viene firmato prima dell'invio.

Nota: Si tenga presente che, nel caso di una fruizione, il ruolo del gateway si inverte diventando *Sender* nel caso della richiesta e *Receiver* nel caso della risposta. Gli schemi di sicurezza disponibili, nel caso della fruizione, rimangono quelli già descritti per Sender e Receiver.

2.15 Trasformazioni

Tra le attività di elaborazione, svolte dal gateway sui flussi di comunicazione in ingresso e uscita, vi è la possibilità di applicare delle *Regole di Trasformazione* che consentono di modificare dinamicamente i contenuti in transito prima che vengano instradati alla relativa destinazione.

2.15.1 Valori dinamici

Le regole di trasformazione possono avvalersi di un contesto di risorse, con valori aggiornati dinamicamente dal gateway, cui attingere per le trasformazioni da attuare. Tali risorse sono utilizzabili quando si procede con la definizione di una regola di trasformazione. Elenchiamo le risorse disponibili:

- *header:NAME* : valore dell'header http, corrispondente all'identificativo NAME, della richiesta.
- *query:NAME* : valore di un parametro della url di invocazione, corrispondente all'identificativo NAME.
- *form:NAME* : valore di un parametro della form, corrispondente all'identificativo NAME.
- *urlRegExp:EXPR* : applicazione di un'espressione regolare, rappresentata dal valore EXPR, alla url di invocazione (l'espressione deve avere un match con l'intera url).
- *xPath:EXPR* : applicazione di un'espressione XPath, rappresentata dal valore EXPR, alla richiesta xml (o soap).
- *jsonPath:EXPR* : applicazione di un'espressione jsonPath, rappresentata dal valore EXPR, alla richiesta json.
- *transaction:id* : l'identificativo UUID della transazione corrente.

- *date:FORMAT* : la data di elaborazione del messaggio; il formato fornito deve essere conforme a quanto richiesto dalla classe java “java.text.SimpleDateFormat” (es. \${date:yyyyMMdd_HHmmssSSS})
- *busta:FIELD* : accesso alle informazioni proprie del profilo di interoperabilità utilizzato; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.protocol.sdk.Busta” (ad es. per il mittente usare *busta.mittente*)
- *property:NAME*: accesso alle proprietà contenute nella traccia (ad esempio l’identificativo SDI); Il valore “NAME” indica il nome della proprietà da utilizzare.
- *tokenInfo:FIELD* : accesso ai claim di un token precedentemente validato; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.pdd.core.token.InformazioniToken” (es. per ottenere il valore del claim “sub” usare \${tokenInfo:sub})
- *tokenClient:FIELD* : identità dell’applicativo client identificato tramite il clientId presente nel token; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.core.id.IDServizioApplicativo” (es. per ottenere il nome dell’applicativo usare \${tokenClient:nome})
- *aa:FIELD* : consente di accedere agli attributi recuperati tramite Attribute Authority; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.pdd.core.token.attribute_authority.InformazioniAttributi” (es. per ottenere il valore dell’attributo “attr1” usare \${aa:attributes[attr1]}, se configurata solamente 1 A.A., altrimenti usare \${aa:attributes[nomeAttributeAuthority][attr1]})
- *transportContext:FIELD* : accesso ai dati della richiesta http; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.utils.transport.http.HttpServletTransportRequestContext” (es. per il principal usare \${transportContext:credential.principal})
- *securityToken:FIELD* : permette di accedere alle informazioni relative ai certificati ed ai security token presenti nella richiesta; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.protocol.sdk.SecurityToken” (es. per accedere al CN del certificato presente nel token ModI “Authorization” usare \${securityToken:authorization.certificate.subject.info(CN)})
- *integration:FIELD* : permette di accedere ai claim di un token di integrazione; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.pdd.core.dynamic.InformazioniIntegrazione” (es. per ottenere il valore del claim “claimCustom” usare \${integration:info[claimCustom]}). Maggiori informazioni sulla funzionalità sono disponibili nella sezione “[Scambio di informazioni tramite un token JSON](#)”.
- *config:NAME* : accesso alle proprietà configurate per l’API; il valore “NAME” indica la proprietà desiderata
- *clientApplicationConfig:NAME* : accesso alle proprietà configurate nell’applicativo fruitore; il valore “NAME” indica la proprietà desiderata
- *clientOrganizationConfig:NAME* : accesso alle proprietà configurate nel soggetto fruitore; il valore “NAME” indica la proprietà desiderata
- *providerOrganizationConfig:NAME* : accesso alle proprietà configurate nel soggetto erogatore; il valore “NAME” indica la proprietà desiderata
- *tokenClientApplicationConfig:NAME* : permette di accedere alla proprietà, configurata nell’applicativo client identificato tramite il clientId presente nel token, con nome “NAME”
- *tokenClientOrganizationConfig:NAME* : permette di accedere alla proprietà, configurata nel soggetto proprietario dell’applicativo client identificato tramite il clientId presente nel token, con nome “NAME”
- *dynamicConfig:FIELD* : permette di accedere alle proprietà degli attori coinvolti nella richiesta (api, applicativi, soggetti); il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.pdd.core.dynamic.DynamicConfig”; maggiori informazioni sulla funzionalità sono disponibili nella sezione “[Accesso alle proprietà delle entità del Registro](#)”.

- *request:FIELD* : permette di accedere al contenuto della richiesta; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.pdd.core.dynamic.ContentReader” (es. per ottenere il digest dell’attachment usare \${request:part.attachmentByIndex(0).contentBase64Digest(SHA-256)})
- *system:NAME* : valore associato alla proprietà di sistema, indicata nella configurazione generale, con nome “NAME”
- *env:NAME* : valore associato alla variabile di sistema con nome “NAME”
- *java:NAME* : valore associato alla variabile java con nome “NAME”
- *envj:NAME* : valore associato alla variabile di sistema o java con nome “NAME”; la variabile viene cercata prima come variabile di sistema e, se non presente, come variabile della jvm

Per le risposte sono inoltre disponibili anche le seguenti risorse:

- *headerResponse:NAME*: valore dell’header http, corrispondente all’identificativo NAME, della risposta.
- *xPathResponse:EXPR*: applicazione di un’espressione XPath, rappresentata dal valore EXPR, alla risposta xml (o soap).
- *jsonPathResponse:EXPR*: applicazione di un’espressione jsonPath, rappresentata dal valore EXPR, alla risposta json.
- *dateResponse:FORMAT*: la data di elaborazione della risposta; il formato fornito deve essere conforme a quanto richiesto dalla classe java “java.text.SimpleDateFormat” (es. \${date:yyyyMMdd_HHmmssSSS})
- *integrationResponse:FIELD* : permette di accedere ai claim di un token di integrazione; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.pdd.core.dynamic.InformazioniIntegrazione” (es. per ottenere il valore del claim “claimCustom” usare \${integrationResponse:info[claimCustom]}). Maggiori informazioni sulla funzionalità sono disponibili nella sezione “*Scambio di informazioni tramite un token JSON*”.
- *response:FIELD* : permette di accedere al contenuto della risposta; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.pdd.core.dynamic.ContentReader” (es. per ottenere il digest dell’attachment usare \${response:part.attachmentByIndex(0).contentBase64Digest(SHA-256)})

L’utilizzo dei suddetti elementi, come placeholder all’interno di template, comporta l’automatica sostituzione con il valore attuale a runtime da parte del gateway.

La sintassi per accedere le proprietà dinamiche sopraelencate è differente in base allo specifico contesto di utilizzo. Se si tratta di un testo interpretato direttamente da GovWay le proprietà saranno direttamente accessibili utilizzando il seguente formato:

- \${header:NAME} o \${headerResponse:NAME}
- \${query:NAME}
- \${form:NAME}
- \${xPath:EXPR} o \${xPathResponse:EXPR}
- \${jsonPath:EXPR} o \${jsonPathResponse:EXPR}
- \${urlRegExp:EXPR}
- \${transaction:id}
- \${date:FORMAT} o \${dateResponse:FORMAT}
- \${busta:FIELD}
- \${property:NAME}
- \${tokenInfo:FIELD}

- \${tokenClient:FIELD}
- \${aa:FIELD}
- \${transportContext:FIELD}
- \${securityToken:FIELD}
- \${integration:FIELD} o \${integrationResponse:FIELD}
- \${config:NAME}
- \${clientApplicationConfig:NAME}
- \${clientOrganizationConfig:NAME}
- \${providerOrganizationConfig:NAME}
- \${tokenClientApplicationConfig:NAME}
- \${tokenClientOrganizationConfig:NAME}
- \${dynamicConfig:FIELD}
- \${request:FIELD} o \${response:FIELD}
- \${system:NAME}
- \${env:NAME}
- \${java:NAME}
- \${envj:NAME}

Nei casi in cui il testo della trasformazione è interpretato da framework esterni (quali Freemarker o Velocity) le proprietà vengono rese disponibili da Govway inizializzando una mappa contenente i valori come oggetti. In questo caso le chiavi della mappa sono le seguenti (tra parentesi sono indicati i tipi di dato corrispondenti):

- header o headerResponse (java.util.Map<String, String>); in caso di molteplici header con stesso nome è disponibile la variabile headerValues o headerResponseValues (java.util.Map<String, List<String>>)
- query (java.util.Map<String, String>); in caso di molteplici parametri con stesso nome è disponibile la variabile queryValues (java.util.Map<String, List<String>>)
- form (java.util.Map<String, String>); in caso di molteplici parametri con stesso nome è disponibile la variabile formValues (java.util.Map<String, List<String>>)
- xPath o xPathResponse (org.openspcoop2.pdd.core.dynamic.PatternExtractor)
- jsonPath o jsonPathResponse (org.openspcoop2.pdd.core.dynamic.PatternExtractor)
- urlRegExp (org.openspcoop2.pdd.core.dynamic.URLRegExpExtractor)
- transactionId (java.lang.String)
- date (java.util.Date)
- busta (org.openspcoop2.protocol.sdk.Busta)
- property (java.util.Map<String, String>)
- tokenInfo (org.openspcoop2.pdd.core.token.InformazioniToken)
- tokenClient (org.openspcoop2.core.id.IDServizioApplicativo)
- aa (org.openspcoop2.pdd.core.token.attribute_authority.InformazioniAttributi)
- transportContext (org.openspcoop2.utils.transport.http.HttpServletTransportRequestContext)
- securityToken (org.openspcoop2.protocol.sdk.SecurityToken)

- integration o integrationResponse (org.openspcoop2.pdd.core.dynamic.InformazioniIntegrazione)
- config (java.util.Map<String, String>)
- clientApplicationConfig (java.util.Map<String, String>)
- clientOrganizationConfig (java.util.Map<String, String>)
- providerOrganizationConfig (java.util.Map<String, String>)
- tokenClientApplicationConfig (java.util.Map<String, String>)
- tokenClientOrganizationConfig (java.util.Map<String, String>)
- dynamicConfig (org.openspcoop2.pdd.core.dynamic.DynamicConfig)
- request o response (org.openspcoop2.pdd.core.dynamic.ContentExtractor)
- system (org.openspcoop2.pdd.core.dynamic.PropertiesReader)
- env (org.openspcoop2.pdd.core.dynamic.PropertiesReader)
- java (org.openspcoop2.pdd.core.dynamic.PropertiesReader)
- envj (org.openspcoop2.pdd.core.dynamic.PropertiesReader)

Nel caso di utilizzo di template “Freemarker” o “Velocity” sono disponibili i seguenti ulteriori oggetti:

- class; permette di definire classi. L'utilizzo varia a seconda del tipo di template engine:
 - velocity: class.forName(«my.package.name»)
 - freemarker: class[«my.package.name»]
- new; permette di istanziare una classe. L'utilizzo varia a seconda del tipo di template engine:
 - velocity: new.instance(«my.package.name»,»Parametro1»,»ParametroN»)
 - freemarker: new(«my.package.name»,»Parametro1»,»ParametroN»)
- attachments (org.openspcoop2.pdd.core.dynamic.AttachmentsReader); consente di ottenere gli allegati registrati sull'API
- context (java.util.Map<String, Object>); permette di accedere al contesto della richiesta.
- errorHandler (org.openspcoop2.pdd.core.dynamic.ErrorHandler); permette di generare risposte personalizzate che segnalano l'impossibilità di proseguire la trasformazione.

Nel caso di utilizzo di template “ZIP”, “TGZ” o “TAR” sono disponibili le seguenti le proprietà dinamiche, interpretate direttamente da GovWay, utilizzabili per accedere a parti della richiesta o della risposta:

- \${content} : payload http del messaggio
- \${soapEnvelope} : soap envelope del messaggio
- \${soapBody} : contenuto del soap body
- \${attachment[index]} : attachment presente in un messaggio multipart alla posizione indicata dall'intero “index”
- \${attachmentId[id]} : attachment presente in un messaggio multipart che possiede il Content-ID indicato

2.15.2 Trasformazione

Nel contesto della configurazione specifica di una erogazione o di una fruizione si può accedere alla funzionalità «Trasformazioni» per inserire una lista di definizioni che applicano trasformazioni ai flussi in entrata e/o uscita. Le trasformazioni create hanno la struttura di una lista ordinata e a ciascun elemento della lista è associato un insieme di criteri di applicabilità. La logica del gateway è quella di analizzare le trasformazioni nell'ordine della lista, selezionando la prima di esse i cui criteri di applicabilità sono tutti soddisfatti.

Tramite il pulsante *Aggiungi* è possibile inserire una nuova trasformazione (Fig. 2.116).

Note: (*) Campi obbligatori

Trasformazione

Nome *

Applicabilità

Risorse Qualsiasi

Content Type

Pattern

SALVA

Figure 2.116: Nuova Trasformazione

La creazione di una trasformazione richiede che vengano inseriti i seguenti dati:

- Nome: identificativo che rappresenta il nome assegnato alla trasformazione
- Applicabilità: sono i campi che vanno a comporre il criterio di applicabilità della trasformazione:
 - Risorse/Azioni: le operazioni sulle quali è applicabile la trasformazione.
 - Content-Type: i content-type sui quali è applicabile la trasformazione.
 - Pattern: il pattern inserito viene confrontato con il messaggio di richiesta del flusso di comunicazione al fine di verificare l'eventuale match. Il pattern può essere espresso nella sintassi «XPath», nel caso di messaggi XML, o JSONPath, nel caso di messaggi JSON.

Le trasformazioni create sono visualizzate nella forma di elenco ordinato (Fig. 2.117). L'icona iniziale di ciascun elemento consente di modificarne la posizione.

Ciascuna regola elencata visualizza i dati che sono stati forniti come criterio di applicabilità. A quelli inseriti in fase di creazione si aggiungono i Soggetti e gli Applicativi, che possono essere forniti accedendo i rispettivi collegamenti.

	Nome	Risorse	Content Type	Pattern	Soggetti	Applicativi
■	Trasformazione Delete	DELETE /api/{nome}/{versione}, DELETE /api/{nome}/{versione}/allegati/{nome_allegato}, DELETE /api/{nome}/{versione}/risorse/{nome_risorsa}, DELETE /api/{nome}/{versione}/servizi/{nome_servizio}, DELETE ...	application/json		Soggetti (0)	Applicativi (0)
■	Trasformazione Standard	Qualsiasi	application/json		Soggetti (0)	Applicativi (0)

Figure2.117: Lista regole di trasformazione

I soggetti/applicativi associati ad una regola saranno confrontati con l'identità del soggetto/applicativo mittente di ciascuna richiesta, identificato tramite l'autenticazione trasporto e/o token.

Accedendo il dettaglio di una regola di trasformazione vengono presentate le due sezioni:

- Trasformazione: per aggiornare il nome o i criteri di applicabilità.
- Regole di Trasformazione: per aggiornare le regole di trasformazione attuate sulla richiesta e sulla risposta.

Regole di Trasformazione della Richiesta

Selezionando il collegamento «Richiesta», nel riquadro delle Regole di Trasformazione, si procede con la definizione formale della trasformazione attuata sulle richieste in ingresso sulle quali è applicabile la trasformazione corrente. Le trasformazioni possono essere applicate sia a livello del trasporto che del contenuto, come mostrano le sezioni visualizzate in Fig. 2.118.

A livello del trasporto è possibile applicare trasformazioni sugli «HTTP Headers», selezionando l'omonimo collegamento e quindi aggiungendo le operazioni da effettuare (Fig. 2.119).

Ciascuna operazione può essere selezionata tra le seguenti:

- add: per aggiungere un nuovo header specificando successivamente nome e valore
- delete: per eliminare un header indicandone successivamente il nome
- update: per modificare un header indicandone successivamente il nome ed il nuovo valore
- updateOrAdd: per modificare un header indicandone successivamente il nome ed il nuovo valore. Nel caso l'header non si presente, verrà aggiunto.

Nota: i valori specificati per gli header http possono contenere le proprietà dinamiche descritte nella sezione *Valori dinamici*. Il campo “Identificazione Fallita” permette di definire il comportamento del Gateway quando non riesce a risolvere parti dinamiche contenute nel valore indicato. Le configurazioni utilizzabili sono:

Trasformazione

Trasporto

[HTTP Headers \(0\)](#)

[URL Parameters \(0\)](#)

Contenuto

Abilitato

SALVA



Figure2.118: Regola di trasformazione della richiesta

HTTP Header

Operazione *

Nome *

Valore * ⓘ

Identificazione Fallita ⓘ

SALVA

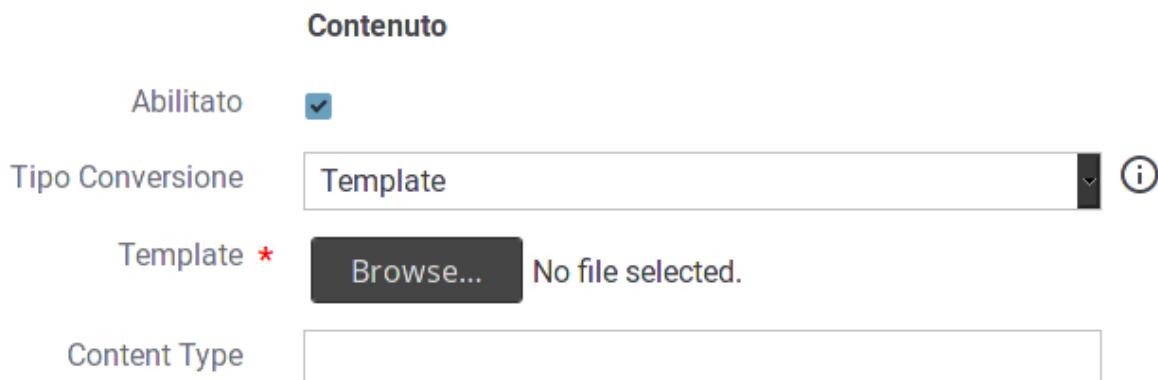


Figure2.119: Operazioni sugli Header HTTP

- Termina con errore: la transazione termina con un errore che riporta la fallita risoluzione della parte dinamica indicata per il valore;
- Continua senza header: la transazione continua senza completare la gestione dell'header.

Sempre a livello del trasporto è possibile applicare trasformazioni anche sui parametri presenti nella Query String, selezionando il collegamento «URL Parameters». La modalità di configurazione è del tutto analoga a quanto appena descritto per gli Header HTTP.

Abilitando l'opzione sul Contenuto è possibile procedere con la definizione di operazioni sul contenuto della richiesta (Fig. 2.120).



The screenshot shows a configuration interface for 'Content' (Contenuto). It includes a checkbox for 'Abilitato' (Enabled) which is checked. The 'Tipo Conversione' (Conversion Type) is set to 'Template', indicated by a dropdown menu with an 'i' icon. Below this, there is a 'Template' field with a 'Browse...' button and a message 'No file selected.'. A 'Content Type' field is also present.

Figure 2.120: Modifica del Contenuto della Richiesta

Per la modifica del contenuto della richiesta devono essere forniti i seguenti dati:

- Tipo Conversione: indica il tipo di trasformazione da applicare al contenuto. Si può scegliere una tra le seguenti opzioni:
 - HTTP Payload Vuoto: opzione presente nel caso REST. Il contenuto della richiesta diventa un payload http vuoto.
 - SOAP Body Vuoto: opzione presente nel caso SOAP. Il contenuto della richiesta diventa un messaggio SOAP con SoapBody vuoto.
 - Template: il contenuto della richiesta viene assegnato utilizzando il template fornito in configurazione, che può contenere parti dinamiche definite tramite una sintassi proprietaria di GovWay.
 - Freemarker Template: il contenuto della richiesta viene assegnato utilizzando il template «Freemarker» (<https://freemarker.apache.org/>) fornito in configurazione.
 - Freemarker Template (Archivio Zip): il file fornito deve essere un archivio zip contenenti dei files che rispettano la sintassi del template engine «Freemarker». Viene richiesta la presenza, all'interno dell'archivio zip, di un file indice che possieda il nome «index.ftl».
 - Velocity Template: il contenuto della richiesta viene assegnato utilizzando il template «Velocity» (<http://velocity.apache.org/>) fornito in configurazione.
 - Velocity Template (Archivio Zip): il file fornito deve essere un archivio zip contenenti dei files che rispettano la sintassi del template engine «Velocity». Viene richiesta la presenza, all'interno dell'archivio zip, di un file indice che possieda il nome «index.vm».

- XSLT: il contenuto della richiesta viene modificato applicando la trasformazione XSLT fornita in configurazione. Questo metodo è applicabile nel caso di messaggi XML o SOAP.
- ZIP Compressor: il contenuto della richiesta verrà trasformato in un archivio zip il cui contenuto viene definito dal file fornito che deve contenere proprietà indicate come nome=valore in ogni linea. Il nome della proprietà corrisponde all'entry name all'interno dell'archivio (es. dir/subDir/entryName1). Il valore della proprietà corrisponde al contenuto dell'entry. È possibile selezionare parti del messaggio, per associarle come contenuto dell'entry, utilizzando le espressioni dinamiche risolte a runtime dal Gateway (sezione *Valori dinamici*).
- TGZ Compressor: il contenuto della richiesta verrà trasformato in un archivio tgz il cui contenuto è definito tramite il file fornito che deve possedere la medesima struttura descritta per il tipo “ZIP”.
- TAR Compressor: il contenuto della richiesta verrà trasformato in un archivio tar il cui contenuto è definito tramite il file fornito che deve possedere la medesima struttura descritta per il tipo “ZIP”.
- Template: nei casi che lo prevedono, con questo elemento si fornisce il template da utilizzare per ottenere il nuovo contenuto della richiesta.
- Content-Type: opzionalmente, tramite questo elemento, è possibile assegnare un content-type alla richiesta modificata.

Nota: i template possono contenere le proprietà dinamiche descritte nella sezione *Valori dinamici*. La sintassi adottata dipende dal template. Una finestra di help contestuale, presente nell'interfaccia, guiderà l'utente nell'applicazione della sintassi corretta.

Conversione da REST a SOAP

Una particolare trasformazione del contenuto della richiesta è quella di convertire il formato da REST a SOAP. Questa funzionalità si ottiene abilitando la sezione «Trasformazione SOAP», presente nel caso di servizi REST. I dati da fornire per la configurazione sono (Fig. 2.121):

- Versione: selezione della versione del protocollo SOAP
- SOAP Action: indicazione della SOAP Action da utilizzare
- Imbustamento SOAP: se il messaggio ottenuto con le operazioni di trasformazione applicate non è in formato SOAP è possibile decidere di far generare al gateway gli elementi di imbustamento. Le opzioni possibili sono:
 - Disabilitato: nessun imbustamento.
 - Utilizza contenuto come SOAP Body: il contenuto attuale viene utilizzato come SOAP Body nel contesto dell'envelope creato.
 - Utilizza contenuto come Attachment: il contenuto attuale viene inserito come attachment relativo al messaggio SOAP generato. Se viene selezionata questa opzione dovranno essere forniti ulteriori dati, quali:
 - * Content Type Attachment: è possibile specificare un Content-Type per l'attachment.
 - * SOAP Body: stabilire quale deve essere il contenuto del SOAP Body. Per questo punto si procede analogamente a quanto già descritto per la trasformazione del contenuto principale della richiesta.

Trasformazione SOAP

Abilitato	<input checked="" type="checkbox"/>
Versione	SOAP 1.1
SOAP Action	test i
Imbustamento SOAP	Utilizza contenuto come SOAP Body v

Figure2.121: Conversione da REST a SOAP

Conversione da SOAP a REST

Una particolare trasformazione del contenuto della richiesta è quella di convertire il formato da SOAP a REST. Questa funzionalità si ottiene abilitando la sezione «Trasformazione REST», presente nel caso di servizi SOAP. I dati da fornire per la configurazione sono (Fig. 2.122):

- Path: path della risorsa cui deve fare riferimento il nuovo messaggio di richiesta REST-
- HTTP Method: il metodo HTTP utilizzato.

Trasformazione Rest

Abilitato	<input checked="" type="checkbox"/>
Path *	<input type="text"/> i
HTTP Method	GET v

Figure2.122: Conversione da SOAP a REST

Regole di Trasformazione della Risposta

Analogamente a quanto visto per la richiesta è possibile utilizzare il link «Risposte», nell'area «Regole di Trasformazione», per procedere con l'impostazione di regole per trasformare le risposte. A differenza del caso della richiesta, dove si può definire un unico meccanismo di trasformazione, in questo caso è possibile definire diverse regole di trasformazione basate sulla casistica delle risposte che si può presentare.

Quando si aggiunge una nuova regola di trasformazione della risposta si procede inserendo le seguenti informazioni (Fig. 2.123):

- Nome: nome assegnato alla regola di trasformazione
- Codice Risposta: Come criterio di applicabilità della regola, è possibile indicare il codice di risposta con le seguenti opzioni:

- Qualsiasi: qualunque codice di risposta ottenuto
- Singolo: si inserisce un specifico codice di risposta per il quale è applicabile la regola
- Intervallo: si inseriscono gli estremi dell'intervallo di codici di risposta per il quale è applicabile la regola
- Content-Type: criterio di corrispondenza con uno dei content-type indicati
- Pattern: espressione XPath o JsonPath da confrontare con il contenuto della risposta per un eventuale match

Erogazioni > api-config v1 (Ente) > Configurazione > Trasformazioni > Trasformazione Standard > Risposte > Aggiungi

Note: (*) Campi obbligatori

Trasformazione

Nome *

Applicabilità

Codice Risposta

Intervallo

Content Type

Pattern

SALVA

Figure2.123: Creazione regola di trasformazione della risposta

Le operazioni di trasformazione sulla risposta sono attuabili in maniera del tutto analoga a quanto già descritto per la richiesta. Diversamente dal caso della richiesta, al posto delle modifiche sui parametri della URL (non presenti nella risposta) è possibile modificare il Codice Risposta restituito.

Nota: Se sulla richiesta si è scelto di attuare la conversione da SOAP a REST, o viceversa, la trasformazione complementare risulterà disponibile anche nella configurazione della risposta.

2.16 Tracciamento

Il tracciamento è la funzionalità del gateway che comporta la registrazione dei dati relativi alle comunicazioni in transito riguardanti i servizi erogati e fruiti.

In questa sezione è possibile personalizzare la configurazione di default del tracciamento definita in accordo a quanto descritto in sezione *Tracciamento*. Le personalizzazioni inserite in questo contesto avranno validità per le sole comunicazioni riguardanti la specifica erogazione/fruizione (Fig. 2.124).

Erogazioni > PetStore v1 (ENTE) > Configurazione > **Tracciamento**

Tracciamento

Transazioni

Stato

Messaggi Diagnostici

Stato

Correlazione Applicativa

Richiesta
[Regole \(0\)](#)

Risposta
[Regole \(0\)](#)

SALVA

Figure2.124: Tracciamento per la singola erogazione/fruizione

Le sezioni presenti nella pagina sono:

- *Transazioni*: l'utente ha l'opzione per mantenere il default definito nella sezione di configurazione generale (sezione *Tracciamento* e dettaglio in *Registrazione della Transazione*) oppure ridefinirlo.

- *Messaggi Diagnostici*: l'utente ha l'opzione per mantenere il default definito nella sezione di configurazione generale (sezione *Tracciamento*) oppure ridefinire il criterio per la sola memorizzazione su Database.
- *Correlazione Applicativa*: consente di impostare delle regole per estrarre dai messaggi in transito, codici, riferimenti, o altri contenuti al fine di arricchire i dati tracciamento generati dal gateway (sezione *Correlazione Applicativa*).

2.17 Correlazione Applicativa

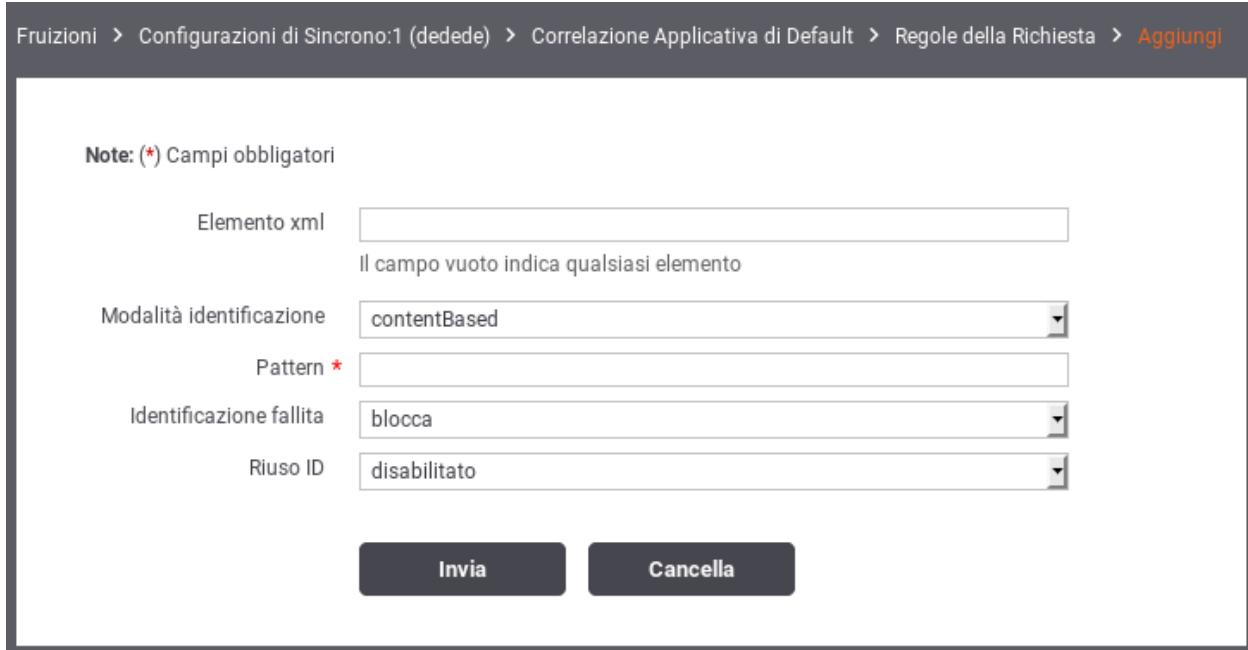
La funzione di *Correlazione Applicativa* consente al gateway che elabora il messaggio di richiesta, di estrarre un identificatore relativo al contenuto applicativo. L'identificatore, se presente, finisce nei sistemi di tracciamento e diagnostici, a completamento delle informazioni già presenti. I dati per configurare la correlazione applicativa consistono in un insieme di regole per l'estrazione di tale identificatore.

Nota: Lunghezza massima per l'identificativo estratto

L'identificativo applicativo estratto deve possedere una lunghezza non superiore ai 255 caratteri. Per maggiori informazioni si rimanda alla sezione *Lunghezza massima dell'identificativo di correlazione applicativa*.

Per accedere alla configurazione della correlazione applicativa, per una data erogazione/fruizione, si utilizza la sezione «Correlazione Applicativa» presente nell'ambito della configurazione del tracciamento di una fruizione/erogazione (sezione *Tracciamento*).

Utilizzando il collegamento *Regole*, presente nel riquadro della Richiesta o Risposta, si accede all'elenco delle regole di correlazione applicativa presenti. Premere il pulsante *Aggiungi* per aggiungere una nuova regola (Fig. 2.125)



The screenshot shows a configuration interface for creating a new rule. The path in the top navigation bar is: Fruizioni > Configurazioni di Sincrono:1 (dedede) > Correlazione Applicativa di Default > Regole della Richiesta > Aggiungi. A note at the top left says: "Note: (*) Campi obbligatori". The form fields are as follows:

- Elemento xml: A text input field containing an empty string, with a note below it: "Il campo vuoto indica qualsiasi elemento".
- Modalità identificazione: A dropdown menu set to "contentBased".
- Pattern *****: An empty text input field.
- Identificazione fallita: A dropdown menu set to "blocca".
- Riuso ID: A dropdown menu set to "disabilitato".

At the bottom are two buttons: "Invia" (Send) and "Cancella" (Cancel).

Figure 2.125: Creazione di una regola di correlazione applicativa

Per la creazione di una regola di correlazione applicativa si devono indicare i seguenti dati:

- *Elemento*: Questo dato serve per capire su quali messaggi è applicabile la regola di correlazione applicativa che si sta definendo. Lasciando il campo vuoto si intende che la regola si applica a tutti i messaggi. In alternativa è possibile indicare:
 - *Nome Azione o Risorsa*: il nome esatto dell’azione o della risorsa su cui verrà applicativa la regola
 - *HttpMethod e Path* (utilizzabile solo su API REST): metodo http e path di una risorsa dell’API; è possibile indicare qualsiasi metodo o qualsiasi path con il carattere speciale “*”. È inoltre possibile definire solamente la parte iniziale di un path attraverso lo “*”. Alcuni esempi:
 - * “POST /resource”
 - * “* /resource”
 - * “POST *”
 - * “* /resource/*”
 - *XPath o JSONPath*: Espressione che può rappresentare un XPath o JSONPath. Se l’espressione ha un match con il contenuto la regola verrà applicata
 - *LocalName dell’elemento xml*: in caso il messaggio sia un xml (soap o rest), è possibile indicare il local name del root element xml su cui verrà applicativa la regola

Nota: Se una richiesta non è applicabile a nessuna regola di correlazione applicativa, fino alla versione 3.3.13.p1 la richiesta terminava con l’errore «Identificativo di correlazione applicativa non identificato; nessun elemento tra quelli di correlazione definiti è presente nel body». Dalla versione successiva è stato modificato il default in modo da accettare la richiesta. È comunque possibile ripristinare il precedente comportamento come descritto nella sezione *Nessuna regola di correlazione applicativa applicabile*.

- *Modalità Identificazione*: rappresenta la modalità di acquisizione dell’identificatore applicativo. Può assumere i seguenti valori:
 - *Url di Invocazione*: il valore viene preso dalla url utilizzata dal servizio applicativo per l’invocazione. La regola per l’estrazione dalla url viene specificata tramite un’espressione regolare inserita nel campo “Espressione Regolare” (l’espressione deve avere un match con l’intera url).
 - *Header HTTP*: Il valore viene estratto dall’header di trasporto avente il nome indicato nel campo successivo.
 - *Contenuto*: Il valore viene estratto direttamente dal messaggio applicativo. La regola per l’estrazione dal messaggio è specificata tramite un’espressione XPath o JSONPath inserita nel campo “Pattern”.
 - *Header di Integrazione*: il valore viene estratto dall’header di integrazione GovWay presente nel valore della proprietà *IDApplicativo*.
 - *Template*: il valore è il risultato dell’istanziazione del template fornito rispetto ai dati della richiesta.
 - *Freemarker Template*: il valore è ottenuto tramite il processamento di un Freemarker Template.
 - *Velocity Template*: il valore è ottenuto tramite il processamento di un Velocity Template.
 - *Disabilitata*: l’identificatore applicativo non viene estratto. Questa opzione è utile quando si vuole disabilitare l’estrazione dell’id applicativo solo per specifici messaggi;
- *Identificazione Fallita*: azione da intraprendere nel caso fallisca l’estrazione dell’identificatore applicativo tramite la regola specificata. Nel caso sia stato indicato *blocca*, tali richieste non verranno accettate con restituzione di un errore al mittente;

Nota: L’estrazione di un identificativo nullo o stringa vuota viene trattato come processo d’identificazione fallito. È possibile modificare questo comportamento seguendo le indicazioni fornite nella sezione *Identificativo*

estratto nullo o stringa vuota.

- *Riuso ID*: opzione per abilitare/disabilitare il riuso dell'identificatore del messaggio (assegnato dal gateway) nel caso in cui vengano inviati messaggi con identificatori applicativi già processati in precedenza.

2.17.1 Lunghezza massima dell'identificativo di correlazione applicativa

L'identificativo applicativo estratto deve possedere una lunghezza non superiore ai 255 caratteri.

Nel caso l'identificativo di correlazione superi la massima lunghezza consentita il comportamento di default di GovWay varia a seconda del criterio di gestione dell'identificazione fallita configurato nella regola di correlazione:

- nel caso di gestione di tipo “blocca” la transazione termina con errore e nel diagnostico viene informato l'utente che è stata superata la massima lunghezza consentita;
- nel caso di gestione di tipo “accetta” la transazione termina con successo e non viene salvata alcun id di correlazione applicativa.

È possibile modificare i comportamenti di default precedentemente indicati abilitando il troncamento dell'identificativo estratto al fine di portare la sua lunghezza alla massima dimensione consentita. Per abilitare il troncamento è possibile registrare una delle seguenti *Proprietà* sull'erogazione o sulla fruizione (i valori associabili alle proprietà sono “true” o “false”):

- *correlation.request.truncate* o *correlation.response.truncate* : consentono di abilitare il troncamento rispettivamente per la richiesta o per la risposta;
- *correlation.truncate*: consente di abilitare il troncamento sia per la richiesta che per la risposta.

Sono inoltre disponibili altre proprietà che consentono una abilitazione a grana più fine sulla singola modalità di gestione:

- *correlation.request.blockIdentificationFailed.truncate* o *correlation.response.blockIdentificationFailed.truncate* : consentono di abilitare il troncamento, rispettivamente per la richiesta o per la risposta, solamente per la gestione di tipo “blocca”;
- *correlation.request.acceptIdentificationFailed.truncate* o *correlation.response.acceptIdentificationFailed.truncate* : consentono di abilitare il troncamento, rispettivamente per la richiesta o per la risposta, solamente per la gestione di tipo “accetta”;
- *correlation.blockIdentificationFailed.truncate* : consente di abilitare il troncamento, sia per la richiesta che per la risposta, solamente per la gestione di tipo “blocca”;
- *correlation.acceptIdentificationFailed.truncate* : consente di abilitare il troncamento, sia per la richiesta che per la risposta, solamente per la gestione di tipo “accetta”.

2.17.2 Nessuna regola di correlazione applicativa applicabile

Una richiesta non applicabile a nessuna regola di correlazione applicativa (vedi configurazione “*Elemento*” descritto in *Correlazione Applicativa*), fino alla versione 3.3.13.p1, terminava con l'errore «Identificativo di correlazione applicativa non identificato; nessun elemento tra quelli di correlazione definiti è presente nel body». Dalla versione successiva è stato modificato il default in modo da accettare la richiesta.

È possibile ripristinare il precedente comportamento registrando una delle seguenti *Proprietà* sull'erogazione o sulla fruizione (i valori associabili alle proprietà sono “true” o “false”):

- *correlation.request.ruleNotFound.abortTransaction*: (default:false) consente di terminazione con errore la transazione, in caso la richiesta non sia applicabile a nessuna regola di correlazione applicativa;

- *correlation.response.ruleNotFound.abortTransaction*: (default:false) consente di terminazione con errore, in caso la risposta non sia applicabile a nessuna regola di correlazione applicativa.

2.17.3 Identificativo estratto nullo o stringa vuota

L'estrazione di un identificativo nullo o stringa vuota viene trattato come processo d'identificazione fallito. È possibile modificare questo comportamento registrando una delle seguenti *Proprietà* sull'erogazione o sulla fruizione (i valori associabili alle proprietà sono “true” o “false”):

- *correlation.request.extractedIdentifierIsNull.abortTransaction*: (default:true) indicazione su come trattare un identificativo nullo estratto dalla richiesta;
- *correlation.response.extractedIdentifierIsNull.abortTransaction*: (default:true) indicazione su come trattare un identificativo nullo estratto dalla risposta;
- *correlation.request.extractedIdentifierIsEmpty.abortTransaction*: (default:true) indicazione su come trattare un identificativo “stringa vuota” estratto dalla richiesta;
- *correlation.response.extractedIdentifierIsEmpty.abortTransaction*: (default:true) indicazione su come trattare un identificativo “stringa vuota” estratto dalla risposta;

2.18 MTOM

Nei casi in cui il mittente e il destinatario si scambiano messaggi con allegati (nell'ambito del protocollo SOAP), utilizzando il protocollo MTOM, GovWay è in grado di gestire tali comunicazioni in modalità trasparente e quindi senza alcun intervento.

In altre situazioni è possibile sfruttare le funzionalità di GovWay per beneficiare delle ottimizzazioni del protocollo MTOM quando uno dei due interlocutori non è in grado di supportare tale protocollo, oppure per effettuare verifiche di congruità dei messaggi in transito basati su MTOM.

Nel caso di una erogazione, per il messaggio di richiesta, le opzioni disponibili sono:

- *disable*. Non viene svolta alcuna azione.
- *unpackaging*. In questo scenario il client fruitore invia dati binari nel formato MTOM ma l'erogatore non supporta tale formato. Il gateway effettua la trasformazione del messaggio inserendo i dati binari in modalità *Base64 encoded* prima che venga inviato al destinatario. Sulla risposta sarà effettuato il processo inverso.
- *verify*. Sia il fruitore che l'erogatore utilizzano MTOM ma si vogliono validare i messaggi. Il gateway effettua, tramite opportuni pattern xpath forniti, la validazione dei messaggi al fine di verificare la conformità del formato del messaggio rispetto a quanto atteso dall'erogatore.

Sempre nel caso di una erogazione, per il messaggio di risposta, le opzioni disponibili sono:

- *disable*. Non viene svolta alcuna azione.
- *packaging*. In questo scenario il client fruitore invia dati binari nella modalità *Base64 encoded* ma l'erogatore richiede il formato MTOM. Il gateway effettua la trasformazione del messaggio secondo il protocollo MTOM prima che venga inviato al destinatario. Sulla risposta sarà effettuato il processo inverso.
- *verify*. Analogo a quanto descritto per il messaggio di richiesta.

Nota: Nel caso si utilizzi la validazione dei contenuti, basata su xsd o wsdl, è possibile che la struttura MTOM non sia stata prevista negli schemi e quindi faccia fallire l'esito della stessa. In questo caso, quando si attiva la validazione è

necessario abilitare l'opzione *Accetta MTOM/XOP-Message* affinché il processo di validazione tenga conto del formato MTOM.

Nota: Nel caso di una fruizione, le opzioni di configurazione disponibili per la richiesta diventano quelle per la risposta e viceversa.

2.19 Registrazione Messaggi

La funzionalità consente di abilitare il salvataggio dei contenuti dei messaggi della richiesta e della risposta transiti su GovWay.

È possibile definire un criterio di registrazione dei messaggi differenziando tra Richiesta e Risposta e abilitando/disabilitando solo la comunicazione desiderata tra:

- *Ingresso*: il messaggio di richiesta o risposta nel momento in cui giunge sul gateway e quindi prima che venga sottoposto al processo di elaborazione previsto.
- *Uscita*: il messaggio di richiesta o risposta nel momento in cui esce dal gateway, per raggiungere il nodo successivo del flusso, e quindi dopo che è stato sottoposto al processo di elaborazione previsto.

Per ciascuno dei messaggi, su cui è stata abilitata la registrazione, è possibile scegliere gli elementi da registrare tra:

- *Headers*: vengono salvati gli header di trasporto (HTTP Headers) associati al messaggio;
- *Payload*: viene salvato il corpo del messaggio (HTTP Payload).

In Fig. 2.126 viene mostrata la pagina di configurazione.

Se a livello di erogazione o fruizione non viene attuata una configurazione specifica, la funzionalità di registrazione dei messaggi eredita la configurazione attuata a livello globale e descritta nella sezione *Registrazione Messaggi*.

In Fig. 2.127 viene mostrata la pagina di configurazione prima di procedere con una personalizzazione a livello di erogazione o fruizione.

Infine nella sezione *Definizione di “white-list” o “black-list” per gli header HTTP da registrare* viene descritto come configurare il prodotto, sia a livello globale che puntuale sulla specifica erogazione o fruizione, per definire delle blackList o delle whiteList rispetto agli header HTTP da registrare.

2.19.1 Definizione di “white-list” o “black-list” per gli header HTTP da registrare

È possibile configurare il prodotto, sia a livello globale che puntuale sulla specifica erogazione o fruizione, per definire delle blackList o delle whiteList rispetto agli header HTTP da registrare.

Nota: In caso di configurazione errata dove vengono definite entrambe le liste, la “white-list” ha priorità sulla “black-list”.

Sulla singola erogazione o fruizione è possibile attuare la configurazione registrando le seguenti *Proprietà* specifiche per il flusso desiderato, definendo una lista di header http separati da virgola:

- *registrazioneMessaggi.richiestaIngresso.whiteList* o *registrazioneMessaggi.richiestaIngresso.blackList*: consente di definire una lista per le richieste in ingresso;
- *registrazioneMessaggi.richiestaUscita.whiteList* o *registrazioneMessaggi.richiestaUscita.blackList*: configurazione per le richieste in uscita;

The screenshot shows the 'Registrazione Messaggi' configuration interface with three tabs: 'Generale', 'Richiesta', and 'Risposta'. Each tab contains a dropdown menu for 'Stato' and sub-sections for 'Ingresso' (Headers and Payload) or 'Uscita' (Headers and Payload). The 'Richiesta' and 'Risposta' tabs show dropdowns set to 'abilitato' (enabled) for all fields. The 'Generale' tab shows 'ridefinito' (redefined) selected in the 'Stato' dropdown.

Generale	Richiesta	Risposta
Stato: ridefinito	Stato: abilitato	Stato: abilitato
Ingresso		
Headers: abilitato	Headers: abilitato	Headers: disabilitato
Payload: abilitato	Payload: abilitato	Payload: disabilitato
Uscita		
Headers: disabilitato	Headers: disabilitato	Headers: abilitato
Payload: disabilitato	Payload: disabilitato	Payload: abilitato

Figure2.126: Personalizzazione della registrazione dei messaggi a livello di erogazione o fruizione

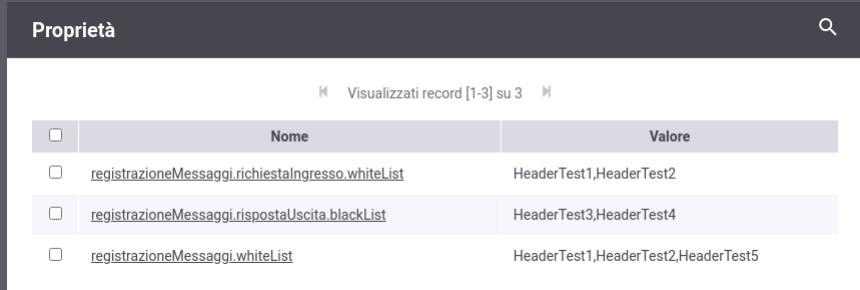
The screenshot shows the 'Registrazione Messaggi' configuration interface with the 'Generale' tab selected. The 'Stato' dropdown is set to 'default (disabilitato)' (disabled).

Figure2.127: Configurazione di default della registrazione dei messaggi a livello di erogazione o fruizione

- *registrazioneMessaggi.rispostaIngresso.whiteList* o *registrazioneMessaggi.rispostaIngresso.blackList*: consente di definire una lista per le risposte in ingresso;
- *registrazioneMessaggi.rispostaUscita.whiteList* o *registrazioneMessaggi.rispostaUscita.blackList*: configurazione per le risposte in uscita.

Sempre a livello di singola erogazione o fruizione è possibile attuare una configurazione che vale per qualsiasi flusso, registrando la seguente *Proprietà*:

- *registrazioneMessaggi.whiteList* o *registrazioneMessaggi.blackList*



Nome	Valore
registrazioneMessaggi.richiestaIngresso.whiteList	HeaderTest1,HeaderTest2
registrazioneMessaggi.rispostaUscita.blackList	HeaderTest3,HeaderTest4
registrazioneMessaggi.whiteList	HeaderTest1,HeaderTest2,HeaderTest5

Figure2.128: Personalizzazione degli header HTTP a livello di erogazione o fruizione

È inoltre possibile effettuare una configurazione simile a livello globale che verrà presa in esame solamente se per un flusso non esiste una configurazione specifica sull'erogazione o fruizione.

Per attuare la configurazione si deve agire sul file <directory-lavoro>/govway_local.properties registrando le seguenti proprietà:

- “white-list”; flussi specifici per le erogazioni:

```
org.openspcoop2.pdd.logger.dump.header.erogazioni.richiesta-ingresso.
  ↵whiteList=HDR1, ...,HDRN
org.openspcoop2.pdd.logger.dump.header.erogazioni.richiesta-uscita.
  ↵whiteList=HDR1, ...,HDRN
org.openspcoop2.pdd.logger.dump.header.erogazioni.risposta-ingresso.
  ↵whiteList=HDR1, ...,HDRN
org.openspcoop2.pdd.logger.dump.header.erogazioni.risposta-uscita.
  ↵whiteList=HDR1, ...,HDRN
```

- “black-list”; flussi specifici per le erogazioni:

```
org.openspcoop2.pdd.logger.dump.header.erogazioni.richiesta-ingresso.
  ↵blackList=HDR1, ...,HDRN
org.openspcoop2.pdd.logger.dump.header.erogazioni.richiesta-uscita.
  ↵blackList=HDR1, ...,HDRN
org.openspcoop2.pdd.logger.dump.header.erogazioni.risposta-ingresso.
  ↵blackList=HDR1, ...,HDRN
org.openspcoop2.pdd.logger.dump.header.erogazioni.risposta-uscita.
  ↵blackList=HDR1, ...,HDRN
```

- “white-list”; flussi specifici per le fruizioni:

```
org.openspcoop2.pdd.logger.dump.header.fruizioni.richiesta-ingresso.
  ↵whiteList=HDR1, ...,HDRN
org.openspcoop2.pdd.logger.dump.header.fruizioni.richiesta-uscita.
```

(continues on next page)

(continua dalla pagina precedente)

```

↳whiteList=HDR1, ..., HDRN
org.openspcoop2.pdd.logger.dump.header.fruizioni.risposta-ingresso.
↳whiteList=HDR1, ..., HDRN
org.openspcoop2.pdd.logger.dump.header.fruizioni.risposta-uscita.
↳whiteList=HDR1, ..., HDRN

```

- “black-list”; flussi specifici per le fruizioni:

```

org.openspcoop2.pdd.logger.dump.header.fruizioni.richiesta-ingresso.
↳blackList=HDR1, ..., HDRN
org.openspcoop2.pdd.logger.dump.header.fruizioni.richiesta-uscita.
↳blackList=HDR1, ..., HDRN
org.openspcoop2.pdd.logger.dump.header.fruizioni.risposta-ingresso.
↳blackList=HDR1, ..., HDRN
org.openspcoop2.pdd.logger.dump.header.fruizioni.risposta-uscita.
↳blackList=HDR1, ..., HDRN

```

- “white-list”; qualsiasi flusso per le erogazioni:

```
org.openspcoop2.pdd.logger.dump.header.erogazioni.whiteList=HDR1, ..., HDRN
```

- “black-list”; qualsiasi flusso per le erogazioni:

```
org.openspcoop2.pdd.logger.dump.header.erogazioni.blackList=HDR1, ..., HDRN
```

- “white-list”; qualsiasi flusso per le fruizioni:

```
org.openspcoop2.pdd.logger.dump.header.fruizioni.whiteList=HDR1, ..., HDRN
```

- “black-list”; qualsiasi flusso per le fruizioni:

```
org.openspcoop2.pdd.logger.dump.header.fruizioni.blackList=HDR1, ..., HDRN
```

- “white-list”; qualsiasi flusso valido sia per le fruizioni che per le erogazioni:

```
org.openspcoop2.pdd.logger.dump.header.whiteList
```

- “black-list”; qualsiasi flusso valido sia per le fruizioni che per le erogazioni:

```
org.openspcoop2.pdd.logger.dump.header.blackList
```

2.20 Proprietà

Ad una API è possibile associare una serie di proprietà consultabili da una qualsiasi delle funzionalità precedentemente descritte.

(Fig. 2.19).

Questa funzionalità è frequentemente utilizzata in combinazione con le *Trasformazioni* per poter permettere all’utente di configurarne il comportamento senza dover modificare e caricare un nuovo file template di trasformazione. All’interno di un template di trasformazione è possibile accedere alle proprietà tramite la sintassi “config” come descritto nella sezione *Valori dinamici*.

Proprietà		
Visualizzati record [1-2] su 2		
	Nome	Valore
<input type="checkbox"/>	nomeProprieta1	valoreProprieta1
<input type="checkbox"/>	nomeProprieta2	valoreProprieta2

ELIMINA **AGGIUNGI**

Figure2.129: Elenco di proprietà di una API

Le proprietà permettono inoltre di effettuare la configurazione di aspetti avanzati di una funzionalità che non rientrano nel suo normale utilizzo. Ad esempio è possibile differenziare il comportamento della validazione dei messaggi, tra richiesta e risposta, utilizzando le proprietà descritte nella sezione *Validazione dei messaggi*.

Profilo “ModI”

Il profilo “ModI” consente in maniera del tutto trasparente alle applicazioni interne al dominio, la conformità delle API (sia in fruizione che in erogazione) alle nuove *Linee Guida AGID di Interoperabilità* (<https://www.agid.gov.it/infrastrutture/sistema-pubblico-connettivita/il-nuovo-modello-interoperabilita>).

La struttura complessiva del processo di configurazione si mantiene analoga a quanto già descritto per il profilo API Gateway. Le differenze, con rispetto al profilo API Gateway, presentate in questa sezione, riguardano vincoli sulle scelte operabili dalla console e le informazioni di configurazione aggiuntive specifiche per la realizzazione degli scenari in accordo al Modello di Interoperabilità.

3.1 Concetti Preliminari

Il Modello di Interoperabilità mantiene sostanzialmente invariato il concetto di *dominio* di un’amministrazione rispetto a quanto prevedeva il precedente modello SPCoop. Resta quindi fondamentale individuare il perimetro d’azione delle interfacce dei servizi rispetto al sistema informativo dell’ente e i propri interlocutori.

Il concetto di dominio, che riveste particolare importanza nella gestione degli aspetti di sicurezza, si sposa perfettamente con i modelli di configurazione di GovWay dove è possibile attivare:

- *erogazioni di API*: richieste che provengono da amministrazioni esterne al dominio e sono dirette ad applicativi interni.
- *fruizioni di API*: richieste che provengono dagli applicativi interni del dominio e sono dirette verso altre amministrazioni.

La govwayConsole, all’atto della registrazione di Soggetti (Enti/Organizzazioni) e Applicativi (Sistemi/Applicazioni di un ente), consente di specificarne il *Dominio*, interno o esterno, al fine della corretta rappresentazione degli scenari di configurazione dei servizi.

Il profilo ModI prevede che i servizi siano basati su SOAP o REST fornendo sempre un descrittore formale delle interfacce basato su uno specifico IDL (Interface Description Language):

- WSDL 1.1 e successivi, per la descrizione delle interfacce SOAP
- OpenAPI 3.0 e successivi, per la descrizione delle interfacce REST

Nel processo di configurazione, tramite la gowayConsole, sono inoltre tenuti in considerazione tutti gli aspetti previsti dalle Linee Guida:

- *Pattern di Interazione*: definiscono la modalità con cui fruitore ed erogatore di un servizio interagiscono. Sono previsti i seguenti pattern:
 - *Bloccante*: il fruitore invia la richiesta e resta bloccato in attesa di ricevere la risposta, completa dei dati attesi, dall'erogatore
 - *Non Bloccante*: il fruitore non resta bloccato dopo aver inviato la richiesta, se non per ricevere una notifica di presa in carico. Per ottenere la risposta sarà necessario effettuare una distinta interazione, prevista nello scenario del servizio.
 - *Accesso CRUD*: pattern orientato alle risorse, utilizzabile solo su tecnologia REST, dove le API vengono utilizzate per eseguire operazioni di tipo CRUD - Create, Read, Update, Delete su risorse del dominio di interesse.
- *Sicurezza Canale*: gestione della sicurezza inherente il canale di comunicazione tra i domini fruitore ed erogatore. La specifica prevede i seguenti due pattern:
 - *ID_AUTH_CHANNEL_01 - Direct Trust Transport-Level Security*: comunicazione basata sul canale TLS dopo aver effettuato il trust del certificato X509 fornito dal dominio erogatore.
 - *ID_AUTH_CHANNEL_02 - Direct Trust mutual Transport-Level Security*: comunicazione basata sul canale TLS dopo aver effettuato il trust dei certificati X509, del fruitore e dell'erogatore, nella modalità di mutua autenticazione.
- *Sicurezza Messaggio*: gestione della sicurezza inherente lo scambio di informazioni tra le applicazioni agli estremi del flusso di comunicazione. I pattern di sicurezza previsti si distinguono per il caso SOAP e per quello REST:
 - *ID_AUTH_SOAP_01 o ID_AUTH_REST_01 - Direct Trust con certificato X.509 su SOAP o REST*: Tramite la validazione del certificato X509, inserito dall'applicazione mittente nel token di sicurezza della richiesta, l'applicativo destinatario verifica la corrispondenza delle identità e la validità del messaggio, prima di procedere con la produzione della risposta attraverso un trust tra fruitore e erogatore basato su certificati x509.
 - *ID_AUTH_REST_01 tramite la Piattaforma Digitale Nazionale Dati (PDND)*: con l'aggiornamento delle linee guida nella “Determinazione n. 128 del 23 maggio 2023”, viene indicato di utilizzare la PDND per ottenere un token conforme al pattern ID_AUTH_REST_01; la costituzione del trust avviene attraverso il materiale crittografico depositato sulla PDND applicando i profili di emissione dei voucher previsti.
 - *ID_AUTH_SOAP_02 o ID_AUTH_REST_02 - Direct Trust con certificato X.509 su SOAP o REST con unicità del messaggio/token*: estensione dei pattern precedenti con l'aggiunta di un meccanismo di filtro che impedisce il processamento di un messaggio duplicato.
 - *INTEGRITY_SOAP_01 o INTEGRITY_REST_01 - Integrità del payload del messaggio SOAP o REST*: pattern che estende i precedenti aggiungendo la gestione della firma del payload come verifica di integrità del messaggio ricevuto.
 - *INTEGRITY_REST_02 - Integrità del payload delle request REST in PDND*: simile al precedente pattern INTEGRITY_REST_01, assume che il trust avvenga tramite il materiale crittografico depositato sulla PDND applicando i profili di emissione dei voucher previsti. All'interno del token viene indicato l'identificativo della chiave pubblica (kid) associata alla chiave privata utilizzata dal client per firmare il token di integrità; identificativo kid generato dalla PDND e recuperabile dall'erogatore tramite le API messe a disposizione dalla PDND stessa.
 - *PROFILE_NON_REPUTATION_01 - Profilo per la non ripudiabilità della trasmissione*: estende i pattern di integrità allo scopo di fornire una conferma al fruitore da parte dell'erogatore della ricezione del contenuto della richiesta. Descrive inoltre la necessità di definire un arco temporale di persistenza dei messaggi utile per soddisfare l'opponibilità ai terzi.

- *Sicurezza Audit*: consente all’erogatore di identificare la specifica provenienza di ogni singola richiesta di accesso ai dati effettuata dal fruitore. Le Linee Guida definiscono 2 pattern utilizzabili sia per API REST che per API SOAP:
 - *AUDIT_REST_01- Inoltro dati tracciati nel dominio del Fruitore*: definisce la struttura del token di audit utilizzabile in alternativa o tramite un criterio di trust realizzato tramite il materiale crittografico depositato sulla PDND o tramite il trust diretto fruitore-erogatore attraverso l’utilizzo di certificati X509. Le Linee Guida indicano che l’erogatore e il fruitore devono individuare i claim da includere nel JWT di audit e suggeriscono i seguenti dati che dovranno essere presenti nel token generato dal fruitore, per ogni richiesta effettuata:
 - * userID, un identificativo univoco dell’utente interno al dominio del fruitore che ha determinato l’esigenza della request di accesso all’e-service dell’erogatore;
 - * userLocation, un identificativo univoco della postazione interna al dominio del fruitore da cui è avviata l’esigenza della request di accesso all’e-service dell’erogatore;
 - * LoA, livello di sicurezza o di garanzia adottato nel processo di autenticazione informatica nel dominio del fruitore.
 - *AUDIT_REST_02 - Inoltro dati tracciati nel dominio del Fruitore con correlazione*: pattern che estende il precedente aggiungendo la correlazione tra il token di autenticazione e il token di audit. Il pattern richiede un trust realizzato tramite il materiale crittografico depositato sulla PDND.
- *URL di Invocazione API*: le linee guida richiedono una indicazione esplicita della tecnologia utilizzata (REST o SOAP) e la versione. Le url con cui vengono esposte le API su GovWay soddisfano entrambi i requisiti.

Tutti questi concetti sono stati recepiti e gestiti nelle maschere di configurazione della govwayConsole, adottando il profilo ModI. Le sezioni seguenti illustrano in dettaglio gli elementi di configurazione integrativi rispetto al profilo API Gateway.

3.2 Sicurezza Canale

I pattern di sicurezza a livello del canale riguardano le modalità di trasporto dei messaggi tra il dominio fruitore e quello erogatore. La specifica tenica del Modello di Interoperabilità prevede, per questa tipologia, i seguenti due pattern:

- [ID_AUTH_CHANNEL_01] Direct Trust Transport-Level Security: comunicazione basata sul canale SSL dopo aver effettuato il trust del certificato X509 fornito dal dominio erogatore.
- [ID_AUTH_CHANNEL_02] Direct Trust mutual Transport-Level Security: comunicazione basata sul canale SSL dopo aver effettuato il trust dei certificati X509, del fruitore e dell’erogatore, nella modalità di mutua autenticazione.

Il concetto di ente/dominio, previsto dalle specifiche del Modello di Interoperabilità, viene riportato su quello di Soggetto nell’ambito delle entità di configurazione di GovWay.

Vediamo nelle sezioni seguenti come si possono effettuare le configurazioni per i pattern di sicurezza canale.

3.2.1 ID_AUTH_CHANNEL_01 - Direct Trust Transport-Level Security

Questo pattern di sicurezza prevede l'utilizzo del canale HTTPS, per le comunicazioni sul confine tra i due domini, con validazione del certificato dell'ente destinatario della comunicazione.

Descriviamo di seguito i passi di configurazione da effettuare:

- La creazione della relativa API prevede che nella sezione «Modi», elemento «Sicurezza Canale», venga selezionato il pattern «ID_AUTH_CHANNEL_01» come indicato in Fig. 3.1.

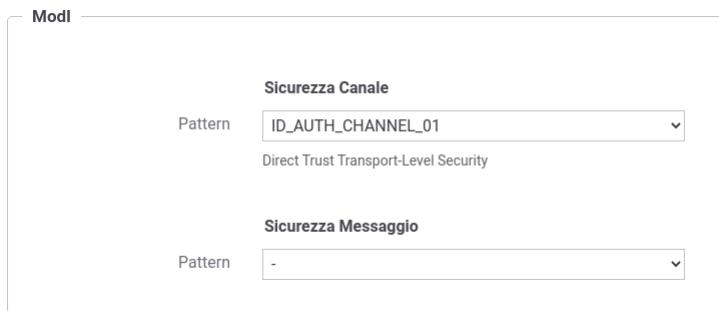


Figure3.1: Selezione del pattern «ID_AUTH_CHANNEL_01» per l'API

- Nel caso si voglia configurare una fruizione, le maschere di configurazione terranno conto degli aspetti di sicurezza sul canale garantendo che l'endpoint specificato nel connettore di uscita sia di tipo HTTPS, indipendentemente dal pattern adottato nella API (TLS sempre obbligatorio). L'autenticazione HTTPS può essere gestita opzionalmente da GovWay o, in alternativa, delegata alla configurazione della JVM sull'application server. Per la gestione in GovWay sono disponibili i campi per la configurazione HTTPS, lasciando opzionalmente la possibilità di impostare l'autenticazione client (vedi sez. [Autenticazione Https](#)).
- Nel caso si voglia configurare una erogazione, il pattern di sicurezza «ID_AUTH_CHANNEL_01» impatta sulla configurazione del Controllo Accessi, previsto nella configurazione specifica dell'erogazione:
 - La sezione «Autenticazione Canale» è impostata a «HTTPS» ammettendo il flag «Opzionale» (Fig. 3.2).

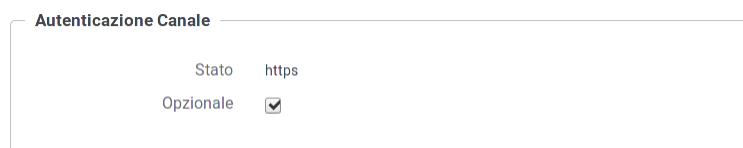


Figure3.2: Autenticazione Canale HTTPS con flag opzionale

- La sezione «Autorizzazione Canale» è per default disabilitata (Fig. 3.3). Abilitando tale sezione sarà possibile inserire i criteri di autorizzazione, come descritto nella sez. [Autorizzazione](#), con la differenza che in questo caso le politiche saranno riferite esclusivamente ai soggetti censiti in configurazione (e non gli applicativi, per i quali si rimanda alla sez. [Sicurezza Messaggio](#)).

Figure3.3: Autorizzazione Canale Disabilitata

3.2.2 ID_AUTH_CHANNEL_02 - Direct Trust mutual Transport-Level Security

Questo pattern di sicurezza prevede l'utilizzo del canale HTTPS con autenticazione client, per le comunicazioni sul confine tra i due domini, con reciproca validazione dei certificati degli enti in gioco.

Descriviamo di seguito i passi di configurazione da effettuare:

- La creazione della relativa API prevede che nella sezione «Modi», elemento «Sicurezza Canale», venga selezionato il pattern «ID_AUTH_CHANNEL_02» come indicato in Fig. 3.4.

Figure3.4: Selezione del pattern «ID_AUTH_CHANNEL_02» per l'API

- Nel caso si voglia configurare una fruizione, le maschere di configurazione terranno conto degli aspetti di sicurezza sul canale garantendo che l'endpoint specificato nel connettore di uscita sia di tipo HTTPS, indipendentemente dal pattern adottato nella API (TLS sempre obbligatorio). L'autenticazione HTTPS può essere gestita opzionalmente da GovWay o, in alternativa, delegata alla configurazione della JVM sull'application server. Per la gestione in GovWay sono disponibili i campi per la configurazione HTTPS, con l'obbligo di impostare l'autenticazione client (vedi sez. *Autenticazione Https*).
- Nel caso si voglia configurare una erogazione, il pattern di sicurezza «ID_AUTH_CHANNEL_02» impatta sulla configurazione del Controllo Accessi, previsto nella configurazione specifica dell'erogazione:
 - La sezione «Autenticazione Canale» è impostata forzatamente a «HTTPS» (Fig. 3.5).

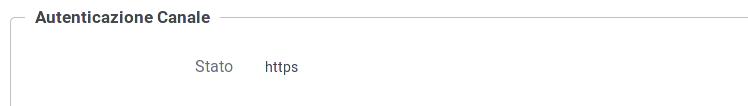


Figure3.5: Autenticazione Canale HTTPS

- Nella sezione «Autenticazione Canale» è possibile attivare l'autorizzazione per richiedente inserendo gli identificativi dei soggetti autorizzati tra quelli identificati tramite il certificato SSL (Fig. 3.6). Abilitando tale sezione sarà possibile inserire i criteri di autorizzazione, come descritto nella sez. *Autorizzazione*, con la differenza che in questo caso le politiche saranno riferite esclusivamente ai soggetti censiti in configurazione (e non gli applicativi, per i quali si rimanda alla sez. *Sicurezza Messaggio*).

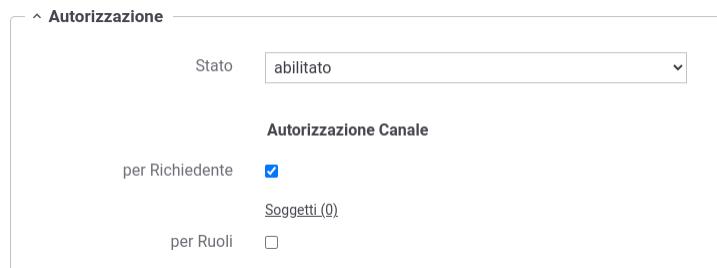


Figure3.6: Autorizzazione Canale su soggetti

3.3 Sicurezza Messaggio

Il pattern di sicurezza sul messaggio definisce le modalità di comunicazione dei messaggi tra componenti interne ai domini delle entità coinvolte. Tali pattern sono distinti per il caso SOAP e per quello REST:

- *ID_AUTH_SOAP_01 o ID_AUTH_REST_01 - Direct Trust con certificato X.509 su SOAP o REST*: tramite la validazione del certificato X509, inserito dall'applicazione mittente nel token di sicurezza, l'applicativo destinatario verifica la corrispondenza delle identità e la validità del messaggio, prima di procedere con il processamento del messaggio attraverso un trust tra fruitore e erogatore basato su certificati x509.
- *ID_AUTH_REST_01 tramite la Piattaforma Digitale Nazionale Dati (PDND)*: con l'aggiornamento delle linee guida nella “Determinazione n. 128 del 23 maggio 2023”, viene indicato di utilizzare la PDND per ottenere un token conforme al pattern ID_AUTH_REST_01; la costituzione del trust avviene attraverso il materiale crittografico depositato sulla PDND applicando i profili di emissione dei voucher previsti.
- *ID_AUTH_SOAP_02 o ID_AUTH_REST_02 - Direct Trust con certificato X.509 su SOAP o REST con unicità del messaggio/token*: estensione dei pattern precedenti con l'aggiunta di un meccanismo di filtro che impedisce il processamento di un messaggio duplicato.
- *INTEGRITY_SOAP_01 o INTEGRITY_REST_01 - Integrità del payload del messaggio SOAP o REST*: pattern che estende i precedenti aggiungendo la gestione della firma del payload come verifica di integrità del messaggio ricevuto.
- *INTEGRITY_REST_02 - Integrità del payload delle request REST in PDND*: simile al precedente pattern INTEGRITY_REST_01, assume che il trust avvenga tramite il materiale crittografico depositato sulla PDND applicando i profili di emissione dei voucher previsti. All'interno del token viene indicato l'identificativo della chiave pubblica (kid) associata alla chiave privata utilizzata dal client per firmare il token di integrità; identificativo kid generato dalla PDND e recuperabile dall'erogatore tramite le API messe a disposizione dalla PDND stessa.
- *PROFILE_NON_REPUTATION_01 - Profilo per la non ripudiabilità della trasmissione*: estende i pattern di integrità allo scopo di fornire una conferma al fruitore da parte dell'erogatore della ricezione del contenuto della richiesta. Descrive inoltre la necessità di definire un arco temporale di persistenza dei messaggi utile per

soddisfare l'opponibilità ai terzi.

- **AUDIT_REST_01 o AUDIT_REST_02 - Inoltro dati tracciati nel dominio del Fruitore:** consente all'erogatore di identificare la specifica provenienza di ogni singola richiesta di accesso ai dati effettuata dal fruitore.

Le applicazioni di un dominio interno o esterno, descritte negli scenari del Modello di Interoperabilità, vengono rappresentate in GovWay tramite la registrazione di Applicativi come entità di configurazione. In accordo al modello di GovWay, ciascun applicativo è associato al soggetto di riferimento che, nell'ottica ModI, rappresenta il dominio di appartenenza. Un applicativo viene identificato attraverso il criterio di trust del pattern di sicurezza scelto:

- trust tramite PDND: l'applicativo viene identificato tramite il “clientId” presente all'interno del token “Authorization” previsto dal pattern “ID_AUTH_REST”;
- trust tra fruitore ed erogatore tramite certificati X509: l'applicativo viene identificato tramite il certificato di firma utilizzato dal fruitore e riferito all'interno del token “Authorization” (claim x5c o x5t#256 o x5u) nel caso di pattern “ID_AUTH_REST” o all'interno dell'header SOAP di WSSecurity nel caso di pattern “ID_AUTH_SOAP”.

Vedremo nelle sezioni seguenti come si possono effettuare le configurazioni relative ai pattern di sicurezza messaggio, mentre di seguito vengono raffigurati i tipici scenari di utilizzo.

Fruizioni di API

Per quanto concerne le fruizioni, le richieste che provengono dagli applicativi interni del dominio e sono dirette verso altre amministrazioni vengono arricchite del token di sicurezza “ModI” associato all'operazione invocata. Gli applicativi vengono identificati attraverso una delle modalità di autenticazione previste da GovWay (vedi sez. *Autenticazione Trasporto*) ed una volta identificato viene utilizzato il materiale crittografico (keystore pkcs12, jks, jwk, private and public key) associatogli per effettuare la firma del token di sicurezza “ModI” (Fig. 3.8). In alternativa il materiale crittografico da utilizzare per la firma può essere definito direttamente nella fruizione o nella token policy.

Nella figura “Fig. 3.7” viene raffigurato lo scenario di fruizione in cui il trust avviene tramite la PDND.

Nella figura “Fig. 3.8” viene invece raffigurato lo scenario di fruizione in cui il trust avviene tra fruitore ed erogatore tramite certificati X509.

Erogazioni di API

In un'erogazione di una API le richieste provengono da amministrazioni esterne al dominio e sono dirette ad applicativi interni. Prima di procedere con l'inoltro della richiesta verso il backend interno, GovWay valida il token di sicurezza ricevuto rispetto al pattern associato all'operazione invocata: verifica firma, validazione temporale, filtro duplicati, verifica integrità del messaggio, verifica del token di audit etc.

Nella figura “Fig. 3.9” viene raffigurato lo scenario di erogazione in cui il trust avviene tramite la PDND:

- il token di autenticazione (ID_AUTH) viene validato rispetto alla chiave pubblica della PDND;
- i token di integrità (INTEGRITY_REST) e/o il token di audit vengono validati scaricando la chiave pubblica del firmatario del token tramite le API della PDND, utilizzando l'identificativo kid presente all'interno del token.

Nella figura “Fig. 3.10” viene invece raffigurato lo scenario di erogazione in cui il trust avviene tra fruitore ed erogatore tramite certificati X509.

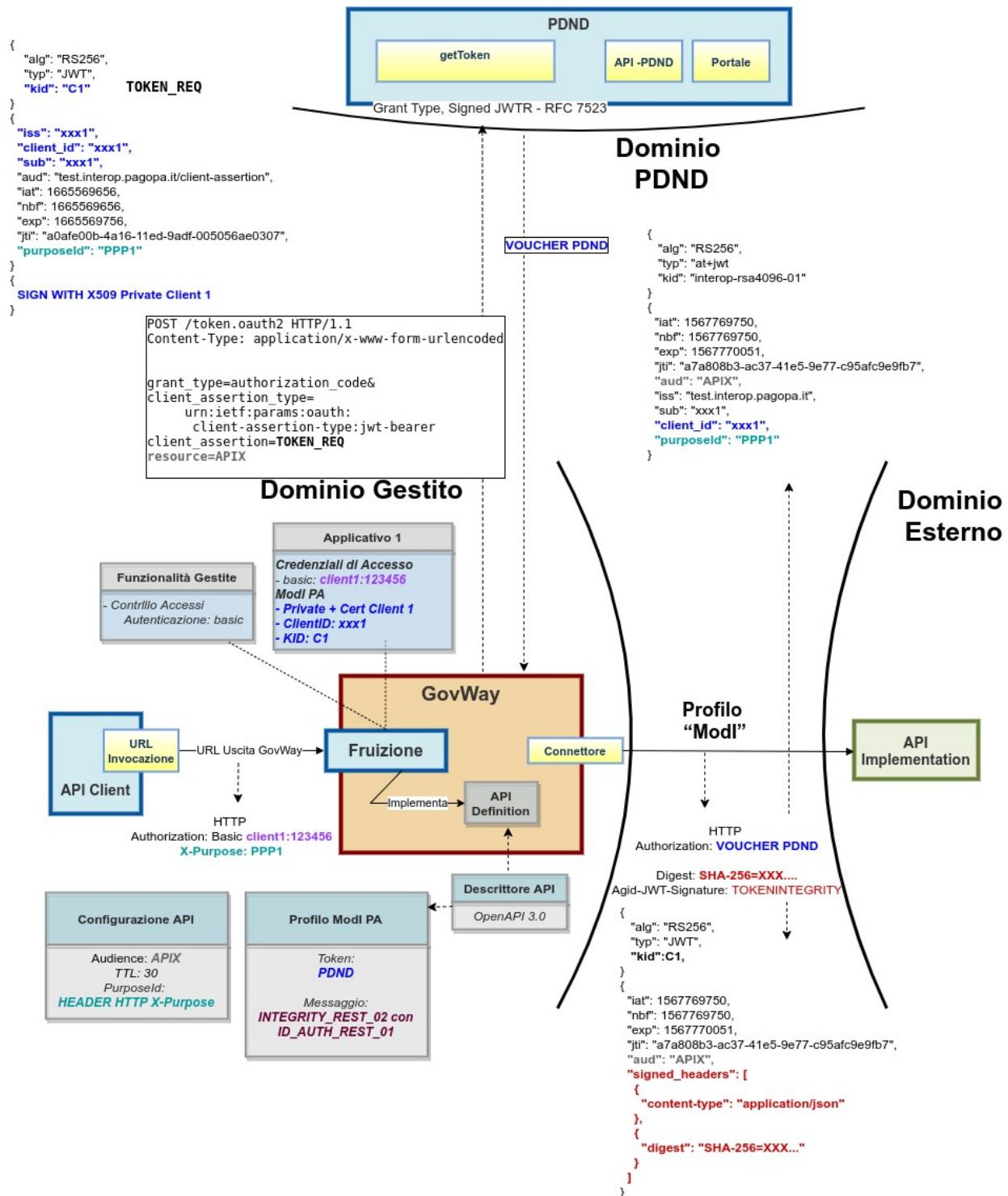


Figure3.7: Fruizione con Profilo di Interoperabilità "Modi": trust tramite PDND

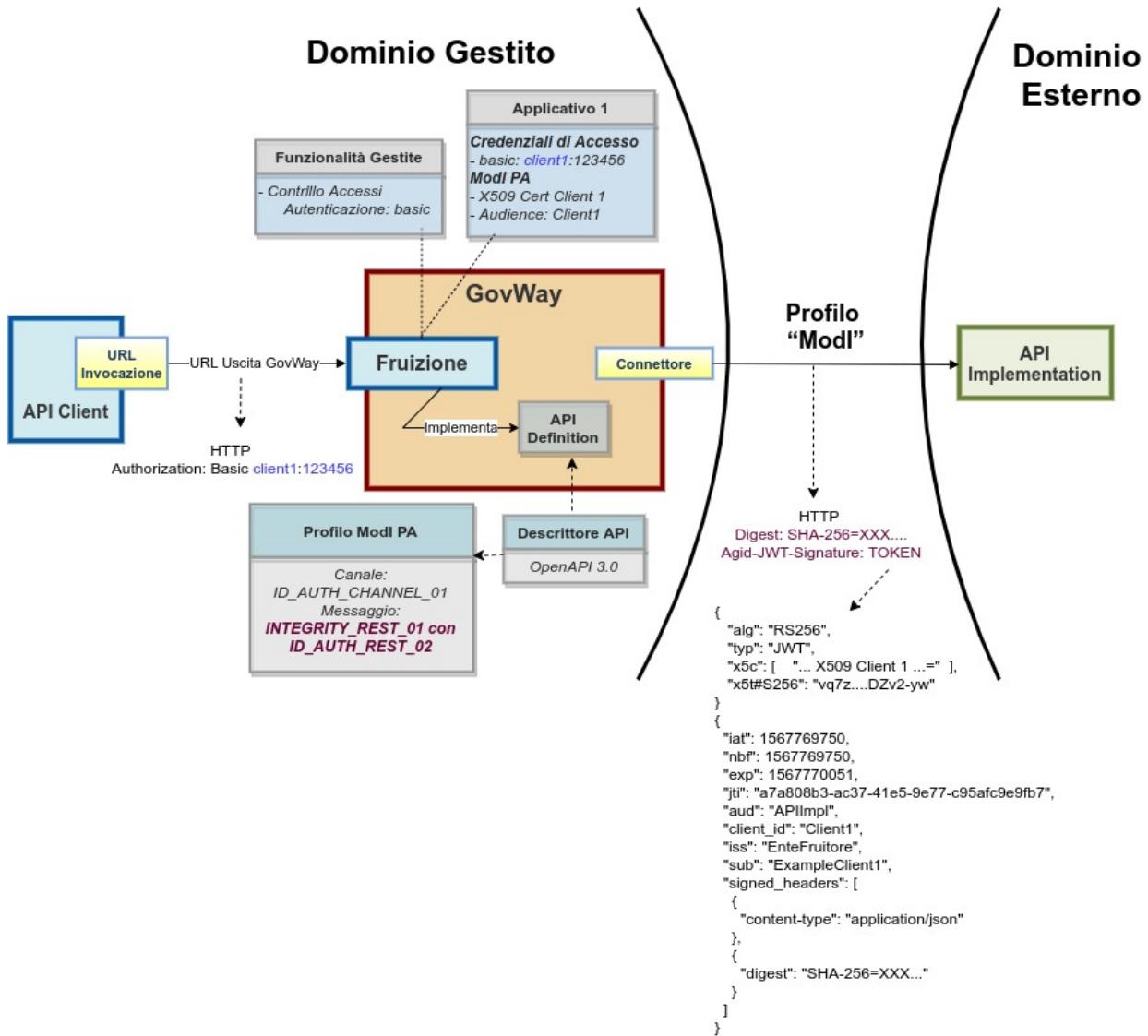


Figure3.8: Fruizione con Profilo di Interoperabilità "ModI": trust tra fruitore ed erogatore tramite certificati X509

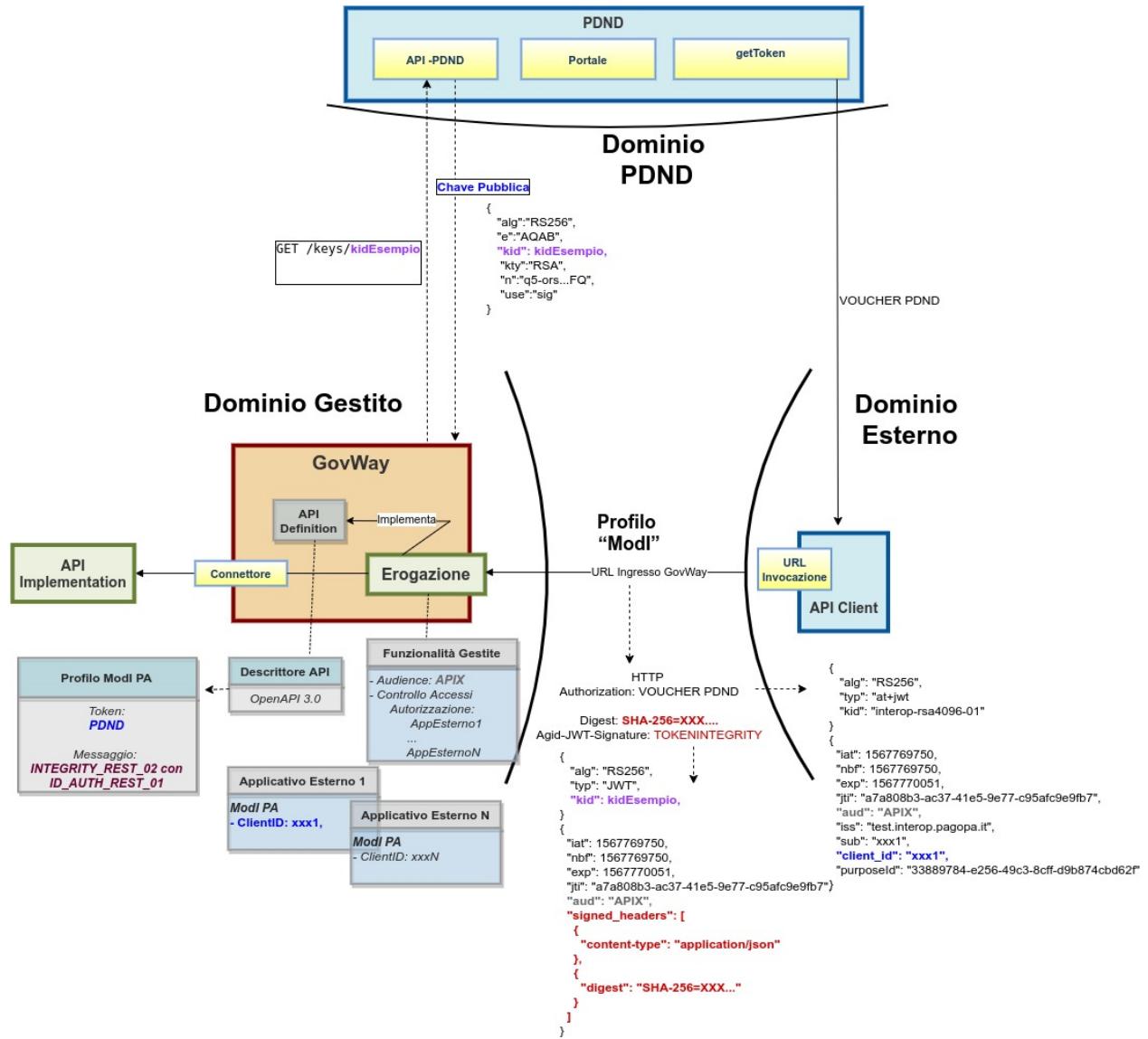


Figure3.9: Erogazione con Profilo di Interoperabilità “ModI”: trust tramite PDND

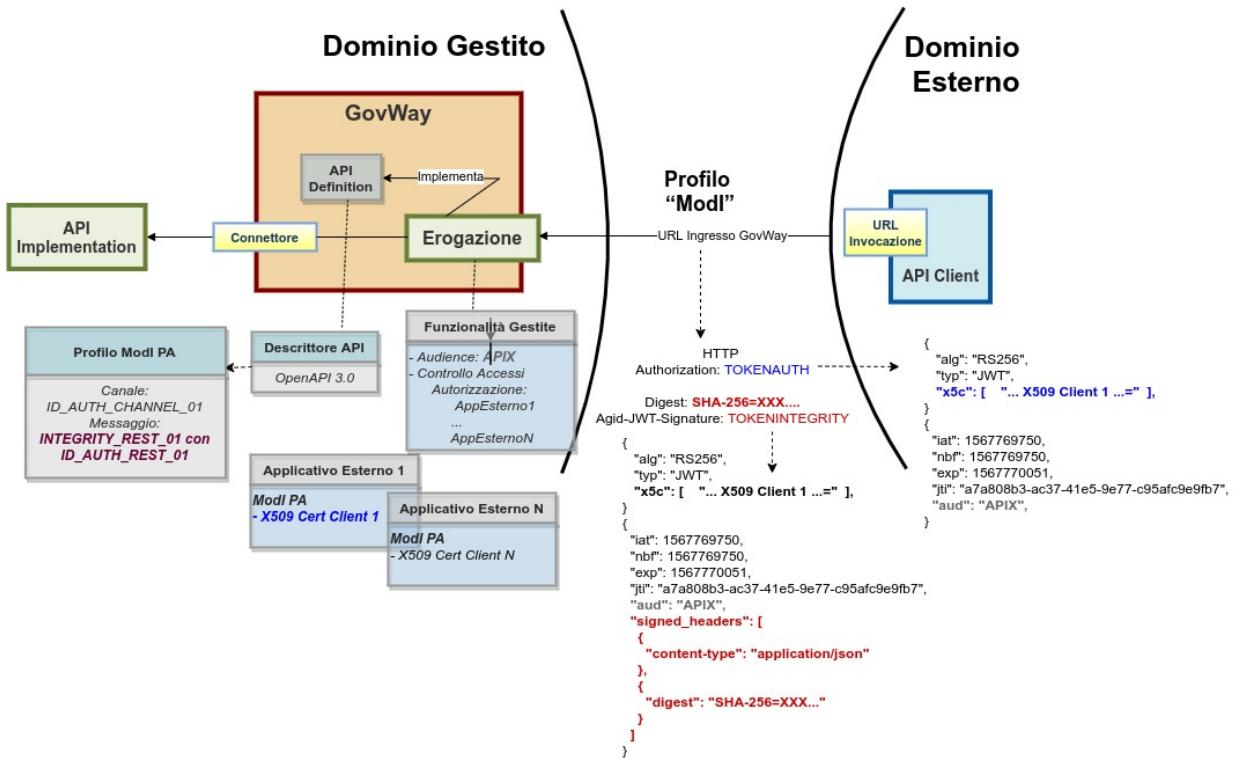


Figure3.10: Erogazione con Profilo di Interoperabilità “ModI”: trust tra fruitore ed erogatore tramite certificati X509

3.3.1 Passi preliminari di configurazione

In questa sezione vengono indicati come effettuare una configurazione iniziale degli aspetti relativi ai criteri di trust tramite PDND e tramite certificati X509, oltre alla registrazione di una coppia di chiavi utilizzata per default per produrre i token di sicurezza “ModI”.

Trust tramite PDND

Per le richieste provenienti da amministrazioni esterne e contenenti token in cui il trust avviene tramite PDND, GovWay deve validare il token “Authorization” al fine di verificare che sia stato effettivamente rilasciato dalla PDND.

Per la validazione del token GovWay utilizza una “*Token Policy Validazione*” con le seguenti caratteristiche:

- Token:
 - Tipo: JWS
 - Posizione: RFC 6750 - Bearer Token Usage (Authorization Request Header Field)
- Validazione JWT:
 - Formato Token: RFC 9068 - JSON Web Token (OAuth2 Access Token)
 - TrustStore: deve contenere la chiave pubblica utilizzata dalla PDND per firmare i token

Con il prodotto viene fornita built-in la token policy “PDND” (Fig. 3.11) da finalizzare nella sezione “TrustStore” nei seguenti aspetti:

- Location: deve essere indicata la well-known URL fornita dalla PDND, oppure in alternativa un percorso nel file system contenente ciò che è stato scaricato tramite la well-known URL:

- ambiente di collaudo: <https://uat.interop.pagopa.it/.well-known/jwks.json>
- ambiente di produzione: <https://interop.pagopa.it/.well-known/jwks.json>

Nota: Le url indicate sopra potrebbero variare; si raccomanda di ottenere sempre dalla PDND le url aggiornate.

Token Policy

Tipo	Validazione
Nome	PDND
Descrizione	<input type="text"/>

Informazioni Generali

Token

Tipo	JWS
Posizione	RFC 6750 - Bearer Token Usage

Elaborazione Token

Validazione JWT	<input checked="" type="checkbox"/>
Token Introspection	<input type="checkbox"/>
OIDC - UserInfo	<input type="checkbox"/>
Token Forward	<input type="checkbox"/>

Validazione JWT

Formato Token	RFC 9068 - JSON Web Token (OAuth2 Access Token)
TrustStore	
Tipo	JWK Set
File *	<input type="text"/> changeit
Riferimento Chiave Pubblica	<input type="text"/> Key ID 'kid' in Token
<small>Per la validazione viene utilizzata la chiave pubblica nel truststore JWKs corrispondente al 'kid' presente nel token</small>	

Figure3.11: Token Policy PDND

Trust tra fruitore ed erogatore tramite certificati X509

Per le richieste provenienti da amministrazioni esterne e contenenti token in cui il trust tra fruitore ed erogatore non avviene tramite PDND, GovWay deve validare il certificato presente all'interno del token di sicurezza al fine di verificare che sia rilasciato da una della CA conosciute, che non sia scaduto e che non sia stato eventualmente revocato.

Per effettuare la validazione del certificato di firma utilizzato dal mittente, deve essere configurato opportunamente GovWay per quanto riguarda le seguenti proprietà presenti nel file «/etc/govway/modipa_local.properties» (dove si assume che “/etc/govway” sia la directory di configurazione indicata in fase di installazione) tutte con prefisso “org.openspcoop2.protocol.modipa.sicurezzaMessaggio.certificati.”:

- *trustStore.path* (obbligatorio): indica il path su file system di un trustStore contenente le CA conosciute;
- *trustStore.tipo* (obbligatorio): indica il tipo di trustStore (JKS);
- *trustStore.password* (obbligatorio): password per accedere al trustStore;
- *trustStore.crls* (opzionale): permette di indicare un elenco, separato da virgola, di file CRL;
- *trustStore.ocspPolicy* (opzionale): in alternativa alla validazione tramite CRL è possibile associare una policy OCSP indicando uno dei tipi registrati nel file *<directory-lavoro>/ocsp.properties* come proprietà “ocsp.<idPolicy>.type”; per ulteriori dettagli si rimanda alle sezioni *ocspInstall* e *ocspConfig*.

La configurazione sopra indicata rappresenta la configurazione di default che verrà proposta durante la gestione di una erogazione o di una fruizione. È sempre possibile ridefinire per ogni singola API tale configurazione

Nota: TrustStore delle comunicazioni HTTPS

Nei pattern di sicurezza per API REST, dove il riferimento al certificato utilizzato viaggia tramite il claim “x5u”, è possibile che GovWay debba effettuare il download del certificato tramite url https che espongono certificati server non validabili tramite le CA note. In tale contesto è possibile configurare un trustStore personalizzato agendo sulle proprietà presenti nel file «/etc/govway/modipa_local.properties» in maniera simile al trustStore dei certificati. Tali proprietà possiedono il prefisso “org.openspcoop2.protocol.modipa.sicurezzaMessaggio.ssl.”.

API PDND

Introduzione

La PDND mette a disposizione delle [API](#) che consentono tra le varie funzionalità:

- *GET /keys/{kid}*: di ottenere la chiave pubblica rispetto al kid indicato nel parametro della url; questa risorsa viene utilizzata da GovWay per poter validare i token con pattern “INTEGRITY_REST_02” e/o pattern di audit “AUDIT_REST_01” o “AUDIT_REST_02” in cui il trust avviene tramite PDND, in cui l'identificativo kid è presente all'interno del token.
- *GET /events/keys*: consente di ottenere informazioni relative alle modifiche delle chiavi crittografiche registrate sulla PDND; la risorsa viene utilizzata da GovWay per mantenere aggiornata la cache locale delle chiavi scaricate dalla PDND.
- *GET /clients/{clientId}*: consente di ottenere le informazioni associate al clientId indicato nel parametro della url; la risorsa viene utilizzata da GovWay per arricchire le informazioni tracciate sul mittente.
- *GET /organizations/{organizationId}*: consente di ottenere le informazioni associate all'organizzazione indicata nel parametro della url; la risorsa viene utilizzata da GovWay per arricchire le informazioni tracciate sul mittente.

L'endpoint di esposizione delle API e la specifica OpenAPI, come indicato nella sezione [API PDND - Dove si trovano?](#), sono reperibili all'interno della sezione «Fruizione > I tuoi client api interop» e variano in funzione dell'ambiente in cui ci si trova.

Per poter fruire delle API delle PDND deve essere registrato sulla PDND un client di tipo “api interop” caricando il certificato di firma che verrà utilizzato per richiedere il token. Al termine della registrazione si otterrà un identificativo univoco della propria identità (“*client_id*” o “*sub*”) e un identificativo associato al certificato caricato (“*kid*”).

Nota: Materiale crittografico differente tra client di tipo “api interop” e “e-service”

La chiave registrata sulla PDND per quanto concerne il client di tipo “api interop” DEVE essere differente da quello che verrà utilizzato per firmare i normali token previsti dai pattern di sicurezza messaggio e audit (es. differente dalla chiave indicata nella sezione “*Chiavi di default per la firma dei token ModI*”).

Configurazione di GovWay

Per consentire a GovWay di utilizzare le risorse precedentemente descritte, viene fornita built-in la fruizione con profilo di interoperabilità “ModI” e nome “api-pdnd” (figura Fig. 3.12) da finalizzare negli aspetti descritti di seguito.

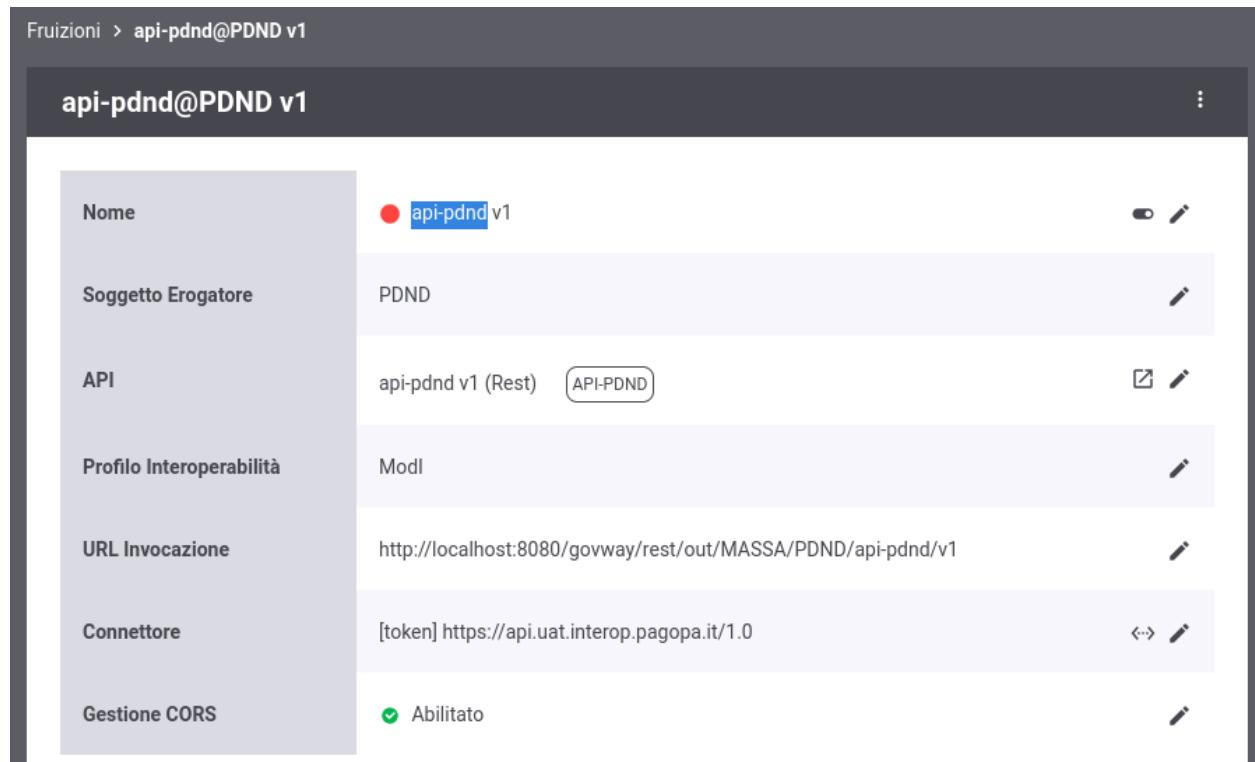


Figure3.12: Fruizione delle API PDND

- *Endpoint di esposizione delle API della PDND*: nella sezione “connettore” deve essere indicata la corretta url di esposizione delle API PDND (figura Fig. 3.13):
 - ambiente di collaudo: <https://api.uat.interop.pagopa.it/1.0>
 - ambiente di produzione: <https://api.interop.pagopa.it/1.0>

Nota: Le url indicate potrebbero variare; si raccomanda di ottenere sempre dalla PDND le url aggiornate come indicato nella sezione [API PDND - Dove si trovano?](#).

- *Token Policy di negoziazione del voucher*: nella precedente sezione “connettore” si è potuto vedere come sia stata associata al connettore una Token Policy di Negoziazione del tipo descritto nella sezione “*Signed JWT*”.

Fruizioni > api-pdnd@PDND v1 > **Connettore**

Connettore

Note: (*) Campi obbligatori

Connettore

Endpoint * ⓘ

Autenticazione Token

Autenticazione Https

Proxy

Ridefinisci Tempi Risposta

Autenticazione Token

Policy * ▾

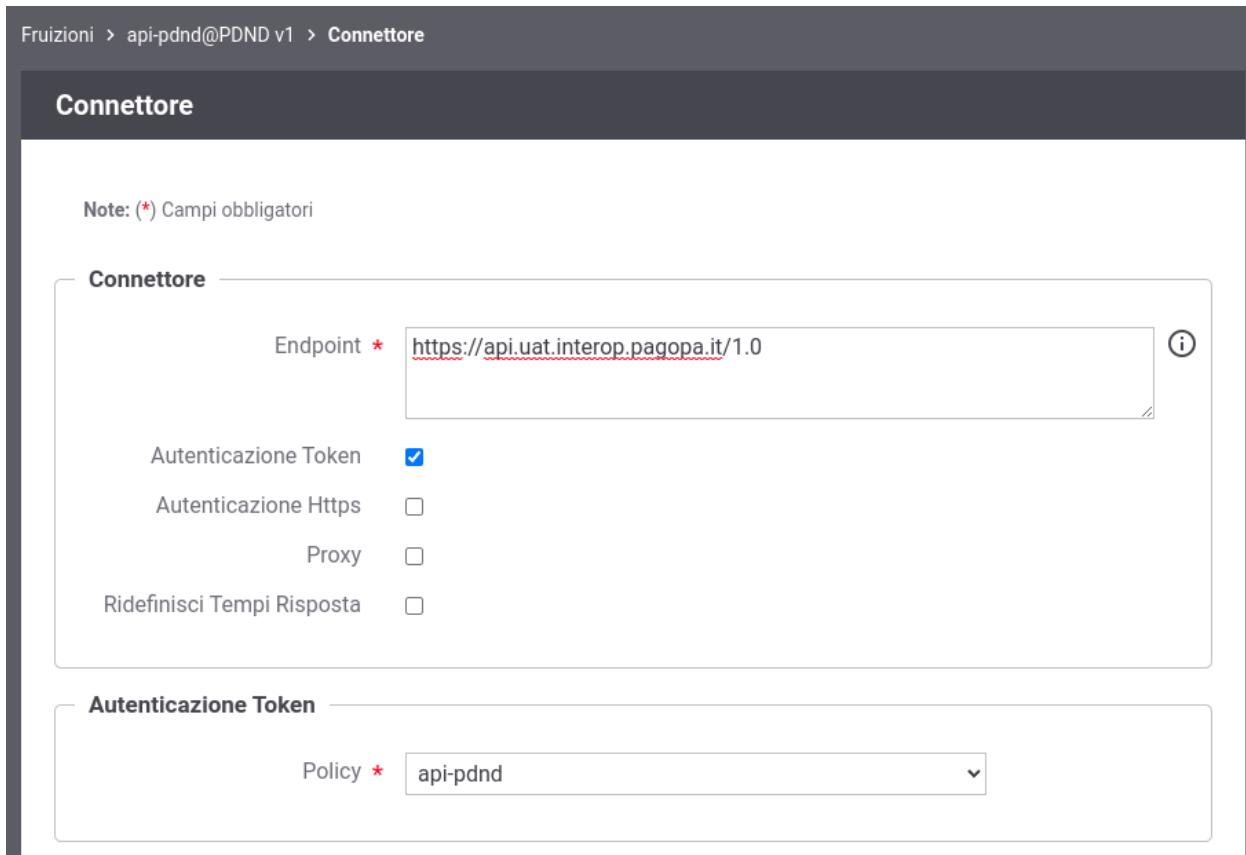


Figure3.13: Fruizione delle API PDND: connettore

La token policy “api-pdnd” riferita (figura Fig. 3.14) deve essere finalizzata nei seguenti aspetti:

- Url: deve essere indicato l’endpoint di negoziazione del voucher esposto dalla PDND:
 - * ambiente di collaudo: <https://auth.uat.interop.pagopa.it/token.oauth2>
 - * ambiente di produzione: <https://auth.interop.pagopa.it/token.oauth2>

Nota: Le url indicate potrebbero variare; si raccomanda di ottenere sempre dalla PDND le url aggiornate come indicato nella sezione [Richiesta di un voucher spendibile presso le API di Interoperabilità](#) dove viene indicato che l’URL dell’endpoint cambia in funzione dell’ambiente e sarà chiaramente visibile sull’interfaccia all’interno del back office.

- Audience: deve essere indicato il corretto valore atteso dal servizio della PDND, valore che cambia in funzione dell’ambiente:
 - * ambiente di collaudo: auth.uat.interop.pagopa.it/client-assertion
 - * ambiente di produzione: auth.interop.pagopa.it/client-assertion

Nota: I valori indicati potrebbero variare; si raccomanda di ottenere sempre dalla PDND i valori aggiornati.

- *Materiale crittografico e dati della PDND*: nella sezione “ModI” devono essere configurati tutti i parametri relativi al materiale crittografico e ai dati identificativi ottenuti dalla PDND in seguito alla registrazione del client di tipo “api interop” (figura Fig. 3.15):
 - Key Id (kid) del Certificato: identificativo kid della chiave pubblica;
 - Identificativo: clientId associato alla chiave pubblica;
 - Chiave Privata e Chiave Pubblica: indica il path su file system rispettivamente delle chiavi private e pubbliche in formato PEM o DER (sono supportati sia i formati pkcs1 che pkcs8);
 - Password Chiave Privata: se la chiave privata è cifrata deve essere indicata la password.

Nota: Tramite il campo “Tipo” è possibile utilizzare un tipo di archivio differente dalla coppia di chiavi pubblica e privata come un keystore “PKCS12”, “JKS” o un archivio json “JWK”.

- *Controllo degli Accessi*: si può notare come la fruizione riporta uno «stato rosso» che evidenzia una configurazione incompleta nella parte relativa al *Controllo degli Accessi*. Procedere con la configurazione del *Controllo degli Accessi* al fine di renderla invocabile secondo la modalità di autenticazione ed autorizzazione desiderata. Le modalità scelte dovranno poi comportare una configurazione adeguata, descritta nel punto successivo, in modo da consentire a GovWay di invocare la fruizione.
- *Fruizione dell’API PDND da parte di GovWay*: la modalità di invocazione della fruizione viene definita tramite le proprietà presenti nel file «/etc/govway/modipa_local.properties» tutte con prefisso “org.openspcoop2.protocol.modipa.sicurezzaMessaggio.certificati.remoteStore.pdnd.”:
 - *baseUrl* (obbligatorio): definisce la url invocata per ottenere la chiave pubblica rispetto ad un identificativo kid;
 - *connectTimeout* e *readTimeout* (obbligatorio): consentono di impostare rispettivamente i limiti temporali per l’instaurazione di una connessione e la ricezione di una risposta dalla PDND;

Token Policy

Tipo	Negoziazione
Nome	api-pdnd
Descrizione	<input style="width: 100%;" type="text"/>

Token Endpoint

Tipo	<input type="text" value="Signed JWT"/>
PDND	<input type="checkbox"/>
URL *	<input type="text" value="https://auth.uat.interop.pagopa.it/token.oauth2"/> (i)
Connection Timeout *	<input type="text" value="5000"/>
Read Timeout *	<input type="text" value="10000"/>
Https	<input type="checkbox"/>
Proxy	<input type="checkbox"/>

JWT KeyStore

Tipo	<input type="text" value="Definito nella fruizione ModI"/>
------	--

JWT Signature

Signature Algorithm	<input type="text" value="RS256"/>
---------------------	------------------------------------

JWT Header

Key Id (kid)	<input type="text" value="Definito nella fruizione ModI"/>
X.509 Certificate	<input type="text" value="-"/>
Digest X.509 Certificate	<input type="text" value="-"/>
Type (typ) *	<input type="text" value="JWT"/>
Content Type (cty)	<input type="checkbox"/>

JWT Payload

Client ID	<input type="text" value="Definito nella fruizione ModI"/>
Issuer	<input type="text" value="ClientID della fruizione ModI"/>
Subject	<input type="text" value="ClientID della fruizione ModI"/>
Audience *	<input type="text" value="auth.uat.interop.pagopa.it/client-assertion"/> (i)
Identifier	<input type="text" value="\${transaction:id}"/> (i)
Time to Live (secondi) *	<input type="text" value="300"/>

Indica la validità temporale, in secondi, a partire dalla data di creazione dell'asserzione

Figure3.14: Fruizione delle API PDND: token policy

Fruizioni > api-pdnd@PDND v1 > Profilo Interoperabilità

Profilo Interoperabilità

Note: (*) Campi obbligatori

ModI - Authorization OAuth

Key Id (kid) del Certificato	<input type="text" value="KID_FORNITO_PDND"/> (i)
Identificativo	<input type="text" value="CLIENT_ID_FORNITO_PDND"/> (i)
KeyStore	<input type="text" value="Ridefinito"/> ▼

KeyStore

Modalità	<input type="text" value="File System"/> ▼
Tipo	<input type="text" value="Key Pair"/> ▼
Chiave Privata *	<input type="text" value="PATH_PRIVATE_KEY"/>
Chiave Pubblica *	<input type="text" value="PATH_PUBLIC_KEY"/>
Password Chiave Privata	<input type="text" value="OPTIONAL_PASSWORD_PRIVATE_KEY"/>

Figure3.15: Fruizione delle API PDND: profilo “ModI”

- *http.username* e *http.password* (opzionale): se definite GovWay invocherà la fruizione utilizzando le credenziali http basic indicate; la keyword speciale “#none#” è utilizzabile per ridefinire la configurazione allo scopo di disabilitare l’invio delle credenziali.
 - *http.header.<nome>* (opzionale): consente di inviare http header personalizzati;
 - *http.queryParameter.<nome>* (opzionale): consente di aggiungere parametri personalizzati alla url invocata;
 - *https.keyStore*, *keyStore.type*, *keyStore.password*, *key.alias*, *key.password* (opzionale): le seguenti proprietà consentono di specificare un certificato tls client con cui GovWay invocherà la fruizione delle API PDND.
 - *https.hostnameVerifier* (opzionale): nel caso in cui la baseUrl indicata sia https consente di attivare o meno la verifica dell’hostname rispetto al CN.
 - *https.trustAllCerts* (opzionale): nel caso in cui la baseUrl indicata sia https disabilita l’autenticazione del certificato server.
 - *https.trustStore*, *https.trustStore.type*, *https.trustStore.password*, *https.trustStore.crl* (opzionale): consente di effettuare una autenticazione del certificato server rispetto ai parametri di truststore indicati.
- *Pull sulla PDND per ottenere gli eventi relativi alle chiavi*: come indicato nella sezione [Endpoint di notifica eventi](#), le API della PDND consentono all’aderente di ottenere una lista di eventi che possono essere utilizzate da GovWay per mantenere aggiornata la cache locale delle chiavi scaricate dalla PDND. Per default la consultazione degli eventi è disabilitata e per abilitarla si deve intervenire sulle proprietà presenti nel file «/etc/govway/govway_local.properties» tutte con prefisso “org.openscoop2.pdd.gestoreChiaviPDND.”:
 - *enabled*: impostare a true la proprietà per abilitare la consultazione degli eventi.
 - *keys.maxLifeMinutes*: indica la vita in minuti di una chiave scaricata dalla PDND e salvata nella cache locale (default: 43200, 30 giorni).
 - *events.keys.limit* indica il numero massimo di eventi recuperati tramite una singola chiamata alla PDND (default: 100).
 - *events.keys.timer.intervalloSecondi*: definisce l’intervallo, in secondi, rispetto al quale vengono controllati eventuali nuovi eventi sulla PDND (default: 3600, un’ora).
 - *cache.keys.timer.intervalloSecondi*: govway dispone di più livelli di cache (che si differenziano se risiedono in RAM o su Database). Questa proprietà definisce l’intervallo, in secondi, rispetto al quale le chiavi presenti nella cache in RAM vengono verificate rispetto alla chiavi presenti nella cache su Database (default: 300, 5 minuti).
 - *Erogazione: maggiori informazioni sul mittente*: le API della PDND consentono anche di ottenere informazioni sull’organizzazione a cui il client afferisce. Tali informazioni possono essere recuperate da GovWay al fine di arricchire le tracce e definire criteri autorizzativi; una volta scaricate vengono mantenute in una cache locale. Per default la consultazione della PDND per ottenere maggiori informazioni sui client è disabilitata e per abilitarla si deve intervenire sulle proprietà presenti nel file «/etc/govway/govway_local.properties» tutte con prefisso “org.openscoop2.pdd.gestorePDND.”:
 - *clientInfo.enabled*: impostare a true la proprietà per abilitare la raccolta delle informazioni sul client;
 - *clientInfo.maxLifeMinutes*: indica la vita in minuti delle informazioni scaricate dalla PDND e salvate nella cache locale (default: 43200, 30 giorni);
 - *clients.error.abortTransaction* indicazione se far fallire la transazione in caso il recupero delle informazioni sul client fallisca (default: false, vedi [Recupero informazioni client tramite API PDND fallito](#));
 - *organizations.error.abortTransaction* indicazione se far fallire la transazione in caso il recupero delle informazioni sull’organizzazione fallisca (default: false, vedi [Recupero informazioni client tramite API PDND fallito](#)).

Nota: La raccolta delle informazioni sul mittente tramite la PDND richiede che la consultazione degli eventi, descritta nel precedente punto, sia stata abilitata nel file «/etc/govway/govway_local.properties» tramite la proprietà “org.openspcoop2.pdd.gestoreChiaviPDND.enabled”

Chiavi di default per la firma dei token ModI

Nelle figure [Fig. 3.8](#) e [Fig. 3.7](#) della sezione [Sicurezza Messaggio](#) è stato descritto come GovWay utilizzerà la chiave privata associata all'applicativo interno che ha scaturito la richiesta per firmare il token di sicurezza aggiunto al messaggio in uscita dal dominio di gestione o per l'access token di richiesta del voucher verso la PDND. È possibile attuare uno scenario differente in cui la chiave privata viene definita nella fruizione come descritto nella sezione [Keystore di firma definito nella fruizione](#); in questo caso è possibile configurare l'utilizzo di un keystore di default indicato nella configurazione descritta di seguito.

Anche per quanto concerne le risposte che GovWay processa in una erogazione, la chiave privata utilizzata per firmare il token di sicurezza aggiunto alla risposta viene preso dalla configurazione di default descritta di seguito.

È sempre possibile ridefinire per ogni singola API il keystore utilizzato in un richiesta di una fruizione o in una risposta di una erogazione.

La coppia di chiavi di default utilizzate per firmare i token ModI possono essere configurate su GovWay tramite le proprietà presenti nel file «/etc/govway/modipa_local.properties» tutte con prefisso “org.openspcoop2.protocol.modipa.sicurezzaMessaggio.certificati.”:

- *keyStore.path* (obbligatorio): indica il path su file system di un keyStore contenente la chiave privata;
- *keyStore.tipo* (obbligatorio): indica il tipo di trustStore (JKS);
- *keyStore.password* (obbligatorio): password per accedere al keyStore;
- *key.alias* (obbligatorio): alias della chiave privata all'interno del keyStore;
- *key.password* (obbligatorio): password della chiave privata all'interno del keyStore;

Inoltre se la chiave pubblica presente nel keystore, definito nella proprietà “keyStore.path”, viene utilizzata come materiale crittografico per registrare un client sulla PDND sono utilizzabili le seguenti due ulteriori proprietà:

- *key.clientId* (opzionale): clientId associato dalla PDND alla chiave pubblica;
- *key.kid* (opzionale): identificativo kid con cui la PDND ha registrato la chiave pubblica.

3.3.2 ID_AUTH_SOAP_01 / ID_AUTH_REST_01

Il pattern ID_AUTH nelle sue varie declinazioni ha lo scopo di identificare e autorizzare l'accesso del fruitore ad un servizio.

Il pattern differisce rispetto al tipo di trust:

- trust tramite PDND: descritto nella sezione “[ID_AUTH_REST_01 tramite la Piattaforma Digitale Nazionale Dati \(PDND\)](#)”, e applicabile sia per API REST che SOAP, consiste nella negoziazione con la PDND di un voucher spendibile verso l'erogatore.
- trust tra fruitore ed erogatore tramite certificati X509: descritto nella sezione “[ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509](#)” differisce nella sostanza se applicato per API REST, in cui viene prodotto un token JWT, o per API SOAP, in cui viene definito un header SOAP WSSecurity. In entrambi i casi il certificato del mittente viene inserito all'interno del token di sicurezza e validato dall'erogatore tramite un trustStore contenente i certificati X509 attesi.

- trust tramite Authorization Server: descritto nella sezione “[ID_AUTH_REST_01 tramite un Authorization Server OAuth differente dalla PDND](#)”, e applicabile sia per API REST che SOAP, consiste nella negoziazione con un authorization server differente dalla PDND di un voucher spendibile verso l’erogatore.

ID_AUTH_REST_01 tramite la Piattaforma Digitale Nazionale Dati (PDND)

Il token di autenticazione ID_AUTH_REST_01 descritto nella sezione “[ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509](#)” viene generato e firmato dall’applicativo mittente. Un token con analoga struttura può essere richiesto sulla Piattaforma Digitale Nazionale Dati (PDND) al fine di ottenere un voucher di autenticazione spendibile verso l’erogatore.

Il token può essere richiesto per tutti i servizi registrati sulla PDND per cui un erogatore ha completato il processo di on-boarding definendo:

- gli attributi che un fruitore deve possedere per poter fare richiesta di fruizione;
- l’identificativo del servizio che il fruitore dovrà riferire per ottenere un token (“*audience*”);
- altri parametri quali la durata del token e il carico di chiamate giornaliere supportate in termini di chiamate complessive e per fruitore.

Per ottenere un token, un applicativo mittente deve:

- essersi registrato sulla PDND caricando il certificato di firma che utilizzerà per richiedere il token. Al termine della registrazione otterrà un identificativo univoco della propria identità (“*client_id*” o “*sub*”) e un identificativo associato al certificato caricato (“*kid*”).
- aver registrato una finalità che descrive la motivazione per cui vuole richiedere la fruizione del servizio e il numero di richieste giornaliere che intende effettuare. La creazione di una finalità si completa con l’ottenimento di un suo identificativo univoco denominato “*purposeId*”.

L’adozione del pattern ID_AUTH_REST_01 rilasciato dalla PDND consente alla ricezione di un messaggio di effettuare i medesimi controlli attuati su un token conforme al pattern descritto nella sezione “[ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509](#)” con la differenza che il token non è più firmato dall’applicativo mittente ma bensì dall’authorization server della PDND e l’identificazione dell’applicativo chiamante non è più attuabile tramite il certificato fornito nell’header del JWT tramite claim “x5c/x5t/x5u” ma bensì tramite l’identificativo presente nel claim “*client_id*”.

Di seguito vengono forniti i dettagli di configurazione necessari ad utilizzare la PDND negli scenari di fruizione o erogazione di un servizio.

Fruizione ID_AUTH_REST_01 (PDND)

Le richieste che provengono dagli applicativi interni del dominio e sono dirette verso altre amministrazioni verranno arricchite del token di sicurezza “ModI” previsto dall’operazione invocata, come indicato precedentemente nella sezione [ID_AUTH_REST_01 tramite la Piattaforma Digitale Nazionale Dati \(PDND\)](#).

Nella figura “Fig. 3.16” viene raffigurato lo scenario di fruizione in cui il trust avviene tramite la PDND.

Di seguito vengono descritti tutti i passi di configurazione specifici per l’implementazione del pattern “ID_AUTH_REST_01” mentre si rimanda alla sezione “[Profilo “API Gateway”](#)” per la normale registrazione e configurazione di una fruizione di API.

API

La registrazione della API deve essere effettuata agendo nella sezione «ModI - Sicurezza Messaggio», come indicato in [Fig. 3.17](#):

- selezionare il “Pattern” «ID_AUTH_REST_01»;

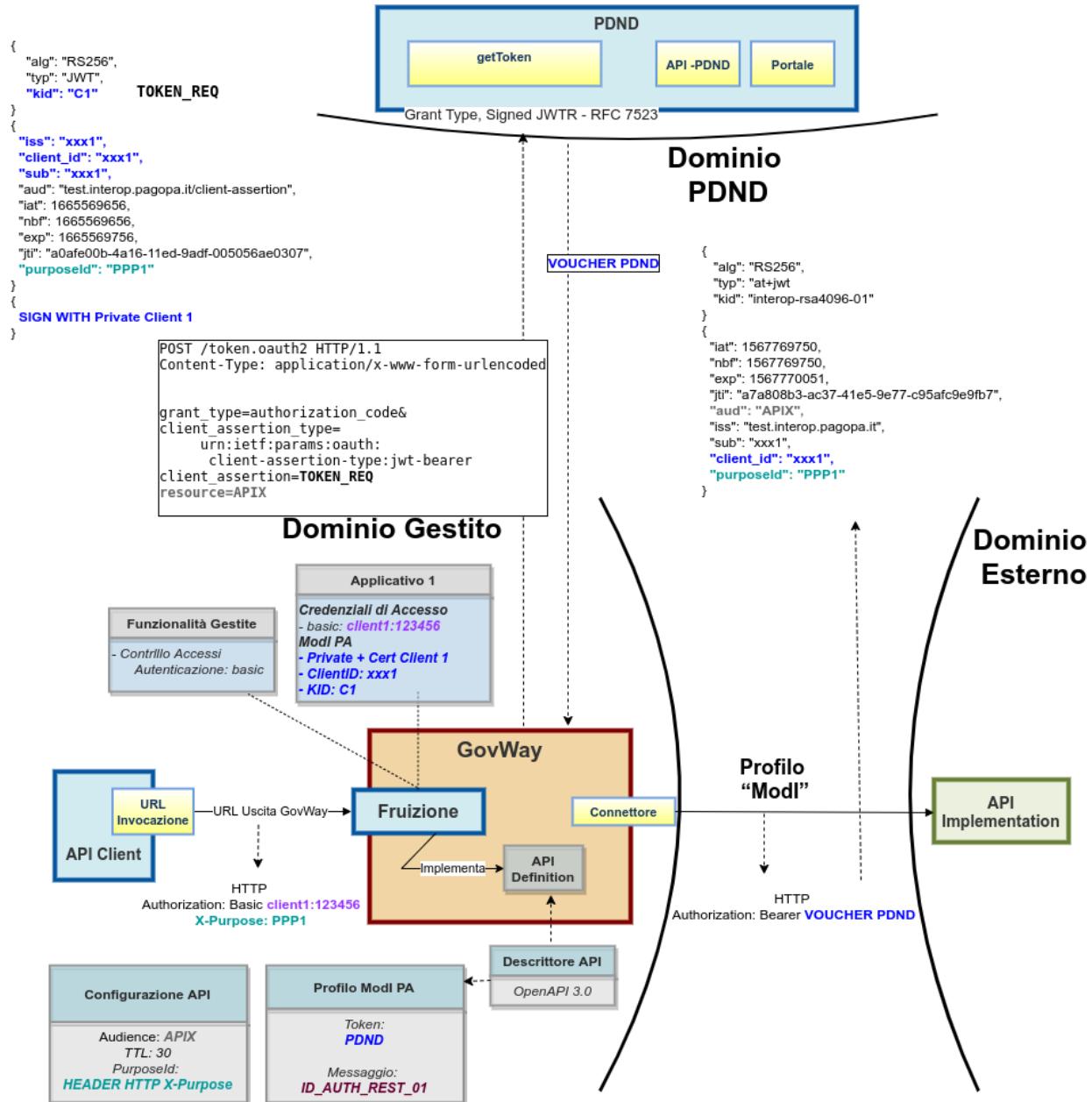


Figure3.16: Fruizione con Profilo di Interoperabilità ‘‘Modi’’, pattern ‘‘ID_AUTH_REST_01’’: trust tramite PDND

- selezionare una “Generazione Token” di tipo “Authorization PDND” per far sì che il Token “ID_AUTH” sia negoziato con la PDND.

Modi

Sicurezza Canale

Pattern: ID_AUTH_CHANNEL_01
Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern: ID_AUTH_REST_01
Direct Trust con certificato X.509

Generazione Token: Authorization PDND
Token ID_AUTH negoziato con la PDND

Informazioni Audit: Dati del dominio del fruitore

Figure3.17: Selezione del pattern «ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization PDND” per l’API

Token Policy di Negoziazione

Per la configurazione delle fruizioni con un pattern di sicurezza via PDND è necessario registrare una Token Policy di Negoziazione del tipo descritto nella sezione “[Signed JWT \(PDND\)](#)”.

Di seguito vengono riportati tutte le informazioni da registrare nella policy:

- Tipo: SignedJWT;
- PDND: flag attivato;
- URL: endpoint esposto dalla PDND su cui è possibile richiedere lo stacco del voucher alla PDND (Fig. 3.18):
 - ambiente di collaudo: <https://auth.uat.interop.pagopa.it/token.oauth2>
 - ambiente di produzione: <https://auth.interop.pagopa.it/token.oauth2>

Nota: Le url indicate potrebbero variare; si raccomanda di ottenere sempre dalla PDND le url aggiornate come indicato nella sezione [Richiesta di un voucher spendibile presso un e-service del catalogo](#).

- JWT Keystore: parametri di accesso al keystore contenente la chiave privata corrispondente alla chiave pubblica registrata sulla PDND durante la registrazione dell’applicativo client. I parametri variano in funzione del tipo di keystore selezionato:
 - “JKS”, “PKCS12”: deve essere definito il path su filesystem dove risiede il keystore, la password per l’accesso al keystore, l’alias con cui è riferita la chiave privata e la password (Fig. 3.19);
 - “JWK Set”: deve essere definito il path su filesystem dove risiede l’archivio json nel formato “JWK Set” e l’identificativo “kid” (alias) con cui è riferita la chiave privata (Fig. 3.20);
 - “Key Pair”: deve essere definito il path su filesystem dove risiedono la chiave privata e pubblica in formato PEM o DER (sono supportati sia i formati pkcs1 che pkcs8) e la password della chiave privata se cifrata (Fig. 3.21);
 - “Definito nell’applicativo ModI”: il keystore utilizzato per firmare l’asserzione JWT inviata alla PDND sarà quello definito nell’applicativo ModI richiedente (scenario descritto nel seguito di questa sezione);

Token Policy > Aggiungi

Note: (*) Campi obbligatori

Token Policy

Nome: * PDND-NegoziatoreToken

Token Endpoint

URL: * http://...../token

Connection Timeout: * 5000

Read Timeout: * 10000

Https:

Proxy:

Figure3.18: Token Policy di Negoziazione PDND (Endpoint)

JWT KeyStore

Nome: * PDND-NegoziatoreToken

File: *

Alias Chiave Privata: *

File: *

Alias Chiave Privata: *

Figure3.19: Token Policy di Negoziazione PDND (Keystore “PKCS12”)

JWT KeyStore

Nome: * PDND-NegoziatoreToken

File: *

Alias Chiave Privata: *

Figure3.20: Token Policy di Negoziazione PDND (Keystore “JWK Set”)

JWT KeyStore

Chiave Privata *

Chiave Pubblica *

>Password Chiave Privata

Figure3.21: Token Policy di Negoziazione PDND (Keystore “Key Pair”)

JWT KeyStore

Tipo

Definito nell'applicativo ModI

Figure3.22: Token Policy di Negoziazione PDND (Keystore definito nell'applicativo ModI)

- “Definito nella fruizione ModI”: il keystore utilizzato per firmare l’asserzione JWT inviata alla PDND sarà quello definito nella fruizione ModI (scenario descritto nel seguito di questa sezione);

JWT KeyStore

Tipo

Definito nella fruizione ModI

Figure3.23: Token Policy di Negoziazione PDND (Keystore definito nella fruizione ModI)

- Tipi PKCS11: gli altri tipi disponibili sono quelli corrispondenti ai tipi di keystore PKCS11 registrati (“*Device PKCS11*”).
- JWT Signature: algoritmo di firma
- JWT Header:
 - Type (typ): lasciare il valore “JWT”;
 - Key Id (kid): deve essere indicato l’identificativo univoco (KID) ottenuto al termine della registrazione dell’applicativo client sulla PDND. Può essere fornito tramite una delle seguenti modalità:
 - * “Personalizzato”: selezionando la modalità “Personalizzato” è possibile indicarlo puntualmente. Il valore può essere definito come costante o contenere parti dinamiche risolte a runtime dal Gateway (“*Valori dinamici*”);
 - * “Definito nell’applicativo ModI”: nel caso in cui è stato indicato un keystore definito nell’applicativo ModI, è possibile selezionare una modalità analoga anche per il KID (Fig. 3.26).
- Questa modalità richiede che oltre al keystore, nell’applicativo ModI richiedente venga abilitata anche la sezione “Authorization OAuth” e venga indicato il KID nel campo “Key Id del Certificato” (Fig. 3.38).
- “Definito nella fruizione ModI”: nel caso in cui è stato indicato un keystore definito nella fruizione ModI, è possibile selezionare una modalità analoga anche per il KID (Fig. 3.27).
- Questa modalità richiede che oltre al keystore, nella fruizione ModI venga abilitata anche la sezione “Authorization PDND” e venga indicato il KID nel campo “Key Id del Certificato” (Fig. 3.36).
- JWT Payload:

JWT Signature

Signature Algorithm



Figure3.24: Token Policy di Negoziazione PDND (Algoritmo di Firma)

JWT Header

Key Id (kid)

X.509 Certificate

Digest X.509 Certificate

Type (typ) *

Content Type (cty)



Figure3.25: Token Policy di Negoziazione PDND (KID personalizzato)

JWT Header

Key Id (kid)

X.509 Certificate

Digest X.509 Certificate

Type (typ) *

Content Type (cty)



Figure3.26: Token Policy di Negoziazione PDND (KID definito nell'applicativo ModI)

JWT Header

Key Id (kid)

X.509 Certificate

Digest X.509 Certificate

Type (typ) *

Content Type (cty)



Figure3.27: Token Policy di Negoziazione PDND (KID definito nella fruizione ModI)

- Client ID, Issuer e Subject: l’identificativo univoco dell’applicativo client (“*client_id*” o “*sub*”) ottenuto al termine della registrazione dell’applicativo sulla PDND deve essere configurato nei tre campi indicati tramite una delle seguenti modalità:

- * indicati nella token policy:

JWT Payload

Client ID *	<input type="text"/>	i
Issuer	<input type="text"/>	i
Subject	<input type="text"/>	i

Figure3.28: Token Policy di Negoziazione PDND (ClientId)

- * in alternativa nel caso in cui sia stato indicato un keystore definito nell’applicativo ModI, è possibile selezionare una modalità analoga anche per la tripla clientId/issuer/subject (Fig. 3.29).

JWT Payload

Client ID	<input type="text" value="Definito nell’applicativo ModI"/>	▼
Issuer	<input type="text" value="ClientID dell’applicativo ModI"/>	▼
Subject	<input type="text" value="ClientID dell’applicativo ModI"/>	▼

Figure3.29: Token Policy di Negoziazione PDND (ClientId definito nell’applicativo ModI)

Questa modalità richiede che oltre al keystore, nell’applicativo ModI richiedente venga abilitata anche la sezione “Authorization OAuth” e venga indicato il clientId nel campo “Identificativo” (Fig. 3.38).

- * infine nel caso in cui sia stato indicato un keystore definito nella fruizione ModI, è possibile selezionare una modalità analoga anche per la tripla clientId/issuer/subject (Fig. 3.30).

JWT Payload

Client ID	<input type="text" value="Definito nella fruizione ModI"/>	▼
Issuer	<input type="text" value="ClientID della fruizione ModI"/>	▼
Subject	<input type="text" value="ClientID della fruizione ModI"/>	▼

Figure3.30: Token Policy di Negoziazione PDND (ClientId definito nella fruizione ModI)

Questa modalità richiede che oltre al keystore, nella fruizione ModI venga abilitata anche la sezione “Authorization PDND” e venga indicato il clientId nel campo “Identificativo” (Fig. 3.27).

- Gli altri campi presenti nella sezione “JWT Payload” rappresentano (Fig. 3.31):

- * Audience: indica il servizio di stacco del voucher della PDND. Il valore, fornito dalla PDND, è indipendente dal servizio per cui si vuole richiedere un voucher e varia solamente in funzione dell’ambiente di validazione o produzione della PDND stessa:

- ambiente di collaudo: auth.uat.interop.pagopa.it/client-assertion
- ambiente di produzione: auth.interop.pagopa.it/client-assertion

Nota: I valori indicati potrebbero variare; si raccomanda di ottenere sempre dalla PDND i valori aggiornati.

- * Identifier: consente di configurare la modalità di valorizzazione del claim “`“jti”` presente all’interno del token di richiesta inviato alla PDND. Si suggerisce di valorizzare il campo con la keyword `“${transaction:id}”` al fine di utilizzare l’identificativo di transazione della richiesta;
- * Time to Live (secondi): consente di indicare la durata del token di richiesta inviato alla PDND (es. 100 sec);
- * Purpose ID: identificativo univoco della finalità, ottenuto dalla PDND, per cui si intende fruire di un servizio. La modalità di configurazione varia a seconda dello scenario che si desidera supportare come descritto nella sezione “*Finalità (purposeId) utilizzata per una fruizione di API*”.
- * Informazioni Sessione: consente di valorizzare il claim “`“sessionInfo”`” previsto dalla PDND. La valorizzazione può essere statica o formata da parti dinamiche risolte a runtime dal Gateway (per maggiori dettagli *Valori dinamici*).

The screenshot shows the configuration interface for a Token Policy. It includes the following fields:

- Audience ***: A text input field with a placeholder for the audience URL.
- Identifier**: A text input field containing the value `“${transaction:id}”`.
- Time to Live (secondi) ***: A text input field with the value `300`.
- Purpose ID ***: A text input field.
- Informazioni Sessione**: A text input field for session information.
- Claims**: A text input field for claims.

Below the fields are descriptive labels:

- “Indica la validità temporale, in secondi, a partire dalla data di creazione dell’asserzione”
- “Indicare per riga i claims (nome=valore) da aggiungere nell’oggetto ‘sessionInfo’”
- “Indicare per riga gli ulteriori claims (nome=valore)”

Figure3.31: Token Policy di Negoziazione PDND (JWT Payload)

- Dati Richiesta:
 - Resource: indicare l’audience/url del servizio per cui si vuole richiedere un voucher;
 - Client ID: indicare il medesimo valore inserito nel campo “Client ID” della sezione “JWT Payload”;

The screenshot shows the configuration interface for the “Dati Richiesta” section. It includes the following fields:

- Scope**: A text input field with a placeholder for scopes separated by commas.
- Audience**: A text input field.
- Client ID**: A text input field with the value `clientId`.
- Resource**: A text input field with the value `resource`.

Figure3.32: Token Policy di Negoziazione PDND (DatiRichiesta)

Per quanto concerne il campo “Client ID”, nel caso in cui sia stato indicato un keystore definito nell’applicativo ModI, è possibile selezionare una modalità analoga anche per il campo “Client ID” (Fig. 3.33).

Dati Richiesta

Scope	<input type="text"/>	i
Elencare più scope separandoli con la virgola		
Audience	<input type="text"/>	i
Client ID	<input type="text" value="ClientID dell'applicativo ModI"/>	i
Resource	<input type="text" value="resource"/>	i

Figure3.33: Token Policy di Negoziazione PDND (DatiRichiesta, ClientId definito nell'applicativo ModI)

Nel caso invece in cui sia stato indicato un keystore definito nella fruizione ModI, è possibile selezionare una modalità analoga anche per il campo “Client ID” (Fig. 3.34).

Dati Richiesta

Scope	<input type="text"/>	i
Elencare più scope separandoli con la virgola		
Audience	<input type="text"/>	i
Client ID	<input type="text" value="ClientID della fruizione ModI"/>	i
Resource	<input type="text"/>	i

Figure3.34: Token Policy di Negoziazione PDND (DatiRichiesta, ClientId definito nella fruizione ModI)

Fruizione

Una volta effettuata la registrazione della Token Policy, per utilizzarla in una fruizione è sufficiente associarla al connettore della fruizione come descritto nella sezione *Autenticazione Token* e mostrato nella figura Fig. 3.35.

Nel caso sia stata configurata l'opzione “definito nella fruizione ModI” per il keystore, il KID o l'identificativo client nella token policy di negoziazione selezionata è possibile configurare tali parametri nella sezione “ModI - Authorization PDND” come mostrato nella figura Fig. 3.36.

Maggiori dettagli sulla configurazione del keystore nella fruizione vengono forniti nella sezione “*Keystore di firma definito nella fruizione*”.

Applicativo Client

La registrazione dell'applicativo avviene come già descritto nella sez. *Creazione di un applicativo*.

Le ulteriori configurazioni descritte di seguito sono necessarie solamente se si intende associare all'applicativo richiedente il keystore utilizzato per la firma del token di sicurezza. Non sono invece necessari ulteriori passi di configurazione se il keystore viene definito nella fruizione o nella token policy e si rimanda rispettivamente alle sezioni “*Keystore di firma definito nella fruizione*” e “*Keystore di firma definito nella token policy*” per ulteriori dettagli di questi scenari.

In questo contesto sarà necessario specificare il dominio «Interno» dell'applicativo e procedere all'inserimento dei dati nel form «ModI - Sicurezza Messaggio - KeyStore» (Fig. 3.37).

I dati da inserire definiscono il keystore contenente la coppia di chiavi utilizzata per firmare i token di sicurezza:

- *Modalità*: il keystore può essere fornito tramite differenti modalità
 - “File System”: deve essere fornito il *Path* assoluto su file system del keystore;
 - “Archivio”: viene effettuato l'upload del keystore;

Figure3.35: Fruizione con pattern «ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization PDND”

Figure3.36: Fruizione con pattern «ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization PDND”

ModI - Sicurezza Messaggio

KeyStore

Abilitato

Modalità

Path *****

Tipo

Password *****

Alias Chiave Privata *****

Password Chiave Privata *****

Certificato No file chosen

Authorization ModI

Identificativo Client (i)

Identificativo dell'Applicativo scambiato nei token di sicurezza

URL (x5u) (i)

URL che riferisce un certificato (o certificate chain) X.509 corrispondente alla chiave firmataria del security token

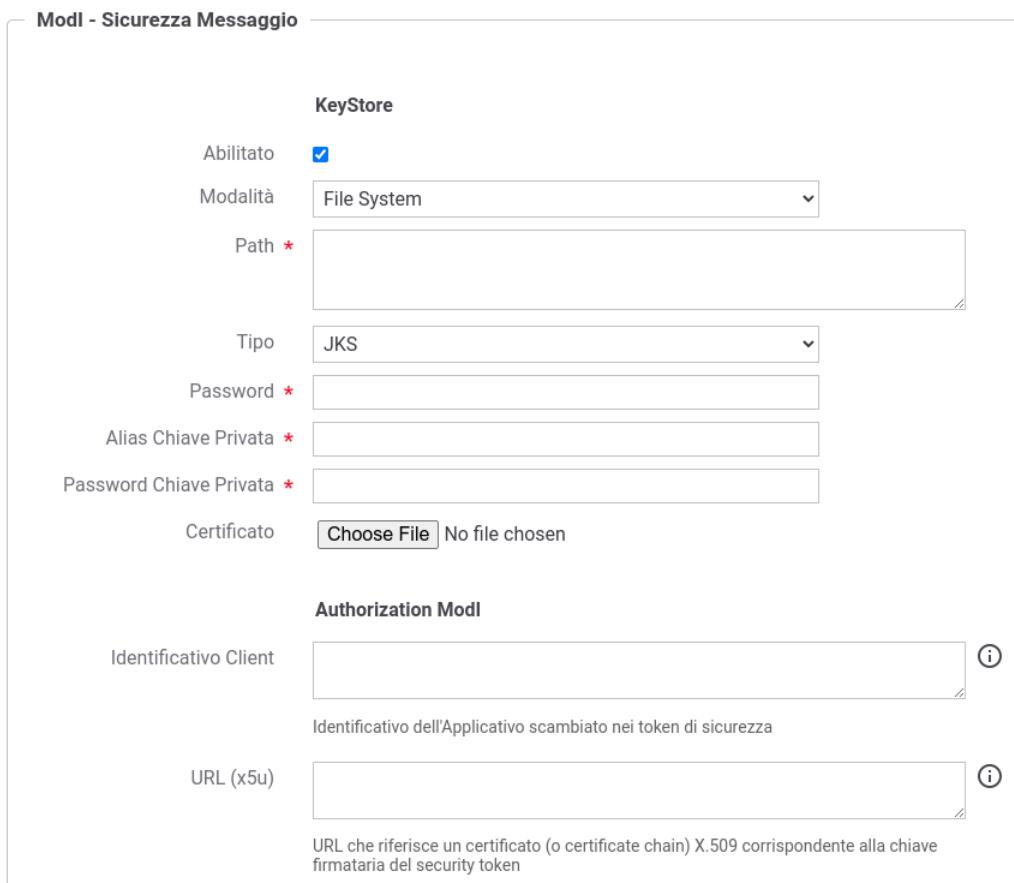


Figure3.37: Dati ModI relativi ad un applicativo interno

- “HSM”: consente di selezionare uno dei tipi di keystore PKCS11 registrati (“*Device PKCS11*”);
- *Tipo*: il formato del keystore:
 - “JKS” o “PKCS12” (disponibile con modalità “File System” e “Archivio”): viene richiesta la definizione della password per l’accesso al keystore nel campo *Password*, l’alias con cui è riferita la chiave privata nel keystore nel campo *Alias Chiave Privata* e la password della chiave privata nel campo *Password Chiave Privata*;
 - “JWK Set” (disponibile con modalità “File System”): deve essere definito il path su filesystem dove risiede l’archivio json nel formato “JWK Set” e l’identificativo “kid” (alias) con cui è riferita la chiave privata nel campo *Alias Chiave Privata*;
 - “Key Pair” (disponibile con modalità “File System”): deve essere definito il path su filesystem dove risiedono la chiave privata e pubblica in formato PEM o DER (sono supportati sia i formati pkcs1 che pkcs8) e la password della chiave privata se cifrata nel campo *Password Chiave Privata*;
 - Tipi PKCS11 (disponibile con modalità “HSM”): i tipi disponibili sono quelli corrispondenti ai tipi di keystore PKCS11 registrati (“*Device PKCS11*”).
- *Certificato*: nel caso di modalità “File System”, con tipi di keystore “JKS” o “PKCS12”, o nel caso di modalità “HSM” consente di caricare il certificato corrispondente alla chiave privata del keystore. Il Certificato, altrimenti disponibile solamente a runtime sui nodi run di GovWay, viene utilizzato sia per motivi di ricerche filtrate sulla console che per consentire l’identificazione dell’applicativo su API erogate da altri soggetti di dominio interno in un contesto MultiTenant (“*Multi-Tenant*”).

Oltre ai dati che definiscono il keystore, nella sezione “Authorization OAuth”, è possibile definire aspetti che riguardano il KID o l’identificativo client da inserire nella richiesta del voucher alla PDND nel caso sia stata configurata l’opzione “definito nell’applicativo ModI” nella token policy di negoziazione per i suddetti campi, come mostrato nella figura Fig. 3.38.

Authorization OAuth

Abilitato

Token Policy di Validazione

Identificativo *

Key Id (kid) del Certificato

Figure3.38: Dati Autorizzazione OAuth relativi ad un applicativo interno

Erogazione ID_AUTH_REST_01 (PDND)

In un’erogazione di una API le richieste provengono da amministrazioni esterne al dominio e sono dirette ad applicativi interni. Prima di procedere con l’inoltro della richiesta verso il backend interno, GovWay valida il token di sicurezza ricevuto rispetto al pattern associato all’operazione invocata: verifica firma, validazione temporale, filtro duplicati, verifica integrità del messaggio, verifica del token di audit etc.

Nella figura “Fig. 3.39” viene raffigurato lo scenario di erogazione in cui il trust avviene tramite la PDND.

Di seguito vengono descritti tutti i passi di configurazione specifici per l’implementazione del pattern “ID_AUTH_REST_01” mentre si rimanda alla sezione “*Profilo “API Gateway”*” per la normale registrazione e configurazione di un’erogazione di API.

API

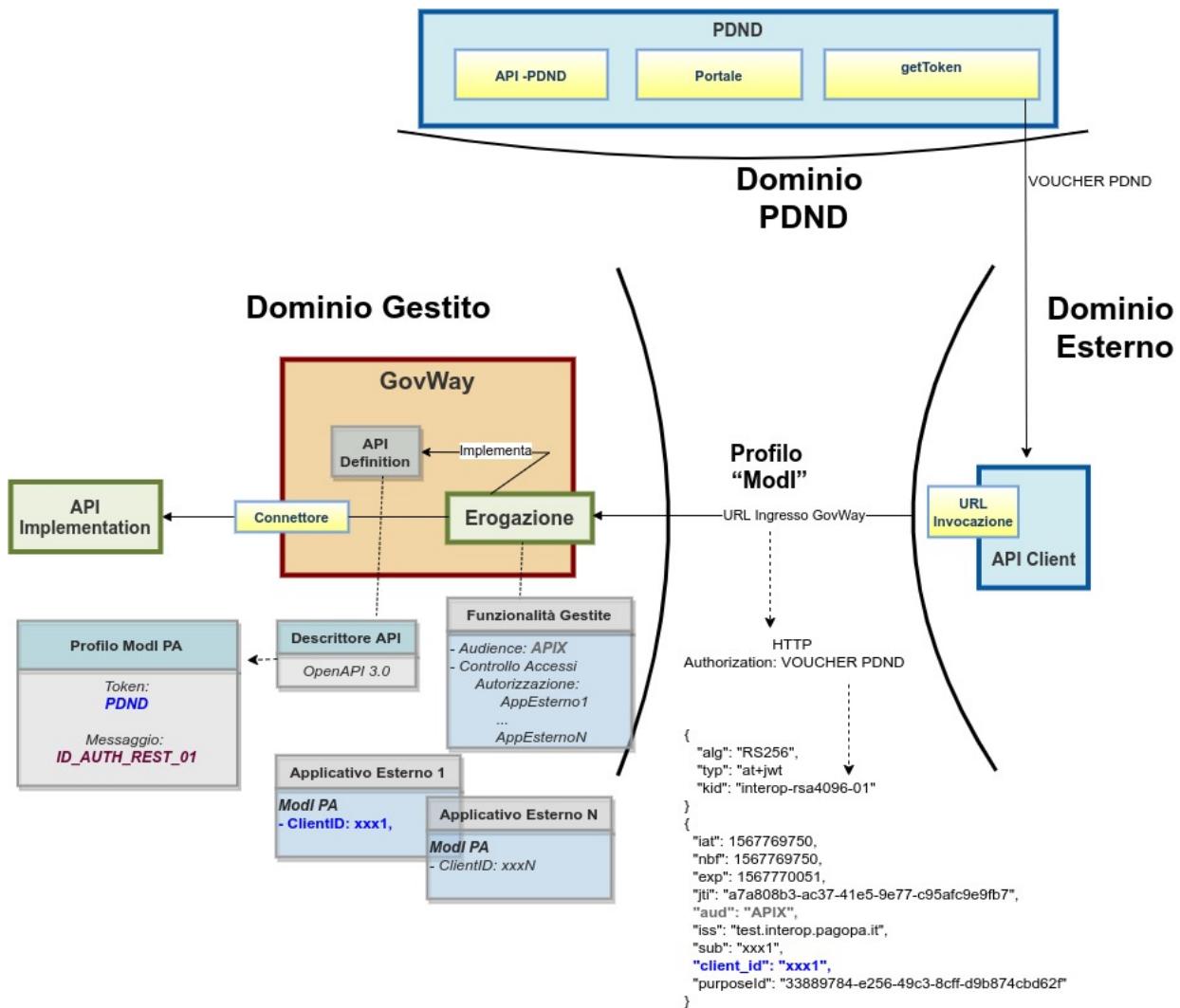


Figure3.39: Erogazione con Profilo di Interoperabilità “ModI”, pattern “ID_AUTH_REST_01”: trust tramite PDND

La registrazione della API deve essere effettuata agendo nella sezione «ModI - Sicurezza Messaggio», come indicato in Fig. 3.40:

- selezionare il “Pattern” «ID_AUTH_REST_01»;
- selezionare una “Generazione Token” di tipo “Authorization PDND” per far sì che il Token “ID_AUTH” sia negoziato con la PDND.

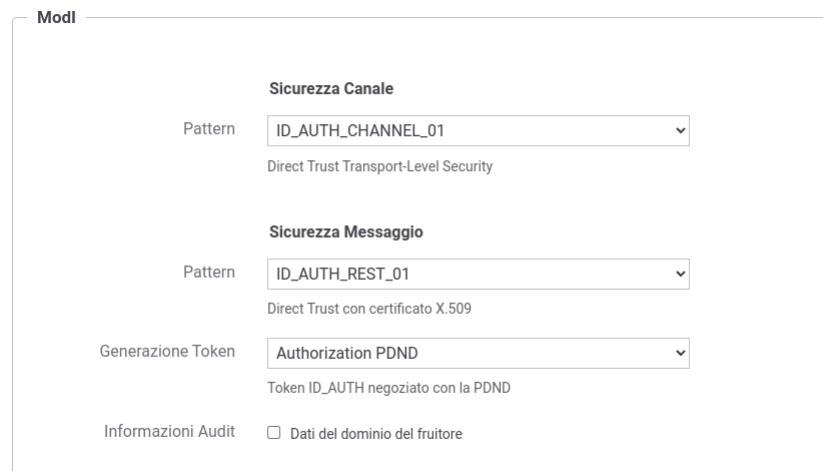


Figure 3.40: Selezione del pattern «ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization PDND” per l’API

Token Policy di Validazione

Per la configurazione di una erogazione con un pattern di sicurezza via PDND viene fornita built-in la token policy “PDND” di cui deve essere stata effettuata la configurazione come descritto nella sezione “[Trust tramite PDND](#)”.

Erogazione

Una volta effettuata la registrazione della Token Policy, per utilizzarla in un’erogazione è necessaria attivarla come policy di autenticazione token nel controllo degli accessi come descritto nella sezione [Autenticazione Token](#).

L’associazione avviene direttamente durante la creazione dell’erogazione come mostrato nella figura Fig. 3.41.

Verifica Audience

Per verificare l’audience presente nel token ricevuto dalla PDND deve essere utilizzata l’[Autorizzazione per Token Claims](#) definendo il claim “aud” uguale al valore atteso (Fig. 3.42).

Identificazione ed Autorizzazione dei fruitori

È possibile registrare gli applicativi dei domini esterni al fine di:

1. identificare puntualmente le componenti esterne coinvolte nella comunicazione abilitando le funzionalità di tracciamento e statistica per tali elementi.
2. abilitare le funzionalità di autorizzazione sugli applicativi identificando puntualmente chi autorizzare dopo il superamento del processo di validazione del token ricevuto (Fig. 3.47).

Rispetto a quanto descritto nella sezione “[ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509](#)” il token ricevuto non è più firmato dall’applicativo mittente ma bensì dall’authorization server della PDND e l’identificazione dell’applicativo chiamante non è più attuabile tramite il certificato fornito nell’header del JWT tramite claim “x5c/x5t/x5u” ma bensì tramite l’identificativo presente nel claim “client_id”.

Per poter identificare gli applicativi chiamanti la modalità di caricamento del certificato di firma, descritto nelle sezioni “[Fruizione ID_AUTH_REST_01 / ID_AUTH_SOAP_01 \(X509\)](#)” e “[Erogazione ID_AUTH_REST_01 /](#)

Informazioni Generali

Soggetto Erogatore: ENTE

API

Nome: * Esempio v1

Tipo: Rest

Nome Erogazione: * Esempio

Versione: 1

Autenticazione Token

Policy: * PDND

Validazione JWT: abilitato

Connettore

Tipo: http

Endpoint: * https://backend

Autenticazione Http:

Autenticazione Token:

Proxy:

Ridefinisci Tempi Risposta:

Opzioni Avanzate:

Debug: govway_connettori.log (i)

Figure3.41: Erogazione con pattern «ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization PDND”

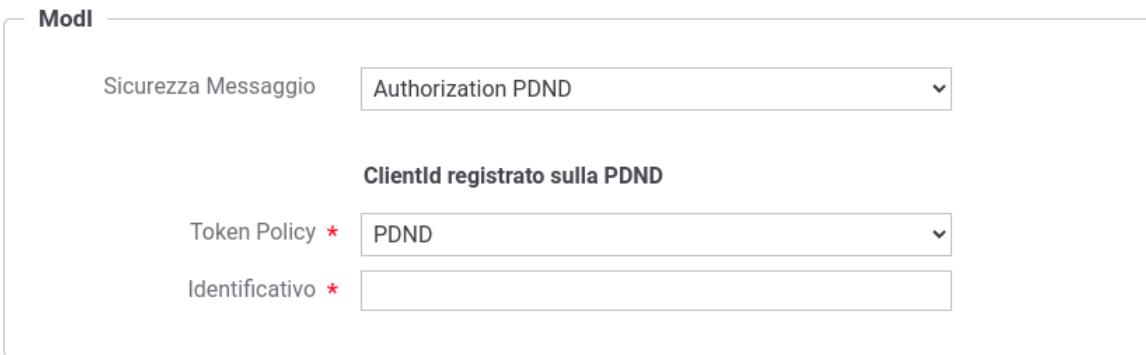
^ Autorizzazione

Stato	<input type="text" value="abilitato"/>
Autorizzazione Canale	
per Richiedente	<input type="checkbox"/>
per Ruoli	<input type="checkbox"/>
Autorizzazione Messaggio	
per Richiedente	<input checked="" type="checkbox"/>
Applicativi (4)	
per Ruoli	<input type="checkbox"/>
Autorizzazione per Token Claims	
Abilitato	<input checked="" type="checkbox"/>
Claims	<input type="text" value="aud=http://demoServer/myService"/> ⓘ

Indicare per riga i claims richiesti (nome=valore); visualizzare 'info' per maggiori dettagli

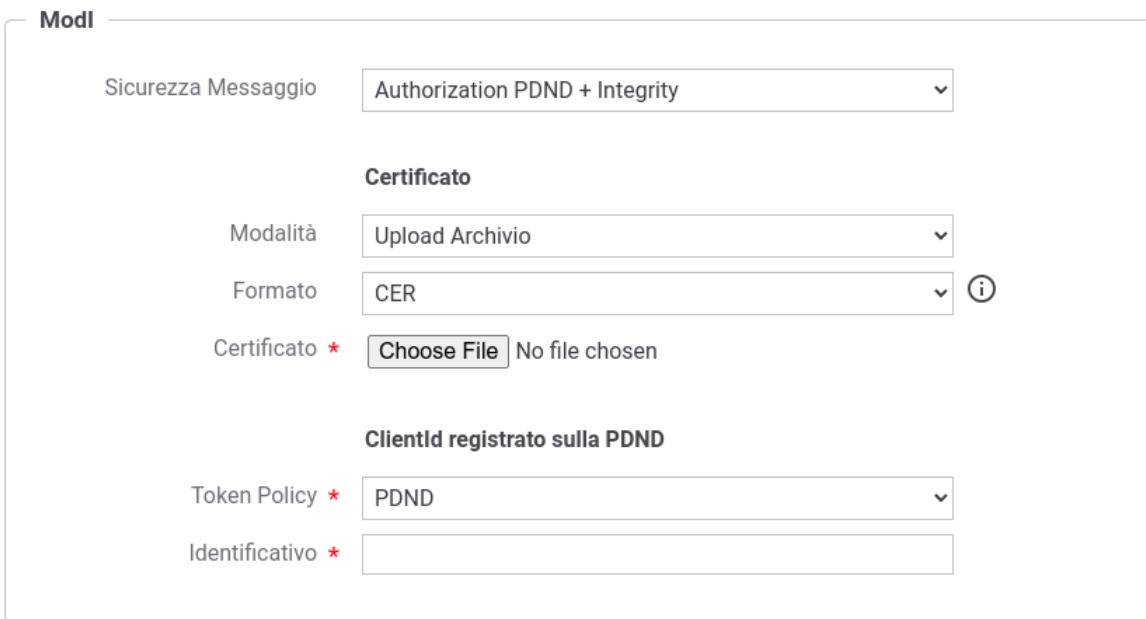
Figure3.42: Autorizzazione dell'audience presente nel token

ID_AUTH_SOAP_01 (X509)”, non è più necessaria mentre si dovranno fornire i dati relativi al token OAuth (Fig. 3.43) o in alternativa aggiungendo tali dati a quelli relativi al certificato (Fig. 3.44).



The screenshot shows the 'Modi' configuration interface. At the top, there is a dropdown menu labeled 'Sicurezza Messaggio' with the value 'Authorization PDND'. Below this, there is a section titled 'ClientId registrato sulla PDND' containing two fields: 'Token Policy *' (set to 'PDND') and 'Identificativo *' (empty). The entire interface is enclosed in a light gray box with a thin border.

Figure3.43: Dati ModI relativi ad un applicativo esterno con configurazione token PDND



The screenshot shows the 'Modi' configuration interface. At the top, there is a dropdown menu labeled 'Sicurezza Messaggio' with the value 'Authorization PDND + Integrity'. Below this, there is a section titled 'Certificato' containing three fields: 'Modalità' (set to 'Upload Archivio'), 'Formato' (set to 'CER'), and 'Certificato *' (containing a 'Choose File' button and the message 'No file chosen'). Below this, there is another section titled 'ClientId registrato sulla PDND' containing two fields: 'Token Policy *' (set to 'PDND') and 'Identificativo *' (empty). The entire interface is enclosed in a light gray box with a thin border.

Figure3.44: Dati ModI relativi ad un applicativo esterno con configurazione sia del certificato di firma che del token PDND

Una configurazione simile è attuabile anche sugli applicativi di dominio interno per poterli riconoscere su installazioni Multi-Tenant (“*Multi-Tenant*”) dove sia il tenant fruitore che quello erogatore viene gestito sullo stesso GovWay (Fig. 3.45).

Una volta registrati gli applicativi client è possibile attuare criteri di autorizzazione dei singoli applicativi accedendo alla configurazione della sezione «Controllo Accessi» e attivando la sicurezza messaggio. Sarà possibile specificare un elenco puntuale di applicativi autorizzati (Fig. 3.47). In alternativa è possibile definire i ruoli che gli applicativi devono possedere.

Modi - Sicurezza Messaggio

KeyStore

Abilitato

Authorization OAuth

Abilitato

Token Policy di Validazione: PDND

Identificativo *

Figure3.45: Dati ModI relativi ad un applicativo interno con configurazione token PDND

ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509

Nota: La sigla che identifica il pattern di sicurezza messaggio varia a seconda se l'API sia di tipo REST, per cui la sigla corrisponde a *ID_AUTH_REST_01*, o SOAP dove viene utilizzata la sigla *ID_AUTH_SOAP_01*.

L'adozione di questo pattern consente, alla ricezione di un messaggio, di validare il certificato fornito dall'applicativo mittente, la porzione di messaggio firmata, la validità temporale nonché la corrispondenza del destinatario della comunicazione.

API

La registrazione della API deve essere effettuata agendo nella sezione «Modi - Sicurezza Messaggio» come indicato rispettivamente per una API REST in [Fig. 3.48](#) e per una API SOAP in [Fig. 3.49](#):

- selezionare il “Pattern” «ID_AUTH_REST_01» su API REST o “ID_AUTH_SOAP_01” su API SOAP;
- selezionare una “Generazione Token” di tipo “Authorization ModI” per far sì che il Token “ID_AUTH” sia generato dalla parte mittente.

Le voci “Header HTTP del Token” (presente solamente su API di tipo REST) e “Applicabilità” consentono di personalizzare l’header HTTP utilizzato e di indicare se il pattern di sicurezza verrà attuato sia sulla richiesta che sulla risposta. Maggiori informazioni vengono fornite rispettivamente nelle sezioni “[Header HTTP del token JWT](#)” e “[Attivazione della sicurezza messaggio su richiesta/risposta](#)”.

Nel contesto della configurazione della specifica operation/risorsa è presente anche la sezione «Sicurezza Messaggio» che consente di intervenire sul pattern di sicurezza messaggio in modo puntuale. È quindi possibile lasciare l’impostazione del pattern al valore già stabilito a livello della API, oppure decidere di ridefinirlo andando a fornire una configurazione specifica per la singola operation/risorsa come descritto nella sezione “[Attivazione di pattern di sicurezza messaggio differenti per la singola operazione](#)”.

Di seguito vengono forniti i dettagli di configurazione richiesti in uno scenario di fruizione o erogazione di un servizio.

Modi - Sicurezza Messaggio

KeyStore

Abilitato

Modalità

Path

Tipo

Password

Alias Chiave Privata

Password Chiave Privata

Certificato No file chosen
ExampleClient2.crt

Authorization ModI

Identificativo Client ⓘ

Identificativo dell'Applicativo scambiato nei token di sicurezza

URL (x5u) ⓘ

URL che riferisce un certificato (o certificate chain) X.509 corrispondente alla chiave firmataria del security token

Authorization OAuth

Abilitato

Token Policy di Validazione

Identificativo

Key Id (kid) del Certificato

Figure3.46: Dati ModI relativi ad un applicativo interno con configurazione sia del certificato di firma che del token PDND

Autorizzazione

Stato	<input type="text" value="abilitato"/>
Autorizzazione Canale	
per Richiedente	<input type="checkbox"/>
per Ruoli	<input type="checkbox"/>
Autorizzazione Messaggio	
per Richiedente	<input checked="" type="checkbox"/>
Applicativi (4)	
per Ruoli	<input type="checkbox"/>
Autorizzazione per Token Claims	
Abilitato	<input type="checkbox"/>

Figure3.47: Autorizzazione di singoli applicativi token per l'accesso all'erogazione

Modi

Sicurezza Canale	
Pattern	<input type="text" value="ID_AUTH_CHANNEL_01"/>
Direct Trust Transport-Level Security	
Sicurezza Messaggio	
Pattern	<input type="text" value="ID_AUTH_REST_01"/>
Direct Trust con certificato X.509	
Generazione Token	<input type="text" value="Authorization ModI"/>
Token ID_AUTH generato dal mittente secondo le Linee Guida 'ModI'	
Header HTTP del Token	<input type="text" value="Authorization Bearer"/>
Applicabilità	<input type="text" value="Richiesta e Risposta"/>
Informazioni Audit	<input type="checkbox"/> Dati del dominio del fruitore

Figure3.48: Pattern di sicurezza messaggio «ID_AUTH_REST_01» per l'API

The screenshot shows a configuration interface for 'Modi' (ModI) security settings. The interface is divided into several sections:

- Sicurezza Canale**: Pattern is set to **ID_AUTH_CHANNEL_01** (Direct Trust Transport-Level Security).
- Sicurezza Messaggio**: Pattern is set to **ID_AUTH_SOAP_01** (Direct Trust con certificato X.509).
- Generazione Token**: Set to **Authorization ModI** (Token ID_AUTH generato dal mittente secondo le Linee Guida 'ModI').
- Applicabilità**: Set to **Richiesta e Risposta**.
- Informazioni Audit**: Includes a checkbox for **Dati del dominio del fruitore**.

Figure3.49: Pattern di sicurezza messaggio «ID_AUTH_SOAP_01» per l'API

Fruizione ID_AUTH_REST_01 / ID_AUTH_SOAP_01 (X509)

Le richieste che provengono dagli applicativi interni del dominio e sono dirette verso altre amministrazioni verranno arricchite del token di sicurezza “ModI” previsto dall’operazione invocata, come indicato precedentemente nella sezione [ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509](#).

Per la configurazione delle fruizioni con i pattern di sicurezza messaggio è necessario registrare ciascun applicativo interno coinvolto al fine principale di associargli una chiave privata e un certificato X509 che GovWay utilizza per firmare il token di sicurezza “ModI” prodotto. Gli applicativi vengono identificati da GovWay tramite una delle modalità di autenticazione supportate descritte nella sezione [Autenticazione Trasporto](#) (Fig. 3.50).

Nella figura “Fig. 3.8” viene raffigurato lo scenario di fruizione in cui il trust avviene tra fruitore ed erogatore tramite certificati x509.

API

La registrazione della API deve essere effettuata seguendo le indicazioni descritte nella sezione [ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509](#)

Fruizione

L’interfaccia per la creazione della fruizione, basata su una API con pattern «ID_AUTH_REST_01» (o «ID_AUTH_SOAP_01»), presenta le sezioni «ModI - Richiesta» e «ModI - Risposta»:

- ModI - Richiesta (Fig. 3.51): la maschera relativa alla richiesta prevede la configurazione del meccanismo di firma digitale del messaggio, ad opera dell’applicativo mittente, e la produzione del relativo token di sicurezza:
 - Algoritmo: l’algoritmo che si vuole utilizzare per la firma digitale del messaggio
 - Riferimento X.509: il metodo da utilizzare per l’inserimento del certificato dell’applicativo nel token di sicurezza. I valori possibili sono (differenziati per il caso REST e SOAP) quelli previsti nelle Linee Guida di Interoperabilità.
 - Certificate Chain: se è stata selezionata la modalità “x5c”, è possibile indicare se nel token di sicurezza verrà incluso solo il certificato utilizzato per la firma o l’intera catena.
 - KeyStore: lo scenario descritto in questa sezione è relativo alla configurazione proposta di default con la voce “Definito nell’applicativo”. Uno scenario differente è attuabile utilizzando la configurazione descritta in [Keystore di firma definito nella fruizione](#).

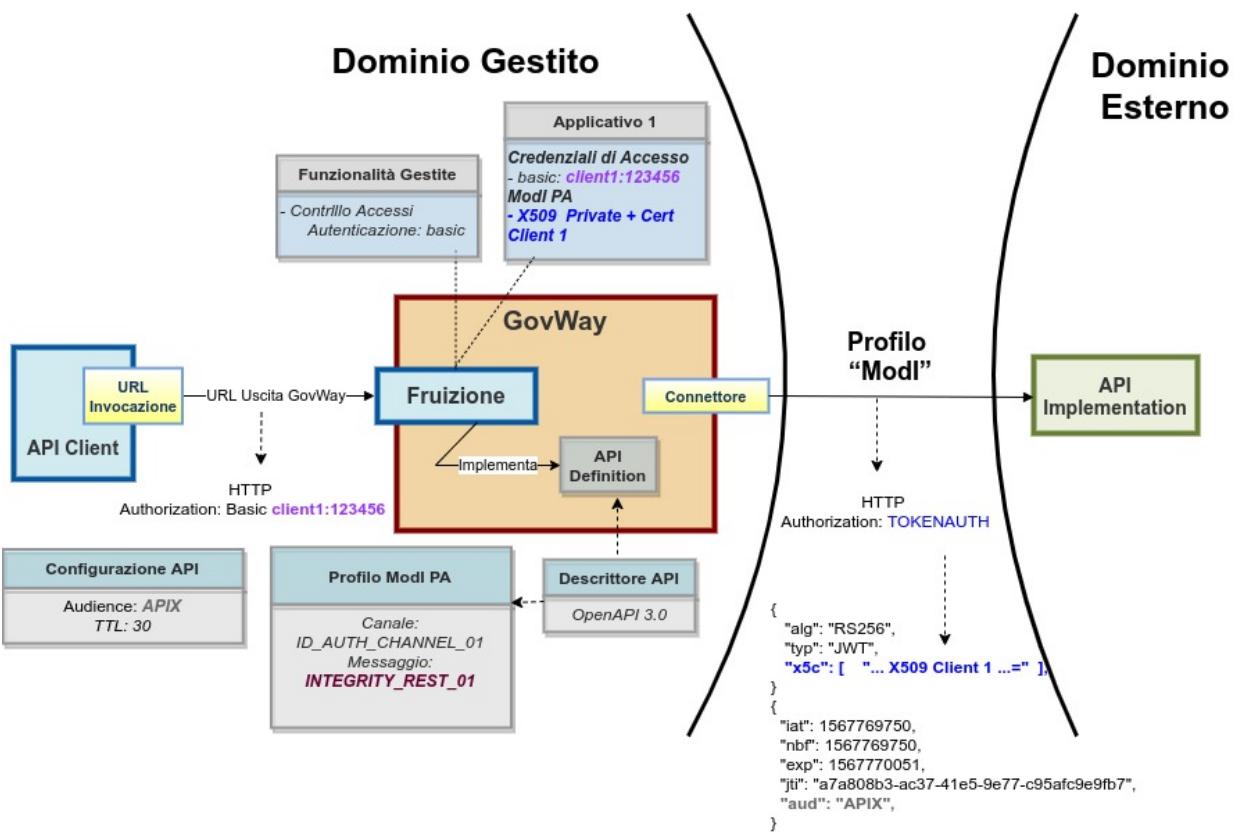


Figure3.50: Fruizione con Profilo di Interoperabilità “ModI”, pattern “ID_AUTH_REST_01”: trust tra fruitore ed erogatore tramite certificati x509

- Time to Live: tempo di validità del token prodotto (in secondi)
- Audience: identificativo dell'applicativo destinatario da indicare come audience nel token di sicurezza; se non viene indicato alcun valore verrà utilizzata la url del connettore. Il valore fornito può contenere parti dinamiche risolte a runtime dal Gateway (per maggiori dettagli [Valori dinamici](#)).

ModI - Richiesta

Sicurezza Messaggio	
Algoritmo	RS256
Riferimento X.509	<input type="text" value="x5c (Certificate)"/> x5t#256 (Certificate SHA-256 Thumbprint) x5u (URL)
Certificate Chain	<input type="checkbox"/>
Time to Live (secondi) *	300
Indica la validità temporale, in secondi, a partire dalla data di creazione del security token	
Audience	http://ente/RestBlocking
Indica a chi è riferito il security token; se non viene fornito un valore verrà utilizzata la url del connettore	

Figure3.51: Dati per la configurazione della sicurezza messaggio sulla richiesta di una fruizione

- ModI - Risposta (Fig. 3.52): la maschera relativa alla risposta prevede la configurazione del meccanismo di validazione del token ricevuto da parte dell'applicativo destinatario:
 - Riferimento X.509: il metodo per la localizzazione del certificato del destinatario nel messaggio di risposta. Si può mantenere la medesima impostazione prevista per il messaggio di richiesta o ridefinirla.
 - TrustStore Certificati: Riferimento al truststore che contiene le CA, i certificati, CRL e policy OCSP da utilizzare per poter verificare i token di sicurezza ricevuti nelle risposte. È possibile mantenere l'impostazione di default che è stata fornita al momento dell'installazione del prodotto, oppure definire un diverso riferimento (opzione «Ridefinito») fornendo Path, Tipo, Password del TrustStore e criteri di verifica tramite CRL o OCSP.
 - Time to Live (secondi): consente di modificare l'intervallo temporale di default (300 secondi) utilizzato per rifiutare i token creati precedentemente all'intervallo indicato.
 - Verifica Audience: Se l'opzione è abilitata, viene effettuata la verifica che il campo Audience, presente nel token di sicurezza della risposta, corrisponda al valore presente nel campo successivo, se indicato, o altrimenti a quello configurato nell'applicativo mittente nella voce “Identificativo Client”.

Applicativo Client

La registrazione dell'applicativo avviene come già descritto nella sez. [Creazione di un applicativo](#).

In questo contesto sarà necessario specificare il dominio «Interno» dell'applicativo e procedere all'inserimento dei dati nel form «ModI» (Fig. 3.53).

I dati da inserire definiscono il keystore contenente la coppia di chiavi utilizzata per firmare i token di sicurezza:

Modi - Risposta

Sicurezza Messaggio

Riferimento X.509

TrustStore Certificati

Time to Live (secondi)

* I token creati precedentemente all'intervallo temporale indicato, in secondi, verranno rifiutati

Verifica Audience La verifica utilizza, se configurato, il valore indicato di seguito altrimenti quello configurato nell'applicativo mittente

TrustStore Certificati

Tipo

Path *

Password *

OCSP Policy

CRL File(s)

Elencare più file separandoli con la ''

Figure3.52: Dati per la configurazione della sicurezza messaggio sulla risposta di una fruizione

ModI - Sicurezza Messaggio

KeyStore

Abilitato

Modalità

Path *****

Tipo

Password *****

Alias Chiave Privata *****

Password Chiave Privata *****

Certificato No file chosen

Authorization ModI

Identificativo Client (i)

Identificativo dell'Applicativo scambiato nei token di sicurezza

URL (x5u) (i)

URL che riferisce un certificato (o certificate chain) X.509 corrispondente alla chiave firmataria del security token

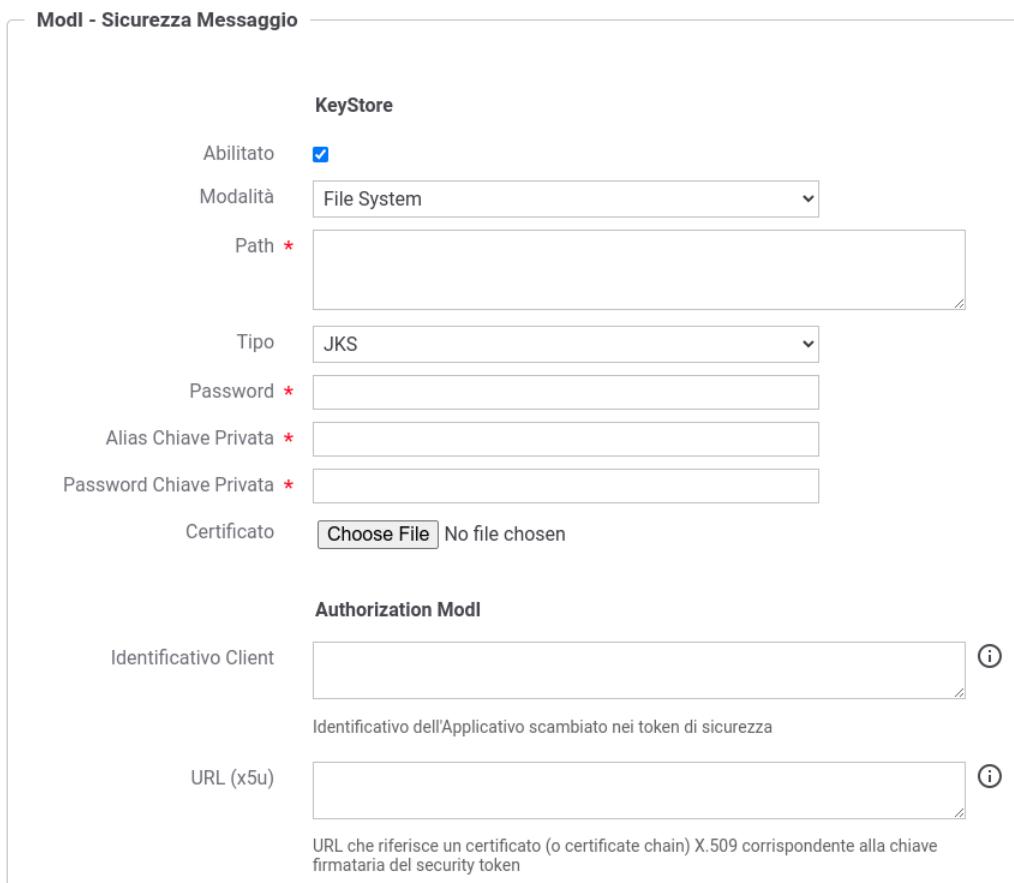


Figure3.53: Dati ModI relativi ad un applicativo interno

- *Modalità*: il keystore può essere fornito tramite differenti modalità
 - “File System”: deve essere fornito il *Path* assoluto su file system del keystore;
 - “Archivio”: viene effettuato l’upload del keystore;
 - “HSM”: consente di selezionare uno dei tipi di keystore PKCS11 registrati (“*Device PKCS11*”);
- *Tipo*: il formato del keystore:
 - “JKS” o “PKCS12” (disponibile con modalità “File System” e “Archivio”): viene richiesta la definizione della password per l’accesso al keystore nel campo *Password*, l’alias con cui è riferita la chiave privata nel keystore nel campo *Alias Chiave Privata* e la password della chiave privata nel campo *Password Chiave Privata*;
 - “JWK Set” o “Key Pair” (disponibile con modalità “File System”): questa modalità non è utilizzabile per fruire di API con una “Generazione Token” di tipo “Authorization ModI” ed utilizzando una applicativo con tale configurazione si otterrà un errore a runtime;
 - Tipi PKCS11 (disponibile con modalità “HSM”): i tipi disponibili sono quelli corrispondenti ai tipi di keystore PKCS11 registrati (“*Device PKCS11*”).
- *Certificato*: nel caso di modalità “File System”, con tipi di keystore “JKS” o “PKCS12”, o nel caso di modalità “HSM” consente di caricare il certificato corrispondente alla chiave privata del keystore. Il Certificato, altrimenti disponibile solamente a runtime sui nodi run di GovWay, viene utilizzato sia per motivi di ricerche filtrate sulla console che per consentire l’identificazione dell’applicativo su API erogate da altri soggetti di dominio interno in un contesto MultiTenant (“*Multi-Tenant*”).

Oltre ai dati che definiscono il keystore, nella sezione “Authorization ModI”, è possibile definire aspetti che riguardano la generazione del token di sicurezza ModI “ID_AUTH” e “INTEGRITY”:

- *Identificativo Client*: identificativo dell’applicativo utilizzato per valorizzare nel token di sicurezza di una richiesta il claim “client_id” per API REST e l’header “wsa:From” per API SOAP (**Attenzione**: se non definito viene utilizzato il nome dell’applicativo). Se è abilitata la funzionalità “Verifica Audience / WSAddressing To” nella configurazione della sicurezza della risposta verrà inoltre verificato che nel token di sicurezza della risposta ricevuto vi sia un claim “aud” per API REST o un header “wsa:To” per API SOAP che possiede un valore identico all’identificato fornito.
- *URL (x5u)*: URL che riferisce un certificato (o certificate chain) X.509 corrispondente alla chiave firmataria del security token. Deve essere obbligatoriamente definito se l’applicativo fruisce di API REST configurate per generare un token di sicurezza tramite il claim “x5u”

Erogazione ID_AUTH_REST_01 / ID_AUTH_SOAP_01 (X509)

In un’erogazione di una API le richieste provengono da amministrazioni esterne al dominio e sono dirette ad applicativi interni. Prima di procedere con l’inoltro della richiesta verso il backend interno, GovWay valida il token di sicurezza ricevuto rispetto al pattern associato all’operazione invocata: verifica firma, validazione temporale, filtro duplicati, verifica integrità del messaggio, verifica del token di audit etc.

Nella figura “Fig. 3.54” viene raffigurato lo scenario di erogazione in cui il trust avviene tra fruitore ed erogatore tramite certificati x509.

API

La registrazione della API deve essere effettuata seguendo le indicazioni descritte nella sezione *ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509*

Erogazione

L’interfaccia per la creazione dell’erogazione, basata su una API con pattern «ID_AUTH_REST_01» (o «ID_AUTH_SOAP_01»), presenta le sezioni «ModI - Richiesta» e «ModI - Risposta»:

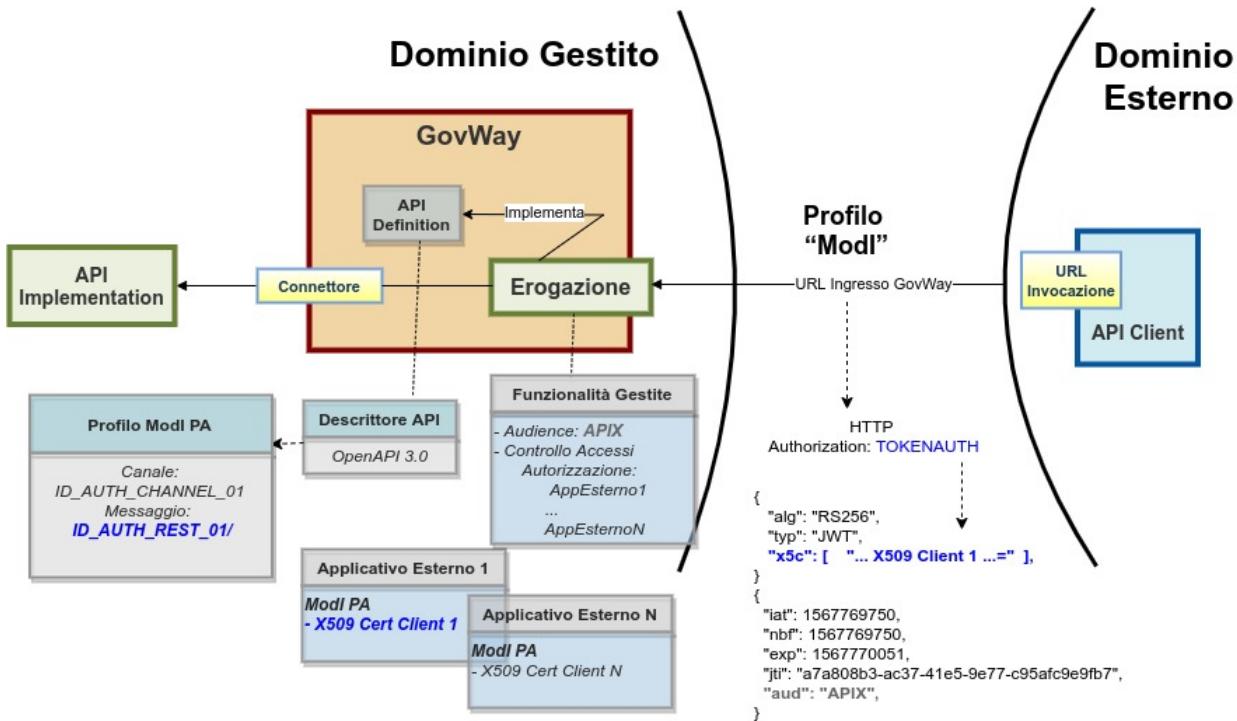


Figure3.54: Erogazione con Profilo di Interoperabilità “ModI”, pattern “ID_AUTH_REST_01”: trust tra fruitore ed erogatore tramite certificati x509

- **ModI - Richiesta (Fig. 3.55):** la maschera relativa alla richiesta prevede la configurazione del meccanismo di validazione del token ricevuto sul messaggio di richiesta:
 - Riferimento X.509 (presente solo per API REST): il metodo per la localizzazione del certificato dell'applicativo mittente nel messaggio di richiesta. Il valore fornito deve corrispondere alla scelta operata dai mittenti. I valori possibili sono quelli previsti nella specifica AGID.
 - TrustStore Certificati: Riferimento al truststore che contiene le CA, i certificati, CRL e policy OCSP da utilizzare per poter verificare i token di sicurezza ricevuti nelle richieste. È possibile mantenere l'impostazione di default che è stata fornita al momento dell'installazione del prodotto, oppure definire un diverso riferimento (opzione «Ridefinito») fornendo Path, Tipo, Password del TrustStore e criteri di verifica tramite CRL o OCSP.
 - Time to Live (secondi): consente di modificare l'intervallo temporale di default (300 secondi) utilizzato per rifiutare i token creati precedentemente all'intervallo indicato.
 - Audience: valore del campo Audience atteso nel token di sicurezza della richiesta.
- **ModI - Risposta (Fig. 3.56):** la maschera prevede la configurazione del meccanismo di firma digitale del messaggio di risposta, e la produzione del relativo token di sicurezza, da inviare all'applicativo mittente:
 - Algoritmo: l'algoritmo che si vuole utilizzare per la firma digitale del messaggio di risposta
 - Riferimento X.509: il metodo da utilizzare per l'inserimento del certificato nel messaggio di risposta. Nelle API di tipo REST si può mantenere la medesima impostazione prevista per il messaggio di richiesta o ridefinirla.
 - Certificate Chain: se è stata selezionata la modalità “x5c”, è possibile indicare se nel token di sicurezza verrà incluso solo il certificato utilizzato per la firma o l'intera catena.
 - Keystore: il keystore da utilizzare per la firma del messaggio di risposta. È possibile mantenere il

Modi - Richiesta

Sicurezza Messaggio

Riferimento X.509

x5c (Certificate)
x5t#256 (Certificate SHA-256 Thumbprint)
URL (x5u)

TrustStore Certificati

Ridefinito

Time to Live (secondi)

Ridefinito

* 300

I token creati precedentemente all'intervallo temporale indicato, in secondi, verranno rifiutati

Audience

testsuite

Se non viene fornito un valore, il valore atteso all'interno del security token corrisponderà all'url di invocazione

TrustStore Certificati

Tipo

JKS

Path *

/etc/govway/keys/xca/trustStore_certificati.jks

Password *

.....

OCSP Policy

-

CRL File(s)

Elencare più file separandoli con la ','

Figure3.55: Dati per la configurazione della sicurezza messaggio sulla richiesta di una erogazione

riferimento al keystore di default, fornito in fase di installazione del prodotto, oppure indicare un diverso riferimento (opzione «Ridefinito») fornendo il path sul filesystem, o in alternativa direttamente l'archivio, unitamente a Tipo, Password, Alias Chiave Privata e Password Chiave Privata.

- Time to Live (secondi): validità temporale del token prodotto.

Modi - Risposta

Sicurezza Messaggio

Algoritmo	RS256
HTTP Headers da firmare *	<input checked="" type="button" value="Digest"/> <input checked="" type="button" value="Content-Type"/> <input checked="" type="button" value="Content-Encoding"/>
Riferimento X.509	Utilizza impostazioni della Richiesta
Certificate Chain	<input type="checkbox"/>
KeyStore	Ridefinito
Time to Live (secondi) *	300

Indica la validità temporale, in secondi, a partire dalla data di creazione del security token della risposta

KeyStore

Modalità	File System
Path *	/etc/govway/keys/xca/ExampleServer.p12
Tip	pkcs12
Password *	123456
Alias Chiave Privata *	ExampleServer
Password Chiave Privata *	123456

Figure3.56: Dati per la configurazione della sicurezza messaggio sulla risposta di una erogazione

Identificazione ed Autorizzazione dei fruitori

È possibile registrare gli applicativi dei domini esterni al fine di:

1. identificare puntualmente le componenti esterne coinvolte nella comunicazione abilitando le funzionalità di tracciamento e statistica per tali elementi.
2. abilitare le funzionalità di autorizzazione sugli applicativi identificando puntualmente chi autorizzare dopo il superamento del processo di autenticazione/autorizzazione canale e validazione del token di sicurezza.

Per abilitare quanto al punto 1 è sufficiente la sola registrazione degli applicativi esterni coinvolti (Fig. 3.57).

Applicativo

Dominio: Esterno

Soggetto:

Nome *

Figure3.57: Registrazione di un applicativo esterno

Dopo aver indicato il dominio «Esterno» per l'applicativo, sarà necessario selezionare il soggetto che identifica il dominio esterno di riferimento.

La registrazione dell'applicativo esterno comprende anche la sezione con i dati relativi alla sicurezza messaggio (Fig. 3.58).

ModI

Sicurezza Messaggio: Authorization ModI

Certificato

Modalità: Upload Archivio

Formato: CER

Certificato *: Choose File No file chosen

Figure3.58: Dati ModI relativi ad un applicativo esterno con upload del certificato

I dati da fornire sono:

- *Modalità*: si seleziona tra il caricamento del certificato e la configurazione manuale
- Caso *Upload Archivio*:
 - *Formato*: formato dell'archivio fornito (CER, JKS; PKCS12)
 - *Certificato*: elemento per l'upload dell'archivio che contiene il certificato
 - *Reply Audience/WSA-To*: identificativo dell'applicativo da confrontare con il valore «Audience» eventualmente presente nelle richieste.
- Caso *Configurazione Manuale* (Fig. 3.59):
 - *Self Signed*: opzione per indicare se il certificato è self-signed oppure rilasciato da una CA
 - *Subject*: il subject del certificato
 - *Issuer*: l'issuer del certificato, nel caso in cui non sia self-signed
 - *Reply Audience/WSA-To*: identificativo dell'applicativo da confrontare con il valore «Audience» eventualmente presente nelle richieste.

Modi

Sicurezza Messaggio Authorization ModI

Certificato

Modalità Configurazione Manuale

Self Signed

Subject *

Issuer

Figure3.59: Dati ModI relativi ad un applicativo esterno con configurazione manuale dei dati di sicurezza

Per abilitare le funzionalità di autorizzazione dei singoli applicativi (punto 2 del precedente elenco) si deve procedere alla configurazione della sezione «Controllo Accessi» relativa all’erogazione. Quando attiva la sicurezza messaggio, questa sezione conterrà il form «Autorizzazione Messaggio» (Fig. 3.60). Qui è possibile specificare un elenco puntuale di applicativi (esterni) autorizzati, ad accedere all’erogazione, tra quelli identificati nella fase di verifica del relativo certificato. Gli applicativi esterni saranno selezionabili tra quelli censiti nella sezione «Applicativi» (Fig. 3.60). In alternativa è possibile definire i ruoli che gli applicativi devono possedere.

^ Autorizzazione

Stato abilitato

Autorizzazione Canale

per Richiedente

per Ruoli

Autorizzazione Messaggio

per Richiedente

[Applicativi \(4\)](#)

per Ruoli

Autorizzazione per Token Claims

Abilitato

Figure3.60: Autorizzazione di singoli applicativi per l’accesso all’erogazione

Nota: L’autorizzazione basata sugli identificativi degli applicativi mittenti del dominio fruitore esterno, è possibile soltanto se è stata effettuata la registrazione degli applicativi interessati, in associazione al soggetto esterno di riferimento.

ID_AUTH_REST_01 tramite un Authorization Server OAuth differente dalla PDND

Il token di autenticazione ID_AUTH_REST_01 descritto nella sezione “[ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509](#)” viene generato e firmato dall’applicativo mittente.

Un token con analoga struttura descritto nella sezione “[ID_AUTH_REST_01 tramite la Piattaforma Digitale Nazionale Dati \(PDND\)](#)” viene invece rilasciato dalla Piattaforma Digitale Nazionale Dati (PDND).

Un token di autenticazione che rispetti la struttura prevista dal pattern “ID_AUTH_REST_01”, descritto nella sezione “[ID_AUTH_REST_01 tramite la Piattaforma Digitale Nazionale Dati \(PDND\)](#)”, può essere rilasciato in maniera simile anche da un authorization server differente dalla PDND. GovWay consente di implementare questo scenario con configurazioni molto simili a quelle già descritte nelle sezioni “[Fruizione ID_AUTH_REST_01 \(PDND\)](#)” e “[Erogazione ID_AUTH_REST_01 \(PDND\)](#)”.

Di seguito vengono forniti i dettagli di configurazione necessari ad utilizzare un authorization server negli scenari di fruizione o erogazione di un servizio.

Fruizione ID_AUTH_REST_01 (Authorization Server)

Di seguito vengono descritti tutti i passi di configurazione che differiscono da quando già descritto nella sezione “[Fruizione ID_AUTH_REST_01 \(PDND\)](#)”.

API

La registrazione della API deve essere effettuata agendo nella sezione «ModI - Sicurezza Messaggio», come indicato in [Fig. 3.61](#):

- selezionare il “Pattern” «ID_AUTH_REST_01»;
- selezionare una “Generazione Token” di tipo “Authorization OAuth” per far sì che il Token “ID_AUTH” sia negoziato con un Authorization Server OAuth.

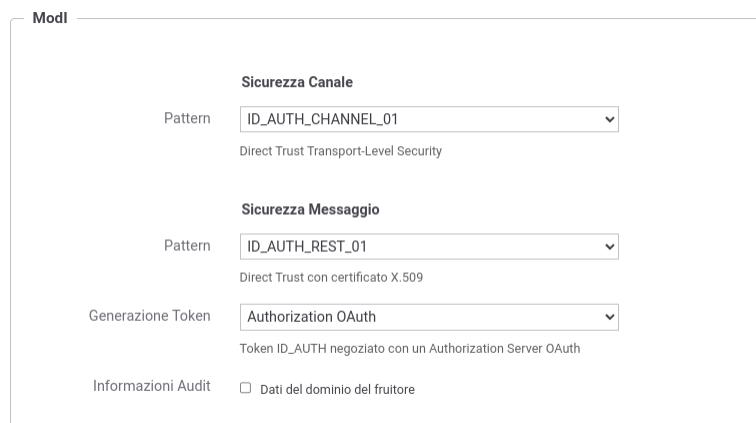


Figure3.61: Selezione del pattern «ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization OAuth” per l’API

Token Policy di Negoziazione

Per la configurazione della fruizione è necessario registrare una Token Policy di Negoziazione con uno qualsiasi dei tipi descritti nella sezione “[Token Policy Negoziazione](#)”.

Fruizione

Una volta effettuata la registrazione della Token Policy, per utilizzarla in una fruizione è sufficiente associarla al connettore della fruizione come descritto nella sezione [Autenticazione Token](#) e mostrato nella figura [Fig. 3.62](#).

Figure3.62: Fruizione con pattern «ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization OAuth”

Erogazione ID_AUTH_REST_01 (Authorization Server)

Di seguito vengono descritti tutti i passi di configurazione che differiscono da quando già descritto nella sezione “[Erogazione ID_AUTH_REST_01 \(PDND\)](#)”.

API

La registrazione della API deve essere effettuata agendo nella sezione «ModI - Sicurezza Messaggio», come indicato in [Fig. 3.63](#):

- selezionare il “Pattern” «ID_AUTH_REST_01»;
- selezionare una “Generazione Token” di tipo “Authorization OAuth” per far sì che il Token “ID_AUTH” sia negoziato con un Authorization Server OAuth.

Token Policy di Validazione

Per la configurazione dell’erogazione è necessario registrare una Token Policy di Validazione descritta nella sezione “[Token Policy Validazione](#)”.

Erogazione

Una volta effettuata la registrazione della Token Policy, per utilizzarla in un’erogazione è necessaria attivarla come policy di autenticazione token nel controllo degli accessi come descritto nella sezione [Autenticazione Token](#).

L’associazione avviene direttamente durante la creazione dell’erogazione come mostrato nella figura [Fig. 3.64](#).

Autorizzazioni ulteriori

Per poter identificare gli applicativi chiamanti si dovranno fornire i dati relativi al token OAuth ([Fig. 3.65](#))

Modi

Sicurezza Canale

Pattern Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern Direct Trust con certificato X.509

Generazione Token Token ID_AUTH negoziato con un Authorization Server OAuth

Informazioni Audit Dati del dominio del fruttore

Figure3.63: Selezione del pattern «ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization OAuth” per l’API

Informazioni Generali

API

Nome * Tipo Rest
Nome Erogazione * Versione

Autenticazione Token

Policy * Validazione JWT abilitato
Token Forward abilitato

Connettore

Endpoint * Autenticazione Http
 Autenticazione Token
 Proxy
 Ridefinisci Tempi Risposta
 Opzioni Avanzate
 Debug

Figure3.64: Erogazione con pattern «ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization OAuth”

Applicativi > Aggiungi

Note: (*) Campi obbligatori

Applicativo

Dominio	Esterno
Soggetto	DemoSoggettoFruitoreEsterno
Nome *	ApplicativoEsterno
Tipi	Client

ModI

Sicurezza Messaggio	Authorization OAuth
Identificativo registrato sull'Authorization Server	
Token Policy *	TestValidazioneJWT
Identificativo *	ClientIDApplicativo-Esempio-123456

Figure3.65: Dati ModI relativi ad un applicativo esterno con configurazione token OAuth

Una configurazione simile è attuabile anche sugli applicativi di dominio interno per poterli riconoscere su installazioni Multi-Tenant (“*Multi-Tenant*”) dove sia il tenant fruitore che quello erogatore viene gestito sullo stesso GovWay (Fig. 3.66).

Authorization OAuth	
Abilitato	<input checked="" type="checkbox"/>
Token Policy di Validazione	TestValidazioneForwardJWS
Identificativo *	IdentificativoClient-Esempio-12345
Key Id (kid) del Certificato	

Figure3.66: Dati ModI relativi ad un applicativo interno con configurazione token OAuth

3.3.3 ID_AUTH_SOAP_02 / ID_AUTH_REST_02

Il pattern ID_AUTH nelle sue varie declinazioni della versione “02” ha lo scopo, oltre a identificare e autorizzare l’accesso del fruitore ad un servizio, anche quello di evitare Replay Attack poichè ogni richiesta non può essere nuovamente processata.

Analizzando il pattern rispetto al tipo di trust:

- trust tramite PDND: in attesa di ulteriori indicazioni, il pattern non sembra essere utilizzabile con un trust tramite PDND, poichè richiederebbe una negoziazione di un nuovo token per ogni richiesta per garantirne l’unicità.
- trust tra fruitore ed erogatore tramite certificati X509: descritto nella sezione “*ID_AUTH_SOAP_02 / ID_AUTH_REST_02 - Direct Trust con certificato X.509 con unicità del messaggio/token*” differisce nella sostanza se applicato per API REST, in cui viene prodotto un token JWT con identificativo univoco inserito nel claim “*jwt*”, o per API SOAP, in cui viene definito un header SOAP WSSecurity e un identificativo univoco presente nell’header SOAP WSAddressing:MessageID. In entrambi i casi il certificato del mittente viene inserito all’interno del token di sicurezza e validato dall’erogatore tramite un trustStore contenente i certificati X509 attesi.

ID_AUTH_SOAP_02 / ID_AUTH_REST_02 - Direct Trust con certificato X.509 con unicità del messaggio/token

Nota: La sigla che identifica il pattern di sicurezza messaggio varia a seconda se l’API sia di tipo REST, per cui la sigla corrisponde a *ID_AUTH_REST_02*, o SOAP dove viene utilizzata la sigla *ID_AUTH_SOAP_02*.

Questo pattern di sicurezza presenta le medesime caratteristiche di *ID_AUTH_SOAP_01/ID_AUTH_REST_01 - Direct Trust con certificato X.509*, con l’unica differenza di prevedere un meccanismo di filtro che impedisce la ricezione di messaggi duplicati da parte di ciascun ricevente.

La registrazione della API deve essere effettuata agendo nella sezione «Modi - Sicurezza Messaggio» come indicato rispettivamente per una API REST in Fig. 3.67 e per una API SOAP in Fig. 3.68

- selezionare il “Pattern” «*ID_AUTH_REST_02*» su API REST o “*ID_AUTH_SOAP_02*” su API SOAP;
- selezionare una “Generazione Token” di tipo “Authorization ModI” per far sì che il Token “*ID_AUTH*” sia generato dalla parte mittente.

Modi

Sicurezza Canale

Pattern Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern Direct Trust con certificato X.509 con unicità del token

Generazione Token Token ID_AUTH generato dal mittente secondo le Linee Guida 'Modl'

Header HTTP del Token

Applicabilità

Informazioni Audit Dati del dominio del fruitore

Figure3.67: Pattern di sicurezza messaggio «ID_AUTH_REST_02» per l'API

Modi

Sicurezza Canale

Pattern Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern Direct Trust con certificato X.509 con unicità del messaggio

Generazione Token Token ID_AUTH generato dal mittente secondo le Linee Guida 'Modl'

Applicabilità

Informazioni Audit Dati del dominio del fruitore

Figure3.68: Pattern di sicurezza messaggio «ID_AUTH_SOAP_02» per l'API

Le configurazioni successive alla registrazione della API sono le medesime già descritte in precedenza per il pattern *ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509*.

3.3.4 INTEGRITY_SOAP_01 / INTEGRITY_REST_01 / INTEGRITY_REST_02

Il pattern INTEGRITY nelle sue varie declinazioni ha lo scopo di garantire l'integrità del payload del messaggio.

Il pattern differisce rispetto al tipo di trust:

- trust tramite PDND: descritto nella sezione “*INTEGRITY_REST_02 - Integrità del payload delle request REST in PDND*”, e applicabile solo per API REST, prevede l'indicazione all'interno del token di un identificativo della chiave pubblica (kid) associata alla chiave privata utilizzata dal client per firmare il token; identificativo kid generato dalla PDND e recuperabile dall'erogatore tramite le API messe a disposizione dalla PDND stessa.
- trust tra fruitore ed erogatore tramite certificati X509: descritto nella sezione “*INTEGRITY_SOAP_01 / INTEGRITY_REST_01 - Integrità payload del messaggio*” differisce nella sostanza se applicato per API REST, in cui viene prodotto un token JWT, o per API SOAP, in cui viene definito un header SOAP WSSecurity. In entrambi i casi il payload viene firmato e ne viene verificata l'integrità dall'erogatore.
- trust ibrido: descritto nella sezione “*ID_AUTH tramite PDND + INTEGRITY_SOAP_01 / INTEGRITY_REST_01*” prevede un trust tramite PDND per il token ID_AUTH e un trust tra fruitore ed erogatore con certificati x509 per il token INTEGRITY.

INTEGRITY_REST_02 - Integrità del payload delle request REST in PDND

Questo pattern di sicurezza consente di estendere il pattern «ID_AUTH_REST_01» aggiungendo un meccanismo che garantisce l'integrità del payload del messaggio.

Il pattern prevede l'indicazione all'interno del token di un identificativo della chiave pubblica (kid) associata alla chiave privata utilizzata dal client per firmare il token. Tale identificativo kid viene generato dalla PDND quando viene registrato il materiale crittografico (chiave pubblica) ed è recuperabile dall'erogatore tramite le API messe a disposizione dalla PDND stessa.

Di seguito vengono forniti i dettagli di configurazione aggiuntivi, rispetto ai passi descritti nella sezione “*ID_AUTH_REST_01 tramite la Piattaforma Digitale Nazionale Dati (PDND)*”, per gli scenari di fruizione o erogazione di un servizio.

Fruizione INTEGRITY_REST_02 (PDND)

Le richieste che provengono dagli applicativi interni del dominio e sono dirette verso altre amministrazioni verranno arricchite del token di sicurezza “ModI” previsto dall'operazione invocata, come indicato precedentemente nella sezione *INTEGRITY_REST_02 - Integrità del payload delle request REST in PDND*.

Nella figura “Fig. 3.69” viene raffigurato lo scenario di fruizione in cui il trust avviene tramite la PDND e viene prodotto il token “Agid-JWT-Signature” previsto dal pattern “INTEGRITY_REST_02”.

Di seguito vengono forniti i dettagli di configurazione aggiuntivi o differenti, rispetto ai passi descritti nella sezione “*Fruizione ID_AUTH_REST_01 (PDND)*”.

API

La registrazione della API deve essere effettuata agendo nella sezione «ModI - Sicurezza Messaggio», come indicato in Fig. 3.70:

- selezionare il “Pattern” «INTEGRITY_REST_02 con ID_AUTH_REST_01»;

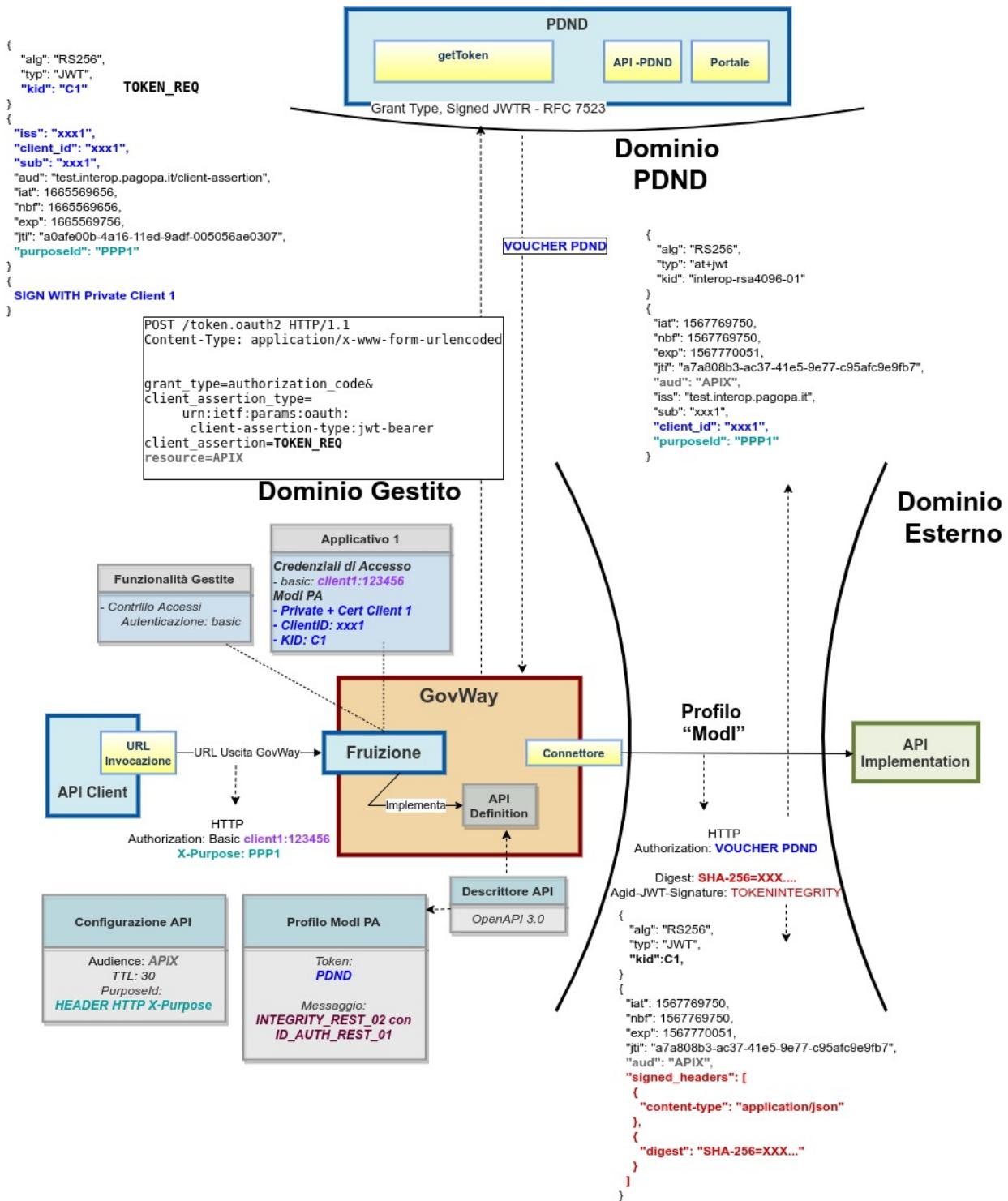


Figure3.69: Fruizione con Profilo di Interoperabilità “Modi”, pattern “INTEGRITY_REST_02”: trust tramite PDND

- selezionare una “Generazione Token” di tipo “Authorization PDND” per far sì che il Token “ID_AUTH” sia negoziato con la PDND.

ModI

Sicurezza Canale	
Pattern	ID_AUTH_CHANNEL_01
Direct Trust Transport-Level Security	
Sicurezza Messaggio	
Pattern	INTEGRITY_REST_02 con ID_AUTH_REST_01
Integrità payload del messaggio	
Generazione Token	Authorization PDND
Token ID_AUTH negoziato con la PDND	
Header HTTP del Token	Agid-JWT-Signature + Authorization Bearer
Applicabilità	Richiesta e Risposta
Digest Richiesta	<input type="checkbox"/> Non ripudiabilità della trasmissione i
Informazioni Audit	<input type="checkbox"/> Dati del dominio del fruitore

Figure3.70: Selezione del pattern «INTEGRITY_REST_02 con ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization PDND” per l’API

Le voci “Header HTTP del Token” e “Applicabilità” consentono di personalizzare l’header HTTP utilizzato e di indicare se il pattern di sicurezza verrà attuato sia sulla richiesta che sulla risposta. Maggiori informazioni vengono fornite rispettivamente nelle sezioni “[Header HTTP del token JWT](#)” e “[Attivazione della sicurezza messaggio su richiesta/risposta](#)”.

Fruizione

Rispetto a quanto indicato nella sezione [Fruizione ID_AUTH_REST_01 \(PDND\)](#) l’interfaccia per la creazione della fruizione, basata su una API con pattern «INTEGRITY_REST_02 con ID_AUTH_REST_01», presenta le sezioni «ModI - Richiesta» e «ModI - Risposta» (compatibilmente con i criteri di applicabilità impostati nell’API):

- ModI - Richiesta (Fig. 3.71): la maschera relativa alla richiesta prevede la configurazione del meccanismo di firma digitale del messaggio, ad opera dell’applicativo mittente, e la produzione del relativo token di sicurezza:
 - Algoritmo: l’algoritmo che si vuole utilizzare per la firma digitale del messaggio;
 - Codifica Digest: consente di selezionare l’algoritmo utilizzato per produrre il digest;
 - HTTP Headers da firmare: indicazione degli eventuali Header HTTP da firmare;
 - KeyStore: lo scenario descritto in questa sezione è relativo alla configurazione proposta di default con la voce “Definito nell’applicativo”, mentre scenari differenti sono attuabili utilizzando le configurazioni descritte in [Keystore di firma definito nella fruizione](#) e [Keystore di firma definito nella token policy](#);

- Time to Live: tempo di validità del token prodotto (in secondi);
- Audience: identificativo dell'applicativo destinatario da indicare come audience nel token di sicurezza; se non viene indicato alcun valore verrà utilizzata la url del connettore. Il valore fornito può contenere parti dinamiche risolte a runtime dal Gateway (per maggiori dettagli [Valori dinamici](#)).
- Claims: consente di personalizzare i claims presenti all'interno del token prodotto. Per maggiori dettagli si rimanda alla sezione “[Payload Claims del token JWT](#)”.

Modi - Richiesta

Sicurezza Messaggio

Algoritmo	RS256
Codifica Digest	Base64
HTTP Headers da firmare *	Digest x Content-Type x Content-Encoding x
KeyStore	Definito nell'applicativo
Time to Live (secondi) *	300
Indica la validità temporale, in secondi, a partire dalla data di creazione del security token	
Audience	<input type="text"/> i
Indica a chi è riferito il security token; se non viene fornito un valore verrà utilizzata la url del connettore	
Claims	<input type="text"/> i
Indicare per riga i claims (nome=valore); visualizzare 'info' per maggiori dettagli	

Figure3.71: Dati per la configurazione della sicurezza messaggio sulla richiesta di una fruizione

- ModI - Risposta: la maschera relativa alla risposta è presente solamente se è stato selezionato un criterio di applicabilità nell'API che prevede la ricezione di un token di sicurezza messaggio anche nella risposta. La sezione consente di configurare il meccanismo di validazione del token ricevuto da parte dell'applicativo destinatario:
 - TrustStore Certificati: il pattern “INTEGRITY_REST_02”, prevede che all'interno del token sia presente un identificativo della chiave pubblica (kid) associata alla chiave privata utilizzata dal destinatario per firmare il token di risposta. L'identificativo kid è stato generato dalla PDND al momento della registrazione del materiale crittografico (chiave pubblica) da parte dell'applicativo destinatario ed è recuperabile dal mittente tramite le [API messe a disposizione dalla PDND stessa](#). Per effettuare la validazione deve essere definito un truststore tramite una delle seguenti modalità alternative:
 - * “Default”: può essere utilizzato il truststore di default descritto nella sezione “[Trust tra fruitore ed erogatore tramite certificati X509](#)”; in questo caso nel truststore deve essere presente un certificato registrato con un alias che corrisponde al “kid” veicolato nel token INTEGRITY.
 - * “Ridefinito” con keystore di tipo “JWK Set” o “JKS”: consente di attuare una configurazione statica dove indicare il path su filesystem di un archivio json contenenti chiavi JWK o di un truststore

contenenti certificati x509. In entrambi i casi l'archivio indicato deve contenere al suo interno una chiave pubblica o un certificato registrato con un alias che corrisponde al “kid” veicolato nel token INTEGRITY. (Fig. 3.72).

Modi - Risposta

Sicurezza Messaggio

TrustStore Certificati: Ridefinito

Time to Live: Default

Verifica Audience: La verifica utilizza, se configurato, il valore indicato di seguito altrimenti quello configurato nell'applicativo mittente

TrustStore Certificati

Path: *

Figure3.72: Truststore “statico” per la validazione del token INTEGRITY sulla risposta di una fruizione

- * “Ridefinito” con keystore di tipo “PDND”: consente di attuare una configurazione dinamica in cui GovWay effettuerà lo scaricamento della chiave pubblica dalla PDND attraverso le [API PDND](#) se l'identificativo kid della chiave non è già presente nella cache locale (Fig. 3.73).

Nella figura “Fig. 3.74” viene raffigurato lo scenario di fruizione durante la fase di validazione del token di risposta tramite un truststore dinamico in cui GovWay utilizza le API PDND per ottenere la chiave pubblica necessaria a validare il token di risposta.

- Time to Live: consente di ridefinire l'intervallo temporale, in secondi, per il quale i token creati precedentemente all'intervallo indicato verranno rifiutati.
- Verifica Audience: Se l'opzione è abilitata, viene effettuata la verifica che il campo Audience, presente nel token di sicurezza della risposta, corrisponda al valore presente nel campo successivo, se indicato, o altrimenti a quello configurato nell'applicativo mittente nella voce “Identificativo Client”.

ModI - Risposta

Sicurezza Messaggio

TrustStore Certificati	Ridefinito
Time to Live	Default
Verifica Audience	<input checked="" type="checkbox"/> La verifica utilizza, se configurato, il valore indicato di seguito altrimenti quello configurato nell'applicativo mittente

TrustStore Certificati

Tipo	PDND
------	------

Figure3.73: Truststore “dinamico” per la validazione del token INTEGRITY sulla risposta di una fruizione

Erogazione INTEGRITY_REST_02 (PDND)

In un’erogazione di una API le richieste provengono da amministrazioni esterne al dominio e sono dirette ad applicativi interni. Prima di procedere con l’inoltro della richiesta verso il backend interno, GovWay valida il token di sicurezza ricevuto rispetto al pattern associato all’operazione invocata: verifica firma, validazione temporale, filtro duplicati, verifica integrità del messaggio, verifica del token di audit etc.

Nella figura “Fig. 3.75” viene raffigurato lo scenario di erogazione in cui il trust avviene tramite la PDND.

Di seguito vengono forniti i dettagli di configurazione aggiuntivi o differenti, rispetto ai passi descritti nella sezione “[Erogazione ID_AUTH_REST_01 \(PDND\)](#)”.

API

La registrazione della API deve essere effettuata agendo nella sezione «ModI - Sicurezza Messaggio», come indicato in [Fig. 3.76](#):

- selezionare il “Pattern” «ID_AUTH_REST_01»;
- selezionare una “Generazione Token” di tipo “Authorization PDND” per far sì che il Token “ID_AUTH” sia negoziato con la PDND.

Erogazione

Rispetto a quanto indicato nella sezione [Erogazione ID_AUTH_REST_01 \(PDND\)](#) l’interfaccia per la creazione dell’erogazione, basata su una API con pattern «INTEGRITY_REST_02 con ID_AUTH_REST_01», presenta le sezioni «ModI - Richiesta» e «ModI - Risposta» (compatibilmente con i criteri di applicabilità impostati nell’API):

- ModI - Richiesta: la maschera relativa alla richiesta prevede la configurazione del meccanismo di validazione del token ricevuto sul messaggio di richiesta:
 - TrustStore Certificati: il pattern “INTEGRITY_REST_02”, prevede che all’interno del token sia presente un identificativo della chiave pubblica (kid) associata alla chiave privata utilizzata dal mittente per firmare il token. L’identificativo kid è stato generato dalla PDND al momento della registrazione del materiale crittografico (chiave pubblica) da parte dell’applicativo mittente ed è recuperabile dall’erogatore tramite le

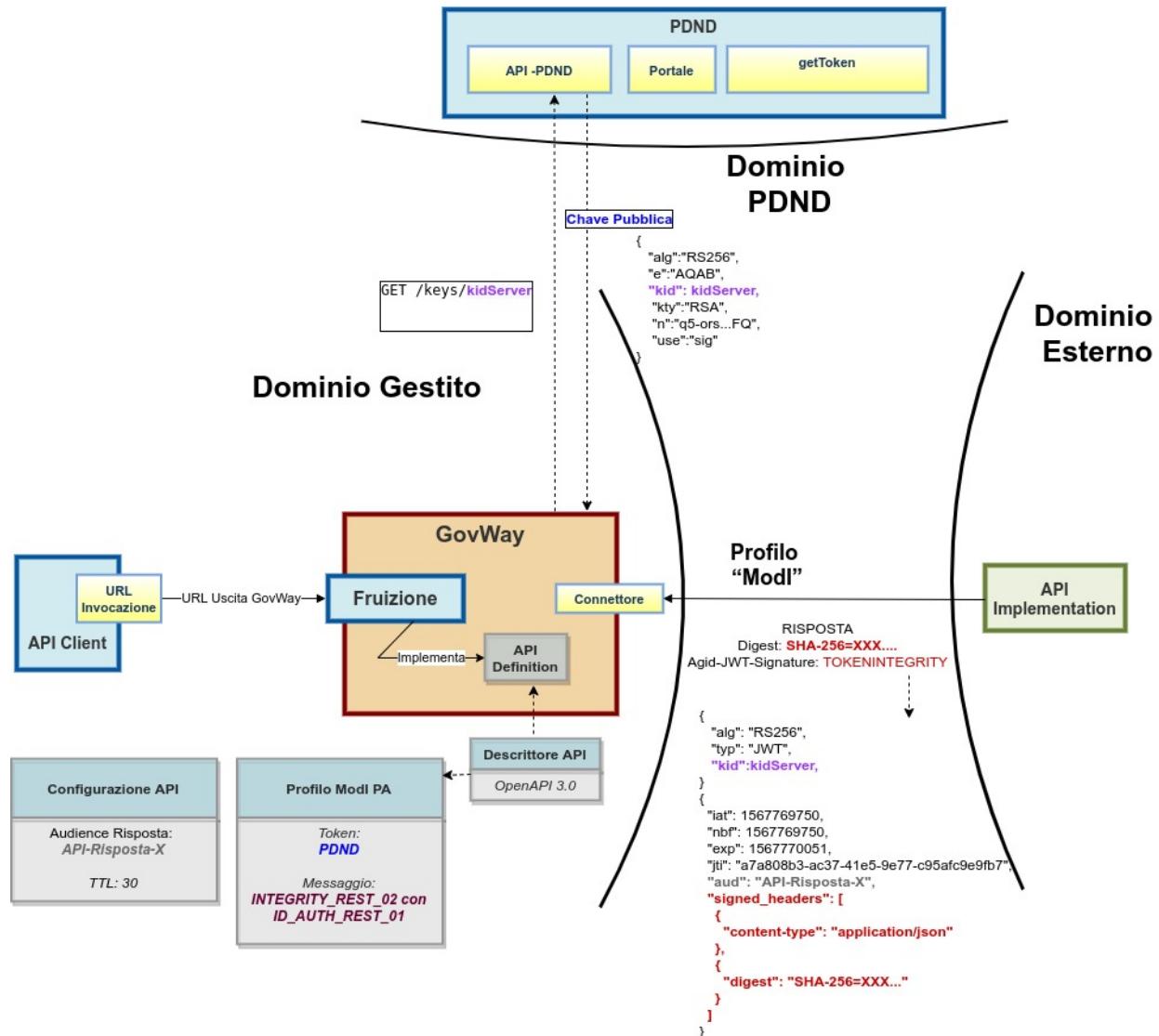


Figure3.74: Fruizione con Profilo di Interoperabilità “ModI”, pattern “INTEGRITY_REST_02”: trust tramite PDND e utilizzo delle API PDND per ottenere la chiave pubblica per validare la risposta

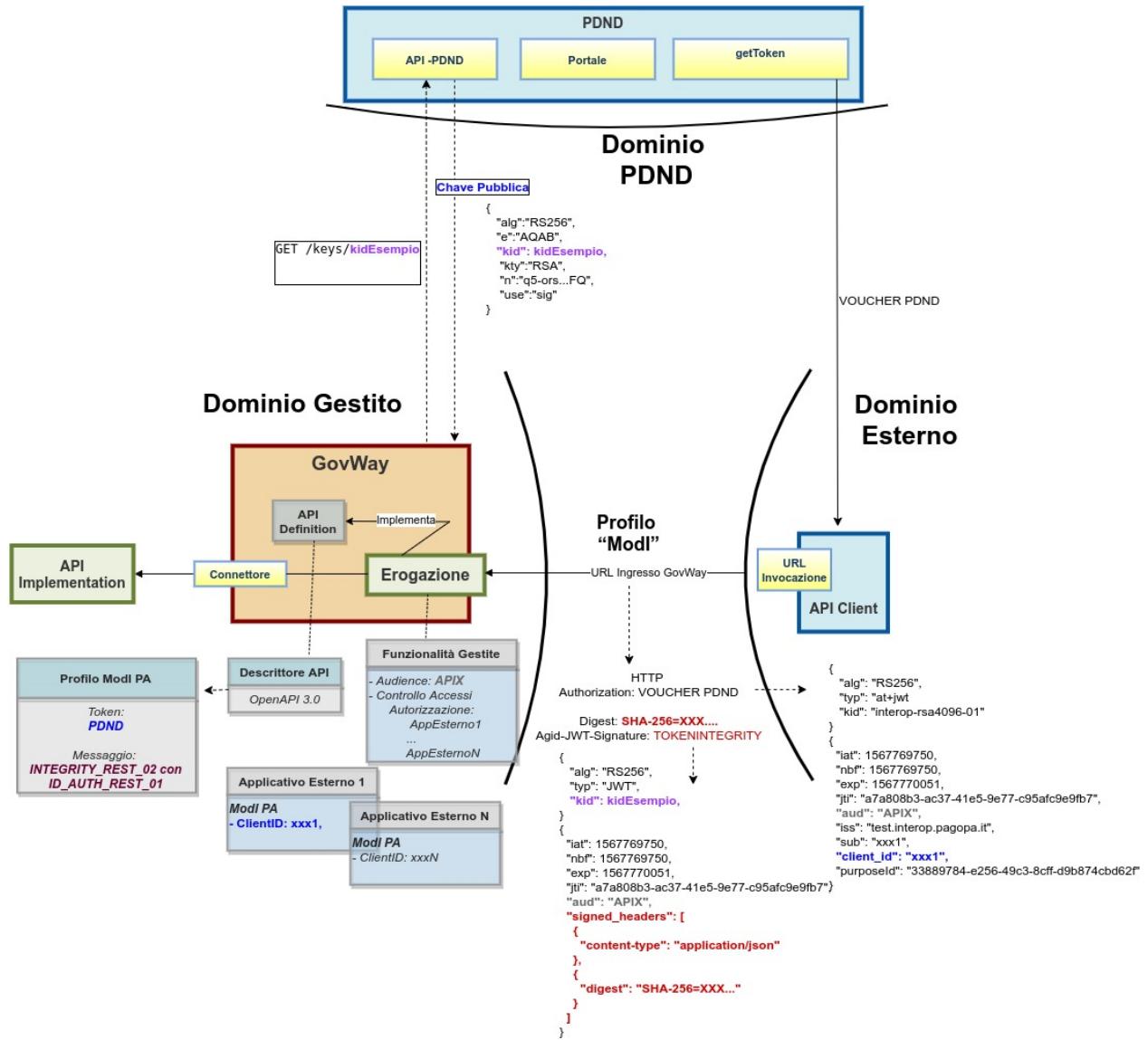


Figure3.75: Erogazione con Profilo di Interoperabilità “ModI”, pattern “INTEGRITY_REST_02”: trust tramite PDND

Modi

Sicurezza Canale

Pattern ▼
Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern ▼
Integrità payload del messaggio

Generazione Token ▼
Token ID_AUTH negoziato con la PDND

Header HTTP del Token ▼

Applicabilità ▼

Digest Richiesta Non ripudiabilità della trasmissione ⓘ

Informazioni Audit Dati del dominio del fruttore

Figure3.76: Selezione del pattern «INTEGRITY_REST_02 con ID_AUTH_REST_01» e “Generazione Token” di tipo “Authorization PDND” per l’API

API messe a disposizione dalla PDND stessa. Per effettuare la validazione deve essere definito un truststore tramite una delle seguenti modalità alternative:

- * “Default”: può essere utilizzato il truststore di default descritto nella sezione “[Trust tra fruitore ed erogatore tramite certificati X509](#)”; in questo caso nel truststore deve essere presente un certificato registrato con un alias che corrisponde al “kid” veicolato nel token INTEGRITY.
- * “Ridefinito” con keystore di tipo “JWK Set” o “JKS”: consente di attuare una configurazione statica dove indicare il path su filesystem di un archivio json contenenti chiavi JWK o di un truststore contenenti certificati x509. In entrambi i casi l’archivio indicato deve contenere al suo interno una chiave pubblica o un certificato registrato con un alias che corrisponde al “kid” veicolato nel token INTEGRITY. ([Fig. 3.77](#)).

Modi - Richiesta

Sicurezza Messaggio

TrustStore Certificati	<input type="text" value="Ridefinito"/>
Time to Live (secondi)	<input type="text" value="Ridefinito"/>
*	<input type="text" value="300"/>
I token creati precedentemente all’intervallo temporale indicato, in secondi, verranno rifiutati	
Audience	<input type="text"/>
Se non viene fornito un valore, il valore atteso all’interno del security token corrisponderà all’url di invocazione	
▼ Coesistenza Token Authorization e Agid-JWT-Signature	
TrustStore Certificati	
Tipo	<input type="text" value="JWK Set"/>
Path *	<input type="text"/>

Figure3.77: Truststore “statico” per la validazione del token INTEGRITY sulla richiesta di una erogazione

- * “Ridefinito” con keystore di tipo “PDND”: consente di attuare una configurazione dinamica in cui GovWay effettuerà lo scaricamento della chiave pubblica dalla PDND attraverso le [API PDND](#) se l’identificativo kid della chiave non è già presente nella cache locale ([Fig. 3.78](#)).
- Time to Live: consente di ridefinire l’intervallo temporale, in secondi, per il quale i token creati precedentemente all’intervallo indicato verranno rifiutati.
- Audience: consente di indicare il valore del claim “aud” atteso nel token di sicurezza INTEGRITY. Se non viene fornito un valore, il valore atteso all’interno del security token corrisponderà all’url di invocazione.
- Coesistenza Token Authorization e Agid-JWT-Signature: consente di indicare da quale header estrarre l’identificativo “jti” da associare alla traccia come “ID del Messaggio” (default: Agid-JWT-Signature). Per

ModI - Richiesta

Sicurezza Messaggio

TrustStore Certificati	Ridefinito
Time to Live	Default
Audience	

Se non viene fornito un valore, il valore atteso all'interno del security token corrisponderà all'url di invocazione

▼ **Coesistenza Token Authorization e Agid-JWT-Signature**

TrustStore Certificati

Tipo	PDND
------	------

Figure3.78: Truststore “dinamico” per la validazione del token INTEGRITY sulla richiesta di una erogazione

maggiori dettagli si rimanda alla sezione “*Configurazione coesistenza degli header in una Erogazione*”.

- ModI - Risposta (Fig. 3.79): la maschera prevede la configurazione del meccanismo di firma digitale del messaggio di risposta, e la produzione del relativo token di sicurezza, da inviare all'applicativo mittente:
 - Algoritmo: l'algoritmo che si vuole utilizzare per la firma digitale del messaggio di risposta;
 - Codifica Digest: consente di selezionare l'algoritmo utilizzato per produrre il digest;
 - HTTP Headers da firmare: indicazione degli eventuali Header HTTP da firmare;
 - Keystore: il keystore da utilizzare per la firma del messaggio di risposta. È possibile mantenere il riferimento al keystore di default, fornito in fase di installazione del prodotto, oppure indicare un diverso riferimento (opzione «Ridefinito») fornendo il path sul filesystem, o in alternativa direttamente l'archivio, unitamente a Tipo, Password, Alias Chiave Privata e Password Chiave Privata.
 - Time to Live (secondi): validità temporale del token prodotto.
 - Claims: consente di personalizzare i claims presenti all'interno del token prodotto. Per maggiori dettagli si rimanda alla sezione “*Payload Claims del token JWT*”.

INTEGRITY_SOAP_01 / INTEGRITY_REST_01 - Integrità payload del messaggio

Nota: La sigla che identifica il pattern di sicurezza messaggio varia a seconda se l'API sia di tipo REST, per cui la sigla corrisponde a *INTEGRITY_REST_01*, o SOAP dove viene utilizzata la sigla *INTEGRITY_SOAP_01*.

Questo pattern di sicurezza consente di estendere «ID_AUTH_REST_01» o «ID_AUTH_REST_02» aggiungendo un meccanismo che garantisce l'integrità del payload del messaggio.

API

ModI - Risposta

Sicurezza Messaggio

Algoritmo	RS256
Codifica Digest	Base64
HTTP Headers da firmare *	Digest <input checked="" type="checkbox"/> Content-Type <input checked="" type="checkbox"/> Content-Encoding <input checked="" type="checkbox"/>
KeyStore	Default
Time to Live (secondi) *	300
Indica la validità temporale, in secondi, a partire dalla data di creazione del security token della risposta	
Claims	<input type="text"/> (i)
Indicare per riga i claims (nome=valore); visualizzare 'info' per maggiori dettagli	

Figure3.79: Dati per la configurazione della sicurezza messaggio sulla risposta di una erogazione

L'attivazione di questo pattern avviene a livello della relativa API, nella sezione «ModI», elemento «Sicurezza Messaggio» come indicato rispettivamente per una API REST in Fig. 3.80 e per una API SOAP in Fig. 3.81:

- selezionare il pattern «INTEGRITY_REST_01 con ID_AUTH_REST_01» nel caso si voglia estendere «ID_AUTH_REST_01», oppure il pattern «INTEGRITY_REST_01 con ID_AUTH_REST_02» nel caso si voglia estendere «ID_AUTH_REST_02» con il meccanismo di garanzia dell'integrità del payload (Fig. 3.80).
- selezionare una “Generazione Token” di tipo “Authorization ModI” per far sì che il Token “ID_AUTH” sia generato dalla parte mittente.

Le voci “Header HTTP del Token” (presente solamente su API di tipo REST) e “Applicabilità” consentono di personalizzare gli header HTTP utilizzati (nomi e coesistenza) e di indicare se il pattern di sicurezza verrà attuato sia sulla richiesta che sulla risposta. Su API di tipo SOAP è possibile selezionare una “Applicabilità” che firmi oltre al body anche gli attachments, se presenti. Maggiori informazioni vengono fornite rispettivamente nelle sezioni “*Header HTTP del token JWT*” e “*Attivazione della sicurezza messaggio su richiesta/risposta*”.

Di seguito vengono forniti i dettagli di configurazione aggiuntivi, rispetto ai passi descritti nella sezione “*ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509*”, per gli scenari di fruizione o erogazione di un servizio.

Modi

Sicurezza Canale

Pattern: ID_AUTH_CHANNEL_01

Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern: INTEGRITY_REST_01 con ID_AUTH_REST_01

Integrità payload del messaggio

Generazione Token: Authorization ModI

Token ID_AUTH generato dal mittente secondo le Linee Guida 'ModI'

Header HTTP del Token: Agid-JWT-Signature + Authorization Bearer

Applicabilità: Richiesta e Risposta

Digest Richiesta: Non ripudiabilità della trasmissione (i)

Informazioni Audit: Dati del dominio del fruitore

Figure3.80: Pattern di sicurezza messaggio «INTEGRITY_REST_01» per un API REST

Modi

Sicurezza Canale

Pattern: ID_AUTH_CHANNEL_01

Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern: INTEGRITY_SOAP_01 con ID_AUTH_SOAP_01

Integrità payload del messaggio

Generazione Token: Authorization ModI

Token ID_AUTH generato dal mittente secondo le Linee Guida 'ModI'

Applicabilità: Richiesta e Risposta

Digest Richiesta: Non ripudiabilità della trasmissione (i)

Informazioni Audit: Dati del dominio del fruitore

Figure3.81: Pattern di sicurezza messaggio «INTEGRITY_SOAP_01» per un API SOAP

Fruizione INTEGRITY_REST_01 / INTEGRITY_SOAP_01 (X509)

Le richieste che provengono dagli applicativi interni del dominio e sono dirette verso altre amministrazioni verranno arricchite del token di sicurezza “ModI” previsto dall’operazione invocata, come indicato precedentemente nella sezione [INTEGRITY_SOAP_01 / INTEGRITY_REST_01 - Integrità payload del messaggio](#).

Per la configurazione delle fruizioni con i pattern di sicurezza messaggio è necessario registrare ciascun applicativo interno coinvolto al fine principale di associargli una chiave privata e un certificato X509 che GovWay utilizza per firmare il token di sicurezza “ModI” prodotto. Gli applicativi vengono identificati da GovWay tramite una delle modalità di autenticazione supportate descritte nella sezione [Autenticazione Trasporto](#) (Fig. 3.50).

Nella figura “Fig. 3.82” viene raffigurato lo scenario di fruizione in cui il trust avviene tra fruitore ed erogatore tramite certificati x509.

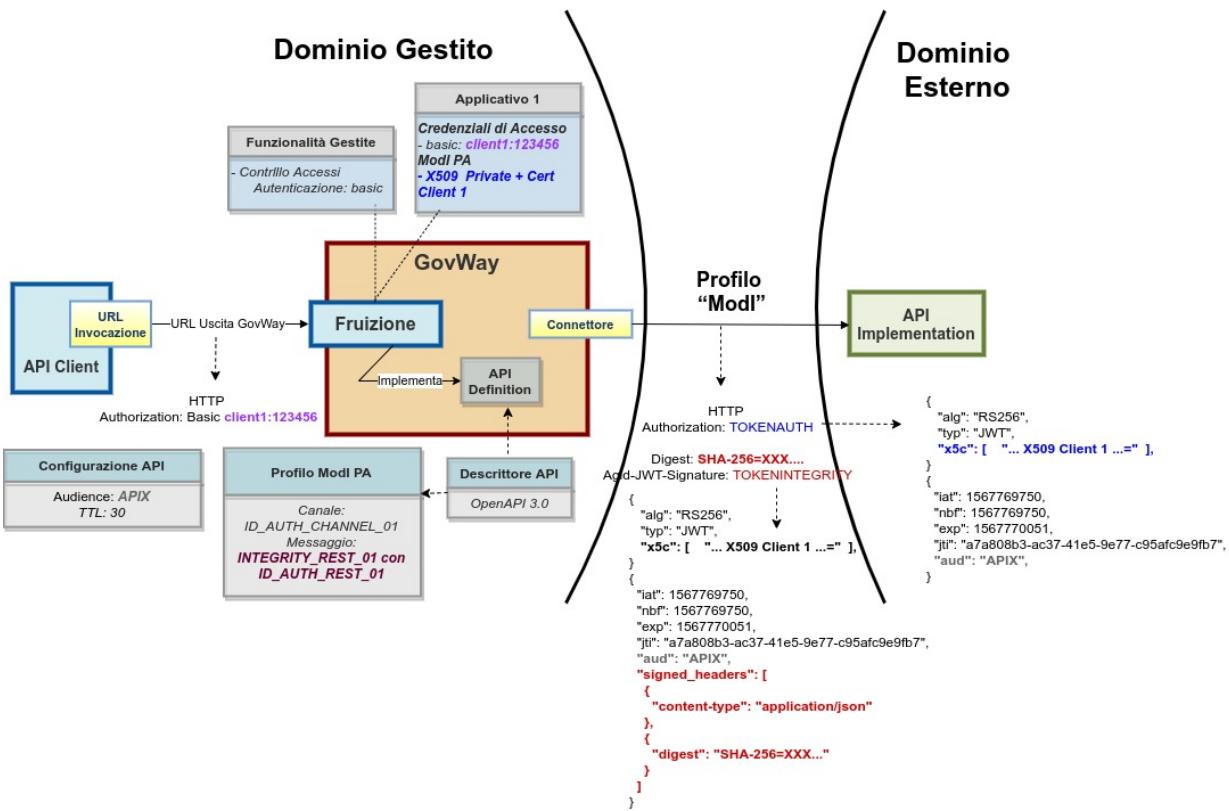


Figure3.82: Fruizione con Profilo di Interoperabilità “ModI”, pattern “INTEGRITY_REST_01”: trust tra fruitore ed erogatore tramite certificati x509

Di seguito vengono forniti i dettagli di configurazione aggiuntivi o differenti, rispetto ai passi descritti nella sezione [“Fruizione ID_AUTH_REST_01 / ID_AUTH_SOAP_01 \(X509\)”](#).

API

La registrazione della API deve essere effettuata seguendo le indicazioni descritte nella sezione [INTEGRITY_SOAP_01 / INTEGRITY_REST_01 - Integrità payload del messaggio](#)

Fruizione

Nel contesto della configurazione di una fruizione di una API di tipo REST, relativamente alla sezione «ModI - Richiesta», oltre ai dati da fornire per la produzione della firma digitale deve essere aggiunta anche l’indicazione degli eventuali Header HTTP da firmare. Tale indicazione viene fornita con il campo «HTTP Headers da firmare» (Fig. 3.83).

Modi - Richiesta

Sicurezza Messaggio

Algoritmo: RS256

HTTP Headers da firmare *: Digest, Content-Type, Content-Encoding

Riferimento X.509: x5c (Certificate), x5t#256 (Certificate SHA-256 Thumbprint), x5u (URL)

Certificate Chain:

Time to Live (secondi) *: 300

Audience: http://ente/RestBlockingIntegrity

Figure3.83: Fruizione «INTEGRITY_REST_01» - Configurazione richiesta con indicazione HTTP Headers da firmare

Erogazione INTEGRITY_REST_01 / INTEGRITY_SOAP_01 (X509)

In un’erogazione di una API le richieste provengono da amministrazioni esterne al dominio e sono dirette ad applicativi interni. Prima di procedere con l’inoltro della richiesta verso il backend interno, GovWay valida il token di sicurezza ricevuto rispetto al pattern associato all’operazione invocata: verifica firma, validazione temporale, filtro duplicati, verifica integrità del messaggio, verifica del token di audit etc.

Nella figura “Fig. 3.84” viene raffigurato lo scenario di erogazione in cui il trust avviene tra fruitore ed erogatore tramite certificati x509.

Di seguito vengono forniti i dettagli di configurazione aggiuntivi o differenti, rispetto ai passi descritti nella sezione “*Erogazione ID_AUTH_REST_01 / ID_AUTH_SOAP_01 (X509)*”.

API

La registrazione della API deve essere effettuata seguendo le indicazioni descritte nella sezione *INTEGRITY_SOAP_01 / INTEGRITY_REST_01 - Integrità payload del messaggio*

Erogazione

Nel contesto della configurazione di una erogazione di una API di tipo REST, relativamente alla sezione «Modi - Risposta», oltre ai dati da fornire per la produzione della firma digitale deve essere aggiunta anche l’indicazione degli eventuali Header HTTP da firmare. Tale indicazione viene fornita con il campo «HTTP Headers da firmare» (Fig. 3.85).

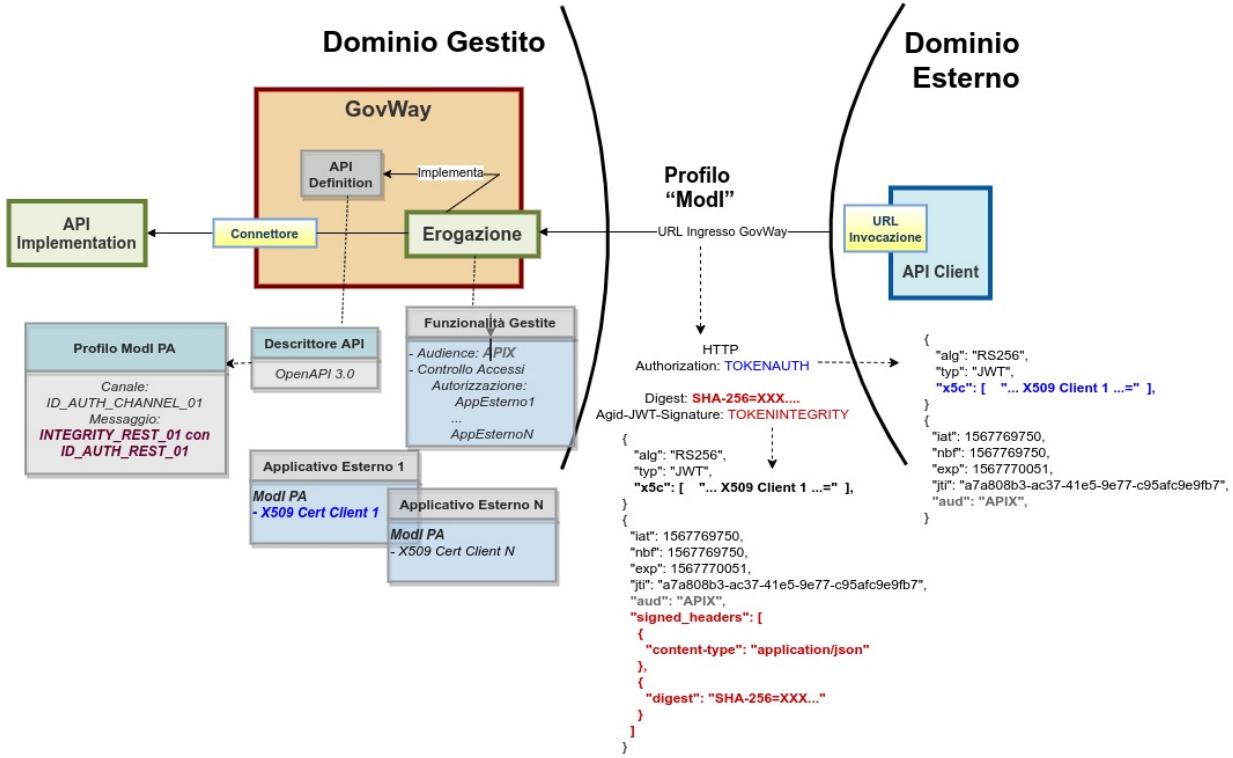


Figure3.84: Erogazione con Profilo di Interoperabilità “ModI”, pattern “INTEGRITY_REST_01”: trust tra fruitore ed erogatore tramite certificati x509

Modi - Risposta

Sicurezza Messaggio

Algoritmo: RS256

HTTP Headers da firmare *: Digest, Content-Type, Content-Encoding

Riferimento X.509: Utilizza impostazioni della Richiesta

Certificate Chain:

KeyStore: Default

Time to Live (secondi) *: 300

Indica la validità temporale, in secondi, a partire dalla data di creazione del security token della risposta

Figure3.85: Erogazione «INTEGRITY_REST_01» - Configurazione risposta con indicazione HTTP Headers da firmare

ID_AUTH tramite PDND + INTEGRITY_SOAP_01 / INTEGRITY_REST_01

Il token di autenticazione ID_AUTH_REST_01 descritto nella sezione “[ID_AUTH_REST_01 tramite la Piattaforma Digitale Nazionale Dati \(PDND\)](#)” è utilizzabile in combinazione con il token di integrità descritto nella sezione “[INTEGRITY_SOAP_01 / INTEGRITY_REST_01 - Integrità payload del messaggio](#)” sia su API di tipo REST che di tipo SOAP.

Per attuare la configurazione su API di tipo REST deve essere utilizzato un pattern di sicurezza “INTEGRITY_REST_01” dove la voce “Header HTTP del Token” deve essere valorizzata solamente con l’header “Agid-JWT-Signature”.

Nessuna particolare indicazione è invece necessaria per attuare la configurazione su API di tipo SOAP dove è sufficiente utilizzare il pattern di sicurezza “INTEGRITY_SOAP_01”.

Su entrambi gli scenari l’autenticazione dell’applicativo chiamante avverrà tramite il token ID_AUTH_REST_01 generato dalla PDND e veicolato su header HTTP “Authorization Bearer”. Invece la gestione dell’integrità del messaggio avverrà secondo le modalità descritte nella sezione “[INTEGRITY_SOAP_01 / INTEGRITY_REST_01 - Integrità payload del messaggio](#)”.

3.3.5 PROFILE_NON_REPUTATION_01

Non ripudiabilità della trasmissione

Questa funzionalità consente di estendere il pattern “[INTEGRITY_SOAP_01 / INTEGRITY_REST_01 / INTEGRITY_REST_02](#)” aggiungendo all’interno del token di sicurezza della risposta il digest della richiesta.

La funzionalità consente di implementare la soluzione per la non ripudiabilità della trasmissione come suggerito nelle linee guida di interoperabilità (Fig. 3.86) all’interno del documento “03 Profili di interoperabilità.pdf”.

D: Risposta

L’erogatore costruisce un messaggio di conferma includendo un identificativo che permetta di associare univocamente al messaggio di richiesta (ad esempio il digest presente nel messaggio di richiesta) e l’istante di trasmissione.

Figure3.86: Punto “D” della soluzione di sicurezza per la non ripudiabilità della trasmissione

Nota: La sigla che identifica il pattern di sicurezza messaggio varia a seconda se l’API sia di tipo REST, per cui la sigla corrisponde a INTEGRITY_REST_01, o SOAP dove viene utilizzata la sigla INTEGRITY_SOAP_01.

L’attivazione di questo profilo avviene a livello della relativa API, nella sezione «ModI», elemento «Sicurezza Messaggio», selezionando la voce «Digest Richiesta» (Fig. 3.87).

Nota: Poichè la funzionalità è un’estensione del pattern “[INTEGRITY_SOAP_01 / INTEGRITY_REST_01 / INTEGRITY_REST_02](#)”, la voce “Digest Richiesta” compare solamente se è stato selezionato uno dei pattern “[INTEGRITY_*](#)”

Nota: Nel caso venga disabilitata la generazione della sicurezza messaggio sulla richiesta o sulla risposta, la funzionalità “Digest della Richiesta” non sarà più attivabile.

Nella figura Fig. 3.88 viene riportato un esempio del payload, relativo al token di sicurezza ModI della risposta per una API REST, contenente il digest della richiesta.

Modi

Sicurezza Canale

Pattern: ID_AUTH_CHANNEL_01

Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern: INTEGRITY_REST_02 con ID_AUTH_REST_01

Integrità payload del messaggio

Generazione Token: Authorization PDND

Token ID_AUTH negoziato con la PDND

Header HTTP del Token: Agid-JWT-Signature + Authorization Bearer

Applicabilità: Richiesta e Risposta

Digest Richiesta: Non ripudiabilità della trasmissione | *i*

Informazioni Audit: Dati del dominio del fruitore

Figure3.87: Pattern di sicurezza messaggio «INTEGRITY» + Digest Richiesta

PAYLOAD: DATA

```
{
  "iat": 1592905216,
  "nbf": 1592905216,
  "exp": 1592905276,
  "jti": "39f616f1-1bb5-47f4-9d14-db1b130e0a35",
  "aud": "AvvocaturaStato/App2",
  "client_id": "Allegati/v1",
  "request_digest": "SHA-256=bd9e1f64cbc5b602eee10dd2202c6cf3cf9bdcfac8305756c79d13cb523048b3",
  "iss": "AgenziaEntrate",
  "sub": "Allegati/v1",
  "signed_headers": [
    {
      "digest": "SHA-256=2d784a4770350388efa147054fe223d1420ede681d46e8d6956c977d897b45b9"
    },
    {
      "content-type": "application/json"
    }
  ]
}
```

Figure3.88: Payload del Token di Sicurezza REST con pattern «INTEGRITY_REST» + Digest Richiesta

Nella figura Fig. 3.89 viene riportato un esempio relativo al token di sicurezza ModI della risposta per una API SOAP. Tutti i digest degli elementi firmati nella richiesta vengono riportati all'interno di un header soap “X-Digest-Richiesta” della risposta. Il nuovo header “X-Digest-Richiesta” sarà aggiunto agli elementi firmati nella risposta.

```
<wsu:Created>2020-06-23T09:51:40.499Z</wsu:Created>
<wsu:Expires>2020-06-23T09:51:40.499Z</wsu:Expires>
</wsu:Timestamp>
</wsse:Security>
<ns2:X-RequestDigest xmlns:ns2="http://amministrazioneesempio.it/nomeinterfacciaservizio"
xmlns:env="http://www.w3.org/2003/05/soap-envelope" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd" env:mustUnderstand="false" wsu:Id="id-fc810b92-431a-4f5f-a917-df88f173472d">
<ds:Reference xmlns:ds="http://www.w3.org/2000/09/xmldsig#" URI="#TS-1d254b27-62ab-4798-bcd7-1b636a7af6f6">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
<ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="wsse soap"/>
</ds:Transform>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
<ds:DigestValue>mNF7nyQtYMEh9r28c5I0dHt+G6x0sna1BG8Nl+KKxw=</ds:DigestValue>
</ds:Reference>
<ds:Reference xmlns:ds="http://www.w3.org/2000/09/xmldsig#" URI="#id-59659b84-bbe8-401c-bba0-25c731086b4b">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
<ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="soap"/>
</ds:Transform>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
<ds:DigestValue>Bz5zFFx1zesgu1nBsug0dk+URQTOKLe5Is+Uj8Fap4=</ds:DigestValue>
</ds:Reference>
<ds:Reference xmlns:ds="http://www.w3.org/2000/09/xmldsig#" URI="#id-8bc48ae-9c13-40ae-b888-8e3a6131b71c">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
<ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="soap"/>
</ds:Transform>
</ds:Transforms>
```

Figure3.89: Payload del Token di Sicurezza SOAP con pattern «INTEGRITY_SOAP» + Digest Richiesta

3.3.6 AUDIT_REST_01 / AUDIT_REST_02

Il pattern AUDIT_REST nelle sue varie declinazioni consente all'erogatore di identificare la specifica provenienza di ogni singola richiesta di accesso ai dati effettuata dal fruitore.

Le Linee Guida indicano che l'erogatore e il fruitore devono individuare i claim da includere nel JWT di audit e suggeriscono i seguenti dati che dovranno essere presenti nel token generato dal fruitore, per ogni richiesta effettuata:

- userID, un identificativo univoco dell'utente interno al dominio del fruitore che ha determinato l'esigenza della request di accesso all'e-service dell'erogatore;
- userLocation, un identificativo univoco della postazione interna al dominio del fruitore da cui è avviata l'esigenza della request di accesso all'e-service dell'erogatore;
- LoA, livello di sicurezza o di garanzia adottato nel processo di autenticazione informatica nel dominio del fruitore.

Le Linee Guida definiscono 2 pattern utilizzabili sia per API REST che per API SOAP, che si aggiungono al precedente pattern legacy di GovWay:

- *AUDIT_REST_01 - Inoltro dati tracciati nel dominio del Fruitore*: definisce la struttura del token di audit utilizzabile tramite due modalità:
 - un criterio di trust realizzato tramite il materiale crittografico depositato sulla PDND;
 - un trust diretto fruitore-erogatore attraverso l'utilizzo di certificati X509.
- *AUDIT_REST_02 - Inoltro dati tracciati nel dominio del Fruitore con correlazione*: pattern che estende il precedente aggiungendo la correlazione tra il token di autenticazione e il token di audit. Il pattern richiede un trust realizzato tramite il materiale crittografico depositato sulla PDND.
- *AUDIT_LEGACY*: soluzione legacy di GovWay già presente nelle precedenti versioni.

AUDIT_REST_01 - Inoltro dati tracciati nel dominio del Fruitore

Questa funzionalità consente di estendere uno qualunque dei pattern di sicurezza “*ID_AUTH_SOAP_01 / ID_AUTH_REST_01*”, “*ID_AUTH_SOAP_02 / ID_AUTH_REST_02*” e “*INTEGRITY_SOAP_01 / INTEGRITY_REST_01 / INTEGRITY_REST_02*” attraverso un nuovo token aggiuntivo adibito a contenere informazioni utili all’erogatore a identificare la specifica provenienza di ogni singola richiesta di accesso ai dati effettuata dal fruitore come descritto nella sezione “*AUDIT_REST_01 / AUDIT_REST_02*”.

L’attivazione di questa funzionalità avviene a livello della relativa API, nella sezione «ModI», elemento «Sicurezza Messaggio», selezionando la voce «Informazioni Audit» (Fig. 3.90).

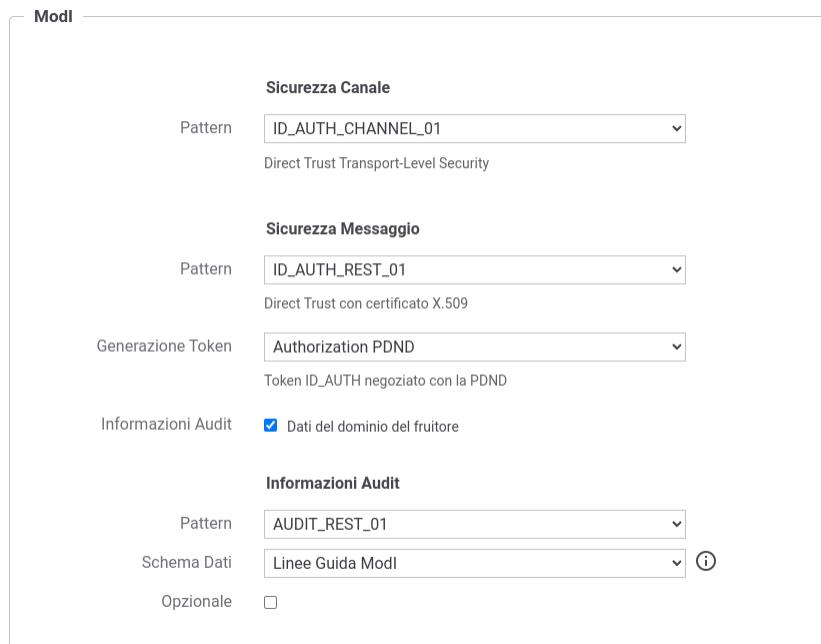


Figure3.90: Pattern di sicurezza messaggio «*ID_AUTH*» + Informazioni Audit «*AUDIT_REST_01*» su API REST

Nota: Nel caso venga disabilitata la generazione della sicurezza messaggio sulla richiesta, la funzionalità “Informazioni Audit” non sarà più attivabile.

Il pattern è utilizzabile anche su API SOAP come mostrato nella figura Fig. 3.91.

Il token di audit per default è considerato obbligatorio, ed è possibile renderlo opzionale attivando il campo con nome “Opzionale” presente nella sezione “Informazioni Audit”.

La voce “Generazione Token” consente di definire il criterio di trust:

- “Authorization PDND” (o “Authorization OAuth”): un criterio di trust realizzato tramite il materiale crittografico depositato sulla PDND;
- “Authorization ModI”: un trust diretto fruitore-erogatore attraverso l’utilizzo di certificati X509.

Nella sezione “*Informazioni UserID, UserLocation e LoA incluse nel token di AUDIT_REST_01*” vengono descritti l’insieme dei dati di default configurati built-in nel prodotto (UserID, UserLocation e LoA) mentre nella sezione “*Informazioni personalizzate da includere nel token di AUDIT_REST_01*” vengono fornite le informazioni utili a definire un insieme di claim alternativo a quello di default.

The screenshot shows the 'ModI' configuration interface with the following settings:

- Sicurezza Canale**:
 - Pattern: ID_AUTH_CHANNEL_01
 - Description: Direct Trust Transport-Level Security
- Sicurezza Messaggio**:
 - Pattern: ID_AUTH_SOAP_01
 - Description: Direct Trust con certificato X.509
- Generazione Token**:
 - Pattern: Authorization PDND
 - Description: Token ID_AUTH negoziato con la PDND
- Informazioni Audit**:
 - Pattern: AUDIT_REST_01
 - Schema Dati: Linee Guida ModI
 - Opzionale:

Figure3.91: Pattern di sicurezza messaggio «ID_AUTH» + Informazioni Audit «AUDIT_REST_01» su API SOAP

Nelle sezioni successive vengono forniti i dettagli di configurazione necessari ad utilizzare il pattern di AUDIT negli scenari di fruizione o erogazione di un servizio.

Informazioni UserID, UserLocation e LoA incluse nel token di AUDIT_REST_01

Il pattern AUDIT_REST nelle sue varie declinazioni consente all'erogatore di identificare la specifica provenienza di ogni singola richiesta di accesso ai dati effettuata dal fruitore.

Le Linee Guida indicano che l'erogatore e il fruitore devono individuare i claim da includere nel JWT di audit “Agid-JWT-TrackingEvidence” e suggeriscono i seguenti dati che dovranno essere presenti nel token generato dal fruitore, per ogni richiesta effettuata:

- userID, un identificativo univoco dell'utente interno al dominio del fruitore che ha determinato l'esigenza della request di accesso all'e-service dell'erogatore;
- userLocation, un identificativo univoco della postazione interna al dominio del fruitore da cui è avviata l'esigenza della request di accesso all'e-service dell'erogatore;
- LoA, livello di sicurezza o di garanzia adottato nel processo di autenticazione informatica nel dominio del fruitore.

Nella figura Fig. 3.92 viene riportato un esempio del payload relativo al token di sicurezza “ModI” di una API REST, contenente le informazioni aggiuntive sull'utente che ha effettuato la richiesta.

Nella configurazione built-in del prodotto, le informazioni da inserire nel token di audit vengono richieste all'applicativo fruitore che invoca una fruizione tramite header http o parametri della url come descritto nella sezione “*Fruizione AUDIT_REST_01*”.

Le informazioni di audit vengono aggiunte, per default, alla traccia ModI e un'erogazione le inoltre al backend attraverso header http come descritto nella sezione “*Erogazione AUDIT_REST_01*”.

```
PAYOUT: DATA
{
  "nbf": 1685625936,
  "exp": 1685626236,
  "jti": "95725a5c-007f-11ee-9ad7-024204bd5e1f",
  "aud": "RestBlockingAuditRest01-JWK/v1",
  "userID": "mariorossi",
  "userLocation": "ufficio234",
  "LoA": "spid",
  "iss": "955dc0eb-007f-11ee-9ad7-024204bd5e1f"
}
```

Figure3.92: Payload del Token di Audit con pattern «AUDIT_REST_01»

Nella sezione “*Informazioni personalizzate da includere nel token di AUDIT_REST_01*” vengono fornite le informazioni utili sia a modificare le configurazioni dei claim userID, userLocation, LoA che a definire un insieme di claim alternativo a quello di default.

Informazioni personalizzate da includere nel token di AUDIT_REST_01

L’insieme di informazioni proposte dalle Linee Guida e le modalità di identificazione dei valori da associare a tali informazioni, all’interno del token di audit, vengono definite tramite la configurazione presente nel file <directory-lavoro>/modipa_local.properties.

La configurazione consente inoltre di definire insiemi alternativi a quello di default proposto dalle Linee Guida.

Ad ogni insieme di dati, concordati tra fruitore ed erogatore, deve essere associato un identificativo univoco interno al file di proprietà (IDPROP) da aggiungere all’elenco, separato da virgola, presente nella proprietà “org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern” che inizialmente presenta solamente lo schema “default” che definisce le informazioni suggerite dalle Linee Guida (userID, userLocation e LoA):

```
# Ogni insieme di dati concordati tra fruitore ed erogatore viene identificato,
→da una keyword da aggiungere alla seguente proprietà (elenco separato da
→virgola):
org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern=default,
→<IDPROP>
```

Di seguito vengono descritte le configurazioni attuabili su un insieme di dati da inserire nel token di audit.

Le proprietà seguenti associano un identificativo univoco e una label all’insieme di dati, label che verrà proposta nella maschera di configurazione delle informazioni di audit in una API (Fig. 3.93).

```
org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.
→nome=Schema-Dati-Esempio
org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.
→label=Schema Dati di Esempio
```

Tramite la proprietà “org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims” devono essere elencati i claim concordati tra erogatore e fruitore. Ogni claim viene identificato da un identificativo univoco interno al file di proprietà (IDCLAIM-X) per l’insieme di dati (IDPROP).

```
org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.
→claims=<IDCLAIM-1>, ..., <IDCLAIM-N>
```

Ogni singolo claim è personalizzabile nei seguenti aspetti:

Modi

Sicurezza Canale

Pattern: ID_AUTH_CHANNEL_01

Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern: ID_AUTH_REST_01

Direct Trust con certificato X.509

Generazione Token: Authorization PDND

Token ID_AUTH negoziato con la PDND

Informazioni Audit: Dati del dominio del fruitore

Informazioni Audit

Pattern: AUDIT_REST_01

Schema Dati: Schema Dati di Esempio (i)

Opzionale:

Figure3.93: Informazioni Audit personalizzato

- Un identificativo univoco e una label da associare al claim rispettivamente tramite le proprietà “`org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.nome`” e “`org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.label`”.
- L’indicazione se il claim è obbligatorio all’interno del token di audit tramite la proprietà “`org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.required`”.
- L’indicazione se il claim veicola un valore riutilizzabile su differenti chiamate tramite la proprietà “`org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.cacheable`”. In caso di proprietà non definita per default il valore del claim sarà processato come riutilizzabile.

Nota: L’intero token di audit verrà salvato in cache e riutilizzato su differenti chiamate solo se tutti i claim inseriti all’interno del token risultano configurati come riutilizzabili.

- L’indicazione se il claim è una stringa json o un tipo primitivo attraverso la proprietà “`org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.stringType`”.
- Una descrizione sintetica dell’informazione rappresentata dal claim che verrà fornita tra i criteri informativi dello schema nella maschera di configurazione dell’API (proprietà “`org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.info`”).
- Una lista di regole, separate dalla virgola, che verranno utilizzate in ordine per individuare il valore del claim da inserire all’interno del token. Le regole possono essere definite tramite valori statici o contenere informazioni dinamiche risolte a runtime e descritte nella sezione “*Valori dinamici*” (proprietà “`org.openscoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.rules`”).

X>.rule”).

- Per ogni regola definita nella precedente proprietà deve essere fornita una descrizione sintetica che verrà visualizzata tra i criteri informativi nella maschera di configurazione della fruizione (proprietà “*org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.rule.info*”).
- L’informazione di un claim di audit può essere propagato verso il backend di una erogazione se viene definito il nome di un header http tramite la proprietà “*org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.forwardBackend*”. Per non propagare alcun header associare un valore vuoto alla proprietà.
- Il valore di un claim di audit viene aggiunto alla traccia ModI se abilito nella proprietà “*org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.trace*”).
- Opzionalmente è inoltre possibile definire per ogni claim le seguenti proprietà che definiscono dei criteri di validazione del valore:
 - espressione regolare (proprietà “*org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.regexp*”)
 - lista di valori ammessi (proprietà “*org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.enum*”)
 - lunghezza minima di caratteri (proprietà “*org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.minLength*”)
 - lunghezza massima di caratteri (proprietà “*org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.<IDPROP>.claims.<IDCLAIM-X>.maxLength*”)

Di seguito viene fornito un esempio di configurazione in cui i valori riportati sono quelli utilizzati per la definizione del claim “*userID*” descritto dalle Linee Guida.

```
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.nome=userID
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.label=UserID
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.required=true
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.stringType=true
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.info=Identificativo univoco dell'utente interno al
  ↳dominio del fruitore che ha determinato l'esigenza della richiesta di
  ↳accesso all'e-service dell'erogatore
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.rule=${header:GovWay-Audit-User}, ${query:govway_audit_
  ↳user}
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.rule.info=Header http 'GovWay-Audit-User', Parametro della
  ↳url 'govway_audit_user'
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.forwardBackend=GovWay-Audit-UserID
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.trace=true
```

Un altro esempio mostra l’utilizzo dei criteri di validazione per definire un claim il cui valore deve essere composto solamente da lettere e numeri e formato esattamente da 3 caratteri.

```

org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.nome=esempioValidazione
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.label=EsempioValidazione
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.required=true
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.stringType=true
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.regexp^ [A-Za-z0-9]+$
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.minLength=3
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.maxLength=3
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.info=Un esempio di validazione tramite regexp e min/max
  ↳length
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.rule=${header:GovWay-Audit-Esempio}
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.rule.info=Header http 'GovWay-Audit-Esempio'
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.forwardBackend=GovWay-Audit-Esempio
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.trace=true

```

Infine l'ultimo esempio mostra l'utilizzo dei criteri di validazione per definire un claim i cui valori vengono definiti da una enumeration.

```

org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.nome=esempioValidazioneByEnum
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.label=EsempioValidazioneByEnum
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.required=true
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.stringType=true
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.enum=CREATE,UPDATE,DELETE
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.info=Un esempio di validazione tramite enum
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.rule=${header:GovWay-Audit-Esempio}
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.rule.info=Header http 'GovWay-Audit-Esempio'
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.forwardBackend=GovWay-Audit-Esempio
org.openspcoop2.protocol.modipa.sicurezzaMessaggio.audit.pattern.default.
  ↳claims.<IDCLAIM-X>.trace=true

```

Fruizione AUDIT_REST_01

Le richieste che provengono dagli applicativi interni del dominio e sono dirette verso altre amministrazioni verranno arricchite del token di sicurezza “Agid-JWT-TrackingEvidence” contenente le informazioni di audit.

Nella figura “Fig. 3.94” viene raffigurato lo scenario di fruizione in cui il trust avviene tramite la PDND e viene prodotto il token “Agid-JWT-TrackingEvidence” previsto dal pattern “AUDIT_REST_01”.

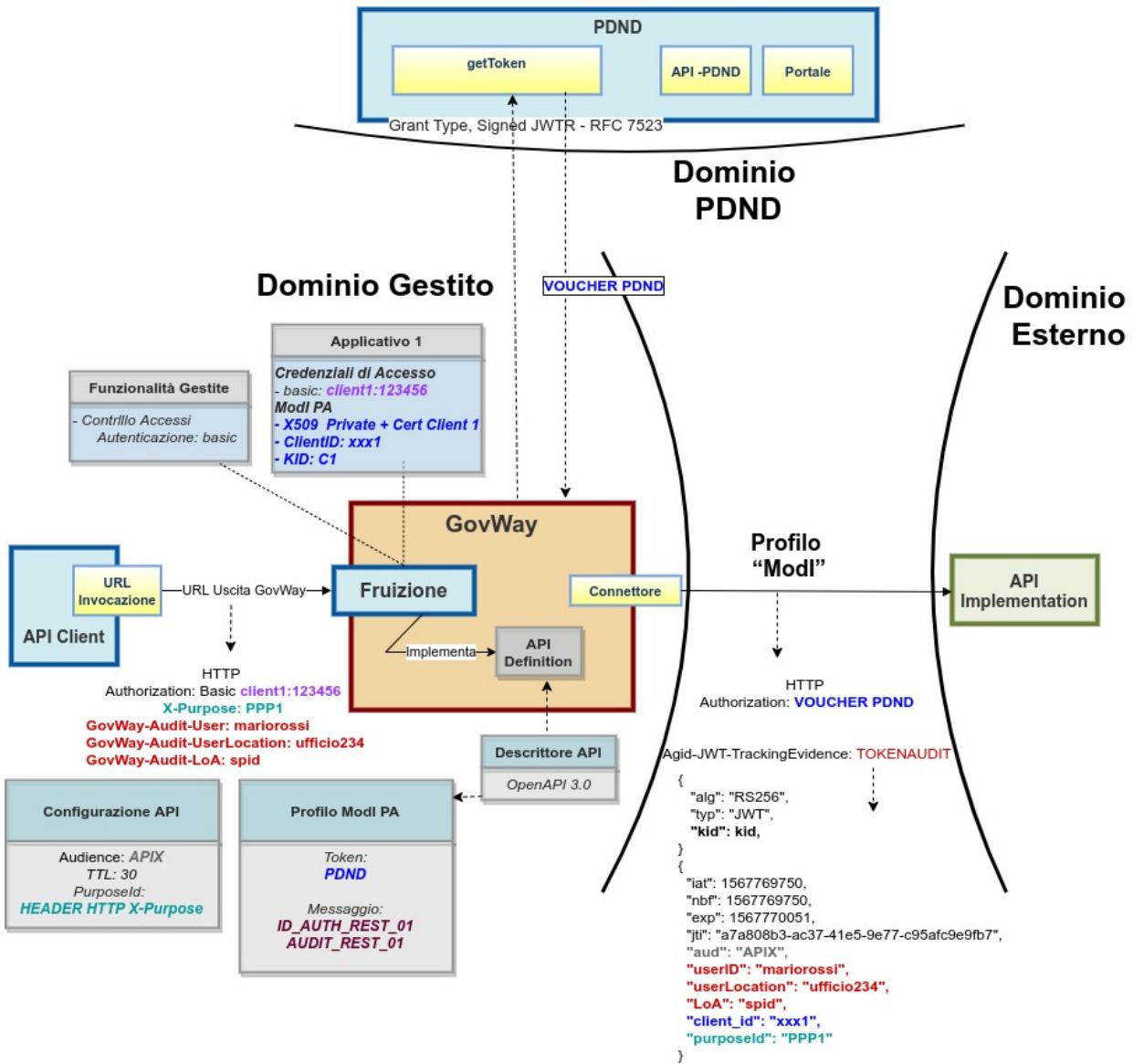


Figure3.94: Fruizione con Profilo di Interoperabilità “ModI”, pattern “AUDIT_REST_01”: trust tramite PDND

Come mostrato nella figura Fig. 3.94 le informazioni di audit sono per default attese nella richiesta pervenuta a GovWay sotto forma di header http o parametro della url:

- userID, un identificativo univoco dell'utente deve essere indicato nella richiesta di fruizione all'interno dell'header http “GovWay-Audit-User” o nel parametro della url con nome “govway_audit_user”;
- userLocation, un identificativo univoco della postazione dell'utente deve essere indicata nell'header http “GovWay-Audit-UserLocation” o nel parametro della url con nome “govway_audit_user_location”;

- LoA, livello di sicurezza adottato nel processo di autenticazione dell’utente può essere indicato nell’header http “GovWay-Audit-LoA” o nel parametro della url con nome “govway_audit_loa”. Questa informazione non è richiesta obbligatoriamente.

Il comportamento di default, per l’acquisizione dei valori utilizzati per le tre informazioni, può essere personalizzato accedendo nella sezione «ModI» di una fruizione, e modificando le voci «Informazioni di Audit» (Fig. 3.95) indicando un valore statico o utilizzando le proprietà dinamiche descritte nella sezione *Valori dinamici*.

Figure3.95: Fruizione - personalizzazione dell’acquisizione delle Informazioni UserID, UserLocation e LoA

Nella sezione «Informazioni di Audit» è inoltre possibile indicare se nel token “Agid-JWT-TrackingEvidence” il claim “aud” deve essere valorizzato con lo stesso audience utilizzato per gli altri token di sicurezza o se fornire un valore differente come mostrato nella figura Fig. 3.96

Figure3.96: Fruizione - personalizzazione dell’Audience all’interno del token “Agid-JWT-TrackingEvidence”

Erogazione AUDIT_REST_01

Nella figura “Fig. 3.97” viene raffigurato lo scenario di erogazione in cui le richieste provenienti dal dominio esterno contengono il token di sicurezza “Agid-JWT-TrackingEvidence” e il trust avviene tramite la PDND.

Per default, le informazioni di audit vengono aggiunte alla traccia ModI e inoltrate al backend attraverso gli header http “GovWay-Audit-UserID”, “GovWay-Audit-UserLocation” e “GovWay-Audit-LoA”. Nella sezione “*Informazioni personalizzate da includere nel token di AUDIT_REST_01*” è possibile personalizzare questi aspetti.

Nella sezione «Informazioni di Audit» è possibile indicare se nel token “Agid-JWT-TrackingEvidence” il claim “aud” deve essere atteso valorizzato con lo stesso audience utilizzato per gli altri token di sicurezza o con uno differente come mostrato nella figura Fig. 3.98

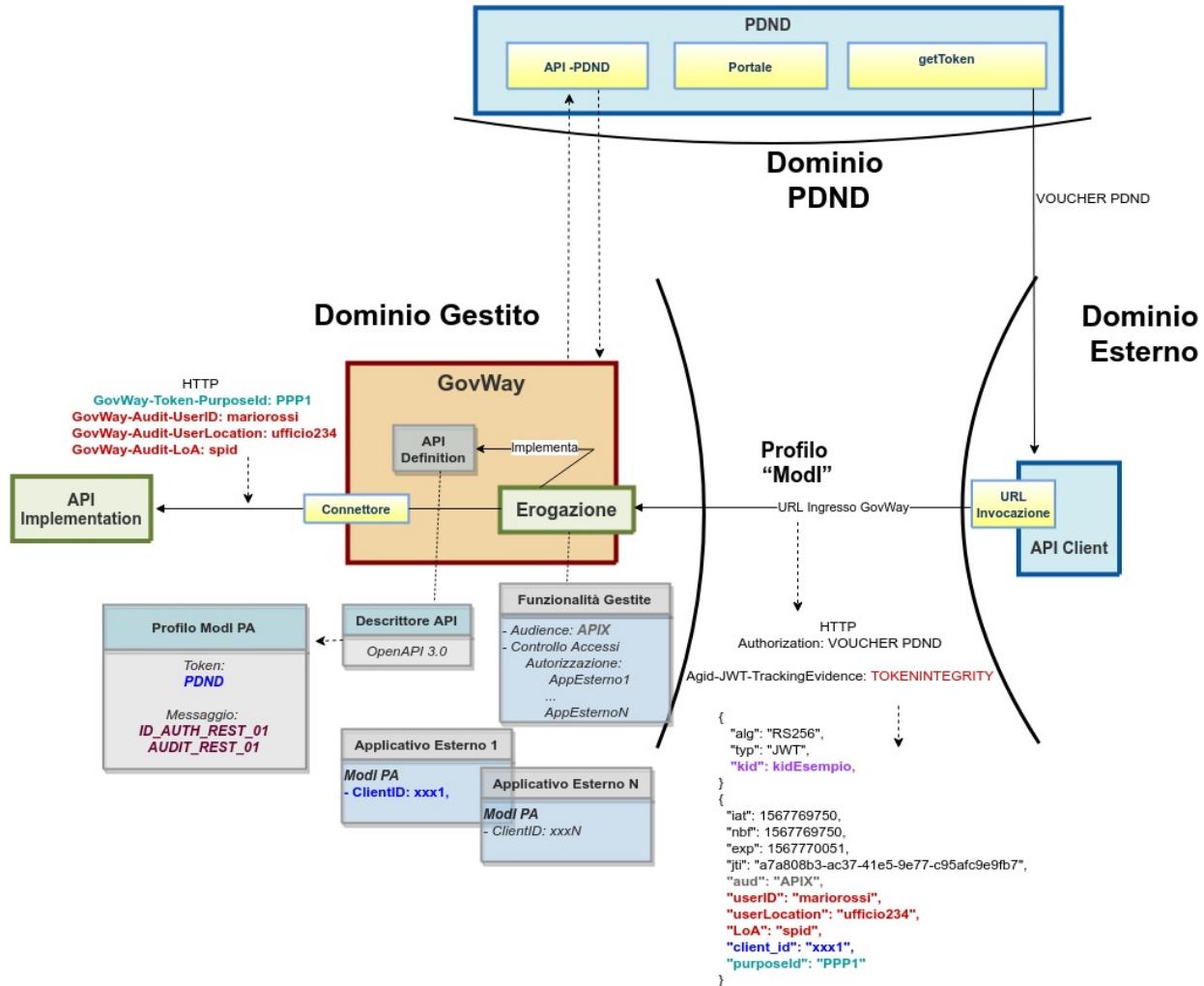


Figure3.97: Erogazione con Profilo di Interoperabilità “ModI”, pattern “AUDIT_REST_01”: trust tramite PDND

Informazioni Audit

Audience: Differente identificativo

*

Figure3.98: Erogazione - personalizzazione dell’Audience all’interno del token “Agid-JWT-TrackingEvidence”

AUDIT_REST_02 - Inoltro dati tracciati nel dominio del Fruitore con correlazione

Il pattern “AUDIT_REST_02” estende il pattern “AUDIT_REST_012” aggiungendo la correlazione tra il token di autenticazione e il token di audit. Il pattern richiede un trust realizzato tramite il materiale crittografico depositato sulla PDND.

Nella figura “Fig. 3.99” viene raffigurato lo scenario.

L’attivazione del pattern “AUDIT_REST_02” rispetto a quanto descritto nella sezione “[AUDIT_REST_01 - Inoltro dati tracciati nel dominio del Fruitore](#)” differisce solamente a livello API, nella sezione «ModI - Informazioni Audit» (Fig. 3.100).

AUDIT_LEGACY

Questa funzionalità consente di estendere il pattern “[INTEGRITY_SOAP_01 / INTEGRITY_REST_01 / INTEGRITY_REST_02](#)” aggiungendo all’interno del token di sicurezza le informazioni sull’utente che ha effettuato la richiesta.

Nota: La sigla che identifica il pattern di sicurezza messaggio varia a seconda se l’API sia di tipo REST, per cui la sigla corrisponde a *INTEGRITY_REST_01*, o SOAP dove viene utilizzata la sigla *INTEGRITY_SOAP_01*.

L’attivazione di questa funzionalità avviene a livello della relativa API, nella sezione «ModI», elemento «Sicurezza Messaggio», selezionando la voce «Informazioni Audit» (Fig. 3.101).

Nota: Poichè la funzionalità è un’estensione del pattern “[INTEGRITY_SOAP_01 / INTEGRITY_REST_01 / INTEGRITY_REST_02](#)”, il pattern “AUDIT_LEGACY” all’interno della sezione “Informazioni Audit” compare solamente se è stato selezionato uno dei pattern “*INTEGRITY_**”.

Nota: Nel caso venga disabilitata la generazione della sicurezza messaggio sulla richiesta, la funzionalità “Informazioni Audit” non sarà più attivabile.

Le informazioni aggiuntive presenti all’interno del token riguardano:

- UserID Utente: identificativo univoco dell’utente all’interno del dominio rappresentato dal “Codice Ente”;
- Indirizzo IP Utente: identifica la postazione da cui l’utente ha effettuato la richiesta;
- Codice Ente: dominio di appartenenza dell’utente.

Nella figura Fig. 3.102 viene riportato un esempio del payload relativo al token di sicurezza “ModI” di una API REST, contenente le informazioni aggiuntive sull’utente che ha effettuato la richiesta.

Nella figura Fig. 3.103 viene riportato un esempio relativo al token di sicurezza “ModI” per una API SOAP. Le informazioni aggiuntive sull’utente che ha effettuato la richiesta sono incluse in una Asserzione SAML.

Fruizione

In una fruizione, le informazioni aggiuntive che vengono aggiunte nel token, sono per default attese nella richiesta pervenuta a GovWay sotto forma di header http o parametro della url:

- UserID Utente: l’identificativo dell’utente deve essere indicato nella richiesta di fruizione all’interno dell’header http “GovWay-CS-User” o del parametro della url con nome “govway_cs_user”;
- Indirizzo IP Utente: la postazione dell’utente deve essere indicata nell’header http “GovWay-CS-IPUser” o del parametro della url con nome “govway_cs_ipuser”;

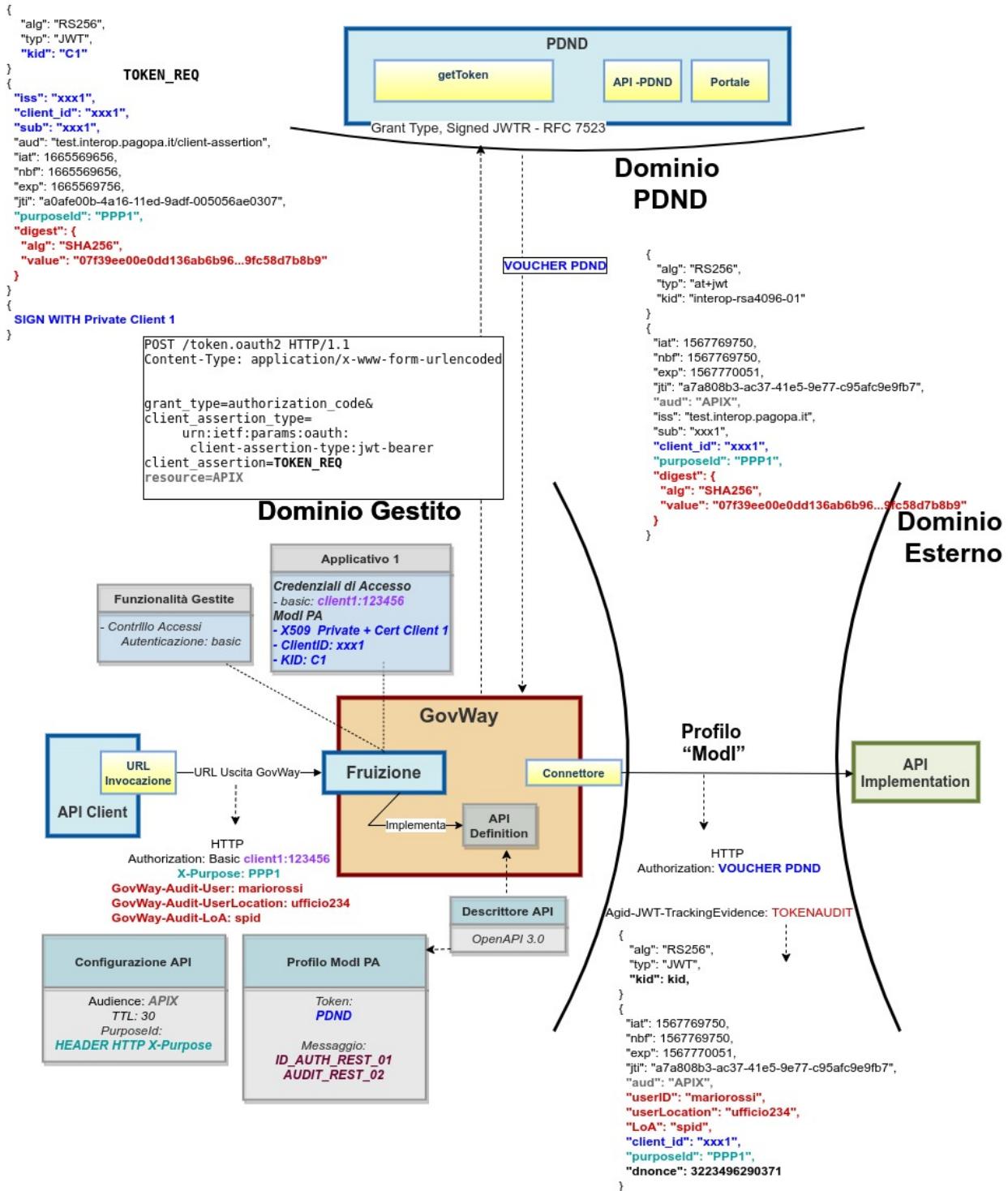


Figure3.99: Profilo di Interoperabilità “ModI”, pattern “AUDIT_REST_02”: trust tramite PDND

Modi

Sicurezza Canale

Pattern: ID_AUTH_CHANNEL_01
Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern: ID_AUTH_REST_01
Direct Trust con certificato X.509

Generazione Token: Authorization PDND
Token ID_AUTH negoziato con la PDND

Informazioni Audit: Dati del dominio del fruitore

Informazioni Audit

Pattern: AUDIT_REST_02
Schema Dati: Linee Guida Modi i

Opzionale:

Figure3.100: Pattern di sicurezza messaggio «ID_AUTH» + Informazioni Audit «AUDIT_REST_02»

Modi

Sicurezza Canale

Pattern: ID_AUTH_CHANNEL_01
Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern: INTEGRITY_REST_01 con ID_AUTH_REST_01
Integrità payload del messaggio

Generazione Token: Authorization Modi
Token ID_AUTH generato dal mittente secondo le Linee Guida 'Modi'

Header HTTP del Token: Agid-JWT-Signature + Authorization Bearer

Applicabilità: Richiesta e Risposta

Digest Richiesta: Non ripudiabilità della trasmissione i

Informazioni Audit: Dati del dominio del fruitore

Informazioni Audit

Pattern: AUDIT_LEGACY i

Figure3.101: Pattern di sicurezza messaggio «INTEGRITY» + Informazioni Audit «AUDIT_LEGACY»

PAYOUT: DATA

```
{
  "iat": 1592905216,
  "nbf": 1592905216,
  "exp": 1592935216,
  "jti": "750e45fd-02b9-4630-9ad8-5fa31f18b53d",
  "aud": "https://api.agenzientrate.it/allegati-demo",
  "client_id": "AvvocaturaStato/App2",
  "iss": "EnteFruitore",
  "sub": "marirossi",
  "user_ip": "10.114.87.24",
  "signed_headers": [
    {
      "digest": "SHA-256=bd9e1f64cbc5b602eee10dd2202c6cf3cf9bdcfac8305756c79d13cb523048b3"
    },
    {
      "content-type": "application/json"
    }
  ]
}
```

Figure3.102: Payload del Token di Sicurezza REST con pattern «INTEGRITY_REST_01» + Informazioni Audit «AUDIT_LEGACY»

```

</wsu:Timestamp>
<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:xsii="http://www.w3.org/2001/XMLSchema-instance" ID="4d961d77-fac5-4f9c-a5a6-3fb7506b058f" IssueInstant="2020-06-23T09:46:40.112Z" Version="2.0" xsi:type="saml2:AssertionType">
  <saml2:Issuer Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">EnteFruitore</saml2:Issuer>
  <saml2:Subject>
    <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">EnteFruitore</saml2:NameID>
    <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches">
      <saml2:SubjectConfirmationData NotBefore="2020-06-23T09:46:40.112Z" NotOnOrAfter="2020-06-23T10:46:40.112Z"/>
    </saml2:SubjectConfirmation>
  </saml2:Subject>
  <saml2:Conditions NotBefore="2020-06-23T09:46:40.112Z" NotOnOrAfter="2020-06-23T10:46:40.112Z"/>
  <saml2:AuthnStatement AuthnInstant="2020-06-23T09:46:40.112Z" SessionNotOnOrAfter="2020-06-23T10:46:40.112Z">
    <saml2:AuthnContext>
      <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</saml2:AuthnContextClassRef>
    </saml2:AuthnContext>
  </saml2:AuthnStatement>
  <saml2:AttributeStatement>
    <saml2:Attribute Name="User" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
      <saml2:AttributeValue xmlns:xsd="http://www.w3.org/2001/XMLSchema">marirossi</saml2:AttributeValue>
    </saml2:Attribute>
    <saml2:Attribute Name="IP-User" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
      <saml2:AttributeValue xmlns:xsd="http://www.w3.org/2001/XMLSchema">10.114.87.24</saml2:AttributeValue>
    </saml2:Attribute>
  </saml2:AttributeStatement>
  </saml2:Assertion>
</wsse:Security>
<wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:env="http://www.w3.org/2003/05/soap-envelope">
```

Figure3.103: Payload del Token di Sicurezza SOAP con pattern «INTEGRITY_SOAP_01» + Informazioni Audit «AUDIT_LEGACY»

- Codice Ente: per default questa informazione assume il valore del soggetto registrato su GovWay, di dominio interno, per il quale si sta effettuando la richiesta di fruizione dell'API.

Il comportamento di default, per l'acquisizione dei valori utilizzati per le tre informazioni aggiuntive, può essere personalizzato accedendo nella sezione «ModI» di una fruizione, e modificando le voci «Informazioni Utente» (Fig. 3.104) indicando un valore statico o utilizzando le proprietà dinamiche descritte nella sezione *Valori dinamici*.

Figure3.104: Personalizzazione dell'acquisizione delle Informazioni Utente

3.3.7 Funzionalità Avanzate

La gestione dei pattern di sicurezza messaggio possono essere personalizzati su diversi aspetti:

- *Attivazione di pattern di sicurezza messaggio differenti per la singola operazione*: è possibile modificare i pattern di sicurezza messaggio applicati puntualmente solamente su una operazione.
- *Attivazione della sicurezza messaggio su richiesta/risposta*: è possibile attivare la sicurezza messaggio puntualmente solamente sulla richiesta o sulla risposta di una operazione. Per API REST è possibile anche definire dei criteri di applicabilità della sicurezza messaggio in base a Content-Type o codici di risposta HTTP.
- *Header HTTP del token JWT*: può essere selezionato l'header http utilizzato per veicolare il token JWT su API REST.
- *Personalizzazione del Token “JWT-Signature”*: può essere attivato un token “JWT-Signature” personalizzato per API REST.
- *Payload Claims del token JWT*: possono essere configurati ulteriori claims da aggiungere nel payload del JWT su API REST.

- *Header SOAP aggiunti nella WSSecurity Signature*: possono essere configurati ulteriori header soap da aggiungere agli elementi inclusi nella firma su API SOAP.
- *Eliminazione token/header contenente la sicurezza messaggio*: è possibile configurare GovWay al fine di non eliminare il token di sicurezza dai messaggi dopo averli validati.
- *Keystore di firma in una fruizione di API*: è possibile attivare differenti scenari di fruizione rispetto a quello di default che prevede l'associazione del keystore di firma sull'applicativo mittente.
- *Finalità (purposeId) utilizzata per una fruizione di API*: l'identificativo univoco della finalità, ottenuto dalla PDND, per cui si intende fruire di un servizio è configurabile in diverse modalità a seconda dello scenario che si desidera supportare.
- *Intermediario*: è possibile indicare un soggetto come intermediario. Questo consente di autorizzare una richiesta proveniente da un soggetto identificato sul canale (l'intermediario), che risulta differente dal soggetto a cui appartiene l'applicativo identificato tramite token di sicurezza.
- *Recupero informazioni client tramite API PDND fallito*: è possibile modificare il comportamento di default per far fallire la transazione in caso il recupero delle informazioni sul client o sull'organizzazione tramite *API PDND* fallisca.
- *Politiche di Rate Limiting basate su informazioni prelevate via API PDND*: le politiche di rate limiting consentono di conteggiare le richieste o filtrare anche rispetto alle informazioni prelevate dalla PDND.

Attivazione di pattern di sicurezza messaggio differenti per la singola operazione

Nella maschera di gestione di un'operazione di una API, all'interno della sezione “ModI”, per default viene fornita una sicurezza messaggio ereditata da quanto indicato sull'API stessa tramite l'opzione “Usa pattern API” (Fig. 3.105).

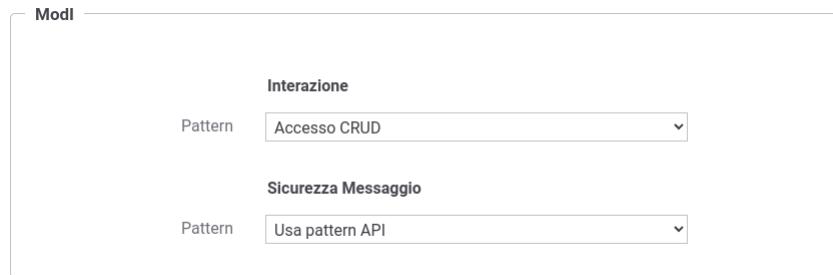


Figure 3.105: Pattern ModI dell'operazione ereditati dall'API

Selezionando la voce “Ridefinito” nel campo “Pattern” della sicurezza messaggio è possibile ridefinire il pattern utilizzato rispetto a quello definito complessivamente sull'API:

- selezionando l'opzione “-” è possibile indicare di non applicare alcun pattern di sicurezza per l'operazione (Fig. 3.106);
- selezionando un qualsiasi altro pattern di sicurezza messaggio è possibile applicare un pattern differente per l'operazione (Fig. 3.107);

ModI

Interazione

Pattern Accesso CRUD

Sicurezza Messaggio

Pattern Ridefinito

-



Figure3.106: Pattern ModI dell'operazione ridefiniti: sicurezza messaggio disabilitata

ModI

Interazione

Pattern Accesso CRUD

Sicurezza Messaggio

Pattern Ridefinito

INTEGRITY_REST_02 con ID_AUTH_REST_01

Integrità payload del messaggio

Generazione Token Authorization PDND

Token ID_AUTH negoziato con la PDND

Header HTTP del Token Aigid-JWT-Signature + Authorization Bearer

Applicabilità Richiesta e Risposta

Digest Richiesta Non ripudiabilità della trasmissione (i)

Informazioni Audit Dati del dominio del fruitore

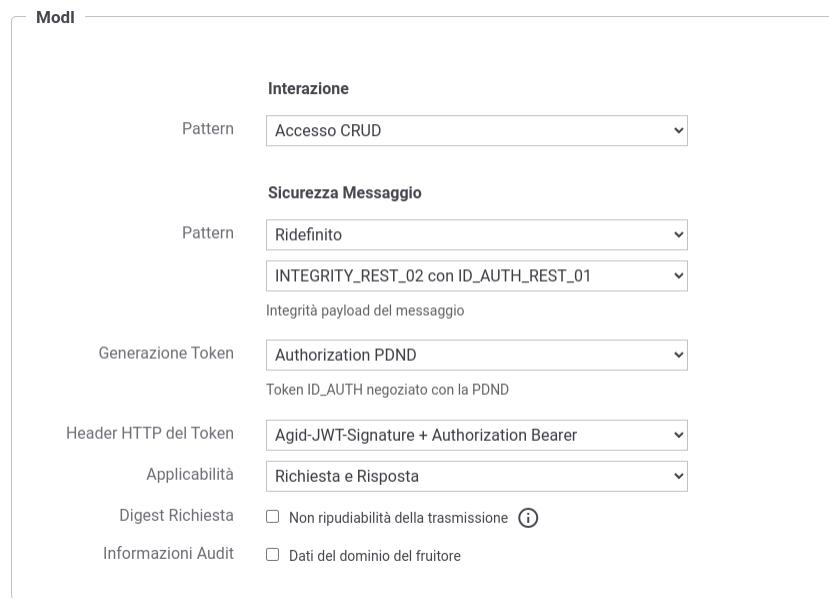


Figure3.107: Pattern ModI dell'operazione ridefiniti

Attivazione della sicurezza messaggio su richiesta/risposta

Insieme all'attivazione di un pattern di sicurezza messaggio è possibile configurarne l'attivazione solamente sulla richiesta o sulla risposta (Fig. 3.108).

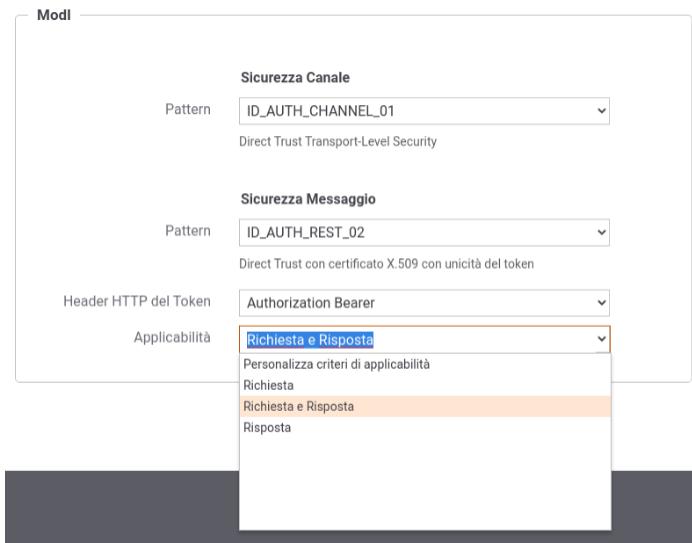


Figure 3.108: Configurazione dell'applicabilità della sicurezza messaggio

Per API REST è possibile anche definire dei criteri di applicabilità della sicurezza messaggio in base a Content-Type o codici di risposta HTTP selezionando la voce “Personalizza criteri di applicabilità”. La personalizzazione dei criteri consente di differenziare la configurazione tra richiesta e risposta come mostrato nella figura Fig. 3.109:

- Richiesta: oltre ad abilitare o disabilitare, è consentito definire una lista di Content-Type solamente per i quali verrà attuata la sicurezza messaggio sulla richiesta.
- Risposta: oltre ad abilitare o disabilitare, è consentito definire una lista di Content-Type e/o una lista di codice di risposta HTTP per i quali verrà attuata la sicurezza messaggio sulla risposta.

La lista di Content-Type per i quali la sicurezza messaggio verrà utilizzata è definibile tramite i seguenti formati:

- type/subtype: indicazione puntuale di un Content-Type
- type/*: hanno un match tutti i Content-Type appartenenti al tipo indicato
- */*+xml: hanno un match tutti i Content-Type che terminano con “+xml”
- regexpType/regexpSubType: hanno un match tutti i Content-Type che soddisfano le espressioni regolari indicate
- empty: valore speciale che rappresenta una richiesta senza Content-Type

La lista dei codici di risposta HTTP per i quali la sicurezza messaggio verrà utilizzata può contenere un codice http puntuale (es. 200) o un intervallo fornendo due codici separati dal trattino (es. 200-299).

Modi

Sicurezza Canale

Pattern ▼
Direct Trust Transport-Level Security

Sicurezza Messaggio

Pattern ▼
Direct Trust con certificato X.509 con unicità del token

Header HTTP del Token ▼

Applicabilità ▼

Sicurezza Messaggio nella Richiesta

Stato ▼

Content-Type * ⓘ

Sicurezza Messaggio nella Risposta

Stato ▼

Content-Type * ⓘ

Codice Risposta * ⓘ

Figure3.109: Configurazione dell'applicabilità della sicurezza messaggio personalizzata per Content-Type e Codici di Risposta

Header HTTP del token JWT

Il pattern di sicurezza, su API di tipo REST, produrrà la generazione di un token JWT firmato inserito all'interno dell'header HTTP previsto dalle *Linee Guida AGID di Interoperabilità* (LG) dove vengono definiti gli header HTTP “Authorization” (Bearer) da usare per l'autenticazione e l'header HTTP “Agid-JWT-Signature” per l'integrità.

La configurazione di default degli header prodotti varia a seconda del pattern di sicurezza selezionato:

- *ID_AUTH_SOAP_01 / ID_AUTH_REST_01 - Direct Trust con certificato X.509*: header HTTP “Authorization”.
- *ID_AUTH_SOAP_02 / ID_AUTH_REST_02 - Direct Trust con certificato X.509 con unicità del messaggio/token*: header HTTP “Authorization”.
- *INTEGRITY_SOAP_01 / INTEGRITY_REST_01 - Integrità payload del messaggio*: nel flusso di richiesta vengono prodotti entrambi gli header, mentre nel flusso di risposta solamente l'header HTTP “Agid-JWT-Signature”.

La voce “Header HTTP del Token” consente di modificare la configurazione di default e variare sia il nome che l'eventuale coesistenza dei 2 header principalmente per due motivi:

- per consentire la retrocompatibilità con il pattern IDAR03, previsto nelle linee guida della versione “bozza” (<https://docs.italia.it/italia/piano-triennale-ict/lg-modellointeroperabilita-docs/it/bozza/>), dove veniva utilizzato con qualsiasi pattern sempre un unico header HTTP “Authorization”;
- per supportare qualsiasi interpretazione del pattern “INTEGRITY_REST_01” e relativa implementazione da parte della controparte con cui si deve interoperare.

Per i motivi suddetti le possibili configurazioni supportate configurabili tramite la voce “Header HTTP del Token” sono le seguenti (Fig. 3.110):

- “Agid-JWT-Signature + Authorization Bearer” : opzione selezionabile solamente con pattern che prevede “INTEGRITY_REST_01”. Prevede la generazione nel flusso di richiesta di entrambi gli header http previsti dalle LG. Nel flusso di risposta è previsto invece solamente l'header “Agid-JWT-Signature”.
- “Agid-JWT-Signature + Authorization Bearer anche nella risposta”: comportamento identico all'opzione precedente, dove però la coesistenza dei due header è prevista anche nel flusso di risposta.
- “Agid-JWT-Signature” : viene generato sempre e solo un unico token di sicurezza, indipendentemente dal pattern di sicurezza selezionato, utilizzando come nome dell'header HTTP il nome proprietario delle LG.
- “Authorization Bearer” : comportamento identico all'opzione precedente, dove però viene utilizzato l'header HTTP “Authorization” e prefisso “Bearer” nel valore. Questa opzione, in presenza di pattern che prevede “INTEGRITY_REST_01”, consente di essere interoperabile con i servizi implementati con la versione “bozza” delle LG.

Sono inoltre disponibili ulteriori opzioni che hanno il medesimo comportamento delle voci precedentemente descritte, con la particolarità che consentono di avere un header per l'integrità del messaggio personalizzato.

- “Custom-JWT-Signature + Authorization Bearer”
- “Custom-JWT-Signature + Authorization Bearer anche nella risposta”
- “Custom-JWT-Signature”

Se viene selezionata una gestione personalizzata, tutta la parte relativa alla gestione dell'integrità (calcolo/verifica Digest, gestione claim “signed_header”) non viene effettuata built-in e viene delegata a livello applicativo. Maggiori dettagli vengono forniti nella sezione *Personalizzazione del Token “JWT-Signature”*.

Nota: Processamento dell'header Agid-JWT-Signature con opzioni che prevedono anche l'header Authorization

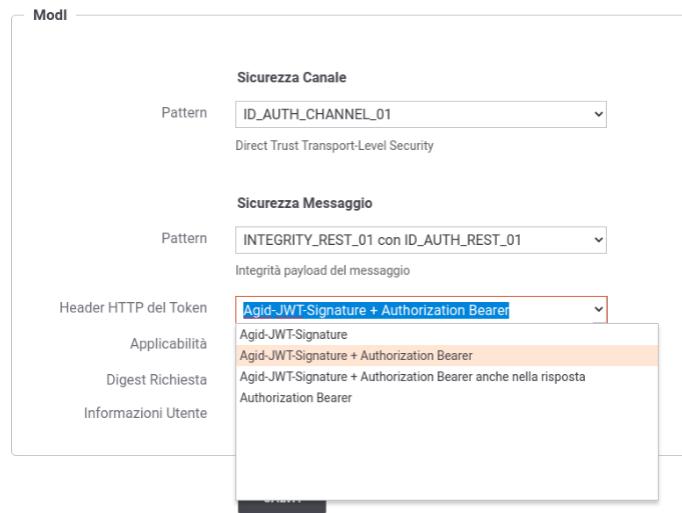


Figure3.110: Selezione dell'Header HTTP del token JWT

Se un'API o una risorsa è stata configurata con un'opzione che prevede la coesistenza dei 2 header, la generazione o la verifica dell'header “Agid-JWT-Signature” avviene solamente se la richiesta o la risposta prevede un payload; ad esempio in una richiesta HTTP GET non verrà generato o atteso l'header. Caso eccezionale riguarda i flussi configurati per firmare header http indipendenti dal payload della richiesta (diversi quindi dai classici header Content-Type, Content-Encoding, Digest); in questi casi l'header “Agid-JWT-Signature” verrà generato anche in assenza di payload poichè nel token saranno inseriti nel claim “signed_headers” gli ulteriori header configurati per essere firmati.

Se in un'API viene selezionata una opzione che prevede la coesistenza dei 2 header, nelle maschere di configurazione ModI delle fruizioni e delle erogazioni saranno presenti ulteriori opzioni che consentono di personalizzare la gestione dei claims presenti all'interno dei due header. Di seguito vengono descritte le opzioni ulteriori presenti nelle fruizioni e nelle erogazioni

Configurazione coesistenza degli header in una Fruizione

La configurazione differisce a seconda se il token di sicurezza deve essere generato (Richiesta) o validato (Risposta). Di seguito vengono descritte le opzioni di configurazione possibili per le due fasi.

Richiesta

Di seguito vengono descritti i parametri di configurazione riguardanti la generazione dei due token nel flusso di richiesta.

- Identificativo “jti”: la configurazione di default prevede la valorizzazione del claim “jti”, presente all'interno dei token di sicurezza portati dai due header, con il medesimo identificativo. Il parametro consente di impostare la generazione di un identificativo differente e di indicare al gateway quale dei due identificativi dovrà associare alla traccia come “ID del Messaggio” tramite la voce “Usa come ID Messaggio” (Fig. 3.112). L'identificativo indicato sarà utilizzabile come criterio di ricerca puntuale tramite la funzionalità disponibile con la console e le API di monitoraggio. L'identificativo “jti” presente nell'altro header verrà comunque associato alla traccia ma non sarà direttamente utilizzabile come criterio di ricerca per individuare la transazione in un secondo momento.
- Audience: la configurazione di default prevede la valorizzazione del claim “aud”, presente all'interno dei token di sicurezza portati dai due header, con il medesimo identificativo. Il parametro consente di impostare la

^ Contemporaneità Token Authorization e Agid-JWT-Signature

Identificativo 'jti'	Stesso identificativo
Audience	Stesso identificativo
Claims 'Authorization'	<input type="text"/>
Claims 'Agid-JWT-Signature'	<input type="text"/>

Indicare per riga i claims (nome=valore); visualizzare 'info' per maggiori dettagli

Figure3.111: Maschera ModI per la configurazione della fruizione in presenza dei due header nella richiesta

^ Contemporaneità Token Authorization e Agid-JWT-Signature

Identificativo 'jti'	Differente identificativo
Usa come ID Messaggio	<input type="text" value="Authorization"/> Agid-JWT-Signature Authorization

Figure3.112: Selezione del claim “jti” da utilizzare come ID del Messaggio

generazione di un identificativo differente indicando l’audience da impostare nel token dell’header Agid-JWT-Signature come mostrato nella figura Fig. 3.113.

Audience	Differente identificativo
*	<input type="text" value="Valore da inserire in Agid-JWT-Signature"/>

Figure3.113: Configurazione del claim “aud” da utilizzare nel token dell’header Agid-JWT-Signature

- Claims “Authorization” e “Agid-JWT-Signature”: consente di modificare i valori di default associati ai claim standard e/o di definirne altri solamente all’interno del token dell’header indicato. Per maggiori dettagli sulle configurazioni disponibili si rimanda alla sezione *Payload Claims del token JWT*.

Risposta

Di seguito vengono descritti i parametri di configurazione riguardanti la validazione dei due token nel flusso di risposta.

- Id “jti” per Filtro Duplicati: consente di indicare da quale header estrarre l’identificativo “jti” da associare alla traccia come “ID del Messaggio” (default: Agid-JWT-Signature). L’identificativo indicato verrà utilizzato per la funzionalità di filtro delle richieste duplicate. Inoltre l’identificativo indicato sarà utilizzabile come criterio di ricerca puntuale tramite la funzionalità disponibile con la console e le API di monitoraggio. L’identificativo “jti” presente nell’altro header verrà comunque associato alla traccia ma non sarà direttamente utilizzabile come criterio di ricerca per individuare la transazione in un secondo momento.
- Audience: la configurazione di default si attende un valore del claim “aud”, presente all’interno dei token di sicurezza portati dai due header, con il medesimo identificativo. Il parametro consente di impostare la generazione di un identificativo differente indicando l’audience atteso nell’header Agid-JWT-Signature come mostrato nella figura Fig. 3.115.

^ Contemporaneità Token Authorization e Agid-JWT-Signature

Id 'jti' per Filtro Duplicati	Authorization
Audience	Stesso identificativo

Figure3.114: Maschera ModI per la configurazione della fruizione in presenza dei due header nella risposta

Audience	Differente identificativo
*	Valore atteso in Agid-JWT-Signature

Figure3.115: Valore atteso per il claim “aud” nel token dell’header Agid-JWT-Signature

Configurazione coesistenza degli header in una Erogazione

La configurazione differisce a seconda se il token di sicurezza deve essere validato (Richiesta) o generato (Risposta). Di seguito vengono descritte le opzioni di configurazione possibili per le due fasi.

Richiesta

Di seguito vengono descritti i parametri di configurazione riguardanti la validazione dei due token nel flusso di richiesta.

^ Contemporaneità Token Authorization e Agid-JWT-Signature

Id 'jti' per Filtro Duplicati	Authorization
Audience	Stesso identificativo

Figure3.116: Maschera ModI per la configurazione dell’erogazione in presenza dei due header nella richiesta

- Id “jti” per Filtro Duplicati: consente di indicare da quale header estrarre l’identificativo “jti” da associare alla traccia come “ID del Messaggio” (default: Agid-JWT-Signature). L’identificativo indicato verrà utilizzato per la funzionalità di filtro delle richieste duplicate. Inoltre l’identificativo indicato sarà utilizzabile come criterio di ricerca puntuale tramite la funzionalità disponibile con la console e le API di monitoraggio. L’identificativo “jti” presente nell’altro header verrà comunque associato alla traccia ma non sarà direttamente utilizzabile come criterio di ricerca per individuare la transazione in un secondo momento.
- Audience: la configurazione di default si attende un valore del claim “aud”, presente all’interno dei token di sicurezza portati dai due header, con il medesimo identificativo. Il parametro consente di impostare la generazione di un identificativo differente indicando l’audience atteso nell’header Agid-JWT-Signature come mostrato nella figura Fig. 3.115.

Risposta

Di seguito vengono descritti i parametri di configurazione riguardanti la generazione dei due token nel flusso di risposta.

^ Contemporaneità Token Authorization e Agid-JWT-Signature

Identificativo 'jti'	Stesso identificativo
Claims 'Authorization'	
Claims 'Agid-JWT-Signature'	

Indicare per riga i claims (nome=valore); visualizzare 'info' per maggiori dettagli

Figure3.117: Maschera ModI per la configurazione dell’erogazione in presenza dei due header nella risposta

- Identificativo “jti”: la configurazione di default prevede la valorizzazione del claim “jti”, presente all’interno dei token di sicurezza portati dai due header, con il medesimo identificativo. Il parametro consente di impostare la generazione di un identificativo differente e di indicare al gateway quale dei due identificativi dovrà associare alla traccia come “ID del Messaggio” tramite la voce “Usa come ID Messaggio” (Fig. 3.112). L’identificativo indicato sarà utilizzabile come criterio di ricerca puntuale tramite la funzionalità disponibile con la console e le API di monitoraggio. L’identificativo “jti” presente nell’altro header verrà comunque associato alla traccia ma non sarà direttamente utilizzabile come criterio di ricerca per individuare la transazione in un secondo momento.
- Claims “Authorization” e “Agid-JWT-Signature”: consente di modificare i valori di default associati ai claim standard e/o di definirne altri solamente all’interno del token dell’header indicato. Per maggiori dettagli sulle configurazioni disponibili si rimanda alla sezione *Payload Claims del token JWT*.

Personalizzazione del Token “JWT-Signature”

Le *Linee Guida AGID di Interoperabilità* (LG) definisce un pattern di sicurezza per garantire l’integrità, su API di tipo REST, che utilizza un token JWT firmato inserito all’interno dell’header HTTP “Agid-JWT-Signature”.

Come è stato descritto nella sezione *Header HTTP del token JWT* è possibile configurare la generazione di un token personalizzato per l’integrità attraverso la voce “Header HTTP del Token” selezionando una delle seguenti opzioni:

- “Custom-JWT-Signature + Authorization Bearer”
- “Custom-JWT-Signature + Authorization Bearer anche nella risposta”
- “Custom-JWT-Signature”

Se viene selezionata una gestione personalizzata, tutta la parte relativa alla gestione dell’integrità (calcolo/verifica Digest, gestione claim “signed_header”) non viene effettuata built-in e viene delegata a livello applicativo. In fase di fruizione è possibile aggiungere all’interno del token personalizzato i claim ricevuti dall’applicativo client (dettagli forniti nella sezione *Payload Claims del token JWT*). In fase di erogazione è possibile far arrivare all’applicativo il token completo (sezione *Eliminazione token/header contenente la sicurezza messaggio*) o i singoli claim (securityToken descritto in *Valori dinamici*) in modo da consentirgli di attuare il processo di validazione personalizzato.

L’attivazione di un token personalizzato per l’integrità comporta la configurazione della voce “Custom-JWT-Signature” nei seguenti aspetti:

- deve essere indicato il nome dell’header HTTP su cui viene veicolato il token personalizzato;
- deve essere scelto se generare il token solamente in presenza di un payload http (comportamento di default) o sempre per qualsiasi risorsa.

Sicurezza Messaggio	
Pattern	INTEGRITY_REST_01 con ID_AUTH_REST_01
Integrità payload del messaggio	
Generazione Token	Authorization PDND
Token ID_AUTH negoziato con la PDND	
Header HTTP del Token	Custom-JWT-Signature + Authorization Bearer
Custom-JWT-Signature *	NomePersonalizzato-JWT-Signature
Presente solo con payload HTTP	
Applicabilità	Richiesta
Informazioni Audit	
<input type="checkbox"/> Dati del dominio del fruitore	

Figure3.118: Attivazione di un token Custom-JWT-Signature per l'integrità della richiesta

Payload Claims del token JWT

Il pattern di sicurezza, su API di tipo REST, produrrà la generazione di un token JWT firmato inserito all'interno dell'header HTTP previsto dalle *Linee Guida AGID di Interoperabilità*. Nel payload del JWT vengono generati i claim di default previsti dal prodotto come quelli temporali (iat, nbf, exp), l'identificativo unico della richiesta (jti), e altri claims che consentono di individuare gli attori (sub, iss, client_id, aud).

Altri claims possono essere aggiunti al payload JWT definendoli nel campo “Claims” tra i criteri di configurazione “ModI” della richiesta, in una fruizione, o della risposta, in una erogazione. Vanno indicati per riga nella forma “nome=valore” come mostrato nella figura Fig. 3.119. Il valore può essere definito come costante o contenere parti dinamiche, definite tramite una sintassi proprietaria di GovWay, che verranno risolte a runtime dal Gateway (per maggiori dettagli *Valori dinamici*). Ulteriori modalità per aggiungere claim proprietari vengono descritti nella sezione *Aggiunta di Claims nei Token*.

Claims	<pre>iss=\${clientOrganizationConfig:iss} sub=\${transportContext:credential.identity} custom_claim=\${config:pCustom}</pre>
--------	--

Indicare per riga i claims (nome=valore); visualizzare 'info' per maggiori dettagli

Figure3.119: Claims aggiuntivi inseriti nel Payload JWT

Nota: Non è consentito indicare i claims “iat, nbf, exp, jti”. In una richiesta non è inoltre consentito indicare ne il claim “aud” ne il claim “client_id” (quest’ultimo prevede un caso eccezionale con il valore \${notGenerate} descritto in seguito). In una risposta, invece, non è consentito indicare il claim “request_digest”.

Di seguito vengono forniti i valori di default inseriti da GovWay nel payload jwt per quanto concerne i claims che individuano gli attori, differenziando tra il token di richiesta generato da una fruizione e il token di risposta generato da una erogazione. Per ogni claim viene anche indicato come modificare il valore di default associato.

- “aud”: indica a chi è riferito il security token
 - fruizione: il valore da inserire nel payload JWT può essere indicato tra i criteri di configurazione “ModI”, nella sezione richiesta. Se non viene fornito un valore verrà utilizzata la url del connettore.

- erogazione: viene utilizzato il primo valore che ha un match in ordine con i seguenti criteri:
 - * claim “aud” indicato nel campo “Claims” tra i criteri di configurazione “ModI” della risposta;
 - * valore configurato nel campo “Identificativo Client” dell’applicativo mittente identificato;
 - * valore presente nel claim “client_id” del payload JWT ricevuto nella richiesta;
 - * valore presente nel claim “sub” del payload JWT ricevuto nella richiesta;
 - * valore “anonymous”

Nota: Il claim “aud”, inserito nei token generati nella richiesta in una fruizione e nella risposta in una erogazione, viene generato come “single case-sensitive string”. È possibile generarlo come “array of case-sensitive strings” indicandolo tra array (es. `[valoreAudience]`) e in questo caso è possibile indicare anche molteplici valori (es. `[valoreAudience1, valoreAudience2]`).

- “iss”: identificativo del soggetto che ha rilasciato (e firmato) il token; viene utilizzato il primo valore che ha un match in ordine con i seguenti criteri:
 - claim “iss” indicato nel campo “Claims” tra i criteri di configurazione “ModI” della richiesta, in una fruizione, o della risposta, in una erogazione;
 - identificativo del soggetto fruitore in una fruizione o l’identificativo del soggetto erogatore in una erogazione
- “sub”: identificativo del mittente a cui è riferito il token; viene utilizzato il primo valore che ha un match in ordine con i seguenti criteri:
 - claim “sub” indicato nel campo “Claims” tra i criteri di configurazione “ModI” della richiesta, in una fruizione, o della risposta, in una erogazione;
 - in una fruizione l’identificativo varia a seconda della modalità di keystore:
 - * nel caso di keystore di firma associato all’applicativo mittente viene utilizzato l’identificativo dell’applicativo;
 - * nel caso di keystore definito nella fruizione viene utilizzato l’identificativo e la versione dell’API fruita;
 - in una erogazione viene utilizzato l’identificativo e la versione dell’API implementata.
- “client_id”: identificativo dell’applicazione client che ha ottenuto il token; viene utilizzato il primo valore che ha un match in ordine con i seguenti criteri:
 - fruizione che richiede un keystore associato all’applicativo mittente:
 - * valore configurato nel campo “Identificativo Client” dell’applicativo mittente identificato;
 - * identificativo dell’applicativo mittente
 - fruizione che richiede un keystore associato alla fruizione:
 - * identificativo e versione dell’API fruita
 - erogazione:
 - * claim “client_id” indicato nel campo “Claims” tra i criteri di configurazione “ModI” della risposta;
 - * identificativo e versione dell’API implementata

Nota: È possibile utilizzare la keyword “\${notGenerate}” come valore dei claims “iss”, “sub” o “client_id”, indicati nel campo “Claims” tra i criteri di configurazione “ModI”, per non far generare il claim all’interno del jwt payload.

Header SOAP aggiunti nella WSSecurity Signature

Il pattern di sicurezza, su API di tipo SOAP, richiede la creazione una WSSecurity Signature dove all'interno vengono firmati gli elementi principali della richiesta (Timestamp, wsa:To) e gli altri elementi richiesti dai profili **ID_AUTH_SOAP_02 / ID_AUTH_REST_02 - Direct Trust con certificato X.509 con unicità del messaggio/token** (wsa:MessageId) e **INTEGRITY_SOAP_01 / INTEGRITY_REST_01 - Integrità payload del messaggio** (Body).

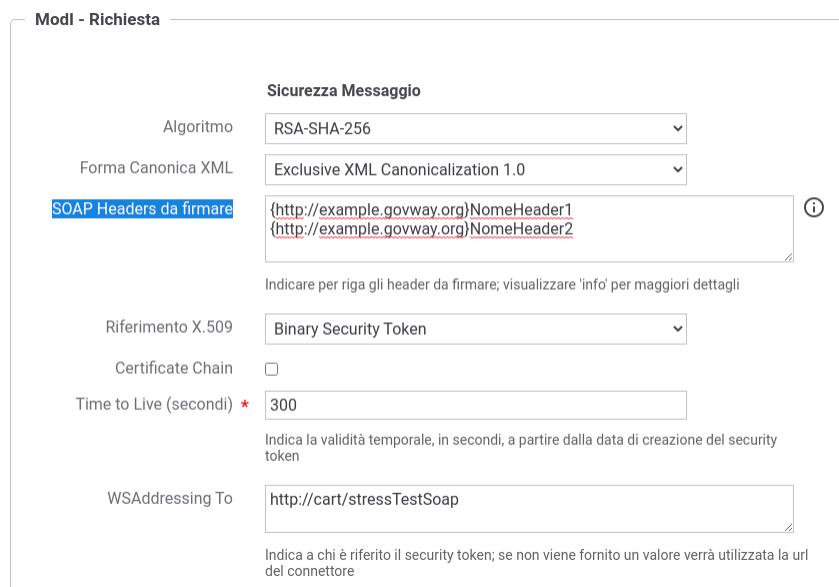
È possibile aggiungere, tra gli elementi firmati, ulteriori header SOAP oltre a quelli previsti dalla specifica ModI. Gli ulteriori header possono essere indicati nell'elemento “SOAP Headers da firmare” presente nella sezione “ModI - Richiesta” di una fruizione o nella sezione “ModI - Risposta” di una erogazione, come mostrato nella figura Fig. 3.120. Gli header devono essere definiti su ogni riga tramite la sintassi:

- {namespace}localName

Ad esempio:

- {<http://example.govway.org>}NomeHeader1
- {<http://example.govway.org>}NomeHeader2

Nota: L'elemento “SOAP Headers da firmare” è disponibile solamente utilizzando la govwayConsole in modalità *avanzata* (sezione *Modalità Avanzata*).



The screenshot shows the 'ModI - Richiesta' configuration screen. The 'SOAP Headers da firmare' field is highlighted, containing the following entries:

```
{http://example.govway.org}NomeHeader1
{http://example.govway.org}NomeHeader2
```

Figure3.120: Configurazione Header SOAP aggiuntivi da aggiungere alla firma

Eliminazione token/header contenente la sicurezza messaggio

I Token di sicurezza, dopo essere stati validati da GovWay, vengono eliminati dai messaggi in modo da rendere trasparente agli applicativi la gestione della sicurezza che è stata effettuata sul Gateway.

È possibile, se necessario, configurare GovWay al fine di non fargli eliminare il token di sicurezza dai messaggi dopo averli validati. Per farlo si deve utilizzare la govwayConsole in modalità avanzata (vedi sezione [Modalità Avanzata](#)).

Per quanto concerne le richieste inoltrate ad un backend, durante la gestione di una erogazione, è possibile disabilitare l'eliminazione del token di sicurezza intervenendo sul connettore dell'erogazione e disabilitando la voce “Sbustamento ModI” all'interno della sezione “Trattamento Messaggio” come mostrato nella figura [Fig. 3.121](#).

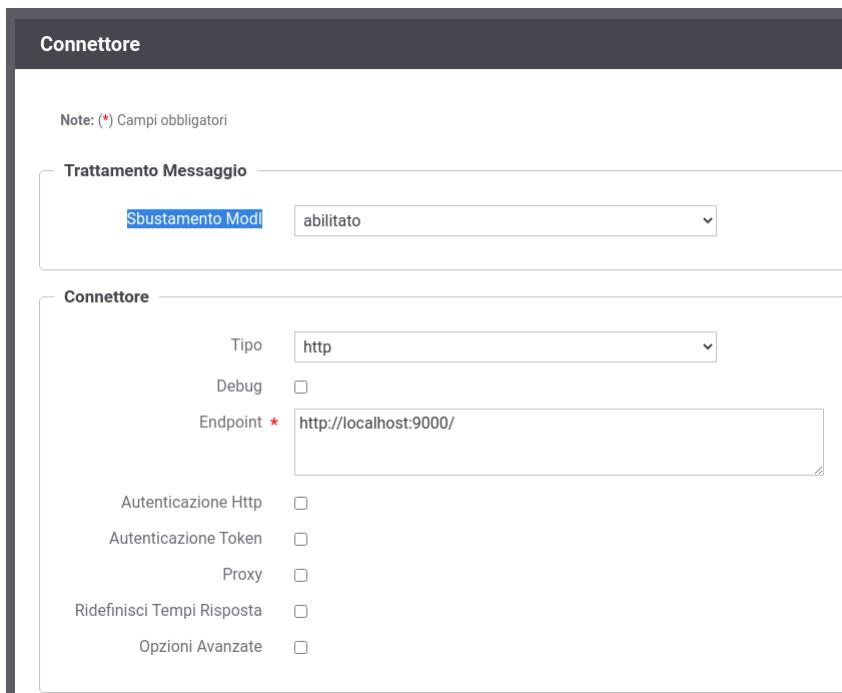


Figure3.121: Funzionalità “Sbustamento ModI” disabilitata per la Richiesta

Sulle risposte ritornate all'applicativo mittente, durante la gestione di una fruizione, è possibile disabilitare l'eliminazione del token di sicurezza intervenendo sull'applicativo e disabilitando la voce “Sbustamento ModI” all'interno della sezione “Trattamento Messaggio” come mostrato nella figura [Fig. 3.122](#).

Keystore di firma in una fruizione di API

Il materiale crittografico utilizzato per una fruizione di API può essere associato a differenti oggetti del registro in funzione del contesto applicativo di utilizzo:

- Materiale associato ad un **applicativo client**: scenario di default descritto nella sezione [Fruizione ID_AUTH_REST_01 / ID_AUTH_SOAP_01 \(X509\)](#) che prevede l'associazione del keystore di firma all'applicativo mittente. In questo scenario si assume che il materiale crittografico identifica l'applicativo il quale lo riutilizzerà su ogni API che necessita di fruire.
- Materiale associato ad una **fruizione**: scenario descritto nella sezione [Keystore di firma definito nella fruizione](#) che prevede l'associazione del keystore di firma alla fruizione di API. In questo scenario si assume che esista

Applicativo

Dominio	Interno
Soggetto	EnteEsterno
Nome *	<input type="text" value="Client1"/>
Tipo	Client

Modalità di Accesso

Tipo	<input type="text" value="http-basic"/>
Utente *	<input type="text" value="Client1"/>
Modifica Password	<input type="checkbox"/>

Ruoli

[visualizza\(0\)](#)

Trattamento Messaggio

Sbustamento ModI	<input type="text" value="abilitato"/>
------------------	--

ModI

Sicurezza Messaggio

Abilitato	<input checked="" type="checkbox"/>
Archivio	

Figure3.122: Funzionalità “Sbustamento ModI” disabilitata per la Risposta

un materiale crittografico dedicato ad ogni fruizione e non riutilizzato come nello scenario precedente. Questo scenario consente agli applicativi che indirizzano la fruizione di uscire verso il dominio esterno tramite un'unica identità rappresentata dal certificato presente nel keystore definito nella fruizione stessa.

- Materiale associato ad una **token policy**: scenario descritto nella sezione *Keystore di firma definito nella token policy* che prevede la definizione di un keystore di firma all'interno di una token Policy. In questo scenario si assume l'esistenza di un unico materiale crittografico che verrà utilizzato per uscire verso il dominio esterno per qualsiasi fruizione di API da qualsiasi applicativo.

Keystore di firma definito nella fruizione

Nello scenario descritto in questa sezione il keystore di firma viene associato alla fruizione e consente quindi a tutti gli applicativi che indirizzano la fruizione di uscire verso il dominio esterno tramite un'unica identità rappresentata dal certificato presente nel keystore. Lo scenario è utilizzabile in quei contesti in cui l'Ente si presenta verso l'API fruita nel dominio esterno sempre tramite un'unica identità ma al suo interno applica un'autorizzazione puntuale identificando il singolo applicativo chiamante.

Per attuare la configurazione è necessario agire nella sezione «ModI - Richiesta»:

- KeyStore: deve essere selezionata la voce “Definito nella fruizione” e successivamente vanno configurati i riferimenti al keystore da utilizzare per la firma:
 - utilizzando il keystore di default (riferimenti descritti in *Chiavi di default per la firma dei token ModI*) non vengono richiesti ulteriori parametri ([Fig. 3.124](#));
 - ridefinendo il keystore dovranno essere indicati i seguenti parametri ([Fig. 3.125](#)):
 - * *Modalità*: il keystore può essere fornito tramite differenti modalità
 - “File System”: deve essere fornito il *Path* assoluto su file system del keystore;
 - “Archivio”: viene effettuato l’upload del keystore;
 - “HSM”: consente di selezionare uno dei tipi di keystore PKCS11 registrati (“*Device PKCS11*”);
 - * *Tipo*: il formato del keystore (jks, pkcs12, tipi di keystore PKCS11 registrati);
 - * *Password*: la password per l’accesso al keystore;
 - * *Alias Chiave Privata*: l’alias con cui è riferita la chiave privata nel keystore;
 - * *Password Chiave Privata*: la password della chiave privata

Per API REST in cui è stata selezionata la voce “Riferimento X.509” nella sezione «ModI - Richiesta» dovrà essere obbligatoriamente anche indicata anche la URL che riferisce un certificato (o certificate chain) X.509 corrispondente alla chiave firmataria del security token ([Fig. 3.126](#)). La url indicata verrà inserita nel token di sicurezza all'interno del claim «x5u».

Nella sezione «ModI - Risposta» se viene abilitata la “Verifica Audience” nel token di sicurezza della risposta deve essere obbligatoriamente indicato il valore atteso ([Fig. 3.127](#)).

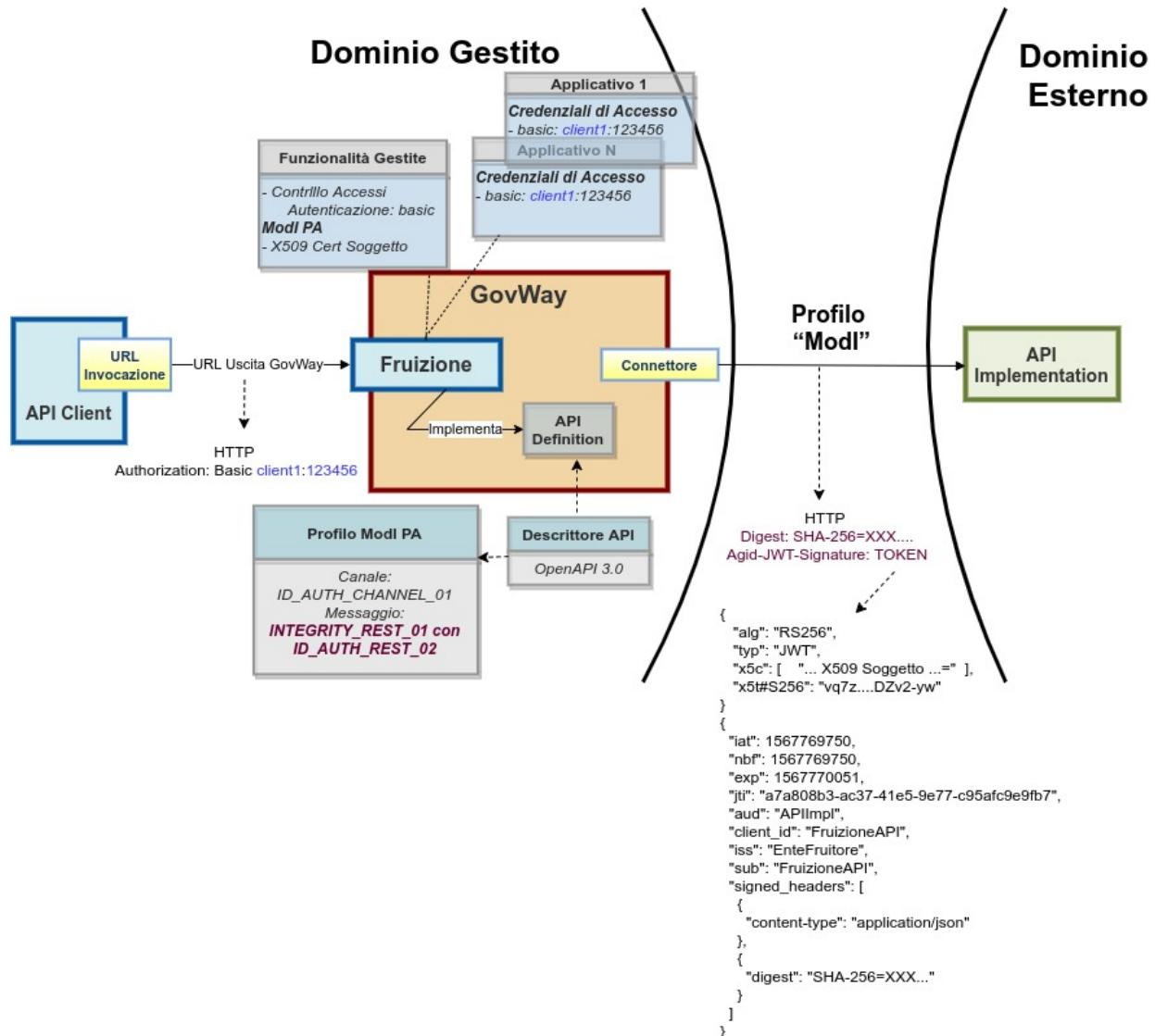


Figure3.123: Fruizione con Profilo di Interoperabilità “ModI”; keystore definito nella fruizione

ModI - Richiesta

Sicurezza Messaggio

Algoritmo	RS256
Codifica Digest	Base64
HTTP Headers da firmare *	Digest <input checked="" type="checkbox"/> Content-Type <input checked="" type="checkbox"/> Content-Encoding <input checked="" type="checkbox"/>
Riferimento X.509	x5c (Certificate) x5t#256 (Certificate SHA-256 Thumbprint) x5u (URL)
Certificate Chain	<input type="checkbox"/>
KeyStore	Definito nella fruizione
	Default
Time to Live (secondi) *	300
Indica la validità temporale, in secondi, a partire dalla data di creazione del security token	
Audience	<input type="text"/> (i)
Indica a chi è riferito il security token; se non viene fornito un valore verrà utilizzata la url del connettore	
Claims	<input type="text"/> (i)
Indicare per riga i claims (nome=valore); visualizzare 'info' per maggiori dettagli	

Figure3.124: Configurazione fruizione “ModI” con keystore di default configurato nella fruizione

Modi - Richiesta

Sicurezza Messaggio

Algoritmo: RS256

Codifica Digest: Base64

HTTP Headers da firmare *: Digest, Content-Type, Content-Encoding

Riferimento X.509: x5c (Certificate), x5t#256 (Certificate SHA-256 Thumbprint), x5u (URL)

Certificate Chain:

KeyStore: Definito nella fruizione, Ridefinito

Time to Live (secondi) *: 300

Indica la validità temporale, in secondi, a partire dalla data di creazione del security token

Audience: ⓘ

Indica a chi è riferito il security token; se non viene fornito un valore verrà utilizzata la url del connettore

Claims: ⓘ

Indicare per riga i claims (nome=valore); visualizzare 'info' per maggiori dettagli

KeyStore

Modalità: File System

Path *:

Tipo: JKS

Password *:

Alias Chiave Privata *:

Password Chiave Privata *:

Figure3.125: Configurazione fruizione “Modi” con keystore ridefinito nella fruizione

ModI - Richiesta

Sicurezza Messaggio

Algoritmo	RS256
Codifica Digest	Base64
HTTP Headers da firmare *	Digest x Content-Type x Content-Encoding x
Riferimento X.509	x5c (Certificate) x5t#256 (Certificate SHA-256 Thumbprint) x5u (URL)
URL (x5u) *	http://<host>/cert.pem
URL che riferisce un certificato (o certificate chain) X.509 corrispondente alla chiave firmataria del security token	
KeyStore	Definito nella fruizione
	Default
Time to Live (secondi) *	300
Indica la validità temporale, in secondi, a partire dalla data di creazione del security token	
Audience	
Indica a chi è riferito il security token; se non viene fornito un valore verrà utilizzata la url del connettore	
Claims	
Indicare per riga i claims (nome=valore); visualizzare 'info' per maggiori dettagli	

Figure3.126: Configurazione fruizione “ModI” con keystore configurato nella fruizione; url del claim “x5u”

Modi - Risposta

Sicurezza Messaggio

Riferimento X.509	Utilizza impostazioni della Richiesta
TrustStore Certificati	Default
TrustStore SSL	Default
Time to Live	Default
Verifica Audience	<input checked="" type="checkbox"/>
* <input type="text"/> (i)	

Figure3.127: Configurazione fruizione “ModI” con verifica dell’audience atteso nella risposta

Keystore di firma definito nella token policy

Nello scenario descritto in questa sezione il keystore di firma viene definito all’interno di una token Policy. Lo scenario è utilizzabile in quei contesti in cui l’Ente si presenta al dominio esterno sempre tramite un’unica identità indipendentemente dall’API fruenda.

Per attuare la configurazione è necessario agire nella sezione «ModI - Richiesta» alla voce “KeyStore” indicando: “Definito nella token policy”

Finalità (purposeId) utilizzata per una fruizione di API

Per ottenere un token dalla PDND un applicativo mittente deve aver registrato una finalità che descrive la motivazione per cui vuole richiedere la fruizione del servizio e il numero di richieste giornaliere che intende effettuare. La creazione di una finalità si completa con l’ottenimento di un suo identificativo univoco denominato «*purposeId*».

Negli scenari di configurazione attuabili su GovWay il *purposeId* può essere indicato in differenti modi, descritti in questa sezione, a seconda della modalità con cui viene definito il campo “Purpose ID” all’interno della Token Policy di negoziazione con la PDND descritta nei passi di configurazione della sezione *Fruizione ID_AUTH_REST_01 (PDND)*.

Le modalità supportate sono le seguenti:

- **statica:** il valore del *purposeId* può essere fornito staticamente nel campo “Purpose ID” della token policy richiedendo quindi la registrazione di una token policy per ogni finalità;
- **fornita dal client:** il valore può contenere una keyword risolta a runtime in modo da valorizzare il claim “*purposeId*” con un valore prelevato dai dati della richiesta. Ad esempio se il censimento dei *purposeId* viene mantenuto a livello applicativo può essere indicato un header HTTP con cui il richiedente può fornire a GovWay il valore da utilizzare (es. \${header:NOME_HEADER_HTTP}). Si rimanda alla sezione “*Valori dinamici*” per le varie modalità dinamiche utilizzabili;
- **proprietà degli oggetti:** di seguito vengono invece fornite alcune indicazioni per mantenere la registrazione del *purposeId* sul registro di GovWay supportando differenti scenari (per maggiori dettagli si rimanda alla sezione *Accesso alle proprietà delle entità del Registro*):
 - *1-1 con la fruizione:* registrazione come proprietà “*purposeId*” della fruizione e riferito nella token policy tramite il valore “\${config:purposeId}”;

Modi - Richiesta

Sicurezza Messaggio

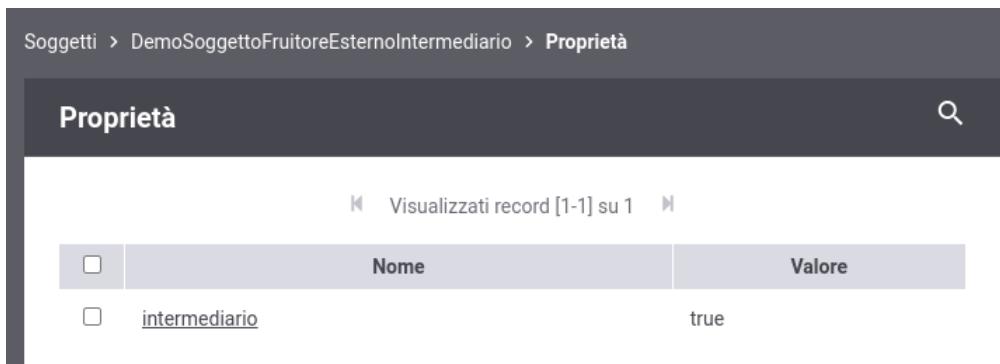
Algoritmo	RS256
Codifica Digest	Base64
HTTP Headers da firmare *	Digest x Content-Type x Content-Encoding x
Riferimento X.509	x5c (Certificate) x5t#256 (Certificate SHA-256 Thumbprint) URL (x5u)
Certificate Chain	<input type="checkbox"/>
KeyStore	Definito nella token policy
Time to Live (secondi) *	300
Indica la validità temporale, in secondi, a partire dalla data di creazione del security token	
Audience	<input type="text"/> (i)
Indica a chi è riferito il security token; se non viene fornito un valore verrà utilizzata la url del connettore	
Claims	<input type="text"/> (i)
Indicare per riga i claims (nome=valore); visualizzare 'info' per maggiori dettagli	

Figure3.128: Configurazione fruizione “ModI” con indicazione di utilizzare il keystore definito nella token policy

- *1-1 con l'applicativo fruitore*: registrazione come proprietà “purposeId” di un applicativo fruitore e riferito nella token policy tramite il valore “\${clientApplicationConfig:purposeId}”;
- *N applicativi fruitore censiti sulla fruizione*: registrazione di N proprietà “<clientApplicationName>.purposeId” sulla fruizione, una per ogni applicativo fruitore il cui nome va indicato come prefisso della proprietà; nella token policy deve essere utilizzato il valore “\${dynamicConfig:apiSearchByClientApplication(purposeId)}”;
- *N fruizioni censite sull'applicativo fruitore*: registrazione di N proprietà “<nomeApiImpl>.v<nomeApiImpl>.purposeId” sull'applicativo, una per ogni fruizione di API che l'applicativo deve fruitore il cui nome va indicato come prefisso della proprietà; nella token policy deve essere utilizzato il valore “\${dynamicConfig:clientApplicationSearch(purposeId)}”.

Intermediario

È possibile indicare un soggetto identificato sul canale come intermediario registrando una proprietà “*intermediario*” valorizzata a “*true*” tra le proprietà del soggetto come mostrato in figura Fig. 3.129.



	Nome	Valore
<input type="checkbox"/>	intermediario	true

Figure3.129: Soggetto intermediario

Questo consente di autorizzare una richiesta proveniente da un soggetto identificato sul canale (l’intermediario), che risulta differente dal soggetto a cui appartiene l’applicativo identificato tramite token di sicurezza.

Recupero informazioni client tramite API PDND fallito

È possibile modificare il comportamento di default, descritto nella sezione [API PDND](#), per far fallire la transazione in caso il recupero delle informazioni sul client o sull’organizzazione tramite [API PDND](#) fallisca.

Per attivare il fallimento della transazione è necessario registrare le seguenti *Proprietà*:

- “*pdnd.readByApiInterop.client.failed.abortTransaction*” valorizzata a “*true*” per richiedere il fallimento della transazione nel caso in cui non sia stato possibile recuperare le informazioni sul client attraverso la risorsa “GET /clients/{clientId}”.
- “*pdnd.readByApiInterop.organization.failed.abortTransaction*” valorizzata a “*true*” per richiedere il fallimento della transazione nel caso in cui non sia stato possibile recuperare le informazioni sull’organizzazione attraverso la risorsa “GET /organizations/{organizationId}”.

Politiche di Rate Limiting basate su informazioni prelevate via API PDND

La figura Fig. 3.130 mostra un esempio di politica di *Rate Limiting* che conteggia le richieste rispetto all'informazione sull'organizzazione ottenuta interrogando le API della PDND.

Valori di Soglia

Ridefinisci Valori di Soglia

Num. Massimo Richieste *

Raggruppamento

Se abilitato, il calcolo del valore della soglia avviene raggruppando le richieste in funzione dei criteri selezionati

Stato	<input type="text" value="abilitato"/>
Risorsa	<input type="checkbox"/>
Richiedente	<input type="checkbox"/>
Token	<input checked="" type="checkbox"/>
Claims *	<input type="text" value="Subject"/> <input type="text" value="ClientId"/> <input style="background-color: #f2e0d2; border: 1px solid #ccc;" type="text" value="PDND OrganizationName"/> <input type="text" value="Username"/> <input type="text" value="eMail"/>
Chiave	<input type="checkbox"/>

Figure3.130: Rate Limiting: conteggio per organizzazione ottenuta tramite API PDND

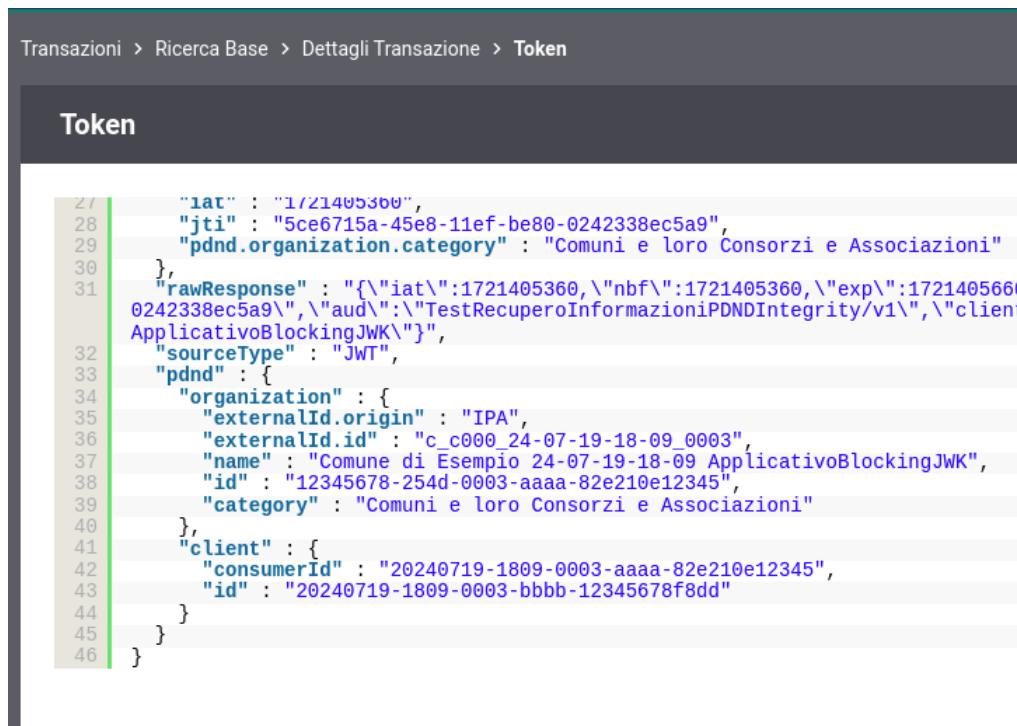
Se viene attivato il conteggio per organizzazione e l'informazione non è disponibile la transazione termina con un errore di Rate Limiting 429 dovuto all'informazione mancante. È possibile modificare il comportamento di default in modo da non far terminare con errore la transazione registrando la *Proprietà pdnd.rateLimitingByOrganization.infoNotAvailable.abortTransaction* valorizzata a *false*.

È inoltre possibile filtrare per informazioni prelevate dalla PDND, visualizzabili anche nel dettaglio di una transazione, nella voce “token” all’interno della sezione “Informazioni Mittente” di (Fig. 3.131). Nella figura Fig. 3.132 viene mostrato un esempio.

3.4 Pattern di Interazione

Le specifiche del Modello di Interoperabilità definiscono i Pattern di Interazione come le modalità secondo le quali un erogatore e un fruitore possono interagire. La distinzione operata a livello della specifica è quella tra il pattern «Bloccante» e quello «Non Bloccante». Solamente per API di tipo REST è disponibile anche un terzo pattern «Accesso CRUD» orientato alle risorse dove le API vengono utilizzate per eseguire operazioni di tipo CRUD - Create, Read, Update, Delete su risorse del dominio di interesse. Per le differenze di dettaglio tra i pattern si rimanda al testo della specifica.

Il pattern di interazione viene definito nell’interfaccia del servizio e conseguentemente GovWay recepisce tale informazioni nell’ambito della configurazione di una API nel contesto del profilo ModI.

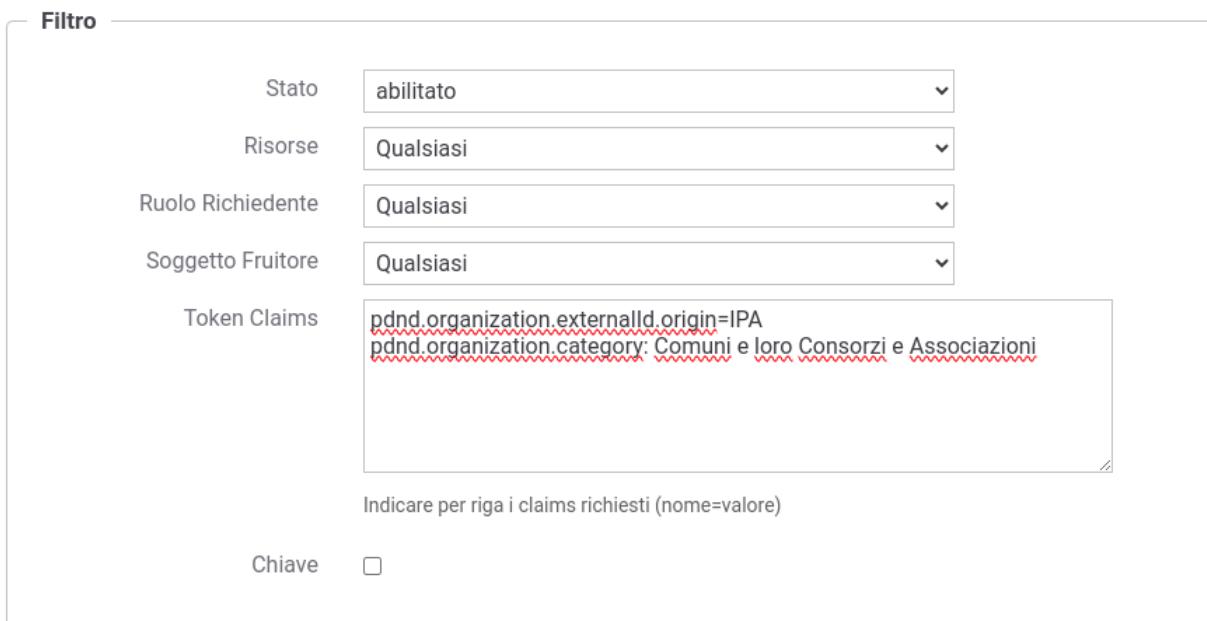


```

27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
    "iat" : "1/21405360",
    "jti" : "5ce6715a-45e8-11ef-be80-0242338ec5a9",
    "pdnd.organization.category" : "Comuni e loro Consorzi e Associazioni"
},
"rawResponse" : "{\"iat\":1721405360,\"nbf\":1721405360,\"exp\":1721405660
0242338ec5a9\", \"aud\": \"TestRecuperoInformazioniPDNDIntegrity/v1\", \"client
ApplicativoBlockingJWK\"}",
"sourceType" : "JWT",
"pdnd" : {
    "organization" : {
        "externalId.origin" : "IPA",
        "externalId.id" : "c_c000_24-07-19-18-09_0003",
        "name" : "Comune di Esempio 24-07-19-18-09 ApplicativoBlockingJWK",
        "id" : "12345678-254d-0003-aaaa-82e210e12345",
        "category" : "Comuni e loro Consorzi e Associazioni"
    },
    "client" : {
        "consumerId" : "20240719-1809-0003-aaaa-82e210e12345",
        "id" : "20240719-1809-0003-bbbb-12345678f8dd"
    }
}
}

```

Figure3.131: Informazioni ottenute tramite API PDND



Filtro

Stato	abilitato
Risorse	Qualsiasi
Ruolo Richiedente	Qualsiasi
Soggetto Fruitore	Qualsiasi
Token Claims	<p>pdnd.organization.externalId.origin=IPA pdnd.organization.category: Comuni e loro Consorzi e Associazioni</p>

Indicare per riga i claims richiesti (nome=valore)

Chiave

Figure3.132: Rate Limiting: filtro per informazioni ottenute tramite API PDND

La configurazione di API con il profilo ModI produce per default servizi con pattern di interazione «Bloccante» su API di tipo SOAP e «Accesso CRUD» su API di tipo REST. Se si desidera, è possibile modificare questa impostazione intervenendo puntualmente sulle singole operation/risorse della API.

La maschera di editing della singola operation/risorsa possiede la sezione ModI per consentire di specificare le seguenti informazioni (Fig. 3.133):

- *Interazione*: specifica il pattern di interazione che si vuole associare alla specifica operation/risorsa
 - *Pattern*: indica il nome del pattern di interazione, a scelta tra Bloccante e Non Bloccante
 - *Tipo*: (solo per il pattern non bloccante) indica se l'interazione prevista è di tipo PUSH (iniziativa del mittente) o PULL (iniziativa del destinatario)
 - *Funzione*: (solo per il pattern non bloccante) indica se l'operation/risorsa ha la funzione di inviare una richiesta, chiedere lo stato di avanzamento dell'elaborazione della risposta o inviare una risposta.
 - *Richiesta Correlata*: (solo per la funzione Richiesta Stato e Risposta) indica l'operation/risorsa correlata che corrisponde all'invio della richiesta.

The screenshot shows a configuration dialog for 'ModI'. At the top, there is a tab labeled 'ModI'. Below it, under the heading 'Interazione', there are three dropdown menus: 'Pattern' (set to 'Non Bloccante'), 'Tipo' (set to 'PULL'), and 'Funzione' (set to 'Richiesta'). Under the heading 'Sicurezza Messaggio', there is a single dropdown menu for 'Pattern' (set to 'Usa pattern API').

Figure3.133: Pattern di interazione ModI per operation/risorse dell'API

Nota: Su API di tipo REST i pattern bloccanti e non bloccanti risultano selezionabili solamente se una risorsa è compatibile con i metodi HTTP e i codici di risposta richiesti dalla specifica.

Nelle sezioni seguenti vengono forniti maggiori dettagli su come siano gestiti i pattern non bloccanti.

3.4.1 Pattern di Interazione PUSH per API SOAP

Il pattern di interazione, denominato PUSH, è utilizzabile nel caso in cui il fruitore abbia a sua volta la possibilità di esporre una interfaccia di servizio per la ricezione delle risposte (Fig. 3.134).

Come riportato dalle Linee Guida di Interoperabilità ModI:

- Al passo (1), il fruitore DEVE indicare l'endpoint della callback utilizzando l'header SOAP custom “X-ReplyTo”;
- Al passo (2), l'erogatore DEVE fornire insieme all'acknowledgement della richiesta nel body, il correlation ID utilizzando l'header SOAP custom X-Correlation-ID;
- Al passo (3), l'erogatore DEVE riutilizzare lo stesso correlation ID fornito al passo (2) sempre utilizzando l'header SOAP custom X-Correlation-ID;

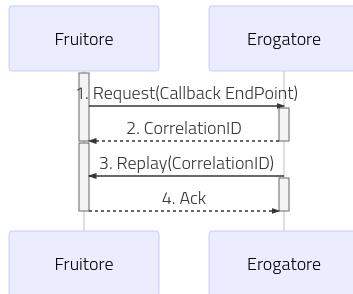


Figure3.134: Flusso previsto in un Pattern di Interazione PUSH

- Al passo (4), il fruitore DEVE riconoscere tramite un messaggio di acknowledgement il ricevimento della risposta.

Configurazione delle API

Per attuare la configurazione su GovWay si deve procedere con la registrazione delle due API che definiscono il servizio di ricezione della richiesta e il servizio di Callback dove l'erogatore deve inoltrare la risposta.

- Richiesta

Effettuata la registrazione delle API, accedere al dettaglio dell'azione relativa al servizio di richiesta ed impostare nella sezione “ModI” un pattern di interazione non bloccante “PUSH” con ruolo “Richiesta” come mostrato nella figura Fig. 3.135:

Figure3.135: Configurazione della richiesta dell'API SOAP (PUSH)

- Risposta

Successivamente, accedere al dettaglio dell'azione relativa al servizio di callback ed impostare nella sezione “ModI” un pattern di interazione non bloccante “PUSH” con ruolo “Risposta”. Definire anche la correlazione verso il servizio e l'azione relativa alla richiesta come mostrato nella figura Fig. 3.136:

The screenshot shows the configuration interface for an 'MRequestResponse' action. The 'Azione' section has 'Nome' set to 'MRequestResponse'. The 'Informazioni Protocollo' section has 'Profilo' set to 'usa profilo servizio'. The 'Modi PA' section contains the following configuration:

- Profilo Interazione: Profilo Non Bloccante, Interazione PUSH, Funzione Risposta, API Richiesta Correlata SOAPBlockingPUSHRequest v1, Servizio SOAPCallback, Azione MRequest.
- Profilo Sicurezza Messaggio: Profilo Usa profilo API.

Figure3.136: Configurazione della risposta dell'API SOAP (PUSH)

Configurazione dell'Erogazione

Sul dominio dell'erogatore deve essere definita sia un'erogazione dell'API relativa al servizio di richiesta che una fruizione del servizio di callback.

- Erogazione del Servizio di Richiesta

Le richieste ricevute sull'erogazione vengono validate da GovWay verificando la presenza dell'header SOAP custom “X-ReplyTo” come previsto dal profilo “ModI”. Effettuata la validazione del messaggio di richiesta, eventualmente gestendo anche gli aspetti di sicurezza descritti nella sezione *Sicurezza Messaggio*, GovWay inoltra il messaggio al backend e rimane in attesa dell'acknowledgement.

Ricevuto il messaggio di acknowledgement GovWay verifica la presenza dell'header SOAP custom “X-Correlation-ID”. Se tale header non risulta presente viene generato da GovWay impostando come valore l'identificativo della transazione, che è stato inoltrato con la richiesta al backend tramite gli header di integrazione descritti nella sezione *Scambio di informazioni nella richiesta inoltrata dal gateway al server* e *Altri header di Integrazione* (per default tramite l'header http “GovWay-Transaction-ID”).

Nota: Header “X-Correlation-ID” generato da GovWay

La generazione dell'header soap “X-Correlation-ID”, se non presente, è disabilitabile intervenendo sulla proprietà «org.openspcoop2.protocol.modipa.soap.push.request.correlationId.header.useTransactionIdIfNotExist» presente nel file “/etc/govway/modipa_local.properties” (si assume che “/etc/govway” sia la directory di configurazione indicata in fase di installazione). Se si disabilita la proprietà, GovWay termina con errore la transazione se rileva l'assenza dell'header soap “X-Correlation-ID” nel messaggio di acknowledgement ricevuto dal backend.

- Fruizione del Servizio di Callback per la Risposta

Le risposte devono essere inoltrate dall'applicativo mittente utilizzando la fruizione del servizio di Callback configurata

su GovWay. Le risposte vengono validate da GovWay verificando la presenza dell'header SOAP custom “X-Correlation-ID”. GovWay permette di fornire l'informazione sull'identificativo di correlazione anche tramite modalità alternative all'header soap (header http, parametri della url...) per poi generare un header soap “X-Correlation-ID” come previsto dalla specifica “ModI” valorizzato con il valore fornito. Le modalità alternative sono le seguenti:

- Header HTTP “X-Correlation-ID”
- Header HTTP “GovWay-Conversation-ID” o parametro della url “govway_conversation_id” previsto per la correlazione tramite identificativo di collaborazione descritta nella sezione *Correlazione tra transazioni differenti*. Questa modalità richiede che sia abilitata l'indicazione dell'identificativo di collaborazione nell'API o sulla singola azione come mostrato nella seguente Fig. 3.137:

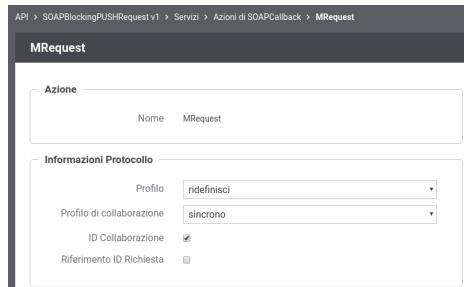


Figure3.137: Abilitazione funzionalità di correlazione govway tramite identificativo di collaborazione

- Header HTTP “GovWay-Relates-To” o parametro della url “govway_relates_to” previsto per la correlazione tramite riferimento della richiesta descritta nella sezione *Correlazione tra transazioni differenti*. Questa modalità richiede che sia abilitata l'indicazione dell'identificativo di riferimento alla richiesta nell'API o sulla singola azione come mostrato nella figura Fig. 3.138:

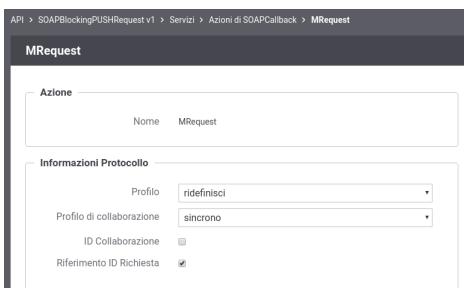


Figure3.138: Abilitazione funzionalità di correlazione govway tramite identificativo della richiesta

Configurazione della Fruizione

Sul dominio del fruitore deve essere definita sia una fruizione dell'API relativa al servizio di richiesta che un'erogazione del servizio di callback.

- Fruizione del Servizio di Richiesta

Le richieste devono essere inoltrate dall'applicativo mittente utilizzando la fruizione del servizio di richiesta configurata su GovWay. Su ogni richiesta GovWay crea, o ne modifica il valore se già presente, dell'header soap “X-ReplyTo” previsto dal profilo “ModI”. L'header viene valorizzato con l'url di invocazione utilizzabile dalla controparte per invocare il servizio di callback configurato su GovWay.

Nota: Header “X-ReplyTo” generato da GovWay

La valorizzazione dell'header soap “X-ReplyTo” da parte di GovWay è disabilitabile intervenendo sulla proprietà «org.openscoop2.protocol.modipa.soap.push.replyTo.header.updateOrCreate» presente nel file “/etc/govway/modipa_local.properties” (si assume che “/etc/govway” sia la directory di configurazione indicata in fase di installazione). Se si disabilita la proprietà, GovWay termina con errore la transazione se rileva l'assenza dell'header soap “X-ReplyTo” nel messaggio di richiesta ricevuto dal backend.

Il messaggio di acknowledgement ricevuto viene anch'esso validato al fine di verificare la presenza dell'header soap “X-Correlation-ID” come previsto dalla specifica “ModI”. L'informazione sull'id di correlazione è ottenibile dall'applicativo mittente sulla risposta, oltre che tramite l'header soap “X-Correlation-ID”, anche tramite gli header di integrazione descritti nella sezione *Scambio di informazioni nella richiesta inoltrata dal gateway al server* e *Altri header di Integrazione* (per default tramite l'header http “GovWay-Conversation-ID”).

- Erogazione del Servizio di Callback per la Risposta

Le risposte ricevute sull'erogazione del servizio di Callback vengono validate da GovWay verificando la presenza dell'header SOAP custom “X-Correlation-ID” come previsto dal profilo “ModI”. Effettuata la validazione del messaggio di risposta, eventualmente gestendo anche gli aspetti di sicurezza descritti nella sezione *Sicurezza Messaggio*, GovWay inoltra il messaggio al backend e rimane in attesa dell'acknowledgement. L'informazione sull'id di correlazione è inoltrato al backend, oltre che tramite l'header soap “X-Correlation-ID”, anche tramite gli header di integrazione descritti nella sezione *Scambio di informazioni nella richiesta inoltrata dal gateway al server* e *Altri header di Integrazione* (per default tramite l'header http “GovWay-Conversation-ID”).

3.4.2 Pattern di Interazione PUSH per API REST

Il pattern di interazione, denominato PUSH, è utilizzabile nel caso in cui il fruitore abbia a sua volta la possibilità di esporre una interfaccia di servizio per la ricezione delle risposte (Fig. 3.139).

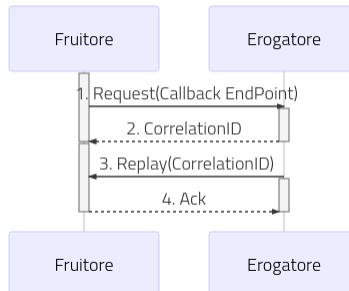


Figure3.139: Flusso previsto in un Pattern di Interazione PUSH

Come riportato dalle Linee Guida di Interoperabilità ModI:

- Al passo (1), il fruitore DEVE indicare l'endpoint della callback utilizzando l'header HTTP custom X-ReplyTo ed usando HTTP method POST;
- Al passo (2), l'erogatore DEVE fornire insieme all'acknowledgement della richiesta nel body, il correlation ID utilizzando l'header HTTP custom X-Correlation-ID; Il codice HTTP di stato DEVE essere HTTP status 202 Accepted a meno che non si verifichino errori;
- Al passo (3), l'erogatore DEVE riutilizzare lo stesso correlation ID fornito al passo (2) sempre utilizzando l'header HTTP custom X-Correlation-ID; Il verbo HTTP utilizzato deve essere POST;
- Al passo (4), il fruitore DEVE riconoscere tramite un messaggio di acknowledgement il ricevimento della risposta; Il codice HTTP di stato DEVE essere HTTP status 200 OK a meno che non si verifichino errori.

Configurazione delle API

Per attuare la configurazione su GovWay si deve procedere con la registrazione delle due API che definiscono il servizio di ricezione della richiesta e il servizio di Callback dove l'erogatore deve inoltrare la risposta.

- Richiesta

Effettuata la registrazione delle API, accedere al dettaglio della risorsa relativa al servizio di richiesta ed impostare nella sezione “ModI” un pattern di interazione non bloccante “PUSH” con ruolo “Richiesta” come mostrato nella figura [Fig. 3.140](#):

The screenshot shows the configuration interface for a REST API. At the top, the URL is API > RESTBlockingPUSHRequest v1 > Risorse > POST /resources/{id_resource}/M. The main title is POST /resources/{id_resource}/M. A note says "Note: (*) Campi obbligatori". The "Risorsa" section contains fields for HTTP Method (POST), Path (*/resources/{id_resource}/M), and Nome (POST_resources.id_resource.M). The "Modi PA" section contains fields for Profilo Interazione (Non Bloccante), Interazione (PUSH), and Funzione (Richiesta). The "Informazioni Protocollo" section contains fields for ID Collaborazione and Riferimento ID Richiesta.

Figure3.140: Configurazione della richiesta dell'API REST (PUSH)

- Risposta

Successivamente, accedere al dettaglio della risorsa relativa al servizio di callback ed impostare nella sezione “ModI” un pattern di interazione non bloccante “PUSH” con ruolo “Risposta”. Definire anche la correlazione verso l'API e l'azione relativa alla richiesta come mostrato nella figura [Fig. 3.141](#):

Configurazione dell'Erogazione

Sul dominio dell'erogatore deve essere definita sia un'erogazione dell'API relativa al servizio di richiesta che una fruizione del servizio di callback.

- Erogazione del Servizio di Richiesta

Le richieste ricevute sull'erogazione vengono validate da GovWay verificando la presenza dell'header HTTP custom “X-ReplyTo” come previsto dal profilo “ModI”. Effettuata la validazione del messaggio di richiesta, eventualmente

The screenshot shows the configuration interface for a REST API response. The top navigation bar indicates the path: API > RESTBlockingPUSHResponse v1 > Risorse > POST /MResponse. The main section is titled 'POST /MResponse'. A note at the top says 'Note: (*) Campi obbligatori'. The 'Risorsa' section contains fields for 'HTTP Method' (set to 'POST'), 'Path' (set to '/MResponse'), and 'Nome' (set to 'POST_MResponse'). A sub-note says 'Se non definito verrà automaticamente generato un identificativo univoco'. The 'Informazioni Protocollo' section includes 'ID Collaborazione' and 'Riferimento ID Richiesta'. The 'Modi PA' section contains a 'Profilo Interazione' group with dropdowns for 'Profilo' (Non Bloccante), 'Interazione' (PUSH), 'Funzione' (Risposta), 'API Richiesta Correlata' (RESTBlockingPUSHRequest v1), and 'Risorsa' (POST /resources/{id_resource}/M). Below this is a 'Profilo Sicurezza Messaggio' group with a dropdown for 'Profilo' (Usa profilo API).

Figure3.141: Configurazione della risposta dell'API REST (PUSH)

gestendo anche gli aspetti di sicurezza descritti nella sezione *Sicurezza Messaggio*, GovWay inoltra il messaggio al backend e rimane in attesa dell'acknowledgement.

Ricevuto il messaggio di acknowledgement GovWay verifica che il codice HTTP di stato sia 202 e verifica la presenza dell'header HTTP custom “X-Correlation-ID”. Se tale header non risulta presente viene generato da GovWay impostando come valore l'identificativo della transazione, che è stato inoltrato con la richiesta al backend tramite gli header di integrazione descritti nella sezione *Scambio di informazioni nella richiesta inoltrata dal gateway al server* e *Altri header di Integrazione* (per default tramite l'header http “GovWay-Transaction-ID”).

Nota: Header “X-Correlation-ID” generato da GovWay

La generazione dell'header HTTP “X-Correlation-ID”, se non presente, è disabilitabile intervenendo sulla proprietà «org.openspcoop2.protocol.modipa.rest.push.request.correlationId.header.useTransactionIdIfNotExists» presente nel file “/etc/govway/modipa_local.properties” (si assume che “/etc/govway” sia la directory di configurazione indicata in fase di installazione). Se si disabilita la proprietà, GovWay termina con errore la transazione se rileva l'assenza dell'header HTTP “X-Correlation-ID” nel messaggio di acknowledgement ricevuto dal backend.

- Fruizione del Servizio di Callback per la Risposta

Le risposte devono essere inoltrate dall'applicativo mittente utilizzando la fruizione del servizio di Callback configurata su GovWay. Le risposte vengono validate da GovWay verificando la presenza dell'header HTTP custom “X-Correlation-ID”. GovWay permette di fornire l'informazione sull'identificativo di correlazione anche tramite modalità alternative all'header HTTP custom per poi generarlo come previsto dalla specifica “ModI” valorizzato con il valore fornito. Le modalità alternative sono le seguenti:

- Header HTTP “GovWay-Conversation-ID” o parametro della url “govway_conversation_id” previsto per la correlazione tramite identificativo di collaborazione descritta nella sezione *Correlazione tra transazioni*

differenti. Questa modalità richiede che sia abilitata l'indicazione dell'identificativo di collaborazione nell'API o sulla singola azione come mostrato nella figura Fig. 3.142:

API > RESTBlockingPUSHRequest v1 > Risorse > POST /resources/{id_resource}/M

POST /resources/{id_resource}/M

Note: (*) Campi obbligatori

Risorsa

HTTP Method: POST
Path: /resources/{id_resource}/M
Nome: POST_resources.id_resource.M
Se non definito verrà automaticamente generato un identificativo univoco
Descrizione:

Informazioni Protocollo

ID Collaborazione:
Riferimento ID Richiesta:

Figure3.142: Abilitazione funzionalità di correlazione govway tramite identificativo di collaborazione

- Header HTTP “GovWay-Relates-To” o parametro della url “govway_relates_to” previsto per la correlazione tramite riferimento della richiesta descritta nella sezione *Correlazione tra transazioni differenti*. Questa modalità richiede che sia abilitata l'indicazione dell'identificativo di riferimento alla richiesta nell'API o sulla singola azione come mostrato nella figura Fig. 3.143:

API > RESTBlockingPUSHRequest v1 > Risorse > POST /resources/{id_resource}/M

POST /resources/{id_resource}/M

Note: (*) Campi obbligatori

Risorsa

HTTP Method: POST
Path: /resources/{id_resource}/M
Nome: POST_resources.id_resource.M
Se non definito verrà automaticamente generato un identificativo univoco
Descrizione:

Informazioni Protocollo

ID Collaborazione:
Riferimento ID Richiesta:

Figure3.143: Abilitazione funzionalità di correlazione govway tramite identificativo della richiesta

Configurazione della Fruizione

Sul dominio del fruitore deve essere definita sia una fruizione dell'API relativa al servizio di richiesta che un'erogazione del servizio di callback.

- Fruizione del Servizio di Richiesta

Le richieste devono essere inoltrate dall'applicativo mittente utilizzando la fruizione del servizio di richiesta configurata su GovWay. Su ogni richiesta GovWay crea, o ne modifica il valore se già presente, dell'header HTTP “X-ReplyTo” previsto dal profilo “ModI”. L'header viene valorizzato con l'url di invocazione utilizzabile dalla controparte per invocare il servizio di callback configurato su GovWay.

Nota: Header “X-ReplyTo” generato da GovWay

La valorizzazione dell'header HTTP "X-ReplyTo" da parte di GovWay è disabilitabile intervenendo sulla proprietà «org.openspcoop2.protocol.modipa.rest.push.replyTo.header.updateOrCreate» presente nel file "/etc/govway/modipa_local.properties" (si assume che "/etc/govway" sia la directory di configurazione indicata in fase di installazione). Se si disabilita la proprietà, GovWay termina con errore la transazione se rileva l'assenza dell'header HTTP "X-ReplyTo" nel messaggio di richiesta ricevuto dal backend.

Il messaggio di acknowledgement ricevuto viene anch'esso validato al fine di verificare la presenza dell'header HTTP "X-Correlation-ID" come previsto dalla specifica "ModI". L'informazione sull'id di correlazione è ottenibile dall'applicativo mittente sulla risposta, oltre che tramite l'header HTTP "X-Correlation-ID", anche tramite gli header di integrazione descritti nella sezione *Scambio di informazioni nella richiesta inoltrata dal gateway al server* e *Altri header di Integrazione* (per default tramite l'header http "GovWay-Conversation-ID").

- Erogazione del Servizio di Callback per la Risposta

Le risposte ricevute sull'erogazione del servizio di Callback vengono validate da GovWay verificando la presenza dell'header HTTP custom "X-Correlation-ID" come previsto dal profilo "ModI". Effettuata la validazione del messaggio di risposta, eventualmente gestendo anche gli aspetti di sicurezza descritti nella sezione *Sicurezza Messaggio*, GovWay inoltra il messaggio al backend e rimane in attesa dell'acknowledgement. L'informazione sull'id di correlazione è inoltrato al backend, oltre che tramite l'header HTTP "X-Correlation-ID", anche tramite gli header di integrazione descritti nella sezione *Scambio di informazioni nella richiesta inoltrata dal gateway al server* e *Altri header di Integrazione* (per default tramite l'header http "GovWay-Conversation-ID").

3.4.3 Pattern di Interazione PULL per API SOAP

Il pattern di interazione, denominato PULL, prevede che il fruitore non fornisca un indirizzo di callback, mentre l'erogatore fornisce un indirizzo interrogabile per verificare lo stato di processamento di una richiesta e, al fine dell'elaborazione della stessa, il risultato (Fig. 3.144).

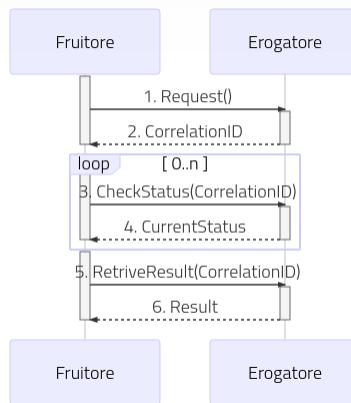


Figure3.144: Flusso previsto in un Pattern di Interazione PULL per API SOAP

Come riportato dalle Linee Guida di Interoperabilità ModI:

- L'interfaccia di servizio dell'erogatore fornisce tre metodi differenti al fine di inoltrare una richiesta, controllarne lo stato ed ottenerne il risultato
- Al passo (1), il fruitore effettua una richiesta;
- Al passo (2), l'erogatore DEVE fornire insieme all'acknowledgement della richiesta nel body, un correlation ID riportato nel header custom SOAP X-Correlation-ID;

- Al passo (3), il fruitore DEVE utilizzare il correlation ID ottenuto al passo (2) per richiedere lo stato di processamento di una specifica richiesta;
- Al passo (4) l'erogatore, quando il processamento non si è ancora concluso fornisce informazioni circa lo stato della lavorazione della richiesta, quando invece il processamento si è concluso risponde indicando in maniera esplicita il completamento;
- Al passo (5), il fruitore utilizza il correlation ID di cui al passo (2) al fine di richiedere il risultato della richiesta;
- Al passo (6), l'erogatore fornisce il risultato del processamento.

Configurazione delle API

Per attuare la configurazione su GovWay si deve procedere con la registrazione dell'API che deve contenere i tre metodi differenti descritti precedentemente.

- Richiesta

Effettuata la registrazione delle API, accedere al dettaglio dell'azione corrispondente alla richiesta ed impostare nella sezione “ModI” un pattern di interazione non bloccante “PULL” con ruolo “Richiesta” come mostrato nella figura [Fig. 3.145](#):

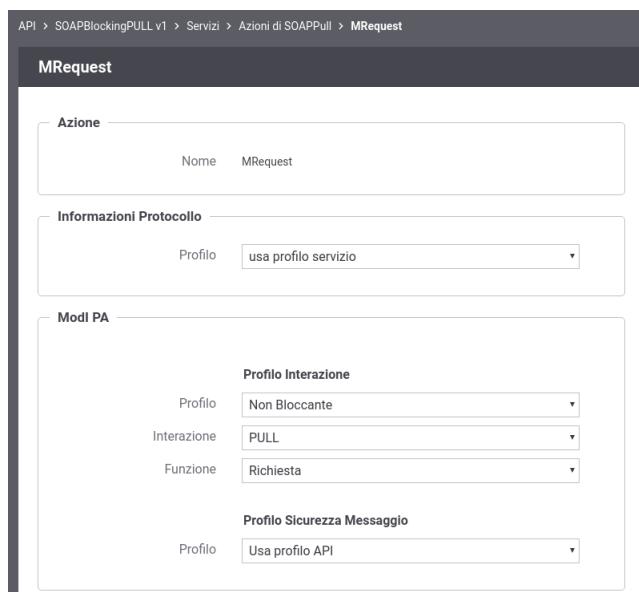


Figure3.145: Configurazione della richiesta dell'API SOAP (PULL)

- Richiesta Stato

Successivamente, accedere al dettaglio dell'azione che consente di richiedere lo stato di processamento ed impostare nella sezione “ModI” un pattern di interazione non bloccante “PULL” con ruolo “Richiesta Stato”. Definire anche la correlazione verso l'azione relativa alla richiesta come mostrato nella figura [Fig. 3.146](#):

- Risposta

Accedere al dettaglio dell'azione corrispondente alla risposta ed impostare nella sezione “ModI” un pattern di interazione non bloccante “PULL” con ruolo “Risposta”. Definire anche la correlazione verso l'azione relativa alla richiesta come mostrato nella figura [Fig. 3.147](#):

API > SOAPBlockingPULL v1 > Servizi > Azioni di SOAPPull > MProcessingStatus

MProcessingStatus

Azione

Nome: MProcessingStatus

Informazioni Protocollo

Profilo: usa profilo servizio

Modi PA

Profilo Interazione

Profilo: Non Bloccante

Interazione: PULL

Funzione: Richiesta Stato

Richiesta Correlata: MRequest

Profilo Sicurezza Messaggio

Profilo: Usa profilo API

Figure3.146: Configurazione della richiesta stato di processamento dell'API SOAP (PULL)

API > SOAPBlockingPULL v1 > Servizi > Azioni di SOAPPull > MResponse

MResponse

Azione

Nome: MResponse

Informazioni Protocollo

Profilo: usa profilo servizio

Modi PA

Profilo Interazione

Profilo: Non Bloccante

Interazione: PULL

Funzione: Risposta

Richiesta Correlata: MRequest

Profilo Sicurezza Messaggio

Profilo: Usa profilo API

Figure3.147: Configurazione della risposta dell'API SOAP (PUSH)

Configurazione dell'Erogazione

Sul dominio dell'erogatore deve essere definita l'erogazione dell'API.

- Richiesta

Le richieste ricevute sull'erogazione vengono inoltrate al backend da GovWay rimanendo poi in attesa dell'acknowledgement.

Ricevuto il messaggio di acknowledgement GovWay verifica la presenza dell'header SOAP custom “X-Correlation-ID”. Se tale header non risulta presente viene generato da GovWay impostando come valore l'identificativo della transazione, che è stato inoltrato con la richiesta al backend tramite gli header di integrazione descritti nella sezione *Scambio di informazioni nella richiesta inoltrata dal gateway al server* e *Altri header di Integrazione* (per default tramite l'header http “GovWay-Transaction-ID”).

Nota: Header “X-Correlation-ID” generato da GovWay

La generazione dell'header soap “X-Correlation-ID”, se non presente, è disabilitabile intervenendo sulla proprietà «org.openspcoop2.protocol.modipa.soap.pull.request.correlationId.header.useTransactionIdIfNotExists» presente nel file “/etc/govway/modipa_local.properties” (si assume che “/etc/govway” sia la directory di configurazione indicata in fase di installazione). Se si disabilita la proprietà, GovWay termina con errore la transazione se rileva l'assenza dell'header soap “X-Correlation-ID” nel messaggio di acknowledgement ricevuto dal backend.

- Richiesta Stato di Processamento

Le richieste che richiedono uno stato del processamento vengono validate da GovWay verificando la presenza dell'header soap “X-Correlation-ID” come previsto dal profilo “ModI”. Effettuata la validazione del messaggio di richiesta, eventualmente gestendo anche gli aspetti di sicurezza descritti nella sezione *Sicurezza Messaggio*, GovWay inoltra il messaggio al backend e rimane in attesa dell'acknowledgement. L'informazione sull'id di correlazione è inoltrato al backend, oltre che tramite l'header soap “X-Correlation-ID”, anche tramite gli header di integrazione descritti nella sezione *Scambio di informazioni nella richiesta inoltrata dal gateway al server* e *Altri header di Integrazione* (per default tramite l'header http “GovWay-Conversation-ID”).

- Risposta

Le risposte vengono gestite da GovWay in maniera simile a quanto indicato per le richieste di stato del processamento.

Configurazione della Fruizione

Sul dominio del fruitore deve essere definita una fruizione dell'API.

- Richiesta

Le richieste devono essere inoltrate dall'applicativo mittente utilizzando la fruizione dell'API configurata su GovWay.

Il messaggio di acknowledgement ricevuto viene validato al fine di verificare la presenza dell'header soap “X-Correlation-ID” come previsto dalla specifica “ModI”. L'informazione sull'id di correlazione è ottenibile dall'applicativo mittente sulla risposta, oltre che tramite l'header soap “X-Correlation-ID”, anche tramite gli header di integrazione descritti nella sezione *Scambio di informazioni nella richiesta inoltrata dal gateway al server* e *Altri header di Integrazione* (per default tramite l'header http “GovWay-Conversation-ID”).

- Richiesta Stato di Processamento

Le richieste che richiedono uno stato del processamento devono essere inoltrate dall'applicativo mittente utilizzando la fruizione dell'API configurata su GovWay. Le richieste vengono validate da GovWay verificando la presenza dell'header soap “X-Correlation-ID”. GovWay permette di fornire l'informazione sull'identificativo di correlazione anche tramite modalità alternative all'header soap per poi generarlo come previsto dalla specifica “ModI” valorizzato con il valore fornito. Le modalità alternative sono le seguenti:

- Header HTTP “X-Correlation-ID”

- Header HTTP “GovWay-Conversation-ID” o parametro della url “govway_conversation_id” previsto per la correlazione tramite identificativo di collaborazione descritta nella sezione *Correlazione tra transazioni differenti*. Questa modalità richiede che sia abilitata l’indicazione dell’identificativo di collaborazione nell’API o sulla singola azione come mostrato nella figura Fig. 3.148:

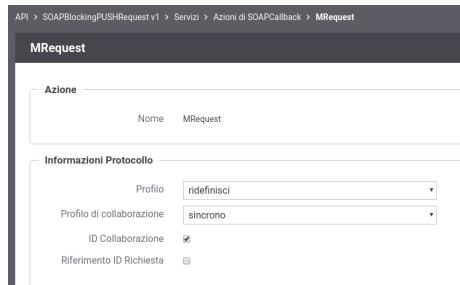


Figure3.148: Abilitazione funzionalità di correlazione govway tramite identificativo di collaborazione

- Header HTTP “GovWay-Relates-To” o parametro della url “govway_relates_to” previsto per la correlazione tramite riferimento della richiesta descritta nella sezione *Correlazione tra transazioni differenti*. Questa modalità richiede che sia abilitata l’indicazione dell’identificativo di riferimento alla richiesta nell’API o sulla singola azione come mostrato nella figura Fig. 3.149:

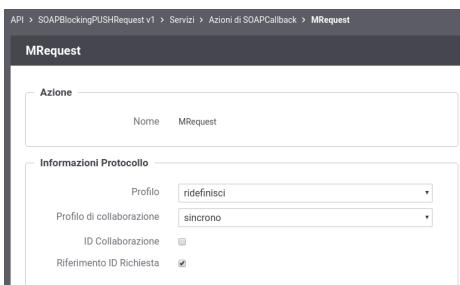


Figure3.149: Abilitazione funzionalità di correlazione govway tramite identificativo della richiesta

- Risposta

Le risposte vengono gestite da GovWay in maniera simile a quanto indicato per le richieste di stato del processamento.

3.4.4 Pattern di Interazione PULL per API REST

Il pattern di interazione, denominato PULL, prevede che il fruitore non fornisca un indirizzo di callback, mentre l’erogatore fornisce un indirizzo interrogabile per verificare lo stato di processamento di una richiesta e, al fine dell’elaborazione della stessa, il risultato (Fig. 3.150).

Come riportato dalle Linee Guida di Interoperabilità ModI:

- L’interfaccia di servizio dell’erogatore fornisce tre metodi differenti al fine di inoltrare una richiesta, controllarne lo stato ed ottenerne il risultato
- Al passo (1), il fruitore DEVE utilizzare il verbo HTTP POST;

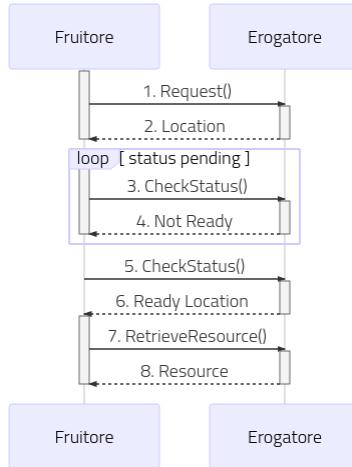


Figure3.150: Flusso previsto in un Pattern di Interazione PULL per API REST

- Al passo (2), l’erogatore DEVE fornire insieme all’acknowledgement della richiesta, un percorso di risorsa per interrogare lo stato di processamento utilizzando HTTP header Location ; Il codice HTTP di stato DEVE essere HTTP status 202 Accepted a meno che non si verifichino errori;
- Al passo (3), il fruitore DEVE utilizzare il percorso di cui al passo (2) per richiedere lo stato della risorsa; Il verbo HTTP utilizzato deve essere GET;
- Al passo (4) l’erogatore indica che la risorsa non è ancora pronta, fornendo informazioni circa lo stato della lavorazione della richiesta; il codice HTTP restituito è HTTP status 200 OK;
- Al passo (6) l’erogatore indica che la risorsa è pronta, utilizzando HTTP header Location ; per indicare il percorso dove recuperare la risorsa, il codice HTTP restituito è HTTP status 303 See Other;
- Al passo (8) l’erogatore risponde con la rappresentazione della risorsa,Il codice HTTP restituito è HTTP status 200 OK;

Configurazione delle API

Per attuare la configurazione su GovWay si deve procedere con la registrazione dell’API che deve contenere le tre risorse differenti descritti precedentemente.

- Richiesta

Effettuata la registrazione delle API, accedere al dettaglio della risorsa corrispondente alla richiesta ed impostare nella sezione “ModI” un pattern di interazione non bloccante “PULL” con ruolo “Richiesta” come mostrato nella figura Fig. 3.151:

- Richiesta Stato

Successivamente, accedere al dettaglio dell’azione che consente di richiedere lo stato di processamento ed impostare nella sezione “ModI” un pattern di interazione non bloccante “PULL” con ruolo “Richiesta Stato”. Definire anche la correlazione verso la risorsa relativa alla richiesta come mostrato nella figura Fig. 3.152:

- Risposta

Accedere al dettaglio dell’azione corrispondente alla risposta ed impostare nella sezione “ModI” un pattern di interazione non bloccante “PULL” con ruolo “Risposta”. Definire anche la correlazione verso la risorsa relativa alla richiesta come mostrato nella figura Fig. 3.153:

API > RESTBlockingPULL v1 > Risorse > **POST /tasks/queue**

POST /tasks/queue

Note: (*) Campi obbligatori

Risorsa

HTTP Method: POST
Path: /tasks/queue
Nome: POST_tasks.queue
Se non definito verrà automaticamente generato un identificativo univoco
Descrizione:

Informazioni Protocollo

ID Collaborazione:
Riferimento ID Richiesta:

Modi PA

Profilo Interazione
Profilo: Non Bloccante
Interazione: PULL
Funzione: Richiesta

Profilo Sicurezza Messaggio
Profilo: Usa profilo API

Figure3.151: Configurazione della richiesta dell'API REST (PULL)

API > RESTBlockingPULL v1 > Risorse > **GET /tasks/queue/{id_task}/**

GET /tasks/queue/{id_task}/

Note: (*) Campi obbligatori

Risorsa

HTTP Method: GET
Path: /tasks/queue/{id_task}/
Nome: GET_tasks.queue.id_task
Se non definito verrà automaticamente generato un identificativo univoco
Descrizione:

Informazioni Protocollo

ID Collaborazione:
Riferimento ID Richiesta:

Modi PA

Profilo Interazione
Profilo: Non Bloccante
Interazione: PULL
Funzione: Richiesta Stato
Richiesta Correlata: POST /tasks/queue

Profilo Sicurezza Messaggio
Profilo: Usa profilo API

Figure3.152: Configurazione della richiesta stato di processamento dell'API REST (PULL)

The screenshot shows the configuration interface for a REST API endpoint. The endpoint is named "GET /tasks/result/{id_task}" with a "GET" method and path "/tasks/result/{id_task}". The configuration includes sections for "Risorsa", "Informazioni Protocollo", and "Modi PA".

- Risorsa:** HTTP Method: GET, Path: /tasks/result/{id_task}/, Name: GET_tasks.result.id_task. A note says: "Se non definito verrà automaticamente generato un identificativo univoco".
- Informazioni Protocollo:** ID Collaborazione: [checkbox], Riferimento ID Richiesta: [checkbox].
- Modi PA:** Profilo Interazione: Profilo: Non Bloccante, Interazione: PULL, Funzione: Risposta, Richiesta Correlata: POST /tasks/queue. Profilo Sicurezza Messaggio: Profilo: Usa profilo API.

Figure3.153: Configurazione della risposta dell'API REST (PUSH)

Configurazione dell'Erogazione

Sul dominio dell'erogatore deve essere definita l'erogazione dell'API.

- Richiesta

Le richieste ricevute sull'erogazione vengono inoltrate al backend da GovWay rimanendo poi in attesa dell'acknowledgement.

Ricevuto il messaggio di acknowledgement GovWay verifica che il codice HTTP di stato sia 202 e verifica la presenza dell'header HTTP "Location".

- Richiesta Stato di Processamento

Le richieste che richiedono uno stato del processamento vengono validate da GovWay verificando che il codice HTTP di stato sia 200 (risposta non ancora pronta) o 303 (risposta pronta ad essere recuperata). Nel caso il codice HTTP sia 303 viene anche verificata la presenza dell'header HTTP "Location".

- Risposta

GovWay valida le risposte verificando che il codice HTTP di stato sia 200.

Nota: Id Correlazione

GovWay estrae dal valore presente nell'header "Location" (per la richiesta e la richiesta stato) e dall'endpoint (per la risposta) l'identificativo di correlazione al fine di correlare la richiesta con le successive operazioni.

Configurazione della Fruizione

Sul dominio del fruitore deve essere definita una fruizione dell'API.

- Richiesta

Le richieste devono essere inoltrate dall'applicativo mittente utilizzando la fruizione dell'API configurata su GovWay.

Il messaggio di acknowledgement ricevuto viene validato al fine di verificare la presenza dell'header http "Location" come previsto dalla specifica "ModI". L'informazione sull'id di correlazione è ottenibile dall'applicativo mittente sulla risposta tramite gli header di integrazione descritti nella sezione [Scambio di informazioni nella richiesta inoltrata dal gateway al server](#) e [Altri header di Integrazione](#) (per default tramite l'header http "GovWay-Conversation-ID").

- Richiesta Stato di Processamento e Risposta

Le successive operazioni devono essere inoltrate dall'applicativo mittente utilizzando la fruizione dell'API configurata su GovWay.

Nota: Id Correlazione

GovWay estrae dal valore presente nell'header "Location" (per la richiesta) e dall'endpoint (per la richiesta stato e per la risposta) l'identificativo di correlazione al fine di correlare la richiesta con le successive operazioni.

Profilo “eDelivery”

Il profilo eDelivery consente di produrre configurazioni di scenari di interoperabilità che si basano sullo standard europeo eDelivery. Per rendere il trattamento dei messaggi conforme a tale standard, GovWay si interfaccia ad una installazione del software Domibus (<https://ec.europa.eu/digital-building-blocks/sites/display/DIGITAL/Domibus>).

Il processo di configurazione rimane strutturalmente analogo a quanto già descritto per la modalità API Gateway. Sono però presenti proprietà specifiche del contesto eDelivery i cui valori devono essere forniti affinché il dialogo con l’access point Domibus possa essere realizzato correttamente.

Nel seguito andiamo a descrivere i passi di configurazione evidenziando, per differenza con il caso API Gateway, gli elementi di eDelivery che dovranno essere gestiti. Al termine della configurazione è necessario procedere con l’export dei dati in formato *PMODE*. Il file prodotto è quello necessario per permettere la configurazione dell’access point Domibus.

4.1 Passi preliminari di configurazione

Per gestire in maniera più semplice i passi di configurazione dei servizi eDelivery è consigliabile impostare l’opportuna modalità operativa della govwayConsole selezionando la voce *eDelivery* sul selettori di modalità presente nella testata dell’applicazione.

Prima di procedere con la configurazione dei servizi si devono verificare i dati relativi ai soggetti interlocutori. Nel caso del soggetto interno al proprio dominio, i dati di configurazione possono essere gestiti alla sezione *Configurazione > Generale* (Fig. 4.1).

Sono presenti valori iniziali, inseriti dal processo di installazione, che devono essere verificati ed eventualmente aggiornati:

- *Base URL Erogazione*: Indirizzo pubblico del Domibus per la ricezione dei messaggi sul canale eDelivery.
- *Base URL Fruizione*: Indirizzo del servizio di GovWay riservato ai client per l’invio di messaggi sul canale eDelivery.

Tramite il collegamento *Visualizza Dati Soggetto* è possibile accedere alla conffigurazione del soggetto interno (Fig. 4.2).

eDelivery	
Base URL Erogazione	<input type="text" value="http://localhost:8080/domibus/services/msh"/>
Base URL Fruizione	<input type="text" value="http://localhost:8080/openspcoop2/as4/PD/"/>
Soggetto	EnteInterno
Visualizza Dati Soggetto	

Figure4.1: Configurazione delle Base URL eDelivery per il soggetto interno

Le proprietà eDelivery da fornire sono le seguenti:

- *Party Info - Id*: Identificativo del soggetto utilizzato nel canale eDelivery.
- *Party Info - Type Name*: Nome assegnato internamente allo schema indicato nel Type Value.
- *Party Info - Type Value*: Schema di generazione riferito all'identificativo del soggetto eDelivery.
- *Party Endpoint - URL*: Indirizzo pubblico del Domibus per la ricezione dei messaggi sul canale eDelivery.
- *Party Endpoint - Common Name*: Valore della omonima proprietà del certificato utilizzato dall'access point Domibus cui afferisce. Questo nome coincide con quello dell'access point.

4.2 Erogazione di servizi in modalità eDelivery

Configurare un'erogazione eDelivery permette ad un'applicazione interna di ricevere i messaggi inviati da un generico access point eDelivery esterno.

Il primo passo di configurazione prevede che venga censito il soggetto esterno mittente dei messaggi. La creazione di tale soggetto si realizza dalla sezione *Registro > Soggetti* della govwayConsole, impostando le proprietà eDelivery già descritte nella sezione precedente per il soggetto interno.

Il passo successivo è quello di registrare le API corrispondenti al servizio eDelivery alla sezione *Registro > API*. Le proprietà eDelivery, presenti nel form di creazione, sono quelle mostrate in Fig. 4.3.

Le proprietà da specificare sono le seguenti:

- *Service Info - Type*: Identificativo assegnato come tipo del servizio (opzionale).
- *Service Info - Name*: Nome del servizio.
- *Payload Profiles - File*: Campo per l'upload del descrittore XML che rappresenta il formato dei messaggi inviati dal mittente. Campo opzionale, utilizzabile per aggiungere nuovi profili rispetto a quelli già presenti nell'installazione standard di Domibus. Per la specifica del formato XML da adottare si consulti la documentazione ufficiale di Domibus.
- *Properties - File*: Campo per l'upload del descrittore XML che definisce le proprietà custom che saranno presenti nei messaggi inviati dal mittente. Campo opzionale, utilizzabile per aggiungere nuove property rispetto a quelle già presenti nell'installazione standard di Domibus. Per la specifica del formato XML da adottare si consulti la documentazione ufficiale di Domibus.

Soggetti > EntelInterno

Note: (*) Campi obbligatori

Soggetto

Nome *****

Descrizione

eDelivery

Party Info

Id *****

Type Name *****

Type Value *****

Party Endpoint

URL *****

Common Name *****

Invia **Cancella**

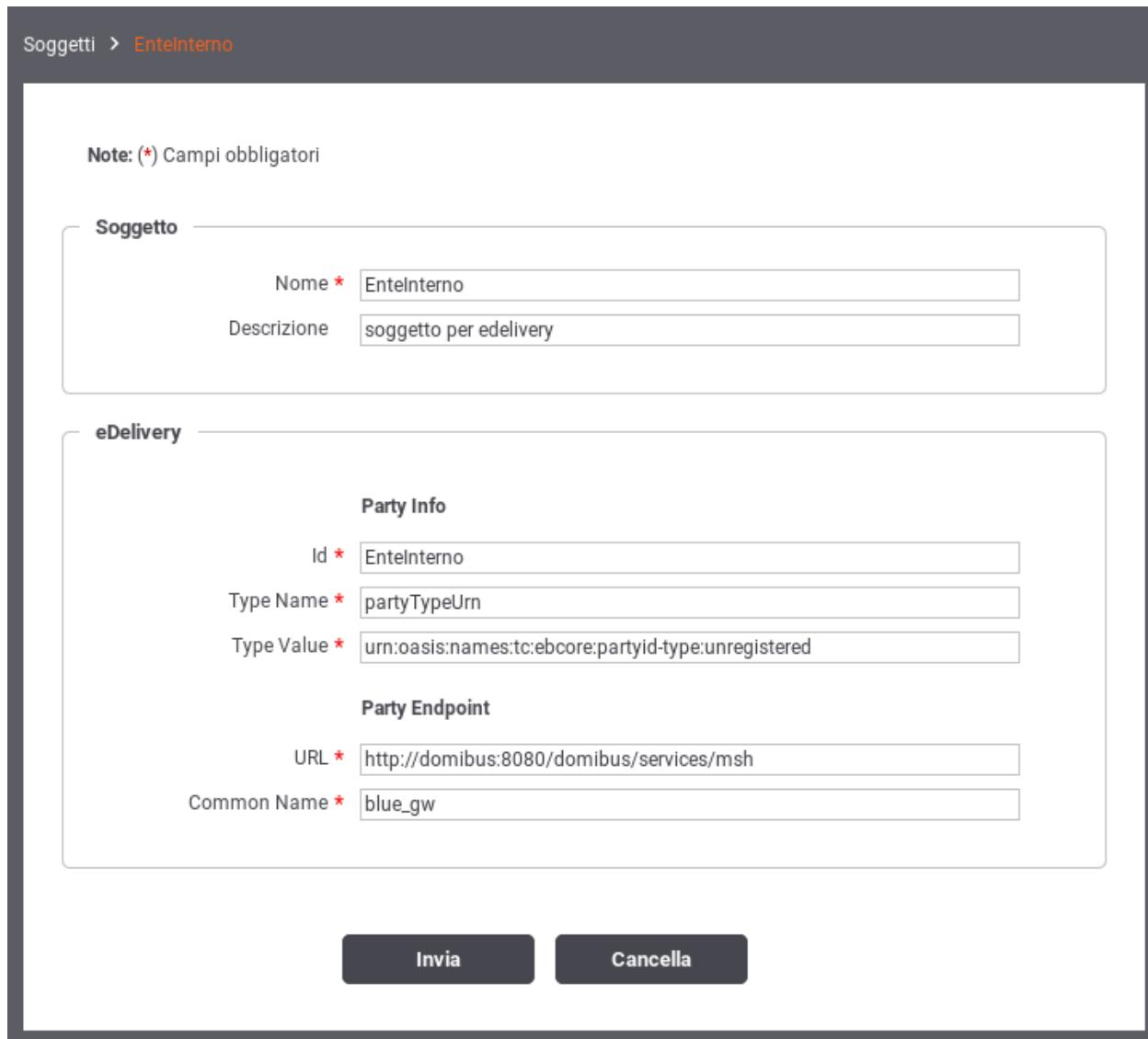


Figure4.2: Configurazione delle proprietà eDelivery per il soggetto interno

The screenshot shows the 'eDelivery' configuration interface. It includes sections for 'Service Info' (Type and Name fields), 'Payload Profiles' (Browse... button, showing 'No file selected.'), and 'Properties' (Browse... button, showing 'No file selected.').

Figure4.3: Registrazione API eDelivery - Proprietà specifiche

Dopo aver effettuato il salvataggio è necessario completare la configurazione del servizio utilizzando il link presente nella colonna *Risorse* o *Servizi*, a seconda che si tratti di un servizio Rest o Soap, in corrispondenza dell'elemento presente nell'indice dei servizi. Per ciascuna delle azioni/risorse elencate per il servizio (o create, nel caso che, non disponendo del descrittore del servizio, si proceda con la configurazione manuale delle azioni), si accede al dettaglio per completare la configurazione delle property eDelivery (Fig. 4.4).

The screenshot shows the 'eDelivery' configuration interface for an action. It includes fields for 'Action Info' (Name: POST_store.pdf) and 'Payload' (Profile: DefaultBinaryProfile, Compress: checked).

Figure4.4: Proprietà eDelivery relative alle azioni delle API

I valori da impostare nel form sono:

- *Action Info - Name*: Nome dell'azione.
- *Payload - Profile*: Payload Profile, tra quelli disponibili, da utilizzare per l'azione.
- *Payload - Compress*: Indicare se l'invio del messaggio farà uso di compressione dei dati.

Dopo aver creato l'API si procede con la configurazione dell'erogazione alla sezione *Registro > Erogazioni* della govwayConsole (Fig. 4.5).



Figure4.5: Proprietà eDelivery relative all'erogazione del servizio

L'unica impostazione eDelivery da fornire in questo contesto è:

- *Security Profile*: profilo di sicurezza adottato dagli access point durante la comunicazione. E" necessario scegliere tra i valori presenti, che corrispondono alle policy standard, già presenti in Domibus con l'installazione.

Nota: L'endpoint fornito alla voce Connettore sarà quello utilizzato da GovWay per la consegna dei messaggi consegnati all'access point Domibus interno.

Nota: Affinché le configurazioni apportate in modalità eDelivery possano essere attuate sull'access point Domibus è necessario procedere alla generazione del PMODE nel modo descritto alla sezione [Generazione del PMODE Domibus](#).

4.3 Fruizione di servizi in modalità eDelivery

Configurare una fruizione eDelivery permette ad un'applicazione interna di inviare messaggi da veicolare verso un generico access point eDelivery esterno.

Il processo di configurazione della fruizione eDelivery prevede inizialmente i medesimi passi già descritti per l'erogazione nella sezione [Erogazione di servizi in modalità eDelivery](#). Dovranno quindi essere configurati i dati eDelivery relativi ai soggetti interlocutori, interno ed esterno, dovranno inoltre essere censite le API relative al servizio da fruire.

Dopo aver censito le API si procede con la configurazione della fruizione creando un nuovo elemento nella sezione *Registro > Fruizioni* della govwayConsole. Analogamente al caso dell'erogazione si dovrà selezionare la security policy necessaria per gli scambi tra gli access point.

Nota: Affinché le configurazioni apportate in modalità eDelivery possano essere attuate sull'access point Domibus è necessario procedere alla generazione del PMODE nel modo descritto alla sezione [Generazione del PMODE Domibus](#).

4.4 Generazione del PMODE Domibus

Affinché il Domibus interno al proprio dominio sia in grado di recepire tutte le configurazioni prodotte nella modalità eDelivery, è necessario che gli venga fornito il relativo file PMODE, così come prevede la configurazione dell'access point eDelivery.

Dopo aver ultimato le configurazioni dei servizi eDelivery, tramite la govwayConsole, si procede all'esportazione del PMODE effettuando i seguenti passaggi (Fig. 4.6):

- Selezionare la voce di menu *Configurazione > Esporta*.
- Selezionare la tipologia archivio *domibus-pmode*.
- Premere il pulsante *Invia* e salvare il file XML che viene restituito.
- Effettuare l'upload del file ottenuto sulla Domibus Console.

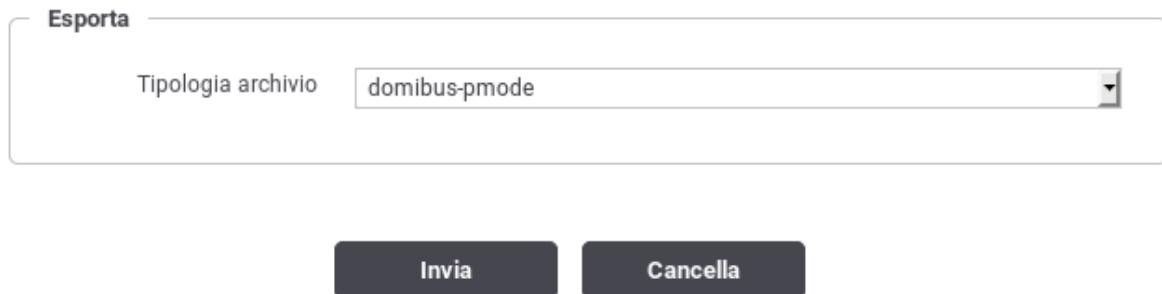


Figure4.6: Esportazione del PMODE

CHAPTER 5

Profilo “SPCoop”

Il profilo SPCoop consente di produrre le configurazioni per i servizi, in accordo alla omonima specifica di cooperazione applicativa della PA italiana. I passi di configurazione, per erogazioni e fruizioni, presentano minime differenze rispetto a quanto descritto per la modalità API Gateway. Nel seguito saranno descritte tali differenze.

5.1 Configurazione di un servizio SPCoop

Il primo passaggio per la configurazione di un servizio SPCoop è quello di creare il relativo Accordo di Servizio. Questi viene creato registrando una nuova API (sezione *Registro > API*). Come illustrato nelle figure seguenti, la particolarità di questa configurazione, rispetto a quanto descritto in precedenza, risiede nella presenza del campo *Soggetto referente*, nel quale deve essere selezionato uno dei soggetti precedentemente registrati.

Se non viene fornito un WSDL, relativo all'accordo di servizio, è necessario definire manualmente l'interfaccia del servizio, analogamente a quanto descritto in sezione *Configurazione manuale delle interfacce*. In questo caso, l'aggiunta del servizio, comprende i profili di collaborazione asincroni oltre alle caratteristiche aggiuntive specifiche del protocollo SPCoop (vedi sezione *Profili di gestione della busta eGov*). La figura seguente mostra i dettagli di questo caso.

La registrazione di una nuova erogazione o fruizione, presenta le seguenti differenze rispetto a quanto descritto per la modalità API Gateway:

- È presente il campo *Tipo* relativamente al servizio
- È presente il campo *Versione Protocollo* per selezionare la versione della specifica SPCoop adottata.

API > Aggiungi

Note: (*) Campi obbligatori

API

Soggetto referente * EnteInterno

Nome * Accordo1

Descrizione

Versione 1

Specifiche delle interfacce

WSDL Browse... No file selected.

Invia Cancella

Figure5.1: Creazione Accordo di Servizio SPCoop

API > Servizi di AccordoServizio:1 (EnteInterno) > [Aggiungi](#)

Note: (*) Campi obbligatori

Servizio

Nome *****

Descrizione

Informazioni Protocollo

Profilo di collaborazione

Filtro duplicati

Conferma ricezione

ID Collaborazione

Consegna in ordine

Scadenza

Invia **Cancella**

Figure5.2: Aggiunta Servizio SPCoop

Informazioni Generali

API	
Nome	AccordoServizio:1 (EnteInterno)
Tipo	Soap
Servizio *	servizio
Servizio	
Tipo	spc
Tipologia Servizio	normale
Versione Protocollo	eGov1.1-lineeGuida1.1

Figure5.3: Creazione erogazione SPCoop

5.2 Profili Asincroni

I servizi con profilo Asincrono richiedono alcuni passi di configurazione ulteriori rispetto alle normali configurazioni dei profili Sincroni e OneWay. Nelle sezioni successive vengono mostrati in dettagli i passi di configurazione ulteriori richiesti dal *Profilo di Collaborazione Asincrono Simmetrico* e dal *Profilo di Collaborazione Asincrono Asimmetrico*.

5.2.1 Profilo di Collaborazione Asincrono Simmetrico

La registrazione di un profilo asincrono simmetrico prevede che vengano correlati tra di loro due azioni di due servizi differenti presenti all'interno del solito accordo di servizio parte comune (API). Di seguito un esempio di tale configurazione.

Ruolo Fruitore

Per poter fruire di un servizio con il profilo asincrono simmetrico come prerequisito è richiesto almeno la registrazione di un applicativo client. La registrazione dell'applicativo fruitore deve prevedere, oltre alle normali configurazioni, la definizione di un connettore attraverso il quale la PdD consegnerà la risposta asincrona. È possibile definire un connettore per la "Risposta Asincrona" impostando la modalità avanzata nella console di gestione e accendo in modifica ad un applicativo client precedentemente registrato. Il link "Risposta Asincrona" consentirà di definire i parametri di accesso al backend per la gestione della risposta asincrona.

Una volta creato l'applicativo è possibile procedere con la registrazione della fruizione del servizio asincrono simmetrico dedicato all'invio della richiesta. Il controllo degli accessi della fruizione deve obbligatoriamente essere configurato per autenticare gli applicativi precedentemente registrati, in modo da identificare l'applicativo chiamante e poterlo associare alla sessione asincrona.

Terminata la registrazione della fruizione, dovrà essere registrata un'erogazione del servizio asincrono dedicato alla ricezione della risposta. La risposta ricevuta verrà consegnata al connettore definito per la risposta asincrona associato all'applicativo chiamante originario salvato nella sessione asincrona.

API > Servizi di aaaa:1 (INPS) > Azioni di serviziocorrelato > **Aggiungi**

Note: (*) Campi obbligatori

Azione

Nome * azioneconrelata

Informazioni Protocollo

Profilo usa profilo servizio

Correlazione asincrona

Correlata al servizio servizio

Correlata all'azione * azione1

SALVA

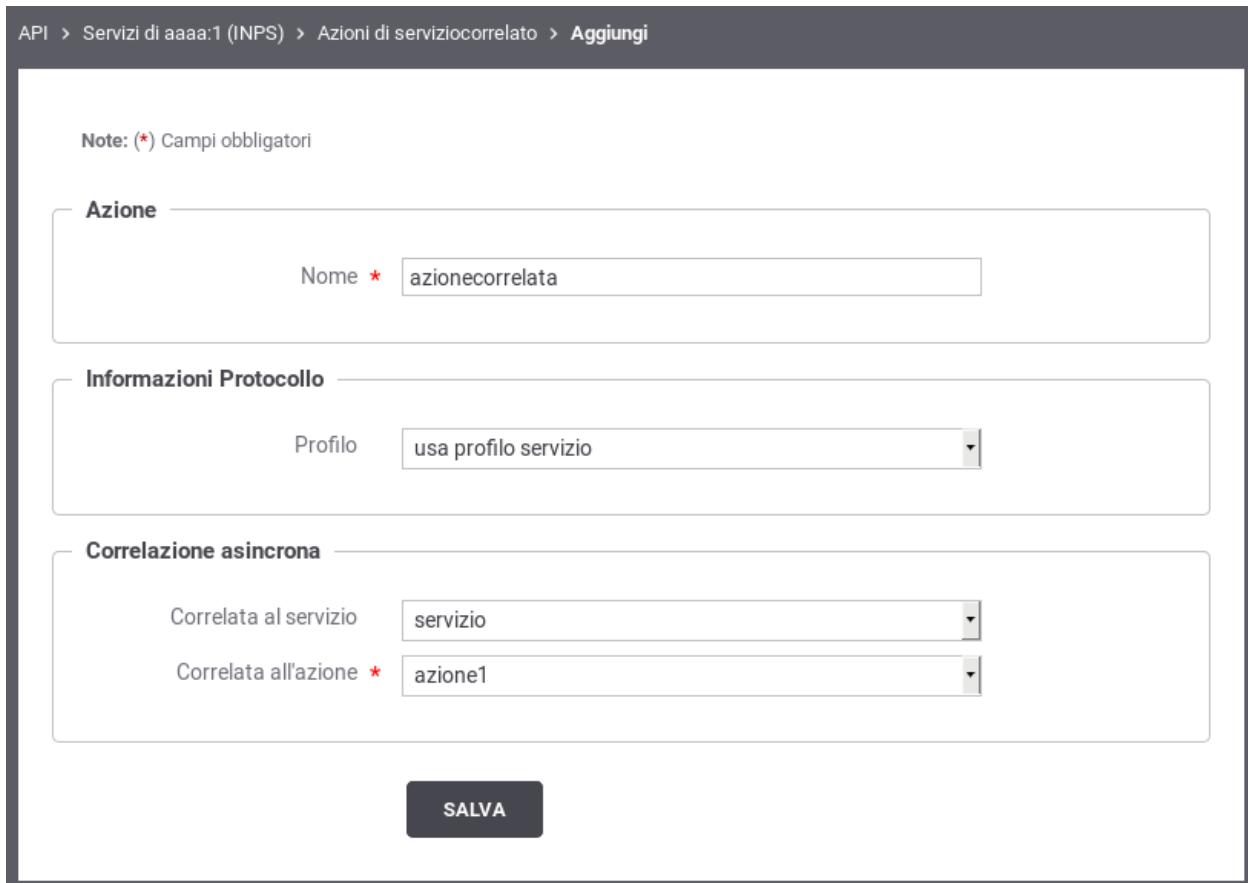


Figure5.4: Correlazione Asincrona Simmetrica

The screenshot shows the configuration page for the 'EsempioClientAsincronoSimmetrico' application. At the top, a note says 'Note: (*) Campi obbligatori'. The main section is titled 'Applicativo' and contains the following fields:

Soggetto	FRUITORE
Nome *	EsempioClientAsincronoSimmetrico
Tipo	Client
Proprietà(0)	
Risposta Asincrona (visualizza)	

Figure5.5: Accesso alla configurazione dell'Applicativo client per la Risposta Asincrona

Nota: Configurazione Servizio Ricezione della Risposta

Durante la registrazione dell'erogazione del servizio di risposta, al connettore richiesto dalla maschera di configurazione può essere fornito un endpoint qualsiasi. Tale endpoint non verrà effettivamente utilizzato poiché la risposta asincrona ricevuta verrà inoltrata al backend configurato come "Risposta Asincrona" dell'applicativo client che ha effettuato la richiesta.

Ruolo Erogatore

Per poter erogare un servizio con il profilo asincrono simmetrico non sono richieste particolari configurazioni. Dovrà essere erogato il servizio relativo alla richiesta e frutto il servizio su cui inviare la risposta.

5.2.2 Profilo di Collaborazione Asincrono Asimmetrico

La registrazione di un profilo asincrono asimmetrico prevede che vengano correlati tra di loro due azioni, normalmente di uno stesso servizio, presenti all'interno dell'accordo di servizio parte comune (API). Di seguito un esempio di tale configurazione.

Applicativi > EsempioClientAsincronoSimmetrico > Risposta Asincrona

Risposta Asincrona

Note: (*) Campi obbligatori

Servizio Applicativo

Nome	EsempioClientAsincronoSimmetrico
------	----------------------------------

Connettore

Abilitato	<input checked="" type="checkbox"/>
Endpoint *	<input type="text" value="http://backendExample/gestioneRispostaAsincrona"/>
Autenticazione Http	<input type="checkbox"/>
Autenticazione Token	<input type="checkbox"/>
AutenticazioneHttps	<input type="checkbox"/>
Proxy	<input type="checkbox"/>
Ridefinisci Tempi Risposta	<input type="checkbox"/>

Figure5.6: Configurazione dell'Applicativo client per la Risposta Asincrona

Note: (*) Campi obbligatori

Azione

Nome ***** azioneCorrelata

Informazioni Protocollo

Profilo ridefinisci

Profilo di collaborazione asincronoAsimmetrico

Filtro duplicati

Conferma ricezione

ID Collaborazione

Consegna in ordine

Scadenza

Correlazione asincrona

Correlata al servizio -

Correlata all'azione azione

SALVA

Figure5.7: Correlazione Asincrona Asimmetrica

Ruolo Fruitore

Per poter fruire un servizio con il profilo asincrono asimmetrico non sono richieste particolari configurazioni. Dovrà essere fruito il servizio su cui inviare la richiesta e richiedere l'esito della risposta.

Ruolo Erogatore

Per poter erogare un servizio con il profilo asincrono asimmetrico come prerequisito è richiesto la registrazione di un applicativo server. Durante la registrazione dell'applicativo possono essere indicati i parametri di accesso al backend a cui consegnare la richiesta asincrona. Una volta creato l'applicativo è possibile definire i parametri di accesso al backend a cui consegnare la richiesta stato asincrona impostando la modalità avanzata nella console di gestione ed entrando in modifica sull'applicativo server precedentemente registrato, dove sarà disponibile il link “Risposta Asincrona”.

Applicativi > EsempioServerAsincronoAsimmetrico

EsempioServerAsincronoAsimmetrico

Note: (*) Campi obbligatori

Applicativo

Soggetto	ENTE
Nome *	EsempioServerAsincronoAsimmetrico
Tipo	Server
Proprietà(0)	
Risposta Asincrona (visualizza)	

Connettore

Endpoint *	http://backendExample/gestioneRichiestaAsincrona
Autenticazione Http	<input type="checkbox"/>
Autenticazione Token	<input type="checkbox"/>
AutenticazioneHttps	<input type="checkbox"/>
Proxy	<input type="checkbox"/>
Ridefinisci Tempi Risposta	<input type="checkbox"/>

Figure5.8: Accesso alla configurazione dell'Applicativo server per la Risposta Asincrona

Una volta creato l'applicativo è possibile procedere con la registrazione dell'erogazione del servizio con asincrono asimmetrico selezionando l'applicativo server precedentemente registrato come connettore di backend come mostrato in figura Fig. 5.10.

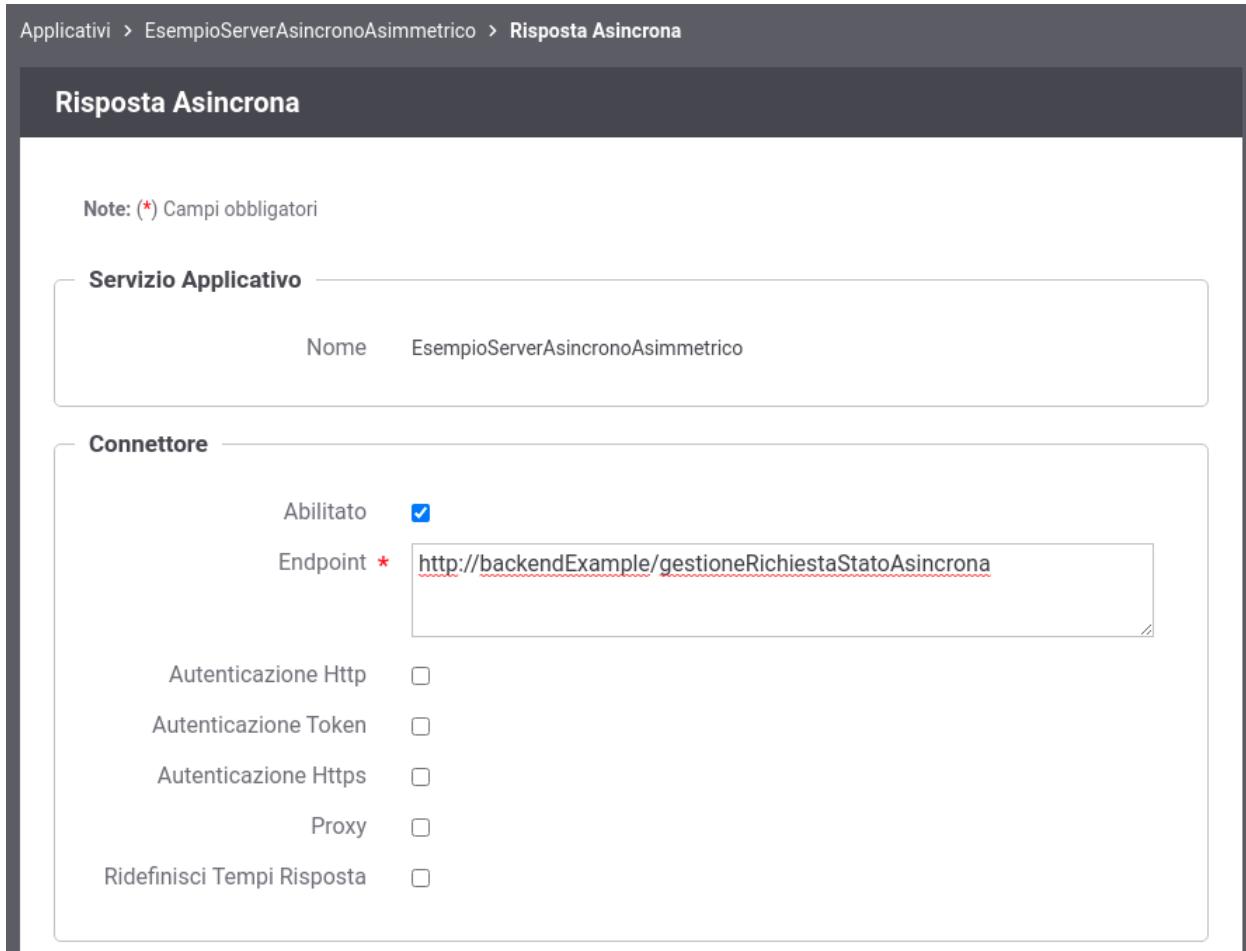


Figure 5.9: Configurazione dell'Applicativo server per la Risposta Asincrona

Erogazioni > Aggiungi

Note: (*) Campi obbligatori

Informazioni Generali

Soggetto Erogatore: ENTE

API

Nome *: ESEMPIOPROFILICOMPLETO v1 (ENTE)

Tipo: Soap

Servizio (Soap) *: AsincronoAsimmetrico

Tipo: spc

Tipologia Servizio: normale

Versione Protocollo: usa versione erogatore

Controllo degli Accessi

Accesso API: autenticato

Connettore

Utilizza Applicativo Server:

Applicativo: EsempioServerAsincronoAsimmetrico

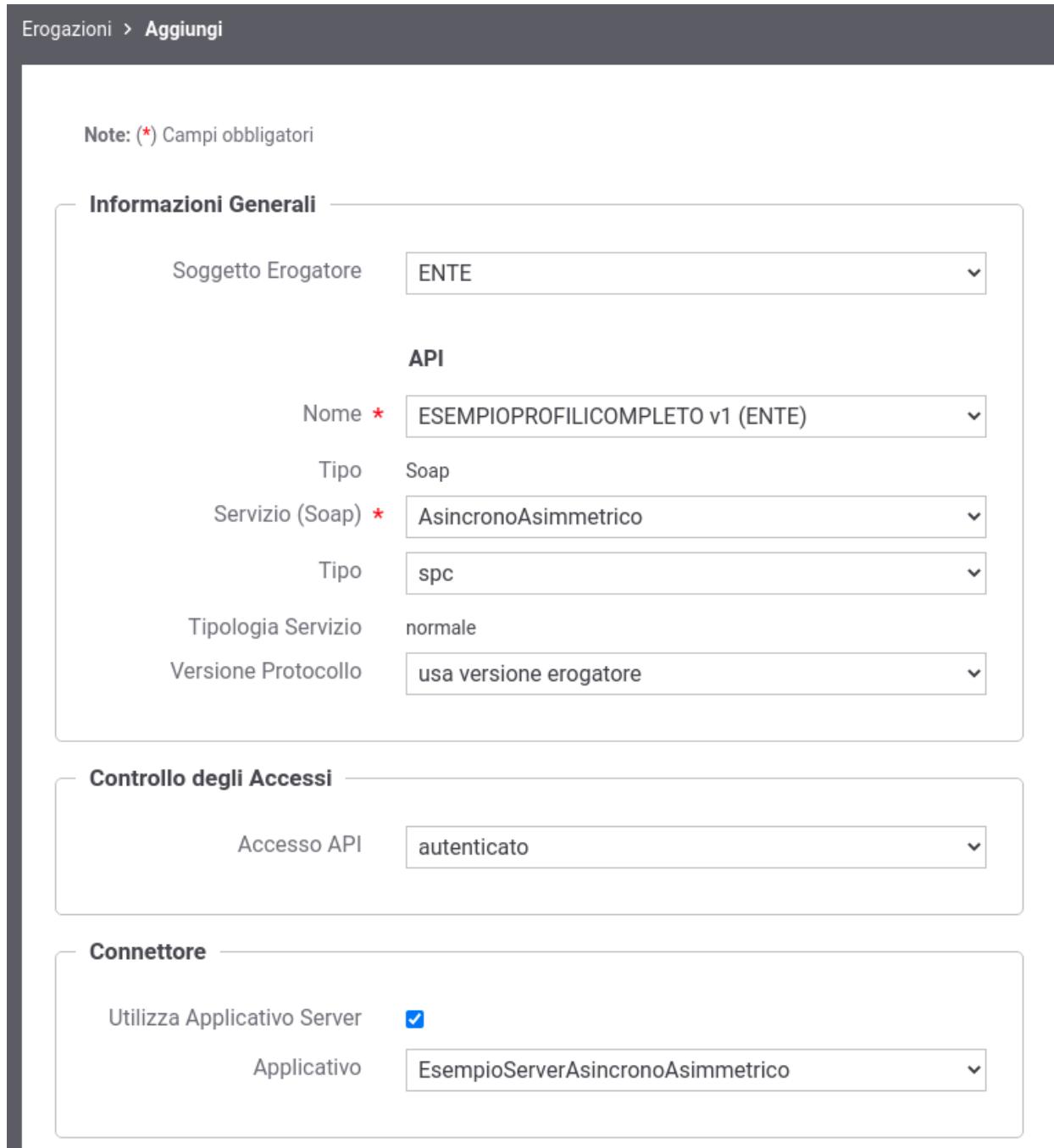


Figure5.10: Selezione dell'Applicativo server per l'erogazione del servizio con profilo asincrono asimmetrico.

5.3 Interfacce WSDL (concettuale, logico ed implementativo)

La specifica SPCoop prevede che nell'accordo di servizio siano specificati i documenti WSDL del servizio applicativo erogatore e, nel caso di profili di collaborazione asincroni asimmetrici, anche quelli del servizio applicativo correlato erogato dal soggetto fruitore.

La Tabella 5.1 riepiloga i documenti necessari alla descrizione formale di un accordo di servizio che possono essere associati agli accordi parte comune e specifica se viene utilizzata la modalità avanzata della console

Table5.1: Descrizione di un accordo di servizio

Nome Documento	Accordo
<i>Specifiche delle Interfacce</i>	
WSDL Definitorio	Parte Comune
WSDL Concettuale	Parte Comune
WSDL Logico Erogatore	Parte Comune
WSDL Logico Fruitore	Parte Comune
<i>Specifiche delle Implementazioni</i>	
WSDL Implementativo Erogatore	Parte Specifica
WSDL Implementativo Fruitore	Parte Specifica

5.4 Profili di gestione della busta eGov

L'interfaccia *completa* fornisce la possibilità di fruire/erogare di servizi SPCoop che non seguono le Linee Guida 1.1 ma si basano sul documento e-Gov 1.1. Questa funzionalità è utile sia per backward compatibility in quei domini dove i servizi non sono ancora stati adeguati al profilo descritto nelle Linee Guida 1.1, sia per usufruire di servizi infrastrutturali quali *consegna affidabile*, *consegna in ordine*, *conversazioni* che non sono presenti nel profilo Linee Guida 1.1.

Fruizione di un servizio.

Supponiamo di essere in un contesto dove vogliamo usufruire di un servizio erogato da un soggetto la cui PdD non è ancora stata adeguata a quanto descritto nelle Linee Guida 1.1. Per usufruire del servizio, il soggetto fruitore deve inviare buste conformi al profilo e-Gov 1.1, nonostante la propria porta di dominio sia già conforme alle Linee Guida 1.1. Per gestire tale contesto è possibile definire il soggetto erogatore con profilo *eGov1.1*. In un successivo momento, la PdD del soggetto erogatore può iniziare ad adeguarsi alle Linee Guida 1.1. Supponiamo che l'adeguamento sia incrementale, fornito per un servizio alla volta. Per usufruire dei servizi erogati da tale soggetto, con la giusta modalità (Linee Guida 1.1 o e-Gov 1.1) è possibile *ridefinire il profilo di gestione all'interno del servizio*.

Erogazione di un servizio.

Poniamoci in un contesto in cui la Porta di Dominio eroga dei servizi che rispettano quanto descritto nelle Linee Guida 1.1. In questo contesto, i soggetti di PdD che non si sono ancora adeguati alle linee guida, non potrebbero usufruire dei servizi. La PdD può essere configurata, in modo da erogare i servizi, per questi soggetti, secondo il profilo *eGov 1.1*. Questa configurazione richiede che al soggetto fruitore venga associato un profilo *eGov 1.1*. In un successivo momento, la PdD di un soggetto fruitore può iniziare ad adeguarsi alle Linee Guida 1.1. Si creano quindi due situazioni di transizione dove devono coesistere entrambe le specifiche:

- Un soggetto fruisce per alcuni servizi erogati secondo le specifiche e-Gov1.1, per altri secondo le Linee Guida 1.1
- Uno o più fruitori accedono al un servizio erogato secondo le specifiche e-Gov1.1, altri secondo le Linee Guida 1.1

In entrambi i casi, per erogare il servizio con la giusta modalità (linee guida o e-gov 1.1) è possibile ridefinire il profilo di gestione impostandolo nella lista dei fruitori del servizio.

5.4.1 Profilo di gestione e-Gov 1.1

Il documento delle linee guida ha deprecato alcune opzioni al fine di snellire la specifica. Per mantenere la compatibilità con la vecchia versione viene sempre offerta la possibilità di specificare tali opzioni all'interno degli accordi di servizio. Tali funzionalità vengono impostate/validate all'interno della busta e-Gov solo se il servizio viene fruito/erogato con profilo *eGov1.1*.

Table5.2: Opzioni della busta eGov

Nome	Default	Funzionalità
Filtro duplicati	true	Funzionalità di filtro delle buste duplicate (Imposta l'attributo inoltro del profilo di trasmissione al valore EGOV_IT_ALPIUUNAVOLTA).
Conferma Ricezione	false	Funzionalità di consegna affidabile delle buste spcoop attraverso l'utilizzo dei riscontri (Imposta l'attributo confermaRicezione del profilo di trasmissione al valore true).
ID Conversazione	false	Aggiunge un elemento Collaborazione alla busta (Diverse istanze di cooperazione possono essere correlate in un'unica conversazione).
Consegna in ordine	false	Consegna in ordine delle buste (Richiede Filtro Duplicati e Conferma Ricezione)
Scadenza		Assegna una scadenza temporale alla busta SPCoop

Di seguito un esempio di creazione di un accordo di servizio che richiede consegna affidabile tramite riscontri, filtro duplicati e id di conversazione per un servizio sincrono.

Opzioni Avanzate

Profilo di collaborazione	<input type="text" value="sincrono"/>
Filtro Duplicati	<input checked="" type="checkbox"/>
Conferma Ricezione	<input checked="" type="checkbox"/>
ID Conversazione	<input checked="" type="checkbox"/>
Consegna in Ordine	<input checked="" type="checkbox"/>
Scadenza	<input type="text"/>

Figure5.11: Controlli avanzati sulle informazioni eGov relative all'accordo di servizio

CHAPTER 6

Profilo “Fatturazione Elettronica”

Il profilo «Fatturazione Elettronica» consente di utilizzare GovWay come nodo di interconnessione al Sistema di Interscambio (SdI), responsabile della gestione dei flussi di fatturazione elettronica.

GovWay supporta la connessione al SdI attraverso lo scenario di interoperabilità su rete Internet basato sull’accesso al servizio *SdICoop*. Il servizio SdICoop prevede un protocollo di comunicazione, basato su SOAP, che veicola messaggi (fatture, archivi, notifiche e metadati) secondo la codifica dettata dalle specifiche tecniche (Per dettagli in merito si faccia riferimento alle Specifiche Tecniche SdI (<https://www.fatturapa.gov.it/it/norme-e-regole/DocumentazioneSDI/>).

Il profilo «Fatturazione Elettronica» consente, ai sistemi di gestione delle fatture di un ente, di non occuparsi della gestione del formato di scambio, previsto dal SdI, mantenendo un grado di interfacciamento notevolmente semplificato. Più in dettaglio:

- I gestionali dell’ente, registrati come applicativi su GovWay, inviano/ricevono le fatture e le notifiche, previste dal colloquio, nel formato originario XML senza ulteriori complessità.
- I metadati presenti nelle comunicazioni con il SdI vengono estratti ed elaborati da GovWay e trasmessi ai gestionali dell’ente tramite appositi *Header di Integrazione SdI*.
- La produzione dei metadati SdI, nel caso delle comunicazioni in uscita (fatturazione attiva), è a carico di GovWay che provvede anche a generare gli identificativi univoci da associare ai messaggi da trasmettere al SdI.

Per la produzione delle configurazioni necessarie a rendere operativo GovWay sono stati realizzati due wizard che guidano l’utente verso il corretto inserimento dei dati necessari. Gli scenari di configurazione supportati sono due e riguardano i casi della *Fatturazione Passiva* e *Fatturazione Attiva*.

6.1 Fatturazione Passiva

Nello scenario di fatturazione passiva si utilizza GovWay per la ricezione delle fatture in arrivo dal SdI. GovWay attua la decodifica del messaggio SdI ricevuto, al fine di estrarre i file fattura in esso contenuti e trasmetterli, nel formato FatturaPA, all'applicativo registrato come destinatario.

Lo scenario complessivo, relativo alla Fatturazione Passiva, è quello illustrato in Fig. 6.1.

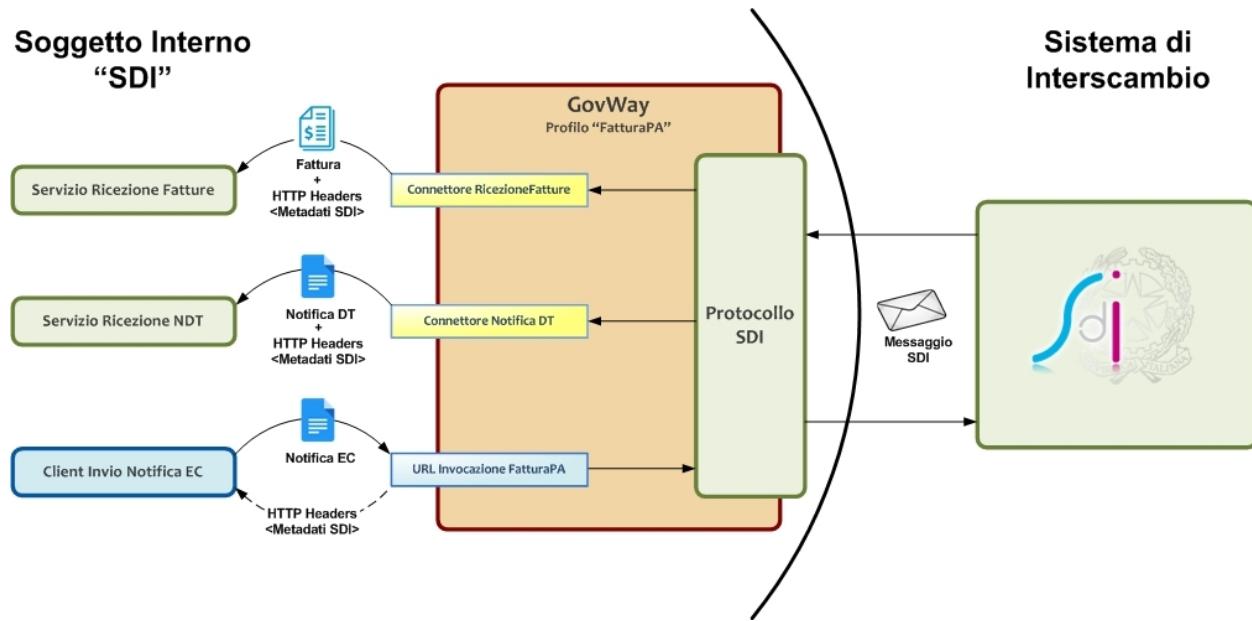


Figure 6.1: Scenario di interoperabilità relativo alla Fatturazione Passiva

Descriviamo per punti i passi significativi di questo scenario:

- *Servizio Ricezione Fatture.* Per consentire a GovWay di consegnare le fatture ricevute dal SdI è necessario esporre un servizio i cui riferimenti per l'accesso dovranno essere configurati nel contesto del *Connettore RicezioneFatture*, presente nella configurazione di GovWay.

Le fatture vengono ricevute da GovWay formato codificato dal protocollo SdI, e comprendono il lotto delle fatture, con i relativi allegati, e un insieme di metadati che descrivono il contesto di invocazione. GovWay si occupa di estrarre le informazioni presenti, elaborando il messaggio SdI, provvedendo quindi a consegnare il lotto di fatture al servizio destinatario, nel formato *FatturaPA* attraverso l'invocazione di una HTTP POST. I metadati raccolti dal messaggio SdI vengono forniti, nel contesto della medesima richiesta, sotto forma di HTTP Headers (fare riferimento alla [Tabella 6.1](#)).

Nota: Nella configurazione di default GovWay non consegna il file Metadati all'applicativo. È possibile attivare la consegna abilitando la proprietà "org.openscoop2.protocol.sdi.fatturazionePassiva.consegnafMetadati" all'interno del file /etc/govway/sdi_local.properties. Il file Metadati verrà consegnato, codificato in base64, nell'header HTTP "GovWay-SDI-FileMetadati".

- *Client Invio Notifica EC.* I sistemi dell'ente, dopo aver ricevuto le fatture, inviano le *Notifiche di Esito Committente*, previste dal protocollo SdI, utilizzando un apposito servizio di GovWay. La URL di invocazione di tale servizio sarà disponibile al termine del processo di configurazione descritto più avanti. GovWay provvede a codificare il messaggio SdI di richiesta contenente il messaggio di notifica ricevuto dall'applicativo mittente.

I metadati prodotti per il messaggio SdI, unitamente all'identificativo messaggio univoco generato, vengono restituiti all'applicativo mittente sotto forma di HTTP Headers (fare riferimento alla [Tabella 6.2](#)).

- *Servizio Ricezione NDT*. Per consentire a GovWay di consegnare le eventuali *Notifiche di Decorrenza Termini* è necessario esporre un servizio i cui riferimenti per l'accesso dovranno essere configurati nel contesto del *Connettore NotificaDT*, presente nella configurazione di GovWay.

GovWay consegna le notifiche DT nel formato originale tramite una HTTP POST, includendo come HTTP Headers i metadati estratti dal messaggio SdI originariamente ricevuto (fare riferimento alla [Tabella 6.3](#)).

Table6.1: Header di Integrazione Ricezione Fattura

Header	Descrizione
GovWay-SDI-FormatoArchivioBase64	Indica se il file fattura è codificato in formato Base64
GovWay-SDI-FormatoArchivioInvioFattura	Indica se è stata utilizzata la modalità di firma CAdES o XAdES (P7M o XML)
GovWay-SDI-FormatoFatturaPA	Indice di versione del formato FatturaPA adottato
GovWay-SDI-IdentificativoSdI	Identificativo assegnato dal SdI alla fattura
GovWay-SDI-MessageId	Identificativo assegnato alla fattura dall'ente trasmittente
GovWay-SDI-NomeFile	Nome del file fattura
GovWay-SDI-NomeFileMetadati	Nome del file di metadati
GovWay-Transaction-ID	Identificativo della transazione assegnato da GovWay

Table6.2: Header di Integrazione Invio Notifica EC

Header	Descrizione
GovWay-Transaction-ID	Identificativo della transazione assegnato da GovWay

Table6.3: Header di Integrazione Ricezione Notifica DT

Header	Descrizione
GovWay-SDI-IdentificativoSdI	Identificativo assegnato dal SdI alla fattura
GovWay-SDI-NomeFile	Nome del file fattura
GovWay-Transaction-ID	Identificativo della transazione assegnato da GovWay

Per produrre le configurazioni necessarie all'utilizzo dello scenario di fatturazione passiva, è possibile utilizzare il wizard messo a disposizione per semplificare l'attività di configurazione di GovWay. I passi da eseguire sono i seguenti:

1. Scaricare il govlet per la fatturazione passiva al seguente indirizzo:
 - <http://www.govway.org/govlets/fatturazione-passiva.zip>
2. Avviare il govlet posizionandosi sulla sezione *Configurazione > Importa* della GovWayConsole e selezionare il file appena scaricato come oggetto da importare.
3. *Soggetto SDI*: al primo step del wizard viene richiesto di indicare, tra gli elementi presenti nella lista a discesa, il soggetto interno destinatario delle fatture. Si tratta di un soggetto appartenente al profilo «FatturaPA».
4. *Servizio SdIRiceviNotifica erogato dal Sistema di Interscambio*: al secondo step viene richiesto di indicare la URL che corrisponde all'endpoint del servizio SdIRiceviNotifica, necessario per l'invio delle *Notifiche di Esito Committente*.

Nota: il valore suggerito dalla maschera di configurazione del govlet fa riferimento alla url del sistema di produzione SDI. Se si vuole configurare un servizio di test è necessario cambiare tale valore ed impostare il riferimento all'ambiente

di test SDI. I certificati, sia per l’ambiente di test che di produzione, devono essere stati inseriti nel truststore di GovWay.

5. *Credenziali per accesso URL NotificaEsito*: al terzo step viene richiesto di fornire il criterio di autenticazione utilizzato dall’applicativo per inviare la notifica di esito committente.
6. *Applicativo per consegna FatturaPA*: al quarto step viene richiesto di fornire i dati di configurazione del connettore, utilizzato da GovWay per la consegna delle fatture. La configurazione del connettore comprende: endpoint, credenziali di autenticazione ed eventualmente i riferimenti del proxy.
7. *Applicativo per consegna NotificaDecorrenzaTermini*: al quinto ed ultimo step viene richiesto di fornire i dati di configurazione del connettore, utilizzato da GovWay per la consegna della notifica di decorrenza termini. La configurazione del connettore comprende: endpoint, credenziali di autenticazione ed eventualmente i riferimenti del proxy.

6.1.1 Ricezione Fatture e Notifiche di Decorrenza Termini

Allo SdI, per raggiungere il servizio di RicezioneFatture su Govway, dovrà essere comunicata la seguente URL:

```
https://<host_govway>/govway/sdi/in/<SoggettoSDI>/RicezioneFatture/v1
```

Le fatture e le notifiche saranno consegnati all’applicativo dell’ente secondo le modalità specificate durante l’esecuzione del Govlet. In fase di consegna delle fatture e delle notifiche verranno generati rispettivamente gli header descritti nelle tabelle precedenti.

6.1.2 Invio della Notifica di Esito Committente

Per l’invio della Notifica di Esito Committente l’applicativo deve utilizzare:

- Una URL così composta:

```
http://<host_govway>/govway/sdi/out/xml2soap/<SoggettoSDI>/CentroServiziFatturaPA/  
→SdIRiceviNotifica/v1?NomeFile=<NomeFileFattura>&IdentificativoSdI=  
→<identificativoSDI>
```

dove:

- *host_govway*: è l’hostname con cui è raggiungibile l’istanza di Govway.
- *SoggettoSDI*: il soggetto interno destinatario delle fatture, come configurato durante l’esecuzione del govlet di fatturazione passiva.
- *NomeFileFattura*: è il nome del file che contiene la fattura cui fa riferimento la notifica EC.
- *identificativoSDI*: è l’identificativo SDI che fa riferimento al lotto della fattura ricevuta.
- L’invocazione deve essere corredata dalle credenziali che sono state indicate durante la configurazione tramite il relativo govlet.
- Utilizzare l’header http *Content-Type* valorizzato con *text/xml* o *application/xml*

Un esempio di invio di una fattura viene fornito tramite il seguente comando curl:

Soggetto Interno al Dominio

In questo esempio si suppone che il nome del soggetto (riferito precedentemente come *SoggettoSDI*) fornito durante la fase di installazione di GovWay sia *Ente*.

```
curl -X POST -basic --user SdIRiceviNotifica:123456 \
--data-binary @IT01234567890_11111_EC_001.xml \
-H "Content-Type: application/xml" \
"http://127.0.0.1:8080/govway/sdi/out/xml2soap/Ente/CentroServiziFatturaPA/ \
→SdIRiceviNotifica/v1?NomeFile=IT01234567890_11111.xml&IdentificativoSdI=345"
```

Nota: La generazione di un nome di file univoco da associare alla notifica di esito committente viene gestita da GovWay.

È possibile disabilitare tale gestione disabilitando la proprietà “org.openscoop2.protocol.sdi.fatturazionePassiva.nomeFile.gestione” nel file “/etc/govway/sdi_local.properties”. Se viene disabilitata la funzionalità (attiva per default), la gestione dei nomi dei file (correttezza sintattica, univocità, ...) è demandata all’Applicativo Client che deve obbligatoriamente fornire il nome da associare alla notifica di esito committente del file attraverso uno dei seguenti modi:

- query parameter “NomeFile”
- header http “SDI-NomeFile”
- header http “GovWay-SDI-NomeFile”

6.2 Fatturazione Attiva

Nello scenario di fatturazione attiva si utilizza GovWay per l’invio delle fatture al SdI. GovWay attua la codifica dei file ricevuti al fine di produrre un messaggio valido per l’invio al SdI.

Lo scenario complessivo, relativo alla Fatturazione Attiva, è quello illustrato in Fig. 6.2.

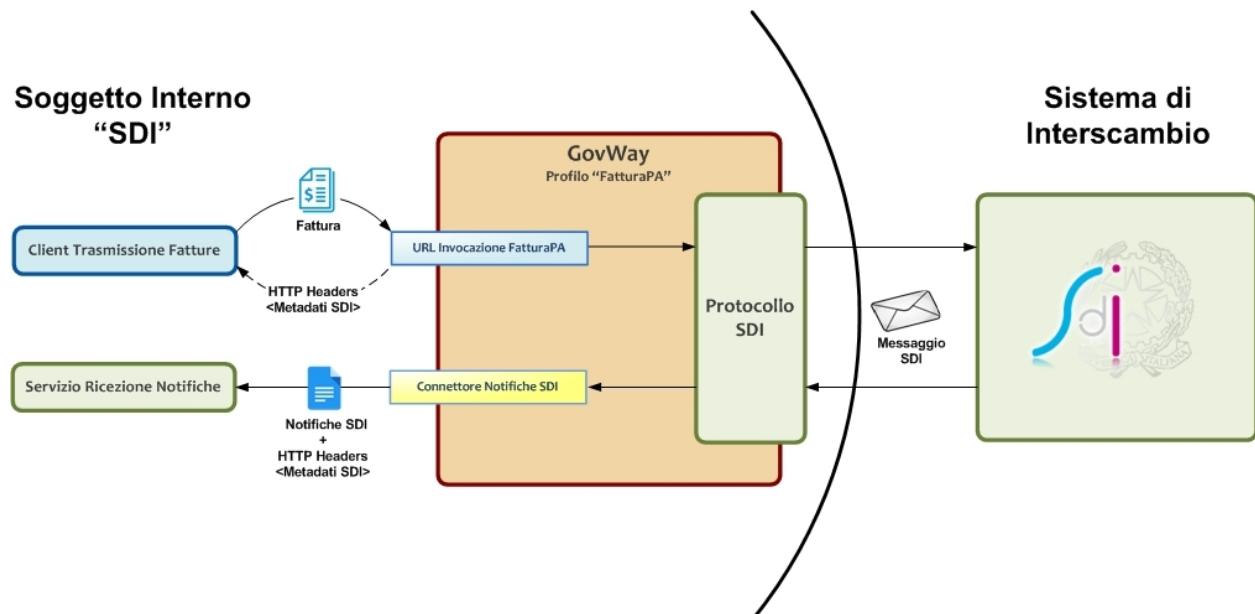


Figure6.2: Scenario di interoperabilità relativo alla Fatturazione Attiva

Descriviamo per punti i passi significativi di questo scenario:

- *Client Trasmissione Fatture.* I sistemi dell’ente possono trasmettere le fatture al SdI tramite un apposito servizio di ricezione di GovWay. La URL di invocazione di tale servizio sarà disponibile al termine del processo di configurazione dello scenario di fatturazione attiva descritto più avanti. Una volta ricevuta la fattura, nel formato previsto da FatturaPA, GovWay provvede a codificare il messaggio SdI di richiesta contenente la fattura da trasmettere. I metadati prodotti per il messaggio SdI, unitamente all’identificativo SdI, vengono restituiti all’applicativo mittente sotto forma di HTTP Headers (fare riferimento alla [Tabella 6.4](#)).
- *Servizio Ricezione Notifiche.* I sistemi dell’ente devono esporre un servizio adibito alla ricezione delle notifiche che il SdI invia successivamente alla trasmissione di una fattura. I riferimenti per l’accesso a tale servizio dovranno essere configurati nel contesto del *Connettore NotificheSDI*, presente nella configurazione di GovWay.

GovWay consegna le notifiche, al servizio dell’ente, nel formato originale tramite una HTTP POST, includendo come HTTP Headers i metadati estratti dal messaggio SdI originariamente ricevuto (fare riferimento alla [Tabella 6.5](#)).

Table6.4: Header di Integrazione “Trasmissione Fatture”

Header	Descrizione
GovWay-SDI-IdentificativoSdI	Identificativo assegnato dal SdI alla fattura
GovWay-SDI-NomeFile	Nome del file fattura
GovWay-Transaction-ID	Identificativo della transazione assegnato da GovWay

Table6.5: Header di Integrazione “Ricezione Notifiche”

Header	Descrizione
GovWay-SDI-IdentificativoSdI	Identificativo assegnato dal SdI alla fattura
GovWay-SDI-NomeFile	Nome del file fattura
GovWay-Transaction-ID	Identificativo della transazione assegnato da GovWay

Per produrre le configurazioni necessarie all’utilizzo dello scenario di fatturazione attiva, è possibile utilizzare il wizard messo a disposizione per semplificare l’attività di configurazione di GovWay. I passi da eseguire sono i seguenti:

1. Scaricare il govlet per la fatturazione attiva al seguente indirizzo
 - <http://www.govway.org/govlets/fatturazione-attiva.zip>
2. Avviare il govlet posizionandosi sulla sezione *Configurazione > Importa* della GovWayConsole e selezionare il file appena scaricato come oggetto da importare.
3. *Soggetto SDI:* al primo step viene richiesto di indicare, tra gli elementi presenti nella lista a discesa, il soggetto interno mittente delle fatture. Si tratta di un soggetto appartenente al profilo «FatturaPA».
4. *Servizio SdIRiceviFile erogato dal Sistema di Interscambio:* al secondo step viene richiesto di indicare la URL che corrisponde all’endpoint del servizio SdIRiceviFile, erogato dal SdI per la trasmissione delle fatture.

Nota: il valore suggerito dalla maschera di configurazione del govlet fa riferimento alla url del sistema di produzione SDI. Se si vuole configurare un servizio di test è necessario cambiare tale valore ed impostare il riferimento all’ambiente di test SDI. I certificati, sia per l’ambiente di test che di produzione, devono essere stati inseriti nel truststore di GovWay.

5. *Credenziali per accesso URL RiceviFile:* al terzo step viene richiesto di fornire il criterio di autenticazione utilizzato dall’applicativo per invocare la url del GovWay per la trasmissione delle fatture.
6. *Applicativo per consegna Notifiche:* al quarto ed ultimo step viene richiesto di fornire i dati di configurazione del connettore, utilizzato da GovWay per la consegna delle notifiche inviate dal SdI, successivamente alla

trasmissione di una determinata fattura. La configurazione del connettore comprende: endpoint, credenziali di autenticazione ed eventualmente i riferimenti del proxy.

6.2.1 Invio della fattura

Per l'invio della fattura l'applicativo mittente deve utilizzare:

- Una URL così composta:

```
http://<host-govway>/govway/sdi/out/xml2soap/<SoggettoSDI>/CentroServiziFatturaPA/
  ↪SdIRiceviFile/v1?Versione=<VersioneFatturaPA>&TipoFile=<TipoFile>&IdPaese=
  ↪<IdPaese>&IdCodice=<IdCodice>
```

dove:

- *host-govway*: è l'hostname con cui è raggiungibile l'istanza di Govway.
- *SoggettoSDI*: il soggetto interno al dominio come configurato durante l'esecuzione del govet di fatturazione passiva.
- *Versione*: versione della fattura che si sta inviando: FPA12 (Fattura 1.2 per Pubbliche Amministrazione), FPR12 (Fattura 1.2 per Privati), SDI11 e SDI10 (Fattura per Pubblica amministrazione versione 1.1. e 1.0).
- *TipoFile*: tipo di fattura: XML (Fattura firmata XADES), P7M (Fattura firmata CADES) o ZIP (archivio di fatture).
- *IdPaese e IdCodice*: dati del trasmittente della fattura.
- L'invocazione deve essere corredata dalle credenziali che sono state indicate durante la configurazione tramite il relativo govet.
- A seconda del tipo di fattura deve essere utilizzato il corretto header http *Content-Type*:
 - XML: è possibile utilizzare *text/xml* o *application/xml*
 - P7M: *application/pkcs7-mime*
 - XML: *application/zip*

Un esempio di invio di una fattura viene fornito tramite il seguente comando curl:

Soggetto Interno al Dominio

In questo esempio si suppone che il nome del soggetto (riferito precedentemente come *SoggettoSDI*) fornito durante la fase di installazione di GovWay sia *Ente*.

```
curl -X POST -basic --user SdIRiceviFile:123456 \
--data-binary @IT01234567890_11111.xml.p7m \
-H "Content-Type: application/pkcs7-mime" \
"http://127.0.0.1:8080/govway/sdi/out/xml2soap/Ente/CentroServiziFatturaPA/SdIRiceviFile/
  ↪v1?Versione=SDI10&TipoFile=P7M&IdPaese=IT&IdCodice=01629370097"
```

Nota: La generazione di un nome di file univoco da associare alla fattura viene gestita da GovWay.

È possibile disabilitare tale gestione disabilitando la proprietà “org.openspcoop2.protocol.sdi.fatturazioneAttiva.nomeFile.gestione” nel file “/etc/govway/sdi_local.properties”. Se viene disabilitata la funzionalità (attiva per default), la gestione dei nomi dei file (correttezza sintattica, univocità, ...) è demandata all'Applicativo Client che deve obbligatoriamente fornire il nome del file da associare alla fattura attraverso uno dei seguenti modi:

- query parameter “NomeFile”

- header http “SDI-NomeFile”
 - header http “GovWay-SDI-NomeFile”
-

6.2.2 Ricezione delle Notifiche dallo Sdi

Allo Sdi dovrà essere comunicata la seguente url che utilizzerà per inoltrare le notifiche:

```
https://<host-govway>/govway/sdi/in/<SoggettoSDI>/TrasmissioneFatture/v1
```

Le notifiche ricevute verranno consegnate secondo le modalità specificate durante l'esecuzione del Govlet. In fase di consegna verranno generati gli header descritti nella [Tabella 6.5](#)

Strumenti

7.1 Runtime

Questa sezione consente di visualizzare dati in tempo reale relativi al contesto di esecuzione del gateway, con la possibilità di effettuare alcune modifiche di stato. Le informazioni presenti sono:

- *Runtime*:
 - *Download*: consente di effettuare il download di un file di testo che contiene tutti i parametri visualizzati nella pagina.
 - *Svuota tutte le Cache*: consente di effettuare il reset contemporaneo di tutte le cache utilizzate dal gateway.
- *Informazioni Generali*: Informazioni sul prodotto e sul software di base.
- *Stato Servizi*: Consente di abilitare/disabilitare in tempo reale i servizi per l'elaborazione delle richieste in ingresso intra ed extra dominio.
- *Informazioni Diagnostica*: Riferimenti ai file di log attivi per il prodotto, con la possibilità di modificare in tempo reale il livello di verbosità degli stessi.
- *Informazioni Tracciamento*: Riferimenti ai file contenenti il tracciamento delle richieste in elaborazione sul gateway, con la possibilità di abilitare/disabilitare le specifiche fonti.
- *Informazioni Database*: Informazioni relative la piattaforma database adottata.
- *Informazioni SSL*: Informazioni sulla configurazione SSL.
- *Informazioni Internazionalizzazione*: Informazioni sulla configurazione del servizio di internazionalizzazione.
- *Informazioni Timezone*: Timezone attivo.
- *Informazioni Java Networking*: Parametri di configurazione inerenti la configurazione del networking a livello Java.
- *Informazioni Modalità Gateway*: Contesti configurati per ciascuna specifica modalità operativa.
- *Cache*: Parametri di configurazione di tutte le cache adottate dal gateway, con la possibilità di effettuare il reset di ciascuna singolarmente.

- *Connessioni Attive*: Evidenza in tempo reale delle connessioni attive verso altri software a supporto (database, broker jms, ecc.)
- *Transazioni Attive*: Riferimenti alle transazioni in corso di elaborazione.
- *Connessioni HTTP Attive*: Evidenza in tempo reale delle connessioni HTTP, aperte in uscita, per l'elaborazione delle richieste in corso.

7.2 Auditing

La funzionalità di *auditing* consente di tracciare il comportamento degli utenti della govwayConsole, al fine di verificare le operazioni eseguite e i loro effetti.

Per gli aspetti di configurazione della funzionalità di auditing si rimanda alla sezione [Auditing](#).

In questa sezione descriviamo le interfacce della govwayConsole dedicate alla consultazione delle informazioni raccolte tramite il servizio di auditing.

Gli utenti della govwayConsole aventi il permesso [A] Auditing (vedi [Utenti](#)) hanno accesso alla funzionalità di consultazione dei dati presenti nel repository del servizio di auditing.

Per accedere al servizio di consultazione selezionare la voce **Auditing** nella sezione **Reportistica** del menu laterale sinistro. La consultazione dei dati di auditing avviene tramite ricerche effettuate impostando i criteri attraverso il form riportato in [Fig. 7.1](#).

Vediamo adesso il significato dei parametri per la ricerca dei dati di auditing:

- *Criteri di Ricerca*
 - **Inizio Intervallo**: Data iniziale che serve ad impostare l'intervallo temporale su cui restringere la ricerca dei dati di auditing. Lasciare il campo vuoto equivale all'impostazione *illimitato*.
 - **Fine Intervallo**: Data finale che serve ad impostare l'intervallo temporale su cui restringere la ricerca dei dati di auditing. Lasciare il campo vuoto equivale all'impostazione *illimitato*.
 - **Utente**: Consente di restringere la ricerca alle sole operazioni effettuate da un determinato utente. Il campo lasciato vuoto equivale a *qualsiasi utente*.
- *Operazione*
 - **Tipo**: Filtro per tipo di operazione, distinguendo tra:
 - * *ADD*: creazione di un'entità
 - * *CHANGE*: modifica di un'entità
 - * *DEL*: cancellazione di un'entità
 - * *LOGIN*: accesso alla govwayConsole
 - * *LOGOUT*: disconnessione dalla govwayConsole
 - **Stato**: Filtro in base allo stato dell'operazione, distinguendo tra:
 - * *requesting*: in fase di richiesta
 - * *error*: terminata con errore
 - * *completed*: terminata correttamente
- *Oggetto*

Reportistica > **Auditing**

Criteri di Ricerca

Inizio intervallo	<input type="text"/>
Indicare una data nel formato 'yyyy-MM-dd'	
Fine intervallo	<input type="text"/>
Indicare una data nel formato 'yyyy-MM-dd'	
Utente	<input type="text"/>

Operazione

Tipo	<input type="text"/>
Stato	<input type="text"/>

Oggetto

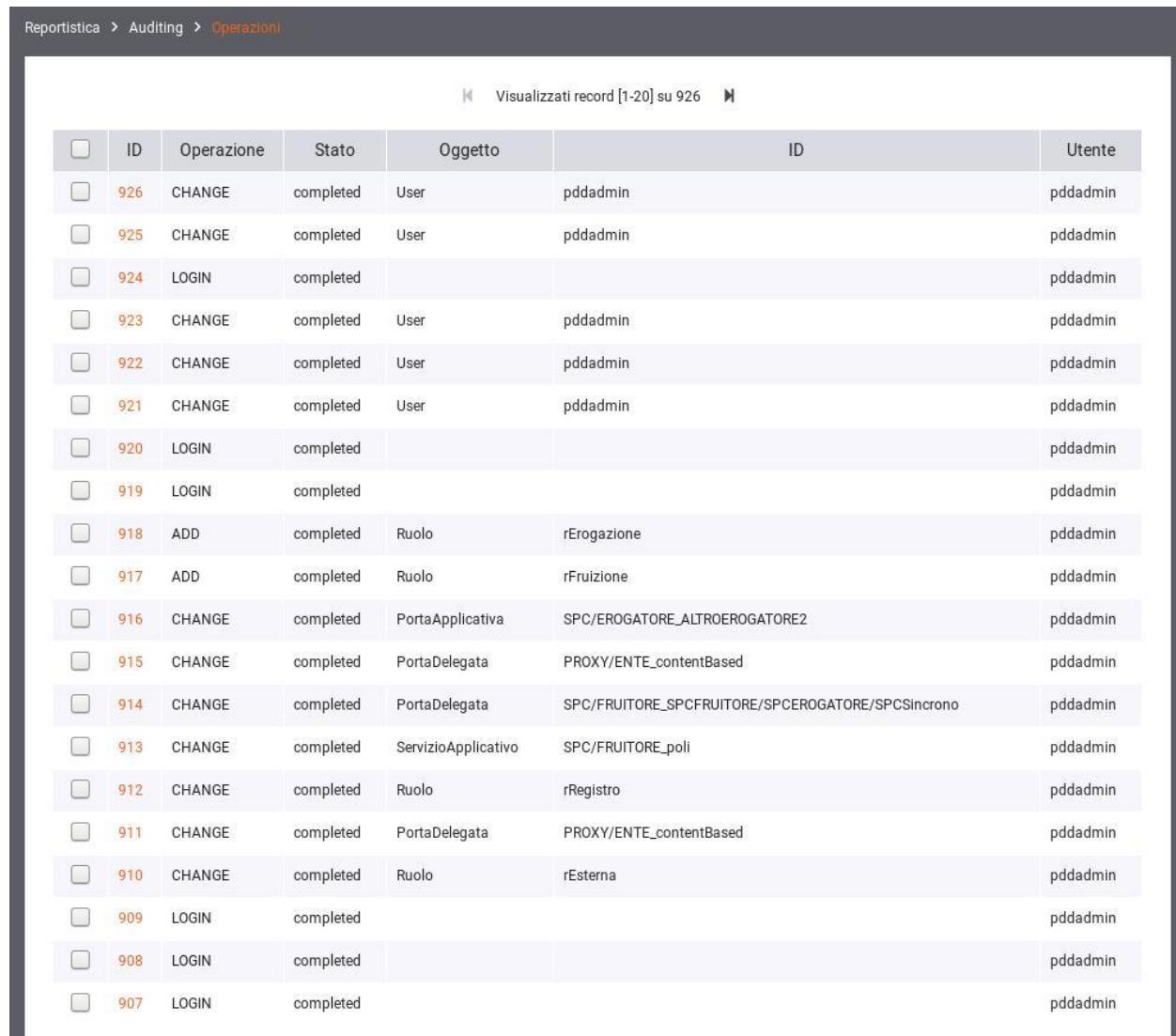
Tipo	<input type="text"/>
Identificativo	<input type="text"/>
Id precedente alla modifica	<input type="text"/>
Contenuto	<input type="text"/>

Invia **Cancella**

Figure7.1: Maschera di ricerca dei dati di auditing

- **Tipo:** campo per restringere la ricerca alle sole operazioni riferite ad un determinato tipo di entità. Il campo è costituito da una lista a discesa popolata con tutte le tipologie di entità gestite dalla govwayConsole.
- **Identificativo:** campo testuale per restringere la ricerca alle sole operazioni effettuate su una specifica entità. La composizione dell'identificativo cambia in base alla tipologia dell'entità. Ad esempio un soggetto è identificato attraverso il tipo e il nome: Tipo/NomeSoggetto.
- **Id precedente alla modifica:** campo testuale analogo al precedente ma utile in quei casi in cui l'operazione che si sta cercando ha modificato i dati che compongono l'identificativo.
- **Contenuto:** pattern per la ricerca sul contenuto dell'entità associata all'operazione. Per utilizzare questo criterio di filtro il servizio di auditing deve essere configurato in modo da effettuare il dump degli oggetti.

Una volta effettuata la ricerca viene mostrata una pagina con la lista dei risultati corrispondenti (vedi Fig. 7.2).



	ID	Operazione	Stato	Oggetto	ID	Utente
	926	CHANGE	completed	User	pddadmin	pddadmin
	925	CHANGE	completed	User	pddadmin	pddadmin
	924	LOGIN	completed			pddadmin
	923	CHANGE	completed	User	pddadmin	pddadmin
	922	CHANGE	completed	User	pddadmin	pddadmin
	921	CHANGE	completed	User	pddadmin	pddadmin
	920	LOGIN	completed			pddadmin
	919	LOGIN	completed			pddadmin
	918	ADD	completed	Ruolo	rErogazione	pddadmin
	917	ADD	completed	Ruolo	rFruizione	pddadmin
	916	CHANGE	completed	PortaApplicativa	SPC/EROGATORE_ALTROEROGATORE2	pddadmin
	915	CHANGE	completed	PortaDelegata	PROXY/ENTE_contentBased	pddadmin
	914	CHANGE	completed	PortaDelegata	SPC/FRUITORE_SPCFRUITORE/SPCEROGATORE/SPCSincrono	pddadmin
	913	CHANGE	completed	ServizioApplicativo	SPC/FRUITORE_poli	pddadmin
	912	CHANGE	completed	Ruolo	rRegistro	pddadmin
	911	CHANGE	completed	PortaDelegata	PROXY/ENTE_contentBased	pddadmin
	910	CHANGE	completed	Ruolo	rEsterna	pddadmin
	909	LOGIN	completed			pddadmin
	908	LOGIN	completed			pddadmin
	907	LOGIN	completed			pddadmin

Figure7.2: Risultato della ricerca dei dati di auditing

Ciascun elemento della lista riporta i dati principali che identificano l'operazione. Selezionando l'identificatore dell'operazione si visualizzano i dati di dettaglio (vedi Fig. 7.3). Dal dettaglio dell'operazione, se è attivo il dump,

si può visualizzare il dettaglio dell'entità coinvolta nell'operazione e gli eventuali documenti binari (ad esempio i file WSDL associati ad un accordo di servizio).

Reportistica > Auditing > Operazioni > **Dettaglio di 916**

Dettaglio Operazione

Time request	2017-08-11 11:12:34.834
Time execute	2017-08-11 11:12:34.917
Tipo operazione	CHANGE
Tipo oggetto	PortaApplicativa
Identificativo	SPC/EROGATORE_ALTROEROGATORE2
Utente	pddadmin
Stato	completed

Documenti Binari (0)

Figure7.3: Dettaglio di una traccia di auditing

Configurazione

Nella sezione del menu *Configurazione* si raggiungono le funzionalità per modificare i parametri di configurazione del gateway.

8.1 Generale

La sezione *Configurazione > Generale* consente di impostare i parametri generali per le funzionalità di base del gateway (Fig. 8.1). In particolare è possibile:

- Attivare e configurare la modalità Multi-Tenant. Abilitando questa modalità sarà ammessa la creazione di ulteriori soggetti interni al dominio GovWay.
- Configurare le Base URL utilizzate per visualizzare le URL di invocazione delle API
- Configurare la gestione del CORS (*cross-origin HTTP request (CORS)*) a livello globale valido per tutte le APIs
- Configurare il Caching Risposta a livello globale valido per tutte le APIs
- Configurare i profili fornendo i riferimenti ai servizi di base per l'elaborazione dei messaggi ed al soggetto interno
- Attivare e configurare la modalità Canali in una installazione composta da più nodi in Load Balancing. Abilitando questa modalità sarà possibile assegnare uno o più canali ad ogni nodo che compone il cluster e suddividere le API in canali di appartenenza. Su ogni nodo saranno autorizzate ad essere invocate solamente le API che possiedono un canale corrispondente alla configurazione del nodo.
- Configurare proprietà di sistema

Configurazione Generale

Note: (*) Campi obbligatori

Multi-Tenant

Stato

URL di Invocazione API

Base URL *****

Base URL Fruizione

[Regole Proxy Pass \(3\)](#)

Gestione CORS

Stato

Access Control

All Allow Origins

Allow Credentials

Allow Methods *****

Allow Request Headers *****

Expose Response Headers

Caching Risposta

Stato

Gestione Profilo

API Gateway

Soggetto ENTE
[Visualizza Dati Soggetto](#)

Modi PA

Soggetto ENTE
[Visualizza Dati Soggetto](#)

Canali

Stato

Proprietà di Sistema

[visualizza](#)

Figure8.2: Maschera per l'impostazione dei parametri di configurazione generale (2/2)

8.1.1 Multi-Tenant

Per abilitare la modalità multi-tenant è sufficiente selezionare il valore «abilitato» sull'elemento Stato.

Dopo aver abilitato l'opzione multi-tenant è possibile creare nuovi soggetti interni al dominio, come indicato alla sezione *Creazione di un soggetto*. In questo contesto, i soggetti avranno come elemento distintivo il dominio, che può essere *Interno* o *Esterno*.

I dettagli sulla configurazione dell'opzione multi-tenant sono riportati nella sezione *Multi-Tenant*.

8.1.2 URL di Invocazione API

Nella sezione “URL di Invocazione API” descritta nella figura Fig. 8.3 è possibile configurare i seguenti aspetti:

- *Base URL*: Indica il prefisso utilizzato per visualizzare le URL di Invocazione delle API.
- *Base URL Fruizione*: permette di differenziare il prefisso utilizzato per visualizzare le URL di Invocazione delle fruizioni dalle erogazioni.
- *Regole Proxy Pass*: tramite questa voce è possibile ridefinire le URL di Invocazioni, per specifiche fruizioni e/o erogazioni, allineandole a regole configurate su un revero proxy che media le comunicazioni http con GovWay (maggiori dettagli disponibili nella sezione *Regole Proxy Pass*).
- *Esporta Configurazione*: consente di esportare per intero la configurazione relativa all'url di invocazione, comprensiva di eventuali regole di proxy pass.

L'url di invocazione di una API su GovWay, al netto di eventuali ridefinizioni definite tramite *Regole Proxy Pass*, segue una convenzione di naming che varia tra fruizione ed erogazione e in base al profilo di interoperabilità, come descritto nella sezione *Contesto di una API erogata o fruita*.

URL di Invocazione API

Base URL *

Base URL Fruizione

[Regole Proxy Pass \(6\)](#)

[Esporta Configurazione](#)

Figure8.3: Configurazione URL di Invocazione API

Contesto di una API erogata o fruita

L'url di invocazione di una API su GovWay, al netto di eventuali ridefinizioni definite tramite *Regole Proxy Pass*, segue una convenzione di naming che varia tra fruizione ed erogazione e in base al profilo di interoperabilità.

Nel seguito di questa documentazione verrà indicata con *<prefix-erogazione>* e *<prefix-fruizione>* rispettivamente la base url configurata per le erogazioni e per le fruizioni, come descritto nella sezione *URL di Invocazione API*.

erogazioni

<prefix-erogazione>/<profilo>[/in]/<soggettoDominioInterno>/<nomeErogazione>/v<versioneErogazione>

- *<profilo>* assume i seguenti differenti valori in funzione del profilo di interoperabilità a cui l'API appartiene:
 - *api*: per il profilo “API Gateway”;
 - *soap*: per API di tipo SOAP su profilo “ModI”;
 - *rest*: per API di tipo REST su profilo “ModI”;
 - *spcoop*: per il profilo “SPCoop”;
 - *sdi*: per il profilo “Fatturazione Elettronica”;
 - *as4*: per il profilo “eDelivery”;
- *<in>* indica una erogazione di API; il contesto */in* può essere omesso.

fruizioni

<prefix-fruizione>/<profilo>/out/<soggettoDominioInterno>/<soggettoErogatore>/<nomeFruizione>/v<versioneFruizione>

- *<profilo>* assume uno dei valori descritti per l'erogazione;
- *<out>* indica una fruizione di API; per il profilo di interoperabilità “Fatturazione Elettronica” deve essere usato “out/xml2soap” al posto di “out”.

Nota: Nell'URL di invocazione può essere omesso anche il profilo, sia per un'erogazione che per una fruizione; in tal caso, l'API verrà ricercata all'interno del profilo “API Gateway”.

Regole Proxy Pass

Questa sezione permette di ridefinire la modalità di visualizzazione delle Url di Invocazione delle API esposte da GovWay per assicurare che, in presenza di un reverse proxy che media le comunicazioni http con GovWay, sia possibile configurare opportunamente le url di invocazione delle API esposte da GovWay allineandole con le eventuali configurazioni specifiche realizzate sul reverse proxy.

Nota: La funzionalità permette di configurare come vengono visualizzate le URL di Invocazione sulla govwayConsole, per allinearsi ad un eventuale reverse proxy che media le comunicazioni http con GovWay. Le API, su GovWay, rimangono raggiungibili solamente sulle url originali e dovrà essere il reverse proxy ad effettuare la conversione rispetto a quella esposta.

Le regole create sono visualizzate nella forma di elenco ordinato (Fig. 8.4). L'icona iniziale di ciascun elemento consente di modificarne la posizione. Per ogni regola viene visualizzato il suo stato, il nome e la descrizione.

Ordine	Stato	Nome	Descrizione
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<u>Domibus</u>	Servizio di ricezione dei messaggi AS4 dell'Access Point Domibus
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<u>ServizioAnagrafica</u>	Ridefinisce le url di invocazione per l'Anagrafica
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<u>HostProduzioneErogazioniModIP</u>	Ridefinisce l'hostname utilizzato per le erogazioni ModIP

Figure8.4: Lista Regole Proxy Pass

Per ogni regola (Fig. 8.5) deve essere obbligatoriamente definita una stringa libera o una espressione regolare utilizzata per individuare l'applicabilità della regola attraverso un confronto con il contesto dell'API. Il contesto è l'URL di Invocazione dell'API senza il prefisso Base URL. Inoltre per ogni regola è possibile indicare altri criteri di applicabilità opzionali quali eventuali profilo di interoperabilità, un soggetto, una tipologia (fruizione/erogazione) o un tipo di api (soap/rest).

Il dettaglio dei campi associati ad una regola sono raggruppati in tre sottosezioni:

Informazioni generiche:

- *Nome*: Identificativo della regola
- *Stato*: Indica se la regola è abilitata o meno.
- *Descrizione*: (Opzionale) Descrizione generica della regola

Le regole di applicabilità vengono definite dai seguenti campi:

Configurazione Generale > Regole di Proxy Pass > **HostProduzioneErogazioniModIPA**

HostProduzioneErogazioniModIPA

Note: (*) Campi obbligatori

Regola

Nome *	HostProduzioneErogazioniModIPA
Stato	abilitato
Descrizione	Ridefinisce l'hostname utilizzato per le erogazioni Mod IPA

Criteri di Applicabilità

Espressione Regolare	<input checked="" type="checkbox"/>
Regola *	.+/in/(.+)/(.+)/v(.+)
Profilo	Modi PA
Ruolo	Erogazione
Tipo API	Rest

Nuova URL di Invocazione

Base URL	http://\${0}/
Contesto	v\${2}/api/\${1}

SALVA

Figure8.5: Creazione Regola Proxy Pass

- *Espressione Regolare*: Indica se la regola sottostante è una espressione regolare o una stringa libera.
- *Regola*: Stringa libera o espressione regolare.
 - L'espressione regolare viene utilizzata per verificarne il match sull'contesto dell'API (url di invocazione senza la Base URL)
 - Nel caso di stringa libera si ha un'applicabilità se il contesto dell'API (url di invocazione senza la Base URL) inizia con la stringa fornita.
- *Profilo*: (Opzionale) Profilo di Interoperabilità per il quale si applica la regola
- *Soggetto*: (Opzionale) Soggetto interno per il quale si applica la regola
- *Ruolo*: (Opzionale) Tipologia di API (Erogazione/Fruizione) per il quale si applica la regola
- *Tipo API*: (Opzionale) Tipo di API (REST/SOAP) per il quale si applica la regola

La nuova url di invocazione viene definita attraverso i campi “Base URL” e “Contesto”.

- *Base URL*: Permette di ridefinire la Base URL utilizzata rispetto a quanto definito nella configurazione generale
- *Contesto*: Indica il contesto da utilizzare dopo la Base URL

Nei campi “Base URL” e “Contesto” è possibile utilizzare le seguenti informazioni dinamiche:

- Se è stata fornita una espressione regolare, nei due campi possono essere utilizzati le keyword “\${posizione}” per impostare un valore dinamico individuato tramite l'espressione regolare fornita. Il primo match, all'interno dell'espressione regolare, è rappresentata da “\${0}” (Ad esempio: [http://server:8080/\\${0}/altro/\\${1}/](http://server:8080/${0}/altro/${1}/))
- Se è abilitata la gestione dei canali (vedi *Canali*) è possibile utilizzare la keyword “\${canale}” per impostare l'identificativo del canale associato all'API. Maggiori esempi vengono forniti nella sezione *Url di Invocazione e Canali*.
- È possibile utilizzare la keyword “\${tag}” per impostare l'identificativo del tag associato all'API. Poichè ad un'API è possibile associare più tag verrà utilizzato quello alla prima posizione ma è possibile indicarne uno differente tramite l'espressione \${tag[posizione]}. Il primo tag, all'interno della lista, è rappresentata da “\${tag[0]}”, ad esempio: [http://server:8080/\\${tag\[0\]}/](http://server:8080/${tag[0]}/)

Esempio 1

Tutte le API REST erogate dal Soggetto “ENTE” tramite il profilo “ModI” possiedono nell'installazione di default la seguente URL di Invocazione:

- <http://localhost:8080/rest/in/ENTE/NomeAPI/v1>

Per modificare la url di invocazione in modo da spostare il nome del soggetto come hostname, e rimodulare il contesto in modo da visualizzare prima la versione, è possibile utilizzare la seguente configurazione di proxy pass:

Criteri di Applicabilità:

- Espressione Regolare: true
- Regola: .+/in/(.+)/(.+)/v(.+)
- Profilo: ModI
- Soggetto: ENTE
- Ruolo: Erogazione
- Tipo API: REST

Nuova URL di Invocazione

- Base URL: [http://\\${0}/](http://${0}/)
- Contesto: v\${2}/api/\${1}

L'url di invocazione prodotta sarà:

- <http://ENTE/v1/api/NomeAPI>

Esempio 2

Supponiamo di voler modificare l'url di invocazione dell'API "PetStore" versione 1 erogata dal soggetto "ENTE" tramite il profilo di interoperabilità "ModI". Nell'installazione di default viene fornita la seguente URL di Invocazione:

- <http://localhost:8080/rest/in/ENTE/PetStore/v1>

Lo scopo è quello di eliminare il nome del soggetto e di togliere la "v" dalla versione. Per farlo è possibile utilizzare la seguente configurazione di proxy pass:

Criteri di Applicabilità:

- Espressione Regolare: false
- Regola: /rest/in/ENTE/PetStore/v1
- Profilo: ModI
- Soggetto: Qualsiasi
- Ruolo: Qualsiasi
- Tipo API: Qualsiasi

Nuova URL di Invocazione

- Base URL:
- Contesto: /rest/in/PetStore/1

L'url di invocazione prodotta sarà:

- <http://localhost:8080/rest/in/PetStore/1>

8.1.3 Gestione CORS

In GovWay è possibile abilitare la gestione del CORS (*cross-origin HTTP request (CORS)*) globalmente in modo che sia valido per tutte le APIs.

La configurazione permette di specificare i seguenti parametri:

- *Stato*: Indicazione se la gestione del CORS è abilitata o meno globalmente su GovWay.
- *Access Control*: tutti i parametri seguenti permettono di configurare il CORS. Per il dettaglio su cosa significa ogni singola voce si rimanda alla specifica CORS <https://www.w3.org/TR/cors/>.
 - *All Allow Origins*: se abilitato, in tutte le risposte viene aggiunto un header http "Access-Control-Allow-Origin" valorizzato con "*"
 - *Allow Origins*: nel caso non venga abilitato il parametro precedente, deve essere indicato una lista di origin che vengono impostate nell'header http "Access-Control-Allow-Origin" aggiunto in ogni risposta
 - *All Allow Methods*: se abilitato, in tutte le risposte di una Preflight Request (OPTIONS) viene aggiunto un header http "Access-Control-Allow-Methods" valorizzato con i metodi richiesti dall'header "Access-Control-Request-Method" della richiesta. L'opzione è attivabile solamente se la voce "All Allow Origins" risulta disabilitata
 - *Allow Methods*: nel caso non venga abilitato il parametro precedente, deve essere indicato una lista di metodi che vengono impostati nell'header http "Access-Control-Allow-Methods" di una risposta Preflight Request (OPTIONS)

- *All Allow Request Headers*: se abilitato, in tutte le risposte di una Preflight Request (OPTIONS) viene aggiunto un header http “Access-Control-Allow-Headers” valorizzato con gli header http richiesti dall’header “Access-Control-Request-Headers” della richiesta. L’opzione è attivabile solamente se la voce “All Allow Origins” risulta disabilitata
- *Allow Request Headers*: nel caso non venga abilitato il parametro precedente, deve essere indicato una lista di header che vengono impostati nell’header http “Access-Control-Allow-Headers” di una risposta Preflight Request (OPTIONS)
- *Allow Credentials*: se abilitato o disabilitato viene impostato relativamente il valore true o false nell’header “Access-Control-Allow-Credentials”
- *Expose Response Headers*: abilita l’accesso a specifici headers, presenti nella risposta, da parte dei client.

Gestione CORS

Stato	<input type="text" value="abilitato"/>
Access Control	
All Allow Origins	<input type="checkbox"/>
Allow Origins *	<input type="text" value="https://www.test-cors.org"/>
All Allow Methods	<input type="checkbox"/>
Allow Methods *	<input type="text" value="GET x PUT x POST x DELETE x PATCH x"/>
All Allow Request Headers	<input type="checkbox"/>
Allow Request Headers *	<input type="text" value="Authorization x Content-Type x SOAPAction x Cache-Control x"/>
Allow Credentials	<input type="checkbox"/>
Expose Response Headers	<input type="text"/>

Figure8.6: Maschera di configurazione generale del CORS

8.1.4 Caching Risposta

In GovWay è possibile abilitare il salvataggio delle risposte in una cache globalmente in modo che sia attivo per tutte le APIs. Questa funzionalità permette ad un backend server di non dover riprocessare le stesse richieste più volte.

La configurazione permette di specificare i seguenti parametri:

- *Stato*: Indicazione se il salvataggio delle risposte in cache è abilitata o meno globalmente su GovWay.
- *Cache Timeout (secondi)*: intervallo di tempo, definito in secondi, per il quale la risposta salvata in cache viene utilizzata come risposte a successive richieste di un client.
- *Dimensione Max Risposta*: se abilitato deve essere definita la dimensione massima (in kb) che una risposta può avere per essere salvata in cache.
- *Generazione Hash*: ad ogni risposta salvata in cache viene associato un valore hash calcolato rispetto ai dati della richiesta che risultano abilitati tra le opzioni seguenti:
 - *URL di Richiesta*: viene utilizzata la URL della richiesta per il calcolo dell'hash.
 - *Payload*: viene utilizzato il payload della richiesta per il calcolo dell'hash.
 - *Headers*: vengono utilizzati gli header della richiesta indicati per il calcolo dell'hash. L'abilitazione di questa opzione comporta l'aggiunta di un elemento per consentire di specificare gli headers da selezionare.
- *Cache Control*: opzioni aggiuntive per la gestione della cache basate sul header HTTP «Cache-Control»:
 - *No Cache*: consente di attivare l'utilizzo della direttiva «no-cache» al fine di effettuare una richiesta evitando di ottenere una risposta dalla cache.
 - *Max Age*: consente di attivare l'utilizzo della direttiva «max-age» che consente di forzare il tempo di vita, al valore fornito, della risposta inserita in cache.
 - *No Store*: consente di attivare l'utilizzo della direttiva «no-store» che consente di impedire l'inserimento in cache della risposta generata dalla richiesta corrente.

Dopo aver salvato la configurazione fornita per il caching della risposta, appare la sezione *Configurazione Avanzata* che comprende il link *Regole*. Seguendo tale link è possibile definire ulteriori criteri avanzati per la gestione della cache.

Nota: In presenza di regole avanzate di configurazione, le risposte salvate in cache saranno solamente quelle che hanno un match con i criteri definiti in una regola.

Come si vede in Fig. 8.8 ciascuna regola è composta dai seguenti campi:

- *Codice Risposta*: codice HTTP ottenuto in risposta. Sono disponibili per la scelta le seguenti opzioni:
 - *Qualsiasi*: indica qualunque valore del codice HTTP restituito
 - *Singolo*: consente di specificare un singolo valore del codice HTTP restituito
 - *Intervallo*: consente di fornire l'intervallo dei valori ammessi per il codice HTTP restituito
- *Cache Timeout (Secondi)*: indica in secondi il timeout applicato agli elementi in cache relativamente ai codici HTTP che soddisfano la regola.
- *Fault*: opzione per specificare se anche i messaggi di fault devono essere inseriti in cache.

Caching Risposta

Stato	abilitato
Cache Timeout (secondi)	300
Dimensione Max Risposta	<input checked="" type="checkbox"/>
Dimensione Max (kb)	1

Generazione Hash

URL di Richiesta	abilitato
Payload	abilitato
Headers	disabilitato

Cache Control

No Cache	<input checked="" type="checkbox"/>
Max Age	<input checked="" type="checkbox"/>
No Store	<input checked="" type="checkbox"/>

Figure8.7: Maschera di configurazione per il Caching della Risposta

Regola

Codice Risposta	Qualsiasi
Cache Timeout (Secondi)	
Fault	<input type="checkbox"/>

SALVA

Figure8.8: Inserimento di una regola per il Caching della Risposta

8.1.5 Profili

Questa sezione viene visualizzata solamente se non è attiva la modalità Multi-tenant. Per ciascun Profilo di Interoperabilità, attivo sul gateway, viene visualizzato il nome del Soggetto interno che eroga/fruisce. Subito sotto il soggetto è presente un collegamento che porta al form di editing del soggetto visualizzato.

8.1.6 Canali

In GovWay è possibile attivare, in una installazione composta da più nodi in Load Balancing, una suddivisione delle API tra i vari nodi utilizzando il concetto di canale, al fine di suddividere il carico tra i nodi. Per maggiori dettagli sull'installazione in Load Balancing si faccia riferimento alla sezione cluster della Guida di Installazione.

Abilitando la modalità “Canali” sarà possibile assegnare uno o più canali ad ogni nodo che compone il cluster ed un canale ad ogni API. Su ogni nodo saranno autorizzate ad essere invocate solamente le API che possiedono un canale corrispondente alla configurazione del nodo.

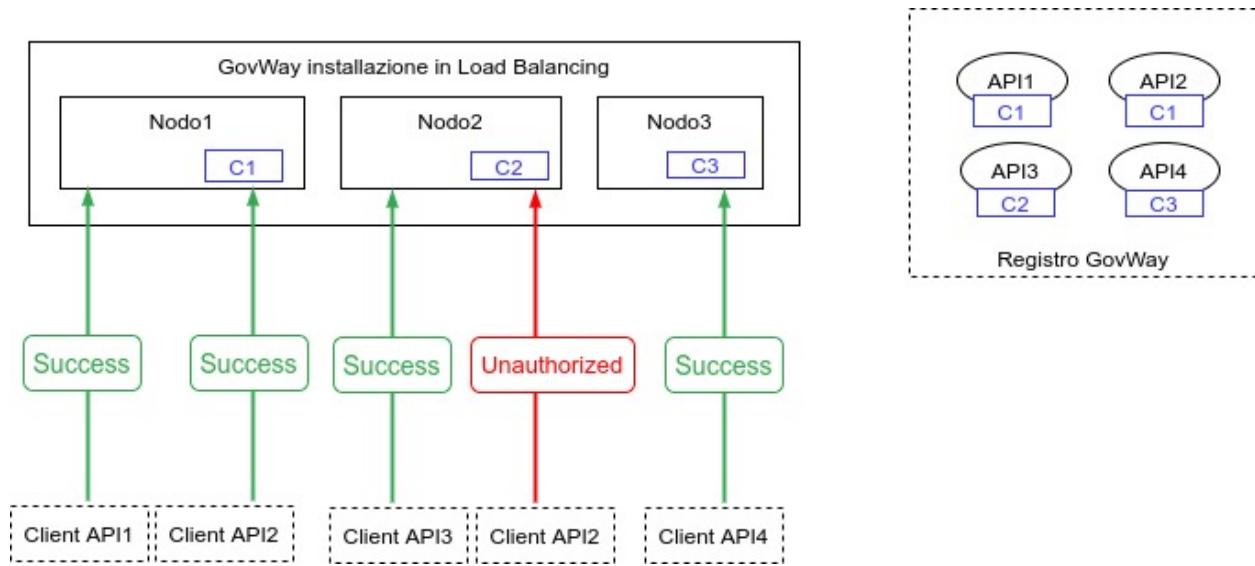


Figure8.9: Suddivisione delle API in Canali

Nelle prossime sezioni verranno descritte:

- *Configurazione dei Canali*: vengono fornite le indicazioni su come abilitare la funzionalità e su come censire i canali ed associarli ai nodi che compongono il cluster.
- *Canale associato all'API*: vengono fornite le indicazioni su come associare ad una erogazione o fruizione di API un canale differente da quello di default.
- *Url di Invocazione e Canali*: viene descritto come personalizzare le url di invocazione visualizzate dalla console per ogni erogazione o fruizione al fine di indirizzare il nodo corretto corrispondente al canale associato all'API.

Configurazione dei Canali

Abilitando la modalità “Canali” sarà possibile assegnare uno o più canali ad ogni nodo che compone il cluster ed un canale ad ogni API.

La prima volta che viene abilitata la funzionalità, la console richiede di configurare un canale di default che verrà associato:

- a tutte le erogazioni o fruizioni di API esistenti alle quali non è stato associato un canale
- a tutti i nodi non registrati

La configurazione richiede (Fig. 8.10):

- *Stato*: indicazione se la gestione è abilitata o meno;
- *Nome*: identificativo univoco del canale di default;
- *Descrizione*: descrizione generica del canale di default.

The screenshot shows a configuration interface for 'Canali'. At the top, there is a dropdown menu labeled 'Stato' with the value 'abilitato'. Below this, a section titled 'Canale di Default' contains two fields: 'Nome' with a red asterisk and the value 'C1', and 'Descrizione' with an empty text area.

Figure8.10: Maschera di abilitazione delle gestione dei canali

Attivata la gestione e definito il canale di default sarà possibile registrare nuovi canali, modificare il canale di default e registrare i nodi che compongono il cluster (Fig. 8.11).

The screenshot shows a management interface for 'Canali'. It includes a dropdown menu for 'Stato' (set to 'abilitato') and another dropdown for 'Default' (set to 'C1'). Below these, there are two links: 'Canali (1)' and 'Nodi (0)'.

Figure8.11: Maschera di gestione dei canali

Dall’elenco dei canali è possibile aggiungere, modificare o eliminare un canale (Fig. 8.12). La registrazione di un

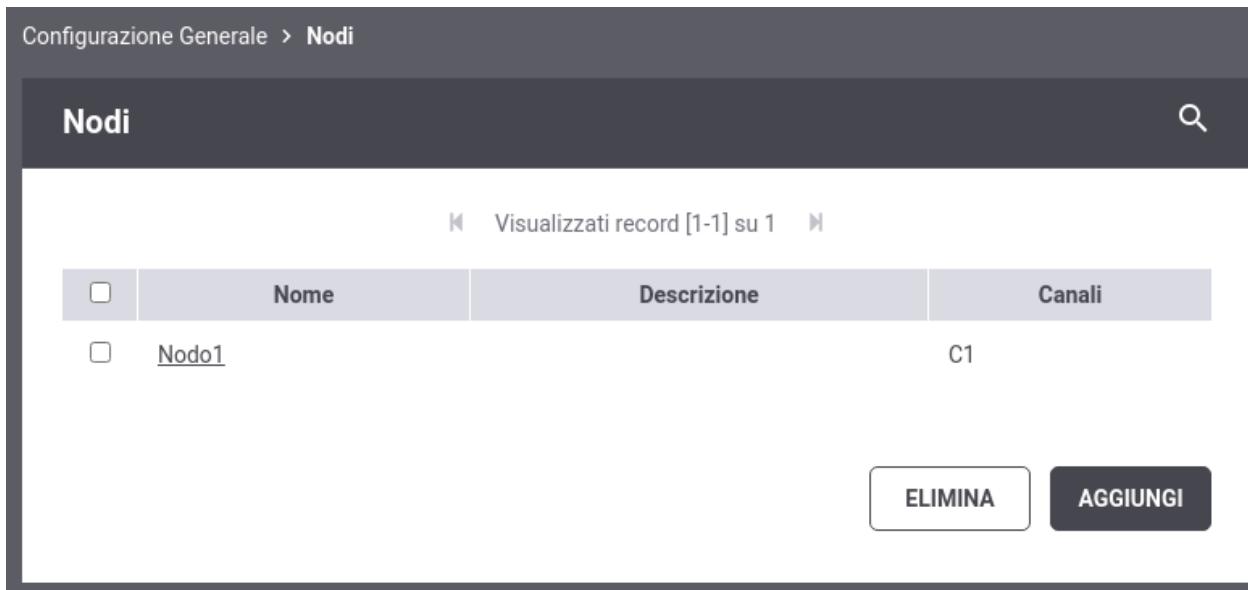
nuovo canale richiede che venga definito un identificativo univoco e opzionalmente una descrizione da associare al canale.



	Nome	Descrizione	Default	Uso
<input type="checkbox"/>	<u>C1</u>		Si	
<input type="checkbox"/>	<u>C2</u>		No	

Figure8.12: Elenco dei canali configurati

Dall'elenco dei nodi è possibile registrare, modificare o eliminare un nodo del cluster (Fig. 8.12).



	Nome	Descrizione	Canali
<input type="checkbox"/>	<u>Node1</u>		C1

Figure8.13: Elenco dei nodi configurati

La registrazione o la modifica di un nodo richiede (Fig. 8.14):

- *Nome*: identificativo univoco del nodo;
- *Descrizione*: descrizione generica del nodo;
- *Canali*: selezione dei canali associati al nodo.

Configurazione Generale > Nodi > Aggiungi

Note: (*) Campi obbligatori

Nodo

Nome *

Descrizione

Canali *

SALVA

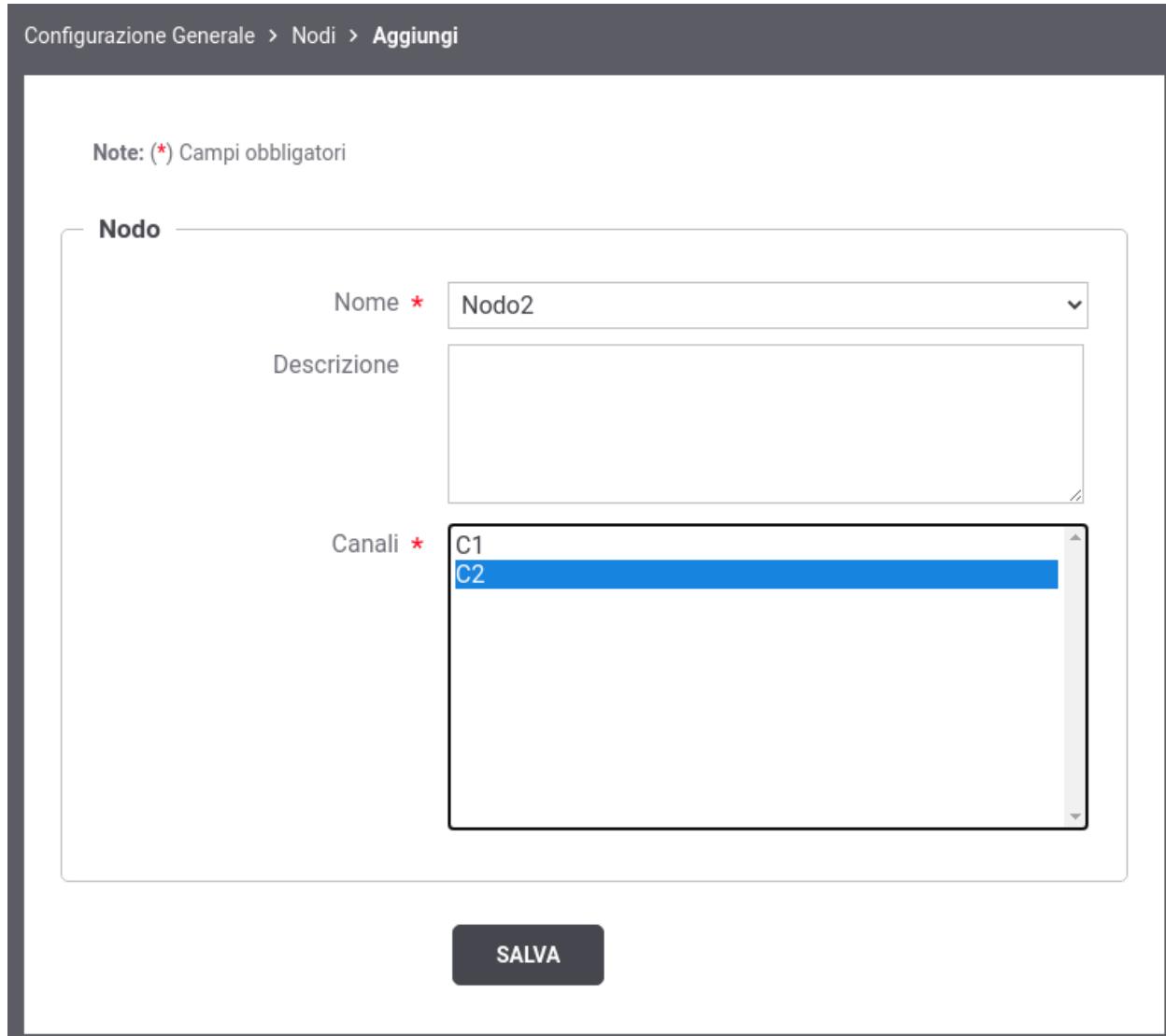


Figure8.14: Registrazione di un nodo

Canale associato all'API

Una volta abilitata la modalità “Canali” accedendo all’elenco delle API (Fig. 8.15) o delle Erogazioni/Fruizioni (Fig. 8.16) verrà visualizzata l’informazione sul canale associato. Tutte le erogazioni/fruizioni esistenti in cui non è stato associato un canale specifico ereditano il canale di default.

API		Nome		
		Visualizzati record [1-4] su 4		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Api1 v1	Examples	<input type="button" value=""/>
		API Rest Open API 3, Canale: C1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Api2 v1	Examples	<input type="button" value=""/>
		API Rest Open API 3, Canale: C1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Api3 v1	Examples	<input type="button" value=""/>
		API Rest Open API 3, Canale: C1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Api4 v1	Examples	<input type="button" value=""/>
		API Rest Open API 3, Canale: C1		

Figure 8.15: Elenco API visualizza l’informazione sul Canale

Un canale, differente da quello di default, può essere associato ad una erogazione o fruizione in due modi:

- definendo un canale nell’API: tutte le erogazioni o fruizioni che implementano l’API ereditano il canale
- associando il canale alla specifica erogazione o fruizione

Sia durante la registrazione di una nuova API che durante l’attivazione di una nuova erogazione o fruizione verrà richiesto all’utente se desidera specificare un canale differente da quello di default (Fig. 8.17).

È possibile modificare il canale associato ad una API esistente accedendo alla maschera di dettaglio dell’API (Fig. 8.18) e cliccando sulla voce “Modifica Canale” si accede ad una maschera identica a quella proposta in fase di creazione (vedi Fig. 8.17).

In ugual modo è possibile associare un canale ad una specifica erogazione o fruizione accedendo alla sua maschera di dettaglio (Fig. 8.19) e cliccando sulla voce “Modifica Canale”.

Erogazioni

Visualizzati record [1-4] su 4

Erogazioni

● **Api1@ENTE v1** Examples
API Rest: Api1 v1, Canale: C1

● **Api2@ENTE v1** Examples
API Rest: Api2 v1, Canale: C1

● **Api3@ENTE v1** Examples
API Rest: Api3 v1, Canale: C1

● **Api4@ENTE v1** Examples
API Rest: Api4 v1, Canale: C1

Figure8.16: Elenco Erogazioni visualizza l'informazione sul Canale

Canale	<input type="text" value="ridefinito"/>
	<input type="text" value="C2"/>

Figure8.17: Associazione di un canale

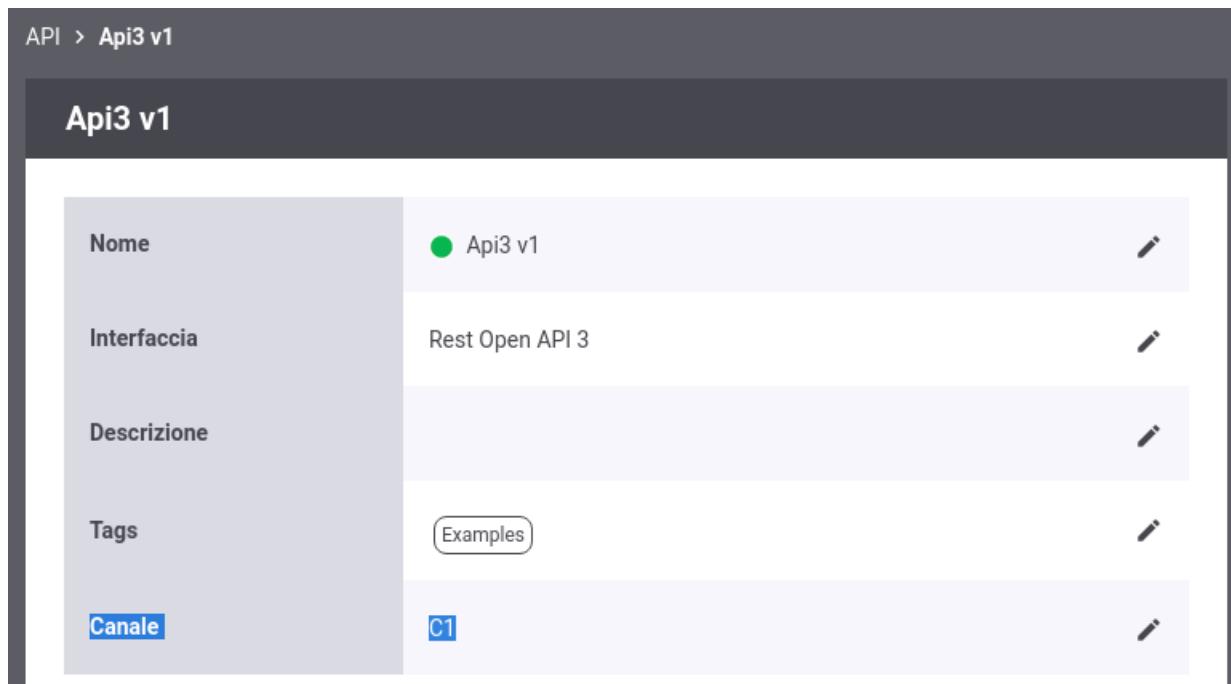


Figure8.18: Modifica del canale associato ad un'API

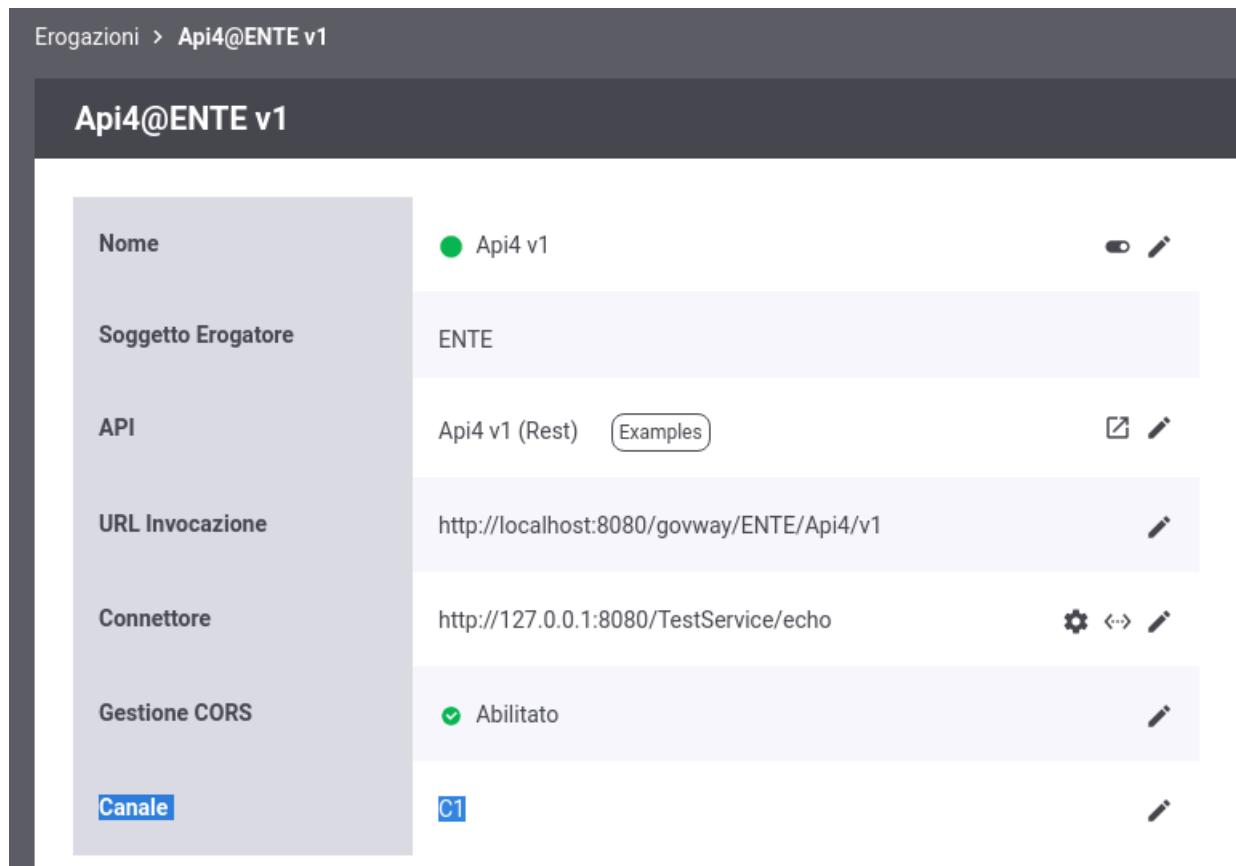


Figure8.19: Modifica del canale associato ad una erogazione

Url di Invocazione e Canali

Su ogni nodo saranno autorizzate ad essere invocate solamente le API che possiedono un canale corrispondente alla configurazione del nodo come raffigurato in Fig. 8.20.

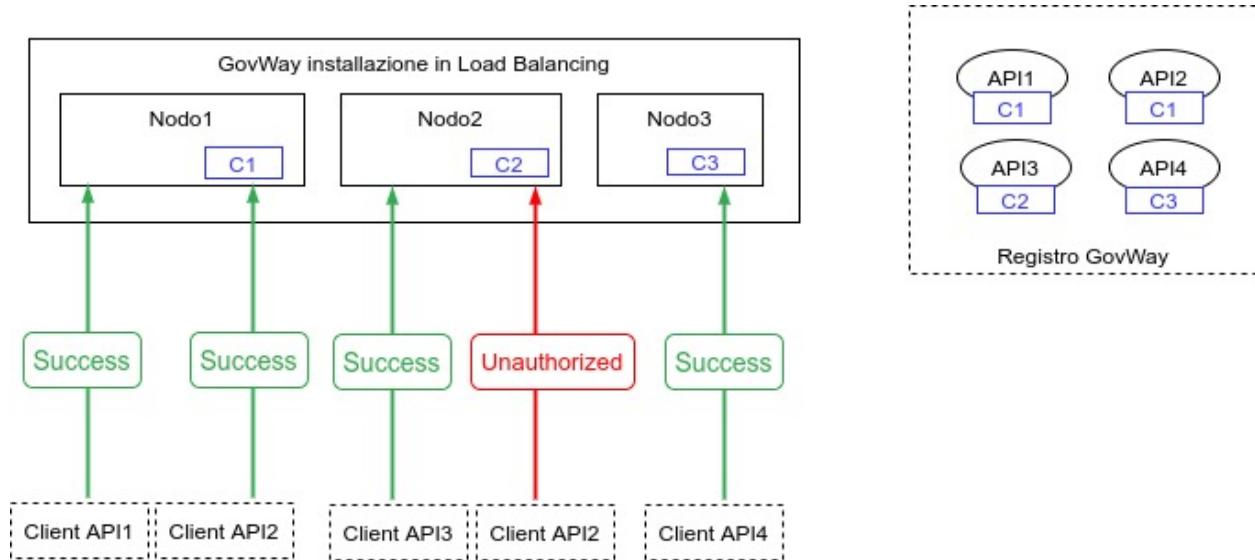


Figure8.20: Suddivisione delle API in Canali

È possibile utilizzare le regole di proxy pass descritte nella sezione *URL di Invocazione API* al fine di far visualizzare una url di invocazione nel dettaglio di una erogazione o fruizione che indirizzi il nodo corretto. La definizione corretta delle regole di proxy pass dipendono dall'architettura reale dei nodi in Load Balancing. Di seguito vengono forniti alcuni esempi al fine di esemplificare la funzionalità.

Ipotesi1: Nome del Canale corrisponde all'hostname di un nodo

In questo primo scenario ogni API sarà invocabile solamente su uno dei nodi che compongono il cluster (Fig. 8.21). Lo scenario prevede che gli identificativi dei canali corrispondano all'hostname di un nodo.

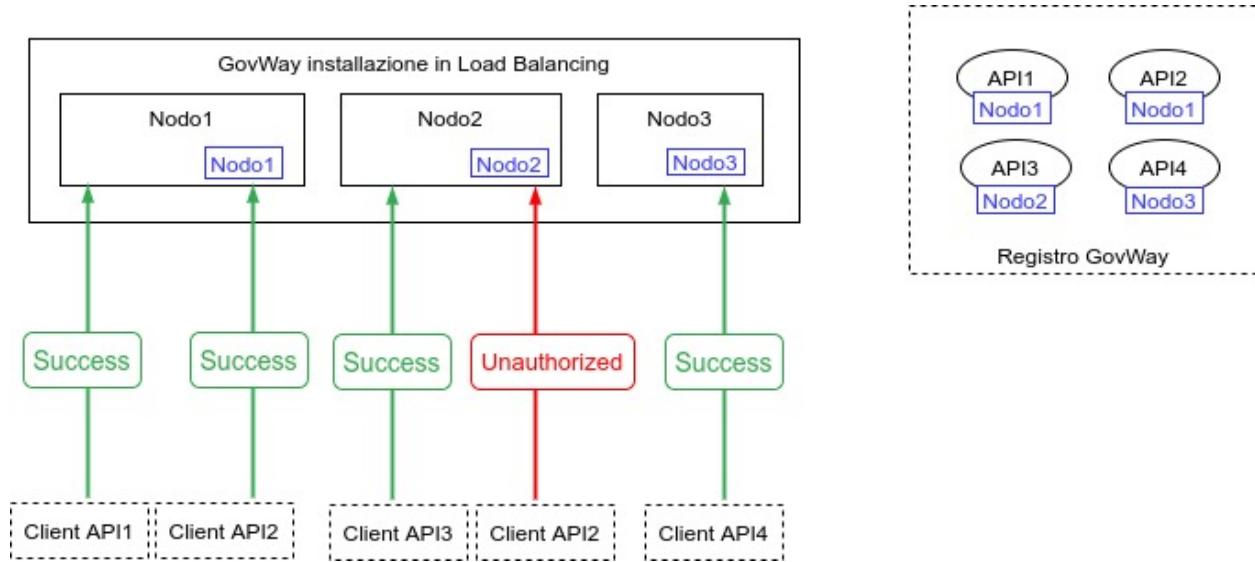


Figure8.21: Suddivisione delle API in Canali, scenario 1

Creando una regola di proxy pass (vedi [URL di Invocazione API](#)) con i seguenti criteri di Applicabilità:

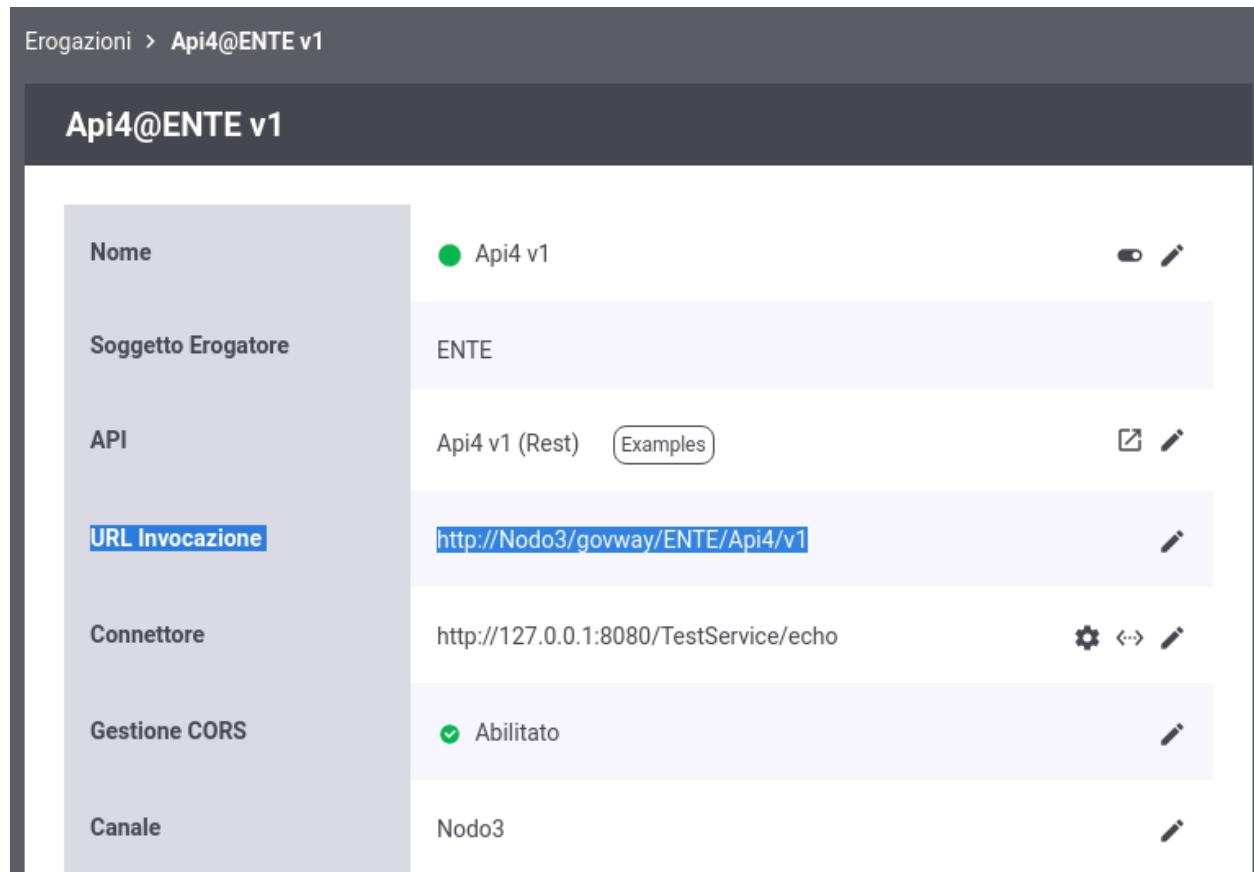
- Espressione Regolare: true
- Regola: (.+)
- Profilo: API Gateway
- Soggetto: Qualsiasi
- Ruolo: Erogazione
- Tipo API: Qualsiasi

Nuova URL di Invocazione

- Base URL: [http://\\${canale}/govway](http://${canale}/govway)
- Contesto: \${0}

L'url di invocazione visualizzata per ogni erogazione indirizzerà il corretto host corrispondente al canale (Fig. 8.22):

- <http://Nodo3/govway/ENTE/Api4/v1>



Nome	Api4 v1	Actions
Soggetto Erogatore	ENTE	
API	Api4 v1 (Rest) Examples	Edit
URL Invocazione	http://Nodo3/govway/ENTE/Api4/v1	Edit
Connettore	http://127.0.0.1:8080/TestService/echo	Edit
Gestione CORS	Abilitato	Edit
Canale	Nodo3	Edit

Figure8.22: Url di Invocazione, scenario 1

Ipotesi2: Nome del Canale corrisponde all'hostname di un load balancer

In questo scenario l'architettura è composta da 3 canali ognuno dei quali è gestito da due nodi GovWay bilanciati da un load balancer (Fig. 8.23). Lo scenario prevede che gli identificativi dei canali corrispondono all'hostname dei load balancer.

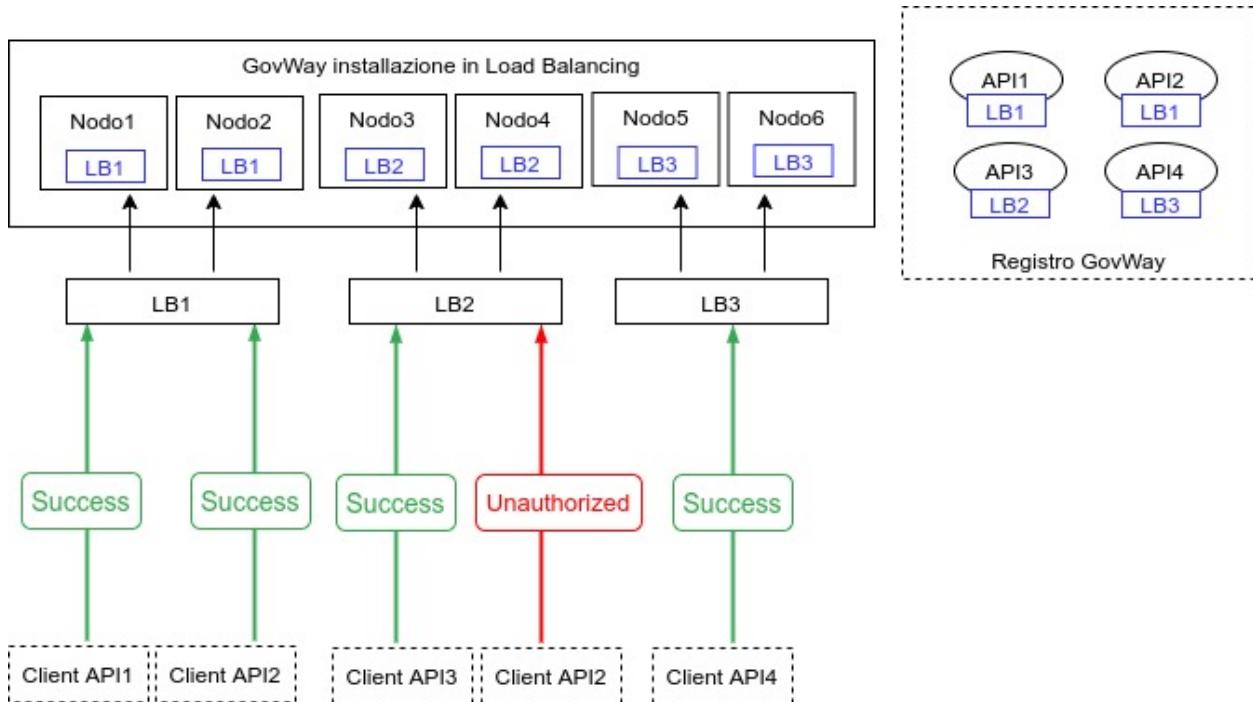


Figure 8.23: Suddivisione delle API in Canali, scenario 2

La configurazione da utilizzare nelle regole di proxy pass è identica a quelle descritte nella Ipotesi 1

Ipotesi3: Nome del Canale corrisponde ad un contesto gestito da un FrontendWeb

In questo scenario ogni nodo in Load Balancing viene acceduto tramite un Frontend Web. Le richieste vengono redirette al corretto nodo in base ad una informazione di contesto presente nella url. Lo scenario prevede che gli identificativi dei canali corrispondono all'informazione di contesto utilizzata dal Frontend Web per inoltrare le richieste al corretto nodo.

Creando una regola di proxy pass (vedi [URL di Invocazione API](#)) con i seguenti criteri di Applicabilità:

- Espressione Regolare: true
- Regola: (.+)
- Profilo: API Gateway
- Soggetto: Qualsiasi
- Ruolo: Erogazione
- Tipo API: Qualsiasi

Nuova URL di Invocazione

- Base URL: <http://frontend/govway>
- Contesto: \${canale}\${0}

L'url di invocazione visualizzata per ogni erogazione conterrà la corretta informazione di contesto che verrà utilizzata dal Frontend Web per smistare le richieste (Fig. 8.25):

- <http://frontend/govway/C2/ENTE/Api3/v1>

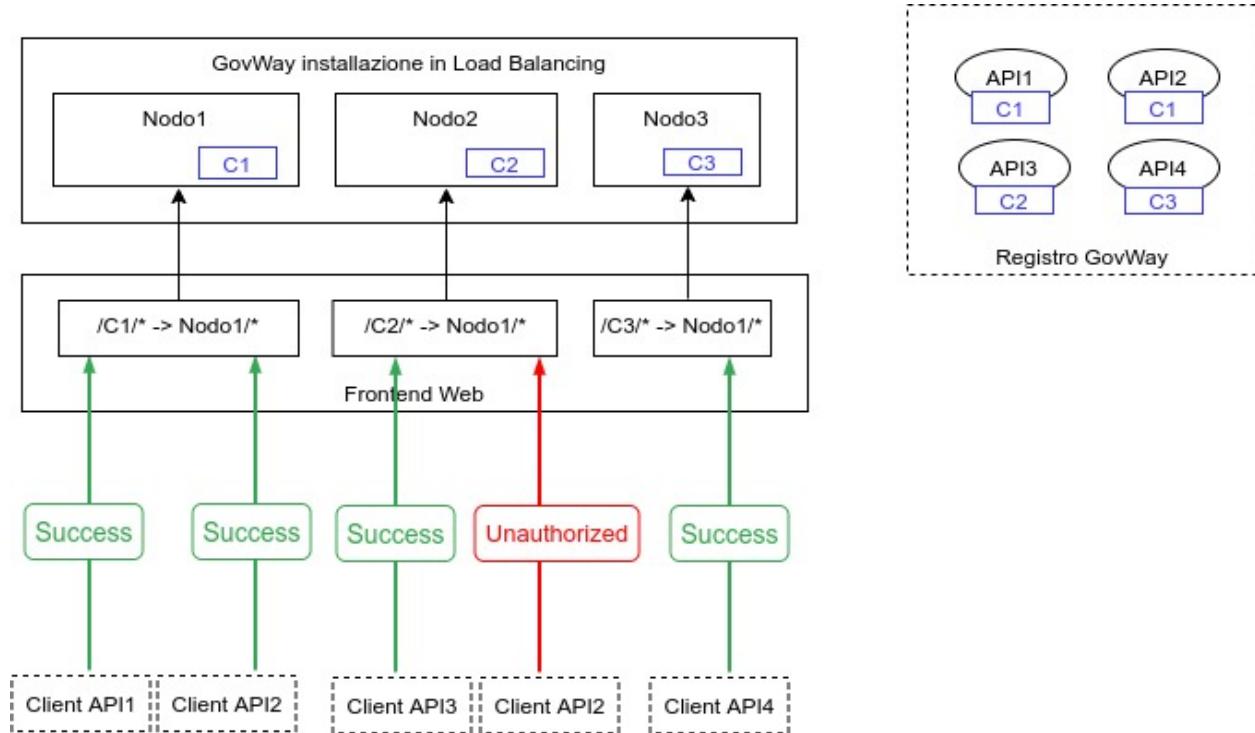


Figure 8.24: Suddivisione delle API in Canali, scenario 3

8.1.7 Proprietà

La funzionalità consente di registrare una serie di proprietà che saranno aggiunte tra le proprietà java nel sistema tramite l’invocazione del metodo:

```
java.lang.System.setProperty(nome, valore);
```

La funzionalità è utilizzabile sia per impostare proprietà utilizzate direttamente da java, come ad es. le proprietà che riguardano il networking, sia per configurare altre funzionalità di GovWay, come ad es. [Gestione Proxy](#).

8.2 Cache

Accedendo alla console di gestione in modalità avanzata ([Modalità Avanzata](#)) nella sezione “*Configurazione > Cache*” ([Fig. 8.27](#)) è possibile configurare i parametri di ogni cache utilizzata da GovWay. Maggiori dettagli vengono forniti nella sezione `govWayCaches` della guida di installazione.

Accedendo alla sezione “*Strumenti > Runtime*” ([Runtime](#)), tramite la voce *Svuota tutte le Cache* è possibile effettuare il reset contemporaneo di tutte le cache utilizzate dal gateway.

L’operazione precedente impatterà su tutte le configurazioni indistintamente. Se si volesse invece eliminare dalle cache le informazioni puntuali di una specifica configurazione si può utilizzare il menù presente sia nell’elenco che nel dettaglio di ogni entità principale di GovWay (API, Erogazione, Soggetto, Applicativo, Ruolo ...) come mostrato nelle figure [Fig. 8.28](#) e [Fig. 8.29](#).

Erogazioni > **Api3@ENTE v1**

Api3@ENTE v1

Nome	Api3 v1	
Soggetto Erogatore	ENTE	
API	Api3 v1 (Rest)	
URL Invocazione	http://frontend/govway/C2/ENTE/Api3/v1	
Connettore	http://127.0.0.1:8080/TestService/echo	
Gestione CORS	Abilitato	
Canale	C2	

Figure8.25: Url di Invocazione, scenario 3

Configurazione Generale > Proprietà di Sistema > **Risultati ricerca**

Risultati ricerca

	Nome	Valore
<input type="checkbox"/>	http.nonProxyHosts	localhost host.example.com
<input type="checkbox"/>	http.proxyHost	proxyHostName
<input type="checkbox"/>	http.proxyPort	8080

Figure8.26: Elenco di proprietà di una configurazione

GovWay - Console di Gestione

Registro

- API
- Erogazioni
- Fruizioni
- Soggetti
- Applicativi
- Ruoli
- Scope

Strumenti

- Runtime
- Auditing
- Coda Messaggi

Configurazione

- Generale
- Cache**
- Tracciamento
- Controllo del Traffico
- Token Policy
- Attribute Authority
- Tags
- Utenti
- Importa
- Esporta

Configurazione Cache

Note: (*) Campi obbligatori

Cache (Dati delle Richieste)

Stato	<input type="text" value="abilitato"/>
Dimensione (Elementi) *	<input type="text" value="15000"/>
Algoritmo	<input type="text" value="LRU"/>
Item Life Time (Secondi)	<input type="text" value="7200"/>
Item Idle Time (Secondi)	<input type="text"/>

Non indicare i secondi per avere un tempo infinito

Cache (Registro API)

Stato	<input type="text" value="abilitato"/>
Dimensione (Elementi) *	<input type="text" value="10000"/>
Algoritmo	<input type="text" value="LRU"/>
Item Life Time (Secondi)	<input type="text" value="7200"/>
Item Idle Time (Secondi)	<input type="text"/>

Non indicare i secondi per avere un tempo infinito

Cache (Configurazione della Porta)

Figure8.27: GovWay Cache



Figure8.28: GovWay Cache: eliminazione puntuale di una configurazione scelta dall'elenco

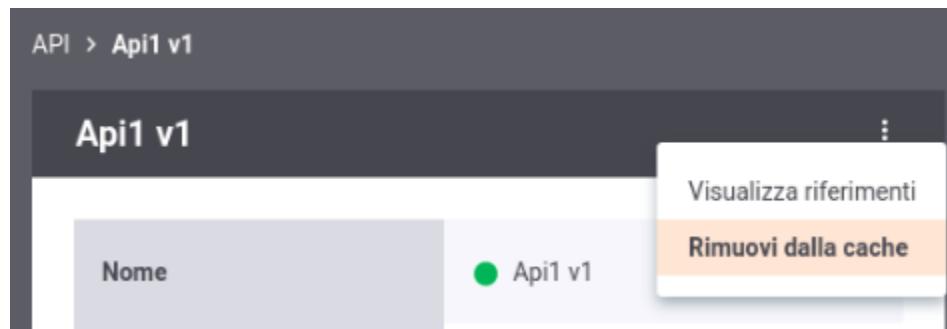


Figure8.29: GovWay Cache: eliminazione puntuale di una configurazione

8.3 Cache PDND

Accedendo alla console di gestione in modalità avanzata (*Modalità Avanzata*) nella sezione “Configurazione > Cache PDND” (Fig. 8.30) è possibile consultare i dati presenti nella cache locale contenente le chiavi pubbliche (JWK) e le informazioni sui client raccolte tramite le *API PDND*.

	Data di Registrazione	Chiave Pubblica	Client Id	Dettagli Organizzazione
<input type="checkbox"/>	2023-05-16 15:35:11.599	40Wnyd6Lkxx5iee12qpoRkuweSMgJONUWwg4ZsrQaw	780e8400-e29b-41d4-a716-446655440012	Comune di Esempio 2 (IPA_c_c002)
<input type="checkbox"/>	2023-05-16 15:35:01.075	55Wnyd6Lkyy5iee12qpoRkuweSMgJONUWwg4Zsr9bb	550e8400-e29b-41d4-a716-446655440000	Comune di Esempio 3 (IPA_c_c003)
<input type="checkbox"/>	2023-05-16 15:35:01.042	AAAC98fda52-9a37-41c0-8696-65e5022e9e44	010e8400-e29b-41d4-a716-446655440099	Comune di Esempio (IPA_c_c001)

Figure 8.30: GovWay Cache PDND

È possibile eliminare una o più chiavi dalla cache attraverso la selezione puntuale e l'utilizzo del pulsante *Elimina*.

Cliccando sul link che riporta il kid di una chiave sarà possibile effettuarne il download.

Infine cliccando sull'ora di registrazione si possono esaminare i dettagli sia riguardanti la chiave pubblica scaricata dalla PDND che le informazioni ottenute relative all'organizzazione afferente del client id (Fig. 8.31).

8.4 Controllo del Traffico

Accedendo la sezione *Configurazione > Controllo del Traffico* si possono impostare i parametri di configurazione relativamente alla funzionalità che consente di stabilire le politiche di accesso alle risorse del gateway, nell'ottica di amministrare le risorse applicative a disposizione, ottimizzando le prestazioni e gestendo le situazioni di picco.

La configurazione della funzionalità di controllo del traffico (Fig. 8.32) si compone dei seguenti gruppi di configurazioni:

- *Limitazione Numero di Richieste Complessive*: consente di fissare un numero limite, riguardo le richieste gestibili simultaneamente dal gateway, bloccando le richieste in eccesso.
- *Controllo della Congestione*: consente di attivare il rilevamento dello stato di congestimento del gateway, in seguito al superamento di una determinata soglia relativamente alle richieste simultanee.
- *Rate Limiting*: sezione per l'impostazione di policy al fine di attivare strategie di controllo del traffico con criteri di selezione specifici della singola richiesta.

Cache PDND > 780e8400-e29b-41d4-a716-446655440012

780e8400-e29b-41d4-a716-446655440012

Data Registrazione	2023-05-16 15:35:11.599
Chiave Pubblica	
Data Ultimo Aggiornamento	2023-05-25 10:13:44.412
Kid	40Wnyvd6Lkxx5iee12qpoRkuweSMgJONUWwg4ZsrOaw
Chiave Pubblica	<pre>{"alg":"RS256","e":"AQAB","kid":"40Wnyvd6Lkxx5iee12qpoRkuweSMgJONUWwg4ZsrOaw","kty":"RSA","n":"q5-orsiThMHNHciDE-6Zbu5bmbfi9QbTsXU4IPsmSAXjcLf7Bg-e0zmYOCg0J0HPG7SqtCV80k_MShiAisXLP3axdbvaxah4YqE-CBPTsDq4hxUirCXJtTd6G3ZVhYatL-XnnXzA2YeqBsItpxn_ZrhRK2-py9-"} download</pre>
Informazioni Client	
Client Id	780e8400-e29b-41d4-a716-446655440012
Dettagli Client	{"consumerId":"00345678-254d-bbbb-aaaa-82e210e12345","id":"780e8400-e29b-41d4-a716-446655440012"}
Dettagli Organizzazione	{"category":"Comuni e loro Consorzi e Associazioni","externalId":{"id":"c_c002","origin":"IPA"},"id":"00345678-254d-bbbb-aaaa-82e210e12345","name":"Comune di Esempio 2"}

Figure8.31: GovWay Cache: dettagli di una chiave

Controllo del Traffico

Note: (*) Campi obbligatori

Limitazione Numero di Richieste Complessive

Stato	<input type="text" value="abilitato"/>
Max Richieste Simultanee *	<input type="text" value="200"/>
Visualizza Informazioni Runtime	

Controllo della Congestione

Stato	<input type="text" value="disabilitato"/>
-------	---

Rate Limiting

Registrazione Policy (0)
Policy Globali (0)

Tempi Risposta

Fruizioni

Connection Timeout *	<input type="text" value="10000"/>
Indicazione in millisecondi (ms)	
Read Timeout *	<input type="text" value="150000"/>
Indicazione in millisecondi (ms)	
Tempo Medio di Risposta *	<input type="text" value="10000"/>
Indicazione in millisecondi (ms)	

Erogazioni

Connection Timeout *	<input type="text" value="10000"/>
Indicazione in millisecondi (ms)	
Read Timeout *	<input type="text" value="120000"/>
Indicazione in millisecondi (ms)	
Tempo Medio di Risposta *	<input type="text" value="10000"/>
Indicazione in millisecondi (ms)	

SALVA

Figure8.32: Maschera per l'impostazione dei parametri di controllo del traffico

- *Tempi Risposta*: sezione per l'impostazione dei valori limite relativi ai tempi di risposta dei servizi, sia nei casi di erogazione che di fruizione.

Le sezioni seguenti dettagliano questi elementi di configurazione.

8.4.1 Limitazione Numero di Richieste Complessive

Il primo livello di configurazione, presente nella pagina di accesso, consente di impostare i seguenti parametri:

- *Stato* (abilitato | disabilitato | warningOnly): Attiva il controllo sul numero di richieste simultanee in elaborazione. Selezionando l'opzione *abilitato* le richieste simultanee ricevute, che eccedono la soglia indicata (parametro *MaxRichiesteSimultanee*) verranno rifiutate restituendo al chiamante un errore. La tipologia di errore restituita è configurabile tramite l'elemento *Tipologia Errore* che appare solamente in caso di controllo abilitato.

Il controllo sul numero di richieste simultanee in elaborazione può anche essere attivato in modalità *WarningOnly* dove, in caso il superamento della soglia, genera solamente un messaggio diagnostico di livello *error* e un evento che segnala l'accaduto.

- *Max Richieste Simultanee*: Corrisponde al numero massimo di richieste simultanee accettate. In genere è possibile fornire un valore accurato dopo aver valutato la portata massima del prodotto installato, in base alle risorse hardware disponibili e ai parametri di dimensionamento delle risorse applicative (ad esempio: numero connessioni al database, dimensioni della memoria java, ecc).

Al superamento di tale valore non verranno accettate ulteriori richieste concorrenti, che verranno quindi rifiutate. Al verificarsi di questa situazione il gateway emette un evento specifico. Queste transazioni vengono marcate con esito *Superamento Limite Richieste* e saranno registrate solamente se previsto dalla configurazione (per default non vengono registrate). Per i dettagli sulla configurazione delle transazioni da registrare in base all'esito consultare la sezione *Tracciamento*.

- *Tipo Errore per API SOAP e Includi Descrizione Errore* (Opzioni presenti solo con console in modalità avanzata): Imposta il tipo di errore restituito al chiamante nel caso di rifiuto dell'elaborazione per superamento della soglia del numero massimo di richieste simultanee. Le opzioni possibili sono le seguenti:

- *Fault*: viene generato un messaggio di Fault contenente un codice ed una descrizione dell'errore rilevato nel caso l'elemento *Includi Descrizione Errore* sia abilitato, o un codice di errore generico altrimenti.
- *Http 429 (Too Many Requests)*
- *Http 503 (Service Unavailable)*
- *Http 500 (Internal Server Error)*

Viene generata una risposta HTTP con il codice selezionato, contenente una pagina html di errore, se l'elemento *Includi Descrizione Errore* è abilitato, o una risposta http vuota altrimenti.

- *Visualizza Informazioni Runtime*: Selezionando questo collegamento si apre una pagina (Fig. 8.33) che mostra in real-time le seguenti informazioni:
 - *Richieste Attive*: il numero di richieste simultanee attualmente in corso di elaborazione.
 - *Stato Gateway*: indica se il gateway ha raggiunto o meno lo stato di congestimento, e quindi superata la soglia sul numero massimo di richieste simultanee.

Nota: L'indicatore è attivo solo nel caso in cui lo stato della successiva opzione *Controllo della Congestione* sia abilitato.

- *Refresh*: collegamento che consente di aggiornare le informazioni presentate nello schermo.



Figure8.33: Dati di congestimento in tempo reale

8.4.2 Controllo della Congestione

Questa sezione consente di impostare i parametri relativi al controllo della congestione. Sono disponibili le seguenti opzioni:

- *Stato* (abilitato | disabilitato): Attiva il controllo sul numero di richieste simultanee al fine di individuare lo stato di congestimento.
- *Soglia di Attivazione (%)*: Selezionando l'opzione *abilitato*, al passo precedente, questo elemento consente di indicare la soglia dello stato di congestimento. La soglia da indicare è in percentuale rispetto al Numero Massimo Richieste Simultanee. Al superamento di tale soglia si entra nello stato di congestimento conseguente emissione di un evento e un messaggio diagnostico al riguardo.

Nota:

Sulla base della percentuale indicata come soglia, una dicitura riporta nella pagina il valore di congestimento calcolato in base al numero massimo di richieste simultanee.

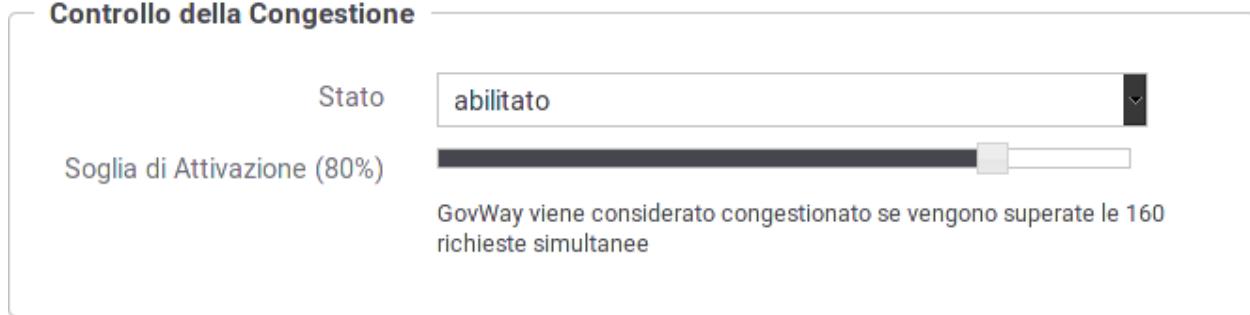


Figure 8.34: Configurazione della soglia di congestimento

8.4.3 Rate Limiting

Questa sezione consente di creare e attivare le policy di controllo del traffico. Gli elementi di configurazione presenti sono:

- *Tipo Errore per API SOAP e Includi Descrizione Errore* (Opzioni presenti solo con console in modalità avanzata): Imposta il tipo di errore restituito al chiamante nel caso venga rilevata una violazione delle policy configurate:
 - *Fault*: viene generato un messaggio di Fault contenente un codice ed una descrizione dell'errore rilevato nel caso l'elemento *Includi Descrizione Errore* sia abilitato, o un codice di errore generico altrimenti.
 - *Http 429 (Too Many Requests)*
 - Http 503 (Service Unavailable)*
 - Http 500 (Internal Server Error)*
 viene generata una risposta HTTP con il codice selezionato contenente una pagina html di errore se l'elemento *Includi Descrizione Errore* è abilitato, od una risposta http vuota altrimenti.
- *Sincronizzazione*: consente di configurare la modalità di conteggio delle policy di rate limiting in modo da supportare scenari di cluster di più nodi. I parametri di configurazione sono identici a quanto descritto nella sezione *Rate Limiting in presenza di un cluster di nodi*.
- *HTTP Headers*: permette di personalizzare gli header HTTP ritornati al client che contengono informazioni sulle policy di rate limiting attive. I parametri di configurazione sono identici a quanto descritto nella sezione *Personalizzazione degli Header HTTP restituiti al client*.
- *Registro Policy*: consente di accedere al Registro delle Policy per visualizzare, modificare e creare le policy di controllo istanziabili per la configurazione del rate limiting. Tra parentesi viene visualizzato il numero di policy attualmente presenti nel registro. Questa funzionalità è descritta nella sezione *Registro Policy*.
- *Policy Globali*: consente di accedere al Registro delle Policy Attivate in ambito globale, cioè operative sul traffico complessivo che transita sul gateway. A queste policy si aggiungono quelle eventualmente definite localmente nella configurazione specifica di ciascuna erogazione/erogazione. Tra parentesi viene visualizzato il numero di policy attualmente attivate. Questa funzionalità è descritta nella sezione *Policy Globali*.

8.4.4 Tempi Risposta

In questa sezione vengono indicati i valori limite di default riguardo i tempi di risposta dei servizi con cui il gateway interagisce durante l'elaborazione delle richieste. Nel caso delle erogazioni, si tratta dei tempi di risposta dei servizi interni, successivamente ad una richiesta di erogazione dall'esterno. Nel caso delle fruizioni, si tratta dei tempi di risposta dei servizi esterni, successivamente ad una richiesta di fruizione da parte di un client interno al dominio. I tempi configurabili sono:

- *Connection Timeout (ms)*: Intervallo di tempo atteso, sulle comunicazioni in uscita, prima di sollevare l'errore Connection Timeout (scadenza del tempo di attesa per stabilire una connessione).
- *Read Timeout (ms)*: Intervallo di tempo atteso, dopo aver stabilito una connessione in uscita, prima di sollevare l'errore di Read Timeout (scadenza del tempo di attesa per ricevere il payload dall'interlocutore).

Nota: È possibile impostare un timeout, in millisecondi, per la ricezione del payload della richiesta configurando la *Proprietà* “*connettori.request.timeoutMs*” sulla singola erogazione o fruizione.

- *Tempo Medio di Risposta (ms)*: Valore di soglia del tempo medio di risposta al fine di valutare la situazione di *Degrado Prestazionale*, condizione per l'applicabilità di eventuali politiche restrittive come documentate più avanti.

8.5 Tracciamento

Accedendo la sezione *Configurazione > Tracciamento* si può configurare quali informazioni, relative alle comunicazioni in transito sul gateway dei servizi erogati o fruitti, verranno registrate (Fig. 8.35):

- *Transazioni*: questa sezione consente di configurare il tipo di tracciamento attuato dal gateway differenziandone il comportamento tra erogazioni e fruizioni rispettivamente tramite i link *Configurazione Erogazioni* e *Configurazione Fruizioni*. Maggiori dettagli vengono forniti nella sezione *Registrazione della Transazione*.
- *Messaggi Diagnostici*: questa sezione consente di specificare il livello di verbosità della diagnostica emessa dal gateway utile a comprendere la fase di elaborazione delle richieste e indagare sulle anomalie occorse. Si può distinguere il livello di verbosità per il salvataggio su *Database* e su *File*.

Nota: Le configurazioni effettuate in questa sezione della console hanno valenza globale e quindi rappresentano il comportamento di default adottato dal gateway nella gestione dei diversi flussi di comunicazione. Tale comportamento può essere ridefinito puntualmente su ogni singola erogazione/fruizione agendo sulla voce di configurazione *Tracciamento*.

8.5.1 Registrazione della Transazione

Il tracciamento è la funzionalità del gateway che comporta la registrazione dei dati relativi alle comunicazioni in transito riguardanti i servizi erogati e fruitti. Nella logica del gateway, tutte le informazioni che riguardano una singola interlocuzione, a partire dalla richiesta pervenuta fino alla conclusione con l'invio dell'eventuale risposta, sono riconducibili ad un'unica entità denominata *Transazione*.

Una transazione registrata dal gateway ha la seguente struttura:

- *Dati di Identificazione Generale*. Sono le informazioni che identificano la comunicazione specifica in termini dei soggetti coinvolti e del servizio richiesto: Soggetto Erogatore, Servizio, Azione, Esito, tempi di risposta ...

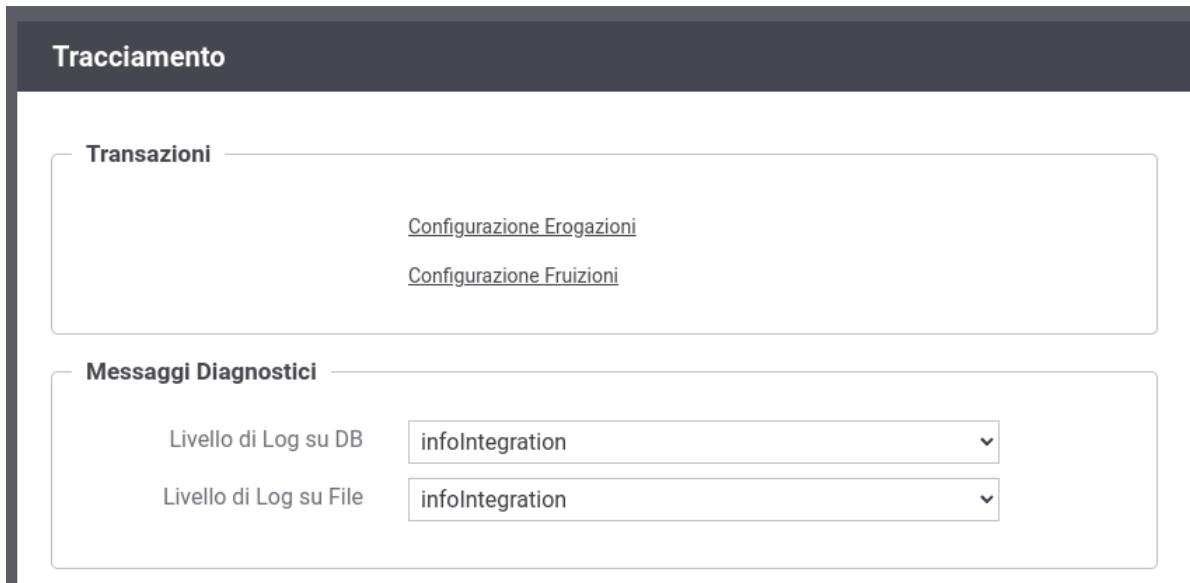


Figure8.35: Configurazione del servizio di tracciamento

- *Dati Mittente.* Sono le informazioni che identificano il richiedente: Soggetto Fruitore, Applicativo, Token, IndirizzoIP, ...
- *Dati della Richiesta.* Sono le informazioni di dettaglio relative alla richiesta: Identificativo del Messaggio, Timestamp di ingresso, Timestamp di uscita, dimensioni del messaggio, ...
- *Dati della Risposta.* Sono le medesime informazioni già citate al punto precedente, ma relative alla comunicazione di risposta.
- *Tracce.* Eventuale traccia di richiesta o di risposta conforme ai profili di interoperabilità che lo richiedono.
- *Messaggi Diagnostici.* La sequenza dei messaggi diagnostici, ordinati cronologicamente, emessi dal gateway nel corso dell'elaborazione dell'intera transazione.
- *Messaggi.* Se abilitata la funzionalità di Registrazione Messaggi saranno presenti i contenuti del payload e degli header HTTP della richiesta e della risposta transiti su GovWay.
- *Fault.* Viene registrato come *Fault di Ingresso* l'eventuale messaggio di errore ricevuto dal gateway durante l'invocazione di un servizio (interno o esterno al dominio gestito) e come *Fault di Uscita* l'eventuale messaggio di errore inoltrato dal gateway al mittente della richiesta (interno o esterno al dominio gestito).

Tutte le informazioni di una transazione sono salvabili da GovWay tramite due modalità:

- database: i dati vengono salvati nella base dati di tracciamento come descritto nella sezione *Tracciamento su Database* e saranno consultabili tramite la Console di Monitoraggio descritta nella sezione mon_intro ;
- file: i dati vengono salvati su file di log come descritto nella sezione *Tracciamento su File*.

La configurazione Fig. 8.36 consente di personalizzare il tracciamento attuato dal gateway per entrambe le modalità tramite le seguenti opzioni:

- *Stato:* permette di abilitare o disabilitare il tracciamento oltre a consentirne una personalizzazione nelle sue varie fasi come descritto nella sezione *Fasi di Tracciamento*;
- *Filtra per Esiti:* consente di specificare quali transazioni memorizzare nell'archivio di monitoraggio in base all'esito rilevato in fase di elaborazione come descritto nella sezione *Tracciamento filtrato per Esiti*.

Nota: La funzionalità di tracciamento su file (FileTrace) possiede un ulteriore stato “configurazioneEsterna”

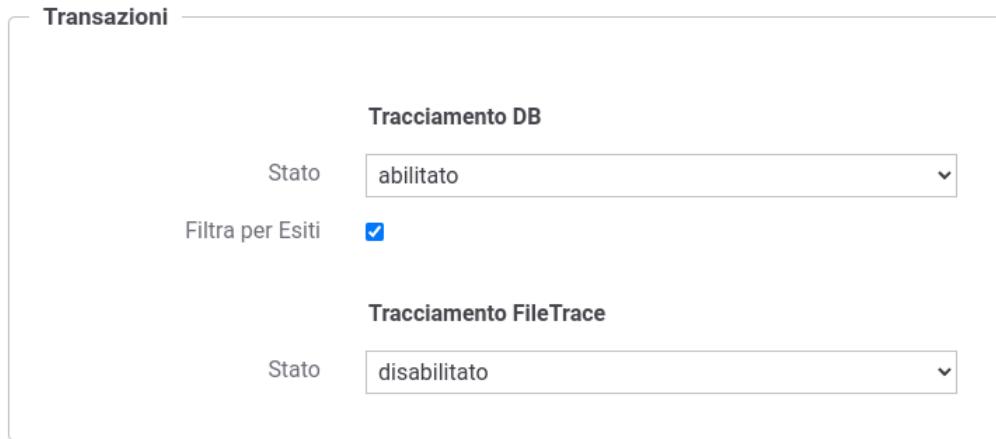


Figure8.36: Configurazione del tipo di tracciamento

che serve ad indicare che lo stato del tracciamento deve essere compreso esaminando la proprietà «org.openscoop2.pdd.transazioni.fileTrace.enabled» del file di configurazione locale «/etc/govway/govway_local.properties».

La figura Fig. 8.37 fornisce un esempio di personalizzazione delle fasi di tracciamento descritta nella sezione *Fasi di Tracciamento*.

Infine utilizzando la console in modalità *avanzata* (sezione *Modalità Avanzata*) è possibile configurare il tracciamento anche rispetto alle seguenti opzioni (Fig. 8.38):

- *Tempi Elaborazione*: vengono salvati tutti i tempi di inizio e fine di una fase di processamento della richiesta (es. autorizzazione, validazione, message-security ...);
- *Token*: vengono salvate le informazioni relative ai token negoziati in una fruizione o ricevuti in una erogazione (senza la parte di “signature” dei token in modo che non siano riutilizzabili come token di autenticazione).

8.5.2 Tracciamento filtrato per Esiti

Ad ogni transazione viene associato un esito di elaborazione come descritto nella sezione `mon_esito_transazione`.

Tramite la configurazione Fig. 8.39 è possibile indicare quali transazioni tracciare rispetto all'esito rilevato in fase di elaborazione.

Gli esiti sono suddivisi nei seguenti gruppi:

- Completate con successo: richieste che sono state gestite correttamente;
- Fault applicativo: richieste gestite correttamente in cui il backend ha restituito un fault applicativo, un SOAP Fault su API SOAP o un Problem Detail su API REST;
- Scartate: richieste che sono state scartate dal gateway immediatamente per differenti motivi:
 - richiedenti una API o un'operazione inesistente ;
 - contenevano una richiesta malformata nel contenuto;
 - credenziali richieste non presenti o non valide (es. cert mTLS o token);
 - API sospese.
- Violazione Policy Rate Limiting: richieste che violano una policy di rate limiting;

Transazioni

Tracciamento DB	
Stato	personalizzato
Richiesta ricevuta	abilitato (bloccante)
Richiesta in consegna	abilitato (non bloccante)
Risposta in consegna	abilitato (bloccante)
Risposta consegnata	abilitato

Tracciamento FileTrace	
Stato	personalizzato
Richiesta ricevuta	abilitato (bloccante)
Richiesta in consegna	disabilitato
Risposta in consegna	disabilitato
Risposta consegnata	abilitato

Figure8.37: Configurazione personalizzata del tipo di tracciamento

Informazioni Registrate

Tempi Elaborazione	disabilitato
Token	abilitato

Figure8.38: Configurazione avanzata delle informazioni salvate in una transazione

- Superamento Limite Richieste Complessive: il numero di richieste complessive supera il limite globale;
- Fallite: richieste che sono state processate con errore e non rientrano nei casi di richieste scartate;
- CORS Preflight: richieste di tipo “CORS Preflight”.

Per ciascun esito è possibile abilitare o disabilitare la registrazione.

Transazioni Registrate

Selezionare gli esiti che verranno registrati nello storico

Registra qualsiasi esito

Completate con successo

Stato

Fault Applicativo

Stato

Fallite

Stato

Scartate

Stato

Violazione Policy Rate Limiting

Stato

Superamento Limite Richieste Complessive

Stato

CORS Preflight

Stato

Figure8.39: Tracciamento filtrato per Esiti

È possibile inoltre, scegliendo l'opzione *personalizzato*, specificare puntualmente quali esiti di dettaglio includere (Fig. 8.40).

È infine possibile specificare l'indicazione di registrare qualsiasi esito (Fig. 8.41).

Scartate

Stato ▼

Autenticazione Fallita
 Token non Presente
 Autenticazione Token Fallita
 Gestione Token Fallita
 API non Individuata
 Operazione non Individuata
 Contenuto Richiesta Malformato
 Richiesta Malformata
 API Sospesa

Figure8.40: Tracciamento filtrato per Esiti personalizzato

Transazioni Registerate

Selezionare gli esiti che verranno registrati nello storico

Registra qualsiasi esito

Figure8.41: Tracciamento senza filtri per Esiti

8.5.3 Fasi di Tracciamento

Ogni richiesta ricevuta dal gateway viene gestita tramite un processo riassumibile nella figura Fig. 8.42 in cui il tracciamento, nella configurazione di default, avviene in fondo al processo dopo aver consegnato la risposta al client.

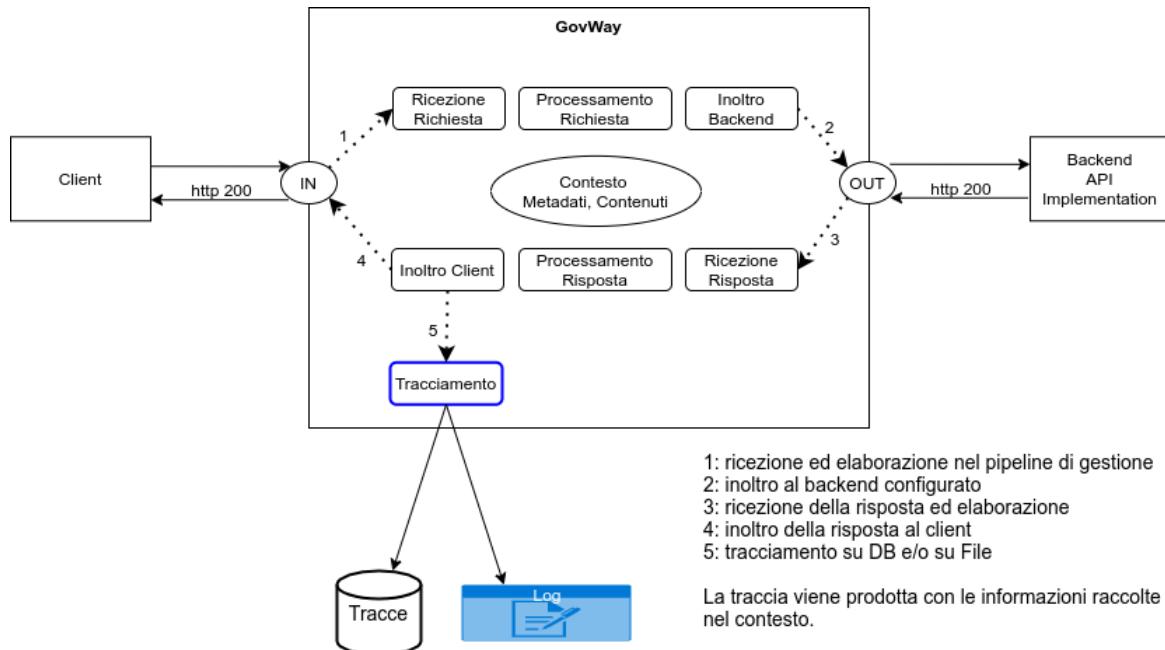


Figure 8.42: Fasi di tracciamento: configurazione di default

GovWay è configurabile per attuare il tracciamento anche su altri punti all'interno del processo di gestione di una richiesta personalizzando le fasi di tracciamento come indicato nella figura Fig. 8.43 e nella sezione [Registrazione della Transazione](#).

Le fasi in cui è possibile attivare il tracciamento sono le seguenti (raffigurate nella figura Fig. 8.44):

- *Richiesta ricevuta*: una volta individuata l'erogazione o la fruizione di API richiesta verrà effettuato il tracciamento prima di iniziare il normale processamento;
- *Richiesta in consegna*: terminato il processamento della richiesta e prima di inoltrarla al backend;
- *Risposta in consegna*: terminato il processamento della risposta e prima di inoltrarla al client;
- *Risposta consegnata*: dopo aver inoltrato la risposta al client.

Tracciamento Fallito

Nelle prime tre fasi (richiesta ricevuta, richiesta in consegna e risposta in consegna) è possibile abilitare il tracciamento con due modalità differenti:

- “abilitato (bloccante)”: la transazione terminerà con un errore ritornato al client se non è possibile effettuare il tracciamento richiesto (es. connessione al database non disponibile);
- “abilitato (non bloccante)”: l'anomalia avvenuta durante il tracciamento verrà registrata nei log di GovWay e il processamento della richiesta potrà continuare senza errori.

Nell'ultima fase non è invece possibile attuare la distinzione visto che una risposta è già stata consegnata al client. Un'eventuale errore di tracciamento verrà registrato nei log di GovWay e in caso di tracciamento su database la traccia verrà riversata nella base dati in un successivo momento, come descritto nella sezione [Tracciamento su Database](#), grazie al *processo di failover*.

Transazioni

Tracciamento DB	
Stato	personalizzato
Richiesta ricevuta	abilitato (bloccante)
Richiesta in consegna	abilitato (non bloccante)
Risposta in consegna	abilitato (bloccante)
Risposta consegnata	abilitato

Tracciamento FileTrace	
Stato	personalizzato
Richiesta ricevuta	abilitato (bloccante)
Richiesta in consegna	disabilitato
Risposta in consegna	disabilitato
Risposta consegnata	abilitato

Figure8.43: Configurazione personalizzata del tipo di tracciamento

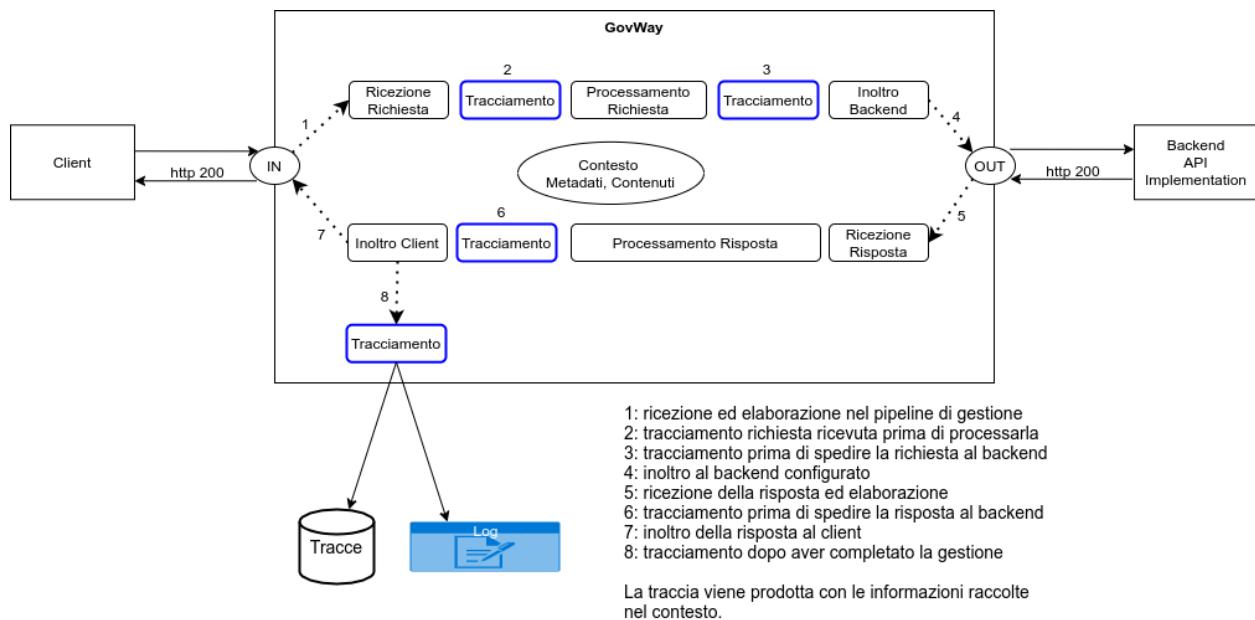


Figure8.44: Fasi di tracciamento: configurazione personalizzata

Nella figura Fig. 8.45 viene raffigurato uno scenario in cui avvengono problematiche (es. connessione al database non disponibile) nella seconda fase di tracciamento al database e tale fase è stata configurata per terminare con errore.

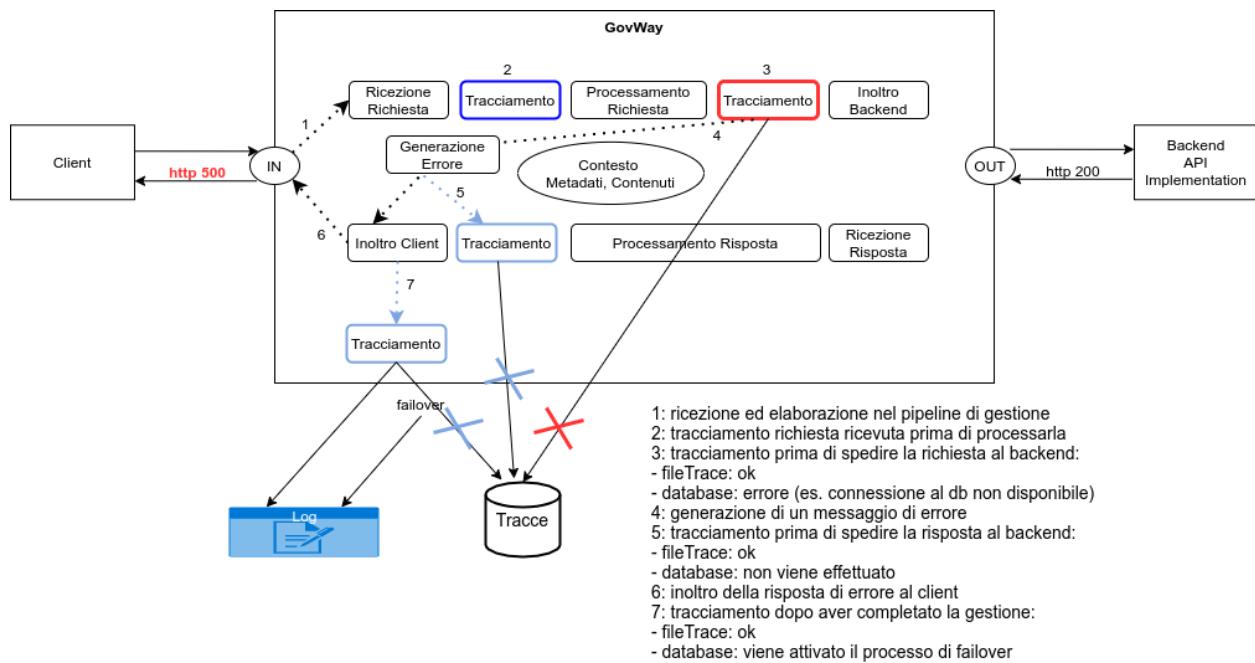


Figure8.45: Fasi di tracciamento: configurazione personalizzata con errore bloccante durante il tracciamento

Invece nella figura Fig. 8.46 viene raffigurato uno scenario in cui avvengono problematiche (es. connessione al database non disponibile) ma le fasi di tracciamento al database sono state configurate come *non bloccanti* e il tracciamento tramite FileTrace può proseguire correttamente.

In entrambi gli scenari raffigurati in Fig. 8.45 e Fig. 8.46 l'ultima fase di tracciamento su database avvierà un *processo di failover*, descritto nella sezione *Tracciamento su Database*, che consentirà di recuperare la traccia e riversarla su database in un secondo momento.

Tracciamento filtrato per Esiti

Come descritto nella sezione *Tracciamento filtrato per Esiti* è possibile indicare quali transazioni tracciare rispetto all'esito rilevato in fase di elaborazione. Tale funzionalità non risulta più utilizzabile se vengono abilitate le seguenti fasi di tracciamento:

- *Richiesta ricevuta*: l'opzione “*Filtra per Esiti*” non sarà attivabile poichè una transazione è già stata emessa prima di poter comprenderne l'esito.
- *Richiesta in consegna e Risposta in consegna*: l'opzione “*Filtra per Esiti*” sarà attivabile però non consentirà di filtrare esiti rispettivamente per errori di consegna o di processamento della risposta.

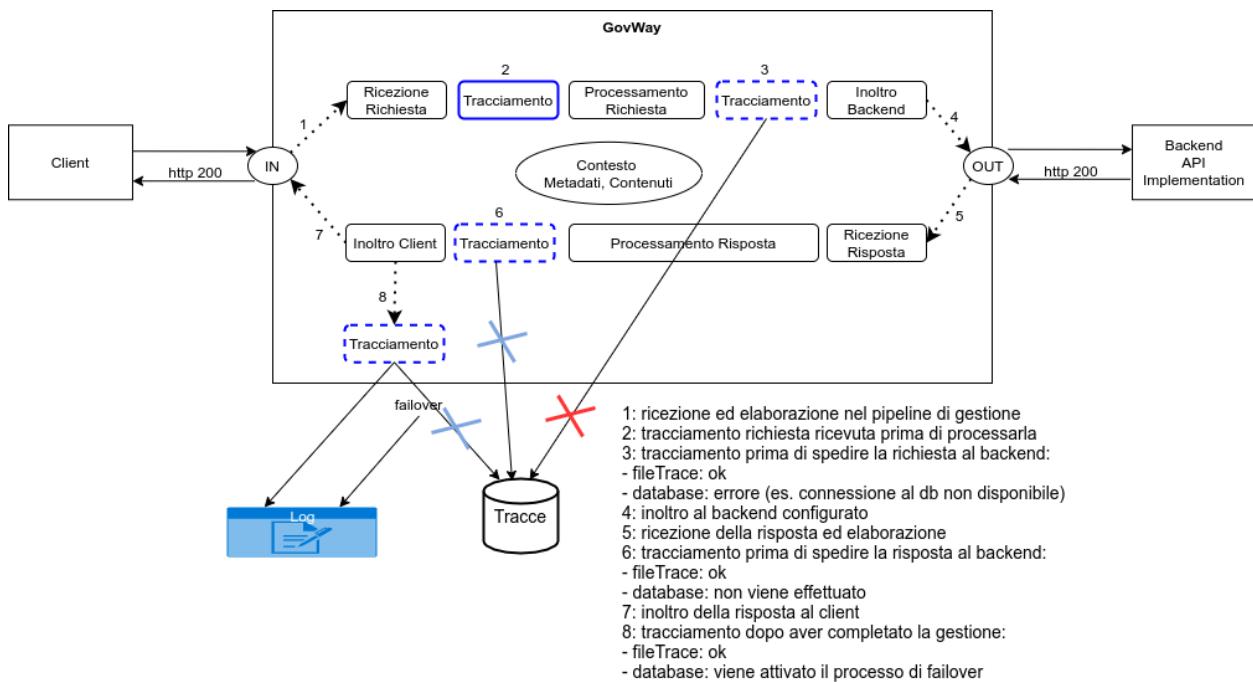


Figure8.46: Fasi di tracciamento: configurazione personalizzata con errore bloccante durante il tracciamento

8.5.4 Tracciamento su Database

Ogni richiesta ricevuta dal gateway viene gestita tramite un processo riassumibile nella figura Fig. 8.47 in cui il tracciamento, nella configurazione di default, avviene in fondo al processo dopo aver consegnato la risposta al client. Nel caso non fosse possibile attuare il tracciamento richiesto (es. connessione al database non disponibile) l'anomalia viene registrata nei log di GovWay e si attiva il *processo di failover* in cui la traccia verrà serializzata su filesystem per poi essere riversata nella base dati in un successivo momento da un timer dedicato al recupero.

Nel caso siano state attivate ulteriori fasi di tracciamento descritte nella sezione *Fasi di Tracciamento*, il *processo di failover* si attiva solamente in caso di problematiche di tracciamento durante l'ultima fase dopo che la risposta al client è già stata consegnata (figura Fig. 8.48).

La figura Fig. 8.49 mostra uno scenario in cui il tracciamento su database non va a buon fine già nelle prime fasi ed essendo configurate come “*bloccanti*” la transazione terminare con errore e viene restituito un fault al client. Nell’ultima fase viene attivato il *processo di failover*.

La figura Fig. 8.50 mostra uno scenario simile al precedente dove però le fasi sono configurate come “*non bloccanti*” e quindi il mancato tracciamento non inficia sulla corretta gestione della richiesta. Nell’ultima fase viene attivato il *processo di failover*.

La configurazione del *processo di failover* è personalizzabile a livello di configurazione locale in “/etc/govway/govway_local.properties” (assumendo sia /etc/govway la directory di configurazione indicata in fase di installazione).

Di seguito un esempio di configurazione effettuabile in cui vengono riportate tutte le opzioni presenti con i valori di default del prodotto.

```
# =====
# Directory dove vengono serializzate le transazioni non registrate
org.openspcoop2.pdd.resources.FileSystemRecovery.repository=/var/govway/
  ↵resources
```

(continues on next page)

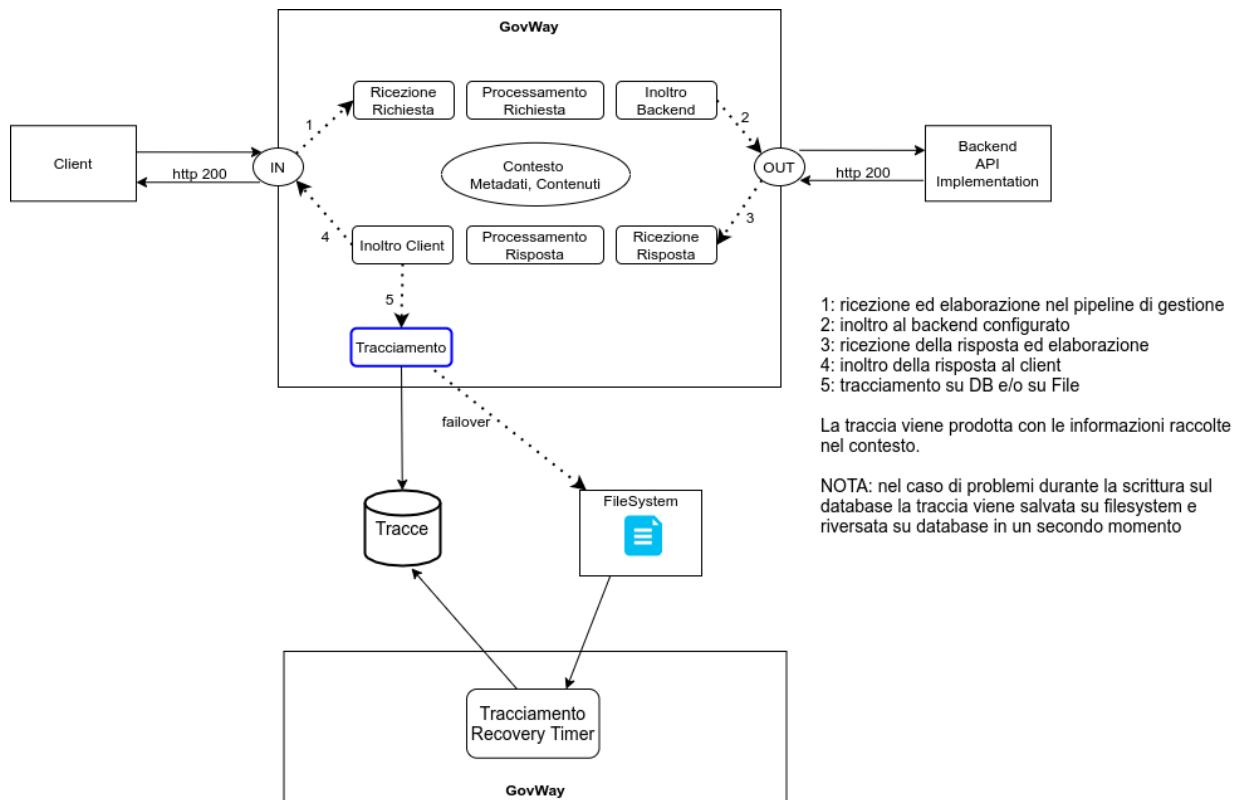


Figure8.47: Fasi di tracciamento su database: configurazione di default

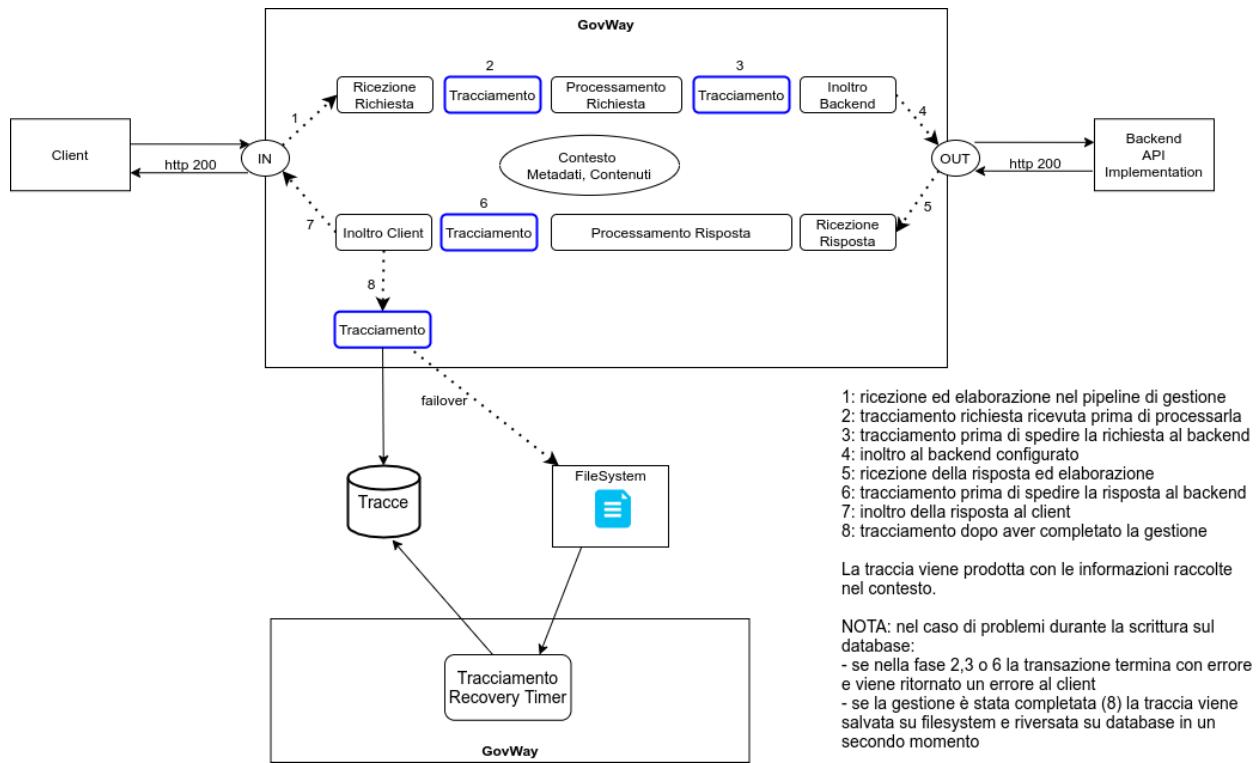


Figure8.48: Fasi di tracciamento su database: configurazione personalizzata

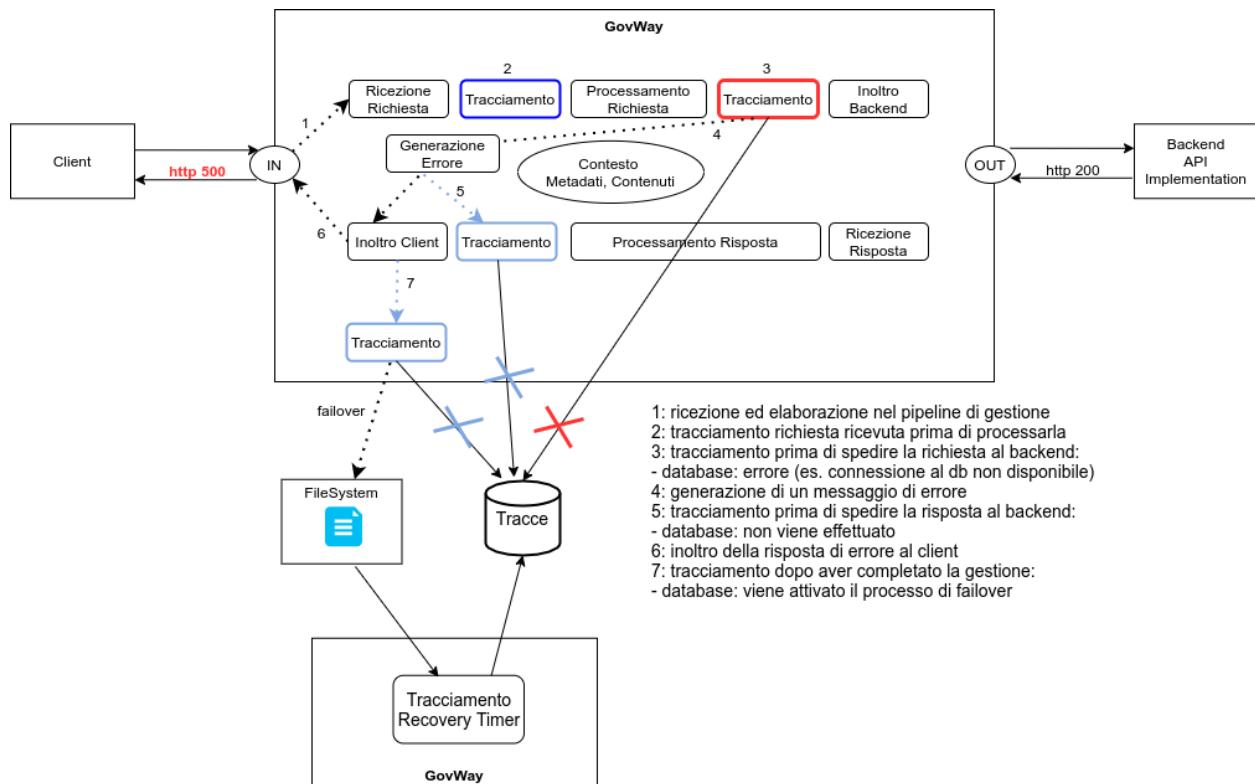


Figure8.49: Fasi di tracciamento su database: configurazione personalizzata con errore durante il tracciamento con fasi bloccanti

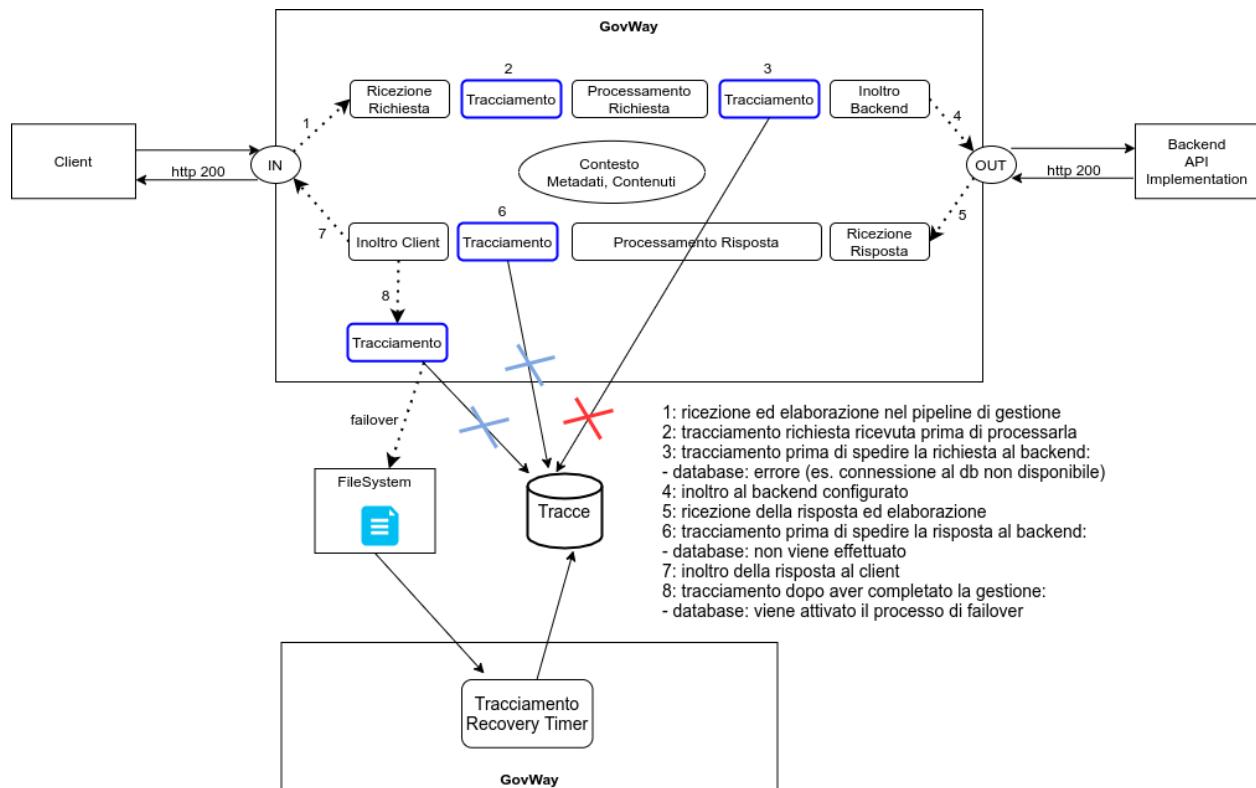


Figure8.50: Fasi di tracciamento su database: configurazione personalizzata con errore durante il tracciamento con fasi non bloccanti

(continua dalla pagina precedente)

```

# Indica se è abilitato il livello di debug durante la gestione
org.openspcoop2.pdd.resources.FileSystemRecovery.debug=false

# Indicazione se il timer dedicato al riversamento delle tracce serializzate
# su db è attivo o meno
org.openspcoop2.pdd.resources.FileSystemRecovery.enabled=true

# Parametri del timer
# - il timeout indica una unità di misura in secondi
# - dopo il numero di tentativi indicati in maxAttempts la traccia verrà
# spostata in una directory 'dlq'
org.openspcoop2.pdd.resources.FileSystemRecovery.timeout=300
org.openspcoop2.pdd.resources.FileSystemRecovery.maxAttempts=10
# =====

```

8.5.5 Tracciamento su File

Le informazioni inerenti le comunicazioni gestite dal gateway vengono registrate su una base dati di tracciamento (*Tracciamento su Database*) e sono consultabili tramite una console di monitoraggio (mon_intro).

È possibile estendere o sostituire il normale tracciamento su database, attivando il tracciamento su file.

La nuova funzionalità consente il tracciamento su file di tutte le informazioni relative alle comunicazioni gestite da GovWay e un successivo processamento del file da strumenti esterni (es. FileBeat) permette così una facile integrazione con sistemi di tracciamento esterni (es. Logstash, Kafka, ...).

La funzionalità consente una completa personalizzazione delle informazioni da riportare su file di log, permettendo anche di definirne il formato e l'ordine in cui vengono salvate. È inoltre possibile suddividere le informazioni in più file di log in modo da facilitare l'invio di informazioni selezionate a destinazioni diverse.

Le informazioni possono essere riversate in uno o più topic, dove ad ogni topic corrisponde tipicamente un file di log. Di seguito un esempio di file prodotto:

```

"req"|"b6cdd758-342c-4599-ae95-33a781730b3f"|"govway"|"2020-06-26 12:46:50:629
"|"+"0200"|"192.168.1.2"|"HTTP/1.1"|"POST"|"https://server:8446/example"|
"application/soap+xml; charset=UTF-8; action=\"test/\"||"10490"|"200"
"req"|"2a9dc253-9dd5-458b-8689-edee7c9ba139"|"govway"|"2020-06-26 12:47:50:561
"|"+"0200"|"192.168.1.2"|"HTTP/1.1"|"POST"|"https://server:8446/app2"|
"application/soap+xml; charset=UTF-8; action=\"test/\"||"1090"|"503"
"req"|"eeedb92b-66b5-451e-8266-ade2cf1f34ce"|"govway"|"2020-06-26 12:47:53:291
"|"+"0200"|"192.168.1.19"|"HTTP/1.1"|"POST"|"https://server:8446/example"|
"application/soap+xml; charset=UTF-8; action=\"test/\"||"11230"|"200"
"req"|"b4355a45-71cc-4293-b3b7-a4622af8ea84"|"govway"|"2020-06-26 12:48:00:102
"|"+"0200"|"192.168.1.22"|"HTTP/1.1"|"POST"|"https://server:8446/example"|
"application/soap+xml; charset=UTF-8; action=\"test/\"||"17999"|"200"

```

Maggiori dettagli sul formato dei log vengono forniti nella sezione (*Configurazione Tracciamento su File*).

Nella sezione *Configurazione dei Topic* viene descritto il formato del file di configurazione, mentre nella sezione *Informazioni Tracciabili* sono riportate tutte le informazioni disponibili.

Per abilitare il tracciamento su file è possibile agire nella configurazione dedicata descritta nella sezione *Registrazione della Transazione* e riportata nella figura Fig. 8.51.

Tracciamento FileTrace

Stato	<input type="text" value="abilitato"/>
Filtra per Esiti	<input checked="" type="checkbox"/>

Figure8.51: Configurazione del tracciamento su file di log

La configurazione consente inoltre di abilitare la produzione dei log su file:

- rispetto agli esiti di una transazione come descritto nella sezione *Tracciamento filtrato per Esiti*;
- rispetto alle varie fasi di tracciamento descritte nella sezione *Fasi di Tracciamento* (Fig. 8.52).

Tracciamento FileTrace

Stato	<input type="text" value="personalizzato"/>
Richiesta ricevuta	<input type="text" value="abilitato"/>
Richiesta in consegna	<input type="text" value="abilitato"/>
Risposta in consegna	<input type="text" value="abilitato"/>
Risposta consegnata	<input type="text" value="abilitato"/>

Figure8.52: Configurazione delle fasi di tracciamento su file di log

È inoltre possibile configurare un ulteriore stato “configurazioneEsterna” (Fig. 8.53) che serve ad indicare che lo stato del tracciamento deve essere compreso esaminando la proprietà «org.openscoop2.pdd.transazioni.fileTrace.enabled» del file di configurazione locale «/etc/govway/govway_local.properties». In questo caso è inoltre utilizzabile per attivare la funzionalità di FileTrace su una singola API la *Proprietà “fileTrace.enabled”* definibile sulla singola erogazione o sulla fruizione attraverso la sua valorizzazione con i valori “true” o “false” rispettivamente per attivare o disattivare la funzionalità.

Tracciamento FileTrace

Stato	<input type="text" value="configurazioneEsterna"/>
Filtra per Esiti	<input type="checkbox"/>

Figure8.53: Configurazione del tracciamento su file di log tramite govway_local.properties

Maggiori dettagli sulla configurazione e sul formato dei log vengono forniti nella sezione (*Configurazione Tracciamento su File*).

Configurazione Tracciamento su File

Nella sezione *Configurazione dei Topic* viene descritto il formato del file di configurazione che definisce i log emessi su file. Il path al file di configurazione deve essere indicato nella sezione “Configurazione FileTrace” presente nella maschera di configurazione del tracciamento descritta nella sezione *Registrazione della Transazione*, ridefinendo il comportamento di default come mostrato in figura Fig. 8.54.

Configurazione FileTrace

Stato: ridefinito

File Configurazione: /tmp/pathAlternativo.properties

Buffer dei Messaggi

Solamente se abilitato sarà possibile accedere ai contenuti dei messaggi

Scambiati con il client	abilitato
Header	abilitato
Payload	abilitato
Scambiati con il server	abilitato
Header	abilitato
Payload	abilitato

Figure8.54: Configurazione personalizzata del tracciamento su file di log

Oltre a consentire l’indicazione di un path su file system dove risiede la configurazione della tracciatura su file è possibile configurare il *buffer dei messaggi* che se abilitato consentirà di accedere ai contenuti delle richieste e delle risposte come descritto nella sezione *Informazioni Tracciabili*.

Nel caso invece non vengano ridefiniti gli aspetti di configurazione di FileTrace nella maschera del tracciamento (Fig. 8.55) una volta attivato il tracciamento su file, come descritto nella sezione *Tracciamento su File*, la configurazione dei topic deve essere definita all’interno del file indicato nella proprietà “org.openspcoop2.pdd.transazioni.fileTrace.config” della configurazione locale “/etc/govway/govway_local.properties” (assumendo sia /etc/govway la directory di configurazione indicata in fase di installazione).

Configurazione FileTrace

Stato: default

Figure8.55: Configurazione di default del tracciamento su file di log

Anche nella configurazione di default su file locale “/etc/govway/govway_local.properties” è possibile configurare il *buffer dei messaggi* che se abilitato consentirà di accedere ai contenuti delle richieste e delle risposte come descritto

nella sezione *Informazioni Tracciabili*.

Di seguito un estratto della configurazione.

```

# =====
# FileTrace
...
#
# File di Configurazione
# Il file può essere indicato con un path assoluto o relativo rispetto alla
# directory di configurazione
org.openspcoop2.pdd.transazioni.fileTrace.config=govway.fileTrace.properties
...
...
# Indicazione se nella funzionalità è consentito l'accesso ai contenuti
# -- Fruizioni --
# inRequest/outResponse
org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPD.enabled=true
#org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPD.payload.enabled=true
#org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPD.headers.enabled=true
# outRequest/inResponse
org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPD.connettore.enabled=true
#org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPD.connettore.payload.
#enabled=true
#org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPD.connettore.headers.
#enabled=true
# -- Erogazioni --
# inRequest/outResponse
org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPA.enabled=true
#org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPA.payload.enabled=true
#org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPA.headers.enabled=true
# outRequest/inResponse
org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPA.connettore.enabled=true
#org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPA.connettore.payload.
#enabled=true
#org.openspcoop2.pdd.transazioni.fileTrace.dumpBinarioPA.connettore.headers.
#enabled=true
...
#
# =====

```

Nel caso non siano ridefiniti gli aspetti di configurazione di FileTrace nella maschera del tracciamento (Fig. 8.55), la configurazione di default indicata nel file “/etc/govway/govway_local.properties” è ridefinibile sulla singola erogazione o fruizione di API attraverso le seguenti *Proprietà*:

- *fileTrace.config* : consente di indicare il path su file system dove risiede la configurazione della tracciatura su file; il file indicato può essere un path assoluto o relativo rispetto alla directory di configurazione (per il formato fare riferimento alla sezione *Configurazione dei Topic*);
- *fileTrace.dumpBinario.enabled*: consente di attivare o disattivare la registrazione dei messaggi scambiati con il client: richiesta ingresso e risposta uscita (i valori associabili alla proprietà sono “true” o “false”);
- *fileTrace.dumpBinario.payload.enabled* e *fileTrace.dumpBinario.headers.enabled*: sovrascrivono la proprietà “*fileTrace.dumpBinario.enabled*” fornendo la possibilità di configurare puntualmente il tipo di informazione scambiata con il client (payload o headers) che sarà resa disponibile per la tracciatura (per default entrambi);
- *fileTrace.dumpBinario.connettore.enabled*: consente di attivare o disattivare la registrazione dei messaggi scambiato con l’implementazione di backend dell’API: richiesta uscita e risposta ingresso (i valori associabili

alla proprietà sono “true” o “false”);

- `fileTrace.dumpBinario.connettore.payload.enabled` e `fileTrace.dumpBinario.connettore.headers.enabled`: sovrscrivono la proprietà “`fileTrace.dumpBinario.connettore.enabled`” fornendo la possibilità di configurare puntualmente il tipo di informazione scambiata con il backend (payload o headers) che sarà resa disponibile per la tracciatura (per default entrambi).

Nota: Solamente se il *buffer* dei messaggi (`fileTrace.dumpBinario.enabled` e/o `fileTrace.dumpBinario.connettore.enabled`) è abilitato sarà possibile accedere ai contenuti dei messaggi come descritto nella sezione *Informazioni Tracciabili*.

Nella sezione *Configurazione dei Topic* viene descritto il formato del file di configurazione, mentre nella sezione *Informazioni Tracciabili* sono riportate tutte le informazioni disponibili.

Configurazione dei Topic

Le informazioni inerenti le comunicazioni gestite dal gateway possono essere riversate in uno o più file di log attraverso la definizione di topic.

La configurazione permette di indicare uno o più topic da generare quando il gateway gestisce erogazioni o fruizioni di API. Nell'esempio seguente vengono registrati due topic sulle erogazioni, dove si vuole salvare le informazioni suddividendole tra richiesta e la risposta. Per quanto concerne la registrazione delle fruizioni si attiva invece solamente un unico topic.

```
# Topic
topic.erogazioni=inputRequest,inputResponse
topic.fruizioni=output
```

Per default tutte le comunicazioni gestite dal gateway vengono veicolati nei topic registrati. È possibile escludere il riversamento nel topic di determinate comunicazioni tramite le seguenti proprietà:

- `log.topic.<erogazioni/fruizioni>.<nomeTopic>.requestSent` : se abilita, sul topic indicato verranno riversate solamente informazioni relative a comunicazioni per le quali il gateway è riuscito a spedire la richiesta verso il backend configurato;
- `log.topic.<erogazioni/fruizioni>.<nomeTopic>.[in/out]/[Request/Response]ContentDefined` : se abilitata, verranno riversate informazioni sul topic solo se la richiesta o risposta indicata, in ingresso o uscita dal gateway, possiede un payload http.
- `log.topic.<erogazioni/fruizioni>.<nomeTopic>.trackingPhases` : consente di indicare un elenco, separato da virgola, di fasi di tracciamento (descritte nella sezione *Fasi di Tracciamento*) su cui il topic verrà applicato; se non presenti un topic verrà applicato su tutte le fasi abilitate. Le fasi sono indicabili tramite le seguenti keyword:
 - `inRequest`: rappresenta la fase “*Richiesta ricevuta*”;
 - `outRequest`: rappresenta la fase “*Richiesta in consegna*”;
 - `outResponse`: rappresenta la fase “*Risposta in consegna*”;
 - `postOutResponse`: rappresenta la fase “*Risposta consegnata*”.

Nell'esempio seguente il topic relativo alle fruizioni viene alimentato solamente se il gateway è riuscito a contattare il backend e la richiesta possedeva un payload http (vengono escluse ad esempio le HTTP GET). Sui topic delle erogazioni viene invece attivato solamente il controllo sul payload http per il topic “`inputRequest`”.

```

# Erogazioni (Filtro per Payload HTTP)
topic.erogazioni.inputRequest.inRequestContentDefined=true

# Fruizioni (Filtro per RequestSent + Payload HTTP)
topic.fruizioni.output.requestSent=true
topic.fruizioni.output.outRequestContentDefined=true

```

Nell'esempio seguente viene definito, per le erogazioni, un topic specifico per la fase di consegna della richiesta e uno per la risposta, oltre ad un altro topic in cui confluiscono log relativi alla fase di ricezione della richiesta e risposta consegnata.

```

# Consegna della richiesta
topic.erogazioni.esempioConsegnaRichiesta.trackingPhases=outRequest
# Consegna della risposta
topic.erogazioni.esempioConsegnaRisposta.trackingPhases=outResponse
# Log della transazione
topic.erogazioni.requests.trackingPhases=inRequest,postOutResponse

```

La generazione dei file di log è gestita dalle seguenti proprietà:

- *log.config.file* : file contenente la configurazione *log4j2* nella quale devono essere definite le Category da associare ad ogni topic;
- *log.severity* (default: info): indica il livello di severità (trace/debug/info/warn/error) utilizzato durante il logging;
- *log.topic.<erogazioni/fruizioni>.<nomeTopic>=<categoryLog4j2>* : assegna la category al topic indicato per le erogazioni o fruizioni.

Nell'esempio seguente viene fornito un esempio di associazione di Category ad ogni topic e un estratto di configurazione log4j2 nella quale viene creato un file per ogni category.

```

# Log4j2 Configuration File
log.config.file=govway.fileTrace.log4j2.properties

# trace/debug/info/warn/error
log.severity=info

# Category per ogni topic delle erogazioni
category.topic.erogazioni.inputRequest=fileTrace.inputRequest
category.topic.erogazioni.inputResponse=fileTrace.inputResponse
# Category per ogni topic delle fruizioni
# sintassi: log.topic.fruizioni.<nomeTopic>=<categoryLog4j2>
category.topic.fruizioni.output=fileTrace.output

```

Estratto della configurazione Log4J2 dove per ogni category viene attivata una rotazione giornaliera:

```

name = fileTracePropertiesConfig

# *** inputRequest **
# Category
logger.fileTrace_inputRequest.name = fileTrace.inputRequest
logger.fileTrace_inputRequest.level = DEBUG
logger.fileTrace_inputRequest.additivity = false
logger.fileTrace_inputRequest.appendRef.rolling.ref = fileTrace.inputRequest.
↳ rollingFile

```

(continues on next page)

(continua dalla pagina precedente)

```

# FileAppender
appender.fileTrace_inputRequest.type = RollingFile
appender.fileTrace_inputRequest.name = fileTrace.inputRequest.rollingFile
appender.fileTrace_inputRequest.fileName = /var/govway/log/fileTrace/
  ↳inputRequest.log
appender.fileTrace_inputRequest.filePattern = /var/govway/log/fileTrace/$$
  ↳{date:yyyy-MM}/inputRequest-%d{MM-dd-yyyy}.log.gz
appender.fileTrace_inputRequest.layout.type = PatternLayout
appender.fileTrace_inputRequest.layout.pattern = %m%n
appender.fileTrace_inputRequest.policies.type = Policies
appender.fileTrace_inputRequest.policies.time.type = TimeBasedTriggeringPolicy
appender.fileTrace_inputRequest.strategy.type = DefaultRolloverStrategy

# ** inputResponse**
# Category
logger.fileTrace_inputResponse.name = fileTrace.inputResponse
logger.fileTrace_inputResponse.level = DEBUG
logger.fileTrace_inputResponse.additivity = false
logger.fileTrace_inputResponse.appenderef.rolling.ref = fileTrace.
  ↳inputResponse.rollingFile
# FileAppender
appender.fileTrace_inputResponse.type = RollingFile
appender.fileTrace_inputResponse.name = fileTrace.inputResponse.rollingFile
appender.fileTrace_inputResponse.fileName = /var/govway/log/fileTrace/
  ↳inputResponse.log
appender.fileTrace_inputResponse.filePattern = /var/govway/log/fileTrace/$$
  ↳{date:yyyy-MM}/inputResponse-%d{MM-dd-yyyy}.log.gz
appender.fileTrace_inputResponse.layout.type = PatternLayout
appender.fileTrace_inputResponse.layout.pattern = %m%n
appender.fileTrace_inputResponse.policies.type = Policies
appender.fileTrace_inputResponse.policies.time.type = TimeBasedTriggeringPolicy
appender.fileTrace_inputResponse.strategy.type = DefaultRolloverStrategy

# ** output **
# Category
logger.fileTrace_output.name = fileTrace.output
logger.fileTrace_output.level = DEBUG
logger.fileTrace_output.additivity = false
logger.fileTrace_output.appenderef.rolling.ref = fileTrace.output.rollingFile
# FileAppender
appender.fileTrace_output.type = RollingFile
appender.fileTrace_output.name = fileTrace.output.rollingFile
appender.fileTrace_output.fileName = /var/govway/log/fileTrace/output.log
appender.fileTrace_output.filePattern = /var/govway/log/fileTrace/${date:yyyy-
  ↳MM}/output-%d{MM-dd-yyyy}.log.gz
appender.fileTrace_output.layout.type = PatternLayout
appender.fileTrace_output.layout.pattern = %m%n
appender.fileTrace_output.policies.type = Policies
appender.fileTrace_output.policies.time.type = TimeBasedTriggeringPolicy
appender.fileTrace_output.strategy.type = DefaultRolloverStrategy

```

Per ogni topic non rimane che definire le informazioni che si desidera tracciare attraverso la proprietà `format.topic.<erogazioni/fruizioni>.<nomeTopic>`. Le informazioni possono essere definite attraverso costanti o

tramite quanto indicato nella sezione *Informazioni Tracciabili*.

Di seguito un esempio:

```
format.topic.erogazioni.inputRequest="req"|"${log:transactionId}"|"govway"|"$  
↳ ${log:inRequestDateZ(yyyy-MM-dd HH:mm:ss:SSS,UTC)}"|"${log:inRequestDate(Z)}"|"  
↳ ${log:forwardedIP}"|"HTTP/1.1"|"${log:httpMethod}"  
format.topic.erogazioni.inputResponse="res"|"${log:transactionId}"|"govway"|"$  
↳ ${log:inRequestDateZ(yyyy-MM-dd HH:mm:ss:SSS,UTC)}"|"${log:inRequestDate(Z)}"|"  
↳ ${log:forwardedIP}"|"HTTP/1.1"|"${log:httpMethod}"|"${log:outHttpStatus}"  
format.topic.fruizioni.output="output"|"${log:transactionId}"|"govway"|"$  
↳ ${log:inRequestDateZ(yyyy-MM-dd HH:mm:ss:SSS,UTC)}"|"${log:inRequestDate(Z)}"|"  
↳ ${log:forwardedIP}"|"HTTP/1.1"|"${log:httpMethod}"|"${log:inHttpStatus}"
```

Le informazioni prodotte ad esempio per il topic `inputRequest` saranno le seguenti:

```
"req"|"b6cddd758-342c-4599-ae95-33a781730b3f"|"govway"|"2020-06-26 12:46:50:629  
↳ "+"|"+0200"|"192.168.1.2"|"HTTP/1.1"|"POST"  
"req"|"2a9dc253-9dd5-458b-8689-edee7c9ba139"|"govway"|"2020-06-26 12:47:50:561  
↳ "+"|"+0200"|"192.168.1.2"|"HTTP/1.1"|"POST"  
"req"|"eeedb92b-66b5-451e-8266-ade2cf1f34ce"|"govway"|"2020-06-26 12:47:53:291  
↳ "+"|"+0200"|"192.168.1.19"|"HTTP/1.1"|"POST"  
"req"|"b4355a45-71cc-4293-b3b7-a4622af8ea84"|"govway"|"2020-06-26 12:48:00:102  
↳ "+"|"+0200"|"192.168.1.22"|"HTTP/1.1"|"POST"
```

Nell'esempio appena riportato si può notare come i 3 topic utilizzano una parte comune. È possibile ottimizzare le informazioni configurate attraverso la definizione di proprietà `format.property.<posizione>.<nomeProprietà>=<valoreProprietà>`. Le proprietà verranno risolte in ordine lessicografico rispetto alla posizione indicata, in modo da garantire la corretta risoluzione se si hanno proprietà che sono definite tramite altre proprietà. Maggiori dettagli vengono forniti nella sezione *Informazioni raggruppate in Proprietà*.

Di seguito il precedente esempio ridefinito tramite proprietà:

```
# properties  
format.property.001.common.govway-id=govway  
format.property.001.common.id="${log:transactionId}"|"${log:property(common.  
↳ govway-id)}"  
format.property.002.common.data="${log:inRequestDateZ(yyyy-MM-dd HH:mm:ss:SSS,  
↳ UTC)}"|"${log:inRequestDate(Z)}"  
format.property.003.common.remoteIP-protocol-method="${log:forwardedIP}"|  
↳ "HTTP/1.1"|"${log:httpMethod}"  
format.property.004.common=${log:property(common.id)}|${log:property(common.  
↳ data)}|${log:property(common.remoteIP-protocol-method)}  
  
# topic  
format.topic.erogazioni.inputRequest="req"|"${log:property(common)}"  
format.topic.erogazioni.inputResponse="res"|"${log:property(common)}"|"$  
↳ ${log:outHttpStatus}"  
format.topic.fruizioni.output="output"|"${log:property(common)}"|"$  
↳ ${log:inHttpStatus}"
```

È infine possibile definire l'escape di caratteri che possono essere presenti nelle informazioni da tracciare tramite la proprietà `format.escape.<char>=<charEscaped>`.

Di seguito un esempio di configurazione che effettua l'escape del carattere “\»” sostituendolo con “\\»”:

```
format.escape."=\\"
```

Nota: In caso di configurazione globale (attivata da file govway_local.properties come indicato in *Configurazione Tracciamento su File*), anche se la configurazione viene modificata non sarà utilizzata dal Gateway fino ad un suo riavvio. È possibile forzare la rilettura immediata accendendo alla voce “Strumenti - Runtime” della console di gestione e selezionando “Aggiorna la configurazione” nella sezione «Informazioni Tracciamento - File Trace» (Fig. 8.56).

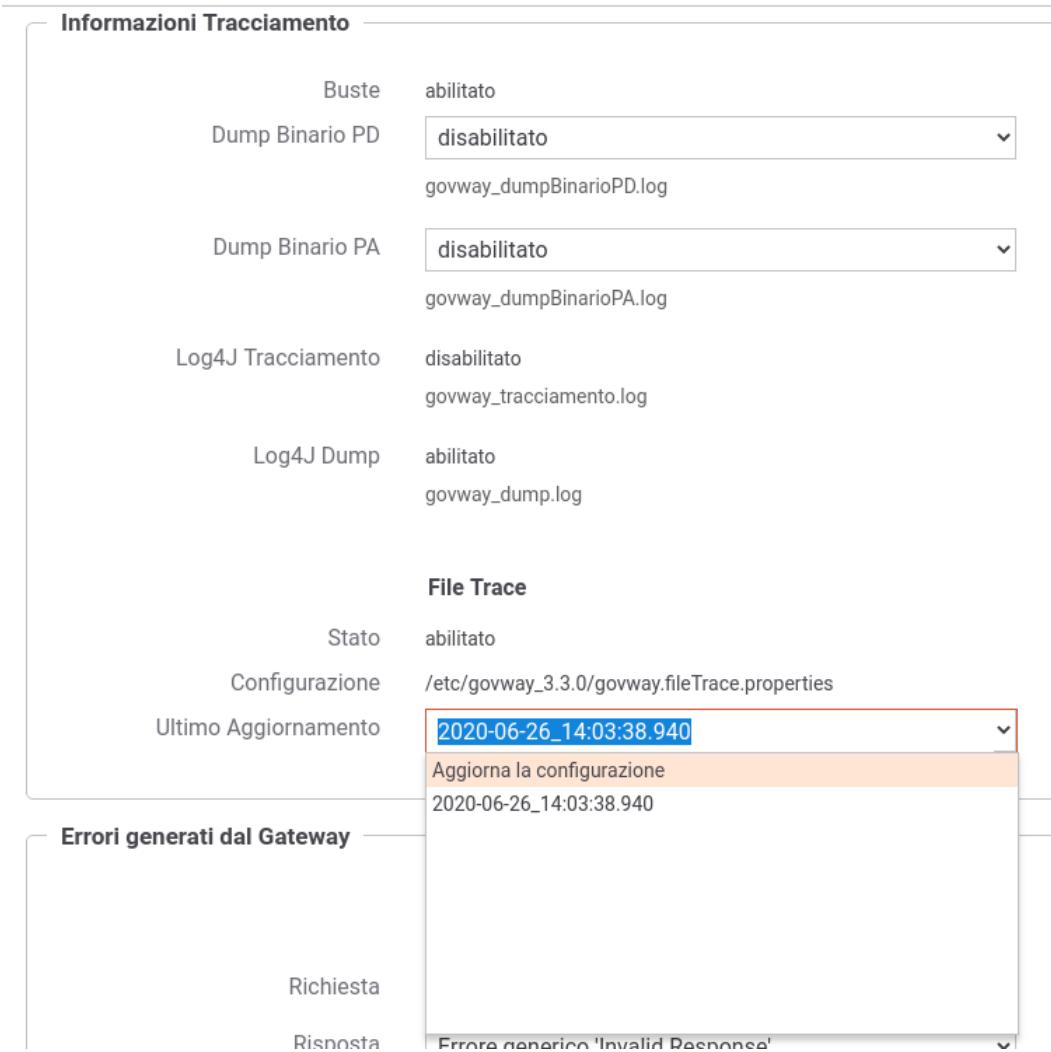


Figure 8.56: Aggiornamento della Configurazione di File Trace

Anche in caso di configurazione locale (attivata tramite le *Proprietà* come indicato in *Configurazione Tracciamento su File*) la configurazione modificata non sarà utilizzata dal Gateway fino ad un suo riavvio. È possibile forzare la rilettura immediata accendendo alla voce “Strumenti - Runtime” della console di gestione e cliccando sulla voce “Svuota le Cache”.

Informazioni Tracciabili

Le informazioni inerenti le comunicazioni gestite dal gateway, che possono essere riversate nei file di log associati ai topic, sono indicabili all'interno del formato di un topic tramite una delle seguenti sintassi:

- `${log:<id>}` : viene registrata la risorsa con l'identificativo indicato.
- `${log:<id>(defaultValue)}` : viene registrata la risorsa con l'identificativo indicato; se la risorsa non è valorizzata, viene registrato il valore di default fornito come parametro
- `${log:<id>(parameters ...)}` : viene registrata la risorsa con l'identificativo indicato, il cui valore può essere personalizzato rispetto ad alcuni parametri.

Le informazioni possono essere registrate codificate in base64 utilizzando il prefisso “logBase64” invece di “log”:

- `${logBase64:<id>}`
- `${logBase64:<id>(defaultValue)}`
- `${logBase64:<id>(parameters ...)}`

L'esempio seguente definisce un topic che utilizza i formati precedentemente indicati. Viene registrato l'identificativo di transazione (informazione acceduta puntualmente), la data di accesso all'API (informazione formattata rispetto ai parametri “yyyy-MM-dd HH:mm:ss:SSS” e “UTC”), il contenuto della richiesta codificato in base64 e l'identificativo di correlazione applicativa se presente o la costante “ExampleDefaultValue” altrimenti.

```
format.topic.erogazioni.example=${log:transactionId}|${log:inRequestDateZ(yyyy-  
MM-dd HH:mm:ss:SSS,UTC):ss:SSS,UTC}"|${logBase64:inRequestContent}|$  
|${log:applicationId(ExampleDefaultValue)}|
```

Le informazioni prodotte saranno le seguenti:

```
"b6cdd758-342c-4599-ae95-33a781730b3f"|"2020-06-26 12:46:50:629  
| eyJtaXR0ZW50ZSI6IkF2dm9jYXR1cmEgR2VuZXJhbGUgZGVsbG8gU3RhdG8iLCJkZXN0a...  
| ExampleDefaultValue  
"2a9dc253-9dd5-458b-8689-edee7c9ba139"|"2020-06-26 12:47:50:561  
| eyJtaXR0ZW50ZSI6IkF2dm9jYXR1cmEgR2VuZXJhbGUgZGVsbG8gU3RhdG8iLCJkZXN0a...  
| ExampleDefaultValue  
"eedeb92b-66b5-451e-8266-ade2cf1f34ce"|"2020-06-26 12:47:53:291  
| eyJtaXR0ZW50ZSI6IkF2dm9jYXR1cmEgR2VuZXJhbGUgZGVsbG8gU3RhdG8iLCJkZXN0a...  
| ApplicationXXX3  
"b4355a45-71cc-4293-b3b7-a4622af8ea84"|"2020-06-26 12:48:00:102  
| eyJtaXR0ZW50ZSI6IkF2dm9jYXR1cmEgR2VuZXJhbGUgZGVsbG8gU3RhdG8iLCJkZXN0a...  
| ExampleDefaultValue
```

Di seguito vengono indicati tutti gli identificativi delle informazioni disponibili con i possibili parametri.

Nota: Gli identificativi per cui non vengono specificati parametri sono sempre disponibili nella modalità con o senza la definizione del valore di default.

Identificativi

- `transactionId`: identificativo della transazione;
- `requestId`: identificativo del messaggio di richiesta;
- `responseId`: identificativo del messaggio di risposta;
- `correlationId`: identificativo che correla molteplici transazioni;

- asyncId: identificativo utilizzato in profili asincroni;
- requestApplicationId: identificativo di correlazione applicativa della richiesta;
- responseApplicationId: identificativo di correlazione applicativa della risposta;
- applicationId: requestApplicationId + responseApplicationId;
- clusterId: identificativo del nodo in una installazione in cluster del gateway.

Esito

- inHttpStatus: codice http di risposta ritornato dal backend contattato dal gateway;
- inHttpReason: http reason associato al codice di risposta ritornato dal backend;
- outHttpStatus: codice http di risposta ritornato al client dal gateway;
- outHttpReason: http reason associato al codice di risposta ritornato al client;
- resultClass: classe a cui appartiene l'esito della transazione tra OK, KO e FAULT;
- resultClassOk: indicazione se l'esito della transazione appartiene alla classe OK (true/false);
- resultClassKo: indicazione se l'esito della transazione appartiene alla classe KO (true/false);
- resultClassFault: indicazione se l'esito della transazione appartiene alla classe FAULT (true/false);
- result: esito della transazione (codifica GovWay);
- resultCode: esito della transazione (codifica numerica di GovWay);
- errorDetail: dettaglio dell'errore avvenuto durante la gestione della transazione (per maggiori informazioni si rimanda al manuale sulla console di monitoraggio nella sezione mon_dettaglioErrore);
- transactionType: tipo della transazione (Standard, Sistema ...).
- trackingPhase: fase di tracciamento (IN_REQUEST, OUT_REQUEST, OUT_RESPONSE, POST_OUT_RESPONSE).

Diagnostici

Di seguito vengono indicati gli identificativi che consentono di accedere ai diagnostici emessi da GovWay durante la gestione della richiesta:

- diagnostics: elenco completo dei messaggi diagnostici emessi;
- errorDiagnostics: elenco dei messaggi diagnostici di sola severità errore.

Ogni diagnostico viene fornito nella forma seguente e separato dagli altri tramite un ritorno a capo (configurazione di default):

<livelloSeverità> <dataEmissione> <codiceDiagnostico> <messaggio>

ad esempio:

```
infoIntegration 2020-09-10T14:15:51.605Z 004074 Autenticazione [basic] in
↳ corso ( BasicUsername 'prova' ) ...
infoIntegration 2020-09-10T14:15:51.606Z 004075 Autenticazione [basic] ↳
↳ effettuata con successo (in cache)
```

L'elenco dei diagnostici sono accessibili anche con i seguenti parametri:

- (separator): consente di indicare un separatore dei diagnostici differente da quello di default (ritorno a capo)
- (separator, format): oltre al separatore, consente di indicare il formato della data (es. yyyy-MM-dd HH:mm:ss:SSS.Z);

- (separator, format, timeZone): oltre al separatore e al formato della data (es. yyyy-MM-dd HH:mm:ss:SSS) consente di indicare il time zone (es. UTC).

Date

- acceptedRequestDate: data in cui la richiesta è pervenuta sul gateway;
- inRequestDate e inRequestStartTime: data in cui la richiesta è iniziata ad essere gestita sul gateway;
- inRequestEndTime: data in cui la richiesta è stata completamente ricevuta sul gateway;
- outRequestDate e outRequestStartTime: data in cui la richiesta viene inoltrata dal gateway al backend;
- outRequestEndTime: data in cui la richiesta è stata consegnata al backend;
- acceptedResponseDate: data in cui la risposta è pervenuta sul gateway;
- inResponseDate e inResponseStartTime: data in cui la risposta è iniziata ad essere gestita sul gateway;
- inResponseEndTime: data in cui la risposta è stata completamente ricevuta sul gateway;
- outResponseStartTime: data in cui la risposta viene ritornata al client;
- outResponseDate e outResponseEndTime: data in cui la risposta è stata completamente consegnata al client.

Tutte le date indicate sono accessibili anche con i seguenti parametri:

- (format): formato della data (es. yyyy-MM-dd HH:mm:ss:SSS.Z);
- (format, timeZone): formato della data (es. yyyy-MM-dd HH:mm:ss:SSS) + time zone (es. UTC).

Elapsed Time

- elapsedTime: tempo di risposta complessivo trascorso tra l'ingresso della richiesta nel gateway e la risposta ritornata al client;
- apiElapsedTime: tempo di risposta del backend;
- gatewayLatency: latenza introdotta dal gateway rispetto al tempo di risposta del backend.

Tutte le informazioni sono ritornate in millisecondi. È possibile ottenere le medesime informazioni in un altro formato di tempo utilizzando i seguenti suffissi:

- <elapsedTime>S: tempo in secondi;
- <elapsedTime>Ms: tempo in millisecondi (è il default);
- <elapsedTime>uS: tempo in microsecondi;
- <elapsedTime>nS: tempo in nanosecondi.

Dominio

- domain: identificativo del dominio interno che ha gestito l'erogazione o la fruizione;
- organizationId: identificativo del soggetto, di dominio interno, che ha gestito l'erogazione o la fruizione (identificativo nel formato previsto dal profilo di interoperabilità);
- organization: nome del soggetto, di dominio interno, che ha gestito l'erogazione o la fruizione;
- organizationType: tipo del soggetto, di dominio interno, che ha gestito l'erogazione o la fruizione;
- role: indica se la transazione rappresenta una “erogazione” o “fruizione”;
- contextPropertiesKeys: nomi delle proprietà definite nel contesto;
- contextProperties: proprietà (nome=valore) definite nel contesto separate da uno spazio;
- contextProperties(propertySeparator, valueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;

- contextProperty(nomeProprietà): valore della proprietà indicata come parametro.

API

- apiProtocol: indica se l'API è di tipo “rest” o “soap”;
- apiId: identificativo dell'API, secondo il formato previsto dal profilo di interoperabilità;
- api: nome dell'API;
- apiVersion: versione dell'API;
- apiType: tipo dell'API;
- apiInterface: identificativo dell'interfaccia implementata dall'erogazione o dalla fruizione (contiene nome, versione e soggetto referente);
- apiInterfaceId: identificativo dell'interfaccia implementata dall'erogazione o dalla fruizione secondo il formato previsto dal profilo di interoperabilità;
- apiPropertiesKeys: nomi delle proprietà definite sull'erogazione o sulla fruizione;
- apiProperties: proprietà (nome=valore) definite sull'erogazione o sulla fruizione separate da uno spazio;
- apiProperties(propertySeparator, valueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;
- apiProperty(nomeProprietà): valore della proprietà indicata come parametro;
- action: identificativo della risorsa (API Rest) o dell'azione (API Soap);
- httpMethod: metodo http invocato;
- outURL: url utilizzata dal gateway per invocare il backend (se presenti, contiene anche i parametri della url);
- inURL: url utilizzata dal client per invocare il gateway (se presenti, contiene anche i parametri della url);
- inFunction: indica il tipo di canale (in, out, out/xml2soap) utilizzato dal client per invocare il gateway;
- collaborationProfileCode: indica il profilo di collaborazione associato all'azione di una API Soap (Oneway/Sincrono/AsincronoSimmetrico/AsincronoAsimmetrico);
- collaborationProfile: indica il profilo di collaborazione associato all'azione di una API Soap con la terminologia del profilo di interoperabilità dell'API;
- profile: profilo di interoperabilità in cui è stata registrata l'API;
- profileLabel: nome descrittivo del profilo di interoperabilità in cui è stata registrata l'API;
- interface: identificativo dell'erogazione o della fruizione;
- outConnectorName: nome del connettore multiplo selezionato per la consegna.

Soggetti

- providerId: identificativo del soggetto erogatore, secondo il formato previsto dal profilo di interoperabilità;
- provider: nome del soggetto erogatore;
- providerType: tipo del soggetto erogatore;
- providerDomain: identificativo del dominio erogatore;
- providerURI: uri associata al soggetto erogatore;
- providerPropertiesKeys: nomi delle proprietà definite sul soggetto fruitore;
- providerProperties: proprietà (nome=valore) definite sul soggetto fruitore separate da uno spazio;

- providerProperties(propertySeparator, valueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;
- providerProperty(nomeProprietà): valore della proprietà indicata come parametro;
- senderId: identificativo del soggetto fruitore, secondo il formato previsto dal profilo di interoperabilità;
- sender: nome del soggetto fruitore;
- senderType: tipo del soggetto fruitore;
- senderDomain: identificativo del dominio fruitore;
- senderURI: uri associata al soggetto fruitore;
- senderPropertiesKeys: nomi delle proprietà definite sul soggetto fruitore;
- senderProperties: proprietà (nome=valore) definite sul soggetto fruitore separate da uno spazio;
- senderProperties(propertySeparator, valueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;
- senderProperty(nomeProprietà): valore della proprietà indicata come parametro.

Mittente

- application: identificativo dell'applicativo richiedente;
- applicationPropertiesKeys: nomi delle proprietà definite sull'applicativo richiedente;
- applicationProperties: proprietà (nome=valore) definite sull'applicativo separate da uno spazio;
- applicationProperties(propertySeparator, valueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;
- applicationProperty(nomeProprietà): valore della proprietà indicata come parametro;
- credentials: credenziali presenti nella richiesta;
- principal: identificato con cui l'applicativo è stato autenticato;
- principalAuthType: tipo di autenticazione (basic/ssl/principal) con cui l'applicativo è stato autenticato;
- clientCertificateSubjectDN: distinguished name del subject relativo al certificato tls client;
- clientCertificateSubjectCN: common name del subject relativo al certificato tls client;
- clientCertificateSubjectDNInfo(String oid): ritorna l'informazione indicata come parametro relativa al subject del certificato tls client;
- clientCertificateIssuerDN: distinguished name dell'issuer relativo al certificato tls client;
- clientCertificateIssuerCN: common name dell'issuer relativo al certificato tls client;
- clientCertificateIssuerDNInfo(String oid): ritorna l'informazione indicata come parametro relativa all'issuer del certificato tls client;
- attribute(nomeAttributo): valore dell'attributo indicato come parametro (informazione disponibile solamente se nell'erogazione/fruitore è stata configurata una sola A.A.);
- attributeByAA(nomeAttributeAuthority, nomeAttributo): valore dell'attributo recuperato tramite l'AttributeAuthority indicata come parametro (informazione disponibile solamente se nell'erogazione/fruitore è stata configurata più di una A.A.);
- clientIP: indirizzo IP del client;
- forwardedIP: indirizzo IP presente nella richiesta in uno degli header http appartenente alla classe «Forwarded-For» o «Client-IP»;

- requesterIP: (o ipRequester) rappresenta l'indirizzo IP del richiedente e assumerà la prima informazione valorizzata, trovata nella richiesta, nel seguente ordine:
 - forwardedIP
 - clientIP
- requester: rappresenta il richiedente della richiesta e assumerà la prima informazione valorizzata, trovata nella richiesta, nel seguente ordine (per maggiori informazioni si rimanda al manuale sulla console di monitoraggio nella sezione mon_richiedente):
 - tokenUsername: username presente nel token;
 - tokenClient: identificativo dell'applicativo identificato tramite il clientId presente nel token;
 - pdndOrganizationName: nome dell'organizzazione del client ottenuto risolvendo il clientId presente nel token tramite la consultazione delle API PDND;
 - application: identificativo dell'applicativo richiedente identificato tramite l'autenticazione di trasporto;
 - tokenClientId: clientId presente nel token nel caso di client credentials grant type (claims clientId e sub presentano lo stesso valore);
 - tokenSubject[@tokenIssuer]: subject presente nel token; viene aggiunto anche un suffisso @tokenIssuer se è presente anche un issuer nel token;
 - principal: identificativo (credenziali) con cui l'applicativo è stato autenticato; se il tipo di autenticazione di trasporto risulta essere "ssl" viene ritornato il valore dell'attributo CN.

Validazione Token

- token: token OAuth2 presente nella richiesta;
- tokenIssuer: issuer presente nel token;
- tokenSubject: subject presente nel token;
- tokenClientId: clientId presente nel token;
- tokenClientApplication: identificativo dell'applicativo identificato tramite il clientId presente nel token;
- tokenClientApplicationPropertiesKeys: nomi delle proprietà definite sull'applicativo identificato tramite il clientId;
- tokenClientApplicationProperties: proprietà (nome=valore) definite sull'applicativo separate da uno spazio;
- tokenClientApplicationProperties(propertySeparator, valueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;
- tokenClientApplicationProperty(nomeProprietà): valore della proprietà indicata come parametro;
- tokenClientOrganizationId: identificativo del soggetto proprietario dell'applicativo identificato tramite il clientId, secondo il formato previsto dal profilo di interoperabilità;
- tokenClientOrganization: nome del soggetto proprietario dell'applicativo identificato tramite il clientId;
- tokenClientOrganizationType: tipo del soggetto proprietario dell'applicativo identificato tramite il clientId;
- tokenClientOrganizationPropertiesKeys: nomi delle proprietà definite sul soggetto proprietario dell'applicativo identificato tramite il clientId;
- tokenClientOrganizationProperties: proprietà (nome=valore) definite sul soggetto separate da uno spazio;
- tokenClientOrganizationProperties(propertySeparator, valueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;
- tokenClientOrganizationProperty(nomeProprietà): valore della proprietà indicata come parametro.

- tokenUsername: username presente nel token;
- tokenMail: eMail presente nel token;
- tokenClaim(nomeClaim): valore del claim indicato come parametro e presente nel token;

Nota: Le informazioni seguenti sono presenti solamente se è stata abilitata la validazione JWT del token

- tokenRaw: JWT token presente nella richiesta;
- tokenHeaderRaw: porzione dell'header relativa al token JWT presente nella richiesta, in formato base64;
- tokenPayloadRaw: porzione del payload relativa al token JWT presente nella richiesta, in formato base64;
- tokenDecodedHeader: contenuto decodificato dell'header presente nel token JWT;
- tokenDecodedPayload: contenuto decodificato del payload presente nel token JWT;
- tokenHeaderClaim(nomeClaim): valore del claim indicato come parametro e presente nell'header del token JWT;
- tokenPayloadClaim(nomeClaim): valore del claim indicato come parametro e presente nel payload del token JWT;
- tokenHeaderClaims(): claims (nome=valore) presenti nell'header del token JWT;
- tokenHeaderClaims(claimSeparator, nameValueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;
- tokenPayloadClaims(): claims (nome=valore) presenti nel payload del token JWT;
- tokenPayloadClaims(claimSeparator, nameValueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;
- tokenCertificateSubjectDN: distinguished name del subject relativo al certificato con cui è stato firmato il token JWT;
- tokenCertificateSubjectCN: common name del subject relativo al certificato con cui è stato firmato il token JWT;
- tokenCertificateSubjectDNInfo(String oid): ritorna l'informazione indicata come parametro relativa al subject del certificato con cui è stato firmato il token JWT;
- tokenCertificateIssuerDN: distinguished name dell'issuer relativo al certificato con cui è stato firmato il token JWT;
- tokenCertificateIssuerCN: common name dell'issuer relativo al certificato con cui è stato firmato il token JWT;
- tokenCertificateIssuerDNInfo(String oid): ritorna l'informazione indicata come parametro relativa all'issuer del certificato con cui è stato firmato il token JWT.

Negoziazione Token

- retrievedAccessToken: access token ottenuto dall'authorization server configurato nella Token Policy associata al connettore;
- retrievedTokenClaim(nomeClaim): valore del claim indicato come parametro e presente nella risposta ritornata dall'authorization server;
- retrievedTokenRequestTransactionId: identificativo della transazione che ha originato la richiesta verso l'authorization server;
- retrievedTokenRequestGrantType: tipo di grant type utilizzato nella negoziazione del token (clientCredentials, usernamePassword, rfc7523_x509, rfc7523_clientSecret);

- retrievedTokenRequestJwtClientAssertion: asserzione jwt generata durante una negoziazione con grant type “rfc7523_x509”;
- retrievedTokenRequestClientId: clientId utilizzato durante la negoziazione del token;
- retrievedTokenRequestClientToken: bearer token utilizzato durante la negoziazione del token;
- retrievedTokenRequestUsername: username utilizzato durante una negoziazione del token con grant type “usernamePassword”;
- retrievedTokenRequestUrl: endpoint dell’authorization server.

Informazioni specifiche dei Profili di Interoperabilità

- requestPropertiesKeys: nomi delle proprietà associate alla traccia della richiesta;
- requestProperties: proprietà (nome=valore), associate alla traccia della richiesta, separate da uno spazio;
- requestProperties(propertySeparator, valueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;
- requestProperty(nomeProprietà): valore della proprietà indicata come parametro;
- responsePropertiesKeys: nomi delle proprietà associate alla traccia della risposta;
- responseProperties: proprietà (nome=valore), associate alla traccia della risposta, separate da uno spazio;
- responseProperties(propertySeparator, valueSeparator): simile alla precedente opzione, consente di indicare i separatori utilizzati;
- responseProperty(nomeProprietà): valore della proprietà indicata come parametro.

ModI

- tokenModI<tokenType>Raw: security token presente nella richiesta;
- tokenModI<tokenType>CertificateSubjectDN: distinguished name del subject relativo al certificato con cui è stato firmato il security token;
- tokenModI<tokenType>CertificateSubjectCN: common name del subject relativo al certificato con cui è stato firmato il security token;
- tokenModI<tokenType>CertificateSubjectDNInfo(String oid): ritorna l’informazione indicata come parametro relativa al subject del certificato con cui è stato firmato il security token;
- tokenModI<tokenType>CertificateIssuerDN: distinguished name dell’issuer relativo al certificato con cui è stato firmato il security token;
- tokenModI<tokenType>CertificateIssuerCN: common name dell’issuer relativo al certificato con cui è stato firmato il security token;
- tokenModI<tokenType>CertificateIssuerDNInfo(String oid): ritorna l’informazione indicata come parametro relativa all’issuer del certificato con cui è stato firmato il security token.

I tipi di token disponibili sono:

- Authorization: security token ricevuto nell’header HTTP “Authorization”;
- Integrity: security token ricevuto nell’header HTTP “Agid-JWT-Signature”;
- Audit: security token ricevuto nell’header HTTP “Agid-JWT-TrackingEvidence”;
- Soap: security token ricevuto nell’header SOAP;

Per i tipi di token “Authorization” e “Integrity”, relativi ad API di tipo REST, sono disponibili anche le seguenti informazioni:

- `tokenModI<tokenType>HeaderRaw`: porzione dell'header relativa al security token presente nella richiesta, in formato base64;
- `tokenModI<tokenType>PayloadRaw`: porzione del payload relativa al security token presente nella richiesta, in formato base64;
- `tokenModI<tokenType>DecodedHeader`: contenuto decodificato dell'header presente nel security token;
- `tokenModI<tokenType>DecodedPayload`: contenuto decodificato del payload presente nel security token;
- `tokenModI<tokenType>HeaderClaim(nomeClaim)`: valore del claim indicato come parametro e presente nell'header del security token;
- `tokenModI<tokenType>PayloadClaim(nomeClaim)`: valore del claim indicato come parametro e presente nel payload del security token;
- `tokenModI<tokenType>HeaderClaims()`: claims (nome=valore) presenti nell'header del security token;
- `tokenModI<tokenType>HeaderClaims(claimSeparator, nameValueSeparator)`: simile alla precedente opzione, consente di indicare i separatori utilizzati;
- `tokenModI<tokenType>PayloadClaims()`: claims (nome=valore) presenti nel payload del security token;
- `tokenModI<tokenType>PayloadClaims(claimSeparator, nameValueSeparator)`: simile alla precedente opzione, consente di indicare i separatori utilizzati;

PDND

Di seguito le indicazioni su come accedere alle informazioni riguardanti il client e l'organizzazione recuperate tramite le API PDND.

- `pdndOrganizationName`: nome dell'organizzazione a cui appartiene il client;
- `pdndOrganizationId`: identificativo PDND (uuid) dell'organizzazione a cui appartiene il client;
- `pdndOrganizationCategory`: categoria in cui è stata classificata dalla PDND l'organizzazione a cui appartiene il client;
- `pdndOrganizationExternalId` e `pdndOrganizationExternalOrigin`: rispettivamente identificativo dell'organizzazione e tipo di repository esterno a cui l'identificativo appartiene;
- `pdndOrganizationJson`: consente di ottenere la risposta json ottenuta dalla PDND invocando l'operazione “GET /organizations/{organizationId}”.
- `pdndClientId`: identificativo PDND (uuid) del client;
- `pdndClientConsumerId`: identificativo PDND (uuid) dell'organizzazione a cui appartiene il client;
- `pdndClientJson`: consente di ottenere la risposta json ottenuta dalla PDND invocando l'operazione “GET /clients/{clientId}”;

Messaggi

- `duplicateRequest`: numero di volte in cui una richiesta con stesso “`requestId`” è stata ricevuta dal gateway;
- `duplicateResponse`: numero di volte in cui una risposta con stesso “`responseId`” è stata ricevuta dal gateway;
- `getInFault`: eventuale SOAP Fault o Problem Detail RFC 7807 ricevuto dal backend;
- `getOutFault`: eventuale SOAP Fault o Problem Detail RFC 7807 ritornato al client.

È inoltre possibile accedere alle seguenti informazioni riguardanti i singoli messaggi in ingresso o uscita dal gateway:

- `<messageType>ContentType`: valore dell'header http “Content-Type”;
- `<messageType>Content`: payload http;
- `<messageType>Size`: dimensione del payload http;

- <messageType>Header(name): valore dell'header http indicato come parametro;
- <messageType>Header(name, multiValueSeparator): elenco di valori, separati con il carattere indicato nel parametro “multiValueSeparator”, relativi agli header http che possiedono il nome indicato dal parametro “name”;
- <messageType>Headers: elenco degli headers http nel formato <nome>=<valore> separati dal carattere “,” ;
- <messageType>Headers(headersSeparator, nameValueSeparator, prefix, suffix): i parametri permettono di personalizzare il formato degli headers http.

I tipi di messaggi disponibili sono:

- inRequest: richiesta ricevuta sul gateway;
- outRequest: richiesta inoltrata al backend;
- inResponse: risposta ricevuta dal backend;
- outResponse: risposta ritornata a client.

Nota: È disponibile inoltre l'operazione “remove<MessageType>Header(name)” (utilizzare la prima lettera maiuscola per MessageType) che consente di ottenere il valore dell'header http indicato come parametro e nello stesso tempo di eliminarlo dalla lista di header ritornati da un successivo accesso alla risorsa “<messageType>Headers”.

Nota: Le informazioni sui 4 tipi di messaggio saranno disponibili solamente se è stata abilitata la funzionalità di dump per ciascun tipo nel file di configurazione locale “/etc/govway/govway_local.properties” (assumendo sia /etc/govway la directory di configurazione indicata in fase di installazione) o tramite le *Proprietà* come indicato in *Configurazione Tracciamento su File*.

Ambiente

- hostAddress(): InetAddress.getLocalHost().getHostAddress();
- hostName(): InetAddress.getLocalHost().getHostName();
- systemProperty(nomeProprietà): valore della proprietà di sistema indicata come parametro;
- javaProperty(nomeProprietà): valore della proprietà java indicata come parametro.

Informazioni raggruppate in Proprietà

Le informazioni associabili ai topic devono essere definite attraverso la sintassi descritta nella sezione *Informazioni Tracciabili*.

Di seguito un esempio di definizione di 2 topic:

```
format.topic.erogazioni.inputRequest="req"|"${log:transactionId}"|"govway"|"${log:inRequestDateZ(yyyy-MM-dd HH:mm:ss:SSS,UTC)}"|"${log:inRequestDate(Z)}"|"${log:forwardedIP}"|"HTTP/1.1"|"${log:httpMethod}"|${log:outHttpStatus}
format.topic.erogazioni.inputResponse="res"|"${log:transactionId}"|"govway"|"${log:inRequestDateZ(yyyy-MM-dd HH:mm:ss:SSS,UTC)}"|"${log:inRequestDate(Z)}"|"${log:forwardedIP}"|"HTTP/1.1"|"${log:httpMethod}"|"${log:outHttpStatus}"
```

Nell'esempio appena riportato si può notare come i 2 topic utilizzano una parte comune ripetuta. È possibile migliorare la scrittura del file di definizione dei topic esplicitando una volta sola l'insieme di informazioni comuni tramite una proprietà che potrà essere riferita nella definizione di ogni topic.

Per definire una proprietà deve essere utilizzata la sintassi:

- “*format.property.<posizione>.<nomeProprietà>=<valoreProprietà>*”

Le proprietà verranno risolte in ordine lessicografico rispetto alla posizione indicata, in modo da garantire la corretta risoluzione se si hanno proprietà che sono definite tramite altre proprietà.

Di seguito il precedente esempio ridefinito tramite proprietà.

```
# properties
format.property.001.common.govway-id=govway
format.property.002.common.id="${log:transactionId}"|"${log:property(common.
˓→govway-id)}"
format.property.003.common.data="${log:inRequestDateZ(yyyy-MM-dd HH:mm:ss:SSS,
˓→UTC)}"|"${log:inRequestDate(Z)}"
format.property.004.common.remoteIP-protocol-method="${log:forwardedIP}"|
˓→"HTTP/1.1"|"${log:httpMethod}"
format.property.005.common=${log:property(common.id)}|${log:property(common.
˓→data)}|${log:property(common.remoteIP-protocol-method)}

# topic
format.topic.erogazioni.inputRequest="req"|"${log:property(common)}"
format.topic.erogazioni.inputResponse="res"|"${log:property(common)}|"|
˓→${log:outHttpStatus}"
```

È inoltre possibile definire l'escape di caratteri che possono essere presenti nelle informazioni da tracciare tramite la proprietà “*format.escape.<char>=<charEscaped>*”.

Di seguito un esempio di configurazione che effettua l'escape del carattere “`\>`” sostituendolo con “`\\\>`”:

```
format.escape.="\\"
```

Infine è possibile cifrare le informazioni registrate su file di log utilizzando una proprietà definita tramite la seguente sintassi:

- “*format.encryptedProperty.<posizione>.<nomeProprietà>=<valoreProprietàDaCifrare>*”

La modalità con cui verranno cifrati i valori della proprietà deve essere indicata tramite un'altra riga definita tramite la seguente sintassi:

```
"format.encrypt.<posizione>.<nomeProprietà>=<encryptionMode>"
```

La modalità “*<encryptionMode>*” indicata deve corrispondere ad una di quelle definite all'interno del file di configurazione stesso dei topic come descritto nella sezione [Cifratura delle Proprietà](#).

Rimane valida la considerazione che le proprietà (semplici o cifrate che siano) verranno risolte in ordine lessicografico rispetto alla posizione indicata, in modo da garantire la corretta risoluzione se si hanno proprietà che sono definite tramite altre proprietà.

Di seguito viene riportato l'esempio precedente modificato per cifrare la parte relativa agli indirizzi IP:

```
# encryption modes
encrypt.encTEST.mode=java
encrypt.encTEST.keystore.type=symm
encrypt.encTEST.key.path=/tmp/symmetric.secretkey
encrypt.encTEST.key.algorithm=AES
encrypt.encTEST.algorithm=AES/CBC/PKCS5Padding
encrypt.encTEST.encoding=base64
```

(continues on next page)

(continua dalla pagina precedente)

```

# properties
format.property.001.commonso.govway-id=govway
format.property.002.commonso.id="${log:transactionId}"|"${log:property(commonso.
↳govway-id)}"
format.property.003.commonso.data="${log:inRequestDateZ(yyyy-MM-dd HH:mm:ss:SSS,
↳UTC)}"|"${log:inRequestDate(Z)}"
format.encryptedProperty.004.commonso.remoteIP-protocol-method="${log:forwardedIP}"|"HTTP/1.1"|"${log:httpMethod}"
format.encrypt.004.commonso.remoteIP-protocol-method=encTEST
format.property.005.commonso=${log:property(commonso.id)}|${log:property(commonso.
↳data)}|${log:property(commonso.remoteIP-protocol-method)}

# topic
format.topic.erogazioni.inputRequest="req"|"${log:property(commonso)}"
format.topic.erogazioni.inputResponse="res"|"${log:property(commonso)}"|"$"
↳{log:outHttpStatus}"

```

Di seguito viene fornito un esempio di informazioni prodotte per il topic “inputRequest” senza cifrare i valori della proprietà “commonso.remoteIP-protocol-method”:

```

"req"|"b6cdd758-342c-4599-ae95-33a781730b3f"|"govway"|"2020-06-26 12:46:50:629
↳"|"+"0200"|"192.168.1.2"|"HTTP/1.1"|"POST"
"req"|"2a9dc253-9dd5-458b-8689-edee7c9ba139"|"govway"|"2020-06-26 12:47:50:561
↳"|"+"0200"|"192.168.1.2"|"HTTP/1.1"|"POST"

```

Introducendo la cifratura si avrà un log simile al seguente:

```

"req"|"b6cdd758-342c-4599-ae95-33a781730b3f"|"govway"|"2020-06-26 12:46:50:629
↳"|"+"0200"|"01q0UhaXq70F2wAfh+xuA==.DTAZdcP3keHRN97tWRoPVmlcMG91aScUFU2/
↳r2T0wg0=
"req"|"2a9dc253-9dd5-458b-8689-edee7c9ba139"|"govway"|"2020-06-26 12:47:50:561
↳"|"+"0200"|"01q0UhaXq70F2wAfh+xuA==.DTAZdcP3keHRN97tWRoPVmlcMG91aScUFU2/
↳r2T0wg0=

```

Cifratura delle Proprietà

Come descritto nella sezione *Informazioni raggruppate in Proprietà* è possibile cifrare le informazioni di una proprietà tramite la sintassi:

- “format.encryptedProperty.<posizione>.<nomeProprietà>=<valoreProprietàDaCifrare>”

La modalità con cui verranno cifrati i valori della proprietà deve essere indicata tramite un’altra riga definita tramite la seguente sintassi:

“format.encrypt.<posizione>.<nomeProprietà>=<encryptionMode>”

La modalità “<encryptionMode>” deve corrispondere ad una tra quelle definite all’interno del file di configurazione dei topic. Di seguito un esempio di definizione di due modalità:

```

# encryption modes
encrypt.encMode1.mode=java
encrypt.encMode1.keystore.type=symmm
encrypt.encMode1.key.path=/tmp/symmetric.secretkey

```

(continues on next page)

(continua dalla pagina precedente)

```

encrypt.encMode1.key.algorithm=AES
encrypt.encMode1.algorithm=AES/CBC/PKCS5Padding
encrypt.encMode1.encoding=base64

encrypt.encMode2.mode=jose
encrypt.encMode2.keystore.type=public
encrypt.encMode2.key.path=/tmp/publicKey.pem
encrypt.encMode2.key.id=myKEY
encrypt.encMode2.key.algorithm=RSA-OAEP-256
encrypt.encMode2.algorithm=A256GCM
encrypt.encMode2.include.cert=false
encrypt.encMode2.include.public.key=true
encrypt.encMode2.include.key.id=true
encrypt.encMode2.include.cert.sha1=false
encrypt.encMode2.include.cert.sha256=false

```

Ogni modalità viene descritta da un insieme di direttive definite tramite la sintassi:

- “*encrypt.<encryptionModeName>.<direttiva>*”

Di seguito vengono fornite tutte le direttive supportate:

- *mode* [required]: indica il tipo di token cifrato prodotto:
 - *jose*: viene prodotto un token JSON Web Encryption (JWE) conforme al RFC 7516;
 - *java*: viene utilizzata la classe Cipher fornita dal package javax.crypto per cifrare i dati che poi verranno serializzati su file di log a seconda della direttiva “*encoding*” fornita;
 - *openssl*: viene prodotto un cipher text, attraverso una chiave derivata da una password, che può essere decifrato utilizzando i comandi di encryption “*openssl*”; richiede un keystore di tipo “*pass*”.
- *encoding* [required; mode=java|openssl]: indica il tipo di codifica utilizzato per la rappresentazione dei dati cifrati:
 - *base64*: rappresentazione base64;
 - *hex*: rappresentazione esadecimale;
- *keystore.type* [required]: indica il tipo di keystore utilizzato dove è presente la chiave di cifratura da utilizzare:
 - *symm*: indica l’utilizzo di una chiave simmetrica fornita attraverso la direttiva:
 - * *key.inline*: chiave simmetrica (es. Chiave AES dovrà essere di 16, 24 o 32 byte);
 - * *key.path*: [ignorata se presente “*key.inline*”] path su filesystem ad una chiave simmetrica (es. Chiave AES dovrà essere di 16, 24 o 32 byte);
 - * *key.encoding*: [optional; base64/hex] consente di indicare la codifica della chiave;
 - *pass*: indica la generazione di una chiave derivata da una password attraverso le seguenti direttive (non utilizzabile con la modalità “*jose*”):
 - * *password*: la password utilizzata per derivare la chiave;
 - * *password.type*: [opzionale; default=openssl-pbkdf2-aes-256-cbc] consente di selezionare l’algoritmo di derivazione tra le seguenti opzioni disponibili:
 - openssl-aes-256-cbc
 - openssl-pbkdf2-aes-128-cbc
 - openssl-pbkdf2-aes-192-cbc

- openssl-pbkdf2-aes-256-cbc
- * *password.iter*: [optional] consente di indicare il numero di iterazioni durante la derivazione della chiave con l'algoritmo pbkdf2.
- *jceks*: indica l'utilizzo di una chiave simmetrica presente in un keystore java di tipo JCEKS indirizzato tramite le seguenti direttive:
 - * *keystore.path*: path su filesystem del keystore;
 - * *keystore.password*: password del keystore;
 - * *key.alias*: alias che identifica la chiave simmetrica nel keystore;
 - * *key.password*: password della chiave simmetrica;
- *public*: indica l'utilizzo di una chiave pubblica asimmetrica fornita attraverso le seguenti direttive:
 - * *key.inline*: chiave pubblica asimmetrica in formato PEM o DER (sono supportati sia i formati pkcs1 che pkcs8);
 - * *key.path*: [ignorata se presente "key.inline"] path su filesystem ad una chiave pubblica asimmetrica in formato PEM o DER (sono supportati sia i formati pkcs1 che pkcs8);
 - * *key.encoding*: [optional; base64/hex] consente di indicare la codifica della chiave;
 - * *key.wrap* [optional; mode=java; boolean true/false]: indicazione se la chiave pubblica debba essere utilizzata per cifrare direttamente i dati (key.wrap=false) o per cifrare una chiave simmetrica AES generata dinamicamente (key.wrap=true);

Nota: La modalità "key.wrap=false" è utilizzabile solamente con informazioni da cifrare «sufficientemente corte» rispetto alla capacità di cifratura della chiave RSA altrimenti si avrà un errore simile al seguente: «too much data for RSA block».

- *jwk*: indica l'utilizzo di keystore JWK che può contenere una chiave simmetrica o una chiave pubblica asimmetrica; le direttive supportate sono le seguenti:
 - * *keystore.path*: path su filesystem del keystore;
 - * *key.alias*: alias che identifica la chiave nel keystore;
 - * *key.wrap* [optional; mode=java; boolean true/false]: indicazione se la chiave pubblica debba essere utilizzata per cifrare direttamente i dati (key.wrap=false) o per cifrare una chiave simmetrica AES generata dinamicamente (key.wrap=true);
- *jks o pkcs12*: indica l'utilizzo di un certificato presente in un keystore java di tipo JKS o PKCS12 indirizzato tramite le seguenti direttive:
 - * *keystore.path*: path su filesystem del keystore;
 - * *keystore.password*: password del keystore;
 - * *key.alias*: alias che identifica il certificato nel keystore;
 - * *key.wrap* [optional; mode=java; boolean true/false]: indicazione se il certificato debba essere utilizzato per cifrare direttamente i dati (key.wrap=false) o per cifrare una chiave simmetrica AES generata dinamicamente (key.wrap=true);
- *<tipoRegistratoPKCS11>*: indica l'utilizzo di uno dei tipi di keystore PKCS11 registrati ("Device PKCS11") all'interno del quale è presente il certificato da utilizzare indicato tramite la direttiva:
 - * *key.alias*: alias che identifica il certificato nel keystore;

- * *key.wrap* [optional; mode=java; boolean true/false]: indicazione se il certificato debba essere utilizzato per cifrare direttamente i dati (key.wrap=false) o per cifrare una chiave simmetrica AES generata dinamicamente (key.wrap=true);
- *key.algorithm* [required]: specifica l'algoritmo utilizzato per generare o gestire le chiavi crittografiche utilizzate durante il processo di cifratura;
- *algorithm* [required]: specifica l'algoritmo utilizzato per cifrare effettivamente i dati;
- *include.key.id* [optional; mode=jose; boolean true/false]: indicazione se inserire nell'header del token JWE (claim "kid") l'alias della chiave utilizzata per la cifratura;
- *key.id* [optional; mode=jose]: indica il nome della chiave che verrà inserito nel claim "kid" presente nell'header del token JWE;
- *include.cert* [optional; mode=jose; boolean true/false]: indicazione se inserire nell'header del token JWE (claim "x5c") il certificato utilizzato per la cifratura;
- *include.cert.sha1* [optional; mode=jose; boolean true/false]: indicazione se inserire nell'header del token JWE (claim "x5t") il digest SHA-1 del certificato utilizzato per la cifratura;
- *include.cert.sha256* [optional; mode=jose; boolean true/false]: indicazione se inserire nell'header del token JWE (claim "x5t#256") il digest SHA-256 del certificato utilizzato per la cifratura;
- *include.public.key* [optional; mode=jose; boolean true/false]: indicazione se inserire nell'header del token JWE (claim "jwk") la chiave pubblica utilizzata per la cifratura.
- *ksm* [optional]: consente di riferire un KSM di "unwrap", tramite il proprio identificativo, definito nel file <directory-lavoro>/byok.properties; maggiori dettagli vengono forniti nella sezione byokInstallKsm;
- *ksm.param.<paramName>* [optional]: come descritto nella sezione byokInstallKsmParametri ogni KSM può richiedere dei parametri di input che possono essere forniti tramite la direttiva "ksm.param.<paramName>".

Rappresentazione dei dati cifrati con mode=java

Come descritto in precedenza indicando la modalità "java" nella direttiva "mode" viene utilizzata la classe Cipher fornita dal package javax.crypto per cifrare i dati che poi verranno serializzati su file di log a seconda della direttiva "encoding" fornita: base64 o hex.

In funzione del tipo di chiave (simmetrica o asimmetrica) e della direttiva key.wrap la rappresentazione dei dati cifrati conterrà più parti che devono essere considerate per poter effettuare l'operazione inversa di decifratura:

- *chiave simmetrica*: il dato cifrato è formato da due parti, separate tramite un punto, entrambe codificate in base64 o hex a seconda dell'encoding selezionato; la prima parte rappresenta il Vettore di Inizializzazione (IV) mentre la seconda sono i dati cifrati:
 - <IV>.<DatiCifrati>
- *chiave pubblica asimmetrica con direttiva key.wrap=true*: il dato cifrato è formato da tre parti, separate tramite un punto, entrambe codificate in base64 o hex a seconda dell'encoding selezionato; la prima parte rappresenta la chiave AES generata dinamicamente e cifrata con la chiave pubblica (wrap), la seconda parte il Vettore di Inizializzazione (IV) della cifratura simmetrica e la terza parte sono i dati cifrati con la chiave simmetrica:
 - <WRAP_KEY>.<IV>.<DatiCifrati>
- *chiave pubblica asimmetrica con direttiva key.wrap=false*: è presente solo una parte contenente i dati cifrati con la chiave pubblica asimmetrica:
 - <DatiCifrati>

8.6 Registrazione Messaggi

Accedendo la sezione *Configurazione > Registrazione Messaggi* si può abilitare il salvataggio dei contenuti dei messaggi della richiesta e della risposta transiti su GovWay.

In Fig. 8.57 viene mostrata la pagina di configurazione.

Registrazione Messaggi

Registrazione Messaggi

Stato: abilitato

[Configurazione Erogazioni](#)

[Configurazione Fruizioni](#)

Figure8.57: Configurazione della funzionalità di Registrazione Messaggi

La configurazione consente di abilitare e configurare la registrazione dei messaggi in transito sul gateway; una volta abilitata l'opzione si possono configurare i dettagli della funzionalità differenziandone il comportamento tra erogazioni e fruizioni rispettivamente tramite i link *Configurazione Erogazioni* e *Configurazione Fruizioni* (es. Fig. 8.58).

Registrazione Messaggi > Configurazione Erogazioni

Configurazione Erogazioni

Richiesta

Stato: abilitato

Ingresso

Headers: disabilitato

Payload: disabilitato

Uscita

Headers: disabilitato

Payload: disabilitato

Risposta

Stato: abilitato

Ingresso

Headers: disabilitato

Payload: disabilitato

Uscita

Headers: disabilitato

Payload: disabilitato

Figure8.58: Configurazione della funzionalità di Registrazione Messaggi per le erogazioni

Dalla sottosezione di configurazione è possibile definire un criterio di registrazione dei messaggi differenziando tra Richiesta e Risposta e abilitando/disabilitando solo la comunicazione desiderata tra:

- *Ingresso*: il messaggio di richiesta o risposta nel momento in cui giunge sul gateway e quindi prima che venga sottoposto al processo di elaborazione previsto.

- *Uscita*: il messaggio di richiesta o risposta nel momento in cui esce dal gateway, per raggiungere il nodo successivo del flusso, e quindi dopo che è stato sottoposto al processo di elaborazione previsto.

Per ciascuno dei messaggi, su cui è stata abilitata la registrazione, è possibile scegliere gli elementi da registrare tra:

- *Headers*: vengono salvati gli header di trasporto (HTTP Headers) associati al messaggio;
- *Payload*: viene salvato il corpo del messaggio (HTTP Payload).

Nota: Le configurazioni effettuate in questa sezione della console hanno valenza globale e quindi rappresentano il comportamento di default adottato dal gateway nella gestione dei diversi flussi di comunicazione. Tale comportamento può essere ridefinito puntualmente su ogni singola erogazione/fruizione agendo sulla voce di configurazione *Registrazione Messaggi*.

8.7 Rate Limiting

Questa sezione descrive come creare e attivare le policy di controllo del traffico:

- *Registro Policy*: Consente di accedere al Registro delle Policy per visualizzare, modificare e creare le policy di controllo istanziabili per la configurazione del rate limiting. Tra parentesi viene visualizzato il numero di policy attualmente presenti nel registro. Questa funzionalità è descritta nella sezione [Registro Policy](#).
- *Policy Globali*: Consente di accedere al Registro delle Policy Attivate in ambito globale, cioè operative sul traffico complessivo che transita sul gateway. A queste policy si aggiungono quelle eventualmente definite localmente nella configurazione specifica di ciascuna erogazione/fruizione. Questa funzionalità è descritta nella sezione [Policy Globali](#).

8.7.1 Registro Policy

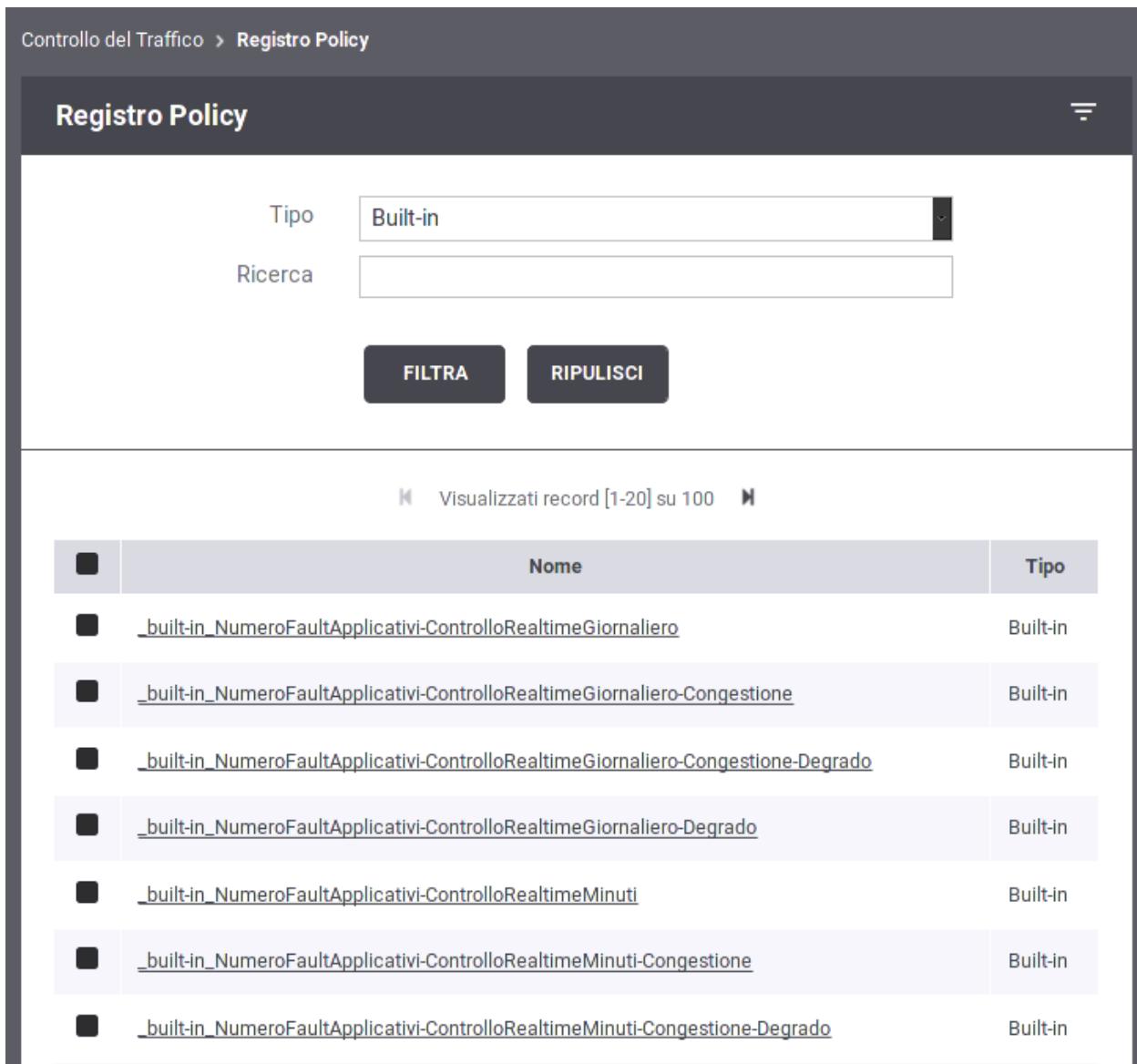
Il Registro delle Policy è il repository dove si possono creare le policy di rate limiting che potranno essere successivamente istanziate. L'accesso alla sezione è possibile grazie all'omonimo collegamento presente nella sezione *Rate Limiting* della pagina principale del controllo del traffico.

La pagina indice del Registro delle Policy mostra l'elenco delle policy già presenti (Fig. 8.59).

Tramite il pulsante “Aggiungi” è possibile aprire la pagina di creazione di una policy di Rate Limiting (Fig. 8.60).

Descriviamo nel seguito i dati che è necessario inserire per la creazione di una policy. Si tenga presente che il sistema propone valori di default per alcuni campi; tali valori cambiano in base alle scelte operate sugli altri campi e possono essere considerati come “consigliati” in base alla combinazione di scelte attuate.

- *Policy*: In questa sezione sono presenti i dati che identificano la policy.
 - *Nome*: Nome assegnato alla policy. Finché il campo non viene modificato dall'utente, viene proposto automaticamente un nome espressivo sulla base delle scelte operate sui rimanenti elementi del form.
 - *Descrizione*: Un testo di descrizione riferito alla policy. Finché il campo non viene modificato dall'utente, viene proposto un testo automatico di descrizione sulla base delle scelte operate sui rimanenti elementi del form.
 - *Metrica*: Si seleziona la metrica che la policy deve monitorare al fine di attuare le eventuali restrizioni. Sono disponibili le seguenti risorse:



The screenshot shows the 'Registro Policy' (Policy Registry) page within the 'Controllo del Traffico' (Traffic Control) section. The interface includes a search bar, a dropdown for 'Tipo' (Type) set to 'Built-in', and buttons for 'FILTRA' (Filter) and 'RIPULISCI' (Clear). Below the search area, a message indicates 'Visualizzati record [1-20] su 100' (Showing records [1-20] of 100). The main content is a table listing eight built-in rate limiting policies, each with a checkbox and a link to its configuration. The columns are 'Nome' (Name) and 'Tipo' (Type). The policies listed are:

	Nome	Tipo
<input type="checkbox"/>	_built-in_NumerofaultApplicativi-ControlloRealtimeGiornaliero	Built-in
<input type="checkbox"/>	_built-in_NumerofaultApplicativi-ControlloRealtimeGiornaliero-Congestione	Built-in
<input type="checkbox"/>	_built-in_NumerofaultApplicativi-ControlloRealtimeGiornaliero-Congestione-Degrado	Built-in
<input type="checkbox"/>	_built-in_NumerofaultApplicativi-ControlloRealtimeGiornaliero-Degrado	Built-in
<input type="checkbox"/>	_built-in_NumerofaultApplicativi-ControlloRealtimeMinuti	Built-in
<input type="checkbox"/>	_built-in_NumerofaultApplicativi-ControlloRealtimeMinuti-Congestione	Built-in
<input type="checkbox"/>	_built-in_NumerofaultApplicativi-ControlloRealtimeMinuti-Congestione-Degrado	Built-in

Figure8.59: Elenco delle Policy di Rate Limiting presenti nel registro

Controllo del Traffico > Registro Policy > Aggiungi

Note: (*) Campi obbligatori

Policy

Nome * NumeroRichieste-ControlloRealtimeOrario

Descrizione * La policy limita il numero totale massimo di richieste consentite durante l'intervallo di tempo specificato in ore (campionamento real-time, finestra corrente).

Metrica Numero Richieste

Valori di Soglia

Modalità di Controllo Realtime

Num. Massimo Richieste *

Intervallo Osservazione

Frequenza Orario

Ore *

Finestra Corrente

Applicabilità

Condizionale

SALVA

Figure8.60: Maschera per la creazione di una policy di Rate Limiting

- * *NumeroRichieste*: La policy effettua il controllo sul numero di richieste gestite. Selezionando questa risorsa si attiveranno i seguenti elementi per la configurazione dei valori di soglia:
 - *Modalità di Controllo*
 - *Numero Massimo di Richieste*
 - *Frequenza Intervallo Osservazione*
 - *Intervallo Osservazione*
 - *Finestra Osservazione*
- * *NumeroRichiesteSimultanee*: La policy effettua il controllo sul numero di richieste simultanee gestite. Selezionando questa metrica si attiveranno i seguenti elementi per la configurazione dei valori di soglia:
 - *Numero Massimo di Richieste*
- * *Dimensione Massima Messaggi*: La policy limita la dimensione massima, in KB, consentita per una richiesta e/o per una risposta. Selezionando questa metrica si attiveranno i seguenti elementi per la configurazione dei valori di soglia:
 - *Dimensione Richiesta*
 - *Dimensione Risposta*
- * *OccupazioneBanda*: La policy effettua il controllo sulla banda occupata da e verso le comunicazioni con il gateway. Selezionando questa risorsa si attiveranno i seguenti elementi per la configurazione dei valori di soglia:
 - *Modalità di Controllo*
 - *Tipo Banda*
 - *Occupazione Massima di Banda (kb)*
 - *Frequenza Intervallo Osservazione*
 - *Intervallo Osservazione*
 - *Finestra Osservazione*
- * *TempoComplessivoRisposta*: La policy controlla la quantità di tempo complessivamente impiegata dal gateway per la ricezione delle risposte dai servizi invocati. Selezionando questa metrica si attiveranno i seguenti elementi per la configurazione dei valori di soglia:
 - *Modalità di Controllo su Realtime (non modificabile)*
 - *Tipo Latenza*
 - *Occupazione Massima di Tempo (secondi)*
 - *Frequenza Intervallo Osservazione*
 - *Intervallo Osservazione*
 - *Finestra Osservazione*
- * *TempoMedioRisposta*: La policy controlla il tempo medio impiegato dal gateway per la ricezione delle risposte dai servizi invocati. Selezionando questa metrica si attiveranno i seguenti elementi per la configurazione dei valori di soglia:
 - *Modalità di Controllo*
 - *Tipo Latenza*
 - *Tempo Medio Risposta (ms)*

- *Frequenza Intervallo Osservazione*

- *Intervallo Osservazione*

- *Finestra Osservazione*

- * *NumeroRichiesteCompletateConSuccesso*

NumeroRichiesteFallite

NumeroFaultApplicativi

La policy effettua il controllo sul numero di richieste gestite dal gateway e terminate con un esito che rientra nella casistica associata alla risorsa selezionata (completate con successo, fallite o fault applicativi). Selezionando questa metrica si attiveranno i seguenti elementi per la configurazione dei valori di soglia:

- *Modalità di Controllo*

- *Numero Massimo di Richieste*

- *Frequenza Intervallo Osservazione*

- *Intervallo Osservazione*

- *Finestra Osservazione*

- *Valori di Soglia:* In questa sezione si specificano i valori di soglia (già anticipati al punto precedente), superati i quali, la policy risulta violata. Alcuni campi presenti in questa sezione cambiano in base alla risorsa monitorata.

- *Simultaneo:* Questa opzione è presente solo per la risorsa “NumeroRichieste”. Attivandola si specifica che il criterio restrittivo entra in funzione al superamento di una soglia sul numero di richieste simultaneamente in gestione.

- *Modalità di Controllo:* Rappresenta la modalità di raccolta dei dati di traffico che saranno usati per la valutazione della policy. Si può scegliere tra le seguenti opzioni:

- * *Realtime:* L'indicatore utilizzato per valutare la policy viene calcolato sulla base di dati raccolti in tempo reale durante l'elaborazione. Questa modalità assicura la massima accuratezza ma occorre tenere presenti le seguenti restrizioni nell'uso:

- 1. I dati “realtime” vengono raccolti in maniera separata sui singoli nodi del cluster. Quindi il controllo effettuato dalla policy riguarderà il traffico sul singolo nodo.

- 2. Si possono impostare criteri di controllo su grana temporale piccola: secondi, minuti, orario, giornaliero.

- * *Statistica:* L'indicatore utilizzato per valutare la policy viene calcolato sulla base delle informazioni statistiche presenti nel database di monitoraggio. L'accuratezza dei dati utilizzati per la valutazione è subordinata alla frequenza di aggiornamento dei dati statistici sul database. Inoltre tale modalità richiede il tracciamento delle transazioni sulle quali viene poi calcolata la statistica (vedi sezione *Tracciamento*). In questa modalità:

- 1. L'indicatore utilizzato per il confronto con la soglia della policy è sempre complessivo rispetto a tutti i nodi del cluster.

- 2. Si possono impostare criteri di controllo con grana temporale ampia: orario, giornaliero, settimanale, mensile.

- 3. Si può utilizzare la tipologia “finestra scorrevole” come valore per la “Finestra Osservazione”, che descriveremo poco più avanti.

- *Numero Massimo di Richieste:* Campo visibile solo per la metrica “NumeroRichieste”. Consente di specificare la soglia per la policy. Quando il numero delle richieste, conteggiate secondo la logica specificata nella policy, supera questo valore, la policy risulta violata.

- *Tipo Banda*: Campo visibile solo per la metrica “OccupazioneBanda”. Consente di specificare la modalità di calcolo della banda occupata per il confronto con la soglia impostata nella policy. Sono disponibili le seguenti opzioni:
 - * *Banda Interna*: Ai fini del conteggio dell’occupazione di banda (in KB) verrà considerato il solo traffico relativo alle comunicazioni con gli applicativi interni al dominio.
 - * *Banda Esterna*: Ai fini del conteggio dell’occupazione di banda (in KB) verrà considerato il solo traffico relativo alle comunicazioni con i servizi esterni al dominio.
 - * *Banda Complessiva*: Ai fini del conteggio dell’occupazione di banda (in KB) verrà considerato tutto il traffico in entrata ed uscita sul gateway.
- *Occupazione Massima di Banda (kb)*: Campo visibile solo per la metrica “OccupazioneBanda”. Consente di specificare la soglia per la policy. Quando la banda, calcolata secondo la logica specificata nella policy, supera questo valore, la policy risulta violata.
- *Tipo Latenza*: Campo visibile solo per le metriche “TempoComplessivoRisposta” e “TempoMedioRisposta”. Consente di specificare la logica di calcolo del tempo di risposta sulla base delle due seguenti opzioni:
 - * *Latenza Servizio*: Per il calcolo del tempo di risposta si considera unicamente il tempo di attesa del gateway dall’invio della richiesta alla ricezione della risposta.
 - * *Latenza Totale*: Per il calcolo del tempo di risposta si considera, oltre alla latenza del servizio, anche il tempo di elaborazione del gateway dal momento dell’ingresso della richiesta fino all’uscita della risposta.
- *Occupazione Massima di Tempo (secondi)*: Campo visibile solo per la metrica “TempoComplessivoRisposta”. Consente di specificare la soglia per la policy. Quando la latenza complessiva, calcolata secondo la logica specificata nella policy, supera questo valore, la policy risulta violata.
- *Tempo Medio Risposta (ms)*: Campo visibile solo per la metrica “TempoMedioRisposta”. Consente di specificare la soglia per la policy. Quando la latenza media, calcolata secondo la logica specificata nella policy, supera questo valore, la policy risulta violata.
- *Frequenza Intervallo Osservazione*

Intervallo Osservazione

Finestra Osservazione

La composizione di questi 3 campi specifica in quale intervallo temporale devono essere selezionati i dati da utilizzare per calcolare l’indicatore che deve essere confrontato con la soglia della policy.

I valori di “Frequenza Intervallo Osservazione” e “Intervallo Osservazione” specificano la frequenza di campionamento dei dati utilizzati per la valutazione delle soglie. In particolare il valore da specificare come Intervallo Osservazione è sempre un numero intero (ad esempio inserendo 8 si campioneranno i dati su finestre di 8 secondi, 8 minuti, ecc, in base all’unità di misura indicata per la frequenza). Il valore selezionato come “Finestra” individua l’esatto intervallo utilizzato nella catena temporale ogni volta che si valuta la policy per una specifica richiesta di servizio.

Per comprendere la logica con cui viene calcolata la finestra di osservazione è necessario introdurre il concetto di Data Attivazione Policy. Si tratta della data in cui la policy è stata applicata ad una richiesta in transito sul gateway. A partire da questa data vengono calcolate le finestre di osservazione in base alla frequenza di campionamento selezionata.

In Fig. 8.61 è mostrato un confronto tra le diverse finestre di osservazione su un campionamento di 2 ore. La determinazione della finestra può essere analogamente trasposta su altre frequenze di campionamento.

Riepilogando:

- * *Corrente*: Indica che per il calcolo dell'indicatore saranno utilizzati i dati che rientrano nella finestra temporale in cui ricade la richiesta in esame.
- * *Precedente*: Indica che per il calcolo dell'indicatore saranno utilizzati i dati che rientrano nella finestra temporale precedente a quella in cui ricade la richiesta in esame.
- * *Scorrevole (disponibile solo nella Modalità Controllo "Statistica")*: Indica che per il calcolo dell'indicatore saranno utilizzati i dati che rientrano in una finestra dinamica che ha come estremo superiore l'ora piena subito precedente all'istante della richiesta in fase di valutazione.

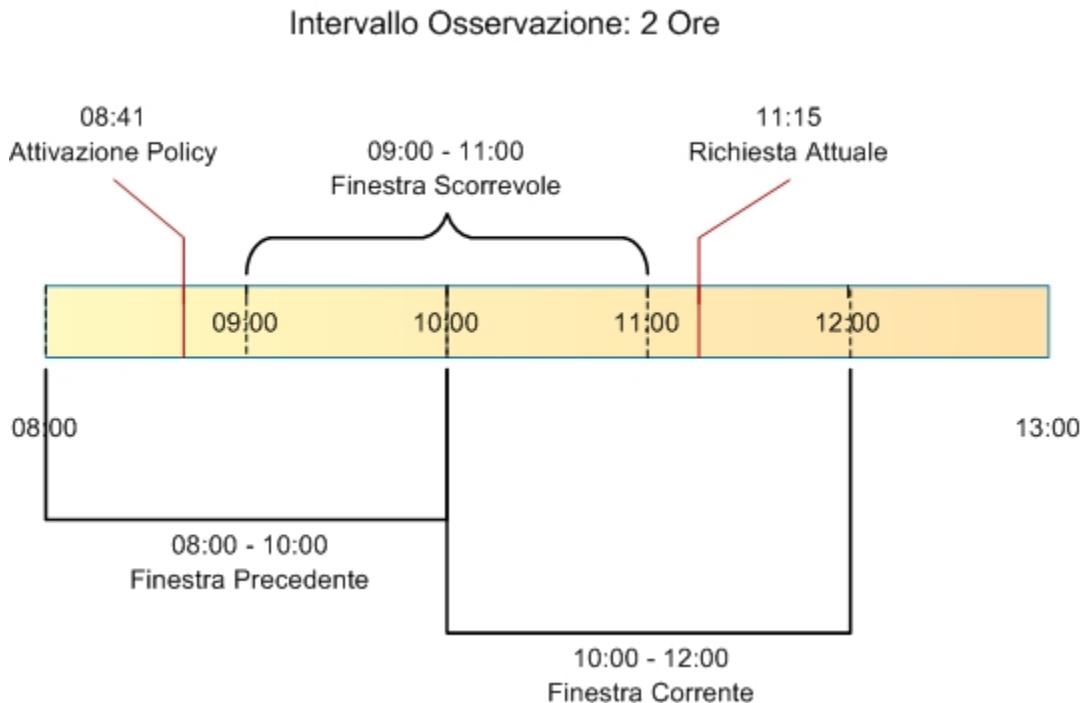


Figure8.61: Finestre di osservazione su un campionamento di 2 ore

- *Applicabilità*: Questa sezione della policy consente di restringere l'applicabilità della policy sulla base di alcuni criteri (Fig. 8.62). Sono presenti i seguenti campi:
 - *Condizionale*: Se questa opzione non è attiva, la policy si applica in maniera incondizionata. Attivando l'opzione, la policy risulterà applicabile sulla base dei criteri specificati nei campi successivi.
 - *In presenza di Congestione del Traffico*: Attivando questa opzione la policy risulta applicabile solo quando sussiste lo stato di congestimento. Affinché questo evento venga rilevato è necessario che sia abilitato il “Controllo della Congestione”, descritto in precedenza, e che risulti superata la soglia impostata sul numero di richieste simultanee.
 - *In presenza di Degrado Prestazionale*: Attivando questa opzione, la policy risulta applicabile solo in caso si rilevi un degrado prestazionale sullo specifico servizio corrispondente alla richiesta in gestione sul gateway. Per la rilevazione del degrado prestazionale si utilizzano le soglie “Tempo Medio di Risposta” impostate sia per le fruizioni che per le erogazioni. Come descritto in precedenza, tali soglie vengono definite per default nella sezione “Configurazione > Controllo del Traffico”, ma possono essere ridefinite al livello del singolo connettore. Per il calcolo del tempo medio di risposta del servizio, da confrontare con la soglia impostata, si utilizza il criterio definito con i campi seguenti:
 - * *Modalità di Controllo*
 - * *Tempo Medio Risposta*

- * *Frequenza Intervallo Osservazione*
- * *Intervallo Osservazione*
- * *Finestra Osservazione*

Per tutti questi campi valgono le medesime descrizioni già riportate nella sezione precedente “Valori di Soglia”.

Applicabilità

Condizionale

Applicata solo in presenza di Congestione del Traffico i

Applicata solo in presenza di Degrado Prestazionale i

Degrado Prestazionale

Modalità di Controllo	<input type="text" value="Realtime"/>
Tempo Medio Risposta	<input type="text" value="Latenza Servizio"/>
Intervallo Osservazione	
Frequenza	<input type="text" value="Orario"/>
Ore *	<input type="text"/>
Finestra	<input type="text" value="Precedente"/>

Figure8.62: Opzioni per l'applicabilità di una policy di rate limiting

Nota: Se si selezionano più opzioni di applicabilità queste si considerano connesse secondo l'operatore logico AND.

8.7.2 Policy Globali

Questa sezione consente di definire le policy di rate limiting che hanno un raggio d'azione che supera la singola erogazione/fruizione ed effettua quindi valutazioni su un campo più ampio.

L'attivazione di una Policy Globale segue in prevalenza il medesimo criterio già descritto nella sezione *Registrazione di una policy* riguardo il caso della configurazione di una singola erogazione/fruizione. Vi sono però alcune differenze che riguardano i criteri di raggruppamento, per il calcolo dei valori di soglia, e i criteri di filtro per l'applicabilità della policy.

Raggruppamento

Come descritto nella sezione [Registrazione di una policy](#) è possibile definire dei criteri di raggruppamento che consentono di verificare i valori di soglia. La logica è del tutto analoga a quella dell'operatore GROUP BY del linguaggio SQL.

I criteri di raggruppamento, per una policy a livello globale, sono maggiori rispetto a quelli descritti in precedenza (Fig. 8.63):

- *Fruizione/Erogazione*
- *Soggetto Erogatore*
- *API*
- *Azione/Risorsa*
- *Soggetto Fruitore*
- *Applicativo Fruitore*
- *Token*
- *Raggruppamento per Chiave*

Valori di Soglia

Ridefinisci Valori di Soglia

Num. Massimo Richieste 100

Raggruppamento

Se abilitato, il calcolo del valore della soglia avviene raggruppando le richieste in [funzione dei criteri selezionati](#)

Stato	abilitato
Fruizione / Erogazione	<input type="checkbox"/>
Soggetto Erogatore	<input type="checkbox"/>
API	<input type="checkbox"/>
Soggetto Fruitore	<input type="checkbox"/>
Applicativo Fruitore	<input type="checkbox"/>
Token	<input type="checkbox"/>
Chiave	<input type="checkbox"/>

Figure 8.63: Definizione criteri di raggruppamento per la policy di rate limiting

Filtro

Abilitando questa sezione è possibile indicare i criteri affinché la policy sia applicabile in base alle caratteristiche di ciascuna richiesta in ingresso. In assenza di filtro, la policy sarà valutata su tutte le richieste in ingresso che riguardano l'erogazione/fruizione che si sta configurando. I criteri di filtro, per una policy a livello globale, sono maggiori rispetto a quelli descritti in precedenza nella sezione [Registrazione di una policy](#) (Fig. 8.64):

- *Stato*: Opzione per abilitare/disabilitare il filtro.
- *Ruolo Gateway*: Opzione per filtrare le richieste di servizio in base al ruolo ricoperto dal gateway nella specifica richiesta: Fruitore o Erogatore.
- *Profilo*: Opzione per filtrare le richieste di servizio in base al profilo di utilizzo del Gateway. Nel caso si sia selezionata un singolo profilo (o se il gateway ne supporta uno solo) viene visualizzato il valore attuale in modo non modificabile.
- *Ruolo Erogatore*: Opzione per filtrare le richieste di servizio in base al ruolo posseduto dal soggetto erogatore. Tramite la lista è possibile selezionare uno tra i ruoli censiti nel registro. La selezione di un ruolo esclude la possibilità di selezionare un soggetto erogatore.
- *Soggetto Erogatore*: Opzione per filtrare le richieste di servizio in base al soggetto erogatore. Tramite la lista è possibile selezionare uno tra i soggetti censiti nel registro. La selezione di un soggetto esclude la possibilità di selezionare un ruolo erogatore.
- *API*: Opzione per filtrare le richieste in base alla API invocata. Tramite la lista è possibile selezionare una tra le API censite nel registro. Se è stato selezionato un soggetto erogatore, saranno elencati solo le API da esso erogate. Analogamente, se è stato selezionato un profilo, saranno elencate solo API relative a tale profilo.
- *Azione/Risorsa*: Opzione per filtrare le richieste di servizio in base all'azione/risorsa invocata. Tramite la lista è possibile selezionare una tra le azioni/risorse censite nel registro. Se è stato selezionato una API, saranno elencati solo le azioni ad essa appartenenti.
- *Ruolo Fruitore*: Opzione per filtrare le richieste di servizio in base al ruolo posseduto dal soggetto fruitore. Tramite la lista è possibile selezionare uno tra i ruoli censiti nel registro. La selezione di un ruolo esclude la possibilità di selezionare un soggetto fruitore.
- *Soggetto Fruitore*: Opzione per filtrare le richieste di servizio in base al soggetto fruitore. Tramite la lista è possibile selezionare uno tra i soggetti censiti nel registro. Se è stato selezionato un servizio, saranno elencati solo i soggetti fruitori del medesimo. La selezione di un soggetto esclude la possibilità di selezionare un ruolo fruitore.
- *Applicativo Fruitore*: Opzione per filtrare le richieste di servizio in base all'applicativo fruitore (opzione non disponibile nel caso di una erogazione). Tramite la lista è possibile selezionare uno tra i servizi applicativi censiti nel registro. Se sono stati selezionati servizi e/o soggetti, la lista presentata sarà filtrata di conseguenza.
- *Filtro per Chiave*: Si tratta di un'opzione avanzata che consente di filtrare le richieste in ingresso sul gateway in base ad una chiave che può essere specificata in maniera personalizzata. Questa parte è già stata descritta in maniera approfondita nella sezione [Registrazione di una policy](#).

Nota: È possibile specificare più di un criterio di filtro; la logica applicata sarà quella dell'operatore AND.

Filtro

Stato	abilitato
Tipologia	Qualsiasi
Profilo	API Gateway
Ruolo Erogatore	Qualsiasi
Soggetto Erogatore	Qualsiasi
API	Qualsiasi
Ruolo Richiedente	Qualsiasi
Soggetto Fruitore	Qualsiasi
Chiave	<input type="checkbox"/>

Figure8.64: Definizione del filtro per l'istanza della policy di rate limiting

8.7.3 Visualizzazione Statistiche Policy

Quando una policy è attivata si ha la possibilità di accedere ad una finestra che fornisce una sintesi dei dati statistici legati all'applicazione della policy in fase di controllo del traffico.

Per visualizzare questa finestra è sufficiente accedere all'elenco delle policy attivate ed utilizzare il collegamento “Visualizza” nella colonna “Runtime” (Fig. 8.65).

Si noti che saranno visualizzati dei dati solo dopo la data di attivazione della policy e cioè dopo che è transitata la prima richiesta cui viene applicata la policy.

I dati statistici riportati sono i seguenti:

- *Criterio di Collezionamento dei dati*: I criteri di raggruppamento utilizzati dalla policy.
- *Dati Generali*:
 - *Il numero istantaneo delle richieste attive*
 - *la data di attivazione della policy (che corrisponde alla data di primo utilizzo della medesima)*
- *Dati collezionati per la risorsa <nomeRisorsa>*: dati di sintesi sulle transazioni cui è stata applicata la policy.

Sono inoltre disponibili i seguenti collegamenti:

- *Refresh*: per aggiornare i dati visualizzati.
- *Reset Contatori*: per azzerare i valori visualizzati (solo nella modalità di controllo realtime).

The screenshot shows a web-based management interface for the OpenSPCoop Enterprise platform. The top navigation bar indicates the path: Configurazione > Controllo del Traffico - Policy > OccupazioneBanda-ControlloRealtimeOrario. The main content area is divided into two sections: 'Informazioni Runtime' and 'PdD OpenSPCoop Enterprise'. The 'Informazioni Runtime' section contains a 'Refresh' button. The 'PdD OpenSPCoop Enterprise' section is titled 'Reset Contatori' and displays runtime statistics for a specific policy. The statistics are as follows:

```
=====
Criterio di Collezione dei Dati
  Disabilitato
Dati Generali
  Richieste Attive: 0
  Data Attivazione Policy: 2017-08-09_15:26:26.223
Dati collezionati per la risorsa 'OccupazioneBanda'
  Modalità di Controllo: realtime
  Finestra Osservazione: corrente
  Intervallo [2017-08-09_15:00:00.000 - 2017-08-09_15:59:59.999]
  Numero Richieste Accettate: 2
  Contatore: 6 kb (6869 bytes)
  Valore Medio: 3 kb (3434 bytes)
  Numero Richieste Bloccate: 0
=====
```

Figure8.65: Dati statistici relativi ad una policy di rate limiting

8.7.4 Filtro o Raggruppamento Personalizzato

Nella sezione *Registrazione di una policy* è possibile utilizzare dei criteri di raggruppamento per il valore di soglia o un filtro di applicabilità personalizzato in modo da definire un comportamento specifico per le proprie esigenze di servizio. Una configurazione personalizzata richiede la realizzazione di un plugin che contiene la logica di filtro e/o il raggruppamento personalizzato; il plugin consiste nell'implementazione di una classe java che implementa l'interfaccia:

```
package org.openspcoop2.pdd.core.controllo_traffico.plugins;
public interface IRateLimiting {
    public String estraiValoreFiltro(Logger log,Dati datiRichiesta) throws
    ↵PluginsException;
    public String estraiValoreCollezionamentoDati(Logger log,Dati
    ↵datiRichiesta) throws PluginsException;
}
```

La classe realizzata deve essere successivamente registrata tramite una entry da aggiungere all'interno del file (da creare se non esiste) */etc/govway/govway_local.classRegistry.properties* di GovWay:

```
org.openspcoop2.pdd.controlloTrafico.rateLimiting.<tipo>=<fully qualified
↪class name>
```

La stringa *<tipo>* diventa utilizzabile come “Tipo Personalizzato” da indicare in fase di configurazione per un criterio di filtro personalizzato (Fig. 8.66) e/o per un criterio di raggruppamento personalizzato (Fig. 8.67).

Filtro per Chiave

Stato	<input checked="" type="checkbox"/>
Tipologia	PluginBased
Tipo Personalizzato *	<input type="text"/>
Valore *	<input type="text"/>

Figure8.66: Filtro Personalizzato

Raggruppamento per Chiave

Stato	<input checked="" type="checkbox"/>
Tipologia	PluginBased
Tipo Personalizzato *	<input type="text"/>

Figure8.67: Raggruppamento Personalizzato

8.8 Token Policy

Per poter definire politiche di controllo degli accessi basate sui Bearer Token o per poterne spedire uno verso l'endpoint associato ad un connettore è necessario creare delle Token Policy da riferire nelle configurazioni degli specifici servizi. La gestione delle Token Policy si effettua andando alla sezione *Configurazione > Token Policy* della govwayConsole. Per creare una nuova policy si utilizza il pulsante *Aggiungi*. Il form di creazione appare inizialmente come quello illustrato in Fig. 8.79.

Figure 8.68: Creazione di una Token Policy

Inizialmente si inseriscono i dati identificativi:

- *Nome*: nome univoco da assegnare alla policy
- *Tipo*: determina il tipo di policy:
 - *Validazione*: definisce una policy utilizzabile per validare Bearer Token nel Controllo degli Accessi (*Autenticazione Token*)
 - *Negoziazione*: definisce i criteri per la negoziazione di un Bearer Token poi utilizzato sui connettori nei quali sarà associata la policy (*Autenticazione Token*)
- *Descrizione*: testo di descrizione generale della policy

I parametri richiesti differiscono a seconda del tipo selezionato. Le sezioni successive dettagliano i due tipi supportati.

8.8.1 Token Policy Negoziazione

Per poter definire politiche che consentono di spedire un Bearer Token verso l'endpoint associato ad un connettore è necessario creare delle Token Policy da riferire nelle configurazioni degli specifici servizi. La gestione delle Token Policy si effettua andando alla sezione *Configurazione > Token Policy* della govwayConsole. Per creare una nuova policy si utilizza il pulsante *Aggiungi*. Il form di creazione appare inizialmente come quello illustrato in [Fig. 8.79](#).

Inizialmente si inseriscono i dati identificativi:

- *Nome*: nome univoco da assegnare alla policy
- *Tipo*: deve essere selezionato il tipo *Negoziazione*
- *Descrizione*: testo di descrizione generale della policy

Al passo successivo si inseriscono le Informazioni Generali. Nella sezione *Token Endpoint* si specifica il tipo di negoziazione e i vari parametri necessari:

- *Tipo*: indica la modalità di negoziazione del token. I valori possibili sono:
 - *Client Credentials*: modalità di negoziazione “Client Credentials Grant” descritta nel RFC 6749 (<https://tools.ietf.org/html/rfc6749#page-40>);
 - *Resource Owner Password Credentials*: modalità di negoziazione “Resource Owner Password Credentials Grant” descritta nel RFC 6749 (<https://tools.ietf.org/html/rfc6749#page-37>);
 - *Signed JWT*: modalità di negoziazione “Client Credentials Grant” descritta nella sezione 2.2 del RFC 7523 (<https://datatracker.ietf.org/doc/html/rfc7523#section-2.2>) che prevede lo scambio di un'asserzione JWT firmata tramite certificato x.509 con l'authorization server;
 - *Signed JWT with Client Secret*: modalità di negoziazione identica alla precedente dove però l'asserzione JWT viene firmata tramite una chiave simmetrica;
 - *Personalizzato*: consente di definire in ogni sua parte la richiesta http inoltrata all'endpoint di negoziazione token.
- *URL*: endpoint del servizio di negoziazione token.
- *Connection Timeout*: Tempo massimo in millisecondi di attesa per stabilire una connessione con il server di negoziazione token.
- *Read Timeout*: Tempo massimo in millisecondi di attesa per la ricezione di una risposta dal server di negoziazione token.
- *Https*: Parametri di configurazione nel caso in cui il server di negoziazione token richieda un accesso Https.
- *Proxy*: Parametri di configurazione nel caso si richieda l'uso di un proxy per l'accesso.

Nel caso sia attivato il flag relativo ad un Proxy o una configurazione Https saranno presentate delle sezioni omonime dove poter inserire i dati di configurazione richiesti.

I parametri di configurazione relativi al tipo di negoziazione del token configurato vengono descritti nelle sezioni “*Client Credentials / Resource Owner Password Credentials*”, “*Signed JWT*” e “*Personalizzazione richiesta http di negoziazione*”.

Nella sezione “Dati Richiesta” potranno invece essere definiti ulteriori criteri che riguardano la richiesta di un token. I dati presenti possono essere definiti tramite costanti o possono contenere parti dinamiche risolte a runtime dal Gateway (per maggiori dettagli [Valori dinamici](#)).

- *Scope*: elenco di scope utente richiesti;
- *Audience*: audience per il quale si vorrebbe ottenere il token;

Token Policy

Tipo *	<input type="text" value="Negoziazione"/>
Nome *	<input type="text"/>
Descrizione	<input type="text"/>

Token Endpoint

Tipo	<input type="text" value="Client Credentials"/>
URL *	<input type="text" value="http://"/> (i)
Connection Timeout *	<input type="text" value="5000"/>
Read Timeout *	<input type="text" value="10000"/>
Https	<input type="checkbox"/>
Proxy	<input type="checkbox"/>

Autenticazione Client

Basic	<input type="checkbox"/>
Bearer	<input type="checkbox"/>
Https	<input type="checkbox"/>

Dati Richiesta

Scope	<input type="text"/> (i)
Elencare più scope separandoli con la virgola	
Audience	<input type="text"/> (i)
Parametri	<input type="text"/> (i)
Indicare per riga gli ulteriori parametri (nome=valore)	
Header HTTP	<input type="text"/> (i)
Indicare per riga gli eventuali header (nome=valore)	

Token Forward

Modalità	<input type="text" value="RFC 6750 - Bearer Token Usage (Authorization Reques: ✓)"/>
----------	--

- *Parametri*: consente di indicare per riga ulteriori parametri (nome=valore) da inserire nella richiesta.
- *Header HTTP*: consente di indicare per riga eventuali header HTTP (nome=valore) da inserire nella richiesta.

Infine nella sezione “Token Forward” si può configurare la modalità di inoltro del token verso l’endpoint del connettore a cui verrà associata la policy che stiamo definendo:

- *RFC 6750 - Bearer Token Usage (Authorization Request Header Field)*: Il token viene inoltrato al destinatario utilizzando l’header Authorization presente nella richiesta HTTP.
- *RFC 6750 - Bearer Token Usage (URI Query Parameter)*: Il token viene inoltrato al destinatario tramite parametro access_token della Query String.
- *Header HTTP*: Il token viene inoltrato al destinatario utilizzando un header HTTP il cui nome deve essere specificato nel campo seguente.
- *Parametro URL*: Il token viene inoltrato al destinatario utilizzando un parametro della Query String il cui nome deve essere specificato nel campo seguente.

Nelle sezioni successive vengono forniti i dettagli relativi alle modalità di negoziazione di un token nel caso sia previsto un jwt firmato o meno.

Client Credentials / Resource Owner Password Credentials

In entrambe le modalità è necessario definire i parametri di configurazione richiesti dall’authorization server per autenticare GovWay come client autorizzato a negoziare il token. Le modalità supportate sono le seguenti:

- *Autenticazione Http Basic*: flag da attivare nel caso in cui il servizio di negoziazione richieda autenticazione di tipo HTTP-BASIC. In questo caso dovranno essere forniti Client-ID e Client-Secret nei campi successivi. È inoltre possibile indicare se la coppia di credenziali deve essere codificata nella richiesta “x-www-form-urlencoded” oppure deve essere inserita in un header HTTP “Authorization”.
- *Autenticazione Bearer*: flag da attivare nel caso in cui il servizio di negoziazione richieda autenticazione tramite un bearer token. Il token dovrà essere indicato nel campo successivo fornito.
- *Autenticazione Https*: flag da attivare nel caso in cui il servizio di negoziazione richieda autenticazione di tipo Https. In questo caso dovranno essere forniti tutti i dati di configurazione nei campi presenti nella sezione “https”.

Se il tipo di negoziazione selezionato è “Resource Owner Password Credentials”, si dovrà inoltre fornire i dati di configurazione specifici dell’autenticazione utente:

- *Username e Password*: Dovranno essere forniti Username e Password dell’utente per cui verrà effettuata la negoziazione del token.

Signed JWT

Nel caso di modalità di negoziazione basata su uno scambio di un JWT firmato con l’authorization server si dovranno fornire tutti i parametri che andranno a definire il JWT firmato.

Innanzitutto se la modalità prevede una firma tramite chiave asimmetrica devono essere forniti i parametri di accesso ad un keystore contenente la chiave privata da utilizzare per la firma tramite una dei seguenti tipi:

- “JKS” o “PKCS12”: viene richiesta l’indicazione del path assoluto del keystore nel campo *File*, la definizione della password per l’accesso al keystore nel campo *Password*, l’alias con cui è riferita la chiave privata nel keystore nel campo *Alias Chiave Privata* e la password della chiave privata nel campo *Password Chiave Privata*;
- “JWK Set”: deve essere definito il path su filesystem dove risiede l’archivio json nel formato “JWK Set” e l’identificativo “kid” (alias) con cui è riferita la chiave privata nel campo *Alias Chiave Privata*;

- “Key Pair”: deve essere definito il path su filesystem dove risiedono la chiave privata e pubblica in formato PEM o DER (sono supportati sia i formati pkcs1 che pkcs8) e la password della chiave privata se cifrata nel campo *Password Chiave Privata*;
- Tipi PKCS11: i tipi disponibili sono quelli corrispondenti ai tipi di keystore PKCS11 registrati (“*Device PKCS11*”).

Nella sezione “JWT Signature” si deve indicare l’algoritmo di firma e l’eventuale chiave segreta nel caso in cui sia prevista una firma tramite chiave simmetrica.

All’interno della sezione “JWT Header” si possono definire quali parametri dovranno essere presenti nella parte non firmata dell’asserzione JWT:

- *Key Id (kid)*: indicazione della chiave utilizzata per attuare la firma dell’asserzione JWT, in una delle seguenti modalità:
 - Alias Chiave Privata (solamente in caso di firma con chiave asimmetrica): nel claim “kid” viene impostato l’alias della chiave privata indicato nella precedente sezione di configurazione;
 - Client ID: viene utilizzato il medesimo valore associato al claim “client_id” inserito nel payload firmato del JWT;
 - Personalizzato: permette di indicare un valore qualsiasi anche formato da parti dinamiche risolte a runtime dal Gateway (per maggiori dettagli *Valori dinamici*).
- *X.509 Certificate* (solamente in caso di firma con chiave asimmetrica):
 - “x5c”: viene inserito il certificato utilizzato per firmare l’asserzione JWT;
 - “x5u”: viene indicata una url dove reperire il certificato di firma.
- *Digest X.509 Certificate* (solamente in caso di firma con chiave asimmetrica): consente di indicare il digest del certificato di firma nella modalità SHA1 (x5t) o SHA256 (x5t#S256);
- *Type*: valore inserito nel claim “typ”;
- *Content Type*: se abilitato, il claim “cty” verrà valorizzato con il content-type associato alla richiesta effettuata all’authorization server.

Nella sezione “JWT Payload” si devono definire i parametri inseriti nella parte firmata dell’asserzione JWT:

- *Client ID*: identificativo del client censito sull’AuthorizationServer che verrà indicato nel claim “client_id” dell’asserzione JWT;
- *Audience*: identifica l’authorization server come destinario dell’asserzione JWT (claim “aud”);
- *Issuer*: dominio del soggetto firmatario dell’asserzione; se non viene fornito un valore il claim “iss” verrà valorizzato con il nome del soggetto associato al dominio di gestione della richiesta;
- *Subject*: il claim “sub” verrà valorizzato con la medesima informazione inserita nel claim “client_id” se nel campo Subject non viene fornito alcun valore;
- *Time to Live*: indica la validità temporale, in secondi, a partire dalla data di creazione dell’asserzione;
- *Claims*: consente di inserire ulteriori claims nel payload JWT firmato, indicandoli per riga nel formato “nome=valore” (modalità descritte nella sezione *Aggiunta di Claims nei Token*).

Tutti i valori definiti nella sezione “JWT Payload” possono contenere parti dinamiche che verranno risolte a runtime dal Gateway (per maggiori dettagli *Valori dinamici*).

Inoltre se non si desidera generare un determinato claim è possibile utilizzare la keyword “\${undefined}” come valore del campo.

Signed JWT (PDND)

La modalità di negoziazione basata su uno scambio di un JWT firmato con l'authorization server, descritta nella sezione “*Signed JWT*”, è stata selezionata come modalità di negoziazione dei token sulla Piattaforma Digitale Nazionale Dati (PDND) descritta nella sezione “*ID_AUTH_REST_01 tramite la Piattaforma Digitale Nazionale Dati (PDND)*”.

Il protocollo di negoziazione, oltre ai parametri standard previsti dal rfc 7523 (<https://datatracker.ietf.org/doc/html/rfc7523#section-2.2>), prevede l'inserimento di alcuni parametri aggiuntivi all'interno del payload del JWT firmato:

- *purposeId*: rappresenta l'identificativo della finalità dell'accordo di adesione recuperabile dalla piattaforma PDND;
- *sessionInfo*: informazioni di sessione che non vengono gestite sulla piattaforma PDND ma consentono di essere inviate al momento della negoziazione per poi essere riportate all'interno dell'access token generato dalla PDND.

il protocollo della PDND prevede inoltre l'aggiunta dei seguenti parametri nella richiesta “x-www-form-urlencoded” :

- *client_id*: se non viene definito alcun valore per il parametro verrà utilizzato il medesimo valore associato al Client ID definito nel payload del JWT;
- *resource*: audience/url che identifica il servizio sulla PDND. Se non viene fornito alcun valore, il parametro non verrà inserito nella richiesta.

I parametri precedentemente descritti sono configurabili attivando la modalità PDND successivamente alla selezione del tipo “Signed JWT” come modalità di negoziazione (Fig. 8.70). Sono configurabili all'interno della sezione “JWT Payload” (Fig. 8.71) e della sezione “Dati Richiesta” (Fig. 8.72).

The screenshot shows a configuration interface for a 'Token Endpoint'. At the top, there is a label 'Token Endpoint'. Below it, a 'Tipo' dropdown menu is open, showing 'Signed JWT' as the selected option. To the right of the dropdown is a small dropdown arrow. Below the dropdown, there is a checkbox labeled 'PDND' which is checked, indicated by a blue checkmark.

Figure8.70: Modalità di negoziazione “Signed JWT” via PDND

Personalizzazione richiesta http di negoziazione

La modalità di negoziazione con tipo “Personalizzato” consente di definire la richiesta http inoltrata all'endpoint di negoziazione token e il parsing della risposta ottenuta.

Nella sezione “Dati Richiesta” vengono definiti i parametri della richiesta HTTP. I dati presenti possono essere definiti tramite costanti o possono contenere parti dinamiche risolte a runtime dal Gateway (per maggiori dettagli *Valori dinamici*).

- *Modo HTTP*: tipo di richiesta (es. GET, POST, ...);
- *Content-Type*: presente solamente per tipi di richieste che prevedono l'invio di un payload, ne definisce il tipo di messaggio inviato;
- *Header HTTP*: consente di indicare per riga eventuali header HTTP (nome=valore) da inserire nella richiesta;
- *Tipo Template Payload*: definisce la modalità con la quale viene successivamente fornito il payload http:
 - *Template*: il payload viene definito tramite un template che può contenere parti dinamiche risolte a runtime definite tramite una sintassi proprietaria di GovWay (per maggiori dettagli *Valori dinamici*);

JWT Payload

Client ID *	<input type="text"/>	(i)
Audience *	<input type="text"/>	(i)
Issuer	<input type="text"/>	(i)
Subject	<input type="text"/>	(i)
Time to Live (secondi) *	<input type="text" value="30000000"/>	
Indica la validità temporale, in secondi, a partire dalla data di creazione dell'asserzione		
Purpose ID *	<input type="text" value="2234546234323"/>	(i)
Informazioni Sessione	<pre>userId=\${header:X-UserId} postazioneChiamante=\${header:X-Postazione}</pre>	
Indicare per riga i claims (nome=valore) da aggiungere nell'oggetto 'sessionInfo'		

Figure8.71: Modalità di negoziazione “Signed JWT” via PDND: claim “purposeId” e “sessionInfo” nel payload JWT

Dati Richiesta

Scope	<input type="text"/>	(i)
Elencare più scope separandoli con la virgola		
Audience	<input type="text"/>	(i)
Client ID	<input type="text" value="9ee921cd-38e8-11ed-b877-024207cc61bd"/>	(i)
Resource	<input type="text" value="4rr921cd-48e9-22ed-b877-024207cc00aa"/>	(i)
Parametri	<input type="text"/>	
Indicare per riga gli ulteriori parametri (nome=valore)		

Figure8.72: Modalità di negoziazione “Signed JWT” via PDND: parametri “client_id” e “resource” nella richiesta “x-www-form-urlencoded”

- *Freemarker Template*: il payload viene definito utilizzando il template «Freemarker» (<https://freemarker.apache.org/>);
- *Velocity Template*: il payload viene definito utilizzando il template «Velocity» (<http://velocity.apache.org/>);
- *Payload HTTP*: payload http inviato nella richiesta.

Nelle figure Fig. 8.73 e Fig. 8.74 vengono rispettivamente mostrati un esempio di configurazione di una richiesta http definita tramite un metodo che non prevede l'invio di un payload e un altro esempio che lo prevede.

Dati Richiesta

Metodo HTTP ▼

Header HTTP i

Indicare per riga gli eventuali header (nome=valore)

Figure8.73: Dati Richiesta http “GET” in una negoziazione token personalizzata

Dati Richiesta

Metodo HTTP ▼

Content-Type * i

Header HTTP i

Indicare per riga gli eventuali header (nome=valore)

Tipo Template Payload ▼

Payload HTTP * i

Figure8.74: Dati Richiesta http “POST” in una negoziazione token personalizzata

Dal token endpoint si attende un formato della risposta che sia conforme al json definito nello standard RFC 6749. Il parser utilizzato per default dalla configurazione, mostrato nella figura Fig. 8.75, è quello che consente di processare i claims definiti nel RFC e riportati nella tabella seguente.

Table8.1: Mapping informazione-claim per il processamento della risposta ottenuta dall'endpoint token di negoziazione

Informazione	RFC 6749 - OAuth2
Token Type	token_type
Access Token	access_token
Refresh Token	refresh_token
Scope	scope
Expires in	expires_in
Expires on	expires_on (**)
Refresh expires in	refresh_expires_in (**)
Refresh expires on	refresh_expires_on (**)
Nota:	
*_in	i claim “*_in” devono indicare un tempo di vita in secondi
*_on	i claim “*_on” devono riportare la data di scadenza come numero di secondi a partire da “1970-01-01T00:00:00Z UTC”
**	claim aggiuntivi generati da implementazioni di authorization server e non direttamente definiti nel RFC 6749

Formato Risposta

Tipo	RFC 6749 - OAuth2
------	-------------------

Figure8.75: Formato Risposta di default: RFC 6749 OAuth

È possibile in alternativa definire un mapping puntuale tra il nome di un claim e l'informazione che GovWay cerca di estrarre dal token. Come mostrato nella figura Fig. 8.76 il mapping viene richiesto per ogni claim definito nella tabella descritta precedentemente.

In alternativa è possibile indicare di utilizzare direttamente il payload di risposta HTTP come access token. In questo scenario, mostrato nella figura Fig. 8.77, dovrà essere definito l'intervallo temporale di validità del token acquisito.

Infine è possibile indicare il nome di una classe che implementa una logica di parsing personalizzata. La classe deve implementare l'interfaccia «org.openscoop2.pdd.core.token.parser.INegoizzazioneTokenParser» (Fig. 8.78).

8.8.2 Token Policy Validazione

Per poter definire politiche di controllo degli accessi basate sui Bearer Token è necessario creare delle Token Policy da riferire nelle configurazioni degli specifici servizi. La gestione delle Token Policy si effettua andando alla sezione *Configurazione > Token Policy* della govwayConsole. Per creare una nuova policy si utilizza il pulsante *Aggiungi*. Il form di creazione appare inizialmente come quello illustrato in Fig. 8.79.

Inizialmente si inseriscono i dati identificativi:

- *Nome*: nome univoco da assegnare alla policy
- *Tipo*: deve essere selezionato il tipo *Validazione*
- *Descrizione*: testo di descrizione generale della policy

Formato Risposta

Tipo	Personalizzato
Token Type *	token_type
Access Token *	access_token
Refresh Token *	refresh_token
Scope *	scope
Expires in	expires_in
	The lifetime in seconds of the access token
Expires on	expires_on
	The expiration time; number of seconds from '1970-01-01T00:00:00Z UTC'
Refresh expires in	refresh_expires_in
	The lifetime in seconds of the refresh token
Refresh expires on	refresh_expires_on
	The expiration time; number of seconds from '1970-01-01T00:00:00Z UTC'

Figure8.76: Personalizzazione del formato della risposta

Formato Risposta

Tipo	Usa payload come Token
Expires in *	300
	The lifetime in seconds of the access token

Figure8.77: Utilizzo del payload di risposta HTTP come access token

Formato Risposta

Tipo	Plugin
ClassName *	org.package.classe

Figure8.78: Personalizzazione del formato della risposta tramite un plugin

Token Policy

Tipo *	Validazione
Nome *	
Descrizione	

Informazioni Generali

Token	
Tipo	Opaco
Discovery Document URL	<input type="checkbox"/>
Posizione	RFC 6750 - Bearer Token Usage
Elaborazione Token	
Token Introspection	<input type="checkbox"/>
OIDC - UserInfo	<input type="checkbox"/>
Token Forward	<input type="checkbox"/>
Endpoint Token	
Connection Timeout *	5000
Read Timeout *	10000
Https	<input type="checkbox"/>
Proxy	<input type="checkbox"/>

Figure8.79: Informazioni generali di una Token Policy

Al passo successivo si inseriscono le Informazioni Generali. Nella sezione *Token* si specifica il tipo di token accettato e il metodo di passaggio.

- *Tipo*: specifica il tipo di token che il gateway attende di ricevere. I valori possibili sono:
 - *JWS*: un JSON Web Token di tipo «Signed».
 - *JWE*: un JSON Web Token di tipo «Encrypt».
 - *Opaco*: un generico token di tipo non specificato.
- *Discovery Document URL*: consente di abilitare la modalità dinamica [OpenID Connect Discovery](#) per recuperare gli endpoint di accesso ai servizi di Introspection, UserInfo e il recupero delle chiavi per una validazione Jwt da una «well-known-url». Se abilitato (Fig. 8.80) le opzioni configurabili sono le seguenti:
 - *Tipo*: indica il formato atteso del payload contenuto nella risposta json:
 - * “OpenID Connect Discovery”: claims definiti in [OpenID Connect Discovery](#) ;
 - * “Personalizzato”: consente di definire un mapping puntuale tra il nome di un claim e l’informazione che GovWay cerca di estrarre dalla risposta (Fig. 8.81);
 - * “Plugin”: consente di indicare il nome di una classe che implementa una logica di parsing personalizzata (deve implementare l’interfaccia «org.openscoop2.pdd.core.token.parser.IDynamicDiscoveryParser»).

Token

Discovery Document URL

Posizione

Elaborazione Token

Token Introspection

OIDC - UserInfo

Token Forward

Discovery Document URL

URL *

Figure8.80: Opzioni “Dynamic Discovery” di una Token Policy

- *Posizione*: indica la modalità di passaggio del token da parte dell’applicativo richiedente. I valori possibili sono:

Token

Tipo	<input type="text" value="JWS"/>
Discovery Document URL	<input checked="" type="checkbox"/>
Posizione	<input type="text" value="RFC 6750 - Bearer Token Usage"/>

Elaborazione Token

Validazione JWT	<input checked="" type="checkbox"/>
Token Introspection	<input checked="" type="checkbox"/>
OIDC - UserInfo	<input checked="" type="checkbox"/>
Token Forward	<input type="checkbox"/>

Discovery Document URL

Tipo	<input type="text" value="Personalizzato"/>
URL *	<input type="text" value="https://example/.well-known/openid-configuration"/>
JWK Set *	<input type="text" value="jwks_uri"/>
Introspection *	<input type="text" value="introspection_endpoint"/>
UserInfo *	<input type="text" value="userinfo_endpoint"/>

Figure8.81: Opzioni “Dynamic Discovery” personalizzata di una Token Policy

- *RFC 6750 - Bearer Token Usage*: la modalità di passaggio del token è una qualsiasi delle tre previste dallo standard RFC 6750 (le tre opzioni successive a questa).
- *RFC 6750 - Bearer Token Usage (Authorization Request Header Field)*: la modalità di passaggio del token è quella che prevede l'inserimento nell'header «Authorization» del messaggio di richiesta. Ad esempio:

```
GET /resource HTTP/1.1
Host: server.example.com
Authorization: Bearer mF_9.B5f-4.1JqM
```

- *RFC 6750 - Bearer Token Usage (Form-Encoded Body Parameter)*: la modalità di passaggio del token è quella di inserirlo nel body della richiesta, eseguita con una POST, utilizzando il parametro *access_token*, come ad esempio:

```
POST /resource HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded

access_token=mF_9.B5f-4.1JqM
```

- *RFC 6750 - Bearer Token Usage (URI Query Parameter)*: la modalità di passaggio del token è quella di utilizzare il parametro *access_token* della Query String, come ad esempio:

```
GET /resource?access_token=mF_9.B5f-4.1JqM HTTP/1.1
Host: server.example.com
```

- *Header HTTP*: la modalità di passaggio del token è quella di inserirlo in un header http custom, il cui nome deve essere fornito nel campo *Nome Header Http*, che appare di seguito.
- *Parametro URL*: la modalità di passaggio del token è quella di inserirlo in un parametro custom della query string. Il nome del parametro deve essere fornito nel campo *Nome Parametro URL*, che appare di seguito.

Nella sezione *Endpoint Token* si specificano eventuali opzioni di accesso agli endpoint:

- *Connection Timeout*: Tempo massimo in millisecondi di attesa per stabilire una connessione;
- *Read Timeout*: Tempo massimo in millisecondi di attesa per la ricezione di una risposta dal server;
- *Https*: Parametri di configurazione nel caso in cui il server richieda un accesso Https;
- *Proxy*: Parametri di configurazione nel caso in cui sia richiesto l'uso di un proxy per l'accesso.

Nella sezione *Elaborazione Token* si specificano le azioni che si possono compiere durante la fase di elaborazione del token ricevuto. Le opzioni disponibili sono:

- Validazione JWT
- Token Introspection
- OIDC - UserInfo
- Token Forward

Le sezioni successive dettagliano questi elementi.

Validazione JWT

Nel caso in cui il token sia di tipo JWT (quindi JWE o JWS), questa opzione attiva la validazione basata su tale standard (Fig. 8.82).

Figure 8.82: Dati di configurazione della validazione JWT

I dati da inserire sono:

- *Formato Token*: indica il formato atteso del payload contenuto nel token JWT. Maggiori dettagli sul mapping vengono forniti in “*Formati dei token*”. I valori possibili sono:
 - *RFC 7519 - JSON Web Token*: claims attesi definiti nel RFC “<https://datatracker.ietf.org/doc/html/rfc7519#section-4>”;
 - *OpenID Connect - ID Token*: definiti nel RFC “https://openid.net/specs/openid-connect-core-1_0.html#IDToken”;
 - *Google - ID Token*: claims definiti in “<https://developers.google.com/identity/protocols/oauth2/openid-connect#obtainuserinfo>”;
 - *Personalizzato*: consente di definire un mapping puntuale tra il nome di un claim e l’informazione che GovWay cerca di estrarre dal token (Fig. 8.87);
 - *Plugin*: consente di indicare il nome di una classe che implementa una logica di parsing personalizzata (deve implementare l’interfaccia “`org.openspcoop2.pdd.core.token.parser.ITokenParser`”).
- *Header*: presente solo in caso di token di tipo JWS, consente di abilitare una validazione dei valori dei claim “*typ*”, “*cty*” o “*alg*” presenti nell’header.

Validazione JWT

Formato Token	RFC 9068 - JSON Web Token (OAuth2 Access Token) <input type="button" value="▼"/>
Header	
<input checked="" type="checkbox"/> Consente di indicare i valori attesi nell'header	
Type (typ)	at+jwt,application/at+jwt
Content Type (cty)	
Algorithm (alg)	
Sia per 'typ', 'cty' o 'alg' è possibile elencare più valori separandoli con la ','	

Figure8.83: Dati di configurazione della validazione di un header JWS

- *TrustStore*: I parametri di configurazione del truststore da utilizzare per il servizio di validazione. Nella configurazione proposta per default il certificato utilizzato per validare il token sarà quello presente all'interno del truststore, corrispondente all'identificativo indicato nel campo "Alias Certificato". In alternativa il certificato è ottenibile tramite le informazioni presenti nel token jwt (x5c, x5t, x5u) attraverso la modalità indicata nel campo "Riferimento X.509". Un certificato ottenuto tramite le informazioni presenti nel token jwt viene sempre validato rispetto al truststore e possono essere abilitati ulteriori criteri di verifica tramite CRL o Policy OCSP (vedi sezione *Online Certificate Status Protocol (OCSP)*).

Token Introspection

Questa sezione consente di attivare la validazione del token ricevuto attraverso un servizio di Token Introspection i cui dati di accesso devono essere forniti in questo contesto (Fig. 8.84).

Per il corretto puntamento al servizio di Token Introspection devono essere forniti in prima istanza i parametri generali legati all'endpoint riferito:

- *Connection Timeout*: Tempo massimo in millisecondi di attesa per stabilire una connessione con il server di validazione token.
- *Read Timeout*: Tempo massimo in millisecondi di attesa per la ricezione di una risposta dal server di validazione token.
- *Https*: Parametri di configurazione nel caso in cui il server di validazione token richieda un accesso Https.
- *Proxy*: Parametri di configurazione nel caso in cui il server di validazione token richieda l'uso di un proxy per l'accesso.

Successivamente devono essere forniti i dati di configurazione specifici del servizio di Token Introspection:

- *Tipo*: tipologia del servizio. A scelta tra i seguenti valori:
 - *RFC 7662 - Introspection*: Servizio di introspection conforme allo standard RFC 7662 "<https://datatracker.ietf.org/doc/html/rfc7662>". Richiede che vengano forniti i seguenti dati:
 - * *URL*: endpoint del servizio di introspection.

The screenshot shows two configuration sections:

- Endpoint Token** (top section):
 - Connection Timeout *****: 10000 (with up/down arrows to adjust)
 - Read Timeout *****: 120000 (with up/down arrows to adjust)
 - Https:
 - Proxy:
- Token Introspection** (bottom section):
 - Tipo *****: RFC 7662 - Introspection
 - URL *****: http://
 - Autenticazione Http Basic:
 - Autenticazione Bearer:
 - AutenticazioneHttps:

Figure8.84: Dati di puntamento al servizio di Token Instrospection

- * *Autenticazione Http Basic*: flag da attivare nel caso in cui il servizio di introspection richieda autenticazione di tipo HTTP-BASIC. In questo caso dovranno essere forniti Username e Password nei campi successivi.
- * *Autenticazione Bearer*: flag da attivare nel caso in cui il servizio di introspection richieda autenticazione tramite un token. Il token dovrà essere indicato nel campo successivo fornito.
- * *AutenticazioneHttps*: flag da attivare nel caso in cui il servizio di introspection richieda autenticazione di tipo Https. In questo caso dovranno essere forniti tutti i dati di configurazione nei campi presenti nella sezione “https”.
- *Google - TokenInfo*: Riferimento al servizio di token introspection di Google. L’unico campo da fornire in questo caso è la URL del servizio. Il sistema precompila questo campo con il valore di default <https://www.googleapis.com/oauth2/v3/tokeninfo>.
- *Personalizzato*: Questa opzione consente di configurare un servizio di Token Introspection personalizzato (Fig. 8.85) attraverso i seguenti dati:
 - * *URL*: la URL del servizio di introspection;
 - * *Autenticazione*: consente di configurare, selezionando il flag opportuno, il tipo di autenticazione richiesta dal servizio di introspection personalizzato;
 - * *Http Method*: il metodo HTTP che deve essere utilizzato per la chiamata al servizio di introspection;
 - * *Posizione Token*: il metodo di passaggio del token al servizio di introspection. Sono supportati i classici metodi: HTTP Authorization Bearer, Header HTTP, Parametro URL e Parametro Form-Encoded Body. Negli ultimi tre casi sarà necessario fornire il nome dell’header o del parametro.

* *Formato Risposta - Tipo*: indica il formato atteso della risposta. Maggiori dettagli sul mapping vengono forniti in “*Formati dei token*”. I valori possibili sono:

- *RFC 7662 - Introspection*: claims attesi definiti nel RFC “<https://datatracker.ietf.org/doc/html/rfc7662#section-2.2>”;
- *Google - TokenInfo*: claims definiti in “<https://developers.google.com/identity/sign-in/web/backend-auth#calling-the-tokeninfo-endpoint>”;
- *Personalizzato*: consente di definire un mapping puntuale tra il nome di un claim e l’informazione che GovWay cerca di estrarre dal token (Fig. 8.87);
- *Plugin*: consente di indicare il nome di una classe che implementa una logica di parsing personalizzata (deve implementare l’interfaccia “org.openspcoop2.pdd.core.token.parser.ITokenParser”).

OIDC - UserInfo

Sezione per attivare la richiesta al servizio di UserInfo per ottenere i dati inerenti l’utente possessore del token ricevuto (Fig. 8.86).

Per il corretto puntamento al servizio di UserInfo devono essere forniti in prima istanza i parametri generali legati all’endpoint riferito, che sono in comune con quelli del servizio di Token Introspection, e quindi già descritti in precedenza.

Successivamente si dovranno fornire i dati di configurazione specifici per il servizio UserInfo, che sono:

- *Tipo*: Si seleziona il tipo di servizio UserInfo riferito. I valori possibili sono:
 - *OpenID Connect - UserInfo*: servizio di UserInfo conforme allo standard OpenID Connect “https://openid.net/specs/openid-connect-core-1_0.html#UserInfo”;
 - *Google - UserInfo*: servizio UserInfo di Google. La URL di default del servizio viene inserita automaticamente.
 - *Personalizzato*: si consente di fornire i dati di configurazione di un servizio personalizzato di UserInfo. I dati di configurazione sono gli stessi già descritti nel caso della configurazione del servizio di Token Introspection personalizzato.
- *URL*: La URL del servizio di UserInfo.
- *Autenticazione*: La configurazione del metodo di autenticazione, quando applicabile.

Token Forward

Azione di elaborazione che consiste nell’inoltro del token ricevuto al destinatario. Una volta attivata questa opzione, devono essere indicate le seguenti informazioni:

- *Originale*: opzione che consente di inoltrare il token originale al destinatario. Attivando questo flag è necessario specificare la modalità di inoltro a scelta tra le seguenti opzioni:
 - *Come è stato ricevuto*: Il token viene inoltrato al destinatario utilizzando lo stesso metodo con cui è stato ricevuto dal gateway.
 - *RFC 6750 - Bearer Token Usage (Authorization Request Header Field)*: Il token viene inoltrato al destinatario utilizzando l’header Authorization presente nella richiesta HTTP.
 - *RFC 6750 - Bearer Token Usage (URI Query Parameter)*: Il token viene inoltrato al destinatario tramite parametro access_token della Query String.

Token Introspection

Tipo	<input type="text" value="Personalizzato"/>
URL *	<input type="text" value="http://"/>
Autenticazione Http Basic	<input type="checkbox"/>
Autenticazione Bearer	<input type="checkbox"/>
AutenticazioneHttps	<input type="checkbox"/>

Configurazione Richiesta

Http Method	<input type="text" value="GET"/>
Posizione Token	<input type="text" value="Parametro URL"/>
Nome Parametro URL *	<input type="text"/>

Formato Risposta

Tipo	<input type="text" value="Personalizzato"/>
Issuer *	<input type="text" value="iss"/>
Subject *	<input type="text" value="sub"/>
Audience *	<input type="text" value="aud"/>
Expire *	<input type="text" value="exp"/>
IssuedAt *	<input type="text" value="iat"/>
NotToBeUsedBefore *	<input type="text" value="nbf"/>
Identifier *	<input type="text" value="jti"/>
ClientId *	<input type="text" value="client_id,azp"/>
Username *	<input type="text" value="preferred_username,username,name"/>
Scope *	<input type="text" value="scope"/>

Figure8.85: Configurazione personalizzata del servizio di Token Instrospection

Endpoint Token

Connection Timeout *	<input type="text" value="10000"/>	<input type="button" value="▲"/> <input type="button" value="▼"/>
Read Timeout *	<input type="text" value="120000"/>	<input type="button" value="▲"/> <input type="button" value="▼"/>
Https	<input type="checkbox"/>	
Proxy	<input type="checkbox"/>	

OIDC - UserInfo

Tipo *	<input type="text" value="OpenID Connect - UserInfo"/>	<input type="button" value="▲"/> <input type="button" value="▼"/>
URL *	<input type="text" value="http://"/>	
Autenticazione Http	<input type="checkbox"/>	
Autenticazione Bearer	<input type="checkbox"/>	
AutenticazioneHttps	<input type="checkbox"/>	

Figure8.86: Dati di puntamento al servizio di UserInfo

- *Header HTTP*: Il token viene inoltrato al destinatario utilizzando un header HTTP il cui nome deve essere specificato nel campo seguente.
- *Parametro URL*: Il token viene inoltrato al destinatario utilizzando un parametro della Query String il cui nome deve essere specificato nel campo seguente.
- *Informazioni Raccolte*: opzione disponibile quando è stata abilitata una delle azioni di validazione del token (introspection, user info o validazione JWT), consente di veicolare i dati ottenuti dal servizio di validazione, al destinatario. Una volta attivato il flag è necessario specificare la modalità di inoltro dei dati selezionando una tra le opzioni seguenti:

Nota: Le informazioni sull'organizzazione PDND sono presenti solamente se è stata attivata l'integrazione con le API PDND descritta nella sezione [API PDND](#)

- *GovWay Headers*: I dati raccolti dal token vengono inseriti nei seguenti header HTTP:

<code>GovWay-Token-Issuer</code>
<code>GovWay-Token-Subject</code>
<code>GovWay-Token-Username</code>
<code>GovWay-Token-Audience</code>
<code>GovWay-Token-ClientId</code>
<code>GovWay-Token-IssuedAt</code>
<code>GovWay-Token-Expire</code>
<code>GovWay-Token-NotToBeUsedBefore</code>
<code>GovWay-Token-Scopes</code>

(continues on next page)

(continua dalla pagina precedente)

```

GovWay-Token-FullName
GovWay-Token-FirstName
GovWay-Token-MiddleName
GovWay-Token-FamilyName
GovWay-Token-EMail
GovWay-Token-PurposeId
GovWay-Token-Jti
GovWay-Token-PDND-OrganizationName
GovWay-Token-PDND-OrganizationCategory
GovWay-Token-PDND-OrganizationExternal

```

- *GovWay JSON*: I dati raccolti dal token vengono inseriti in un oggetto JSON, il cui JsonSchema è il seguente:

```

{
  "required" : [ "id" ],
  "properties": {
    "id": {"type": "string"},
    "issuer": {"type": "string"},
    "subject": {"type": "string"},
    "username": {"type": "string"},
    "audience": {
      "type": "array",
      "items": {"type": "string"}
    },
    "clientId": {"type": "string"},
    "iat": {
      "type": "string",
      "format": "date-time"
    },
    "expire": {
      "type": "string",
      "format": "date-time"
    },
    "nbf": {
      "type": "string",
      "format": "date-time"
    },
    "roles": {
      "type": "array",
      "items": {"type": "string"}
    },
    "scope": {
      "type": "array",
      "items": {"type": "string"}
    },
    "userInfo": {
      "type": "object",
      "properties": {
        "fullName": {"type": "string"},
        "firstName": {"type": "string"},
        "middleName": {"type": "string"}
      }
    }
  }
}

```

(continues on next page)

(continua dalla pagina precedente)

```

        "familyName": {"type": "string"},  

        "email": {"type": "string"}  

    },  

    "additionalProperties": false  

},  

"jti": {"type": "string"},  

"purposeId": {"type": "string"},  

"pdnd": {  

    "type": "object",  

    "properties": {  

        "organization": {  

            "type": "object",  

            "properties": {  

                "name": {"type": "string"},  

                "category": {"type": "string"},  

                "external": {"type": "string"},  

                "externalOrigin": {"type": "string"},  

                "externalId": {"type": "string"}  

            },  

            "additionalProperties": false  

        }  

    },  

    "additionalProperties": false  

}  

},  

"additionalProperties": false  

}  

"claims": {  

    "type": "array",  

    "items": {  

        "name": {"type": "string"},  

        "value": {"type": "string"}  

    },  

    "additionalProperties": false  

},  

"processTime": {  

    "type": "string",  

    "format": "date-time"  

}  

},  

"additionalProperties": false  

}

```

Il JSON risultante viene inserito nell'Header HTTP *GovWay-Token*.

- *GovWay JWS*: I dati raccolti dal token vengono inseriti in un oggetto JSON, come descritto al punto precedente. In questo caso il token JSON viene inserito successivamente in un JWT e quindi firmato. Il JWS risultante viene inserito nell'Header HTTP *GovWay-JWT*.
- *JSON*: Le informazioni ottenute dai servizi di introspection, userinfo o il json estratto dal token jwt dopo la validazione, vengono inseriti negli header http o nelle proprietà della url indicati.

Nota: Le informazioni sono esattamente quelle recuperate dai servizi originali (o presenti nel token originale nel caso di validazione jwt).

-
- *JWS/JWE*: Uguale alla modalità JSON con la differenza che negli header http, o nelle proprietà della url,

vengono inseriti dei JWT firmati (caso JWS) o cifrati (caso JWE) contenenti al loro interno il JSON.

Formati dei token

Nella funzionalità “*Token Policy Validazione*” viene attuato un parsing del token ricevuto nel caso sia abilitata la “*Validazione JWT*” per estrarre le informazioni principali che vengono registrate da GovWay e possono essere inoltrate al backend sotto forma di header di integrazione (“*Token Forward*”). Un parsing delle informazione avviene inoltre anche se risulta attivata la funzionalità “*Token Introspection*” e/o “*OIDC - UserInfo*”. Ogni funzionalità precedentemente indicata richiede che venga indicato il formato del token per poter interpretare correttamente le informazioni presenti. Di seguito viene fornita una tabella di mapping tra le informazioni che GovWay cerca di estrarre dal token e i nomi dei claims rispetto al formato impostabile nelle funzionalità suddette.

Table8.2: Mapping informazione-claim per ogni formato di token

Informazione	RFC 9068 - OAuth2 Access Token	RFC 7519 - JSON Token	RFC 7662 - Introspection	OpenID Connect ID Token	Google - Token	ID
Issuer	iss	iss	iss	iss	iss	
Subject	sub	sub	sub	sub	sub	
Audience	aud	aud	aud	aud	aud	
Expire	exp	exp	exp	exp	exp	
IssuedAt	iat	iat	iat	iat	iat	
NotToBeUsedBefore	nbf	nbf	nbf	non supportato	non supportato	
Identifier	jti	jti	jti	non supportato	non supportato	
Scope	scope	scope	scope	scope	scope	
ClientId	client_id	non supportato	client_id	azp	azp	
Username	non supportato	non supportato	username	preferred_username	name	
User Full name	non supportato	non supportato	non supportato	name	name	
User First name	non supportato	non supportato	non supportato	given_name	given_name	
User Middle name	non supportato	non supportato	non supportato	middle_name	middle_name	
User Family name	non supportato	non supportato	non supportato	family_name o last_name	family_name	
User eMail	non supportato	non supportato	non supportato	email	email	

All'interno di ogni funzionalità presente in “*Validazione JWT*” è anche inoltre possibile indicare un formato personalizzato che consente di definire un mapping puntuale tra il nome di un claim e l'informazione che GovWay cerca di estrarre dal token. Per ogni campo possono essere indicati più claims, separandoli con la virgola, ed in tal caso nel token verranno cercati nell'ordine in cui sono definiti. L'indicazione di un claim per ogni informazione non è vincolante rispetto alla presenza di tale claim all'interno del token.

8.9 Attribute Authority

Le Attribute Authority (AA) sono regolate dalle «Linee guida dei gestori di attributi qualificati» rilasciate da AGID ed operano erogando API che gestiscono gli attributi qualificati di persone fisiche o giuridiche.

Un attributo qualificato descrive una proprietà di un'identità e si definisce qualificato perché è attestato da un soggetto (Attribute Authority) cui la legge conferisce tale potere. La descrizione e il formato di ogni attributo è specifico dell'Attribute Authority alla quale è possibile richiedere attributi solo mediante la stipula di una convenzione. Inoltre le singole AA definiscono nelle proprie specifiche di integrazione quali siano gli elementi obbligatori che devono essere

Formato Token	Personalizzato
Issuer *	iss
Subject *	sub
Audience *	aud
Expire *	exp
IssuedAt *	iat
NotToBeUsedBefore *	nbf
Identifier *	jti
ClientId *	azp,client_id
Username *	preferred_username,username,name
Scope *	scope
Role *	role
User - Full name *	name
User - First name *	given_name
User - Middle name *	middle_name
User - Family name *	family_name,last_name
User - eMail *	email,e-mail,e_mail,mail

Per ogni campo possono essere indicati più claims, separandoli con la virgola.
Nel token verranno cercati nell'ordine in cui sono definiti.
L'indicazione di un nome non è vincolante rispetto alla presenza all'interno del token.

Figure8.87: Personalizzazione del formato del token

presenti nelle richieste tra cui l'informazione necessaria ad identificare il soggetto per cui si stanno richiedendo gli attributi.

GovWay supporta l'interazione con le Attribute Authority nella fase di verifica dell'autorizzazione all'accesso ad una API, permettendo di utilizzare gli attributi ottenuti dalle AA nelle politiche di accesso alle API.

Per poter definire politiche di controllo degli accessi basate sugli attributi è necessario registrare una o più AA. Queste potranno poi essere riferite nella configurazione delle singole API.

La gestione delle AA si effettua dalla sezione *Configurazione > Attribute Authority* della govwayConsole. Per registrare una nuova si utilizza il pulsante *Aggiungi*. Il form di creazione appare inizialmente come quello illustrato in Fig. 8.88.

The screenshot shows a web-based configuration interface. At the top, a dark header bar contains the text 'Attribute Authority > Aggiungi'. Below this, the main form area has a light gray background. A note 'Note: (*) Campi obbligatori' is displayed in a small font. The form is titled 'Attribute Authority'. It contains two text input fields: 'Nome' (Name) with a red asterisk (*) indicating it is required, and 'Descrizione' (Description). Both fields have a placeholder text 'Nome' and 'Descrizione' respectively.

Figure8.88: Registrazione di una Attribute Authority

Inizialmente si inseriscono i dati identificativi:

- *Nome*: nome univoco da assegnare all'AA
- *Descrizione*: testo di descrizione generale

Le sezioni successive dettagliano i criteri con cui si compone una richiesta di attributi e l'endpoint a cui deve essere spedita. Infine deve essere istruito GovWay su come interpretare la risposta di attributi ricevuta dall'AA.

8.9.1 Endpoint di una Attribute Authority

In questa sezione vengono descritti i parametri di connessione alla AA.

- *URL*: endpoint dell'AA a cui è possibile inviare una richiesta di attributi. Il valore può essere definito come costante o contenere parti dinamiche risolte a runtime dal Gateway (per maggiori dettagli [Valori dinamici](#)).
- *Connection Timeout*: Tempo massimo in millisecondi di attesa per stabilire una connessione con il server.
- *Read Timeout*: Tempo massimo in millisecondi di attesa per la ricezione di una risposta dal server.
- *Https*: Parametri di configurazione nel caso in cui l'AA richieda un accesso Https.
- *Proxy*: Parametri di configurazione nel caso in cui l'AA richieda l'uso di un proxy per l'accesso.

Successivamente devono essere forniti i dati di configurazione specifici dell'autenticazione client, se richiesto dall'AA:

- *Autenticazione Http Basic*: flag da attivare nel caso in cui l'AA richieda autenticazione di tipo HTTP-BASIC. In questo caso dovranno essere forniti Username e Password nei campi successivi.
- *Autenticazione Bearer*: flag da attivare nel caso in cui l'AA richieda autenticazione tramite un bearer token. Il token dovrà essere indicato nel campo successivo fornito.

Endpoint

URL * http:// i

Connection Timeout * 10000

Read Timeout * 120000

Https

Proxy

Autenticazione Client

Basic

Bearer

Https

Figure8.89: Endpoint di un Attribute Authority

- **Autenticazione Https:** flag da attivare nel caso in cui l'AA richieda autenticazione di tipo Https. In questo caso dovranno essere forniti tutti i dati di configurazione nei campi presenti nella sezione “https”.

Nel caso sia attivato il flag relativo ad un Proxy o una configurazione Https saranno presentate delle sezioni omonime dove poter inserire i dati di configurazione richiesti.

8.9.2 Richiesta di Attributi

Ogni singola AA definisce nella propria interfaccia quali siano gli elementi obbligatori che devono essere presenti nelle richieste, tra cui l'informazione necessaria ad identificare il soggetto a cui si riferiscono gli attributi richiesti.

La sezione seguente consente di definire come GovWay debba formare la richiesta che verrà inoltrata all'endpoint configurato nella sezione *Endpoint di una Attribute Authority*.

- *Posizione:* indica dove risiede la richiesta di attributi nella comunicazione HTTP:
 - *Authorization Bearer:* richiesta inserita nell'header HTTP “Authorization” con prefisso “Bearer”;
 - *HTTP Payload:* richiesta veicolata come payload http;
 - *Header HTTP:* richiesta inserita in un header HTTP il cui nome viene definito nel campo successivo fornito;
 - *Parametro URL:* richiesta inserita come parametro della url il cui nome viene definito nel campo successivo fornito.
- *Http Method:* consente di selezionare il tipo di richiesta HTTP da utilizzare tra quelle compatibili con la *Posizione* della richiesta di attributi.
- *Tipo Richiesta:* indica il formato della richiesta:
 - *JWS:* token JWT firmato (<https://datatracker.ietf.org/doc/html/rfc7515>);
 - *JSON:* consente di definire la richiesta di attributi tramite la definizione di un template;

- *Personalizzata*: simile alla precedente opzione, consente inoltre di impostare il Content-Type associato alla richiesta.

Richiesta nel formato JWT

Nel caso di richiesta di tipo *JWS* si devono fornire le informazioni necessarie a produrre il JWT firmato così suddivise:

- *JWS KeyStore*: consente di fornire i parametri di accesso al keystore contenente la chiave privata ed il certificato da utilizzare per firmare, tramite una dei seguenti tipi:
 - “JKS” o “PKCS12”: viene richiesta l’indicazione del path assoluto del keystore nel campo *File*, la definizione della password per l’accesso al keystore nel campo *Password*, l’alias con cui è riferita la chiave privata nel keystore nel campo *Alias Chiave Privata* e la password della chiave privata nel campo *Password Chiave Privata*;
 - “JWK Set”: deve essere definito il path su filesystem dove risiede l’archivio json nel formato “JWK Set” e l’identificativo “kid” (alias) con cui è riferita la chiave privata nel campo *Alias Chiave Privata*;
 - “Key Pair”: deve essere definito il path su filesystem dove risiedono la chiave privata e pubblica in formato PEM o DER (sono supportati sia i formati pkcs1 che pkcs8) e la password della chiave privata se cifrata nel campo *Password Chiave Privata*;
 - Tipi PKCS11: i tipi disponibili sono quelli corrispondenti ai tipi di keystore PKCS11 registrati (“*Device PKCS11*”).
- *JWS Header*: consente di indicare quali dati debbano essere inseriti nella parte header (non firmati) del JWT; tra i parametri impostabili vi sono l’algoritmo di firma e l’indicazione se deve essere inserito il certificato utilizzato per la firma nell’header (x5c).
- *JWS Payload*: consente di impostare i valori dei claim presenti nella parte body (firmata) del JWT. Vengono fornite differenti modalità con le quali poter definire il payload:
 - *RFC7515*: consente di definire i claims standard (“iss”, “sub” e “aud”) e la validità temporale del JWT. Inoltre è possibile definire ulteriori claims da inserire nel body indicandoli per riga (nome=valore) nel campo “Claims” (modalità descritte nella sezione *Aggiunta di Claims nei Token*). I claim “iss”, “sub”, “aud” e gli eventuali claims aggiuntivi possono essere definiti tramite costanti o possono contenere parti dinamiche risolte a runtime dal Gateway (per maggiori dettagli *Valori dinamici*).
 - *Template*: il payload viene definito tramite un template che può contenere parti dinamiche risolte a runtime definite tramite una sintassi proprietaria di GovWay.
 - *Freemarker Template*: il payload viene definito utilizzando il template «Freemarker» (<https://freemarker.apache.org/>).
 - *Velocity Template*: il payload viene definito utilizzando il template «Velocity» (<http://velocity.apache.org/>).

Richiesta in altri formati

Nel caso di richiesta di tipo *Json* o *Personalizzata* si deve fornire un template che definisce la richiesta di attributi. Il tipo di template utilizzabile è selezionabile tra i seguenti:

- *Template*: il contenuto della richiesta viene definito tramite un template che può contenere parti dinamiche risolte a runtime definite tramite una sintassi proprietaria di GovWay;
- *Freemarker Template*: il contenuto della richiesta viene definito utilizzando il template «Freemarker» (<https://freemarker.apache.org/>);
- *Velocity Template*: il contenuto della richiesta viene definito utilizzando il template «Velocity» (<http://velocity.apache.org/>).

Valori dinamici utilizzabili nei Template

Richiesta

Posizione	Authorization Bearer
Http Method	GET
Tipo Richiesta	JWS

JWS KeyStore

Tipo	JKS
File *	<input type="text"/>
Password *	<input type="text"/>
Alias Chiave Privata *	<input type="text"/>
Password Chiave Privata *	<input type="text"/>

JWS Header

Signature Algorithm	RS256
Key Id (kid)	<input type="checkbox"/>
X.509 Certificate	<input type="text"/>
Digest X.509 Certificate	<input type="text"/>
Type (typ)	JWT
Content Type (cty)	<input type="checkbox"/>

JWS Payload

Modalità	RFC7515
Issuer *	<input type="text"/> (i)
Subject *	<input type="text"/> (i)
Audience *	<input type="text"/> (i)
Time to Live (secondi) *	300

Indica la validità temporale, in secondi, a partire dalla data di creazione dell'asserzione

Claims	<input type="text"/> (i)
--------	---------------------------------------

Figure8.90: Richiesta di Attributi nel formato JWS con modalità “RFC7515”

Richiesta

Posizione	HTTP Payload
Http Method	POST
Tipo Richiesta	JSON

Payload

Tipo Template	Template
Contenuto *	<input type="text"/>
Content-Type *	application/json

Figure8.91: Richiesta di Attributi nel formato JSON

I costrutti utilizzabili nei template sono gli stessi utilizzabili per la funzionalità di trasformazione, come descritti nella sezione “*Valori dinamici*”, arricchiti di un’ulteriore istruzione che consente di individuare gli attributi da richiedere, così come configurati sulla specifica fruizione o erogazione di API nella quale è stata riferita l’AA :

- *requiredAttributes:METHOD* : il valore “METHOD” fornito deve rappresentare un metodo valido all’interno della classe “org.openspcoop2.pdd.core.token.attribute_authority.RequiredAttributes”
 - Se la richiesta è definita tramite template con la sintassi specifica di GovWay, gli attributi saranno direttamente accessibili utilizzando il formato “\${requiredAttributes:METHOD}”; ad es. per ottenere la lista degli attributi in un formato utilizzabile all’interno di un array json usare \${requiredAttributes:jsonList()} oppure \${requiredAttributes:formatList(<, >)}
 - Se la richiesta è definita tramite template Freemarker o Velocity, l’oggetto contenente gli attributi da richiedere è presente nel contesto con chiave di accesso “aa”.

Di seguito un esempio di template GovWay che definisce una richiesta JSON in cui l’identità della persona fisica per cui si richiedono gli attributi viene prelevata dal token OAuth e gli attributi richiesti sono quelli configurati nell’erogazione di API:

```
{
  "attributes": [${requiredAttributes:jsonList()}],
  "fiscalCode": "${tokenInfo:username}"
}
```

8.9.3 Risposta della Attribute Authority

Ogni singola AA utilizza un proprio formato per la descrizione degli attributi nella risposta fornita. Il tipo della risposta deve essere definito nel campo “*Tipo Risposta*”. Di seguito vengono descritte le opzioni richieste per ogni tipo.

- **JWS:** la risposta viene gestita come token JWT firmato (<https://datatracker.ietf.org/doc/html/rfc7515>) presente nel payload http. Deve essere indicato il claim che contiene gli attributi richiesti ed è possibile elencare più claim separandoli tramite virgola.

Nella sezione “*TrustStore*” devono essere indicati i dati che consentono di accedere al truststore da utilizzare per validare il token jws. Nella configurazione proposta per default il certificato utilizzato per validare il token sarà quello presente all’interno del truststore, corrispondente all’identificativo indicato nel campo “*Alias Certificato*”. In alternativa il certificato è ottenibile tramite le informazioni presenti nel token jwt (x5c, x5t, x5u) attraverso la modalità indicata nel campo “*Riferimento X.509*”. Un certificato ottenuto tramite le informazioni presenti nel token jwt viene sempre validato rispetto al truststore e possono essere abilitati ulteriori criteri di verifica tramite CRL o Policy OCSP (vedi sezione *Online Certificate Status Protocol (OCSP)*).

Risposta

Tipo Risposta: JWS

Attributi *

Indicare il claim che contiene gli attributi.
È possibile elencare più claims separandoli con la virgola

TrustStore

Tipo: JKS

File *

Password *

Riferimento X.509: Alias in TrustStore

Per la validazione viene utilizzato il certificato nel truststore corrispondente all’alias indicato

Alias Certificato *

Figure8.92: Risposta di Attributi nel formato JWS

- **JSON:** la risposta viene processata come messaggio JSON. Se gli attributi sono contenuti in uno o più elementi devono esserne elencati i nomi separandoli tramite virgola. Invece lasciando vuoto il campo “*Attributi*” tutti gli elementi presenti saranno interpretati come attributi.
- **Personalizzata:** la risposta viene processata tramite la classe indicata nel campo “*ClassName*”. La classe fornita deve implementare l’interfaccia “org.openspcoop2.pdd.core.token.attribute_authority.IRetrieveAttributeAuthorityResponseParser”.

Risposta

Tipo Risposta	<input type="text" value="JSON"/>
Attributi	<input type="text"/>
Se gli attributi sono contenuti in uno o più elementi, elencarne i nomi separandoli con la virgola. Lasciando vuoto questo campo tutti gli elementi ritornati saranno interpretati come attributi	

Figure8.93: Risposta di Attributi nel formato JSON

Risposta

Tipo Risposta	<input type="text" value="Personalizzata"/>
ClassName *	<input type="text"/>

Figure8.94: Risposta di Attributi in un formato personalizzato

8.10 Tags

La sezione *Configurazione > Tags* è dedicata alla gestione dei tags che possono essere utilizzati per la classificazione delle API presenti nel registro.

I tags possono essere creati direttamente durante la registrazione di una API, oppure da questa sezione in maniera più sistematica e assegnando loro un tipo, Soap o Rest, che indica l'ambito di utilizzo del tag stesso.

La sezione mostra l'elenco dei tags disponibili (Fig. 8.95).

L'elenco dei tag può essere filtrato impostando, nella barra dei filtri a comparsa, un pattern per il nome o un tipo. Oltre ad aggiungere ed eliminare i tag esistenti è possibile esportarli in blocco.

Col pulsante *Aggiungi* si apre il form per creare un nuovo tag (Fig. 8.96).

Per creare un tag si inseriscono i seguenti dati:

- *Nome*: il nome del tag
- *Descrizione*: descrizione del tag
- *Tipo*: serve per indicare per quali API è possibile utilizzare il tag: SOAP, REST o Qualsiasi.

	Nome	Tipo
<input type="checkbox"/>	altroTag	Qualsiasi
<input type="checkbox"/>	Anagrafica	Qualsiasi
<input type="checkbox"/>	PagamentiTelematici	Qualsiasi
<input type="checkbox"/>	PagamentiTelematiciREST	Rest
<input type="checkbox"/>	PagamentiTelematiciSOAP	Soap
<input type="checkbox"/>	tagTest	Qualsiasi
<input type="checkbox"/>	tagTest1	Qualsiasi
<input type="checkbox"/>	tagTest2	Qualsiasi

ESPORTA **ELIMINA** **AGGIUNGI**

Figure8.95: Elenco dei tags

Tags > Aggiungi

Note: (*) Campi obbligatori

Tag

Nome *	<input type="text"/>
Descrizione	<input type="text"/>
Tipo	<input type="text" value="Qualsiasi"/>

SALVA

Figure8.96: Creazione di un tag

8.11 Utenti

La sezione *Configurazione > Utenti* è dedicata alla gestione degli utenti dei cruscotti grafici govwayConsole e govwayMonitor.

Prima di descrivere le funzionalità relative alla gestione utenti è necessario fare una premessa sull'organizzazione dei permessi che sono assegnabili ad un utente.

Le funzionalità delle console grafiche sono partizionate in gruppi cui corrispondono puntuali permessi che possono essere concessi agli utenti per limitarne l'operatività. Vediamo quali sono i gruppi funzionali, e conseguentemente i permessi associabili a ciascun utente:

- *Registro*
 - *Gestione API [S]* - Gestione delle entità di configurazione dei servizi, quali: API, Erogazioni, Fruizioni, ecc.
- *GovWay Monitor*
 - *Monitoraggio [D]* - Accesso alle funzionalità di monitoraggio della console govwayMonitor.
 - *Reportistica [R]* - Accesso alle funzionalità di reportistica della console govwayMonitor.
- *Strumenti*
 - *Auditing [A]* - Accesso alle funzionalità di consultazione delle tracce del servizio di Auditing.
- *Configurazione*
 - *[C]* - Accesso alle funzionalità di configurazione. Queste funzionalità sono quelle presenti nel menu di navigazione nel gruppo *Configurazione* e riguardano: tracciamento, controllo del traffico, import-export, ecc.
 - *[U]* - Possibilità di gestire gli utenti delle console. Gli utenti con questo permesso, sono di fatto dei superutenti in quanto possono assumere l'identità di un qualunque utente del sistema.

L'applicazione, al termine dell'installazione, contiene una utenza (credenziali indicate durante l'esecuzione dell'installer) che permette di effettuare tutte le principali operazioni di gestione.

Gli utenti in possesso del permesso [U] possono creare dei nuovi utenti.

La maschera di creazione di un nuovo utente è quella mostrata in [Fig. 8.97](#).

Le informazioni da inserire sono:

- *Informazioni Utente*
 - *Nome*
- *Permessi di Gestione*: sezione che consente di assegnare i permessi all'utente e quindi decidere quali funzionalità rendergli accessibili.
- *Profilo di Interoperabilità*: sezione che consente di decidere quali, tra i profili disponibili, rendere accessibili all'utente.
- *Visibilità dati tramite govwayMonitor*: questa sezione è visibile solo se è stato abilitato uno dei permessi «GovWay Monitor» ([Fig. 8.98](#)). In questo contesto è possibile stabilire la visibilità dell'utente sulla console GovWay Monitor riguardo i seguenti:
 - *Soggetti*: opzione visibile solo se attiva la modalità multi-tenant, consente di limitare la visibilità delle entità di monitoraggio ai soli soggetti interni indicati in una whitelist. Per configurare la whitelist è necessario salvare l'utente da creare e successivamente accedere in editing. In alternativa è possibile attivare il flag «Tutti» per non assegnare limitazioni.

Figure 8.97: Creazione nuovo utente

- **API:** consente di limitare la visibilità delle entità di monitoraggio alle sole API indicate in una whitelist. Per configurare la whitelist è necessario salvare l'utente da creare e successivamente accedere in editing. In alternativa è possibile attivare il flag «Tutti» per non assegnare limitazioni.
- **Profilo Utente:** questa sezione è visibile solamente se è stato abilitato uno dei permessi «GovWay Monitor» (Fig. 8.98) o **Registro** (Fig. 8.99) e consente di impostare alcuni criteri di default associati all'utenza durante l'utilizzo della console:
 - Modalità Interfaccia (solo per govwayConsole): consente di decidere quale modalità, tra standard e avanzata, è quella di default per l'utente;
 - Profilo Interoperabilità: consente di impostare un profilo di interoperabilità di default all'utente;
 - Soggetto Operativo: voce presente solamente se è stato selezionato un profilo di interoperabilità, consente di associare un soggetto operativo di default all'utente;
 - Home Page (solo per govwayMonitor): definisce la homepage visualizzata a login effettuato, consentendo di scegliere tra la pagina di ricerca delle transazioni o un report statistico;
 - Intervallo Temporale (solo per govwayMonitor): voce presente solo se è stato selezionato un report statistico per l'homepage della console di monitoraggio, consentendone di indicare l'intervallo temporale.
- **Password:** sezione per l'impostazione della password dell'utente.

Nota: I criteri minimi di sicurezza che una password deve soddisfare sono configurabili agendo sul file <directory-lavoro>/consolePassword.properties:

Figure8.98: Creazione nuovo utente (Sezione govwayMonitor)

Figure8.99: Creazione nuovo utente (Sezione Profilo Utente)

```

# Abilitare l'opzione seguente per poter autenticare:
# La password deve rispettare tutti i vincoli impostati

# Deve soddisfare le seguenti espressioni regolari
#passwordVerifier.regularExpression.EXP1=reg1
#..
#passwordVerifier.regularExpression.EXPN=regn

# Non deve contenere il nome di login dell'utente
passwordVerifier.notContainsLogin=true

# Non deve corrispondere ad una delle seguenti parole riservate
#passwordVerifier.restrictedWords=root, admin, administrator, amministratore

# Deve essere composta almeno da x caratteri
passwordVerifier.minLength=8

# Non deve essere composta da più di x caratteri
#passwordVerifier.maxLength=20

# Deve contenere almeno una lettera minuscola (a - z)
passwordVerifier.lowerCaseLetter=true

# Deve contenere almeno una lettera maiuscola (A - Z)
passwordVerifier.upperCaseLetter=true

```

(continues on next page)

(continua dalla pagina precedente)

```

# Deve contenere almeno un numero (0 - 9)
passwordVerifier.includeNumber=true

# Deve contenere almeno un carattere non alfabetico (ad esempio, !, $, #, %, @)
passwordVerifier.includeNotAlphanumericSymbol=true

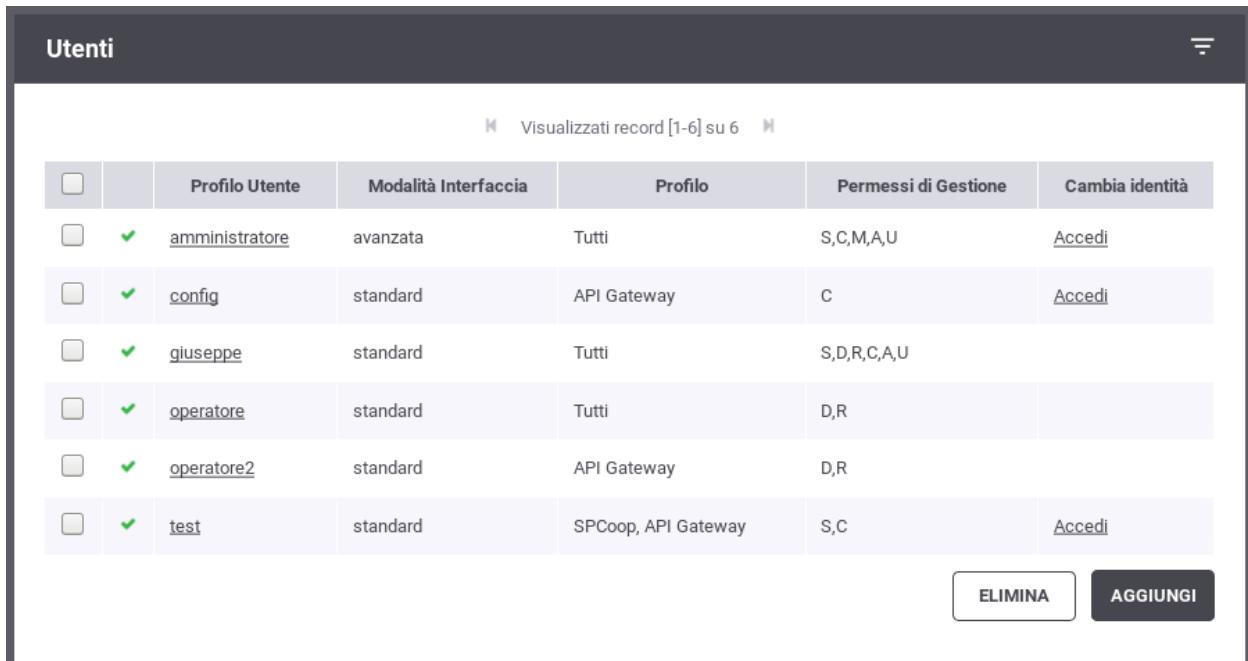
# Tutti i caratteri utilizzati devono essere differenti
#passwordVerifier.allDistinctCharacters=true

# La password dovrà essere aggiornata ogni 90 giorni
# Impostare un valore <=0 per disabilitare la verifica
#passwordVerifier.expireDays=90
passwordVerifier.expireDays=-1

# Abilita lo storico delle password non consentendo di aggiornare la password
#corrente con una precedentemente già impostata.
passwordVerifier.history=true

```

La pagina indice della sezione Utenti visualizza gli utenti già presenti nel sistema con i relativi permessi e i link per modificarli o assumerne l'identità (Fig. 8.100)



		Profilo Utente	Modalità Interfaccia	Profilo	Permessi di Gestione	Cambia identità
<input type="checkbox"/>	<input checked="" type="checkbox"/>	amministratore	avanzata	Tutti	S,C,M,A,U	Accedi
<input type="checkbox"/>	<input checked="" type="checkbox"/>	config	standard	API Gateway	C	Accedi
<input type="checkbox"/>	<input checked="" type="checkbox"/>	giuseppe	standard	Tutti	S,D,R,C,A,U	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	operatore	standard	Tutti	D,R	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	operatore2	standard	API Gateway	D,R	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	test	standard	SPCoop, API Gateway	S,C	Accedi

Figure 8.100: Lista degli utenti

Nota: La password generata e assegnata all'utente viene visualizzata solamente nell'avviso visualizzato in seguito alla creazione (Fig. 8.101) e successivamente non è più consultabile.

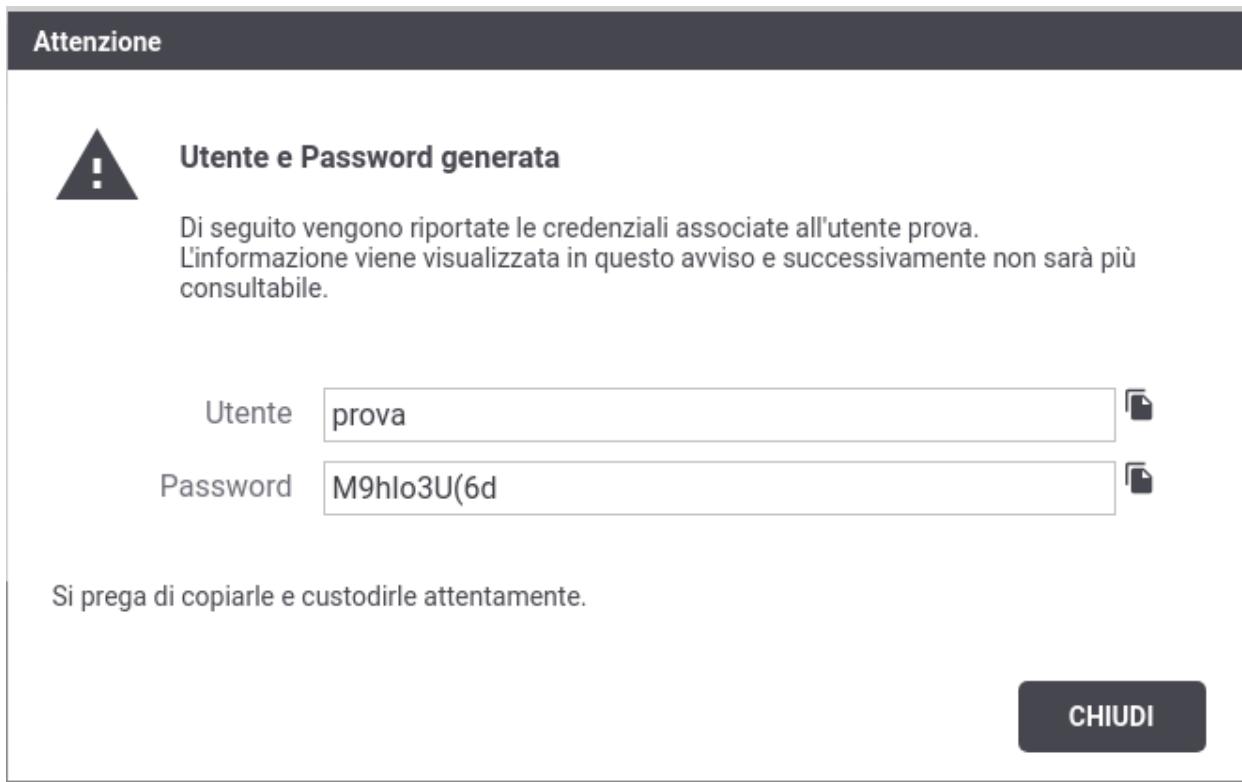


Figure8.101: Avviso di copia delle credenziali dell'utente

Nel caso di smarrimento della password è necessario procedere con la generazione di una nuova password (Fig. 8.102).

Figure8.102: Aggiornamento delle credenziali dell'utente

8.12 Importa

L'importazione di entità nel registro può essere effettuata tramite la sezione accessibile con la voce di menu *Importa* presente nella sezione *Configurazione*.

Gli archivi che possono essere importati devono essere nel formato atteso da govway e sono ottenibili:

- attraverso un'esportazione effettuabile tramite govwayConsole come indicato nella sezione *Esporta*
- scaricando le govlets disponibili sul sito del progetto che permettono di pre-configurare GovWay per una specifica API

Il form che compare per l'importazione è quello riportato in Fig. 8.103. I passi da eseguire sono i seguenti:

- *Validazione Documenti* (disponibile solamente con interfaccia in modalità avanzata, per default è abilitato): Se attivato, questo flag indica che i documenti presenti nell'archivio vengono validati prima di essere importati (es. wsdl, xsd, openapi 3, swagger 2 ...).
- *Aggiornamento*: Se attivato, questo flag indica che l'archivio da importare costituisce un aggiornamento del registro attuale. Gli elementi presenti nell'archivio, che risultano già esistere sul registro di GovWay, verranno aggiornati solamente se il flag viene abilitato.
- *Policy di Configurazione*: eventuali policy globali (Token, Rate Limiting) presenti nell'archivio verranno importate solamente se il flag viene abilitato.
- *Configurazione di GovWay*: una eventuale configurazione presente nell'archivio verrà importata solamente se il flag viene abilitato.
- Selezionare dal filesystem il file che corrisponde all'archivio che deve essere importato.

Figure 8.103: Importazione di entità nel registro

Nota: Attraverso l'abilitazione di configurazioni avanzate relative ai Profili di Interoperabilità può comparire una ulteriore scelta iniziale che serve ad indicare la modalità cui fanno riferimento le entità contenute nell'archivio da importare.

Ad esempio, per il Profilo SPCoop se viene abilitata la proprietà “org.openspcoop2.protocol.spcoop.packageSICA” nel file locale “/etc/govway/spcoop_local.properties”, verrà richiesto quale tipo di archivio si vuole importare a scelta tra:

- *spcoop*: il formato standard basato sulle specifiche SPCoop
- *govlet*: il formato di govway

8.13 Esporta

L'esportazione dei dati di configurazione dalla govwayConsole è possibile nei modi seguenti:

- Selezionando singolarmente le entità di configurazione da esportare, come ad esempio «Erogazioni» o «API», e premendo il pulsante *Esporta* (Fig. 8.104).

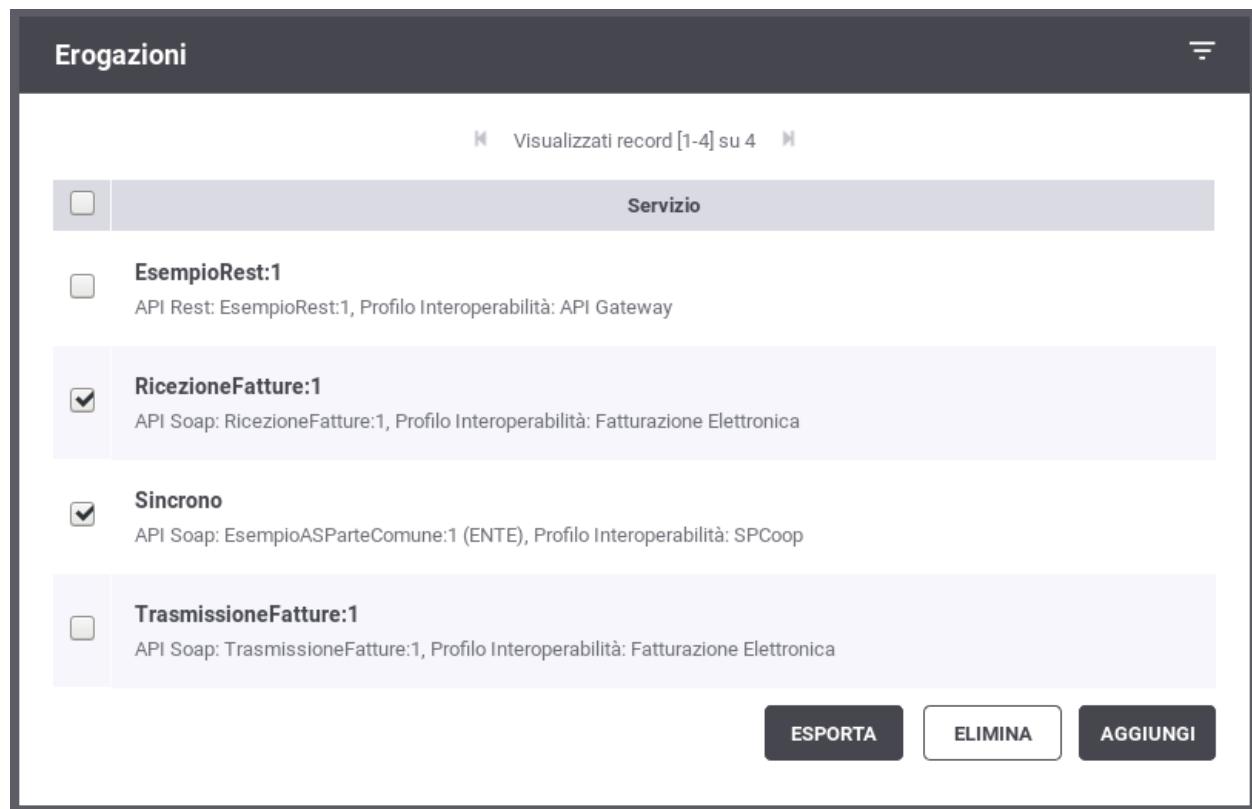


Figure8.104: Esportazione di singole entità del registro

Dopo aver selezionato il pulsante «Esporta», una seconda maschera (Fig. 8.105) riporta le seguenti informazioni:

- *Profilo Interoperabilità*: indicazione del profilo cui fa riferimento l'esportazione.
- *Soggetto*: indicazione del dominio cui fa riferimento l'esportazione.
- *Tipologia archivio*: se previsto dal Profilo di Interoperabilità, fa selezionare la tipologia di archivio da produrre. Il default è il formato *Govlet* standard di esportazione di Govway.
- *Policy di Configurazione*: se il flag viene abilitato vengono incluse nell'archivio esportato le policy globali (Token, Rate Limiting) condivise tra più API

- *Elementi di Registro*: se il flag viene abilitato vengono incluse nell’archivio esportato anche gli elementi del registro riferiti da quelli selezionati.

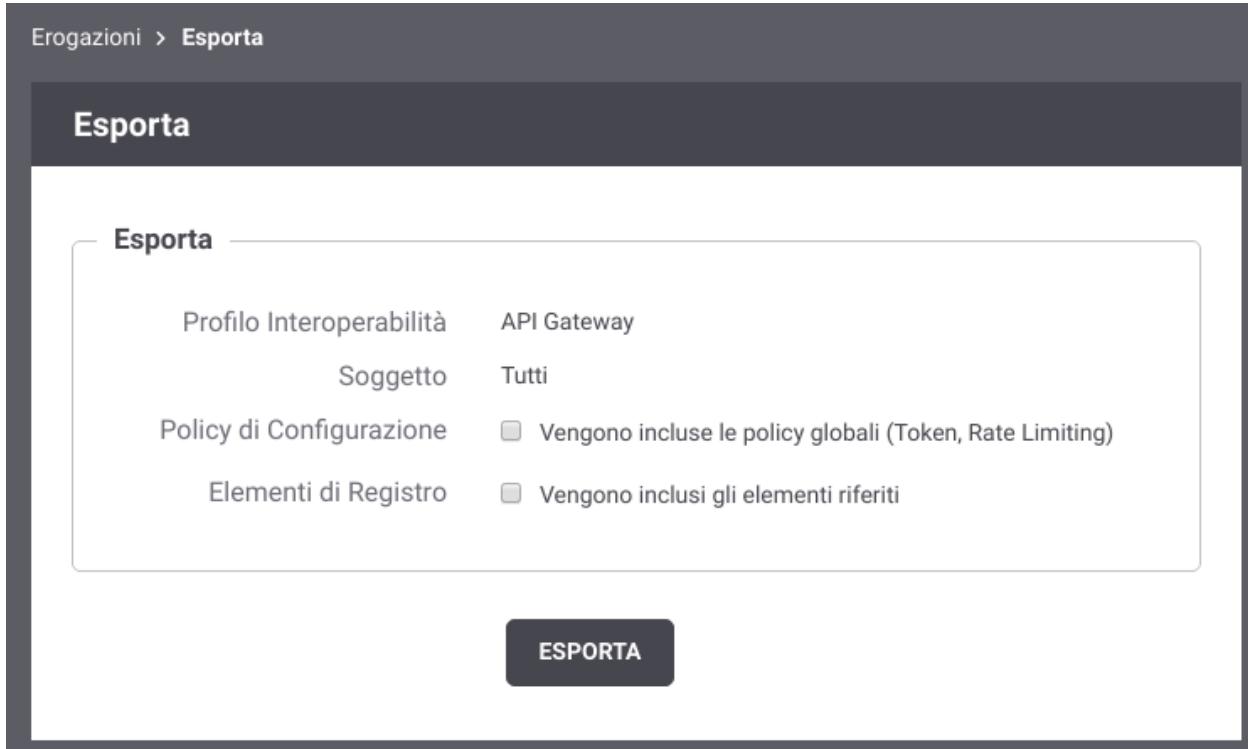


Figure8.105: Esportazione di entità nel registro: parametri

- Tramite la voce di menu *Configurazione > Esporta* che presenta le opzioni mostrate in Fig. 8.106.

Le opzioni presenti sono:

- *Profilo Interoperabilità*: indica quale profilo riguarda l’esportazione che si sta effettuando
- *Tipologia Archivio*: nei casi che lo prevedono, consente di specificare il formato dell’archivio di esportazione da produrre.
- *Modalità*: consente di specificare cosa esportare tra le seguenti possibilità:
 - * *Esportazione completa*: esportazione dell’intero repository di configurazione (limitatamente al profilo di interoperabilità selezionato, se diverso da «Tutti»).
 - * *Registro*: esporta solo le entità del registro (erogazioni, fruizioni, api, ecc)
 - * *Configurazione*: esporta solo le entità della sezione Configurazione (token policy, tracciamento, ecc).

Il formato dell’archivio prodotto come risultato dell’esportazione dipende dalla modalità cui fanno riferimento le entità selezionate.

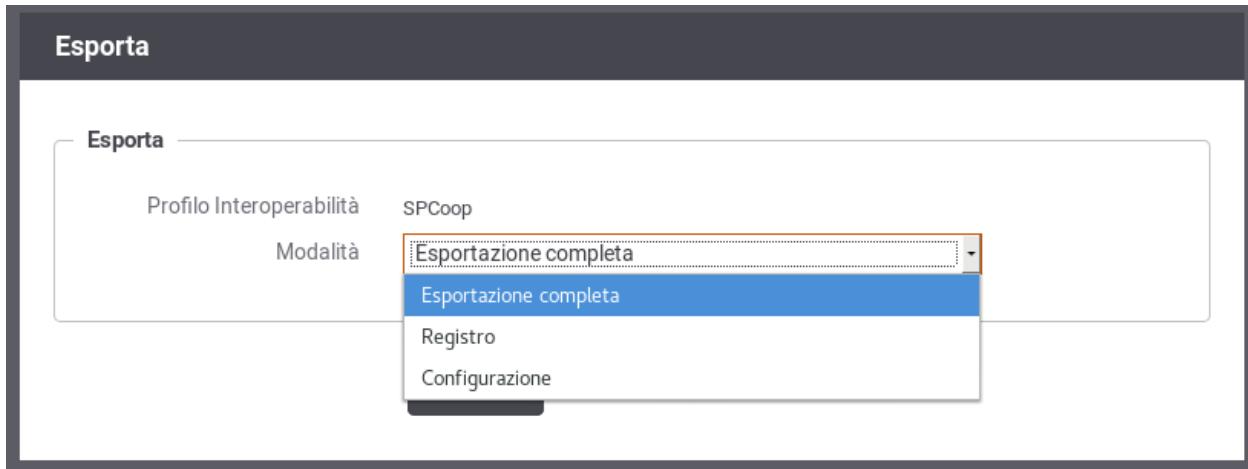


Figure8.106: Esportazione di entità nel registro

8.14 Auditing

In questa sezione descriviamo le modalità di configurazione del servizio di auditing, al fine di definire quali informazioni devono essere tracciate, con che formato e con che livello di dettaglio.

Gli utenti con permesso [C] Configurazione (vedi sezione [Utenti](#)) hanno la possibilità di configurare il servizio di auditing, al fine di stabilire cosa tracciare, con che formato e con che livello di dettaglio.

L'accesso alla funzionalità di configurazione del servizio di auditing avviene tramite la voce *Auditing* nella sezione *Configurazione* del menu laterale sinistro.

Se la maschera si presenta come in Fig. 8.107 il servizio di auditing è disabilitato e quindi nessun dato verrà tracciato.



Figure8.107: Servizio di auditing disabilitato

Modificando lo *Stato* del servizio di auditing in **Abilitato** appariranno ulteriori campi nel form (vedi Fig. 8.108) per effettuare le impostazioni.

La configurazione del servizio di auditing avviene tramite la creazione di una lista di **Filtri**, ciascuno dei quali stabilisce

Configurazione > **Auditing**

Auditing

Stato audit: abilitato

Comportamento di Default

Audit: abilitato

Dump: disabilitato

Formato dump: JSON

Log4j Auditing: abilitato

Filtri

visualizza(0)

Invia **Cancella**

Figure8.108: Servizio di auditing abilitato

un criterio per stabilire se una data informazione deve o non deve essere tracciata. Alle informazioni cui non si applica nessuno dei filtri definiti, viene applicato il comportamento di default, i cui parametri sono presenti nella schermata principale del servizio. Facendo riferimento alla Fig. 8.108 vediamo quali sono i parametri per specificare il comportamento di default:

- **Audit** (abilitato/disabilitato): Se abilitato, tutte le informazioni, cui non risulta applicabile nessuno dei filtri impostati, verranno tracciate dal servizio di auditing.
- **Dump** (abilitato/disabilitato): Questo campo viene preso in considerazione quando *Audit = abilitato*. Stabilisce, nei casi in cui non si applica nessun filtro, se oltre a tracciare i campi che descrivono l'operazione, devono essere tracciate anche le strutture dati coinvolte.
- **Formato Dump** (JSON/XML): Stabilisce il formato in cui vengono memorizzate le strutture dati di cui si è scelto di effettuare il dump. Le opzioni possibili sono tra il formato standard JSON (<http://www.json.org>) e la sua rappresentazione in formato XML.
- **Log4J Auditing** (abilitato/disabilitato): Questa opzione consente di abilitare/disabilitare l'appender log4j relativo ai dati tracciati dal servizio di auditing.

Una volta stabilito il comportamento di default si potranno definire i filtri specifici. Per passare alla sezione di gestione dei filtri si seleziona *Visualizza* nella sezione Filtri. Nell'area di gestione filtri viene mostrata la lista dei filtri esistenti con la possibilità di modificare/cancellare gli esistenti o inserirne di nuovi. Si può aggiungere un nuovo filtro premendo il pulsante *Aggiungi*. In Fig. 8.109 è mostrata la maschera per la creazione di un nuovo filtro di auditing.

Configurazione > Auditing > Filtri > **Aggiungi**

Filtro Generico

Utente:

Tipo operazione:

Tipo oggetto:

Stato operazione:

Filtro per Contenuto

Stato:

Azione

Stato:

Dump:

Invia **Cancella**

Figure8.109: Creazione di un filtro per il servizio di auditing

Facendo riferimento alla Fig. 8.109 vediamo in dettaglio il significato dei campi di un filtro:

- *Filtro Generico*

- **Utente:** è possibile specificare in questo campo uno username relativo ad un utente della govwayConsole del quale si vogliono tracciare le operazioni effettuate. Lasciare il campo di testo vuoto equivale a *Qualsiasi Utente*
- **Tipo Operazione** (ADD/CHANGE/DEL): Specifica il tipo di operazione che si vuole tracciare distinguendo tra operazioni di creazione, modifica e cancellazione. Lasciare il campo vuoto equivale a *Qualsiasi Tipo*.
- **Tipo Oggetto:** Questo campo è costituito da una lista contenente tutte le entità gestibili tramite l'interfaccia govwayConsole (ad esempio: Accordo di Servizio, Porta Delegata, ecc). Consente di restringere il tracciamento alle sole operazioni riguardanti una determinata entità. Lasciare il campo vuoto equivale a *Qualsiasi Tipo Oggetto*.
- **Stato Operazione** (requesting/error/completed): Consente di restringere le operazioni da tracciare in base al loro stato:
 - * *requesting*: indica un'operazione in fase di richiesta e non ancora completata
 - * *error*: Indica un'operazione completata che ha restituito un errore
 - * *completed*: Indica un'operazione che è terminata correttamenteLasciare il campo vuoto equivale a *Qualsiasi Stato Operazione*.

- *Filtro per contenuto*

- **Stato** (abilitato/disabilitato): Opzione che consente di abilitare il filtro basato sul contenuto degli oggetti coinvolti nell'operazione. Se l'opzione viene abilitata compariranno i 2 campi descritti ai passi successivi.
 - **Tipo** (normale/espressioneRegolare): Describe se la stringa riportata nel campo Dump deve essere interpretata come pattern o come espressione regolare.
 - **Dump:** Campo di testo per inserire il pattern (o espressione regolare) sulla base del quale verranno filtrate le operazioni. Il sistema di auditing tracerà soltanto le operazioni che coinvolgeranno entità il cui contenuto corrisponde alla stringa specificata.
- **Azione:** indica quale azione deve essere effettuata al verificarsi delle condizioni del filtro
 - **Stato** (abilitato/disabilitato): Se abilitato, al verificarsi delle condizioni impostate nel filtro, i dati dell'operazione verranno tracciati.
 - **Dump** (abilitato/disabilitato): Se *Stato = abilitato* è possibile specificare se si deve effettuare anche il dump delle entità coinvolte nell'operazione. Ad esempio, se viene tracciata un'operazione di modifica di un Accordo di Servizio, si decide se si vuole effettuare anche il dump dell'Accordo di Servizio oggetto della modifica.

Errori generati da GovWay

La gestione dei casi di errore, nelle comunicazioni mediate da un Gateway, deve tener conto di ulteriori casi di errore che possono presentarsi rispetto al dialogo diretto tra gli applicativi. Oltre agli errori già previsti nelle interfacce dell'API, gli applicativi client possono pertanto ricevere due tipi di errori generati direttamente da GovWay:

- *Errori Client*: identificabili da un codice http 4xx su API REST o da un fault code “Client” su API SOAP. Indicano che GovWay ha rilevato problemi nella richiesta effettuata dal client (es. errore autenticazione, autorizzazione, validazione contenuti...).
- *Errori Server*: identificabili dai codici http 502, 503 e 504 per le API REST o da un fault code “Server” generato dal Gateway e restituito con codice http 500 per le API SOAP.

La codifica degli errori prodotta dal Gateway permette alle applicazioni client di discriminare tra errori causati da una richiesta errata, per i quali è quindi necessario intervenire sull'applicazione client prima di effettuare nuovi invii, ed errori dovuti allo stato dei servizi invocati, per i quali è invece possibile continuare ad effettuare la richiesta. Maggiori dettagli sulla logica di re-invio delle richieste vengono riportati nella sezione *Classificazione degli Errori*.

Per ciascun errore GovWay riporta le seguenti informazioni:

- Un codice http su API REST o un fault code su API SOAP come descritto in precedenza.
- Un codice di errore, indicato nell'header http “GovWay-Transaction-ErrorType”, che riporta l'errore rilevato dal gateway (es. AuthenticationRequired, TokenExpired, InvalidRequestContent...).
- Un identificativo di transazione, indicato nell'header http “GovWay-Transaction-ID”, che identifica la transazione in errore, utile principalmente per indagini diagnostiche.
- Un payload http, contenente maggiori dettagli sull'errore, opportunamente codificato per API REST (*REST Problem Details - RFC 7807*) o SOAP (*SOAP Fault*).

Nota: Il codice di errore e l'identificativo di transazione vengono riportati sia tramite header http che all'interno del payload.

Di seguito viene riportato un esempio di errore generato in seguito al rilevamento di una richiesta non conforme all'interfaccia API REST:

```
HTTP/1.1 400 Bad Request
Server: GovWay
Transfer-Encoding: chunked
GovWay-Transaction-ErrorType: InvalidRequestContent
GovWay-Transaction-ID: b76b4d1b-cd9d-43a0-bea2-1f352f1e71dd
Content-Type: application/problem+json
Date: Thu, 28 May 2020 15:59:14 GMT

{
  "type": "https://govway.org/handling-errors/400/InvalidRequestContent.html",
  "title": "InvalidRequestContent",
  "status": 400,
  "detail": "Request content not conform to API specification",
  "govway_id": "b76b4d1b-cd9d-43a0-bea2-1f352f1e71dd"
}
```

Lo stesso tipo di errore, rilevato per una API SOAP, viene riportato di seguito:

```
HTTP/1.1 500 Internal Server Error
Server: GovWay
Transfer-Encoding: chunked
GovWay-Transaction-ErrorType: InvalidRequestContent
GovWay-Transaction-ID: b76b4d1b-cd9d-43a0-bea2-1f352f1e71dd
Content-Type: text/xml
Date: Thu, 28 May 2020 15:59:14 GMT

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client.InvalidRequestContent</faultcode>
      <faultstring>Received request is not conform to API specification</faultstring>
      <faultactor>http://govway.org/integration</faultactor>
      <detail>
        <problem xmlns="urn:ietf:rfc:7807">
          <type>https://govway.org/handling-errors/400/InvalidRequestContent.html</
          type>
          <title>InvalidRequestContent</title>
          <status>400</status>
          <detail>Request content not conform to API specification</detail>
          <govway_id>9876b03e-0377-4a02-9fb8-07094b0cdf06</govway_id>
        </problem>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

9.1 Classificazione degli Errori

Una risposta con codice 2xx indica che l'operazione ha avuto successo mentre codici diversi indicano un problema imputabile al client (4xx su API REST o da un fault code “Client” su API SOAP) o un errore dipendente dallo stato del servizio (5xx su API REST o da un fault code “Server” su API SOAP).

La tabella [Tabella 9.1](#) riporta l'elenco dei possibili codici di errore restituiti da GovWay. Per ognuno di questi, nella colonna “Retry” è indicato se sia possibile o meno effettuare nuovi invii della stessa richiesta che ha ottenuto errore. Le indicazioni fornite sono le seguenti:

- Sì: il client può effettuare nuovamente la stessa richiesta;
- Sì, se idempotente: il client può effettuare nuovamente la stessa richiesta, ma solo se l'operazione sul backend applicativo è implementata in maniera idempotente.
- No: il client deve risolvere il problema segnalato prima di effettuare una nuova richiesta (ripetere la stessa richiesta produrrebbe sempre lo stesso esito).

Table9.1: Gestione degli Errori

REST / SOAP	GovWay-Transaction-ErrorType	Retry
400 / Client	Errori 400 (Bad Request)	No
401 / Client	Errori 401 (Authentication Error)	No
403 / Client	Errori 403 (Authorization Deny)	No
404 / Client	Errori 404 (NotFound)	No
409 / Client	Errori 409 (Conflict)	No
413 / Client	Errori 413 (Payload Too Large)	No
429 / Client	LimitExceeded - 429 (Rate Limiting)	No, fino a reset *
429 / Client	TooManyRequests - 429 (Rate Limiting)	Sì
502 / Server	Errori 502 (Bad Gateway)	Sì, se idempotente
502 / Server	ResponseSizeExceeded - 502 (Bad Gateway)	No
503 / Server	Errori 503 (Service Unavailable)	Sì
504 / Server	Errori 504 (Endpoint Request Timed-out)	Sì, se idempotente

[*] Se vengono attivate policy di *Rate Limiting* che prevedono un limite di richieste all'interno di una finestra temporale, GovWay genera un header HTTP che indica al client il numero di secondi che mancano alla nuova finestra temporale dove saranno resettati i contatori delle richieste effettuate. I nomi degli header, che cambiano in funzione delle policy attivate, vengono descritte nella sezione [Informazioni restituite dal gateway nella risposta all'applicativo client](#).

Nei casi in cui è prevista la rispedizione, GovWay genera un header “Retry-After” che indica al client il numero di secondi di attesa prima di ripetere la richiesta.

9.1.1 Errori 400 (Bad Request)

In questa sezione vengono riportati tutti i possibili codici di errore generati da GovWay relativi ad una richiesta client malformata.

Nella configurazione di default di GovWay, le casistiche di errore “AttachmentsRequestFailed”, “MessageSecurityRequestFailed”, “InteroperabilityRequestManagementFailed”, “TransformationRuleRequestFailed” e “ConnectorNotFound” sono tutte restituiti al client con il solo codice di errore [BadRequest](#). La scelta è finalizzata ad evitare disclosure di informazioni relative al domino interno.

È possibile abilitare temporaneamente la generazione dei codici puntuali accendendo alla voce “Strumenti - Runtime” della console di gestione e selezionando “Errore Puntuale” per la “Richiesta” nella sezione «Errori generati dal Gateway - Codici di errore “GovWay-Transaction-ErrorType”» ([Fig. 9.1](#)).



Figure9.1: Attivazione temporanea degli errori specifici 400 (Bad Request)

L'abilitazione permanente può essere invece effettuata disabilitando la seguente proprietà sul file di proprietà esterno /etc/govway/errori_local.properties:

```
# Gateway non in grado di gestire la richiesta: AttachmentsRequestFailed, ↵
↳ MessageSecurityRequestFailed, InteroperabilityRequestManagementFailed, ↵
↳ TransformationRuleRequestFailed, ConnectorNotFound
WRAP_400_INTERNAL_BAD_REQUEST.enabled=false
```

ContentTypeNotProvided

GovWay ha rilevato una richiesta verso una API SOAP che non possiede un header http “Content-Type”.

ContentTypeNotSupported

GovWay ha rilevato una richiesta verso una API SOAP che possiede un header http “Content-Type” non supportato.

Il valore supportato per SOAP 1.1 è “text/xml” mentre per SOAP 1.2 è “application/soap+xml”. Sono supportati anche i formati multipart “SOAP With Attachments” (Multipart/Related; type=text/xml; boundary=...) e MTOM (Multipart/Related; type=>application/xop+xml;>; start-info=>text/xml;>; boundary=...).

SapMustUnderstandUnknown

GovWay ha rilevato una richiesta verso una API SOAP che contiene un SOAP Header con attributo “mustUnderstand” e senza un actor/role definito che risulta sconosciuto a GovWay.

Nota: L'errore viene generato solamente se GovWay è stato configurato per riconoscere e trattare solamente alcuni SOAP Header specifici. La configurazione di default di govway è di far passare tutti i SOAP Header; per modificarla agire sul file di proprietà esterno /etc/govway/govway_local.properties aggiungendo le seguenti proprietà

```
# Possibili valori: true/false
org.openspcoop2.pdd.services.BypassMustUnderstandHandler.allHeaders=false

# Sintassi per filtri specifici:
# org.openspcoop2.pdd.services.BypassMustUnderstandHandler.header.LOCAL_
↳ NAME=NAMESPACE_URI
# Se si deve definire più header con stesso local name e differente namespace,
↳ si può utilizzare la seguente sintassi:
# org.openspcoop2.pdd.services.BypassMustUnderstandHandler.header.LOCAL_NAME!
```

(continues on next page)

(continua dalla pagina precedente)

```

↳ NUMERO_PROGRESSIVO=NAMESPACE_URI
# Esempio per Bypass per WS-Security:
#org.openspcoop2.pdd.services.BypassMustUnderstandHandler.header.
↳ Security=http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
↳ secext-1.0.xsd
# Esempio per Bypass per WS-Reliability
#org.openspcoop2.pdd.services.BypassMustUnderstandHandler.header.
↳ Sequence=http://schemas.xmlsoap.org/ws/2005/02/rm

```

SoapVersionMismatch

La versione SOAP rilevata differisce tra quella indicata nell'header http "Content-Type" e il namespace dell'Envelope.

UnprocessableRequestContent

GovWay ha rilevato un payload differente da quello indicato nell'header http "Content-Type".

RequestReadTimeout

Rilevato errore "Read Timed Out" durante la lettura della richiesta.

NotSupportedByProtocol

L'errore viene sollevato quando la richiesta non è compatibile con il Profilo di Interoperabilità e/o il protocollo (SOAP/REST) a cui appartiene l'API invocata su GovWay.

CorrelationInformationNotFound

L'errore indica che nella richiesta non è stato possibile per GovWay estrarre il *Riferimento ID Richiesta* da utilizzare per effettuare una correlazione asincrona tra operazioni differenti.

La correlazione è attivabile su una API tramite la funzionalità descritta nella sezione *Correlazione tra transazioni differenti*.

Nota: In mancanza di un *Riferimento ID Richiesta*, GovWay per default non solleva alcun errore. È possibile forzare la generazione dell'errore intervenendo sul file di proprietà esterno /etc/govway/trasparente_local.properties aggiungendo le seguenti proprietà

```

# Fruizioni
org.openspcoop2.protocol.trasparente.pd.riferimentoIdRichiesta.required=true
# Erogazioni
org.openspcoop2.protocol.trasparente.pa.riferimentoIdRichiesta.required=true

```

L'errore viene anche sollevato se GovWay non rileva il riferimento alla richiesta nelle collaborazioni asincrone del Profilo di Interoperabilità SPCoop. Per maggiori dettagli si rimanda alla sezione *Profili Asincroni*.

ApplicationCorrelationIdentificationRequestFailed

La funzionalità di correlazione applicativa, abilitata sull'API invocata, non è riuscita ad estrarre l'informazione richiesta.

Maggiori dettagli sulla funzionalità sono descritti nella sezione [Correlazione Applicativa](#).

InvalidRequestContent

La funzionalità di validazione dei contenuti applicativi, abilitata sull'API invocata, ha rilevato un contenuto della richiesta non conforme alla specifica dell'API.

Maggiori dettagli sulla funzionalità sono descritti nella sezione [Validazione dei messaggi](#).

UnexpectedInteroperabilityHeader

GovWay ha rilevato, in una fruizione di API, una richiesta già contenente l'header di interoperabilità previsto dal profilo.

L'errore viene generato solo se l'API appartiene ad un Profilo di Interoperabilità differente dal profilo "API Gateway". Per maggiori dettagli sui profili di interoperabilità è possibile consultare le sezioni:

- [Profilo "ModI"](#)
- [Profilo "eDelivery"](#)
- [Profilo "SPCoop"](#)
- [Profilo "eDelivery"](#)

InteroperabilityInvalidRequest

L'errore segnala che GovWay ha rilevato una richiesta non conforme al Profilo di Interoperabilità a cui appartiene l'API invocata.

Per maggiori dettagli sui profili di interoperabilità è possibile consultare le sezioni:

- [Profilo "ModI"](#)
- [Profilo "eDelivery"](#)
- [Profilo "SPCoop"](#)
- [Profilo "eDelivery"](#)

AttachmentsRequestFailed

GovWay ha rilevato un payload della richiesta non utilizzabile con la gestione degli attachments abilitata sull'API.

Maggiori dettagli sulla funzionalità di gestione degli attachments sono presenti nella sezione [MTOM](#).

MessageSecurityRequestFailed

GovWay ha rilevato un payload della richiesta non utilizzabile con la gestione della sicurezza messaggio abilitata sull'API.

Maggiori dettagli sulla funzionalità di sicurezza messaggio sono presenti nella sezione *Sicurezza a livello del messaggio*.

InteroperabilityRequestManagementFailed

L'errore segnala che GovWay non è riuscito a completare la generazione di un header conforme al Profilo di Interoperabilità a cui appartiene l'API invocata.

Per maggiori dettagli sui profili di interoperabilità è possibile consultare le sezioni:

- *Profilo “ModI”*
- *Profilo “eDelivery”*
- *Profilo “SPCoop”*
- *Profilo “eDelivery”*

TransformationRuleRequestFailed

GovWay ha rilevato un payload della richiesta non utilizzabile con la funzionalità di trasformazione attivata sull'API.

Maggiori dettagli sulla funzionalità di trasformazione sono presenti nella sezione *Trasformazioni*.

ConnectorNotFound

Non è stato possibile individuare il connettore che implementa l'API tra quelli associati all'erogazione.

Maggiori dettagli sulla funzionalità che consente di individuare il connettore da utilizzare, rispetto ai parametri della richiesta, sono presenti nella sezione *Consegna Condizionale*.

BadRequest

L'errore segnala che la richiesta verso l'API invocata è malformata.

9.1.2 Errori 401 (Authentication Error)

In questa sezione vengono riportati tutti i possibili codici di errore generati da GovWay relativi a autenticazione fallita. Rientrano in questa casistica gli errori avvenuti durante le fasi di autenticazione degli applicativi (Sezione *Autenticazione Trasporto*) e di verifica del token OAuth (Sezione *Autenticazione Token*).

AuthenticationRequired

La richiesta non possiede le credenziali relative all'autenticazione configurata sul Controllo degli Accessi dell'API.

Maggiori dettagli sulla funzionalità di autenticazione sono descritti nella sezione [Autenticazione Trasporto](#).

AuthenticationFailed

La richiesta possiede delle credenziali non valide relative all'autenticazione configurata sul Controllo degli Accessi dell'API.

Maggiori dettagli sulla funzionalità di autenticazione sono descritti nella sezione [Autenticazione Trasporto](#).

ProxyAuthenticationRequired

La richiesta non possiede le credenziali necessarie per poter autenticare il frontend su cui è stata effettuata l'autenticazione degli applicativi chiamanti.

Scenario in cui si presenta l'errore

Nel caso in cui la terminazione ssl viene gestita su un frontend http (Apache httpd, IIS, etc) GovWay necessita di ricevere le credenziali per attuare il processo di autenticazione descritto nella sezione [Autenticazione Trasporto](#). Nel caso di utilizzo di una integrazione “mod_jk” tra frontend e application server, GovWay riceve i certificati gestiti sul frontend http in maniera trasparente. Negli altri casi invece deve essere configurato opportunamente il frontend http per inoltrare i certificati client o il DN attraverso header HTTP a GovWay.

Nell'ambito di tale configurazione è possibile abilitare l'autenticazione del frontend in modo da accettare gli header http contenenti le credenziali solamente da un frontend autenticato.

Maggiori dettagli sulla funzionalità sono descritti nella sezione `install_ssl_server_frontend` della Guida di Installazione.

ProxyAuthenticationFailed

La richiesta possiede delle credenziali non valide per poter autenticare il frontend su cui è stata effettuata l'autenticazione degli applicativi chiamanti.

Scenario in cui si presenta l'errore

Nel caso in cui la terminazione ssl viene gestita su un frontend http (Apache httpd, IIS, etc) GovWay necessita di ricevere le credenziali per attuare il processo di autenticazione descritto nella sezione [Autenticazione Trasporto](#). Nel caso di utilizzo di una integrazione “mod_jk” tra frontend e application server, GovWay riceve i certificati gestiti sul frontend http in maniera trasparente. Negli altri casi invece deve essere configurato opportunamente il frontend http per inoltrare i certificati client o il DN attraverso header HTTP a GovWay.

Nell'ambito di tale configurazione è possibile abilitare l'autenticazione del frontend in modo da accettare gli header http contenenti le credenziali solamente da un frontend autenticato.

Maggiori dettagli sulla funzionalità sono descritti nella sezione `install_ssl_server_frontend` della Guida di Installazione.

ForwardProxyAuthenticationRequired

Il frontend che ha effettuato l'autenticazione degli applicativi non ha inoltrato a GovWay le credenziali nell'header http concordato.

Scenario in cui si presenta l'errore

Nel caso in cui la terminazione ssl viene gestita su un frontend http (Apache httpd, IIS, etc) GovWay necessita di ricevere le credenziali per attuare il processo di autenticazione descritto nella sezione [Autenticazione Trasporto](#). Nel caso di utilizzo di una integrazione “mod_jk” tra frontend e application server, GovWay riceve i certificati gestiti sul frontend http in maniera trasparente. Negli altri casi invece deve essere configurato opportunamente il frontend http per inoltrare i certificati client o il DN attraverso header HTTP a GovWay.

Maggiori dettagli sulla funzionalità sono descritti nella sezione `install_ssl_server_frontend` della Guida di Installazione.

TokenAuthenticationRequired

La richiesta non possiede un token *oAuth2*. Il token viene richiesto dall'autenticazione configurata sul Controllo degli Accessi dell'API.

Maggiori dettagli sulla funzionalità sono descritti nella sezione [Autenticazione Token](#).

TokenAuthenticationFailed

La richiesta possiede un token *oAuth2* non valido rispetto all'autenticazione configurata sul Controllo degli Accessi dell'API.

Maggiori dettagli sulla funzionalità sono descritti nella sezione [Autenticazione Token](#).

TokenExpired

La richiesta possiede un token *oAuth2* scaduto.

Maggiori dettagli sulla funzionalità configurata nel Controllo degli Accessi dell'API sono descritti nella sezione [Autenticazione Token](#).

TokenNotBefore

La richiesta possiede un token *oAuth2* non ancora utilizzabile. Nel claim “notBefore” è presente una data futura.

Maggiori dettagli sulla funzionalità “Autenticazione Token” configurata nel Controllo degli Accessi dell'API sono descritti nella sezione [Autenticazione Token](#).

TokenInTheFuture

La richiesta possiede un token *oAuth2* nel quale il claim “iat” possiede una data futura.

Maggiori dettagli sulla funzionalità “Autenticazione Token” configurata nel Controllo degli Accessi dell'API sono descritti nella sezione [Autenticazione Token](#).

TokenRequiredClaimsNotFound

Il token *oAuth2* presente nella richiesta non contiene tutti i claim configurati come obbligatori nel Controllo degli Accessi dell'API.

Maggiori dettagli sulla funzionalità sono descritti nella sezione [Autenticazione Token](#).

Authentication

La richiesta non soddisfa l'autenticazione indicata nel Controllo degli Accessi dell'API ([Autenticazione Trasporto](#)).

9.1.3 Errori 403 (Authorization Deny)

In questa sezione vengono riportati tutti i possibili codici di errore generati da GovWay, relativi ad autorizzazione negata.

AuthorizationContentDeny

La richiesta non soddisfa i criteri di autorizzazione del contenuto attivati sul Controllo degli Accessi dell'API.

Maggiori dettagli sulla funzionalità di autorizzazione dei contenuti sono descritti nella sezione [Autorizzazione Contenuti](#).

AuthorizationContentPolicyDeny

La richiesta non soddisfa i criteri di autorizzazione del contenuto attivati, sul Controllo degli Accessi dell'API, tramite una policy XAML (o un template).

Maggiori dettagli sulla funzionalità di autorizzazione dei contenuti sono descritti nella sezione [Autorizzazione Contenuti](#).

AuthorizationDeny

La richiesta non soddisfa i criteri di autorizzazione per richiedente, attivati sul Controllo degli Accessi dell'API.

Maggiori dettagli sulla funzionalità di autorizzazione per richiedente sono descritti nella sezione [Autorizzazione](#).

AuthorizationPolicyDeny

La richiesta non soddisfa i criteri di autorizzazione definiti nella policy XAML (o in un template), attivati sul Controllo degli Accessi dell'API.

Maggiori dettagli sulla funzionalità di autorizzazione per policy sono descritti nella sezione [XACML-Policy](#).

AuthorizationTokenDeny

La richiesta non soddisfa i criteri di autorizzazione, relativi ai claims presenti nel token *oAuth2*, attivati sul Controllo degli Accessi dell'API. Maggiori dettagli sulla funzionalità di autorizzazione per token claims sono descritti nella sezione [Autorizzazione per Token Claims](#).

AuthorizationMissingScope

Il token *oAuth2* presente nella richiesta non contiene tutti gli scope richiesti sul Controllo degli Accessi dell'API. Maggiori dettagli sulla funzionalità di autorizzazione per scope sono descritti nella sezione [Autorizzazione](#).

AuthorizationMissingRole

La richiesta non soddisfa i criteri di autorizzazione per ruolo, attivati sul Controllo degli Accessi dell'API. Maggiori dettagli sulla funzionalità di autorizzazione per ruolo sono descritti nella sezione [Autorizzazione](#).

Authorization

La richiesta non soddisfa i criteri di autorizzazione attivati sul Controllo degli Accessi dell'API. Maggiori dettagli sulla funzionalità di autorizzazione sono descritti nella sezione [Autorizzazione](#).

9.1.4 Errori 404 (NotFound)

In questa sezione vengono riportati tutti i possibili codici di errore generati da GovWay relativi a richieste verso API o risorse inesistenti.

UndefinedOperation

L'operazione richiesta non risulta associata all'API registrata su GovWay.

Maggiori dettagli sulla registrazione di una API su GovWay sono descritti nella sezione [Definizione delle API](#).

UnknownAPI

L'API richiesta non risulta esistere su GovWay.

Maggiori dettagli sulla registrazione di una API su GovWay sono descritti nella sezione [Definizione delle API](#).

NotFound

Il contenuto della richiesta non indirizza una API esistente su GovWay.

Questo tipo di errore avviene nei Profili di Interoperabilità per i quali l'API non viene indirizzata nella URL ma all'interno del contenuto della richiesta (es. [Profilo "SPCoop"](#)).

9.1.5 Errori 409 (Conflict)

In questa sezione vengono riportati tutti i possibili codici di errore generati da GovWay relativi a richieste già processate.

ConflictInQueue

La richiesta risulta già in elaborazione su GovWay.

Questo tipo di errore avviene nei Profili di Interoperabilità che richiedono un filtro dei duplicati. Un identificativo univoco viene associato al messaggio all'interno dell'header previsto dal profilo di interoperabilità (es. *Profilo "ModI"*).

Conflict

La richiesta risulta già stata elaborata su GovWay.

Questo tipo di errore avviene nei Profili di Interoperabilità che richiedono un filtro dei duplicati. Un identificativo univoco viene associato al messaggio all'interno dell'header previsto dal profilo di interoperabilità (es. *Profilo "ModI"*).

9.1.6 Errori 413 (Payload Too Large)

In questa sezione vengono riportati i possibili codici di errore generati da GovWay relativi a richieste che non vengono processate poichè possiedono un payload più grande del livello di soglia impostato tramite le policy di *Rate Limiting* definite utilizzando la metrica “Dimensione Massima Messaggio”.

RequestSizeExceeded

La richiesta non viene processata poichè possiede un payload più grande del livello di soglia impostato tramite la policy di *Rate Limiting* definita utilizzando la metrica “Dimensione Massima Messaggio”.

9.1.7 Errori 429 (Rate Limiting)

In questa sezione vengono riportati tutti i possibili codici di errore generati da GovWay relativi alle funzionalità di Rate Limiting.

LimitExceeded

L'errore segnala che è stato superato il numero massimo di richieste (o di banda) nell'intervallo temporale configurato sulla policy di Rate Limiting dell'API invocata.

Maggiori informazioni sul Rate Limiting sono consultabili nella sezione *Rate Limiting*.

TooManyRequests

L'errore segnala che è stato superato il numero totale massimo di richieste simultaneemesse sull'API invocata.

Maggiori informazioni sul Rate Limiting sono consultabili nella sezione [Rate Limiting](#).

9.1.8 Errori 502 (Bad Gateway)

In questa sezione vengono riportati tutti i possibili codici di errore generati da GovWay relativi a errori emersi durante la gestione della risposta.

Nella configurazione di default di GovWay, gli errori descritti in questa sezione, con l'eccezione del codice «ResponseSizeExceeded», sono tutti restituiti al client con il solo codice di errore [InvalidResponse](#). La scelta è finalizzata ad evitare disclosure di informazioni relative al domino interno.

È possibile abilitare temporaneamente la generazione dei codici puntuali accendendo alla voce “Strumenti - Runtime” della console di gestione e selezionando “Errore Puntuale” per la “Risposta” nella sezione «Errori generati dal Gateway - Codici di errore “GovWay-Transaction-ErrorType”» (Fig. 9.2).

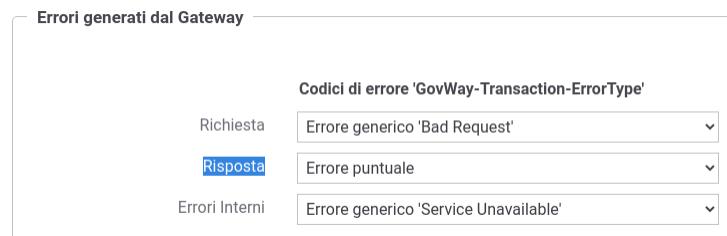


Figure9.2: Attivazione temporanea degli errori specifici 502 (Bad Gateway)

L'abilitazione permanente può essere invece effettuata disabilitando le seguenti proprietà sul file di proprietà esterno /etc/govway/errori_local.properties:

```
WRAP_502_BAD_RESPONSE.enabled=false
WRAP_502_INTERNAL_RESPONSE_ERROR.enabled=false
```

InvalidResponse

Risposta non valida ricevuta dal backend che implementa l'API.

ResponseSizeExceeded

La risposta non viene processata poichè possiede un payload più grande del livello di soglia impostato tramite la policy di [Rate Limiting](#) definita utilizzando la metrica “Dimensione Massima Messaggio”.

UnprocessableResponseContent

GovWay ha rilevato un payload differente da quello indicato nell'header http “Content-Type” della risposta.

AttachmentsResponseFailed

GovWay ha rilevato un payload della risposta non utilizzabile con la gestione degli attachments abilitata sull'API.

Maggiori dettagli sulla funzionalità di gestione degli attachments sono presenti nella sezione [*MTOM*](#).

ApplicationCorrelationIdentificationResponseFailed

La funzionalità di correlazione applicativa, abilitata sull'API invocata, non è riuscita ad estrarre l'informazione richiesta dalla risposta.

Maggiori dettagli sulla funzionalità sono descritti nella sezione [*Correlazione Applicativa*](#).

MessageSecurityResponseFailed

GovWay ha rilevato un payload della risposta non utilizzabile con la gestione della sicurezza messaggio abilitata sull'API.

Maggiori dettagli sulla funzionalità di sicurezza messaggio sono presenti nella sezione [*Sicurezza a livello del messaggio*](#).

InvalidResponseContent

La funzionalità di validazione dei contenuti applicativi, abilitata sull'API invocata, ha rilevato un contenuto della risposta non conforme alla specifica dell'API.

Maggiori dettagli sulla funzionalità sono descritti nella sezione [*Validazione dei messaggi*](#).

InteroperabilityResponseManagementFailed

L'errore segnala che GovWay non è riuscito a completare la generazione di un header conforme al Profilo di Interoperabilità a cui appartiene l'API invocata.

Per maggiori dettagli sui profili di interoperabilità è possibile consultare le sezioni:

- [*Profilo “ModI”*](#)
- [*Profilo “eDelivery”*](#)
- [*Profilo “SPCoop”*](#)
- [*Profilo “eDelivery”*](#)

InteroperabilityInvalidResponse

L'errore segnala che GovWay ha rilevato una risposta non conforme al Profilo di Interoperabilità a cui appartiene l'API invocata.

Per maggiori dettagli sui profili di interoperabilità è possibile consultare le sezioni:

- *Profilo “ModI”*
- *Profilo “eDelivery”*
- *Profilo “SPCoop”*
- *Profilo “eDelivery”*

UnexpectedInteroperabilityResponseHeader

GovWay ha rilevato, in una erogazione di API, una risposta già contenente l'header di interoperabilità previsto dal profilo.

L'errore viene generato solo se l'API appartiene ad un Profilo di Interoperabilità differente dal profilo “API Gateway”. Per maggiori dettagli sui profili di interoperabilità è possibile consultare le sezioni:

- *Profilo “ModI”*
- *Profilo “eDelivery”*
- *Profilo “SPCoop”*
- *Profilo “eDelivery”*

InteroperabilityResponseError

L'errore segnala la ricezione di una risposta, conforme al Profilo di Interoperabilità, che segnala degli errori rilevati dalla controparte.

Per maggiori dettagli sui profili di interoperabilità è possibile consultare le sezioni:

- *Profilo “ModI”*
- *Profilo “eDelivery”*
- *Profilo “SPCoop”*
- *Profilo “eDelivery”*

TransformationRuleResponseFailed

GovWay ha rilevato un payload della risposta non utilizzabile con la funzionalità di trasformazione attivata sull'API.

Maggiori dettagli sulla funzionalità di trasformazione sono presenti nella sezione *Trasformazioni*.

ExpectedResponseNotReceived

Una risposta non è presente nel payload ritornato dal backend che implementa l'API.

ConflictResponse

La risposta risulta già stata elaborata su GovWay.

Questo tipo di errore avviene nei Profili di Interoperabilità che richiedono un filtro dei duplicati. Un identificativo univoco viene associato al messaggio all'interno dell'header previsto dal profilo di interoperabilità (es. *Profilo "ModI"*).

BadResponse

Risposta non valida ricevuta dal backend che implementa l'API.

GatewayError

Il gateway non è momentaneamente in grado di gestire la risposta.

9.1.9 Errori 503 (Service Unavailable)

In questa sezione vengono riportati tutti i possibili codici di errore generati da GovWay relativi ad errori emersi durante la gestione della richiesta. Gli errori sono classificabili in:

- indisponibilità temporanea del backend che implementa l'API; l'errore viene identificato dal codice di errore “APIUnavailable”;
- accesso all'API sospeso su GovWay; l'errore viene identificato dal codice di errore “APISuspended”;
- Il gateway non è al momento correttamente operativo; rientrano in questa casistica i codici di errore: “GatewayInactive”, “GatewayUnavailable”, “GatewayError”.

Nella configurazione di default di GovWay, gli errori che indicano una indisponibilità temporanea del gateway sono tutti restituiti al client con il solo codice di errore *APIUnavailable*. La scelta è finalizzata ad evitare disclosure di informazioni relative al domino interno.

È possibile abilitare temporaneamente la generazione dei codici puntuali accendendo alla voce “Strumenti - Runtime” della console di gestione e selezionando “Errore Puntuale” per gli “Errori Interni” nella sezione «Errori generati dal Gateway - Codici di errore “GovWay-Transaction-ErrorType”» (Fig. 9.3).



Figure9.3: Attivazione temporanea degli errori specifici 503 (Service Unavailable)

L'abilitazione permanente può essere invece effettuata disabilitando la seguente proprietà sul file di proprietà esterno /etc/govway/errori_local.properties:

```
# Gateway momentaneamente indisponibile: GatewayInactive, GatewayUnavailable, ↵
˓→GatewayError
WRAP_503_INTERNAL_ERROR.enabled=false
```

APIUnavailable

GovWay ha rilevato una indisponibilità temporanea del backend che implementa l'API.

È possibile utilizzare la funzionalità descritta nella sezione *Verifica Connattività Connettore* per verificare puntualmente la connattività tramite la console di gestione.

APISuspended

L'API invocata risulta sospesa.

Maggiori dettagli sulla funzionalità di sospensione di una API vengono forniti nella sezione *Sospensione API*.

GatewayInactive

Il gateway non è attualmente disponibile.

GatewayUnavailable

Il gateway è temporaneamente non disponibile.

GatewayError

Il gateway non è momentaneamente in grado di gestire la richiesta.

9.1.10 Errori 504 (Endpoint Request Timed-out)

In questa sezione è presente il codice di errore generato da GovWay quando avviene un “Read Timed Out” durante l'invocazione del backend che implementa l'API.

EndpointReadTimeout

Rilevato errore “Read Timed Out” durante l'invocazione del backend che implementa l'API.

È possibile configurare il tempo di attesa di una risposta agendo sui parametri di configurazione del connettore descritti nella sezione *Tempi Risposta*.

9.2 REST Problem Details - RFC 7807

Negli errori generati da GovWay, relativi alla gestione di richieste verso API di tipo REST, il payload http ritornato al chiamante contiene un oggetto *Problem Details* come definito nella specifica *RFC 7807* (<https://tools.ietf.org/html/rfc7807>).

Gli elementi valorizzati sono i seguenti:

- *type*: riferisce una pagina della seguente documentazione (*Classificazione degli Errori*) che descrive l'errore.
- *title*: contiene un codice che specifica la problematica rilevata dal gateway (es. `AuthenticationRequired`, `TokenExpired`, `InvalidRequestContent`...). Tutti i codici di errore vengono descritti nella sezione *Classificazione degli Errori*.
- *status*: contiene il codice http ritornato al chiamante.
- *detail*: fornisce informazioni di dettaglio sull'errore avvenuto.
- *govway_id*: identificativo di transazione che permette di individuare la transazione terminata in errore tramite la Console di Monitoraggio.

Di seguito viene riportato un esempio:

```
{
  "type": "https://govway.org/handling-errors/400/InvalidRequestContent.html",
  "title": "InvalidRequestContent",
  "status": 400,
  "detail": "Request content not conform to API specification",
  "govway_id": "b76b4d1b-cd9d-43a0-bea2-1f352f1e71dd"
}
```

L'oggetto *Problem Details* generato dal Gateway possiede per default il formato *json*.

Viene utilizzato il formato *xml* (Appendice “A” del RFC 7807) solamente se la richiesta presenta anch’essa tale formato.

Nota: Un applicativo client può indicare al Gateway quale formato desidera attraverso l'header http *Accept*.

Di seguito viene riportato un esempio di oggetto *Problem Details* nel formato *xml*:

```
<problem xmlns="urn:ietf:rfc:7807">
  <type>https://govway.org/handling-errors/400/InvalidRequestContent.html</type>
  <title>InvalidRequestContent</title>
  <status>400</status>
  <detail>Request content not conform to API specification</detail>
  <govway_id>a1e047bf-3775-4f74-9492-dba972e7afb2</govway_id>
</problem>
```

Claim aggiuntivi

Se viene abilitata la generazione di un codice specifico di errore, come descritto nella sezione *Attivazione di Codici di Errore Specifici*, viene valorizzato anche il claim *govway_status*.

È inoltre possibile valorizzare il claim *instance* con l'identificativo dell'erogazione o della fruizione invocata seguendo le indicazioni descritte di seguito. Per default tale elemento non viene valorizzato.

È possibile abilitare temporaneamente la valorizzazione del claim *instance* accendendo alla voce “Strumenti - Runtime” della console di gestione e abilitando lo stato della sezione «Claim “instance” nei Problem “(Fig. 9.5)».



Figure9.4: Attivazione temporanea del claim *instance* nel Problem Detail

Una abilitazione permanente è invece attuabile agendo sul file di proprietà esterno /etc/govway/govway_local.properties abilitando la seguente proprietà:

```
org.openspcoop2.pdd.errori.instance=true
```

Di seguito viene riportato un esempio di errore generato in seguito al rilevamento di una richiesta non conforme all'interfaccia API REST, dove è stato abilitata sia la generazione di un codice di errore specifico che la valorizzazione dell'elemento *instance*:

```
{
  "type": "https://govway.org/handling-errors/400/InvalidRequestContent.html",
  "title": "InvalidRequestContent",
  "status": 400,
  "detail": "Request content not conform to API specification",
  "instance": "gw_ENTE/gw_api-monitor/v1",
  "govway_id": "b76b4d1b-cd9d-43a0-bea2-1f352f1e71dd",
  "govway_status": "integration:GOVWAY-418"
}
```

9.3 SOAP Fault

Negli errori generati da GovWay, relativi alla gestione di richieste verso API di tipo SOAP, il payload http ritornato al chiamante contiene un SOAP Fault.

Gli elementi del fault sono valorizzati come segue:

- *faultactor* (Soap 1.1) o *Role* (Soap 1.2) possiede il valore <http://govway.org/integration>.
- *faultcode* (Soap 1.1) o *Code/Subcode* (Soap 1.2): contiene uno standard SOAP fault code (Server/Client per Soap 1.1, Receiver/Sender per Soap 1.2) concatenato con un codice di errore di GovWay che specifica la problematica rilevata (es. AuthenticationRequired, TokenExpired, InvalidRequestContent ...). Tutti i codici di errore vengono descritti nella sezione [Classificazione degli Errori](#).
- *faultstring* (Soap 1.1) o *Reason* (Soap 1.2): fornisce informazioni di dettaglio sull'errore avvenuto.
- *detail*: è presente l'oggetto *Problem Details*, nella rappresentazione xml descritta nella sezione [REST Problem Details - RFC 7807](#).

Nota: Il formato di errore (Soap 1.1 o Soap 1.2) assume lo stesso formato della richiesta.

Di seguito viene riportato un esempio di errore rilevato per una API SOAP 1.1:

```
HTTP/1.1 500 Internal Server Error
Server: GovWay
Transfer-Encoding: chunked
GovWay-Transaction-ErrorType: InvalidRequestContent
```

(continues on next page)

(continua dalla pagina precedente)

GovWay-Transaction-ID: b76b4d1b-cd9d-43a0-bea2-1f352f1e71dd
 Content-Type: text/xml
 Date: Thu, 28 May 2020 15:59:14 GMT

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client.InvalidRequestContent</faultcode>
      <faultstring>Received request is not conform to API specification</faultstring>
      <faultactor>http://govway.org/integration</faultactor>
      <detail>
        <problem xmlns="urn:ietf:rfc:7807">
          <type>https://govway.org/handling-errors/400/InvalidRequestContent.html</
          type>
          <title>InvalidRequestContent</title>
          <status>400</status>
          <detail>Request content not conform to API specification</detail>
          <govway_id>9876b03e-0377-4a02-9fb8-07094b0cdf06</govway_id>
        </problem>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Lo stesso tipo di errore, rilevato per una API SOAP 1.2, viene riportato di seguito:

HTTP/1.1 500 Internal Server Error
 Server: GovWay
 Transfer-Encoding: chunked
 GovWay-Transaction-ErrorType: InvalidRequestContent
 GovWay-Transaction-ID: b76b4d1b-cd9d-43a0-bea2-1f352f1e71dd
 Content-Type: application/soap+xml
 Date: Thu, 28 May 2020 15:59:14 GMT

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value>env:Sender</env:Value>
        <env:Subcode>
          <env:Value xmlns:integration="http://govway.org/integration/fault">
            integration:InvalidRequestContent
          </env:Value>
        </env:Subcode>
      </env:Code>
      <env:Reason>
        <env:Text xml:lang="en-US">Operation undefined in the API specification</
        env:Text>
      </env:Reason>
      <env:Role>http://govway.org/integration</env:Role>
      <env:Detail>
        <problem xmlns="urn:ietf:rfc:7807">
```

(continues on next page)

(continua dalla pagina precedente)

```

<type>https://govway.org/handling-errors/400/InvalidRequestContent.html</
↪type>
<title>InvalidRequestContent</title>
<status>400</status>
<detail>Request content not conform to API specification</detail>
<govway_id>9876b03e-0377-4a02-9fb8-07094b0cdf06</govway_id>
</problem>
</env:Detail>
</env:Fault>
</env:Body>
</env:Envelope>

```

9.4 Attivazione di Codici di Errore Specifici

Nella configurazione di default di GovWay, gli errori restituiti ai client non contengono dettagli che possano causare disclosure di informazioni relative al dominio interno. In alcuni casi, per facilitare il supporto alla risoluzione di problemi, è comunque possibile abilitare la generazione di codici più specifici di errore ritornati al client nell'header http "GovWay-Transaction-ErrorStatus" e nel claim "govway_status" del *REST Problem Details - RFC 7807*.

È possibile abilitare temporaneamente la generazione dei codici specifici accendendo alla voce "Strumenti - Runtime" della console di gestione e abilitando "Http Header / Problem Detail" nella sezione «Codici di errore specifici "GovWay-Transaction-ErrorStatus"» (Fig. 9.5).

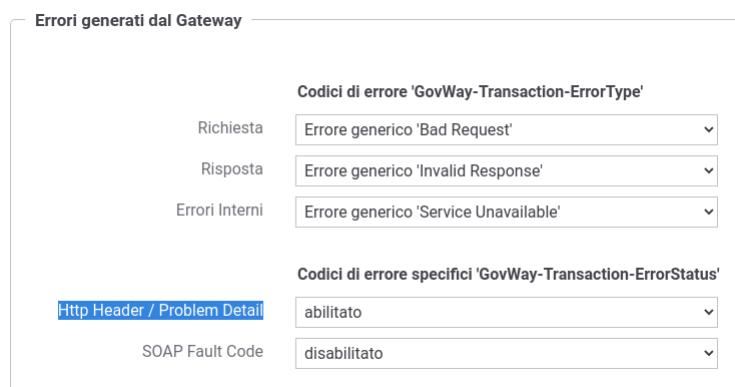


Figure9.5: Attivazione temporanea dei codici di errore specifici di GovWay

L'abilitazione permanente è invece possibile abilitando la seguente proprietà sul file di proprietà esterno /etc/govway/govway_local.properties:

```
org.openspcoop2.pdd.errori.status=true
```

Di seguito viene riportato un esempio di errore generato in seguito al rilevamento di una richiesta non conforme all'interfaccia API REST, dove è stata abilitata la generazione di un codice di errore specifico:

```

HTTP/1.1 400 Bad Request
Server: GovWay
Transfer-Encoding: chunked

```

(continues on next page)

(continua dalla pagina precedente)

```

GovWay-Transaction-ErrorType: InvalidRequestContent
GovWay-Transaction-ErrorStatus: integration:GOVWAY-418
GovWay-Transaction-ID: b76b4d1b-cd9d-43a0-bea2-1f352f1e71dd
Content-Type: application/problem+json
Date: Thu, 28 May 2020 15:59:14 GMT

{
  "type": "https://govway.org/handling-errors/400/InvalidRequestContent.html",
  "title": "InvalidRequestContent",
  "status": 400,
  "detail": "Request content not conform to API specification",
  "govway_id": "b76b4d1b-cd9d-43a0-bea2-1f352f1e71dd",
  "govway_status": "integration:GOVWAY-418"
}

```

Il codice di errore specifico può essere generato anche all'interno del SOAP Fault come "Fault Code" al posto di quello di default generato da GovWay e descritto nella sezione *Classificazione degli Errori*.

È possibile abilitare temporaneamente la generazione all'interno del SOAP Fault Code accendendo alla voce "Strumenti - Runtime" della console di gestione e abilitando "SOAP Fault Code" nella sezione «Codici di errore specifici "GovWay-Transaction-ErrorStatus"» (Fig. 9.6).

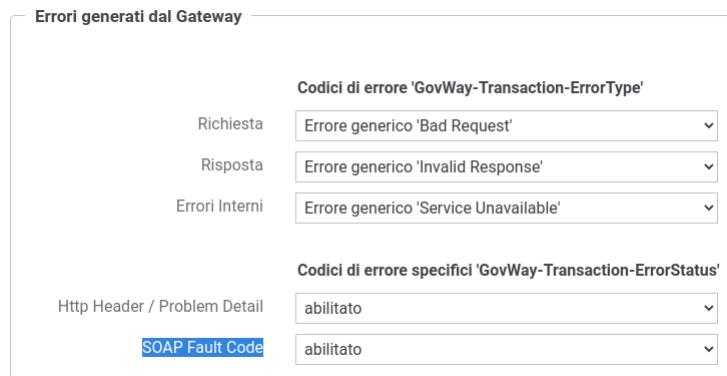


Figure9.6: Attivazione temporanea dei codici di errore specifici di GovWay come SOAP Fault Code

L'abilitazione permanente è invece possibile abilitando la seguente proprietà sul file di proprietà esterno /etc/govway/govway_local.properties:

```
org.openspcoop2.pdd.errori.soap.useGovWayStatusAsFaultCode=true
```

Di seguito viene riportato un esempio di errore generato in seguito al rilevamento di una richiesta non conforme all'interfaccia API SOAP, dove è stata abilitata sia la generazione di un codice di errore specifico che la generazione del SOAP Fault Code specifico:

```

HTTP/1.1 500 Internal Server Error
Server: GovWay
Transfer-Encoding: chunked
GovWay-Transaction-ErrorType: InvalidRequestContent
GovWay-Transaction-ErrorStatus: integration:GOVWAY-418

```

(continues on next page)

(continua dalla pagina precedente)

GovWay-Transaction-ID: b76b4d1b-cd9d-43a0-bea2-1f352f1e71dd
Content-Type: text/xml
Date: Thu, 28 May 2020 15:59:14 GMT

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode xmlns:integration="http://govway.org/integration/fault">
        integration:Client.GOVWAY-423
      </faultcode>
      <faultstring>Received request is not conform to API specification</faultstring>
      <faultactor>http://govway.org/integration</faultactor>
      <detail>
        <problem xmlns="urn:ietf:rfc:7807">
          <type>https://govway.org/handling-errors/400/InvalidRequestContent.html</type>
          <title>InvalidRequestContent</title>
          <status>400</status>
          <detail>Request content not conform to API specification</detail>
          <govway_id>9876b03e-0377-4a02-9fb8-07094b0cdf06</govway_id>
          <govway_status>integration:GOVWAY-418</govway_status>
        </problem>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Di seguito vengono riportate le casistiche di errore che possono verificarsi sul Gateway, con i relativi codici.

Nota: Alcuni degli errori riportati sono scaturiti da funzionalità disponibili nel Gateway attraverso configurazioni avanzate non descritte nel presente manuale.

Table9.2: Codici di Errore GovWay

Codice	Descrizione
integration:GOVWAY-401	Identifica la richiesta di una erogazione o fruizione inesistente
integration:GOVWAY-402	Rientrano in questa casistica gli errori avvenuti durante la fase di autenticazione di una fruizione (sezione <i>Autenticazione Trasporto</i>)
integration:GOVWAY-403	Azione non identificabile tramite i meccanismi configurati. (sezione <i>Modalità di identificazione dell'azione</i>)
integration:GOVWAY-404	Rientrano in questa casistica gli errori avvenuti durante la fase di autorizzazione di una fruizione (sezione <i>Autorizzazione</i>)
integration:GOVWAY-405	Servizio richiesto non esistente (richiede una configurazione non documentata)
integration:GOVWAY-406	Indica che non sono disponibili messaggi (richiede accesso alla MessageBox via Integration Manager, configurazione non documentata)
integration:GOVWAY-407	Il messaggio richiesto non esiste (richiede accesso alla MessageBox via Integration Manager, configurazione non documentata)

continues on next page

Table 9.2 – continua dalla pagina precedente

Codice	Descrizione
integration:GOVWAY-408	Indica che non esiste una API utilizzabile per correlare la richiesta con una precedente transazione (es. utilizzato con i profili asincroni descritti nella sezione <i>Profili Asincroni</i>)
integration:GOVWAY-409	Indica che non è possibile correlare la richiesta con una precedente transazione (es. utilizzato con i profili asincroni descritti nella sezione <i>Profili Asincroni</i>)
integration:GOVWAY-410	L'API invocata possiede il profilo <i>asincrono simmetrico</i> e la configurazione della fruizione non presenta meccanismi di autenticazione dell'applicativo client. L'identificazione di un applicativo fruitore è fondamentale nel profilo asincrono simmetrico per consegnare la risposta (<i>Profili Asincroni</i>)
integration:GOVWAY-411	Indica una configurazione errata dove l'applicativo mittente non possiede una configurazione per la spedizione della risposta asincrona e l'API possiede il profilo <i>asincrono simmetrico</i> (<i>Profili Asincroni</i>)
integration:GOVWAY-412	L'API è stata invocata senza fornire il riferimento ad un messaggio attualmente in carico sul Gateway, mentre la configurazione lo richiede. richiede una configurazione non documentata)
integration:GOVWAY-413	L'API è stata invocata fornendo il riferimento ad un messaggio attualmente in carico sul Gateway, mentre la configurazione non lo richiede. richiede una configurazione non documentata)
integration:GOVWAY-414	L'API invocata è stata configurata con un profilo differente da <i>oneway</i> e richiede la funzionalità di <i>consegna in ordine</i> (sezione <i>Profili di gestione della busta eGov</i>)
integration:GOVWAY-415	L'API invocata è stata configurata per utilizzare la funzionalità di <i>consegna in ordine</i> ma non presenta altre caratteristiche obbligatorie con questa funzionalità (es. confermaRicezione,filtroDuplicati,collaborazione) (sezione <i>Profili di gestione della busta eGov</i>)
integration:GOVWAY-416	Rientrano in questa casistica gli errori avvenuti durante la fase di correlazione applicativa della richiesta (sezione <i>Correlazione Applicativa</i>)
integration:GOVWAY-417	Tale errore viene sollevato se l'interfaccia API e/o gli schemi associati (xsd,json,yaml) contengono errori che non ne consentono l'utilizzo durante la validazione dei contenuti (sezione <i>Validazione dei messaggi</i>)
integration:GOVWAY-418	La validazione dei contenuti ha rilevato una richiesta non conforme all'interfaccia API (sezione <i>Validazione dei messaggi</i>)
integration:GOVWAY-419	La validazione dei contenuti ha rilevato una risposta non conforme all'interfaccia API (sezione <i>Validazione dei messaggi</i>)
integration:GOVWAY-420	Viene sollevato questo errore se un applicativo invoca una fruizione di una API fornendo un messaggio contenente già un header di protocollo. (es. se viene inviato un messaggio contenente un'header eGov (sezione <i>Profilo "SPCoop"</i>)
integration:GOVWAY-421	Indica che il messaggio di richiesta fornito via Integration Manager non è un messaggio SOAP Valido (configurazione non documentata)
integration:GOVWAY-422	Il messaggio di richiesta presente nell'http body (Accesso al servizio out/xml2soap) o il messaggio indicato nella richiesta via IntegrationManager (Accesso al servizio via Integration Manager con imbustamento SOAP) non è utilizzabile, tramite la funzionalità di Imbustamento, per ottenere un messaggio SOAP valido (configurazione non documentata)
integration:GOVWAY-423	L'azione identificata tramite i meccanismi configurati non risulta esistere all'interno dell'API invocata. (sezione <i>Modalità di identificazione dell'azione</i>)

continues on next page

Table 9.2 – continua dalla pagina precedente

Codice	Descrizione
integration:GOVWAY-424	La funzionalità avanzata <i>Allega Body</i> ha generato un errore (configurazione non documentata)
integration:GOVWAY-425	La funzionalità avanzata <i>Scarta Body</i> ha generato un errore (configurazione non documentata)
integration:GOVWAY-426	Errore generico che può avvenire durante la gestione della richiesta, dovuto comunque a dati forniti nella richiesta stessa (es. Valore SOAPAction scorretto)
integration:GOVWAY-427	Indica che il Gateway ha rilevato la presenza di SOAPHeader Element che non è in grado di processare e che richiedono obbligatoriamente il processamento (mustUnderstand=1 e actor non presente)
integration:GOVWAY-428	Rientrano in questa casistica gli errori avvenuti durante la fase di autorizzazione del contenuto (configurazione non documentata)
integration:GOVWAY-429	Errore che viene ritornato dal Gateway se la richiesta presenta un header http <i>Content-Type</i> non supportato (per API SOAP)
integration:GOVWAY-430	Errore che viene ritornato dal Gateway se rileva una busta soap che possiede un namespace differente da quello atteso per la versione SOAP corrispondente al <i>Content-Type</i> (per API SOAP)
integration:GOVWAY-431	Rientrano in questa casistica gli errori avvenuti durante il recupero delle credenziali fornite tramite un Proxy (configurazione non documentata)
integration:GOVWAY-432	Errore che viene ritornato dal Gateway se la richiesta presenta un contenuto malformato (es. xml malformato in una API SOAP)
integration:GOVWAY-433	Indica che la richiesta non presenta un header http <i>Content-Type</i> (obbligatorio in API SOAP)
integration:GOVWAY-434	Rientrano in questa casistica gli errori avvenuti durante la fase di correlazione applicativa della risposta (sezione <i>Correlazione Applicativa</i>)
integration:GOVWAY-435	L'errore viene sollevato se viene rilevata una configurazione <i>Local Forward</i> non corretta (configurazione non documentata)
integration:GOVWAY-436	L'errore viene sollevato se viene rilevato un tipo di fruitore non supportato dalla modalità di utilizzo del Gateway fruista (configurazione non documentata)
integration:GOVWAY-437	L'errore viene sollevato se viene rilevato un tipo di erogatore non supportato dalla modalità di utilizzo del Gateway fruista (configurazione non documentata)
integration:GOVWAY-438	L'errore viene sollevato se viene rilevato un tipo di servizio non supportato dalla modalità di utilizzo del Gateway fruista (configurazione non documentata)
integration:GOVWAY-439	L'errore viene sollevato se viene rilevata una configurazione che richiede una funzionalità non supportata nella modalità di utilizzo del Gateway (configurazione non documentata)
integration:GOVWAY-440	Errore che viene ritornato dal Gateway se la risposta presenta un contenuto malformato (es. xml malformato in una API SOAP)
integration:GOVWAY-441	La richiesta indirizza una configurazione non invocabile direttamente, configurazione creata tramite le indicazioni descritte nella sezione <i>Differenziare le configurazioni specifiche per risorsa/azione</i>
integration:GOVWAY-442	La richiesta pervenuta sul Gateway non presenta un riferimento ad una precedente transazione, mentre la configurazione lo richiede (sezione <i>Correlazione tra transazioni differenti</i>). Nell'installazione di default del Gateway, l'errore indicato non viene mai sollevato poiché non è obbligatorio fornire il riferimento ad una precedente transazione.

continues on next page

Table 9.2 – continua dalla pagina precedente

Codice	Descrizione
integration:GOVWAY-443	L'errore viene ritornato dal Gateway se non viene rilevato un token durante l'invocazione di una fruizione (sezione <i>Autenticazione Token</i>)
integration:GOVWAY-444	L'errore viene ritornato dal Gateway se viene rilevato un token non valido durante l'invocazione di una fruizione (sezione <i>Autenticazione Token</i>)
integration:GOVWAY-445	Rientrano in questa casistica gli errori avvenuti durante la fase di autorizzazione di una fruizione, riguardanti la gestione di un token (sezione <i>Autorizzazione</i>)
integration:GOVWAY-446	Il Gateway ritorna tale codice se la fruizione o l'erogazione invocata risulta sospesa
integration:GOVWAY-450	La richiesta pervenuta sul Gateway non indirizza una erogazione specifica e non è utilizzabile per identificarne alcuna (configurazione non documentata)
integration:GOVWAY-451	Il soggetto invocato non esiste (configurazione non documentata)
integration:GOVWAY-452	Indica che il messaggio ricevuto è già stato gestito in precedenza (es. filtro duplicati attivo descritto nella sezione <i>Profilo "SPCoop"</i>)
integration:GOVWAY-453	L'applicativo erogatore associato all'erogazione non esiste (configurazione non documentata)
integration:GOVWAY-454	Viene sollevato questo errore se il messaggio ritornato come risposta dall'applicativo erogatore, in una erogazione, contiene già un header di protocollo. (es. se viene ritornato un messaggio contenente un'header eGov (sezione <i>Profilo "SPCoop"</i>))
integration:GOVWAY-455	L'errore indica che la richiesta presenta al suo interno degli identificativi di API differenti da quelli dell'erogazione invocata (es. busta eGov contiene dei dati di servizio non allineati all'erogazione invocata)
integration:GOVWAY-500	Errore generico
integration:GOVWAY-516	Errore ritornato dal gateway se non riesce ad inoltrare il messaggio all'endpoint configurato
integration:GOVWAY-517	Errore ritornato dal gateway se non viene ritornata una risposta dall'endpoint contattato e il profilo ne prevede una (es. profilo sincrono nelle API SOAP)
integration:GOVWAY-518	Indica che l'applicativo erogatore ha ritornato un SOAPFault (API SOAP)
integration:GOVWAY-537	La richiesta pervenuta è già presente in carico sul Gateway ed è attualmente in fase di processamento (configurazione non documentata)
integration:GOVWAY-538	La richiesta asincrona pervenuta è già presente in carico sul Gateway ed è attualmente in fase di processamento (configurazione non documentata)
integration:GOVWAY-539	La ricevuta della richiesta asincrona pervenuta è già presente in carico sul Gateway ed è attualmente in fase di processamento (configurazione non documentata)
integration:GOVWAY-CC00	Errore generico avvenuto durante la gestione del Controllo del Traffico (sezione <i>Controllo del Traffico</i>)
integration:GOVWAY-CC01	Il Gateway ha rilevato il superamento del massimo numero di richieste simultanee configurato (sezione <i>Limitazione Numero di Richieste Complessive</i>)
integration:GOVWAY-CP00	Indica che la funzionalità di Rate-Limiting ha rilevato una policy sconosciuta (sezione <i>Rate Limiting</i>)
integration:GOVWAY-CP01	Indica che la funzionalità di Rate-Limiting ha rilevato una violazione di una policy di tipo "NumeroRichieste-RichiesteSimultanee" (sezione <i>Rate Limiting</i>). Il codice di errore può presentare il suffisso -CC se la policy è configurata insieme a controlli di congestione e/o il suffisso -DP se configurata con meccanismi di degrado.

continues on next page

Table 9.2 – continua dalla pagina precedente

Codice	Descrizione
integration:GOVWAY-ERR-CP01	Errore emerso durante la gestione da parte del Gateway della policy di Rate-Limiting di tipo “NumeroRichieste-RichiesteSimultanee” (sezione <i>Rate Limiting</i>).
integration:GOVWAY-CP02	Indica che la funzionalità di Rate-Limiting ha rilevato una violazione di una policy di tipo “NumeroRichieste-ControlloRealtime*” (sezione <i>Rate Limiting</i>). Il codice di errore può presentare il suffisso -CC se la policy è configurata insieme a controlli di congestione e/o il suffisso -DP se configurata con meccanismi di degrado (sezione <i>Rate Limiting</i>).
integration:GOVWAY-ERR-CP02	Errore emerso durante la gestione da parte del Gateway della policy di Rate-Limiting di tipo “NumeroRichieste-ControlloRealtime*” (sezione <i>Rate Limiting</i>).
integration:GOVWAY-CP03	Indica che la funzionalità di Rate-Limiting ha rilevato una violazione di una policy di tipo “OccupazioneBanda-*” (sezione <i>Rate Limiting</i>). Il codice di errore può presentare il suffisso -CC se la policy è configurata insieme a controlli di congestione e/o il suffisso -DP se configurata con meccanismi di degrado (sezione <i>Rate Limiting</i>).
integration:GOVWAY-ERR-CP03	Errore emerso durante la gestione da parte del Gateway della policy di Rate-Limiting di tipo “OccupazioneBanda-*” (sezione <i>Rate Limiting</i>).
integration:GOVWAY-CP04	Indica che la funzionalità di Rate-Limiting ha rilevato una violazione di una policy di tipo “TempoComplessivioRisposta” (sezione <i>Rate Limiting</i>). Il codice di errore può presentare il suffisso -CC se la policy è configurata insieme a controlli di congestione e/o il suffisso -DP se configurata con meccanismi di degrado.
integration:GOVWAY-ERR-CP04	Errore emerso durante la gestione da parte del Gateway della policy di Rate-Limiting di tipo “TempoComplessivioRisposta” (sezione <i>Rate Limiting</i>).
integration:GOVWAY-CP05	Indica che la funzionalità di Rate-Limiting ha rilevato una violazione di una policy di tipo “TempoMedioRisposta-*” (sezione <i>Rate Limiting</i>). Il codice di errore può presentare il suffisso -CC se la policy è configurata insieme a controlli di congestione e/o il suffisso -DP se configurata con meccanismi di degrado.
integration:GOVWAY-ERR-CP05	Errore emerso durante la gestione da parte del Gateway della policy di Rate-Limiting di tipo “TempoMedioRisposta-*” (sezione <i>Rate Limiting</i>).
integration:GOVWAY-CP06	Indica che la funzionalità di Rate-Limiting ha rilevato una violazione di una policy di tipo “NumeroRichiesteCompletaeConSuccesso” (sezione <i>Rate Limiting</i>). Il codice di errore può presentare il suffisso -CC se la policy è configurata insieme a controlli di congestione e/o il suffisso -DP se configurata con meccanismi di degrado.
integration:GOVWAY-ERR-CP06	Errore emerso durante la gestione da parte del Gateway della policy di Rate-Limiting di tipo “NumeroRichiesteCompletaeConSuccesso” (sezione <i>Rate Limiting</i>).
integration:GOVWAY-CP07	Indica che la funzionalità di Rate-Limiting ha rilevato una violazione di una policy di tipo “NumeroRichiesteFallite” (sezione <i>Rate Limiting</i>). Il codice di errore può presentare il suffisso -CC se la policy è configurata insieme a controlli di congestione e/o il suffisso -DP se configurata con meccanismi di degrado.
integration:GOVWAY-ERR-CP07	Errore emerso durante la gestione da parte del Gateway della policy di Rate-Limiting di tipo “NumeroRichiesteFallite” (sezione <i>Rate Limiting</i>).

continues on next page

Table 9.2 – continua dalla pagina precedente

Codice	Descrizione
integration:GOVWAY-CP08	Indica che la funzionalità di Rate-Limiting ha rilevato una violazione di una policy di tipo “NumeroFaultApplicativi” (sezione <i>Rate Limiting</i>). Il codice di errore può presentare il suffisso <i>-CC</i> se la policy è configurata insieme a controlli di congestione e/o il suffisso <i>-DP</i> se configurata con meccanismi di degrado.
integration:GOVWAY-ERR-CP08	Errore emerso durante la gestione da parte del Gateway della policy di Rate-Limiting di tipo “NumeroFaultApplicativi” (sezione <i>Rate Limiting</i>).
protocol:GOVWAY-109	Rientrano in questa casistica gli errori avvenuti durante la fase di autenticazione di una erogazione, se non vengono rilevate credenziali (sezione <i>Autenticazione Trasporto</i>)
protocol:GOVWAY-117	Rientrano in questa casistica gli errori avvenuti durante la fase di autenticazione di una erogazione, se vengono rilevate credenziali non corrette (sezione <i>Autenticazione Trasporto</i>)
protocol:GOVWAY-1350	Rientrano in questa casistica eventuali errori generici avvenuti durante la fase di autorizzazione di una erogazione (sezione <i>Autorizzazione</i>) o sicurezza del messaggio (sezione <i>Sicurezza a livello del messaggio</i>)
protocol:GOVWAY-1351	L'errore viene ritornato dal Gateway se viene rilevato che il messaggio presenta al suo interno un mittente differente da quello identificato dalle credenziali (configurazione non documentata)
protocol:GOVWAY-1352	Rientrano in questa casistica gli errori avvenuti durante la fase di autorizzazione di una erogazione, quando la richiesta non viene autorizzata (sezione <i>Autorizzazione</i>)
protocol:GOVWAY-[1353-1354]	L'errore viene ritornato dal Gateway se viene rilevato che la firma della busta, prevista dalla modalità utilizzata, non è rispettivamente valida o presente (configurazione non documentata)
protocol:GOVWAY-1355	L'errore viene ritornato dal Gateway se viene rilevato che la firma del messaggio non è valida (sezione <i>Sicurezza a livello del messaggio</i>)
protocol:GOVWAY-1356	L'errore viene ritornato dal Gateway se viene rilevato che il messaggio non è firmato (sezione <i>Sicurezza a livello del messaggio</i>)
protocol:GOVWAY-[1357-1360]	L'errore viene ritornato dal Gateway se viene rilevato che la firma degli allegati non sono valide o presenti (configurazione non documentata)
protocol:GOVWAY-1361	L'errore viene ritornato dal Gateway se viene rilevato che la cifratura del messaggio non è valida (sezione <i>Sicurezza a livello del messaggio</i>)
protocol:GOVWAY-1362	L'errore viene ritornato dal Gateway se viene rilevato che il messaggio non è cifrato (sezione <i>Sicurezza a livello del messaggio</i>)
protocol:GOVWAY-[1363-1364]	L'errore viene ritornato dal Gateway se viene rilevato che le cifrature degli allegati non sono valide o presenti (configurazione non documentata)
protocol:GOVWAY-1365	L'errore viene ritornato dal Gateway se viene rilevato che il messaggio non contiene l'attesa configurazione di sicurezza (sezione <i>Sicurezza a livello del messaggio</i>)
protocol:GOVWAY-1366	L'errore viene ritornato dal Gateway se non viene rilevato un token durante l'invocazione di una erogazione (sezione <i>Autenticazione Token</i>)
protocol:GOVWAY-1367	L'errore viene ritornato dal Gateway se viene rilevato un token non valido durante l'invocazione di una erogazione (sezione <i>Autenticazione Token</i>)
protocol:GOVWAY-1368	Rientrano in questa casistica gli errori avvenuti durante la fase di autorizzazione di una erogazione, riguardanti la gestione di un token (sezione <i>Autorizzazione</i>)
protocol:GOVWAY-[1-6]	Rientrano in questa casistica gli errori generici avvenuti durante il processamento e la validazione di una richiesta di erogazione

continues on next page

Table 9.2 – continua dalla pagina precedente

Codice	Descrizione
protocol:GOVWAY-[51-60]	Gli errori che rientrano in questa casistica vengono generati durante la validazione della richiesta se sono presenti informazioni non valide per quanto concerne gli attributi <i>mustUnderstand</i> e <i>actor</i> di un header SOAP (es. busta egov nella modalità descritta in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[100-120]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sul mittente (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[150-170]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sul destinatario (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[200-205]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sul profilo di collaborazione (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[250-265]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sul servizio (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[300-315]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sulla correlazione asincrona (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[350-355]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sulla collaborazione (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[400-406]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sull'azione (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[450-455]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sulla correlazione asincrona per quanto riguarda l'azione (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[500-506]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sull'identificativo messaggio (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[550-556]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sul riferimento messaggio (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[600-610]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sull'ora registrazione (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[650-661]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sulla scadenza (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[700-717]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sul filtro duplicati e sulla conferma della ricezione (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[750-766]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sulla consegna in ordine (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[800-817]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sul servizio applicativo
protocol:GOVWAY-[850-879]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sui riscontri (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[900-971]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sulla lista trasmissioni (es. busta egov in sezione <i>Profilo “SPCoop”</i>)

continues on next page

Table 9.2 – continua dalla pagina precedente

Codice	Descrizione
protocol:GOVWAY-[1000-1035]	Errore rilevato durante la validazione della richiesta che riguarda informazioni sulla lista eccezioni (es. busta egov in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[1300-1329]	Errore rilevato durante la validazione del messaggio per quanto concerne la parte di SOAPFault previsto dal protocollo (es. busta egov errore in sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-[1400-1404]	Errore rilevato durante la validazione del messaggio per quanto concerne la parte di attachments previsto dal protocollo (es. busta egov con attachments, sezione <i>Profilo “SPCoop”</i>)
protocol:GOVWAY-2000	Errore generico rilevato durante la validazione del messaggio

Funzionalità Avanzate

10.1 Modalità Avanzata

L’interfaccia della govwayConsole, fin qui descritta, fa riferimento all’operatività nella *modalità standard*. La modalità standard prevede varie semplificazioni, sulle opzioni visualizzate nelle schermate, mirate al compimento delle operazioni di uso comune.

Nel caso si avesse la necessità di ricorrere a configurazioni più specifiche, non contemplate nella modalità standard, è possibile passare alla visualizzazione dell’interfaccia nella *Modalità Avanzata* utilizzando la voce omonima del menu a discesa che compare selezionando l’icona in alto a destra (nella testata della govwayConsole) come mostrato nella figura Fig. 10.1.

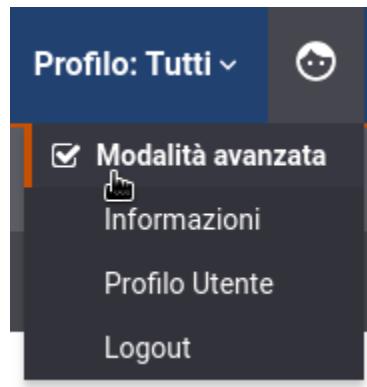


Figure 10.1: Selezione Modalità Avanzata

Nota: La modifica della modalità attuata tramite il menù a discesa non è persistente e al successivo login verrà nuovamente presentata la modalità di default associata al profilo utente. Per modificarlo si rimanda alla sezione *Profilo Utente*.

Operando in modalità avanzata, in ciascuno dei contesti di configurazione già descritti in questo manuale, compariranno opzioni aggiuntive per le quali sono previsti valori di default nel caso della modalità standard.

Nella modalità avanzata sarà disponibile la funzionalità aggiuntiva *Elimina*, presente nel menu di Configurazione, che consente di utilizzare package di esportazione per l'eliminazione selettiva di entità dal registro.

Nota: Non tutte le funzionalità disponibili in modalità avanzata sono descritte nel presente manuale.

10.2 Configurazione manuale delle interfacce

Nel caso non si disponga del descrittore della API, è possibile in alternativa fornire manualmente la specifica delle interfacce. Dopo aver salvato la nuova API, senza aver fornito il descrittore delle interfacce, si procede individuando il nuovo elemento nella lista delle API presenti e cliccando sul collegamento presente nella colonna *Servizi*, nel caso SOAP, o *Risorse* nel caso REST.

Nel caso SOAP, si procede aggiungendo il nuovo servizio tramite il pulsante *Aggiungi*. Il form da compilare è quello mostrato nella figura seguente.

API > Servizi di Hello:1 > **Aggiungi**

Note: (*) Campi obbligatori

Servizio

Nome *****

Descrizione

Informazioni Protocollo

Profilo di collaborazione

ID Collaborazione

Riferimento ID Richiesta

Invia **Cancella**

Figure 10.2: Aggiunta di un servizio alla API SOAP

I dati da fornire sono i seguenti:

- *Nome* del servizio
- *Descrizione* del servizio
- *Profilo di collaborazione* del servizio, a scelta tra oneway e sincrono
- *ID Conversazione*. Flag per consentire di specificare nelle richieste un valore che identifica una conversazione.
- *Riferimento ID Richiesta*. Flag per consentire di specificare nelle richieste un identificativo relativo ad un messaggio precedente.

Al passo successivo, utilizzando il collegamento nella colonna *Azioni*, relativamente al servizio appena creato, si procede con l'aggiunta delle azioni. Il form da compilare è quello mostrato nella figura seguente.

Note: (*) Campi obbligatori

Azione

Nome *

Informazioni Protocollo

Profilo

usa profilo servizio

Invia Cancella

Figure10.3: Aggiunta di un'azione alla API SOAP

I dati da fornire sono i seguenti:

- *Nome* dell'azione
- *Profilo*. Si può scegliere se utilizzare le impostazioni già fornite a livello del servizio, oppure ridefinirle indicando nuovamente Profilo di collaborazione, ID Conversazione e Riferimento ID Richiesta.

Nel caso REST, si procede aggiungendo la nuova risorsa tramite il pulsante *Aggiungi*. Il form da compilare è quello mostrato nella figura seguente.

I dati da fornire sono i seguenti:

- *HTTP Method* relativo alla risorsa (GET, POST, DELETE, ecc.)

The screenshot shows a user interface for adding a new resource to an API. The top navigation bar indicates the path: API > api-config v1 > Risorse > Aggiungi. A note at the top left of the form states: "Note: (*) Campi obbligatori". The form is divided into two main sections: "Risorsa" and "Informazioni Protocollo".

Risorsa

- HTTP Method: Qualsiasi (dropdown menu)
- Path: (text input field)
- Nome *: (text input field, marked with a red asterisk)
- Descrizione: (text area)

Informazioni Protocollo

- ID Conversazione: (checkbox)
- Riferimento ID Richiesta: (checkbox)

SALVA (Save button)

Figure10.4: Aggiunta di una risorsa alla API REST

- *Path* della risorsa
- *Nome* della risorsa
- *Descrizione* della risorsa
- *ID Conversazione*. Flag per consentire di specificare nelle richieste un valore che identifica una conversazione.
- *Riferimento ID Richiesta*. Flag per consentire di specificare nelle richieste un identificativo relativo ad un messaggio precedente.

10.3 Versionamento delle API e delle Erogazioni/Fruizioni

Su GovWay vi è una gestione del versionamento effettuato su due componenti:

- API
- Erogazione o Fruizione dell'API

Come descritto nella sezione [Versionamento delle API](#), sulla singola erogazione/fruizione è possibile modificare la versione dell'API implementata solamente se ne esiste più di una versione. Questa modifica si riflette automaticamente anche sulla versione dell'erogazione/fruizione, e sull'url di invocazione, se non esiste già una erogazione/fruizione con la nuova versione.

Utilizzando la console in modalità avanzata ([Modalità Avanzata](#)) è invece possibile modificare puntualmente la versione dell'erogazione/fruizione e di conseguenza l'url di invocazione tramite il bottone “modifica” evidenziato nella figura [Fig. 10.5](#).

Accedendo alla modifica del nome dell'erogazione/fruizione con la console in modalità avanzata, è possibile modificare la versione ([Fig. 10.6](#)).

Effettuata la modifica l'erogazione possiederà una versione indipendente dalla versione dell'API implementata. L'url di invocazione riflette la versione dell'erogazione come evidenziato nella figura [Fig. 10.7](#).

10.4 Modalità di identificazione dell'azione

Nel contesto dei servizi Soap, sia erogazioni che fruizioni, si ha la possibilità di selezionare una tra diverse opzioni che riguardano la modalità di identificazione dell'azione. Dopo aver acceduto la sezione [URL di Invocazione](#), relativamente alla fruizione o erogazione, si può selezionare una tra le seguenti opzioni:

- *Contenuto* (Soap e Rest): il dato viene ricavato dal messaggio di richiesta utilizzando come criterio l'espressione XPath o JsonPath indicata nel campo *Pattern* sottostante.
- *Header HTTP* (Soap e Rest): il dato viene ricavato da un valore passato come Http Header. Il campo sottostante consente di specificare il nome di tale header.
- *Header di Integrazione* (Soap e Rest): il dato viene ricavato dall'header di integrazione fornito con il messaggio di richiesta. Per conoscere come gli applicativi client forniscono tale informazione vedere la sezione [Scambio di informazioni nella richiesta del client verso il gateway](#).
- *Specifica di Interfaccia dell'API* (Soap e Rest): il dato viene ricavato in automatico sulla base delle informazioni fornite con la richiesta (messaggio e parametri) confrontandole con la descrizione dell'interfaccia dell'API.
- *Url di Invocazione* (Soap): il dato viene ricavato dinamicamente dalla url di invocazione utilizzando come criterio l'espressione regolare inserita nel campo *Espressione Regolare* sottostante (l'espressione deve avere un match con l'intera url).
- *SOAPAction* (Soap): Questa opzione consente di ricavare il dato dal campo *SOAPAction* presente nell'header di trasporto delle comunicazioni SOAP.

Erogazioni > PetStore@ENTE v2

PetStore@ENTE v2

Nome	PetStore v2	 
Soggetto Erogatore	ENTE	
API	PetStore v2 (Rest)	  
URL Invocazione	http://127.0.0.1:8080/govway/ENTE/PetStore/v2	
Connettore	http://petstore.swagger.io/v2	  
Gestione CORS	Abilitato	

CONFIGURA



Figure10.5: Nuova Versione di una Erogazione

Erogazioni > PetStore@ENTE v2 > Informazioni Generali

Informazioni Generali

Note: (*) Campi obbligatori

Informazioni Generali

Nome *	PetStore
Versione	6
<u>Allegati (0)</u>	

Figure10.6: Scelta di una nuova versione per una Erogazione

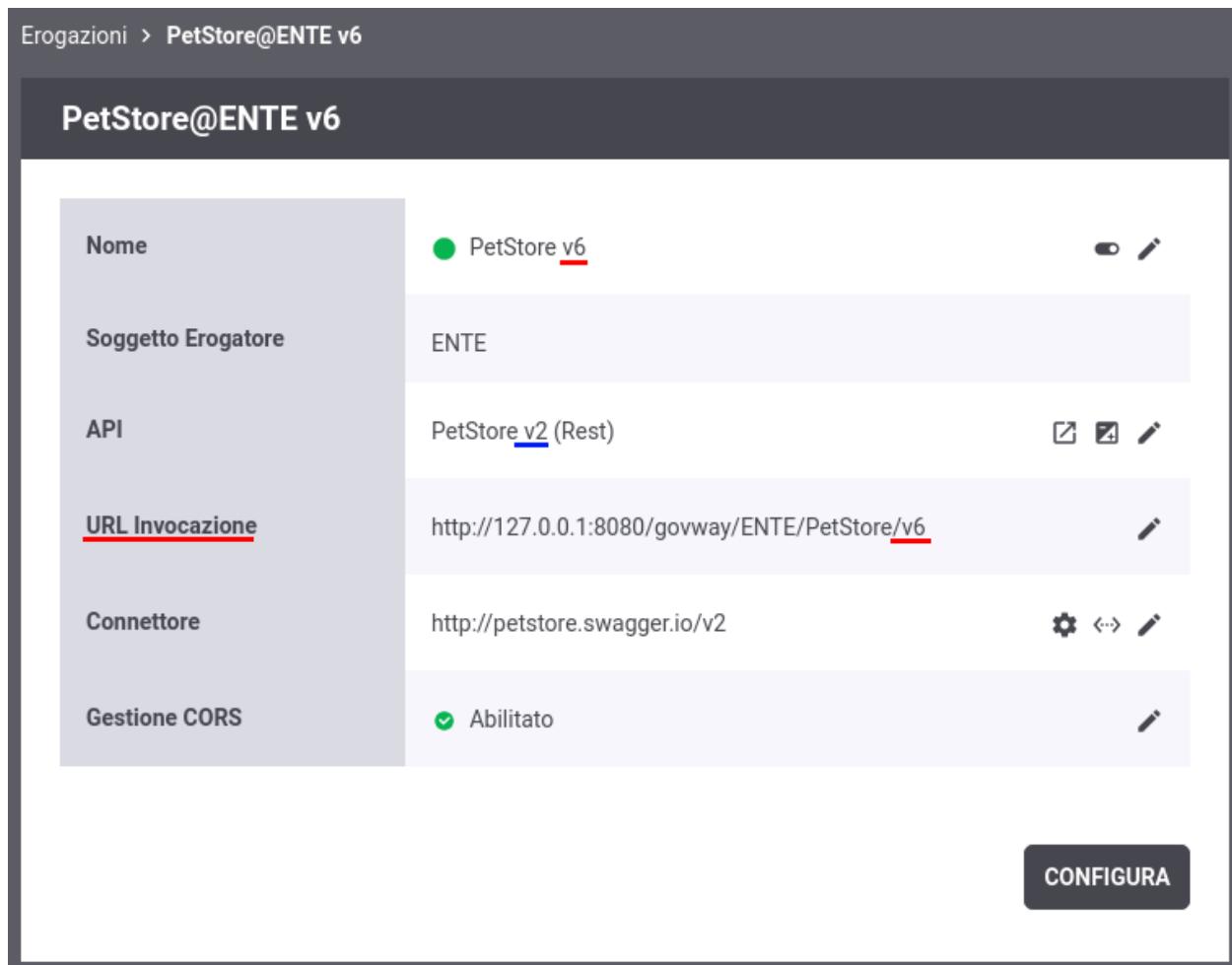


Figure10.7: Nuova versione dell'erogazione differente dalla versione dell'API

Attivando il flag *Identificazione tramite API*, in caso di fallimento dell'identificazione dell'azione nella modalità prevista al passo precedente, si tenterà di utilizzare la modalità «Specifica di Interfaccia dell'API» come seconda opzione.

Il campo *Azioni* illustra l'elenco delle azioni presenti per semplice comodità.

10.5 Multi-Tenant

I processi di configurazione, descritti in questo manuale, sono ottimizzati nell'ottica di mantenere sempre sottinteso il soggetto interno al dominio. In tal senso, le fruizioni e le erogazioni si intendono sempre in soggettiva riguardo un singolo soggetto interno amministrato dall'utente in sessione.

Multi-tenant è un'opzione che consente di estendere l'ambito delle configurazioni prodotte dalla GovWayConsole a più di un soggetto interno al dominio. Tale opzione si attiva nella configurazione generale (sezione [Generale](#)).

Per gestire la compresenza di più soggetti interni al dominio, per la configurazione di erogazioni e fruizioni, è possibile scegliere quali soggetti interni rendere ammissibili ([Fig. 10.8](#)):

- *Fruizioni (Soggetto Erogatore)*
 - *Tutti*: indica che tutti i soggetti interni, censiti nel registro di GovWay, sono selezionabili come soggetto erogatore, in una fruizione.
 - *Escludi Soggetto Fruitore*: indica che tutti i soggetti interni, tranne il soggetto fruitore, sono selezionabili come soggetto erogatore, in una fruizione.
 - *Solo Soggetti Esterni*: indica che il soggetto erogatore di una fruizione deve essere un soggetto esterno.
- *Erogazioni (Soggetti Fruitori)*
 - *Tutti*: indica che tutti i soggetti interni, censiti nel registro di GovWay, sono selezionabili come soggetti fruitori, in una erogazione.
 - *Escludi Soggetto Erogatore*: indica che tutti i soggetti interni, tranne il soggetto erogatore, sono selezionabili come soggetti fruitori, in una erogazione.
 - *Solo Soggetti Esterni*: indica che i soggetti fruitori di una erogazione devono essere soggetti esterni.

	Stato
Soggetto Erogatore	abilitato
Soggetti Fruitori	abilitato

Fruizioni

Soggetto Erogatore: Solo Soggetti Esterni

Erogazioni

Soggetti Fruitori: Escludi Soggetto Erogatore

Figure 10.8: Elementi di configurazione della modalità multi-tenant

L'utente che ha l'opzione multi-tenant attiva, visualizza sulla testata un menu a discesa che consente di selezionare l'utente interno al dominio sul quale vuole operare ([Fig. 10.9](#)). Se viene selezionato un soggetto dalla lista, l'operatività

sulla console risulterà identica alla situazione con un unico soggetto interno. Selezionando l'opzione «Tutti» sarà richiesto nei singoli contesti di specificare il soggetto interno.

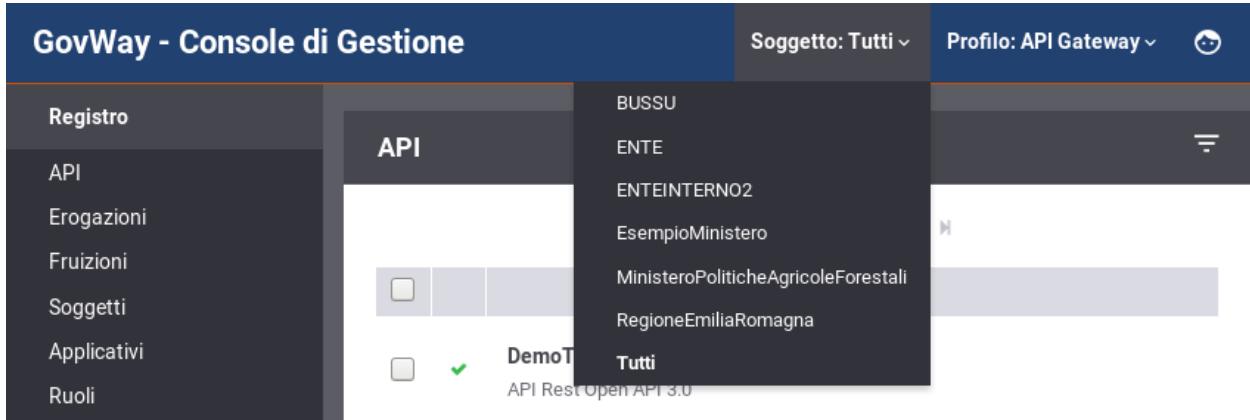


Figure10.9: Selezione del soggetto operativo in modalità multi-tenant

10.6 Header di Integrazione

In base alle configurazioni prodotte per i servizi, è previsto in diverse situazioni che gli applicativi scambino dei dati con il gateway.

Nel caso degli applicativi server lo scopo è quello di ricevere dal gateway i metadati che riguardano la richiesta gestita.

Per gli applicativi client tale scambio si rende necessario al fine di fornire al gateway specifici parametri necessari a elaborare la richiesta.

Per consentire lo scambio di tali informazioni, funzionali all'integrazione tra applicativi e gateway, sono previste alcune strutture dati, che indichiamo di seguito con il termine *Header di Integrazione*, che possono essere trasmesse in differenti modalità:

- *Trasporto*: le informazioni sono contenute nell'header di trasporto
- *Url Based*: le informazioni sono incapsulate nella url
- *SOAP*: le informazioni sono incluse in uno specifico header SOAP proprietario di GovWay
- *WS-Addressing*: le informazioni sono incluse in un header SOAP secondo il formato standard WS-Addressing

Nel seguito descriviamo le strutture degli header di integrazione attive per default con l'installazione del prodotto. Tali strutture variano in funzione del ruolo dell'applicativo. Per l'applicativo client è possibile fornire informazioni al gateway tramite le modalità *Trasporto* e *Url Based*. L'applicativo server, invece, riceve le informazioni dal gateway solamente tramite la modalità *Trasporto*.

10.6.1 Scambio di informazioni nella richiesta inoltrata dal gateway al server

Le informazioni fornite dal gateway all'applicativo erogatore, sia per quanto concerne fruizioni che per erogazioni, sono riassunte nella [Tabella 10.1](#).

Table10.1: Scambio di informazioni nella richiesta inoltrata dal gateway al server

Nome Header Trasporto	Descrizione
GovWay-Message-ID	Identificativo del messaggio assegnato da GovWay
GovWay-Relates-To	Identificativo del messaggio riferito
GovWay-Conversation-ID	Identificativo della conversazione
GovWay-Transaction-ID	Identificativo della transazione assegnato da GovWay

Inoltre, solamente per quanto concerne le erogazioni, all'applicativo interno al dominio vengono inoltrate ulteriori meta-informazioni riguardanti la transazione gestita sul gateway descritte nella [Tabella 10.2](#).

Table10.2: Scambio di informazioni nella richiesta inoltrata dal gateway al server per una Erogazione

Header	Descrizione
GovWay-Sender-Type	Codice che identifica il tipo del mittente
GovWay-Sender	Identificativo del mittente
GovWay-Provider-Type	Codice che identifica il tipo del destinatario
GovWay-Provider	Identificativo del destinatario
GovWay-Service-Type	Codice che identifica il tipo del servizio
GovWay-Service	Identificativo del servizio
GovWay-Service-Version	Progressivo di versione del servizio
GovWay-Action	Identificativo dell'azione
GovWay-Application-Message-ID	Identificativo del messaggio assegnato dall'applicativo
GovWay-Application	Identificativo dell'applicativo
GovWay-Token-Sender-Type	Codice che identifica il tipo del dominio mittente dell'applicativo identificato tramite autenticazione token
GovWay-Token-Sender	Identificativo del dominio mittente dell'applicativo identificato tramite autenticazione token
GovWay-Token-Application	Identificativo dell'applicativo identificato tramite autenticazione token

10.6.2 Informazioni restituite dal gateway nella risposta all'applicativo client

Header Standard

Le informazioni restituite dal gateway all'applicativo client, sia per le fruizioni che per le erogazioni, sono riassunte nella [Tabella 10.3](#).

Table10.3: Header restituiti dal gateway nella risposta all'applicativo client

Nome Header Trasporto	Descrizione
GovWay-Message-ID	Identificativo del messaggio assegnato da GovWay
GovWay-Relates-To	Identificativo del messaggio riferito
GovWay-Conversation-ID	Identificativo della conversazione
GovWay-Application-Message-ID	Identificativo del messaggio assegnato dall'applicativo (solo nel caso di Fruizione)
GovWay-Transaction-ID	Identificativo della transazione assegnato da GovWay

Header RateLimiting

All'applicativo client vengono inoltre restituiti ulteriori header http informativi se l'applicativo erogatore non è disponibile o se sono stati attivati meccanismi di Rate Limiting (sezione [Rate Limiting](#)).

Table10.4: Ulteriori header restituiti dal gateway nella risposta all'applicativo client

Nome Header Trasporto	Descrizione	Motivazione
Retry-After	Indica al client il numero di secondi dopo i quali ripresentarsi poiché il servizio contattato non è al momento disponibile.	Le principali cause della generazione di tale header sono imputabili alla non raggiungibilità un applicativo erogatore, alla violazione di politiche di RateLimiting o a quando un servizio è temporaneamente disabilitato
X-RateLimit-Limit	Indica il numero massimo di richieste effettuabili	Rate-Limiting attivato con policy di tipo "NumeroRichieste-ControlloRealtime*" (sezione Rate Limiting)
X-RateLimit-Remaining	Numero di richieste rimanenti prima del prossimo reset	Rate-Limiting attivato con policy di tipo "NumeroRichieste-ControlloRealtime*" (sezione Rate Limiting)
X-RateLimit-Reset	Numero di secondi mancante al prossimo reset	Rate-Limiting attivato con policy di tipo "NumeroRichieste-ControlloRealtime*" (sezione Rate Limiting)
GovWay-RateLimit- ConcurrentRequest-Limit	Indica il numero massimo di richieste concorrenti inviabili	Rate-Limiting attivato con policy di tipo "NumeroRichieste-RichiesteSimultanee" (sezione Rate Limiting)
GovWay-RateLimit- ConcurrentRequest-Remaining	Indica il numero massimo di richieste concorrenti ancora inviabili	Rate-Limiting attivato con policy di tipo "NumeroRichieste-RichiesteSimultanee" (sezione Rate Limiting)
GovWay-RateLimit- BandwithQuota-Limit	Indica la massima banda occupabile	Rate-Limiting attivato con policy di tipo "OccupazioneBanda-*" (sezione Rate Limiting)
GovWay-RateLimit- BandwithQuota-Remaining	Indica la banda ancora occupabile prima del prossimo reset	Rate-Limiting attivato con policy di tipo "OccupazioneBanda-*" (sezione Rate Limiting)
GovWay-RateLimit- BandwithQuota-Reset	Numero di secondi mancante al prossimo reset	Rate-Limiting attivato con policy di tipo "OccupazioneBanda-*" (sezione Rate Limiting)

continues on next page

Table 10.4 – continua dalla pagina precedente

Nome Header Trasporto	Descrizione	Motivazione
GovWay-RateLimit-AvgTimeResponse-Limit	Tempo medio di risposta atteso	Rate-Limiting attivato con policy di tipo “TempoMedioRisposta-*” (sezione <i>Rate Limiting</i>)
GovWay-RateLimit-AvgTimeResponse-Reset	Numero di secondi mancante al prossimo reset	Rate-Limiting attivato con policy di tipo “TempoMedioRisposta-*” (sezione <i>Rate Limiting</i>)
GovWay-RateLimit-TimeResponseQuota-Limit	Tempo complessivo di risposta occupabile	Policy creata con risorsa di tipo “TempoComplessivioRisposta” (sezione <i>Registro Policy</i>)
GovWay-RateLimit-TimeResponseQuota-Remaining	Tempo di risposta ancora occupabile prima del prossimo reset	Policy creata con risorsa di tipo “TempoComplessivioRisposta” (sezione <i>Registro Policy</i>)
GovWay-RateLimit-TimeResponseQuota-Reset	Numero di secondi mancante al prossimo reset	Policy creata con risorsa di tipo “TempoComplessivioRisposta” (sezione <i>Registro Policy</i>)
GovWay-RateLimit-RequestSuccessful-Limit, GovWay-RateLimit-RequestFailed-Limit, GovWay-RateLimit-Fault-Limit	Indica il numero massimo di richieste effettuabili	Policy creata rispettivamente con risorsa di tipo “NumeroRichiesteCompletateConSuccesso”, “NumeroRichiesteFallite” e “NumeroFaultApplicativi” (sezione <i>Registro Policy</i>)
GovWay-RateLimit-RequestSuccessful-Remaining, GovWay-RateLimit-RequestFailed-Remaining, GovWay-RateLimit-Fault-Remaining	Numero di richieste rimanenti prima del prossimo reset	Policy creata rispettivamente con risorsa di tipo “NumeroRichiesteCompletateConSuccesso”, “NumeroRichiesteFallite” e “NumeroFaultApplicativi” (sezione <i>Registro Policy</i>)
GovWay-RateLimit-RequestSuccessful-Reset, GovWay-RateLimit-RequestFailed-Reset, GovWay-RateLimit-Fault-Reset	Numero di secondi mancante al prossimo reset	Policy creata rispettivamente con risorsa di tipo “NumeroRichiesteCompletateConSuccesso”, “NumeroRichiesteFallite” e “NumeroFaultApplicativi” (sezione <i>Registro Policy</i>)

Header di Sicurezza

È stata introdotta una politica di generazione automatica degli header HTTP indicati nella Tabella 10.5, se non ritornati dal backend che implementa l'API, con lo scopo di evitare alcune vulnerabilità a cui possono essere soggette le implementazioni delle API.

Nota: Il caching viene disabilitato per evitare che delle risposte vengano inopportunamente messe in cache, come indicato nelle Linee Guida - raccomandazioni tecniche per REST “RAC_REST_NAME_010”. Il mancato rispetto di questa raccomandazione può portare all'esposizione accidentale di dati personali.

Table10.5: Header restituiti dal gateway nella risposta all'applicativo client, se non ritornati dal Backend

Nome Header Trasporto	Valore
X-Content-Type-Options	nosniff
Cache-Control	no-cache, no-store, must-revalidate
Pragma	no-cache
Expires	0
Vary	*

È possibile configurare una gestione personalizzata degli header di sicurezza per la singola API registrando le seguenti *Proprietà* sull'erogazione o sulla fruizione:

- *securityHeaders.enabled* : consente di disabilitare la generazione degli headers di sicurezza. I valori associabili alla proprietà sono “true” o “false”. Per default questo controllo è abilitato.
- *securityHeaders.default* : consente di disabilitare la generazione degli headers di sicurezza di default. I valori associabili alla proprietà sono “true” o “false”. Per default questo controllo è abilitato.
- *securityHeaders* : lista di nomi di header http, separati con la virgola. Per ogni header indicato deve essere registrata una ulteriore proprietà dove va indicato il valore da associare all'header:
 - *securityHeaders.<nomeHeader> = <valoreHeader>*

10.6.3 Scambio di informazioni nella richiesta del client verso il gateway

Le informazioni che possono essere fornite dal client al gateway sono riassunte nella tabella [Tabella 10.6](#) e riguardano le modalità *Trasporto* e *Url Based* attive di default.

Table10.6: Scambio di informazioni nella richiesta del client verso il gateway

Nome Header Trasporto	Nome Url Property	Descrizione
GovWay-Action	govway_action	Identificativo dell'azione invocata. Tale informazione deve essere fornita dal client se il servizio è stato configurato in modalità di identificazione dell'azione <i>input-based</i> . (Sezione Modalità di identificazione dell'azione)
GovWay-Relates-To	govway_relates_to	Identificativo di un precedente messaggio a cui la richiesta in essere si riferisce. (Sezione Correlazione tra transazioni differenti)
GovWay-Conversation-ID	govway_conversation_id	Identificativo di una conversazione a cui la richiesta in essere si riferisce (Sezione Correlazione tra transazioni differenti)

Esiste inoltre la possibilità per il client di fornire informazioni di integrazione tramite un json, come descritto nella sezione [Scambio di informazioni tramite un token JSON](#).

10.6.4 Altri header di Integrazione

Per attivare ulteriori header di integrazione è richiesto l'accesso alla govwayConsole in modalità *avanzata* (Sezione *Modalità Avanzata*).

Nota: Gli header di trasporto relativi alle funzionalità di Rate-Limiting e Service-Unavailable, descritti nella sezione *Informazioni restituite dal gateway nella risposta all'applicativo client*, vengono generati solamente nella modalità *Header HTTP*.

A partire dall'erogazione o fruizione di una API, accedendo alla sezione *Configurazione dell'API* in modalità avanzata compare una sezione precedentemente non documentata denominata *Opzioni Avanzate*. All'interno di tale sezione è possibile agire sulla configurazione della voce *Metadati* nella sezione *Integrazione* per attivare gli header di integrazione desiderati :

Nota: Per ogni tipo di header di integrazione descritto di seguito è possibile indicare, tramite una voce di configurazione dedicata, se deve essere generato solamente nei messaggi inoltrati al dominio interno (richiesta inoltrata al server nelle erogazioni o risposta restituita al client nelle fruizioni) o anche verso il dominio esterno.

- *Header HTTP*: vengono generati gli header di trasporto descritti nelle precedenti sezioni.
- *Parametri della Url*: le informazioni precedentemente descritte vengono aggiunte alla url tramite i parametri descritti nella [Tabella 10.7](#).

Table10.7: Informazioni generate dal gateway nella url della richiesta inoltrata al server

Nome Parameter	Query URL	Descrizione
govway_message_id		Identificativo del messaggio assegnato da GovWay
govway_relates_to		Identificativo del messaggio riferito
govway_conversation_id		Identificativo della conversazione
govway_transaction_id		Identificativo della transazione assegnato da GovWay
govway_sender_type		Codice che identifica il tipo del mittente
govway_sender		Identificativo del mittente
govway_provider_type		Codice che identifica il tipo del destinatario
govway_provider		Identificativo del destinatario
govway_service_type		Codice che identifica il tipo del servizio
govway_service		Identificativo del servizio
govway_service_version		Progressivo di versione del servizio
govway_action		Identificativo dell'azione
govway_application_mess		Identificativo del messaggio assegnato dall'applicativo
govway_application		Identificativo dell'applicativo identificato tramite autenticazione trasporto
govway_token_sender_type		Codice che identifica il tipo del dominio mittente dell'applicativo identificato tramite autenticazione token
govway_token_sender		Identificativo del dominio mittente dell'applicativo identificato tramite autenticazione token
govway_token_applicatior		Identificativo dell'applicativo identificato tramite autenticazione token

- *Header SOAP GovWay*: le informazioni precedentemente descritte vengono incluse come attributi in uno specifico header SOAP proprietario di GovWay che possiede il nome *integration* associato al namespace <http://govway.org/integration>. Di seguito un esempio di tale header:

```

<gw:integration
    ...
    transactionId="a2c6fd66-ec0b-407c-8a21-25b4920e7c73"
    SOAP_ENV:actor="http://govway.org/integration"
    SOAP_ENV:mustUnderstand="0"
    xmlns:SOAP_ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:gw="http://govway.org/integration"/>

```

Nella tabella [Tabella 10.8](#) vengono descritti i nome degli attributi.

Table10.8: Informazioni generate dal gateway nell'header soap proprietario di GovWay

Nome Attributo	Descrizione
messageId	Identificativo del messaggio assegnato da GovWay
relatesTo	Identificativo del messaggio riferito
conversationId	Identificativo della conversazione
transactionId	Identificativo della transazione assegnato da GovWay
senderType	Codice che identifica il tipo del mittente
sender	Identificativo del mittente
providerType	Codice che identifica il tipo del destinatario
provider	Identificativo del destinatario
serviceType	Codice che identifica il tipo del servizio
service	Identificativo del servizio
serviceVersion	Progressivo di versione del servizio
action	Identificativo dell'azione
applicationMessage	Identificativo del messaggio assegnato dall'applicativo
application	Identificativo dell'applicativo
tokenSenderType	Codice che identifica il tipo del dominio mittente dell'applicativo identificato tramite autenticazione token
tokenSender	Identificativo del dominio mittente dell'applicativo identificato tramite autenticazione token
tokenApplication	Identificativo dell'applicativo identificato tramite autenticazione token

Nota: Utilizzabile solamente con API di tipologia SOAP

- *WS-Addressing*: all'interno del messaggio Soap vengono generati gli header *To*, *From*, *Action*, *MessageID* e *RelatesTo* associati al namespace <http://www.w3.org/2005/08/addressing>. I valori utilizzati per i vari header sono i seguenti:

– *To*

```

http://<providerType>_<provider>.govway.org/services/<serviceType>_<service>/<serviceVersion>

```

– *From*

```

http://[<application>.]<senderType>_<sender>.govway.org

```

– *Action*

```
http://<providerType>_<provider>.govway.org/services/<serviceType>_<service>/<serviceVersion>/<action>
```

- *MessageID* di Protocollo, ritornato in una risposta di una fruizione o inserito nella consegna della richiesta di una erogazione

```
uuid:<messageId>
```

- *MessageID* di Integrazione, atteso nella richiesta inviata dal client in una fruizione o nella risposta ritornata dal backend in una erogazione. Viene utilizzato ad es. per la funzionalità di correlazione applicativa

```
uuid:<applicationMessageId>
```

- *RelatesTo*

```
uuid:<relatesTo>
```

Nota: Utilizzabile solamente con API di tipologia SOAP

- *Template:* consente di definire tramite un template freemarker o velocity come le informazioni siano inserite nel messaggio di richiesta, di risposta o in entrambi. Il tipo di template (freemarker/velocity) e il path del file template possono essere specifici per singole API indicandoli nelle proprietà “integrazione.template.richiesta/risposta.tipo” e “integrazione.template.richiesta/risposta.file”. In alternativa è possibile definire il tipo e il file template a livello globale agendo sul file locale di configurazione *govway_local.properties* tramite la definizione delle proprietà “org.openscoop2.pdd.integrazione.template.<pd/pa>.<request/response>.tipo” e “org.openscoop2.pdd.integrazione.template.<pd/pa>.<request/response>.file”.
- *Header HTTP di Autenticazione:* consente di generare Header HTTP utilizzabili dal backend per autenticare l’API Gateway. I nomi degli header generati ed i loro valori possono essere specifici per singole API indicandoli nelle proprietà “integrazione.autenticazione.headers” e “integrazione.autenticazione.header.<NOME_HEADER>”. In alternativa è possibile definire gli header a livello globale agendo sul file locale di configurazione *govway_local.properties* tramite la definizione delle proprietà “org.openscoop2.pdd.integrazione.autenticazione.<pd/pa>.request.headers” e “org.openscoop2.pdd.integrazione.autenticazione.<pd/pa>.request.header.<NOME_HEADER>”.
- *OpenSPCoop 2.x o OpenSPCoop2 1.x:* sono disponibili header di integrazione compatibili con le versioni di OpenSPCoop 2.x e 1.x:
 - Header HTTP: le informazioni sono veicolate all’interno di header HTTP. È possibile indicare se i nomi degli header debbano possedere o meno il prefisso “X-“;
 - Parametri Url: le informazioni sono veicolate come parametri della url
 - Header SOAP: le informazioni sono incluse in uno specifico header SOAP proprietario di OpenSPCoop 2.x o 1.x

10.6.5 Scambio di informazioni tramite un token JSON

Oltre alle modalità di interscambio di informazioni standard, descritte nelle precedenti sezioni, il client può fornire altre informazioni al gateway tramite un json il cui formato non è prestabilito da GovWay ma può essere definito in maniera arbitraria dal client.

Nella configurazione di default, GovWay si attende il json all'interno dell'header http "GovWay-Integration" codificato in base64. La presenza dell'header http non è obbligatoria ma se presente il json viene acceduto e le informazioni presenti al suo interno vengono rese disponibili tramite la keyword "integration" come informazione dinamica descritta nella sezione *Valori dinamici*. Le informazioni possono essere poi utilizzate nelle varie funzionalità del gateway nelle quali è possibile utilizzare i *Valori dinamici* come ad esempio nella generazione di token di sicurezza ModI descritti nella sezione *Payload Claims del token JWT*.

È possibile configurare una modalità di scambio differente registrando le seguenti *Proprietà* sull'erogazione o sulla fruizione:

- *integrationInfo.enabled* : consente di disabilitare la lettura del json di integrazione. I valori associabili alla proprietà sono "true" o "false". Per default questo controllo è abilitato.
- *integrationInfo.type* : consente di indicare dove risiede il json di integrazione. I valori associabili alla proprietà sono "http_header" (default) o "query_parameter".
- *integrationInfo.name* : nome dell'header http o del parametro della url dove risiede il json di integrazione (default: "GovWay-Integration").
- *integrationInfo.encode* : tipo di codifica utilizzata per trasmettere il json di integrazione. I valori associabili alla proprietà sono:
 - "base64" (default)
 - "hex"
 - "jwt": json atteso come payload del jwt
 - "plain": nessuna codifica
- *integrationInfo.required* : indica se l'header http o il parametro della url deve essere obbligatoriamente presente nella richiesta. I valori associabili alla proprietà sono "true" o "false". Per default questo controllo è disabilitato.

Inoltre anche tra le informazioni restituite nella risposta dal server a GovWay può essere presente l'header http "GovWay-Integration". Come per la richiesta la sua presenza non è obbligatoria e le informazioni presenti al suo interno vengono rese disponibili tramite la keyword "integrationResponse" come informazione dinamica descritta nella sezione *Valori dinamici*.

Anche per la risposta è possibile configurare una modalità di scambio differente registrando le seguenti *Proprietà* sull'erogazione o sulla fruizione:

- *responseIntegrationInfo.enabled* : consente di disabilitare la lettura del json di integrazione della risposta. I valori associabili alla proprietà sono "true" o "false". Per default questo controllo è abilitato.
- *responseIntegrationInfo.name* : nome dell'header http dove risiede il json di integrazione (default: "GovWay-Integration").
- *responseIntegrationInfo.encode* : tipo di codifica utilizzata per trasmettere il json di integrazione. I valori associabili sono gli stessi utilizzabili sulla richiesta: "base64", "hex", "jwt" e "plain".
- *responseIntegrationInfo.required* : indica se l'header http della risposta deve essere obbligatoriamente presente. I valori associabili alla proprietà sono "true" o "false". Per default questo controllo è disabilitato.

10.7 Connettori

I connettori rappresentano le entità di configurazione che consentono a GovWay di indirizzare le comunicazioni verso gli attori dei flussi di erogazione/fruizione gestiti. Nel nostro contesto possiamo distinguere due tipologie di comunicazioni:

- *GovWay —> Applicativo Esterno*, nel caso di fruizioni
- *GovWay —> Applicativo Interno*, nel caso di erogazioni

I connettori di GovWay permettono di configurare differenti aspetti della comunicazione http:

- *Autenticazione http*: tale funzionalità permette di impostare delle credenziali http basic (username e password).
- *Autenticazione token*: tale funzionalità permette di inoltrare un Bearer Token.
- *Autenticazione API Key*: funzionalità che consente di inoltrare al backend una chiave di identificazione “Api Key” veicolata all’interno di un header http “X-API-KEY” (<https://swagger.io/docs/specification/authentication/api-keys/>). È possibile abilitare anche la modalità “App ID” che prevede oltre all’ApiKey un identificatore dell’applicazione oltre a personalizzare i nomi degli header http utilizzati.
- *Autenticazione https*: se l’utente lo desidera può personalizzare tutti gli aspetti che riguardano una comunicazione sicura su https.
- *Proxy*: è possibile configurare un proxy http che media la comunicazione.
- *Ridefinisci Tempi Risposta*: permette di ridefinire i tempi di risposta che sono stati configurati a livello generale, nell’ambito del controllo del traffico (vedi sezione *Tempi Risposta*).

Attivando la *modalità avanzata* dell’interfaccia saranno inoltre disponibili le seguenti opzioni:

- *Data Transfer Mode*: tramite questa configurazione è possibile indicare se la comunicazione deve avvenire in modalità transfer-encoding-chunked (streaming) o content length fisso.
- *Redirect*: tramite questa configurazione è possibile indicare se un eventuale redirect ritornato dal server contattato deve essere seguito o meno.
- *Debug*: è possibile abilitare un log verboso di tutta la comunicazione.

La govwayConsole, tramite l’interfaccia in modalità *avanzata*, consente anche di configurare le comunicazioni attraverso connettori non basati sul protocollo HTTP (o HTTPS). GovWay offre built-in i seguenti ulteriori connettori:

- *JMS*: connettore basato sul protocollo JMS
- *File*: connettore che permette di serializzare il messaggio di richiesta su FileSystem ed opzionalmente generare una risposta.
- *Null*: connettore per test. Si comporta come un servizio Oneway ricevendo richieste senza rispondere
- *NullEcho*: connettore per test. Si comporta come un servizio Sincrono rispondendo con un messaggio identico alla richiesta

Nel seguito vengono descritte alcune funzionalità specifiche dei connettori HTTP e HTTPS. Inoltre viene fornita una descrizione del connettore built-in JMS e File.

10.7.1 Autenticazione Http

Quando si configura l'autenticazione per un connettore è possibile scegliere la modalità di autenticazione HTTP-BASIC. Tale funzionalità permette di impostare delle credenziali (username e password) che verranno iniettate nella comunicazione http tramite header “Authorization” (<https://tools.ietf.org/html/rfc2617#section-2>).

Connettore

- Abilitato
- Endpoint *
- Autenticazione Http
- AutenticazioneHttps
- Proxy
- Ridefinisci Tempi Risposta

Autenticazione Http

- Utente *
- Password *

Figure 10.10: Dati di configurazione di un'autenticazione Http

10.7.2 Autenticazione Token

Quando si configura l'autenticazione per un connettore è possibile scegliere la modalità di autenticazione per token. Tale funzionalità permette di iniettare un Token Bearer nella comunicazione http tramite la modalità definita all'interno della policy selezionata (es. tramite header “Authorization”). Per ulteriori dettagli su come registrare una policy di negoziazione del Bearer Token si rimanda alla sezione [Token Policy Negoziazione](#).

10.7.3 Autenticazione API Key

Quando si configura l'autenticazione per un connettore è possibile scegliere la modalità di autenticazione API Key che consente di inoltrare al backend una chiave di identificazione “Api Key” veicolata all'interno di un header http “X-API-KEY” (<https://swagger.io/docs/specification/authentication/api-keys/>).

La configurazione consente inoltre di abilitare la modalità “App ID” che prevede oltre all’ApiKey un identificatore dell'applicazione veicolato nell'header http “X-APP-ID”; modalità denominata “Multiple API Keys” nella specifica “OAS3 API Keys”.

Infine è possibile ridefinire i nomi degli header http utilizzati non selezionando l'opzione “Nomi Standard OAS3”.

Connettore

Endpoint *

Autenticazione Token

AutenticazioneHttps

Proxy

Ridefinisci Tempi Risposta

Autenticazione Token

Policy *

Figure10.11: Dati di configurazione di un'autenticazione Token

Connettore

Utilizza Applicativo Server

Endpoint * ⓘ

Autenticazione Http

Autenticazione Token

Autenticazione API Key

AutenticazioneHttps

Proxy

Ridefinisci Tempi Risposta

Autenticazione API Key

Nomi Standard OAS3

App ID

API Key *

Figure10.12: Dati di configurazione di un'autenticazione API Key

Connettore

Utilizza Applicativo Server

Endpoint * i

Autenticazione Http

Autenticazione Token

Autenticazione API Key

AutenticazioneHttps

Proxy

Ridefinisci Tempi Risposta

Autenticazione API Key

Nomi Standard OAS3

App ID

API Key *

App ID *

Figure10.13: Dati di configurazione di un'autenticazione API Key + App ID

Autenticazione API Key

Nomi Standard OAS3

App ID

API Key

Header HTTP *

Valore *

App ID

Header HTTP *

Valore *

Figure10.14: Dati di configurazione di un'autenticazione API Key + App ID con personalizzazione dei nomi degli header HTTP

10.7.4 Autenticazione Https

Per un endpoint https viene utilizzata per default la configurazione https impostata nella JVM dell'Application Server (proprietà javax.net.ssl.*) descritta nella sezione `install_ssl_client_direct`.

È possibile configurare GovWay in modo che utilizzi una configurazione https differente da quella ereditata dalla JVM (es. keystore e truststore, versione TLS ...) attraverso l'abilitazione dell'autenticazione https come mostrata in figura [Fig. 10.15](#).

Un'alternativa alla soluzione descritta in questa sezione viene documentata nella sezione [Repository delle configurazioni https](#), dove la configurazione avviene tramite file di proprietà.

L'autenticazione https personalizzata consente di agire ai seguenti livelli di autenticazione:

- **Autenticazione Server**, è possibile definire le trusted keys e indicare se si desidera verificare l'hostname rispetto al certificato server contenuto nella sessione SSL.
- **Autenticazione Client**, è opzionale; se abilitata permette di definire il keystore contenente la chiave privata che si deve utilizzare durante la sessione SSL.

Facendo riferimento alla maschera raffigurata in [Fig. 10.15](#) andiamo a descrivere il significato dei parametri:

- *Connettore*

- **Url**: indirizzo endpoint del connettore
- **Tipologia** (es. TLSv1.2): Tipo e versione del protocollo di trasporto. Sono selezionabili tutti i tipi supportati dalla versione della jvm utilizzata.
- **Verifica Hostname** (true/false): Attiva la verifica in fase di autenticazione server della corrispondenza tra l'hostname indicato nella url e quello presente nel certificato server ritornato dal server (nel subject CN=hostname)

- *Autenticazione Server*

I certificati server saranno validati tramite la configurazione indicata di seguito. Per accettare qualsiasi certificato restituito dal server è possibile disattivare la **Verifica**.

- **Tipo** (jks, pkcs12): Tipologia del TrustStore (default: jks). Se registrati saranno disponibili anche i tipi di keystore PKCS11; per ulteriori dettagli si rimanda alla sezione [Device PKCS11](#).
- **Path**: Path dove è localizzato il truststore contenente i certificati server trusted.
- **Password**: Password per l'accesso al TrustStore.
- **OCSP Policy**: Policy OCSP da utilizzare per validare il certificato server; per ulteriori dettagli si rimanda alla sezione [Online Certificate Status Protocol \(OCSP\)](#).
- **CRL File(s)**: Path dove è presente una CRL da utilizzare per validare i certificati server. L'indicazione di una CRL è opzionale e ne possono essere indicate più di una separando i path con la virgola.

- *Autenticazione Client (opzionale)*

Abilitando la checkbox **Abilitato** è possibile configurare il certificato client che verrà inoltrato al server.

- **Dati di Accesso al KeyStore** (usa valori del TrustStore, Ridefinisci): Consente di riutilizzare i medesimi riferimenti del TrustStore anche per il KeyStore o in alternativa ridefinirli.
- **Tipo (solo se Dati di Accesso ridefiniti)** (jks, pkcs12): Tipologia del Keystore (default: jks). Se registrati saranno disponibili anche i tipi di keystore PKCS11; per ulteriori dettagli si rimanda alla sezione [Device PKCS11](#).
- **Path (solo se Dati di Accesso ridefiniti)**: Path dove è localizzato il Keystore.
- **Password (solo se Dati di Accesso ridefiniti)**: Password per l'accesso al Keystore.

Connettore

Utilizza Applicativo Server

Endpoint *

Autenticazione Http

Autenticazione Token

AutenticazioneHttps

Proxy

Ridefinisci Tempi Risposta

Autenticazione Https

Tipologia

Verifica Hostname

Autenticazione Server

Verifica

Path *

Tipo

Password *

CRL File(s)

Elencare più file separandoli con la ;

Autenticazione Client

Abilitato

Dati Accesso al KeyStore

Path *

Tipo

Password *

Password Chiave Privata *

Alias Chiave Privata

Figure10.15: Dati di configurazione di un'autenticazione Https

- **Password Chiave Privata:** Password per accedere alla chiave privata presente nel keystore.
- **Alias Chiave Privata:** Alias che individua la chiave privata, presente nel keystore, da utilizzare. L'indicazione di un alias è opzionale e se non fornito viene utilizzata la prima chiave trovata.

10.7.5 Repository delle configurazioni https

Per un endpoint https viene utilizzata per default la configurazione https impostata nella JVM dell'Application Server (proprietà `javax.net.ssl.*`) descritta nella sezione `install_ssl_client_direct`.

È possibile configurare GovWay in modo che utilizzi una configurazione https differente o definendo tramite la console di gestione un'autenticazione personalizzata (soluzione descritta nella sezione [Autenticazione Https](#)) o creando un repository di configurazioni definite tramite file di proprietà. Quest'ultima modalità viene descritta in questa sezione.

Per abilitare l'utilizzo di un repository delle configurazioni https devono essere registrate le seguenti *Proprietà* sull'erogazione o sulla fruizione:

- `connettori.httpsEndpoint.jvmConfigOverride.enabled`: consente di abilitare o disabilitare la funzionalità; i valori associabili alle proprietà sono “true” o “false”. Per default la funzionalità è disabilitata.
- `connettori.httpsEndpoint.jvmConfigOverride.repository`: consente di indicare la directory contenente i file di proprietà (default: “/etc/govway/https/erogazioni” per le erogazioni e “/etc/govway/https/fruizioni” per le fruizioni).
- `connettori.httpsEndpoint.jvmConfigOverride.config`: consente di indicare il nome del file di proprietà da utilizzare che deve esistere all'interno del repository riferito nella precedente proprietà (default: “<TIPO_EROGATORE><NOME_EROGATORE>.properties” per le erogazioni e “<TIPO_FRUITORE><NOME_FRUITORE>.properties” per le fruizioni). Il nome del file indicato può contenere delle macro, risolte a runtime dal gateway, per creare dei path dinamici (per ulteriori dettagli si rimanda alla sezione [Valori dinamici](#)).

I valori di default, per quanto concerne l'abilitazione della funzionalità e il repository delle configurazione, possono essere impostati anche a livello globale nel file “govway_local.properties” tramite le proprietà descritte di seguito che variano per erogazioni e fruizioni.

Proprietà a livello globale per le erogazioni:

```
org.openspcoop2.pdd.connettori.consegnaContenutiApplicativi.http.urlHttps.  
↳overrideDefaultConfiguration=false  
org.openspcoop2.pdd.connettori.consegnaContenutiApplicativi.http.urlHttps.  
↳repository=<directory-lavoro>/https/erogazioni
```

Proprietà a livello globale per le fruizioni:

```
org.openspcoop2.pdd.connettori.inoltroBuste.http.urlHttps.  
↳overrideDefaultConfiguration=false  
org.openspcoop2.pdd.connettori.inoltroBuste.http.urlHttps.repository=  
↳<directory-lavoro>/https/fruizioni
```

Proprietà per la personalizzazione della configurazione https

Le proprietà che consentono di personalizzare l'autenticazione server sono le seguenti:

- `trustAllCerts` (true/false): se abilitata non viene attuata l'autenticazione del certificato server e viene accettato qualsiasi certificato restituito dal server;
- `trustStoreLocation`: path dove è localizzato il TrustStore contenente i certificati server trusted;
- `trustStoreType` (jks, pkcs12): tipologia del TrustStore; se registrati potranno essere indicati anche i tipi di keystore PKCS11 (per ulteriori dettagli si rimanda alla sezione [Device PKCS11](#));

- trustStorePassword: password per l’accesso al TrustStore;
- trustStoreOCSPPolicy: policy OCSP da utilizzare per validare il certificato server; per ulteriori dettagli si rimanda alla sezione *Online Certificate Status Protocol (OCSP)*;
- trustStoreCRLs: path dove è presente una CRL da utilizzare per validare i certificati server; ne possono essere indicate più di una separando i path con la virgola;
- trustManagementAlgorithm: consente di indicare un algoritmo differente da quello di default (PKIX).

Le proprietà che consentono di personalizzare l’autenticazione client, indicando il keystore contenente la chiave privata che si deve utilizzare durante la sessione TLS, sono le seguenti:

- keyStoreLocation: path dove è localizzato il Keystore;
- keyStoreType (jks, pkcs12): tipologia del Keystore; se registrati potranno essere indicati anche i tipi di keystore PKCS11 (per ulteriori dettagli si rimanda alla sezione *Device PKCS11*);
- keyStorePassword: password per l’accesso al Keystore;
- keyPassword: password per accedere alla chiave privata presente nel keystore;
- keyAlias: alias che individua la chiave privata, presente nel keystore, da utilizzare. L’indicazione di un alias è opzionale e se non fornito viene utilizzata la prima chiave trovata.
- keyManagementAlgorithm: consente di indicare un algoritmo differente da quello di default (SunX509).

Sono inoltre disponibili le seguenti proprietà:

- hostnameVerifier (true/false): attiva la verifica in fase di autenticazione server della corrispondenza tra l’hostname indicato nella url e quello presente nel certificato server ritornato dal server (nel subject CN=<hostname>).
- sslType: tipo e versione del protocollo di trasporto (es. TLSv1.2). Sono selezionabili tutti i tipi supportati dalla versione della jvm utilizzata.
- secureRandom (true/false): indicazione se deve essere utilizzato un “Secure Random”.
- secureRandomAlgorithm: consente di indicare un algoritmo “secure random” differente da quello di default.

10.7.6 Proxy

Funzionalità che consente di configurare un proxy http che media la comunicazione. Oltre ai classici parametri hostname e porta, è possibile anche indicare delle credenziali http basic (username e password) che verranno iniettate nella comunicazione http tramite header “Proxy-Authorization”.

10.7.7 Tempi Risposta

Tramite questa sezione è possibile ridefinire i tempi di risposta che sono stati configurati a livello generale, nell’ambito del controllo del traffico (vedi sezione *Tempi Risposta*).

Connettore

Abilitato	<input checked="" type="checkbox"/>
Endpoint *	<input type="text" value="https://127.0.0.1:8080/TestService/echo"/>
Autenticazione Http	<input type="checkbox"/>
AutenticazioneHttps	<input type="checkbox"/>
Proxy	<input checked="" type="checkbox"/>
Ridefinisci Tempi Risposta	<input type="checkbox"/>

Proxy

Hostname *	<input type="text" value="proxy"/>
Porta *	<input type="text" value="8080"/>
Username	<input type="text"/>
Password	<input type="text"/>

Figure10.16: Dati di configurazione di un Proxy Http

10.7.8 Configurazione Http Avanzata

Richiede accesso alla govwayConsole in modalità *avanzata*

Tramite questa sezione è possibile indicare sia quale modalità di comunicazione (streaming o meno) deve essere utilizzata, sia se deve avvenire una eventuale gestione dei redirect http.

Facendo riferimento alla maschera raffigurata in Fig. 10.17 andiamo a descrivere il significato dei parametri:

- *Data Transfer Mode* tramite questa configurazione è possibile indicare se la comunicazione deve avvenire in modalità transfer-encoding-chunked (streaming) o con content length fisso.
 - **Modalità Data Transfer** (default, content-length, transfer-encoding-chunked): indica il tipo di trasferimento dati; scegliendo la voce default verrà utilizzato il comportamento configurato a livello globale nel file govway_local.properties tramite le opzioni:
 - * org.openspcoop2.pdd.connettori.inoltroBuste.httpTransferLength
 - * org.openspcoop2.pdd.connettori.consegnaContenutiApplicativi.httpTransferLength
 - **Chunk Length (Bytes)** (presente solamente se la modalità è transfer-encoding-chunked): indica la dimensione in bytes di ogni singolo http chunk.
- *Redirect* tramite questa configurazione è possibile indicare se un eventuale redirect ritornato dal server contattato deve essere seguito o meno.
 - **Gestione Redirect** (default, abilitato, disabilitato): consente di personalizzare il comportamento sul singolo connettore; scegliendo la voce default verrà utilizzato il comportamento configurato a livello globale nel file govway_local.properties tramite le opzioni:

Connettore

Tipo	http
Debug	<input type="checkbox"/>
Endpoint *	http://127.0.0.1:8080/TestService/echo
Autenticazione Http	<input type="checkbox"/>
Proxy	<input type="checkbox"/>
Ridefinisci Tempi Risposta	<input type="checkbox"/>
Opzioni Avanzate	<input checked="" type="checkbox"/>

Opzioni Avanzate

Modalità Data Transfer	transfer-encoding-chunked
Chunk Length (Bytes)	
Gestione Redirect	abilitato
Max Numero di Redirect	

Figure10.17: Configurazione http avanzata

- * org.openspcoop2.pdd.connettori.inoltroBuste.followRedirects
 - * org.openspcoop2.pdd.connettori.consegnaContenutiApplicativi.followRedirects
- **Massimo Numero di Redirect** (presente solamente se la gestione redirect è abilitata): indica il massimo numero di redirect seguiti.

10.7.9 Debug

Richiede accesso alla govwayConsole in modalità *avanzata* (sezione *Modalità Avanzata*).

Se viene abilitato il debug, GovWay produce un log verboso di tutta la comunicazione nel file

- */var/log/govway/govway_connettori.log*

(assumendo che */var/log/govway* sia la directory di logging configurata)

The screenshot shows a configuration dialog for a connector. The 'Connettore' tab is selected. The 'Tipo' dropdown is set to 'http'. The 'Debug' checkbox is checked. The 'Endpoint' field contains 'http://127.0.0.1:8080/TestService/echo'. There are several other checkboxes for 'Autenticazione Http', 'Proxy', 'Ridefinisci Tempi Risposta', and 'Opzioni Avanzate', all of which are unchecked.

Figure10.18: Debug

10.7.10 Header HTTP

MIME encoded-word encoding

I valori degli header che includono caratteri non ASCII vengono codificati per default attraverso la codifica MIME encoded-word definita nel RFC 2047 con encoding style "Q" (Quoted-Printable).

È possibile disabilitare la codifica o personalizzarne lo stile su una singola erogazione o fruizione di API attraverso la definizione delle seguenti *Proprietà*:

- *connettori.header.value.encodingRFC2047.enabled* (true/false default:true): consente di abilitare o disabilitare la codifica;
- *connettori.header.value.encodingRFC2047.type* (Q/B default:Q): specifica lo stile di codifica da utilizzare tra Quoted-Printable o Base64;
- *connettori.header.value.encodingRFC2047.charset* (default: US-ASCII): consente di indicare il charset da utilizzare per determinare se una codifica del valore di un header è necessaria o meno;

È inoltre possibile definire delle proprietà specifiche per il trattamento delle richieste o delle risposte che sovrascrivono il comportamento indicato nelle proprietà generiche:

- *connettori.header.value.encodingRFC2047.request.enabled*;
- *connettori.header.value.encodingRFC2047.response.enabled*;
- *connettori.header.value.encodingRFC2047.request.type*;
- *connettori.header.value.encodingRFC2047.response.type*;
- *connettori.header.value.encodingRFC2047.request.charset*;
- *connettori.header.value.encodingRFC2047.response.charset*.

RFC 7230 section 3.2. Header Fields

Viene attuata una validazione sia del nome che del valore di un header, in conformità con quanto indicato nella specifica “RFC 7230 - sezione 3.2”. Se viene rilevato un errore, l’header HTTP non verrà inoltrato.

È possibile disabilitare la validazione su una singola erogazione o fruizione di API attraverso la definizione della seguente *Proprietà*:

- *connettori.header.validation.enabled* (true/false default:true): consente di abilitare o disabilitare la validazione;

È inoltre possibile definire delle proprietà specifiche per il trattamento delle richieste o delle risposte che sovrascrivono il comportamento indicato nelle proprietà generiche:

- *connettori.header.validation.request.enabled*;
- *connettori.header.validation.response.enabled*.

10.7.11 Connettore JMS

Il connettore JMS consente di configurare i parametri per abilitare la comunicazione tra GovWay e gli applicativi attraverso il protocollo JMS.

In Fig. 10.19 è mostrata la maschera di configurazione del connettore JMS.

In riferimento alla Fig. 10.19 descriviamo in dettaglio il significato dei campi per la configurazione:

- **Nome**: identificatore JNDI della risorsa queue/topic JMS
- **Tipo** (Queue/Topic): Si specifica se la risorsa JMS è di tipo queue o topic
- **Send As** (TextMessage/BytesMessage): Si sceglie la codifica del messaggio da inviare tramite broker JMS, tra TextMessage e BytesMessage.
- **Utente**: Username relativo alle credenziali per l’autenticazione e la negoziazione di una connessione sul Broker JMS
- **Password**: Password relativa alle credenziali per l’autenticazione e la negoziazione di una connessione sul Broker JMS
- **Connection Factory**: Identificatore della risorsa JNDI per la creazione di una connessione verso il broker JMS
- **Initial Context Factory**: Class Name per l’inizializzazione del server JNDI per la lookup della Connection Factory e della Coda
- **Url Pkg Prefixes**: Lista sperata da “:” per specificare i prefissi dei package da utilizzare per l’inizializzazione del Context JNDI
- **Provider Url**: Indirizzo che localizza il server JNDI

Connettore

Tipo	<input type="text" value="jms"/> <input type="button" value="▼"/>
Debug	<input type="checkbox"/>
Ridefinisci Tempi Risposta	<input type="checkbox"/>

Dati Configurazione Coda

Nome *	<input type="text" value="http://127.0.0.1:8080/TestService/echo"/> <input type="button" value="▼"/>
Tipo	<input type="text" value="queue"/> <input type="button" value="▼"/>
Send As	<input type="text" value="TextMessage"/> <input type="button" value="▼"/>

Dati Configurazione Connessione

Connection Factory *	<input type="text"/>
Utente	<input type="text"/>
Password	<input type="text"/>

Contesto JNDI

Initial Context Factory	<input type="text"/>
Url Pgk Prefixes	<input type="text"/>
Provider Url	<input type="text"/>

Figure10.19: Dati di configurazione di un connettore JMS

10.7.12 Connettore File

Il connettore permette di serializzare la richiesta su FileSystem ed opzionalmente di generare una risposta.

Il connettore File supporta:

- **Richiesta**, è possibile serializzare il messaggio di richiesta su file-system fornendo un path che può contenere anche parti dinamiche risolte a runtime da GovWay. È permesso anche abilitare l'eventuale sovrascrittura del file, se risulta già esistente, e la creazione automatica delle directory padre, se non esistono.
- **Risposta**, è opzionale; se abilitata permette di generare una risposta costruita utilizzando il contenuto di un file indirizzabile a sua volta tramite gli stessi meccanismi dinamici della richiesta. Il file contenente la risposta può essere eliminato una volta consumato (opzione configurabile). L'utente può inoltre indicare un tempo di attesa (ms) qualora il file non sia immediatamente disponibile.

Facendo riferimento alla maschera raffigurata in Fig. 10.20 andiamo a descrivere il significato dei parametri:

- *Richiesta*
 - **File**: indirizzo su file-system (path) dove verrà serializzato il messaggio di richiesta. È possibile fornire delle macro per creare dei path dinamici (per ulteriori dettagli si rimanda alla sezione *Valori dinamici*).
 - **File (Permessi)**: consente di impostare i permessi del file, indicato nel campo precedente, tramite il formato “[o/a]+/-rwx”.
 - **File Headers** (opzionale): indirizzo su file-system (path) dove verranno serializzati gli header di trasporto associati alla richiesta. È possibile fornire delle macro per creare dei path dinamici (per ulteriori dettagli si rimanda alla sezione *Valori dinamici*).
 - **File Headers (Permessi)**: consente di impostare i permessi del file, indicato nel campo precedente, tramite il formato “[o/a]+/-rwx”.
 - **Overwrite If Exists** (true/false): abilita l'eventuale sovrascrittura del file, se risulta già esistere.
 - **AutoCreate Parent Directory** (true/false): abilita la creazione automatica delle directory padre, se non esistono.
- *Risposta (Opzionale)*
 - **Generazione** (true/false): abilita la generazione di una risposta. Tutte le successive opzioni della sezione “Risposta” sono configurabili solamente se la generazione è abilitata.
 - **File**: indirizzo su file-system (path) dove verrà letto il messaggio di risposta. È possibile fornire delle macro per creare dei path dinamici (per ulteriori dettagli si rimanda alla sezione *Valori dinamici*).
 - **File Headers** (opzionale): indirizzo su file system (path) dove verranno letti gli header di trasporto da associare alla risposta. È possibile fornire delle macro per creare dei path dinamici (per ulteriori dettagli si rimanda alla sezione *Valori dinamici*).
 - **Delete After Read** (true/false): abilita l'eventuale eliminazione del file una volta utilizzato per la generazione della risposta.
 - **Wait Time If Not Exists (ms)** (opzionale): indica un tempo di attesa (ms) qualora il file per la generazione della risposta non sia immediatamente disponibile.
- *Informazioni Dinamiche*. Per creare dei path dinamici rispetto alla transazione in corso di elaborazione, possono essere utilizzate le macro descritte nella sezione sezione *Valori dinamici*. Di seguito vengono riportati solo alcuni esempi:
 - **{date:FORMAT}** indica la data di elaborazione del messaggio. Il formato fornito deve essere conforme a quanto richiesto dalla classe java “java.text.SimpleDateFormat”. Ad esempio: {date:yyyyMMdd_HHmmssSSS}.
 - **{transaction:id}** indica l'identificativo della transazione (UUID).

Connettore

Utilizza Applicativo Server

Tipo

Consente di salvare la richiesta su filesystem e restituire una risposta

Debug govway_connatori.log

Richiesta

File *****

File (Permessi)

File Headers

File Headers (Permessi)

AutoCreate Parent Dir

Overwrite If Exists

Risposta

Generazione

File *****

File Headers

Delete After Read

WaitTime ifNotExists (ms)

Figure10.20: Dati di configurazione di un connettore File

- **{busta:FIELD}** permette di utilizzare informazioni di protocollo riguardanti la transazione in corso. Il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe di openscoop “org.openscoop2.protocol.sdk.Busta”. Ad esempio per ottenere il mittente della busta usare **{busta:mittente}**.
- **{header:NAME}** permette di utilizzare informazioni, relative alla transazione in corso, inserite negli header http generati da GovWay (maggiori dettagli in sezione *Header di Integrazione*). Il valore “NAME” indica il nome dell’header da utilizzare. Ad esempio per utilizzare il nome del mittente è possibile usare **{header:GovWay-Sender}**. Un altro esempio valido nello scenario della fatturazione elettronica (sezione *Profilo “Fatturazione Elettronica”*) potrebbe essere quello di utilizzare il nome originale del file fattura utilizzando la sintassi **{header:GovWay-SDI-NomeFile}**
- **{query:NAME}** permette di utilizzare informazioni, relative alla transazione in corso, inserite nei query parameter aggiunti all’endpoint da GovWay (maggiori dettagli in sezione *Header di Integrazione*). Il valore “NAME” indica il nome della proprietà da utilizzare. Ad esempio per utilizzare il nome del mittente è possibile usare **{query:govway_sender}**.
- **{property:NAME}** permette di utilizzare informazioni, relative alla transazione in corso, specifiche della sezione relativa al profilo utilizzato all’interno della traccia (es. sezione “Informazioni Fatturazione Elettronica”). Il valore “NAME” indica il nome della proprietà da utilizzare. Un esempio valido nello scenario della fatturazione elettronica (sezione *Profilo “Fatturazione Elettronica”*) potrebbe essere quello di utilizzare l’identificativo sdi utilizzando la sintassi **{property:IdentificativoSdi}**

10.8 Device PKCS11

Nella sezione pkcs11Install del manuale di installazione è documentato come configurare GovWay per poter utilizzare token PKCS11.

Una volta registrati, i token saranno selezionabili tra i tipi di keystore disponibili (es. Fig. 10.21) per tutte le funzionalità che richiedono l’utilizzo di una chiave X.509.

Autenticazione Client	
Abilitato	<input checked="" type="checkbox"/>
Dati Accesso al KeyStore	Ridefinisci
Tipo	softhsm-example
Alias Chiave Privata	JKS PKCS12 softhsm-example
Algoritmo *	softhsm-example

Figure10.21: Esempio di configurazione di un token PKCS11 su connettore https

Nota: Le funzionalità che richiedono l’utilizzo della parte pubblica di un certificato X.509 non consentiranno di selezionare i keystore PKCS11 registrati, a meno che durante la registrazione non siano state abilitate le opzioni “usableAsTrustStore” e “usableAsSecretKeyStore”. Per maggiori dettagli si rimanda alla sezione pkcs11Install.

10.9 Online Certificate Status Protocol (OCSP)

Nella sezione `ocspInstall` del manuale di installazione è documentato come configurare GovWay per poter utilizzare policy OCSP.

Una volta registrate, le policy saranno selezionabili (es. Fig. 10.22) per tutte le funzionalità che richiedono una validazione di un certificato X.509.

Autenticazione Https

Tipologia: **TLSv1.3**

Verifica Hostname:

Autenticazione Server

Verifica:

OCSP Policy: **Certificate Only**

CRL File(s):

Elencare più file separandoli con la ','

Autenticazione Client

Abilitato:

Figure 10.22: Esempio di configurazione di una policy OCSP su connettore https

10.10 Correlazione tra transazioni differenti

Richiede accesso alla `govwayConsole` in modalità *avanzata* (sezione *Modalità Avanzata*).

Come descritto anche nella sezione *Configurazione manuale delle interfacce*, durante la configurazione di un API di tipo SOAP o REST è possibile specificare i parametri descritti di seguito rispettivamente in un servizio/azione o in una risorsa.

- *ID Conversazione*. Flag per consentire di specificare nelle richieste un valore che identifica una conversazione.
- *Riferimento ID Richiesta*. Flag per consentire di specificare nelle richieste un identificativo relativo ad un messaggio precedente.

Tali parametri consentono agli applicativi client di fornire tali informazioni tramite gli header di integrazione descritti nella sezione *Scambio di informazioni nella richiesta del client verso il gateway*

Le informazioni fornite saranno associate alla traccia della transazione gestita, e quindi utilizzabili in fase di monitoraggio tramite le modalità di ricerca basate su identificativi descritte nella Guida alla Console di Monitoraggio.

10.11 Opzioni Avanzate per Erogazioni/Fruizioni

A partire dall'erogazione o fruizione di una API, accedendo alla sezione Configurazione, descritta nella sezione *Configurazione dell'API*, in modalità avanzata compare una sezione precedentemente non documentata denominata *Opzioni Avanzate*.

All'interno di tale sezione è possibile configurare (Fig. 10.23):

- *Integrazione - Metadati*: per default non impostato, consente di attivare gli header di integrazione desiderati utilizzando le keyword, separate da virgola, descritta nella sezione *Altri header di Integrazione*.
- *Rate Limiting*: per default non impostato, consente di personalizzare le impostazioni che riguardano gli Header HTTP informativi restituiti ai client (vedi sezione *Header HTTP informativi restituiti ai client: quote e finestre temporali*) e il tipo di Rate Limiting in presenza di un cluster di nodi (vedi sezione *Rate Limiting in presenza di un cluster di nodi*).
- *Handlers*: consente di attivare handler sul pipeline relativo alla gestione delle richieste o delle risposte di GovWay.
- *SOAP With Attachments - Gestione Body*: presente solamente per API di tipo SOAP consente tramite la voce “allega” di spostare il contenuto presente nel body in un attachment o di eliminare il body dalla richiesta prima di inoltrare il messaggio.

10.12 Gestione Proxy

I connettori descritti nella sezione *Connettori* rappresentano le entità di configurazione che consentono a GovWay di indirizzare le comunicazioni verso gli attori dei flussi di erogazione/fruizione gestiti. Come già descritto in tale sezione possiamo distinguere due tipologie di comunicazioni:

- *GovWay —> Applicativo Esterno*, nel caso di fruizioni
- *GovWay —> Applicativo Interno*, nel caso di erogazioni

In alcune architetture potrebbe essere presente tra GovWay e l'applicativo da contattare un proxy che media le comunicazioni.

- *Proxy HTTP*, se la comunicazione è mediata da un proxy http l'indirizzo remoto dell'applicativo viene censito su GovWay e la mediazione tramite il proxy sarà trasparente seguendo le indicazioni di configurazione descritte nella sezione *Proxy*.
- *Proxy Applicativo*, in scenari più complessi possono essere presenti reverse proxy che intervengono nella gestione delle connessioni https, utilizzando certificati client e/o trustStore differenti per diversi contesti applicativi. In queste situazioni l'endpoint indicato nella configurazione del connettore su GovWay non è l'indirizzo remoto dell'applicativo ma bensì l'indirizzo del reverse proxy il quale a sua volta si occuperà di inoltrare la richiesta agli indirizzi a lui noti. In questa situazione, è necessario configurare gli endpoint delle API sia su GovWay (indirizzo del reverse proxy), che sul reverse proxy (indirizzo dell'Erogatore finale)

Per semplificare la gestione, in uno scenario architettonale con *Proxy Applicativo*, GovWay può passare l'indirizzo remoto dell'applicativo al proxy tramite un header HTTP o un parametro della url. In questo modo il censimento degli applicativi viene effettuato esclusivamente su GovWay.

Per abilitare e configurare la funzionalità “govway-proxy” si deve agire a livello di proprietà java, configurabili accedendo alla sezione “Configurazione Generale -> Proprietà di Sistema”, aggiungendo una proprietà “govway-proxy-enable” con valore “true” (Figura Fig. 10.24).

Una volta abilitata la funzionalità la configurazione è attuabile tramite le seguenti proprietà:

Opzioni Avanzate

Integrazione

Metadati ▼ ⓘ

Rate Limiting

Sincronizzazione ▼

HTTP Headers ▼

Handlers

Handlers per la Richiesta

[Pre-In \(precedente alla ricezione dei dati\) \(0\)](#)
[In \(dopo aver ricevuto i dati\) \(0\)](#)
[InProfileInfo \(dopo aver identificato i soggetti e l'API richiesta\) \(0\)](#)
[Out \(precedente all'inoltro dei dati\) \(0\)](#)
[Post-Out \(successiva all'inoltro dei dati\) \(0\)](#)

Handlers per la Risposta

[Pre-In \(precedente alla ricezione dei dati\) \(0\)](#)
[In \(dopo aver ricevuto i dati\) \(0\)](#)
[Out \(precedente all'inoltro dei dati\) \(0\)](#)
[Post-Out \(successiva all'inoltro dei dati\) \(0\)](#)

SOAP With Attachments

Gestione Body ▼

Figure10.23: Opzioni Avanzate di una API

Configurazione Generale > Proprietà di Sistema > Risultati ricerca

Risultati ricerca

Visualizzati record [1-1] su 1

	Nome	Valore
<input type="checkbox"/>	govway-proxy-enable	true

ELIMINA AGGIUNGI

Figure10.24: Configurazione delle Proprietà di Sistema

- govway-proxy: endpoint a cui verranno inoltrate le richieste. L'endpoint può contenere parti dinamiche che verranno risolte dal Gateway (per ulteriori dettagli fare riferimento alla sezione [Valori dinamici](#));
- govway-proxy-header: se configurato verrà utilizzato un header http, con il nome indicato, per inoltrare al proxy l'indirizzo remoto;
- govway-proxy-header-base64: nel caso sia stato configurato un header http, l'indirizzo remoto sarà codificato in base64 se viene abilitata la seguente proprietà;
- govway-proxy-query: se configurato verrà utilizzato un parametro della url, con il nome indicato, per inoltrare al proxy l'indirizzo remoto;
- govway-proxy-query-base64: nel caso sia stato configurato un parametro della url, l'indirizzo remoto sarà codificato in base64 se viene abilitata la seguente proprietà.

È inoltre configurabile l'indicazione (true/false) se la funzionalità proxy deve essere attivata anche verso gli endpoint registrati nelle token policy e nelle attribute authority, tramite le seguenti proprietà:

- govway-proxy-token-dynamic-discovery: servizio “dynamic-discovery” definito in una [Token Policy Validazione](#);
- govway-proxy-token-jwt-validation: location dei certificati da utilizzare per la “validazione jwt” di un token in una [Token Policy Validazione](#);
- govway-proxy-token-introspection: servizio “introspection” definito in una [Token Policy Validazione](#);
- govway-proxy-token-userinfo: servizio “user-info” definito in una [Token Policy Validazione](#);
- govway-proxy-token-retrieve: [Token Policy Negoziazione](#);
- govway-proxy-attribute-authority: [Attribute Authority](#) da cui vengono recuperati gli attributi;
- govway-proxy-attribute-authority-response-jwt-validation: location dei certificati da utilizzare per la “validazione jwt” della risposta in una [Attribute Authority](#).

Nota:

La configurazione dei parametri che riguardano l'header http o il parametro della url non sono obbligatori e se non presenti viene utilizzata la configurazione di default (header http "GovWay-APIAddress" non codificato in base64) ridefinibile nel file di configurazione locale "/etc/govway/govway_local.properties" tramite una configurazione come quella riportata di seguito (assumendo sia /etc/govway la directory di configurazione indicata in fase di installazione). Analogi discorsi vale per l'attivazione della funzionalità proxy verso gli endpoint registrati nelle token policy e nelle attribute authority, la quale è per default disabilitata.

```
# -----
# GovWay Proxy
#
# Default behaviour
org.openspcoop2.pdd.connatori.govwayProxy.enable=false
#
# Default configuration (HTTP)
org.openspcoop2.pdd.connatori.govwayProxy.header.enable=true
org.openspcoop2.pdd.connatori.govwayProxy.header.nome=GovWay-APIAddress
org.openspcoop2.pdd.connatori.govwayProxy.header.base64=false
#
# Default configuration (query URL)
org.openspcoop2.pdd.connatori.govwayProxy.urlParameter.enable=false
org.openspcoop2.pdd.connatori.govwayProxy.urlParameter.nome=govway_api_address
org.openspcoop2.pdd.connatori.govwayProxy.urlParameter.base64=false
#
# Default configuration (Token e Attributes)
org.openspcoop2.pdd.connatori.govwayProxy.tokenDynamicDiscovery.enable=false
org.openspcoop2.pdd.connatori.govwayProxy.tokenJwtValidation.enable=false
org.openspcoop2.pdd.connatori.govwayProxy.tokenIntrospection.enable=false
org.openspcoop2.pdd.connatori.govwayProxy.tokenUserInfo.enable=false
org.openspcoop2.pdd.connatori.govwayProxy.tokenRetrieve.enable=false
org.openspcoop2.pdd.connatori.govwayProxy.attributeAuthority.enable=false
org.openspcoop2.pdd.connatori.govwayProxy.attributeAuthority.responseJwtValidation.
  ↪enable=false
# -----
```

Nota: Anche l'abilitazione stessa della funzionalità "govway-proxy" può essere effettuata nel file di configurazione locale tramite la proprietà "org.openspcoop2.pdd.connatori.govwayProxy.enable" ed in questo caso non è necessario registrare la proprietà di sistema "govway-proxy-enable"

L'endpoint utilizzato per il proxy, indicato nella proprietà "govway-proxy", può essere ridefinito tramite le seguenti proprietà dalla più generica fino alla più specifica:

- govway-<ruolo>-proxy: l'endpoint indicato verrà utilizzato solamente se govway agisce nel ruolo indicato; "<ruolo>" può assumere i valori "fruizioni" o "erogazioni".
- profilo-<profilo>-govway-proxy o profilo-<profilo>-govway-<ruolo>-proxy: rispetto alla precedente proprietà è possibile restringere l'utilizzo dell'endpoint ad un determinato Profilo di Interoperabilità; "<profilo>" può assumere i valori "trasparente" (Profilo API Gateway), "modipa" (Profilo ModI), "spcoop" (Profilo SPCoop), "as4" (Profilo eDelivery), "sdi" (Profilo Fatturazione Elettronica).
- dominio-<nomeSoggetto>-govway-proxy o dominio-<nomeSoggetto>-govway-<ruolo>-proxy: l'endpoint indicato verrà utilizzato solamente per il soggetto interno indicato in "<nomeSoggetto>".
- dominio-<profilo>-<nomeSoggetto>-govway-proxy o dominio-<profilo>-<nomeSoggetto>-govway-<ruolo>-

proxy: rispetto alla precedente proprietà è possibile restringere l'utilizzo dell'endpoint per il soggetto interno indicato in “<nomeSoggetto>” relativamente al solo Profilo di Interoperabilità indicato in “<profilo>”.

- dominio-<tipoSoggetto>-<nomeSoggetto>-govway-proxy o dominio-<tipoSoggetto>-<nomeSoggetto>-govway-<ruolo>-proxy: rispetto alle precedenti due proprietà è possibile restringere l'utilizzo dell'endpoint per il soggetto interno indicato in “<nomeSoggetto>” relativamente al solo tipo indicato in “<tipoSoggetto>”. Questa opzione è utile nei profili di interoperabilità dove ai soggetti è possibile associare più tipi, come ad es. in SPCoop dove sono utilizzabili i tipi “spc”, “aoo”, “test”.
- tag-<nomeTag>-govway-proxy o tag-<nomeTag>-govway-<ruolo>-proxy: l'endpoint indicato verrà utilizzato solamente se l'API appartiene al tag indicato in “<nomeTag>”.

Anche i parametri di configurazione relativamente all'utilizzo dell'header, al parametro della url possono essere ridefiniti, quando viene ridefinito un endpoint, con lo stesso criterio. Analogi discorsi vale per l'attivazione della funzionalità proxy verso gli endpoint registrati nelle token policy e nelle attribute authority.

Nella figura Fig. 10.25 viene fornito un esempio di configurazione di un proxy relativamente alle sole fruizioni. L'endpoint del proxy è lo stesso per tutti i soggetti interni gestiti (dove è stato abilitato il multi-tenant) con la sola differenza che nel contesto della url è presente il nome del soggetto interno. In questo esempio l'endpoint remoto viene inserito nell'header HTTP GovWay-APIAddress codificato in base64.

	Nome	Valore
<input type="checkbox"/>	govway-fruizioni-proxy	https://proxy/\${busta:mittente}
<input type="checkbox"/>	govway-fruizioni-proxy-header	GovWay-APIAddress
<input type="checkbox"/>	govway-fruizioni-proxy-header-base64	true
<input type="checkbox"/>	govway-proxy-enable	true

Figure10.25: GovWay Proxy per le fruizioni con endpoint dinamico

Nella figura Fig. 10.26 viene fornito un esempio di configurazione di un proxy relativamente alle sole fruizioni dove l'endpoint del proxy differisce sulla porta a seconda del soggetto interno.

Configurazione Generale > Proprietà di Sistema

Proprietà di Sistema

Visualizzati record [1-4] su 4

	Nome	Valore
<input type="checkbox"/>	dominio-Ente1-govway-fruizioni-proxy	https://proxy:8652
<input type="checkbox"/>	dominio-Ente2-govway-fruizioni-proxy	https://proxy:8653
<input type="checkbox"/>	dominio-Ente3-govway-fruizioni-proxy	https://proxy:8654
<input type="checkbox"/>	govway-proxy-enable	true

ELIMINA **AGGIUNGI**

Figure10.26: GovWay Proxy per le fruizioni con endpoint differente per Soggetto Interno

10.13 Autenticazione e Autorizzazione Principal (Security Constraint)

In precedenza, relativamente alla configurazione del controllo degli accessi, ed in particolare del meccanismo di autenticazione, si è indicata anche la possibilità di utilizzare il tipo *principal*. Questa configurazione richiede che l'autenticazione sia delegata all'application server o qualunque altra modalità che permetta a GovWay di accedere al principal tramite la api *HttpServletRequest.getUserPrincipal()*.

In precedenza, relativamente all'autorizzazione, si è descritta la possibilità di utilizzare ruoli con fonte *esterna*. Questa fonte richiede che la gestione dei ruoli sia delegata all'Application Server o a qualunque altra modalità che permetta a GovWay di accedere ai ruoli tramite la api *HttpServletRequest.isUserInRole()*.

Le modalità di configurazione di utenti e ruoli sull'application server variano in funzione della versione utilizzata e pertanto si rimanda alla documentazione del prodotto.

È inoltre richiesto che l'applicazione utente sia protetta tramite un *security-constraint*.

Per abilitare la protezione è necessario intervenire all'interno dell'archivio *govway.ear*, editando il file definito nel war “*govway.war/WEB-INF/web.xml*”, dove è presente una configurazione di security constraint di default progettata per le url dei servizi esposti da GovWay nei vari profili di interoperabilità. Il metodo di autenticazione impostato per default è *HTTP-BASIC*. Per abilitare la protezione è sufficiente scommentare l'intera sezione che viene riportata di seguito.

```
<!-- start Security Constraint Authentication Container
<security-constraint>
    <web-resource-collection>
        <web-resource-name>AuthenticationContainer</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>*</role-name>
    </auth-constraint>
</security-constraint>

<security-constraint>
    <web-resource-collection>
        <web-resource-name>AuthenticationContainer</web-resource-name>
        <url-pattern>/IntegrationManager/PD</url-pattern>
        <url-pattern>/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/check</url-pattern>
        <url-pattern>/api/IntegrationManager/PD</url-pattern>
        <url-pattern>/api/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/api/check</url-pattern>
        <url-pattern>/modipa/IntegrationManager/PD</url-pattern>
        <url-pattern>/modipa/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/modipa/check</url-pattern>
        <url-pattern>/soap/IntegrationManager/PD</url-pattern>
        <url-pattern>/soap/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/soap/check</url-pattern>
        <url-pattern>/rest/IntegrationManager/PD</url-pattern>
        <url-pattern>/rest/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/rest/check</url-pattern>
        <url-pattern>/spcoop/IntegrationManager/PD</url-pattern>
        <url-pattern>/spcoop/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/spcoop/check</url-pattern>
        <url-pattern>/sdi/IntegrationManager/PD</url-pattern>
        <url-pattern>/sdi/IntegrationManager/MessageBox</url-pattern>
```

(continues on next page)

(continua dalla pagina precedente)

```

<url-pattern>/sdi/check</url-pattern>
<url-pattern>/as4/IntegrationManager/PD</url-pattern>
<url-pattern>/as4/IntegrationManager/MessageBox</url-pattern>
<url-pattern>/as4/check</url-pattern>
<http-method>GET</http-method>
</web-resource-collection>
</security-constraint>

<security-constraint>
    <web-resource-collection>
        <web-resource-name>AuthenticationContainer</web-resource-name>
        <url-pattern>/IntegrationManager/PD</url-pattern>
        <url-pattern>/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/check</url-pattern>
        <url-pattern>/api/IntegrationManager/PD</url-pattern>
        <url-pattern>/api/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/api/check</url-pattern>
        <url-pattern>/modipa/IntegrationManager/PD</url-pattern>
        <url-pattern>/modipa/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/modipa/check</url-pattern>
        <url-pattern>/soap/IntegrationManager/PD</url-pattern>
        <url-pattern>/soap/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/soap/check</url-pattern>
        <url-pattern>/rest/IntegrationManager/PD</url-pattern>
        <url-pattern>/rest/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/rest/check</url-pattern>
        <url-pattern>/spcoop/IntegrationManager/PD</url-pattern>
        <url-pattern>/spcoop/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/spcoop/check</url-pattern>
        <url-pattern>/sdi/IntegrationManager/PD</url-pattern>
        <url-pattern>/sdi/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/sdi/check</url-pattern>
        <url-pattern>/as4/IntegrationManager/PD</url-pattern>
        <url-pattern>/as4/IntegrationManager/MessageBox</url-pattern>
        <url-pattern>/as4/check</url-pattern>
        <http-method>TRACE</http-method>
        <http-method>HEAD</http-method>
        <http-method>DELETE</http-method>
        <http-method>POST</http-method>
        <http-method>CONNECT</http-method>
        <http-method>OPTIONS</http-method>
        <http-method>PUT</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>*</role-name>
    </auth-constraint>
</security-constraint>

<security-role>
    <role-name>*</role-name>
</security-role>

```

(continues on next page)

(continua dalla pagina precedente)

```
<login-config>
    <auth-method>BASIC</auth-method>
</login-config>
end Security Constraint Authentication Container -->
```

10.14 Aggiunta di Claims nei Token

In diverse funzionalità (*Token Policy Negoziazione - Signed JWT, Attribute Authority - Richiesta di Attributi, Payload Claims del token JWT*) è stata documentata la possibilità di aggiungere claim personalizzati nel payload JWT prodotto da GovWay indicandoli per riga nel formato “nome=valore”.

Tutti i valori definiti possono contenere parti dinamiche che verranno risolte a runtime dal Gateway (per maggiori dettagli *Valori dinamici*).

Le coppie di valori indicate consentono di aggiungere ulteriori claim nel payload JWT come tipo semplice stringa. Ad esempio definendo un valore come “claimTest=valoreClaim” verrà effettuata la seguente aggiunta al payload JWT:

```
{
    ...
    "claimTest": "valoreClaim"
}
```

Un valore definito tramite una parte dinamica, comporta un errore a runtime, se tale risoluzione non è possibile. Ad esempio definendo un valore come “claimTest=\${header:X-Example}”, se poi l’header http “X-Example” non esiste nella richiesta la transazione abortisce con errore. Per aggiungere il claim solamente se la risoluzione dinamica del valore viene effettuata con successo è possibile usare la forma opzionale “?{..}”. Ad esempio definendo un valore come “claimTest=?{header:X-Example}”, se poi l’header http “X-Example” non esiste nella richiesta, l’unico effetto è quello che non sarà aggiunto al JWT Payload il claim “claimTest”.

Fornendo un valore che inizia e termina con le parentesi graffe si definisce un oggetto json. Ad esempio definendo un valore come “claimTest={«prova»:>valoreProva, «prova2»:>\${header:X-Example}}” verrà effettuata la seguente aggiunta al payload JWT:

```
{
    ...
    "claimTest": {
        "prova": "valoreProva",
        "prova2": "<ValorePrelevatoHeaderHTTPIndicato>"
    }
}
```

Se il valore inizia e termina con le parentesi quadre si definisce invece un array json. Ad esempio definendo un valore come “claimTest=[«valoreProva», «valoreProva2», «\${header:X-Example}]” verrà effettuata la seguente aggiunta al payload JWT:

```
{
    ...
    "claimTest": ["valoreProva", "valoreProva2",
        ↵ "zValorePrelevatoHeaderHTTPIndicato"]
}
```

Per definire tipi primitivi json (boolean,int,long,float,double) è necessario attuare un cast nella forma “cast(<valore> as <tipoPrimitivo>)”. Ad esempio definendo dei valori come “claimTest=cast(true as boolean)”

e “claimTest2=cast(\${header:X-Example} as long)” verrà effettuata la seguente aggiunta al payload JWT (si suppone presente nella richiesta un header http “X-Example” valorizzato con “678”):

```
{  
  ...  
  "claimTest": true,  
  "claimTest2": 678  
}
```

Per convertire una lista json di tipi primitivi in lista di stringhe è possibile attuare un cast nella forma “cast(<valore> as string array)”. Ad esempio definendo dei valori come “claimTest=cast([1,2,3] as string array)” verrà effettuata la seguente aggiunta al payload JWT:

```
{  
  ...  
  "claimTest": ["1", "2", "3"]  
}
```

10.15 Accesso alle proprietà delle entità del Registro

In diverse funzionalità (*Trasformazioni - Valori dinamici, Autorizzazione Contenuti, Autorizzazione per Token Claims ...*) è stata documentata la possibilità di accedere alle proprietà registrate nelle entità presenti sul Registro; vengono di seguito riportate le keyword utilizzabili (il valore “NAME” indica la proprietà desiderata):

- *config:NAME* : proprietà configurata per l’API erogata o fruitore;
- *clientApplicationConfig:NAME* : proprietà configurata nell’applicativo fruitore;
- *clientOrganizationConfig:NAME* : proprietà configurata nel soggetto fruitore;
- *providerOrganizationConfig:NAME* : proprietà configurata nel soggetto erogatore;
- *tokenClientApplicationConfig:NAME* : proprietà configurata nell’applicativo client identificato tramite il clientId presente nel token;
- *tokenClientOrganizationConfig:NAME* : proprietà configurata nel soggetto proprietario dell’applicativo client identificato tramite il clientId presente nel token.

Oltre agli accessi diretti alle proprietà di una singola entità, GovWay fornisce un’ulteriore modalità di accesso definita tramite la keyword “dynamicConfig:FIELD” che consente di accedere alle proprietà delle entità coinvolte nella richiesta (api, applicativi, soggetti) verificando la presenza di una proprietà desiderata definita in funzione prima dell’entità più specifica e poi via via tramite quella più generica; il valore “FIELD” fornito deve rappresentare un field valido all’interno della classe “org.openscoop2.pdd.core.dynamic.DynamicConfig”.

Di seguito viene fornita una descrizione dei principali metodi forniti e l’ordine di ricerca della proprietà, indicata come parametro “pName”, dalla più specifica alla più generica; non appena viene individuata una proprietà viene utilizzato il suo valore e l’algoritmo di ricerca termina.

- *dynamicConfig:apiSearchByClientApplication(pName)*: effettua una ricerca tra le proprietà di un’erogazione o fruizione di API (config) attraverso i seguenti criteri:
 - <clientOrganizationName>.<clientApplicationName>.<pName>
 - <clientApplicationName>.<pName>
 - <clientOrganizationName>.<pName>
 - <pName>

- *dynamicConfig:clientApplicationSearch(pName)*: effettua una ricerca tra le proprietà di un'applicativo fruitore (clientApplicationConfig) attraverso i seguenti criteri:
 - <nomeErogatore>.<nomeApiImpl>.v<versioneApiImpl>.<pName>
 - <nomeApiImpl>.v<nomeApiImpl>.<pName>
 - <nomeErogatore>.<pName>
 - <pName>
- *dynamicConfig:clientOrganizationSearch(pName)*: effettua una ricerca tra le proprietà di un soggetto fruitore (clientOrganizationConfig) attraverso i seguenti criteri:
 - <nomeErogatore>.<nomeApiImpl>.v<versioneApiImpl>.<pName>
 - <nomeApiImpl>.v<nomeApiImpl>.<pName>
 - <nomeErogatore>.<pName>
 - <pName>
- *dynamicConfig:apiSearchByTokenClientApplication(pName)*: effettua una ricerca tra le proprietà di un'erogazione o fruizione di API (config) attraverso i seguenti criteri:
 - <tokenClientOrganizationName>.<tokenClientApplicationName>.<pName>
 - <tokenClientApplicationName>.<pName>
 - <tokenClientOrganizationName>.<pName>
 - <pName>
- *dynamicConfig:tokenClientApplicationSearch(pName)*: effettua una ricerca tra le proprietà di un'applicativo fruitore identificato tramite il clientId presente nel token (tokenClientApplicationConfig) attraverso i seguenti criteri:
 - <nomeErogatore>.<nomeApiImpl>.v<versioneApiImpl>.<pName>
 - <nomeApiImpl>.v<nomeApiImpl>.<pName>
 - <nomeErogatore>.<pName>
 - <pName>
- *dynamicConfig:tokenClientOrganizationSearch(pName)*: effettua una ricerca tra le proprietà di un soggetto proprietario dell'applicativo client identificato tramite il clientId presente nel token (tokenClientOrganizationConfig) attraverso i seguenti criteri:
 - <nomeErogatore>.<nomeApiImpl>.v<versioneApiImpl>.<pName>
 - <nomeApiImpl>.v<nomeApiImpl>.<pName>
 - <nomeErogatore>.<pName>
 - <pName>
- *dynamicConfig:providerSearch(pName)*: effettua una ricerca tra le proprietà di un soggetto erogatore (providerOrganizationConfig) attraverso i seguenti criteri:
 - <nomeApiImpl>.v<nomeApiImpl>.<pName>
 - <pName>

10.16 Espressioni XPath su messaggi JSON

In diverse funzionalità (*Correlazione Applicativa*, *Registrazione di una policy*, *Modalità di identificazione dell'azione*) è stata documentata la possibilità di utilizzare espressioni jsonPath o XPath per estrarre contenuti dai messaggi JSON o XML in transito sul Gateway.

L'estrazione dei contenuti da messaggi JSON si basa su espressioni JSONPath che allo stato attuale non hanno la stessa «potenza» delle espressioni XPath. Ad esempio:

- non è possibile ottenere il nome di un claim, come invece in XPath è possibile ottenere il local-name di un elemento tramite la funzione “local-name”
- non si dispongono delle complesse funzioni per le elaborazioni sulle stringhe (ad es. in xpath è disponibile la funzione “substring-before”)
- ...

Per ovviare a tali limitazioni GovWay fornisce la possibilità di utilizzare espressioni XPath su messaggi JSON attraverso la seguente sintassi:

```
xpath [namespace(prefix1:uri1, ... ,prefixN:uriN) ] <espressioneXPathStandard>
```

Nel caso il gateway rilevi una espressione che inizi con il prefisso “xpath” da applicare su un messaggio JSON, effettua una trasformazione del messaggio in una rappresentazione xml. Ad esempio per il messaggio JSON:

```
{
  "prova": "test1",
  "prova2": 23
}
```

Per estrarre il valore del field “prova” è possibile utilizzare le seguenti espressioni, la prima jsonPath e le successive xpath:

- \$.prova
- xpath //prova/text()
- xpath /json2xml/prova/text()

Le espressioni xpath sono utilizzabili poichè il messaggio JSON viene convertito nel seguente messaggio xml (inserito all'interno dell'elemento radice “json2xml”):

```
<json2xml>
  <prova>test1</prova>
  <prova2>23</prova2>
</json2xml>
```

Mentre nell'esempio precedente sono sufficienti le funzionalità offerte dal jsonPath per estrarre il valore del field “prova”, ricorrere all'utilizzo di XPath è necessario se ad esempio vogliamo ottenere il nome di un field. Nell'esempio seguente l'espressione fornita consente di estrarre il nome dell'ultimo field presente nella struttura json “prova2”. Tale risultato è ottenibile solamente utilizzando l'espressione xPath:

```
xpath local-name(/json2xml/*[last()])
```

In alcuni contesti i servizi REST non vengono implementati a partire da interfacce progettate ad hoc (OpenAPI, Swagger...) ma sono frutto di una trasformazioni di esistenti servizi SOAP. In questi scenari, i servizi REST veicolano messaggi JSON ottenuti attraverso la trasformazione dei relativi messaggi XML utilizzati su SOAP. Per poter utilizzare espressioni xPath devono essere affrontate le problematiche di risoluzione dei prefissi e dei namespace. In questi

contesti i messaggi JSON presenteranno field che possiedono nel nome il carattere ":" ereditato dalla rappresentazione xml. Di seguito un esempio di messaggio json ottenuto da una trasformazione di un messaggio xml equivalente:

```
{
  "m:NomeAzioneTestRequest": {
    "bodyWithNS" : "true",
    "xmlns:m" : "http://testNamespace",
    "prodotto" : {
      "codice" : "26",
      "altro:codice3" : "34",
      "xmlns:altro" : "http://testNamespaceAltro"
    }
  }
}
```

Supponendo di voler estrarre il nome del field "NomeAzioneTestRequest" e da questo eliminare anche il suffisso "Request" è possibile utilizzare la seguente espressione xPath:

```
xpath namespace(m:http://testNamespace, altro:http://altro) substring-before(local-name(/→/json2xml/*), \"Request\")
```

Si può notare come tra il prefisso "xpath" e l'espressione xpath vera e propria (substring-before(...)) siano stati definiti i namespace che coinvolgono i field presenti nella struttura json che avevano il carattere ":".

La struttura xml, ottenuta dalla conversione del messaggio json, su cui viene applicata l'espressione xpath è la seguente:

```
<json2xml xmlns:m="http://testNamespace" xmlns:altro="http://altro" xmlns:__xmlns=→"http://govway.org/utils/json2xml/xmlns">
  <m:NomeAzioneTestRequest>
    <bodyWithNS>true</bodyWithNS>
    <__xmlns:m>http://testNamespace</__xmlns:m>
    <prodotto>
      <codice>26</codice>
      <altro:codice3>34</altro:codice3>
      <__xmlns:altro>http://testNamespaceAltro</__xmlns:altro>
    </prodotto>
  </m:NomeAzioneTestRequest>
</json2xml>
```

Nota: Il prefisso "xmlns:" viene gestito automaticamente da GovWay, il quale gli associa un namespace di default "<http://govway.org/utils/json2xml/xmlns>". Tale namespace è possibile ridefinirlo aggiungendo all'elenco dei namespace anche un mapping per "xmlns".

10.17 Validazione dei messaggi con OpenAPI 3.x

Nella sezione *Validazione dei messaggi* è stata descritta la funzionalità di validazione dei messaggi applicativi in transito sul gateway.

Dalla versione 3.3.1, per la validazione dei messaggi riguardanti API REST con specifiche di interfaccia OpenAPI 3.x, viene utilizzata la libreria openapi4j (<https://openapi4j.github.io/openapi4j/>). È possibile ritornare al precedente motore di validazione registrando la seguente *Proprietà* sull'erogazione o sulla fruizione:

- *validation.openapi4j.enabled=false*

Dalla versione 3.3.5.p1 è inoltre possibile utilizzare un ulteriore motore di validazione, utilizzando la libreria “swagger-request-validator” (<https://bitbucket.org/atlassian/swagger-request-validator>). È possibile attivare il nuovo motore di validazione registrando la seguente *Proprietà* sull'erogazione o sulla fruizione:

- *validation.swaggerRequestValidator.enabled=true*

Se invece si vuole modificare il tipo di validazione effettuata con i motori “openapi4j” o “swagger-request-validator” è possibile farlo abilitando (true) o disabilitando (false) la specifica funzionalità registrando una delle seguenti *Proprietà* (per default tutte le proprietà elencate sono abilitate):

- *validation.openapi.validateAPISpec* (default: true): prima di procedere con la validazione del messaggio, viene controllato che l'interfaccia OpenAPI 3.x sia sintatticamente valida;
- *validation.openapi.validateRequestPath* (default: true): viene effettuata la validazione dei parametri che definiscono il path della risorsa;
- *validation.openapi.validateRequestQuery* (default: true): viene effettuata la validazione della query url;
- *validation.openapi.validateRequestHeaders* (default: true): viene effettuata la validazione degli header http della richiesta;
- *validation.openapi.validateResponseHeaders* (default: true): viene effettuata la validazione degli header http della risposta;
- *validation.openapi.validateRequestCookies* (default: true): viene effettuata la validazione dei cookie presenti nella richiesta;
- *validation.openapi.validateRequestBody* (default: true): viene effettuata la validazione del payload http della richiesta;
- *validation.openapi.validateResponseBody* (default: true): viene effettuata la validazione del payload http della risposta;
- *validation.openapi.mergeAPISpec* (default: true): eventuali schemi esterni json o yaml vengono aggiunti all'OpenAPI principale prima di procedere con la validazione.

Per il motore di validazione “openapi4j” sono disponibili le ultiori proprietà:

- *validation.openapi4j.validateMultipartOptimization* (default: false): attiva il processamento ottimizzato delle richieste multipart/form-data (o mixed). L'ottimizzazione sfrutta l'ipotesi che le parti «binarie», che non richiedono una validazione rispetto ad uno schema, sono tipicamente serializzate dopo i metadati (plain o json) e possono quindi essere «saltate» terminando l'analisi dello stream dopo aver validato i metadati in modo da avere benefici prestazionali visto che tipicamente le parti binarie rappresentano la maggior dimensione del messaggio in termini di bytes. Poichè se attivata l'ottimizzazione non consente di individuare se esistono part non definite nella specifica (in presenza di “additionalProperties=false”) il comportamento di default è quello di non usare l'ottimizzazione.

Per il motore di validazione “swagger-request-validator” sono disponibili le ultiori proprietà:

- `validation.swaggerRequestValidator.validateWildcardSubtypeAsJson` (default: true): consente di indicare se le richieste associate a media type definiti con il carattere “*” nel subtype (es. application/*) debbano essere validate come richieste json;
- `validation.swaggerRequestValidator.validateRequestUnexpectedQueryParam` (default: false): se abilitata vengono segnalati gli eventuali parametri non definiti nella specifica;
- `validation.swaggerRequestValidator.resolveFullyApiSpec` (default: false): indica se sostituire inline i \$ref nello schema con le loro definizioni. Per default viene utilizzato il valore “false” poichè quando vengono risolti inline non vengono gestiti correttamente i singoli attributi degli schemi combinati (oneOf, allOf ecc..). La risoluzione inline consente però di avere delle performance maggiori.
- `validation.swaggerRequestValidator.injectingAdditionalPropertiesFalse` (default: false): se abilitata, viene riattivato il transformer della libreria che aggiunge additionalProperties=false in tutti gli oggetti degli schemi. È necessario disattivarlo per poter validare correttamente gli schemi che definiscono tale proprietà a true. La libreria lo utilizza come workaround per validare strutture allOf.

10.18 Cifratura delle Password

Gli oggetti censiti nel registro di GovWay che possiedono una password sono i seguenti:

- le utenze delle console di gestione e monitoraggio (descritte nella sezione [Utenti](#));
- gli applicativi e i soggetti registrati con credenziali “http-basic” (sezione [Credenziali “http-basic”](#));
- gli applicativi e i soggetti registrati con credenziali “api-key”; in questo caso viene cifrata la chiave di identificazione univoca (sezione [Credenziali “api-key”](#)).

Le password vengono cifrate per default con un algoritmo di cifratura: SHA-512-based Unix crypt (\$6\$).

Nota: Per garantire la retrocompatibilità con le utenze esistenti precedenti alla versione 3.3.2 di GovWay, la verifica delle password viene attuata anche usando il precedente algoritmo. La verifica in modalità “backward compatibility” può essere disattivata, una volta migrate tutte le password al nuovo formato di cifratura, agendo sul file <directory-lavoro>/consolePassword.properties:

```
# Abilitare l'opzione seguente per poter autenticare:
# - le utenze delle console esistenti memorizzate con la precedente cifratura
#   ↳MD5
# - le password 'basic' degli applicativi/soggetti memorizzati in chiaro
passwordEncrypt.backwardCompatibility=false
```

È possibile modificare il tipo di cifratura configurando i parametri presenti nel file <directory-lavoro>/consolePassword.properties:

```
# Tipo di cifratura (enum org.openscoop2.utils.crypt.CryptType)
passwordEncrypt.type=SHA2_BASED_UNIX_CRYPT_SHA512
# In alternativa alla definizione di un tipo, è possibile fornire una classe
# che implementa l'interfaccia org.openscoop2.utils.crypt.ICrypt
#passwordEncrypt.customType=className

# Charset utilizzato per le password
#passwordEncrypt.charsetName=UTF-8

# Parametri per il calcolo del 'salt'
```

(continues on next page)

(continua dalla pagina precedente)

```

passwordEncrypt.salt.length=16
passwordEncrypt.salt.secureRandom=true
#passwordEncrypt.salt.secureRandomAlgorithm=SHA1PRNG

# Parametri per il calcolo del Digest
#passwordEncrypt.digestAlgorithm=
#passwordEncrypt.iteration=intNumber

# Output format
#passwordEncrypt.base64Encoding=true/false

```

I tipi di cifratura supportati sono:

- *SHA2_BASED_UNIX_CRYPT_SHA256* e *SHA2_BASED_UNIX_CRYPT_SHA512*: SHA2-based Unix crypt in variante SHA-256 e SHA-512; consentono la personalizzazione del “salt” e del numero di iterazioni (“rounds”).
- *LIBC_CRYPT_MD5* e *LIBC_CRYPT_MD5_APACHE*: libc crypt() MD5 «\$1\$» e variante Apache «\$apr1\$»; consentono la personalizzazione del “salt”.
- *DES_UNIX_CRYPT*: Unix crypt(3) DES; consente la personalizzazione del “salt”.
- *RFC2307_MD5*, *RFC2307_SMD5*, *RFC2307_SHA* e *RFC2307_SSHA*: RFC2307 in variante MD5, SMD5, SHA e SSHA; non consente alcuna personalizzazione.
- *JASYPT_BASIC_PASSWORD* e *JASYPT_STRONG_PASSWORD*: Jasypt in variante basic e strong; non consente alcuna personalizzazione.
- *JASYPT_CUSTOM_PASSWORD*: Jasypt custom configurabile per “salt”, numero di iterazioni, algoritmo di digest e codifica base64/hex.
- *PBE_KEY_SPEC*: PBE Key Spec configurabile per “salt”, numero di iterazioni, algoritmo di digest e codifica base64/hex.
- *B_CRYPT* e *S_CRYPT*: BCrypt e SCrypt; non consente alcuna personalizzazione.
- *PLAIN*: le password vengono salvate in chiaro

10.19 Visualizzazione delle Informazioni Confidenziali Cifrate

Le informazioni confidenziali descritte nella sezione *Credenziali richieste per l'acceso ad entità terze* e i valori delle proprietà descritte nella sezione *Proprietà censite nel registro di GovWay*, in presenza di byokInstall, verranno salvate cifrate sulla base dati e non sarà più possibile visualizzarle in chiaro tramite la console.

Per consentire all'utente di effettuare la decifratura su richiesta delle informazioni cifrate tramite la console, è possibile aggiungere la seguente proprietà agendo sul file *<directory-lavoro>/console_local.properties*:

```

# Consentire all'utente di visualizzare in chiaro le informazioni cifrate.
console.visualizzaInformazioniCifrate.enabled=true

```

Consentendo la decifratura su richiesta, nella maschera di visualizzazione di un'informazione confidenziale, accanto al lucchetto chiuso saranno visualizzate due ulteriori icone (Fig. 10.27) che consentono le seguenti azioni:

- “Copia”: il valore decifrato sarà disponibile per l'azione “incolla”;
- “Visualizza”: il valore decifrato verrà visualizzato all'utente (Fig. 10.28).

Figure10.27: Decifratura su richiesta delle informazioni confidenziali.

Figure10.28: Decifratura su richiesta delle informazioni confidenziali: visualizzazione informazione

10.20 Cifratura delle Informazioni Confidenziali

In architetture dove risiede una suddivisione fisica tra le istanze run e le istanze manager contenenti le console di gestione, la cifratura delle informazioni confidenziali può avvenire attraverso due modalità:

- la modalità standard, descritta nella sezione `byokInstallSecurityGovWay`, in cui viene attivato un unico *Security Engine* che prevede la condivisione della chiave master da parte di tutti i nodi run e manager;
- una differente modalità in cui la chiave master risiede solamente sui nodi run e i nodi manager richiedono l'operazione di wrap/unwrap ai nodi run.

Di seguito vengono fornite le indicazioni necessarie ad attivare la seconda modalità.

Nota: La seconda modalità può essere utilizzata per scenari di test o in ambienti progettati per far sì che le funzioni di wrap/unwrap esposte dai nodi run siano sufficientemente protette da criteri di sicurezza elevati, accessibili solamente dai nodi manager e dove il livello di sicurezza di accesso a questi ultimi è equivalente a quello dei corrispettivi nodi run.

La seconda modalità richiede la configurazione del file `<directory-lavoro>/govway.nodirun.properties` come descritto nella sezione cluster-console.

Devono inoltre essere abilitate le funzioni di wrap e unwrap (per default disabilitate) attraverso la creazione delle seguenti due proprietà sul file `<directory-lavoro>/govway_local.properties`:

```
org.openspcoop2.pdd.byok.jmx.wrap.enabled=true
org.openspcoop2.pdd.byok.jmx.unwrap.enabled=true
```

Tanto premesso la configurazione dei KSM utilizzati dai nodi manager potranno riferire speciali costanti:

- `#{govway-runtime:endpoint-wrap}`: consente di riferire l'endpoint esposto dai nodi run che implementa l'operazione di wrap;

- \${govway-runtime:endpoint-unwrap}: consente di riferire l'endpoint esposto dai nodi run che implementa l'operazione di unwrap;
- \${govway-runtime:username} e \${govway-runtime:password}: consente di riferire la credenziale http-basic configurata in un nodo run e descritta nella sezione cluster-console.

L'attivazione del *Security Engine* dovrà avvenire definendo due proprietà differenti nel file *<directory-lavoro>/byok.properties*:

- *govway.security*: identificativo del (byokInstallSecurityEngine) che devono utilizzare i nodi manager;
- *govway.security.runtime*: identificativo del (byokInstallSecurityEngine) utilizzato sui nodi run.

Di seguito un esempio in cui viene attivata la cifratura tramite il Security Engine “gw-pbkdf2” sui nodi run e l'invocazione remota da parte dei nodi manager:

```
#_
=====
govway.security=gw-remote
govway.security.runtime=gw-pbkdf2
#_
=====

# =====
# Security Engine
#
# 'gw-pbkdf2' usato dai nodi run
security.gw-pbkdf2.ksm.wrap=openssl-pbkdf2-wrap
security.gw-pbkdf2.ksm.unwrap=openssl-pbkdf2-unwrap
security.gw-pbkdf2.ksm.param.encryptionPassword=${envj:read(GOVWAY_ENCRYPTION_PASSWORD)}
#
# 'gw-remote' usato dai nodi manager
security.gw-remote.ksm.wrap=govway-remote-wrap
security.gw-remote.ksm.unwrap=govway-remote-unwrap
security.gw-remote.ksm.param.endpointWrap=${govway-runtime:endpoint-wrap}
security.gw-remote.ksm.param.endpointUnwrap=${govway-runtime:endpoint-unwrap}
security.gw-remote.ksm.param.username=${govway-runtime:username}
security.gw-remote.ksm.param.password=${govway-runtime:password}
# =====

# =====
# KSM 'gw-pbkdf2' usato dai nodi run
#
# WRAP
ksm.govway-openssl-pbkdf2-wrap.label=OpenSSL 'pbkdf2' Wrap
ksm.govway-openssl-pbkdf2-wrap.type=openssl-pbkdf2-wrap
ksm.govway-openssl-pbkdf2-wrap.mode=wrap
ksm.govway-openssl-pbkdf2-wrap.encryptionMode=local
ksm.govway-openssl-pbkdf2-wrap.input.param1.name=encryptionPassword
ksm.govway-openssl-pbkdf2-wrap.input.param1.label=Encryption Password
ksm.govway-openssl-pbkdf2-wrap.local.impl=openssl
ksm.govway-openssl-pbkdf2-wrap.local.keystore.type=pass
ksm.govway-openssl-pbkdf2-wrap.local.password=${ksm:encryptionPassword}
ksm.govway-openssl-pbkdf2-wrap.local.password.type=openssl-pbkdf2-aes-256-cbc
ksm.govway-openssl-pbkdf2-wrap.local.encoding=base64
#
```

(continues on next page)

(continua dalla pagina precedente)

```

# UNWRAP
ksm.govway-openssl-pbkdf2-unwrap.label=OpenSSL 'pbkdf2'
ksm.govway-openssl-pbkdf2-unwrap.type=openssl-pbkdf2-unwrap
ksm.govway-openssl-pbkdf2-unwrap.mode=unwrap
ksm.govway-openssl-pbkdf2-unwrap.encryptionMode=local
ksm.govway-openssl-pbkdf2-unwrap.input.param1.name=encryptionPassword
ksm.govway-openssl-pbkdf2-unwrap.input.param1.label=Encryption Password
ksm.govway-openssl-pbkdf2-unwrap.local.impl=openssl
ksm.govway-openssl-pbkdf2-unwrap.local.keystore.type=pass
ksm.govway-openssl-pbkdf2-unwrap.local.password=${ksm:encryptionPassword}
ksm.govway-openssl-pbkdf2-unwrap.local.password.type=openssl-pbkdf2-aes-256-cbc
ksm.govway-openssl-pbkdf2-unwrap.local.encoding=base64
# =====

# =====
# KSM 'gw-remote' usato dai nodi manager
#
# WRAP
ksm.govway-remote-wrap.label=GovWay Remote Wrap
ksm.govway-remote-wrap.type=govway-remote-wrap
ksm.govway-remote-wrap.mode=wrap
ksm.govway-remote-wrap.encryptionMode=remote
ksm.govway-remote-wrap.input.param1.name=endpointWrap
ksm.govway-remote-wrap.input.param1.label=Endpoint Wrap Key
ksm.govway-remote-wrap.input.param2.name=username
ksm.govway-remote-wrap.input.param2.label=Username
ksm.govway-remote-wrap.input.param3.name=password
ksm.govway-remote-wrap.input.param3.label=Password
ksm.govway-remote-wrap.http.endpoint=${ksm:endpointWrap}&paramValue=${ksm-base64-
˓→urlencoded-key}
ksm.govway-remote-wrap.http.method=GET
ksm.govway-remote-wrap.http.username=${ksm:username}
ksm.govway-remote-wrap.http.password=${ksm:password}
#
# UNWRAP
ksm.govway-remote-unwrap.label=GovWay Remote Unwrap
ksm.govway-remote-unwrap.type=govway-remote-unwrap
ksm.govway-remote-unwrap.mode=unwrap
ksm.govway-remote-unwrap.encryptionMode=remote
ksm.govway-remote-unwrap.input.param1.name=endpointUnwrap
ksm.govway-remote-unwrap.input.param1.label=Endpoint Unwrap Key
ksm.govway-remote-unwrap.input.param2.name=username
ksm.govway-remote-unwrap.input.param2.label=Username
ksm.govway-remote-unwrap.input.param3.name=password
ksm.govway-remote-unwrap.input.param3.label=Password
ksm.govway-remote-unwrap.http.endpoint=${ksm:endpointUnwrap}&paramValue=${ksm-urlencoded-
˓→key}
ksm.govway-remote-unwrap.http.method=GET
ksm.govway-remote-unwrap.http.username=${ksm:username}
ksm.govway-remote-unwrap.http.password=${ksm:password}
ksm.govway-remote-unwrap.http.response.base64Encoded=true
# =====

```

10.21 Health Check

Govway espone un servizio built-in di health check al contesto:

```
/govway/check
```

Il servizio, invocabile con una semplice GET, restituisce una risposta vuota con codice HTTP 200 nel caso in cui GovWay sia correttamente in funzione.

```
> curl -v http://localhost:8080/govway/check
> GET /govway/check HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.66.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Connection: keep-alive
< Content-Length: 0
< Date: Mon, 10 Oct 2022 14:55:45 GMT
```

Nel caso in cui GovWay non si sia avviato correttamente viene restituito un codice HTTP 503:

```
> curl -v http://localhost:8080/govway/check
> GET /govway/check HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.66.0
> Accept: */*
>
< HTTP/1.1 503 Service Unavailable
< Connection: keep-alive
< Content-Type: text/plain
< Content-Length: 35
< Date: Mon, 10 Oct 2022 16:13:59 GMT
<
API Gateway GovWay non inizializzato
```

Se invece vengono rilevati errori dopo che GovWay si è avviato correttamente viene restituito un codice HTTP 500 e nel payload viene riportata la motivazione dell'errore rilevato:

```
> curl -v http://localhost:8080/govway/check
> GET /govway/check HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.66.0
> Accept: */*
>
< HTTP/1.1 500 Internal Server Error
< Connection: keep-alive
< Content-Type: text/plain
< Content-Length: 203
< Date: Mon, 10 Oct 2022 16:17:40 GMT
<
Risorse di sistema non disponibili: [Database] Connessione al database GovWay
↳ non disponibile: javax.resource.ResourceException: IJ000453: Unable to get
↳ managed connection for java:/org.govway.datasource
```

Tra i controlli previsti è inclusa anche una verifica dell'aggiornamento frequente dei campionamenti statistici. La configurazione di default, riportata di seguito, prevede un livello di soglia (1) in modo da verificare che le statistiche siano sempre aggiornate almeno all'intervallo precedente: ad esempio, all'ora precedente per il campionamento orario o al giorno precedente per il giornaliero. Può essere indicato anche un valore “0” per imporre che le statistiche siano aggiornate (anche parzialmente) all'ultimo intervallo; questo controllo stringente potrebbe creare dei falsi positivi.

```
# =====
# Health check
...
# Report statistici
# Consente di verificare l'aggiornamento frequente dei campionamenti statistici.
# Se viene rilevata una data più vecchia rispetto a quanto indicato nella
# soglia, il servizio di health check ritorna un errore.
# Funzionamento della soglia:
# - se viene indicato un valore '1' si richiede che le statistiche siano
# aggiornate all'intervallo precedente: ad esempio, all'ora precedente per il
# campionamento orario o al giorno precedente per il giornaliero;
# - può essere indicato anche un valore '0' per imporre che le statistiche siano
# aggiornate (anche parzialmente) all'ultimo intervallo; questo controllo
# stringente potrebbe creare dei falsi positivi.
org.openscoop2.pdd.check.healthCheck.reportStatistici.intervalloOrario.
  verifica=true
org.openscoop2.pdd.check.healthCheck.reportStatistici.intervalloOrario.
  soglia=1
org.openscoop2.pdd.check.healthCheck.reportStatistici.intervalloGiornaliero.
  verifica=true
org.openscoop2.pdd.check.healthCheck.reportStatistici.intervalloGiornaliero.
  soglia=1
# =====
```

Di seguito un esempio di errore derivante da un campionamento statistico non correttamente aggiornato:

```
> curl -v http://localhost:8080/govway/check
> GET /govway/check HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.66.0
> Accept: */*
>
< HTTP/1.1 500 Internal Server Error
< Connection: keep-alive
< Content-Type: text/plain
< Content-Length: 203
< Date: Mon, 10 Oct 2022 16:17:40 GMT
<
Statistics HealthCheck failed
Hourly statistical information found whose last update is older than the
allowed threshold (1); last generation date: 2024-07-29_15:00:00.000
```

Oltre ai controlli standard, previsti per default sul servizio, è possibile abilitarne uno ulteriore che si occupa di invocare una API di HealthCheck configurata su GovWay. Per attivare questo controllo è necessario editare il file `<directory-lavoro>/govway_local.properties` e abilitare l'health check per API REST e/o per API SOAP, indicando il corretto endpoint di esposizione dell'API:

```
# -----
# Health check
# Se abilitato, il servizio /govway/check invocherà anche l'API REST e/o SOAP
# per verificare il corretto funzionamento di GovWay
# API REST
org.openspcoop2.pdd.check.healthCheck.apiRest.enabled=true
org.openspcoop2.pdd.check.healthCheck.apiRest.endpoint=http://localhost:8080/
  ↪govway/ENTE/api-rest-status/v1/status
# API SOAP
org.openspcoop2.pdd.check.healthCheck.apiSoap.enabled=true
org.openspcoop2.pdd.check.healthCheck.apiSoap.endpoint=http://localhost:8080/
  ↪govway/ENTE/api-soap-status/v1
...
# -----
```