

Dall'esperienza della Porta di Dominio italiana, l'**API Gateway** conforme alle normative della **Pubblica Amministrazione**

Quick Start Guide

Copyright © 2005-2018 *Link.it srl*

Indice

1 Profilo API Gateway	1
1.1 Erogazione API REST	3
1.2 Modalità Multi-Tenant	8
1.3 Erogazione API SOAP	11
1.4 Fruizione API	15
2 Configurazioni differenti per gruppi di risorse di una API	19
3 Sospensione di una API	26
4 Gestione CORS	30
5 Controllo degli Accessi	34
5.1 OAuth	34
5.1.1 Validazione tramite Introspection	35
5.1.2 Validazione JWT	43
5.1.3 Autenticazione e OIDC UserInfo	47
5.1.4 Autorizzazione per Scope	53
5.1.5 Autorizzazione sui Claims	60
5.1.6 Autorizzazione XACML	65
5.1.7 Token Forward	71
5.1.8 Registrazione Authorization Server	75
5.2 Autenticazione	77
5.2.1 Autenticazione Https	78
5.2.1.1 Identificazione dei Mittenti	83
5.2.2 Autenticazione Http Basic	88
5.2.3 Autenticazione Container	88
5.3 Autorizzazione	88
5.3.1 Autorizzazione Puntuale	89
5.3.2 Autorizzazione per Ruoli	89
5.3.3 XACML	89
6 Rate Limiting	89
6.1 Numero massimo di Richieste	89
6.2 Numero massimo di Richieste Concorrenti	89
6.3 Massima Banda Occupabile	89
6.4 Tempo Medio di Risposta	89
6.5 Numero massimo di Fault Applicativi	89

7 Validazione Messaggi	89
7.1 Validazione API REST	89
7.2 Validazione API SOAP	89
8 Caching Risposte	90
9 Sicurezza Messaggi	90
9.1 WSSecurity Signature	90
9.2 WSSecurity Encrypt	90
9.3 WSSecurity SAML	90
9.4 JWT Signature	90
9.5 JWT Encrypt	90
10 Registrazione Messaggi	90
11 Tracciamento	90
11.1 Correlazione Applicativa su API REST	90
11.2 Correlazione Applicativa su API SOAP	90
11.3 Disattivazione	91
11.4 Livello di Log	91
12 MTOM	91
12.1 Packaging	91
12.2 Unpackaging	91
12.3 Validazione	91
12.4 Verifica	91
13 Profilo FatturaPA	91
13.1 Fatturazione Attiva	91
13.2 Fatturazione Passiva	91
14 Profilo SPCoop	91
14.1 Profilo Oneway	92
14.2 Profilo Sincrono	92
14.3 Profilo Asincrono Simmetrico	92
14.4 Profilo Asincrono Asimmetrico	92
15 Analisi Statistica	92
15.1 Distribuzione Temporale	92
15.2 Distribuzione per Esiti	92

Elenco delle figure

1	Selezione del profilo <i>API Gateway</i>	1
2	Erogazione di una API Rest tramite GovWay	3
3	Registrazione di una API	4
4	Risorse di una API	4
5	Registrazione di una erogazione di API	5
6	URL di Invocazione dell'API erogata	6
7	Tracce delle invocazioni transitate sul Gateway	7
8	Dettaglio di una invocazione transitata sul Gateway	8
9	Scenario Multi-Tenant	9
10	Configurazione Multi-Tenant Abilitato	9
11	Registrazione nuovo Soggetto	10
12	Selezione del Soggetto	10
13	URL di Invocazione dell'API erogata	11
14	Erogazione di una API SOAP tramite GovWay	12
15	Registrazione di una API	12
16	Servizi di una API	13
17	Registrazione di una erogazione di API	14
18	URL di Invocazione dell'API erogata	14
19	Fruizione di una API tramite GovWay	16
20	Registrazione nuovo Soggetto	16
21	Registrazione di una fruizione di API	17
22	URL di Invocazione dell'API fruita	18
23	Configurazioni differenti per gruppi di risorse di una API	19
24	Situazione iniziale con unico gruppo 'Predefinito'	20
25	Registrazione Gruppo 'Creazione e Modifica'	20
26	Registrazione Gruppo 'Eliminazione'	21
27	Gruppi Registrati	21
28	Configurazioni dei Gruppi	22
29	Reset Cache delle Configurazioni di GovWay	22
30	Tracce delle invocazioni transitate sul Gateway	23
31	Dettaglio di una invocazione fallita bloccata dal Gateway	24
32	Registrazione Gruppo 'Eliminazione'	24
33	Tracce delle invocazioni transitate sul Gateway	26
34	Sospensione di una API	27
35	Sospensione di una erogazione	27
36	Erogazione sospesa	28
37	Stato disabilitato riportato nell'elenco delle erogazioni	28

38	Tracce delle invocazioni transitate sul Gateway	29
39	Gruppo di una erogazione sospeso	29
40	Stato disabilitato di un gruppo riportato nell'elenco delle erogazioni	29
41	Scenario cross-origin HTTP request (CORS)	30
42	CORS - Configurazione di default	31
43	Verifica CORS	32
44	Verifica CORS: richiesta OPTIONS	32
45	Personalizzazione Gestione CORS di una erogazione	33
46	Personalizzazione Gestione CORS: definizione di uno specifico 'origin'	33
47	Verifica CORS: definizione di uno specifico 'origin'	34
48	Scenario OAuth	35
49	Configurazione OAuth2 per PetStore	36
50	Tracce delle invocazioni terminate con errore 'Gestione Token Fallita'	37
51	Ottenimento Token: Playground Google, Step 1	38
52	Ottenimento Token: Playground Google, Step 2	38
53	Ottenimento Token: Playground Google, Step 3	39
54	Traccia di una invocazione terminata con successo	40
55	Informazioni ottenute tramite Introspection del Token	40
56	Configurazione Registrazione Messaggi per visualizzare Header HTTP	42
57	Dettaglio della transazione con contenuti	42
58	Header HTTP prodotti da GovWay contenenti le informazioni sul Token	43
59	Scenario OAuth con validazione JWT	44
60	Configurazione OAuth2 - Validazione JWT	45
61	Ottenimento Token: Playground Google, Step 3	45
62	Traccia di una invocazione terminata con successo	46
63	Informazioni presenti in un Token JWT	47
64	Scenario OAuth con accesso servizio UserInfo	48
65	Configurazione OAuth2 - Autenticazione	49
66	Tracce delle invocazioni terminate con errore 'Autenticazione Fallita'	50
67	Diagnostici di una invocazione terminata con errore	50
68	Informazioni presenti nel Token	51
69	Configurazione OAuth2 - Autenticazione	51
70	Traccia di una invocazione terminata con successo	52
71	Informazioni presenti in un Token JWT	52
72	Scenario OAuth con autorizzazione per Scope	53
73	Ottenimento Token: Playground Google, scelta scope API Calendar	54
74	Ottenimento Token: Playground Google, autorizzazione scope API Calendar	55
75	Ottenimento Token: Playground Google, Step 3	55
76	Configurazione OAuth2 - Registrazione Scope	56

77	Configurazione OAuth2 - Lista degli Scope registrati	56
78	Configurazione OAuth2 - Autorizzazione	57
79	Configurazione OAuth2 - Autorizzazione - Scope	58
80	Configurazione OAuth2 - Autorizzazione - Elenco Scope	58
81	Tracce delle invocazioni terminate con errore 'Autorizzazione Negata'	59
82	Diagnostici di una invocazione terminata con errore	59
83	Scope presenti nel Token	60
84	Configurazione OAuth2 - Autorizzazione degli scope con opzione 'Almeno uno'	60
85	Scenario OAuth con autorizzazione sui Claims	61
86	Configurazione OAuth2 - Autorizzazione	62
87	Tracce delle invocazioni terminate con errore 'Autorizzazione Negata'	63
88	Diagnostici di una invocazione terminata con errore	63
89	Configurazione OAuth2 - Autorizzazione dei claims corretta	64
90	Scenario OAuth con autorizzazione XACMLPolicy	65
91	Configurazione OAuth2 - Autorizzazione XACML Policy	68
92	Tracce delle invocazioni terminate con errore 'Autorizzazione Negata'	69
93	Diagnostici di una invocazione terminata con errore	69
94	Configurazione Registrazione Messaggi per visualizzare Header HTTP	71
95	Dettaglio della transazione con contenuti	72
96	Header HTTP prodotti da GovWay contenenti le informazioni sul Token	72
97	Modalità di Forward delle Informazioni Raccolte	73
98	Modalità di Forward del Token Originale	73
99	Token Policy di esempio: Google (1/2)	76
100	Token Policy di esempio: Google (2/2)	77
101	Scenario con autenticazione Https	78
102	Configurazione Autenticazione Https	78
103	Traccia dell'invocazione contenente il subject del certificato client	79
104	Ricerca di transazioni con mittente identificato fornendo l'intero subject del certificato client	80
105	Ricerca di transazioni con mittente identificato fornendo una parte del subject del certificato client	81
106	Tracce delle invocazioni terminate con errore 'Autenticazione Fallita'	82
107	Diagnostici di una invocazione terminata con errore	82
108	Scenario con autenticazione Https e identificazione dei mittenti	83
109	Registrazione nuovo Soggetto	84
110	Registrazione nuovo Soggetto	85
111	Traccia dell'invocazione contenente il soggetto mittente	86
112	Ricerca di transazioni di un soggetto mittente	87
113	Traccia dell'invocazione contenente l'applicativo mittente	88
114	Ricerca di transazioni di un applicativo mittente	88

Elenco delle tabelle

1	Registrazione scope	56
---	-------------------------------	----

1 Profilo API Gateway

Gli scenari che descriviamo in questa sezione mostrano come configurare l'API Gateway per gestire qualunque generica API basata su scambio di messaggi SOAP e REST.

Profilo di Utilizzo delle Console

Prima di procedere con gli scenari descritti nei successivi paragrafi selezionare il profilo *API Gateway* nell'apposito menù situato in alto a destra presente nell'intestazione delle console.

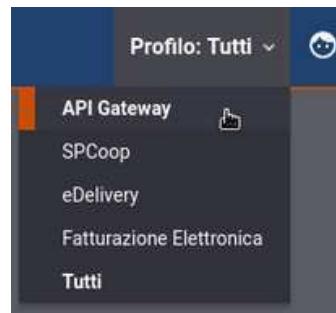


Figura 1: Selezione del profilo *API Gateway*

Per quanto concerne la tipologia di servizi **REST**, il servizio di esempio utilizzato per mostrare le funzionalità dell'API Gateway è lo *Swagger Petstore* (web site: <https://petstore.swagger.io/>) disponibile all'indirizzo <http://petstore.swagger.io/v2/>. L'interfaccia API è scaricabile in <https://petstore.swagger.io/v2/swagger.json>. Per simulare un aggiornamento di un animale all'interno del negozio è utilizzabile il seguente comando:

```
curl -v -X PUT "http://petstore.swagger.io/v2/pet" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento viene confermato con un codice http 200 e una risposta json equivalente alla richiesta:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Content-Type: application/json
Connection: close
Server: Jetty(9.2.9.v20150224)

{
    "id":3,
    "category": {"id":22,"name":"dog"},
    "name": "doggie",
    "photoUrls": ["http://image/dog.jpg"],
    "tags": [{"id":23,"name":"white"}],
    "status": "available"
}
```

Per i servizi di tipologia **SOAP**, il servizio di esempio utilizzato per mostrare le funzionalità dell'API Gateway è *Credit Card Verification* (web site: http://wiki.cdyne.com/index.php/Credit_Card_Verification) disponibile all'indirizzo <http://ws.cdyne.com/creditcardverify/luhnchecker.asmx>. L'interfaccia WSDL del servizio è scaricabile in <https://ws.cdyne.com/creditcardverify-luhnchecker.asmx?wsdl>. Per simulare una richiesta il cui fine è validare un numero di carta di credito è utilizzabile il seguente comando, che genera una richiesta SOAP 1.1:

```
curl -v -X POST "http://ws.cdyne.com/creditcardverify/luhnchecker.asmx" \
-H 'Content-Type: text/xml; charset=UTF-8' \
-H 'SOAPAction: "http://ws.cdyne.com/CheckCC"' \
-d '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<CheckCC xmlns="http://ws.cdyne.com/">
<CardNumber>4111111111111111</CardNumber>
</CheckCC>
</soapenv:Body>
</soapenv:Envelope>'
```

L'esito della verifica viene ritornato con un codice http 200 e una risposta contenente i dettagli della carta:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: text/xml; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Thu, 15 Nov 2018 11:50:12 GMT
Content-Length: 393
Connection: keep-alive
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<CheckCCResponse xmlns="http://ws.cdyne.com/">
<CheckCCResult>
<CardType>VISA</CardType>
<CardValid>true</CardValid>
</CheckCCResult>
</CheckCCResponse>
</soap:Body>
</soap:Envelope>
```

Per simulare la medesima richiesta utilizzando SOAP 1.2 è possibile usare il comando:

```
curl -v -X POST "http://ws.cdyne.com/creditcardverify/luhnchecker.asmx" \
-H 'Content-Type: application/soap+xml; charset=utf-8' \
-d '<soap12:Envelope xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
<soap12:Header/>
<soap12:Body>
<CheckCC xmlns="http://ws.cdyne.com/">
<CardNumber>4111111111111111</CardNumber>
</CheckCC>
</soap12:Body>
</soap12:Envelope>'
```

L'esito della verifica viene ritornato con un codice http 200 e una risposta contenente i dettagli della carta:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
```

```
Content-Type: application/soap+xml; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Thu, 15 Nov 2018 11:50:12 GMT
Content-Length: 393
Connection: keep-alive
```

```
<soap12:Envelope xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CheckCCResponse xmlns="http://ws.cdyne.com/">
      <CheckCCResult>
        <CardType>VISA</CardType>
        <CardValid>true</CardValid>
      </CheckCCResult>
    </CheckCCResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1 Erogazione API REST

Procediamo adesso con la descrizione dei passi di configurazione necessari a registrare una API REST implementata da un applicativo interno al proprio dominio di gestione. L'applicativo implementa lo *Swagger Petstore* descritto in Sezione 1. In questo scenario di esempio si suppone che l'indirizzo <http://petstore.swagger.io/> dove viene erogato il servizio sia interno al proprio dominio di gestione.

L'API, per questo primo esempio di utilizzo del Gateway, viene registrata in modo che sia accessibile in forma anonima da qualunque client invocando l'url esposta da GovWay. Una rappresentazione di questo scenario è mostrata nella figura Figura 2. Prima di procedere con la configurazione effettuare il download dell'interfaccia API disponibile in <https://petstore.swagger.io/v2/swagger.json>.

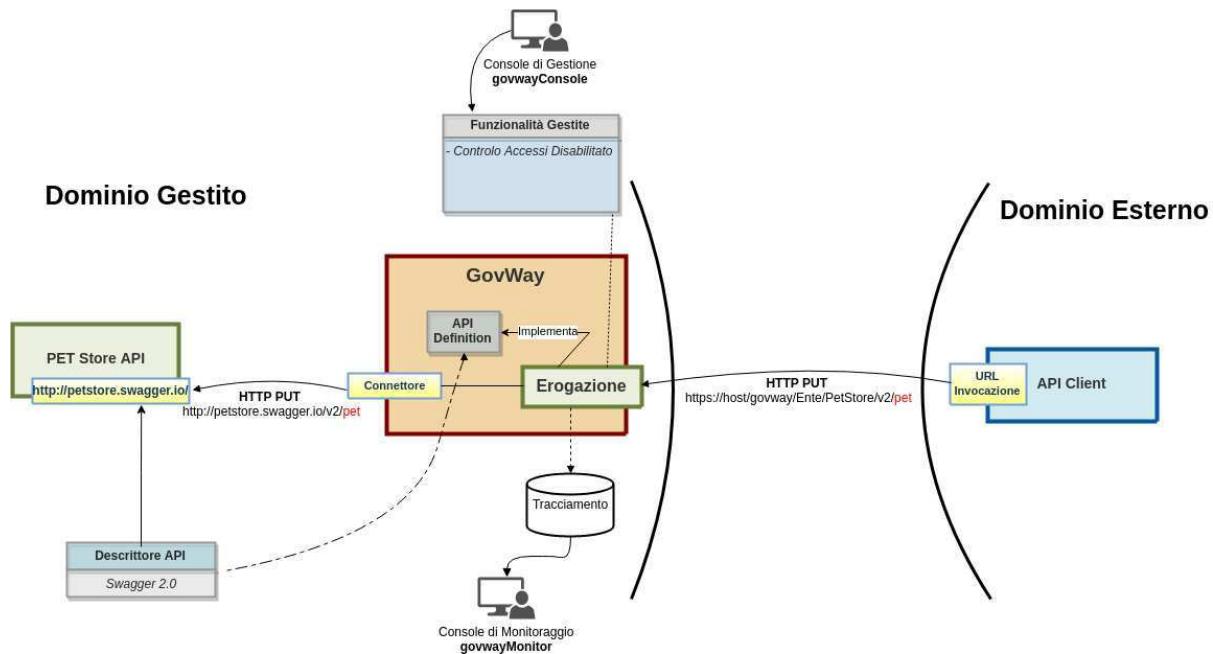


Figura 2: Erogazione di una API Rest tramite GovWay

Per registrare l'API su Govway, utilizzando la console *govwayConsole*, procedere come segue:

1. Registrazione API.

Accedere alla sezione '*API*' e selezionare il pulsante '*Aggiungi*'. Fornire i seguenti dati:

- *Tipo*: selezionare la tipologia '*REST*'.
- *Nome*: indicare il nome dell'API che si sta registrando, ad esempio '*PetStore*'.
- *Descrizione*: optionalmente è possibile fornire una descrizione generica dell'API.
- *Versione*: indicare la versione dell'API che si sta registrando; nell'esempio utilizziamo la versione 2 del PetStore.
- *Formato Specifica*: selezionare '*Swagger 2.0*' tra i formati supportati.
- *Swagger 2.0*: caricare l'interfaccia API scaricata dall'indirizzo <https://petstore.swagger.io/v2/swagger.json>.

The screenshot shows the 'Aggiungi' (Add) API registration form. The 'API' section contains fields for Tipo (Rest), Nome (PetStore), Descrizione (Servizio di esempio per API REST), and Versione (2). The 'Specifiche delle interfacce' section contains a dropdown for Formato Specifica (Swagger 2.0) and a file input field showing 'Choose File' and 'swagger.json'. A 'SALVA' (Save) button is at the bottom.

Figura 3: Registrazione di una API

Effettuato il salvataggio, l'API sarà consultabile all'interno dell'elenco delle API registrate. Accedendo al dettaglio si potranno visionare le risorse che tale API dispone come si può vedere dalla figura Figura 4.

	Method	Path	Descrizione
	POST	/pet	
	PUT	/pet	
	GET	/pet/findByStatus	
	GET	/pet/findByTags	
	DELETE	/pet/{petId}	
	GET	/pet/{petId}	
	POST	/pet/{petId}	

Figura 4: Risorse di una API

2. Registrazione Erogazione

Accedere alla sezione '*Erogazioni*' e selezionare il pulsante '*Aggiungi*'. Fornire i seguenti dati:

- *Nome*: selezionare l'API precedentemente registrata '*PetStore v2*'.
- *Autenticazione - Stato*: per esporre l'API in modo che sia invocabile da qualunque client in forma anonima selezionare lo stato '*disabilitato*'.
- *Connettore - Endpoint*: indicare la *base uri* dove viene erogata l'API nel dominio interno. Per il nostro esempio utilizzare la url:
 - *http://petstore.swagger.io/v2*

The screenshot shows the 'Aggiungi' (Add) form for registering an API. The form has three main sections: 'Informazioni Generali', 'Autenticazione', and 'Connettore'. In the 'Informazioni Generali' section, 'Soggetto Erogatore' is set to 'Amministrazione' and 'Nome' is set to 'PetStore v2'. In the 'Autenticazione' section, 'Trasporto' is set to 'disabilitato'. In the 'Connettore' section, 'Endpoint' is set to 'http://petstore.swagger.io/v2'. A 'SALVA' (Save) button is at the bottom.

Figura 5: Registrazione di una erogazione di API

Effettuato il salvataggio, l'API erogata sarà consultabile all'interno dell'elenco delle erogazioni. Accedendo al dettaglio si potrà conoscere l'*url di invocazione* che deve essere comunicata ai client che desiderano invocare l'API.

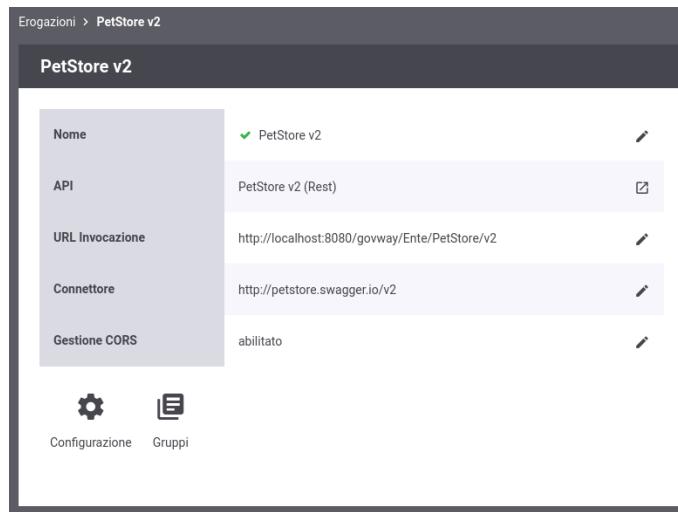


Figura 6: URL di Invocazione dell'API erogata

3. Invocazione API tramite GovWay

Al termine di questi passi di configurazione il servizio REST sarà raggiungibile dai client utilizzando l'url di invocazione:

- `http://host:port/govway/<soggetto-dominio-interno>/PetStore/v2/<uri-risorsa>`

Soggetto Interno al Dominio

In questo esempio si suppone che il nome del soggetto fornito durante la fase di installazione di GovWay sia *Ente*.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento viene confermato con un codice http 200 e una risposta json equivalente alla richiesta:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Content-Type: application/json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Message-ID: 84e1d9a4-c181-436f-b7f0-4cabf55c370d
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824
```

```
{
    "id":3,
    "category":{ "id":22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
```

```
    "status":"available"
}
```

Traccia della comunicazione

L'invocazione restituisce al client, sotto forma di header HTTP, l'id di transazione con cui è stata salvata la traccia contenente tutti i dati dell'invocazione sul Gateway.

4. Consultazione Tracce

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway (figura Figura 7) e conoscere il dettaglio di una singola invocazione (figura Figura 8).

		Data Ingresso Richiesta ▾	Tipologia	Esito	Fruitore	Erogatore	API	Azione
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-14 14:00:59	Erogazione	Ok		Ente	PetStore v2	PUT_pet
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-14 14:00:58	Erogazione	Ok		Ente	PetStore v2	PUT_pet
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-14 14:00:57	Erogazione	Ok		Ente	PetStore v2	PUT_pet
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-14 14:00:55	Erogazione	Ok		Ente	PetStore v2	PUT_pet
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-14 13:30:27	Erogazione	Ok		Ente	PetStore v2	PUT_pet

Figura 7: Tracce delle invocazioni transitate sul Gateway

The screenshot displays the 'Dettaglio Transazione' (Transaction Detail) page from the GovWay interface. The page is organized into several sections:

- Informazioni Generali (General Information):**
 - Tipologia: Erogazione (API Gateway)
 - Erogatore: Ente
 - API: PetStore v2
 - Azione: PUT_pet
 - Profilo Collaborazione: Sincrono
 - Esito: Ok
 - Diagnostici: Visualizza | Esporta
- Dettagli Richiesta (Request Details):**
 - ID Messaggio: 5d55e710-c795-4d78-ad2c-6da3f4c32101
 - Data Ingresso: 2018-11-14 14:00:59.536
 - Data Uscita: 2018-11-14 14:00:59.540
 - Bytes Ingresso: 225 B
 - Bytes Uscita: 225 B
- Dettagli Risposta (Response Details):**
 - Data Ingresso: 2018-11-14 14:00:59.765
 - Data Uscita: 2018-11-14 14:00:59.768
 - Bytes Ingresso: 150 B
 - Bytes Uscita: 150 B
- Informazioni Mittente (Sender Information):**
 - Metodo HTTP: PUT
 - URL Invocazione: [in] /govway/in/Ente/PetStore/v2/pet
 - Indirizzo Client: 127.0.0.1
 - Codice Risposta Client: 200
- Informazioni Avanzate (Advanced Information):**
 - ID Transazione: ab361e6b-f41f-4a53-a194-60cb19f6b30f
 - Dominio (ID): domain/gw/Ente
 - Dominio (Soggetto): Ente
 - Connettore: http://petstore.swagger.io/v2/pet
 - Codice Risposta: 200
 - Latenza Totale: 232 ms
 - Latenza Servizio: 225 ms
 - Latenza Gateway: 7 ms
 - Porta Inbound: Ente/PetStore/v2
 - Applicativo Erogatore: gw_Ente/gw_PetStore/v2

Figura 8: Dettaglio di una invocazione transitata sul Gateway

1.2 Modalità Multi-Tenant

GovWay supporta nativamente il multi-tenant grazie al quale è possibile gestire più domini. Una API che deve essere erogata su più domini viene registrata solamente una volta e può poi essere implementata da tutti i soggetti dei vari domini gestiti. Un applicativo client, per indirizzare una specifica API di un dominio, deve semplicemente indicare il nome del soggetto nella url di invocazione. Una rappresentazione di uno scenario multi-tenant è mostrata nella figura Figura 9

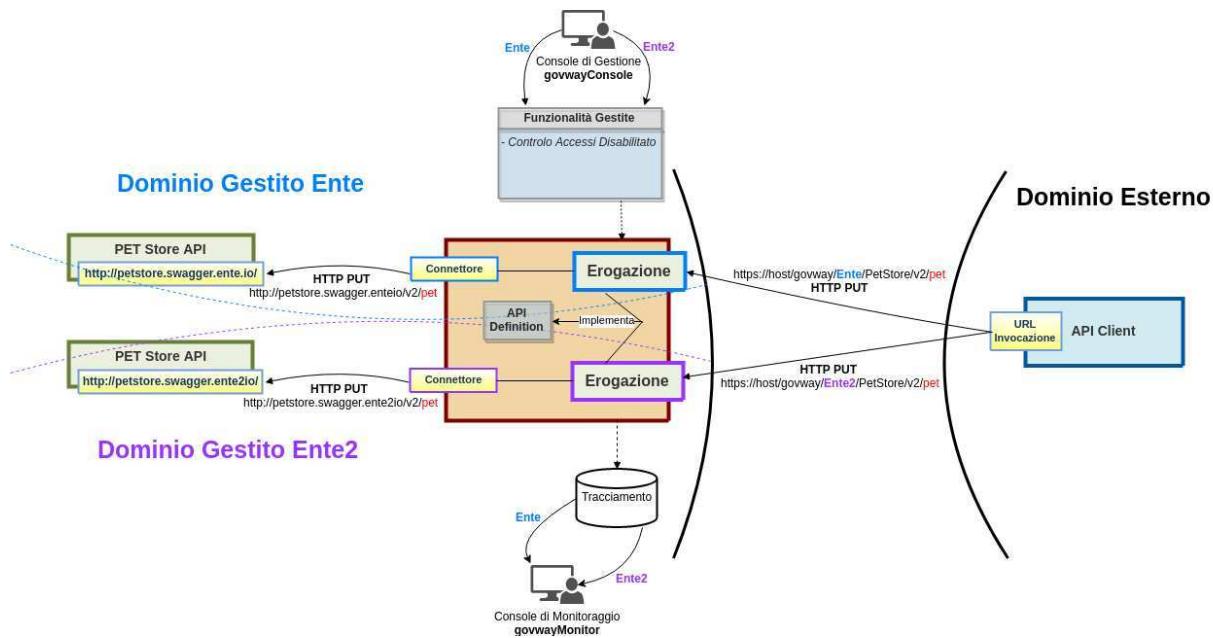


Figura 9: Scenario Multi-Tenant

Di seguito vengono descritti i passi necessari a gestire più domini (multi-tenant) su GovWay al fine di erogare l'API già registrata nell'esempio descritto nella sezione Sezione 1.1 all'interno di un ulteriore dominio gestito dal soggetto *Ente2*.

1. Abilitazione Multi-Tenant

GovWay viene installato per default con la funzionalità multi-tenant disabilitata e quindi l'unico dominio gestito è quello del soggetto fornito in fase di installazione. Per abilitare il multi-tenant accedere alla sezione '*Configurazione*' e selezionare la voce '*Generale*'. Nella maschera visualizzata selezionare il valore '*abilitato*' nella sezione '*Multi-Tenant*'.



Figura 10: Configurazione Multi-Tenant Abilitato

2. Registrazione nuovo Soggetto

Accedere alla sezione '*Soggetti*' e selezionare il pulsante '*Aggiungi*'. Fornire i seguenti dati:

- Dominio*: selezionare la voce '*Interno*'.
- Nome*: indicare il nome del Soggetto che rappresenta il nuovo dominio in gestione, ad esempio '*Ente2*'.
- Descrizione*: opzionalmente è possibile fornire una descrizione generica del soggetto.

Soggetti > Aggiungi

Note: (*) Campi obbligatori

Soggetto

Dominio: Interno

Nome *: Ente2

Descrizione:

SALVA

Figura 11: Registrazione nuovo Soggetto

3. Selezione del Dominio da gestire

Sia nella console di gestione (*govwayConsole*) che nella console di monitoraggio (*govwayMonitor*), una volta abilitato il Multi-Tenant, prima di procedere con qualsiasi operazione deve essere selezionato il soggetto per cui si intende gestire il dominio attraverso l'apposito menù situato in alto a destra nell'intestazione delle console.



Figura 12: Selezione del Soggetto

4. Registrazione Erogazione

Procedere con la registrazione della API '*PetStore v2*' così come già descritto nella sezione Sezione 1.1. Accedere alla sezione '*Erogazioni*' e selezionare il pulsante '*Aggiungi*'. Fornire i seguenti dati:

- *Nome*: selezionare l'API precedentemente registrata '*PetStore v2*'.
- *Autenticazione - Stato*: per esporre l'API in modo che sia invocabile da qualunque client in forma anonima selezionare lo stato '*disabilitato*'.
- *Connettore - Endpoint*: indicare la *base uri* dove viene erogata l'API nel dominio interno. Per il nostro esempio utilizzare sempre la url:
 - <http://petstore.swagger.io/v2>

Effettuato il salvataggio, l'API erogata sarà consultabile all'interno dell'elenco delle erogazioni. Accedendo al dettaglio si potrà conoscere l'*url di invocazione* che deve essere comunicata ai client che desiderano invocare l'API.

Nome del Soggetto presente nella url di invocazione

Come si può vedere dalla figura Figura 13 il soggetto *Ente2* compare nella url indicata.

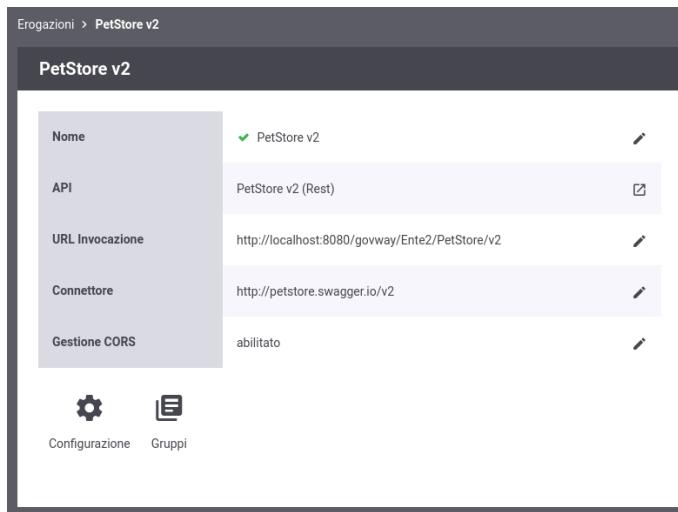


Figura 13: URL di Invocazione dell'API erogata

5. Invocazione API tramite GovWay

Al termine di questi passi di configurazione il servizio REST sarà raggiungibile dai client utilizzando l'url di invocazione:

- `http://host:port/govway/Ente2/PetStore/v2/<uri-risorsa>`

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente2/PetStore/v2/pet" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

6. Consultazione Tracce

La consultazione delle tracce per ogni dominio gestito è identica a quanto descritto nella sezione Sezione 1.1, previa selezione del soggetto in gestione tramite il menù situato in alto a destra.

1.3 Erogazione API SOAP

Procediamo adesso con la descrizione dei passi di configurazione necessari a registrare una API SOAP implementata da un applicativo interno al proprio dominio di gestione. L'applicativo implementa il servizio *Credit Card Verification* descritto in Sezione 1. In questo scenario di esempio si suppone che l'indirizzo <http://ws.cdyne.com/creditcardverify/luhnchecker.asmx> dove viene erogato il servizio sia interno al proprio dominio di gestione.

L'API, per questo esempio, viene registrata in modo che sia accessibile in forma anonima da qualunque client invocando l'url esposta da GovWay. Una rappresentazione di questo scenario è mostrata nella figura Figura 14. Prima di procedere con la configurazione effettuare il download dell'interfaccia WSDL disponibile in <https://ws.cdyne.com/creditcardverify/luhnchecker.asmx?wsdl>

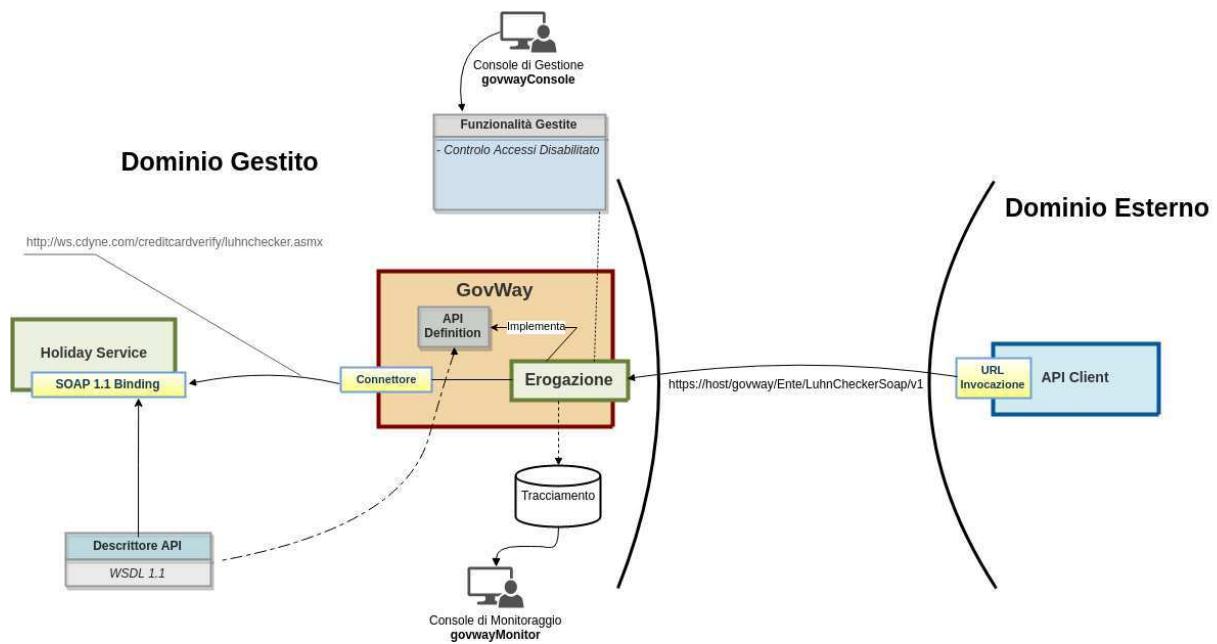


Figura 14: Erogazione di una API SOAP tramite GovWay

Per registrare l'API su Govway, utilizzando la console *govwayConsole*, procedere come segue:

1. Registrazione API.

Accedere alla sezione '*API*' e selezionare il pulsante '*Aggiungi*'. Fornire i seguenti dati:

- Tipo*: selezionare la tipologia '*SOAP*'.
- Nome*: indicare il nome dell'API che si sta registrando, ad esempio '*CreditCardVerification*'.
- Descrizione*: opzionalmente è possibile fornire una descrizione generica dell'API.
- Versione*: indicare la versione dell'API che si sta registrando; nell'esempio utilizziamo la versione *1*.
- WSDL*: caricare l'interfaccia WSDL scaricata dall'indirizzo <https://ws.cdyne.com/creditcardverify/luhnchecker.asmx?wsdl>

Screenshot della schermata di registrazione di una nuova API ('Aggiungi') nella console *govwayConsole*.

API > Aggiungi

Note: (*) Campi obbligatori

API

Tipo	Soap
Nome *	CreditCardVerification
Descrizione	Servizio di esempio per API SOAP
Versione	1

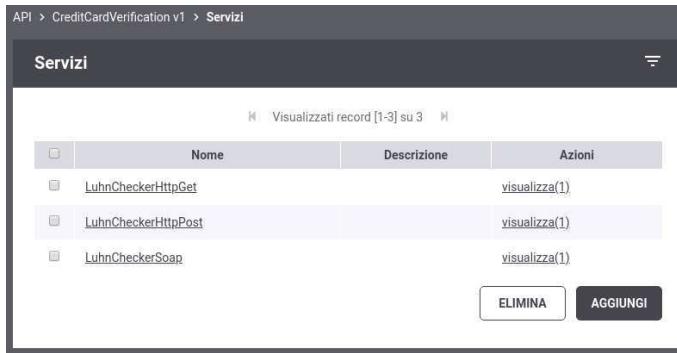
Specifiche delle interfacce

WSDL: Choose File No file chosen
luhnchecker.asmx?wsdl

SALVA

Figura 15: Registrazione di una API

Effettuato il salvataggio, l'API sarà consultabile all'interno dell'elenco delle API registrate. Accedendo al dettaglio si potranno visionare i servizi che tale API dispone che corrispondono ai *port type* presenti nell'interfaccia wsdl caricata. Come si può vedere dalla figura Figura 16 l'interfaccia *Credit Card Verification* possiede tre differenti servizi che corrispondono a differenti modalità di utilizzo. Nel seguito di questo esempio verrà utilizzato esclusivamente il servizio *LuhnCheckerSoap*.



Nome	Descrizione
LuhnCheckerHttpGet	visualizza(1)
LuhnCheckerHttpPost	visualizza(1)
LuhnCheckerSoap	visualizza(1)

Figura 16: Servizi di una API

2. Registrazione Erogazione

Accedere alla sezione '*Erogazioni*' e selezionare il pulsante '*Aggiungi*'. Fornire i seguenti dati:

- *Nome*: selezionare l'API precedentemente registrata '*CreditCardVerification v1*'.
- *Servizio*: selezionare uno dei servizi (port type) definiti nell'API precedentemente registrata '*LuhnCheckerSoap*'.
- *Autenticazione - Stato*: per esporre l'API in modo che sia invocabile da qualunque client in forma anonima selezionare lo stato '*disabilitato*'.
- *Connettore - Endpoint*: indicare l'endpoint dove viene erogata l'API nel dominio interno. Per il nostro esempio utilizzare la url:
 - <http://ws.cdyne.com/creditcardverify/luhnchecker.asmx>

Note: (*) Campi obbligatori

Informazioni Generali

API

- Nome: CreditCardVerification v1
- Tipo: Soap
- Servizio *: LuhnCheckerSoap

Autenticazione

Trasporto

- Stato: disabilitato

Connettore

- Endpoint *: http://ws.cdyne.com/creditcardverify/luhnchecker.asmx
- Autenticazione Http:
- AutenticazioneHttps:
- Proxy:
- Ridefinisci Tempi Risposta:

SALVA

Figura 17: Registrazione di una erogazione di API

Effettuato il salvataggio, l'API erogata sarà consultabile all'interno dell'elenco delle erogazioni. Accedendo al dettaglio si potrà conoscere l'*url di invocazione* che deve essere comunicata ai client che desiderano invocare l'API.

Erogazioni > LuhnCheckerSoap v1	
LuhnCheckerSoap v1	
Nome	LuhnCheckerSoap v1
API	CreditCardVerification v1 (Soap)
URL Invocazione	http://localhost:8080/govway/Ente/LuhnCheckerSoap/v1
Connettore	http://ws.cdyne.com/creditcardverify/luhnchecker.asmx
Gestione CORS	abilitato

Configurazione Gruppi

Figura 18: URL di Invocazione dell'API erogata

3. Invocazione API tramite GovWay

Al termine di questi passi di configurazione il servizio SOAP sarà raggiungibile dai client utilizzando l'url di invocazione:

- `http://host:port/govway/<soggetto-dominio-interno>/LuhnCheckerSoap/v1`

Soggetto Interno al Dominio

In questo esempio si suppone che il nome del soggetto fornito durante la fase di installazione di GovWay sia *Ente*.

```
curl -v -X POST "http://127.0.0.1:8080/govway/Ente/LuhnCheckerSoap/v1" \
-H 'Content-Type: text/xml; charset=UTF-8' \
-H 'SOAPAction: "http://ws.cdyne.com/CheckCC"' \
-d '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<CheckCC xmlns="http://ws.cdyne.com/">
<CardNumber>4111111111111111</CardNumber>
</CheckCC>
</soapenv:Body>
</soapenv:Envelope>'
```

L'esito della verifica viene ritornato con un codice http 200 e una risposta contenente i dettagli della carta:

```
HTTP/1.1 200 OK
Connection: keep-alive
Server: GovWay
GovWay-Message-ID: b62dc163-e788-4dc2-9cee-40c77b0a7a29
GovWay-Transaction-ID: fc155be0-c1ac-4e2e-93f7-d69a30258069
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
Date: Thu, 15 Nov 2018 13:34:22 GMT

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<CheckCCResponse xmlns="http://ws.cdyne.com/">
<CheckCCResult>
<CardType>VISA</CardType>
<CardValid>true</CardValid>
</CheckCCResult>
</CheckCCResponse>
</soap:Body>
</soap:Envelope>
```

Per simulare la medesima richiesta utilizzando un messaggio SOAP 1.2 è possibile usare la stessa url di invocazione :

```
curl -v -X POST "http://127.0.0.1:8080/govway/Ente/LuhnCheckerSoap/v1" \
-H 'Content-Type: application/soap+xml; charset=utf-8' \
-d '<soap12:Envelope xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
<soap12:Header/>
<soap12:Body>
<CheckCC xmlns="http://ws.cdyne.com/">
<CardNumber>4111111111111111</CardNumber>
</CheckCC>
</soap12:Body>
</soap12:Envelope>'
```

4. Consultazione Tracce

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway e recuperare i dettagli di una singola invocazione così come già descritto nella sezione Sezione 1.1.

1.4 Fruizione API

Procediamo adesso con la descrizione dei passi di configurazione necessari, ad un applicativo client interno al dominio di gestione, per poter fruire di una API REST esterna. L'API REST esterna utilizzata sarà lo *Swagger Petstore* descritto in Sezione 1 e poichè si suppone che tale scenario sia già stato provato non è necessario registrare nuovamente l'API.

In GovWay ad ogni dominio, interno o esterno, viene associato ad un Soggetto. Nella sezione Sezione 1.2 viene descritto come registrare più soggetti relativi a domini interni. In questo esempio, invece, procederemo con la registrazione di un soggetto esterno che rappresenta il gestore del dominio a cui appartiene il PetStore.

La fruizione di API, per questo primo esempio di utilizzo, viene registrata in modo che sia accessibile in forma anonima da qualunque client invocando l'url esposta da GovWay. Una rappresentazione di questo scenario è mostrata nella figura Figura 19.

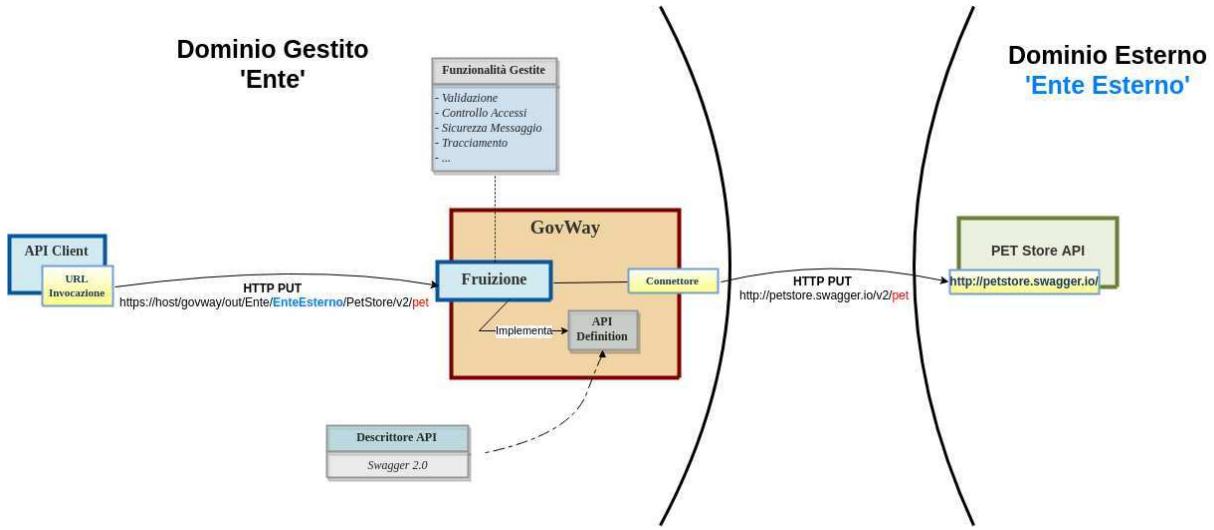


Figura 19: Fruizione di una API tramite GovWay

Per registrare l'API su Govway, utilizzando la console *govwayConsole*, procedere come segue:

1. Registrazione nuovo Soggetto del dominio esterno

Accedere alla sezione 'Soggetti' e selezionare il pulsante 'Aggiungi'. Fornire i seguenti dati:

- Dominio:** selezionare la voce 'Esterno'.
- Nome:** indicare il nome del Soggetto che rappresenta il nuovo dominio esterno, ad esempio 'EnteEsterno'.
- Tipologia:** selezionare la voce 'Erogatore'.
- Descrizione:** opzionalmente è possibile fornire una descrizione generica del soggetto.

Soggetto	
Dominio	Esterno
Nome *	EnteEsterno
Tipologia	Erogatore
Descrizione	

Figura 20: Registrazione nuovo Soggetto

2. Registrazione Fruizione

Accedere alla sezione 'Fruizioni' e selezionare il pulsante 'Aggiungi'. Fornire i seguenti dati:

- *API - Nome*: selezionare l'API precedentemente registrata '*PetStore v2*'.
- *Soggetto Erogatore - Nome*: selezionare il soggetto precedentemente registrato '*EnteEsterno*'.
- *Autenticazione - Stato*: per esporre l'API in modo che sia invocabile da qualunque client in forma anonima selezionare lo stato '*disabilitato*'.
- *Connettore - Endpoint*: indicare la *base uri* dove viene erogata l'API nel dominio esterno. Per il nostro esempio utilizzare la url:
 - *http://petstore.swagger.io/v2*

The screenshot shows the 'Aggiungi' (Add) screen for creating a new API usage ('Fruizione'). The form is divided into three main sections: 'Informazioni Generali', 'Autenticazione', and 'Connettore'. In 'Informazioni Generali', the 'Nome' field is set to 'PetStore v2' and the 'Soggetto Erogatore' field is set to 'EnteEsterno'. In 'Autenticazione', the 'Trasporto' section shows 'Stato' set to 'disabilitato'. In 'Connettore', the 'Endpoint' field contains the value 'http://petstore.swagger.io/v2'. At the bottom right of the form is a 'SALVA' (Save) button.

Figura 21: Registrazione di una fruizione di API

Effettuato il salvataggio, l'API erogata sarà consultabile all'interno dell'elenco delle fruizioni. Accedendo al dettaglio si potrà conoscere l'*url di invocazione* che deve essere comunicata ai client che desiderano invocare l'API.

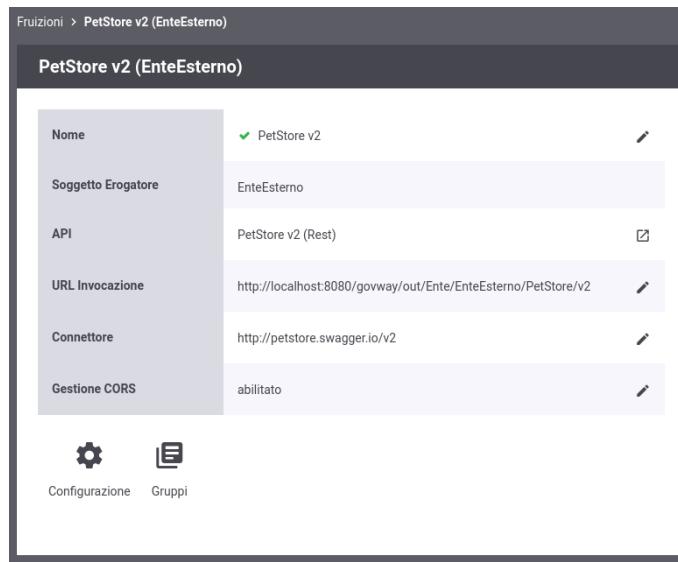


Figura 22: URL di Invocazione dell'API fruita

3. Invocazione API tramite GovWay

Al termine di questi passi di configurazione il servizio REST sarà raggiungibile dai client utilizzando l'url di invocazione:

- `http://host:port/govway/out/<soggetto-dominio-interno>/EnteEsterno/PetStore/v2/<uri-risorsa>`

Soggetto Interno al Dominio

In questo esempio si suppone che il nome del soggetto fornito durante la fase di installazione di GovWay sia *Ente*.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/out/Ente/EnteEsterno/PetStore/v2/pet" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento viene confermato con un codice http 200 e una risposta json equivalente alla richiesta:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Content-Type: application/json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Message-ID: 84e1d9a4-c181-436f-b7f0-4cabf55c370d
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
    "id":3,
    "category":{ "id":22, "name": "dog" },
    "name": "doggie",
```

```

    "photoUrls": ["http://image/dog.jpg"],
    "tags": [{"id": 23, "name": "white"}],
    "status": "available"
}

```

4. Consultazione Tracce

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway e recuperare i dettagli di una singola invocazione così come già descritto nella sezione Sezione 1.1.

2 Configurazioni differenti per gruppi di risorse di una API

Nei precedenti esempi tutte le risorse delle API REST o le azioni dei servizi SOAP vengono gestite dal Gateway tramite un'unica configurazione di default. Le funzionalità che verranno descritte nelle successive sezioni della guida (es. Sezione 5, Sezione 6, Sezione 7 ...) possono essere attivate tramite un'unica configurazione su tutte le risorse/azioni dell'API o possono essere distinte a seconda delle caratteristiche applicative di ogni singola risorsa o azione.

Di seguito, per fornire un esempio di raggruppamento delle risorse, ipotizziamo di classificare le operazioni del servizio *Swagger Petstore*, descritto in Sezione 1, per il metodo http:

- *POST, PUT*: per queste operazioni viene richiesta un'autenticazione *http basic*
- *DEL*: per queste operazioni viene richiesta un'autenticazione *https*
- *GET*: queste operazioni sono utilizzabili in forma anonima

Metodologia di classificazione solo a titolo di esempio

la classificazione per metodo http e i tipi di autenticazione utilizzati sono solamente a titolo di esempio per descrivere la possibilità di definire configurazioni differenti per gruppi di risorse.

Una rappresentazione di questo scenario è mostrata nella figura Figura 23.

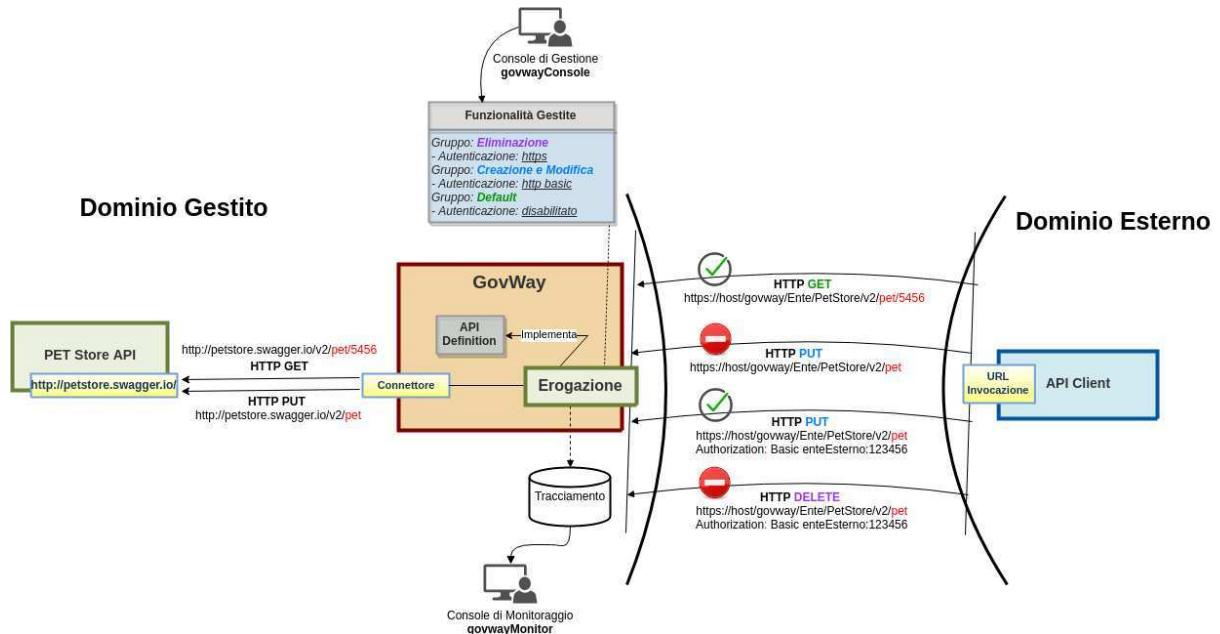


Figura 23: Configurazioni differenti per gruppi di risorse di una API

Per classificare in gruppi le risorse dell'API *Swagger Petstore*, utilizzando la console *govwayConsole*, procedere come segue:

1. Registrazione Gruppo 'Creazione e Modifica'

Accedere alla sezione '*Erogazioni*' e selezionare l'API precedentemente registrata '*PetStore v2*'. Dopodichè accedere, dal dettaglio dell'erogazione, alla sezione '*Gruppi*' dove vengono visualizzati i gruppi in cui sono state classificate le risorse. Per default è presente un solo gruppo *Predefinito* a cui sono associate tutte le risorse (vedi figura Figura 24).



Figura 24: Situazione iniziale con unico gruppo 'Predefinito'

Selezionare il pulsante '*Aggiungi*' e fornire i seguenti dati:

- Nome Gruppo*: permette di associare un nome al gruppo delle risorse. Per il nostro esempio utilizzare il nome 'Creazione e Modifica'.
- Risorse*: tramite la selezione multipla è possibile scegliere una o più risorse che dovranno appartenere al gruppo. Per il nostro esempio selezionare tutte le risorse con il metodo http *POST* e *PUT*.
- Modalità*: indica se deve essere clonata la configurazione a partire dal gruppo indicato o se bisogna creare una configurazione ex-novo. Per riprodurre lo scenario di esempio precedentemente descritto selezionare *Nuova*.
- Autenticazione - Stato*: per esporre l'API in modo che sia invocabile da client identificati tramite credenziali http-basic selezionare lo stato '*http basic*'.

The screenshot shows a modal dialog titled 'Aggiungi' (Add). It has two sections: 'Configurazione' (Configuration) and 'Autenticazione' (Authentication). In the Configuration section, 'Nome Gruppo' is set to 'Creazione e Modifica'. Under 'Risorse', a dropdown menu lists several methods: POST /pet, PUT /pet, GET /pet/findByStatus, GET /pet/findByTags, DELETE /pet/{petId}, GET /pet/{petId}, POST /pet/{petId}, POST /pet/{petId}/uploadImage, GET /store/inventory, and POST /store/order. The 'Modalità' dropdown is set to 'Nuova'. In the Authentication section, 'Trasporto' (Transport) is set to 'http-basic' and there is an optional checkbox. At the bottom is a 'SALVA' (Save) button.

Figura 25: Registrazione Gruppo 'Creazione e Modifica'

2. Registrazione Gruppo 'Eliminazione'

Procedere, come descritto in precedenza, per registrare un ulteriore gruppo fornendo i seguenti dati:

- *Nome Gruppo*: 'Eliminazione'.
- *Risorse*: Selezionare tutte le risorse con il metodo http *DEL*.
- *Modalità*: Per riprodurre lo scenario di esempio precedentemente descritto selezionare *Nuova*.
- *Autenticazione - Stato*: selezionare lo stato '*https*'.

Figura 26: Registrazione Gruppo 'Eliminazione'

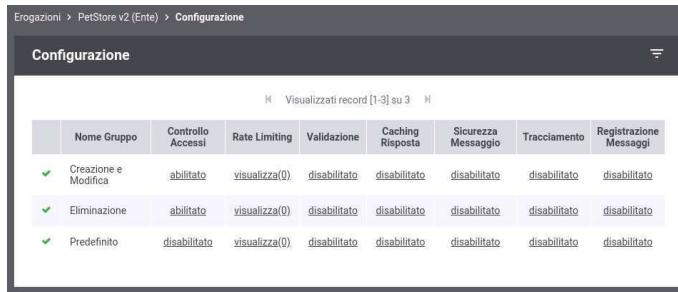
3. Verifica Gruppi Esistenti

Dal dettaglio dell'erogazione, accedere alla sezione '*Gruppi Risorse*' dove adesso verranno visualizzati tre gruppi, i due gruppi creati in precedenza ed il gruppo predefinito che adesso contiene solamente le risorse con metodo http GET (vedi figura Figura 27). In questa sezione sarà possibile agire sui gruppi anche in un secondo momento aggiungendo o eliminando risorse da un gruppo o creandone di nuovi.

Figura 27: Gruppi Registrati

Sempre dal dettaglio dell'erogazione, accedere alla sezione '*Configurazione*' dove vengono visualizzati i tre gruppi. In questa sezione sarà possibile configurare per ogni gruppo le funzionalità descritte nelle successive sezioni della guida (es. Sezione 5, Sezione 6, Sezione 7 ...). Si può notare come i due gruppi creati per l'esempio possiedano un *Controllo Accessi*

abilitato, mentre il gruppo *Predefinito* che contiene solo le risorse GET possiede tale funzionalità disabilitata. (vedi figura Figura 28).



The screenshot shows a table titled 'Configurazione' under the section 'Erogazioni > PetStore v2 (Ente) > Configurazione'. The table lists three groups: 'Creazione e Modifica', 'Eliminazione', and 'Predefinito'. The columns represent various configuration settings: Nome Gruppo, Controllo Accessi, Rate Limiting, Validazione, Caching Risposta, Sicurezza Messaggio, Tracciamento, and Registrazione Messaggi. The 'Predefinito' group is highlighted in red.

	Nome Gruppo	Controllo Accessi	Rate Limiting	Validazione	Caching Risposta	Sicurezza Messaggio	Tracciamento	Registrazione Messaggi
✓	Creazione e Modifica	abilitato	visualizza()	disabilitato	disabilitato	disabilitato	disabilitato	disabilitato
✓	Eliminazione	abilitato	visualizza()	disabilitato	disabilitato	disabilitato	disabilitato	disabilitato
✓	Predefinito	disabilitato	visualizza()	disabilitato	disabilitato	disabilitato	disabilitato	disabilitato

Figura 28: Configurazioni dei Gruppi

4. Reset Cache delle Configurazioni di GovWay

Le configurazioni accedute da GovWay vengono mantenute in una cache dopo il primo accesso per 2 ore. Siccome nei precedenti punti abbiamo modificato una configurazione utilizzata nelle sezioni precedenti se non sono trascorse 2 ore dall'ultimo utilizzo è necessario forzare un reset della cache. Per farlo accedere alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'. (vedi figura Figura 29).

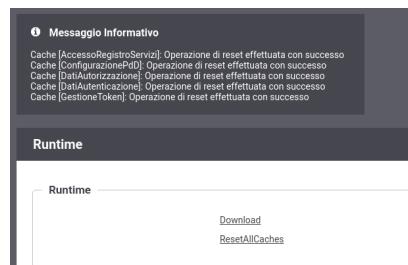


Figura 29: Reset Cache delle Configurazioni di GovWay

5. Invocazione Anonima di una Risorsa del gruppo 'Predefinito' completata con successo

Effettuando una richiesta di un animale tramite http method *GET* si può vedere come la richiesta completa con successo:

```
curl -v -X GET "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet/1" \
-H "accept: application/json"
```

L'esito dell'aggiornamento viene confermato con un codice http 200 e una risposta json equivalente alla richiesta:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Content-Type: application/json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Message-ID: 84e1d9a4-c181-436f-b7f0-4cabf55c370d
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
  "id":1,
  "category": { "id":1, "name":"Akuke" },
  "name":"roy",
  "photoUrls":["https://goo.gl/images/fxk2BX"],
```

```

    "tags": [{"id":0, "name":"Naughty Dog"}], "
  status": "available"
}

```

6. Invocazione Anonima di una Risorsa del gruppo 'Creazione e Modifica' terminata con errore

Effettuando una modifica di un animale tramite http method *PUT* si può vedere come la richiesta termina con errore causato dal fatto che non si sono fornite credenziali *http basic*:

```

curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
  "id": 3,
  "category": { "id": 22, "name": "dog" },
  "name": "doggie",
  "photoUrls": [ "http://image/dog.jpg" ],
  "tags": [ { "id": 23, "name": "white" } ],
  "status": "available"
}'

```

L'esito dell'aggiornamento termina con un codice http 401 e una risposta contenente un json di errore generato dal Gateway (*Problem Details* come definito nella specifica *RFC 7807*: <https://tools.ietf.org/html/rfc7807>):

```

HTTP/1.1 401 Unauthorized
Connection: keep-alive
WWW-Authenticate: Basic realm="GovWay"
Server: GovWay
Transfer-Encoding: chunked
GovWay-Transaction-ID: 15a60a91-edc1-4b7c-b7f0-b31739d543a0
Content-Type: application/problem+json
Date: Thu, 15 Nov 2018 16:07:10 GMT

{
  "type": "https://httpstatuses.com/401",
  "title": "Unauthorized",
  "status": 401,
  "detail": "Autenticazione fallita, credenziali non fornite",
  "govway_status": "protocol:GOVWAY-109"
}

```

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Dalla figura Figura 30 si può vedere come le transazioni con metodo http *PUT* sono terminate con errore con esito *Autenticazione Fallita*. Accedendo al dettaglio della singola invocazione fallita è possibile esaminare i diagnostici emessi da GovWay nei quali viene evidenziato il motivo del fallimento (figura Figura 31).

		Data Ingresso Richiesta	Tipologia	Esito	Fruitore	Erogatore	API	Azione
	●	2018-11-15 17:07:10	Erogazione	Autenticazione Fallita		Ente	PetStore v2	PUT_pet
	●	2018-11-15 17:03:43	Erogazione	Ok		Ente	PetStore v2	GET_pet.petId
	●	2018-11-15 17:03:09	Erogazione	Autenticazione Fallita		Ente	PetStore v2	PUT_pet

Figura 30: Tracce delle invocazioni transitate sul Gateway

Storio > Intervallo Temporale > Dettagli Transazione > Messaggi Diagnostici			
Data	Severità	Funzione	Messaggio
2018-11-15 17:07:10.917	infoIntegration	RicezioneBuste	Ricevuta richiesta applicativa
2018-11-15 17:07:10.919	infoIntegration	RicezioneBuste	Autenticazione basic in corso ...
2018-11-15 17:07:10.919	errorIntegration	RicezioneBuste	Autenticazione basic fallita: Autenticazione fallita, credenziali non fornite
2018-11-15 17:07:10.920	errorProtocol	RicezioneBuste	Generato messaggio di cooperazione di Errore con identificativo {bbbf9e08-9711-4709-a7e-2a603e95bad}
2018-11-15 17:07:10.921	infoIntegration	RicezioneBuste	Richiesta tipo: "http://httpstatus.es.com:401", title: "Unauthorized", status: 401, detail: "Autenticazione fallita, credenziali non fornite", protocol: GOWAY-109" consegnata al mittente con mittente di trasporto: 401

Figura 31: Dettaglio di una invocazione fallita bloccata dal Gateway

7. Invocazione di una Risorsa del gruppo 'Creazione e Modifica' con credenziali 'http basic' completata con successo

Per verificare che l'invocazione http descritta al punto precedente termini con successo in presenza di credenziali http basic si deve procedere con l'assegnazione di una credenziale ad un soggetto esterno al dominio. Di seguito viene descritto come fare tale assegnazione per completare l'esempio. Si rimanda poi alla sezione Sezione 5.2 per ulteriori dettagli sugli aspetti dell'autenticazione.

Accedere al soggetto *EnteEsterno* creato in precedenza durante l'esempio descritto nella sezione Sezione 1.4 e associargli delle credenziali 'http basic' come ad esempio un username *enteEsterno* ed una password *123456* (vedi figura Figura 32).

Soggetti > EnteEsterno

EnteEsterno

Note: (*) Campi obbligatori

Soggetto

Dominio: Esterno
Nome: * EnteEsterno
Descrizione:

Modalità di Accesso

Tipo: http-basic
Utente: * enteEsterno
Password: * 123456

Ruoli

Ruoli (0)

SALVA

Figura 32: Registrazione Gruppo 'Eliminazione'

Dopo aver associato le credenziali al soggetto effettuare il reset della cache delle configurazioni del Gateway come descritto in precedenza prima di procedere con l'invocazione.

Effettuando una modifica di un animale tramite http method *PUT* con le credenziali *http basic* si può vedere come la richiesta termina con successo:

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet" --basic --user ↵
enteEsterno:123456 \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
  "id": 3,
```

```

    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}

```

L'esito dell'aggiornamento viene confermato con un codice http 200 e una risposta json equivalente alla richiesta:

```

HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Content-Type: application/json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Message-ID: 84e1d9a4-c181-436f-b7f0-4cabf55c370d
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
  "id":3,
  "category": {"id":22,"name":"dog"},
  "name":"doggie",
  "photoUrls":["http://image/dog.jpg"],
  "tags":[{"id":23,"name":"white"}],
  "status":"available"
}

```

8. Invocazione di una Risorsa del gruppo 'Eliminazione' con credenziali 'http basic' terminata con errore

Effettuando una eliminazione di un animale tramite http method *DEL* si può vedere come la richiesta termina con errore causato dal fatto che non si sono fornite credenziali *https*:

```

curl -v -X DELETE "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet/545646489" --basic ←
  --user enteEsterno:123456 \
-H "accept: application/json"

```

L'esito dell'eliminazione termina con un codice http 401 e una risposta contenente un json di errore generato dal Gateway (*Problem Details* come definito nella specifica *RFC 7807*: <https://tools.ietf.org/html/rfc7807>):

```

HTTP/1.1 401 Unauthorized
Connection: keep-alive
Server: GovWay
Transfer-Encoding: chunked
GovWay-Transaction-ID: 15a60a91-edc1-4b7c-b7f0-b31739d543a0
Content-Type: application/problem+json
Date: Thu, 15 Nov 2018 16:07:10 GMT

{
  "type": "https://httpstatuses.com/401",
  "title": "Unauthorized",
  "status": 401,
  "detail": "Autenticazione fallita, credenziali non fornite",
  "govway_status": "protocol:GOVWAY-109"
}

```

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Dalla figura Figura 33 si può vedere come le transazioni con metodo http *DEL* sono terminate con errore con esito *Autenticazione Fallita*.

		Data Ingresso Richiesta ▾	Tipologia	Esito	Fruitore	Erogatore	API	Azione
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-16 10:22:09	Erogazione	Autenticazione Fallita		Ente	PetStore v2	DELETE_pet.petId
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-16 10:21:02	Erogazione	Autenticazione Fallita		Ente	PetStore v2	DELETE_pet.petId
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-16 10:20:56	Erogazione	Ok	EnteEsterno	Ente	PetStore v2	PUT_pet
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-16 10:20:44	Erogazione	Autenticazione Fallita		Ente	PetStore v2	PUT_pet
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-16 10:18:41	Erogazione	Ok		Ente	PetStore v2	GET_pet.petId
<input type="checkbox"/>	<input checked="" type="radio"/>	2018-11-16 10:18:32	Erogazione	Ok	EnteEsterno	Ente	PetStore v2	PUT_pet

Figura 33: Tracce delle invocazioni transitate sul Gateway

Ripristino Erogazione API con il solo gruppo predefinito per proseguo degli scenari

Nei scenari descritti nelle successive sezioni verrà utilizzato sempre il gruppo predefinito per mostrare la funzionalità. Per tale motivo si consiglia di ripristinare la situazione iniziale eliminando i due gruppi creati in questa sezione accedendo al dettaglio dell'erogazione dell'API *PetStore* nella sezione '*Gruppi*'.

3 Sospensione di una API

Una erogazione o una fruizione di API, precedentemente configurata, può essere temporaneamente sospesa. L'effetto di una sospensione è quella di bloccare sul gateway le richieste e di ritornare al client oltre all'informazione che il servizio non è disponibile una indicazione su quando può riprovare tramite l'header http standard *Retry-After*. Una sospensione è utile in diversi scenari quali ad esempio:

- *Aggiornamento applicativo erogatore*: Durante il periodo di aggiornamento di un applicativo erogatore una sospensione dell'erogazione permette di non intasare di richieste, che andrebbero in errore, il backend applicativo.
- *Problema applicativo client*: Supponiamo che un applicativo client produca delle richieste, verso un dominio esterno, che generano errori dovuti a problemi del software del client. Una volta identificato il problema, per evitare di intasare di richieste errate il Dominio esterno può essere funzionale sospendere la fruizione dell'API fino a che il problema non viene risolto.

Una rappresentazione di questo scenario è mostrata nella figura Figura 34.

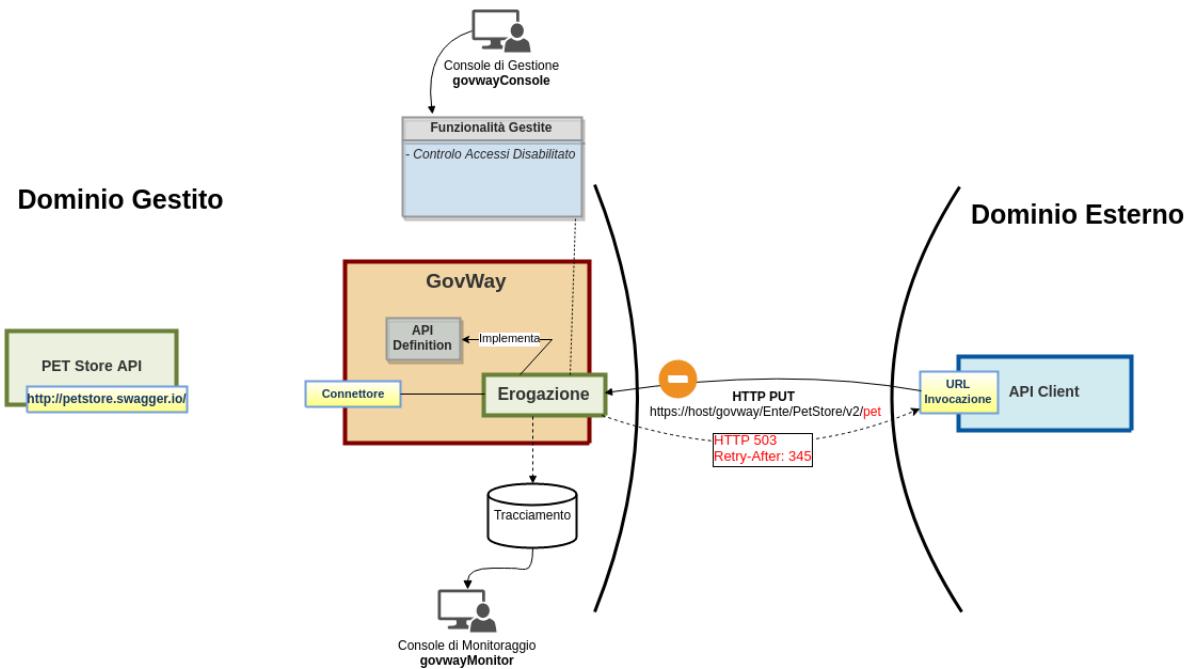


Figura 34: Sospensione di una API

Per sospendere una erogazione o fruizione di API, utilizzando la console *govwayConsole* dal dettaglio dell'erogazione o della fruizione accedere alla sezione '*Configurazione*'. Cliccando sull'icona di stato verde comparirà una finestra di dialogo dove viene richiesto di confermare la sospensione. La figura Figura 35 mostra una sospensione in corso dell'erogazione registrata nella sezione Sezione 1.1.

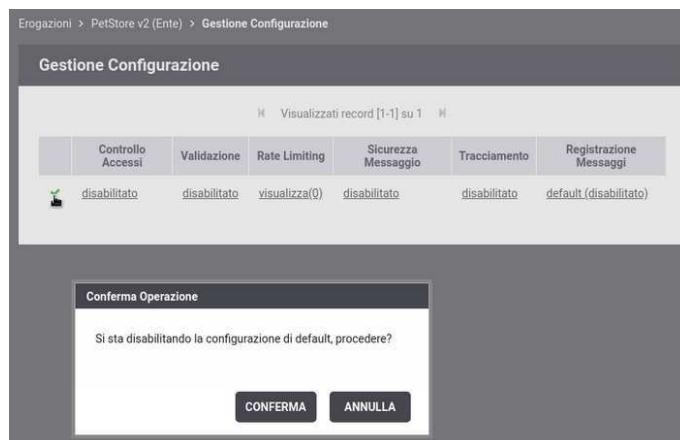


Figura 35: Sospensione di una erogazione

Procedendo con la conferma l'erogazione sarà a tutti gli effetti sospesa come mostra anche l'icona di stato rossa (vedi figura Figura 36).



Figura 36: Erogazione sospesa

L'informazione sullo stato di sospensione di una erogazione o una fruizione viene fornita, tramite l'icona di stato, anche nell'elenco principale come mostrato nella figura Figura 37.

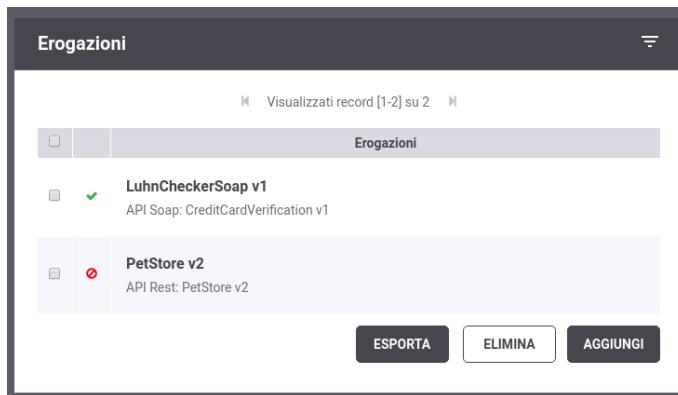


Figura 37: Stato disabilitato riportato nell'elenco delle erogazioni

Effettuando una modifica di un animale tramite http method *PUT* si può vedere come la richiesta termina con errore causato dal fatto che l'erogazione risulta sospesa:

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento termina con un codice http 503, un header http *Retry-After* contenente l'indicazione sul numero di secondi dopo i quali un client dovrebbe ripresentarsi e una risposta contenente un json di errore generato dal Gateway (*Problem Details* come definito nella specifica *RFC 7807*: <https://tools.ietf.org/html/rfc7807>):

```
HTTP/1.1 503 Service Unavailable
Connection: keep-alive
Retry-After: 338
Server: GovWay
Transfer-Encoding: chunked
GovWay-Transaction-ID: 15a60a91-edc1-4b7c-b7f0-b31739d543a0
Content-Type: application/problem+json
Date: Thu, 15 Nov 2018 16:07:10 GMT
```

```
{
  "type": "https://httpstatuses.com/503",
  "title": "Service Unavailable",
  "status": 503,
  "detail": "Porta disabilitata",
  "govway_status": "integration:GOVWAY-446"
}
```

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Dalla figura Figura 38 si può vedere come le transazioni generate dopo la sospensione sono terminate con esito *API Sospesa*.

		Data Ingresso Richiesta	Tipologia	Esito	Fruitore	Erogatore	API	Azione
	■	2018-11-16 12:09:25	Erogazione	API Sospesa		Ente	PetStore v2	PUT_pet
	■	2018-11-16 12:09:20	Erogazione	API Sospesa		Ente	PetStore v2	PUT_pet

Figura 38: Tracce delle invocazioni transitate sul Gateway

Se per una erogazione o fruizione di API è stata effettuata la classificazione delle risorse in gruppi, come mostrato nella sezione Sezione 2, la sospensione può essere effettuata sul singolo gruppo.

La figura Figura 39 mostra un esempio di sospensione, nello scenario Sezione 2, del solo gruppo '*Predefinito*'.

Erogazioni > PetStore v2 (Ente) > Configurazione							
Configurazione							
	Nome Gruppo	Controllo Accessi	Rate Limiting	Validazione	Caching Risposta	Sicurezza Messaggio	Tracciamento
✓	Creazione e Modifica	abilitato	visualizza()	disabilitato	disabilitato	disabilitato	disabilitato
✓	Eliminazione	abilitato	visualizza()	disabilitato	disabilitato	disabilitato	disabilitato
●	Predefinito	disabilitato	visualizza()	disabilitato	disabilitato	disabilitato	disabilitato

Figura 39: Gruppo di una erogazione sospeso

L'informazione sullo stato di sospensione parziale (relativa a non tutti i gruppi) di una erogazione o una fruizione viene fornita, tramite un icona di stato gialla, anche nell'elenco principale come mostrato nella figura Figura 40.

Erogazioni	
	Erogazioni
■	✓ LuhnCheckerSoap v1 API Soap: CreditCardVerification v1
■	▼ PetStore v2 API Rest: PetStore v2

Figura 40: Stato disabilitato di un gruppo riportato nell'elenco delle erogazioni

4 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.

Una rappresentazione di questo scenario è mostrata nella figura Figura 41.

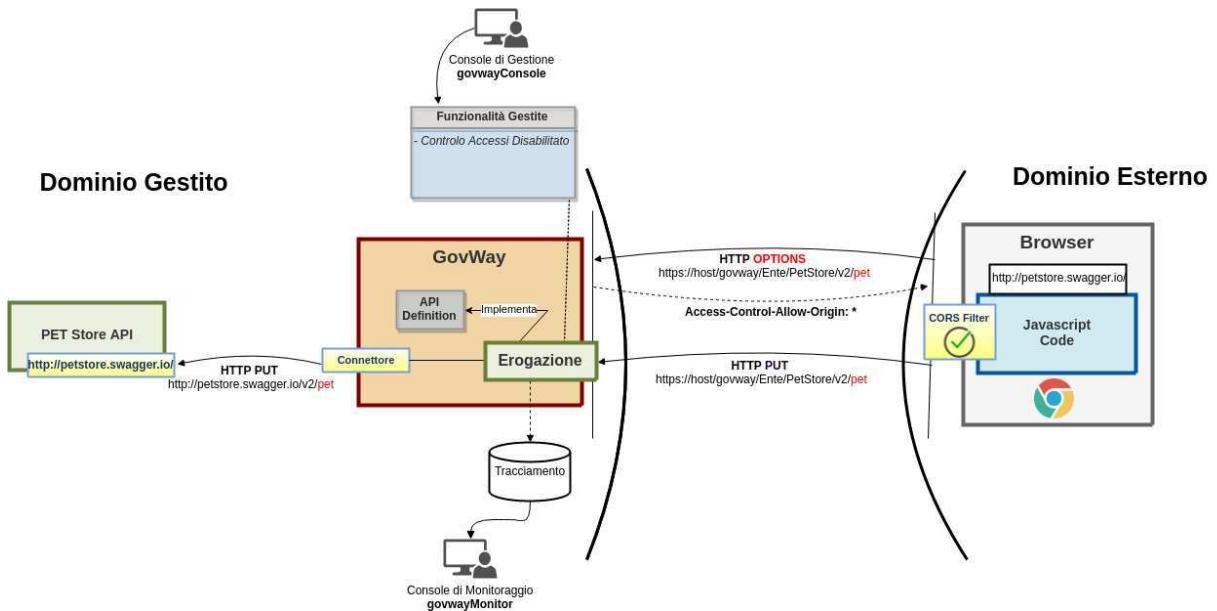


Figura 41: Scenario cross-origin HTTP request (CORS)

In GovWay è abilitata per default una gestione globale del CORS. I dettagli sulla configurazione globale sono accedibili tramite la voce del menu 'Configurazione - Generale' all'interno della sezione 'Gestione CORS'. Per il dettaglio sul significato di ogni voce si rimanda alla specifica CORS <https://www.w3.org/TR/cors/>. Sono abilitati per default:

- *Access-Control-Allow-Origin*: Qualsiasi origine (*)
- *Access-Control-Allow-Methods*: i metodi http POST, PUT, GET, DELETE e PATCH
- *Access-Control-Allow-Headers*: gli header http 'Authorization', 'Content-Type' e 'SOAPAction'

La figura Figura 42 mostra la configurazione globale attiva per default.

The screenshot shows the 'Gestione CORS' (CORS Management) configuration page. At the top, there are dropdown menus for 'Stato' (Enabled) and 'Tipo' (Managed by Gateway). Below these are sections for 'Access Control': 'All Allow Origins' (checked), 'Allow Headers' (containing Authorization, Content-Type, SOAPAction), and 'Allow Methods' (containing GET, PUT, POST, DELETE, PATCH). There is also an 'Allow Credentials' checkbox.

Figura 42: CORS - Configurazione di default

Tramite il tool on-line disponibile all’indirizzo <https://www.test-cors.org/> è possibile verificare il funzionamento dello scenario descritto nella figura Figura 41. Configurare il tool con i seguenti parametri per utilizzare il servizio descritto nella sezione Sezione 1.1:

- *HTTP Method: PUT*

- *Request Headers:*

- accept: application/json
- Content-Type: application/json

- *Request Content:*

```
{  
    "id": 3,  
    "category": { "id": 22, "name": "dog" },  
    "name": "doggie",  
    "photoUrls": [ "http://image/dog.jpg" ],  
    "tags": [ { "id": 23, "name": "white" } ],  
    "status": "available"  
}
```

- *Remote URL: http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet*

Se si attiva la modalità *Developers Tool* (es. su Chrome ’More Tools - Developers Tool’) è possibile vedere le richieste effettuate dal browser oltre agli header http scambiati.

Nella figura Figura 43 è possibile vedere come siano state effettuate due richieste http di cui la prima è stata iniziata dal browser (Initiator: corsclient.js).

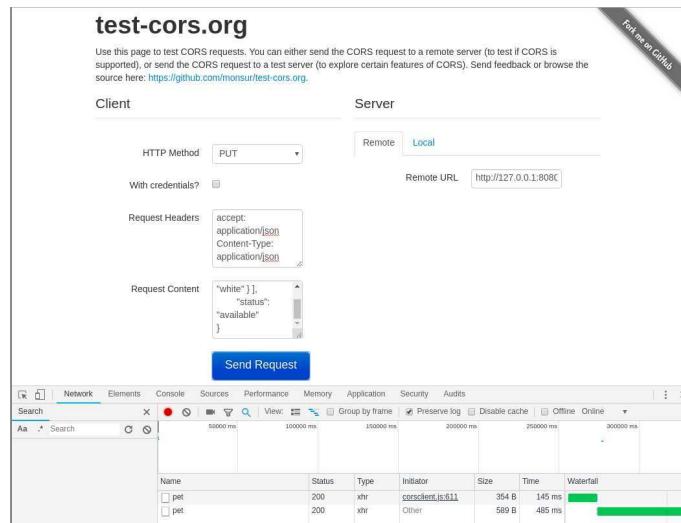


Figura 43: Verifica CORS

La figura Figura 44 evidenzia gli header scambiati nella prima richiesta OPTIONS; tra gli header della risposta vi sono gli header relativi alla configurazione di default del CORS di GovWay tra cui l'header 'Access-Control-Allow-Origin' impostato al valore '*'.

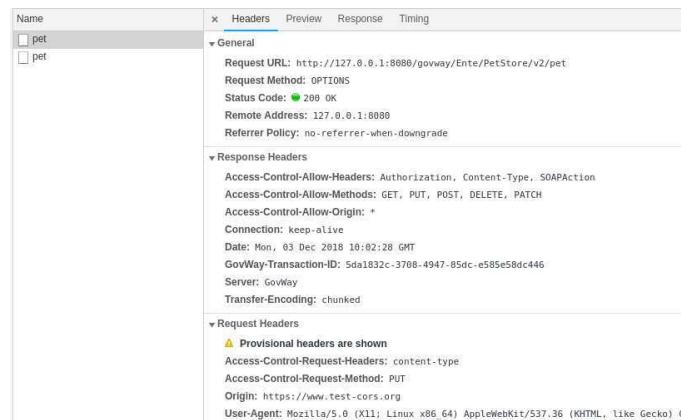


Figura 44: Verifica CORS: richiesta OPTIONS

Vediamo adesso come modificare la gestione del CORS di una singola erogazione o fruizione di API utilizzando la console *govwayConsole*. Per farlo accedere al dettaglio di un'erogazione o di una fruizione e cliccare sull'icona di modifica presente nella riga relativa alla gestione del CORS.

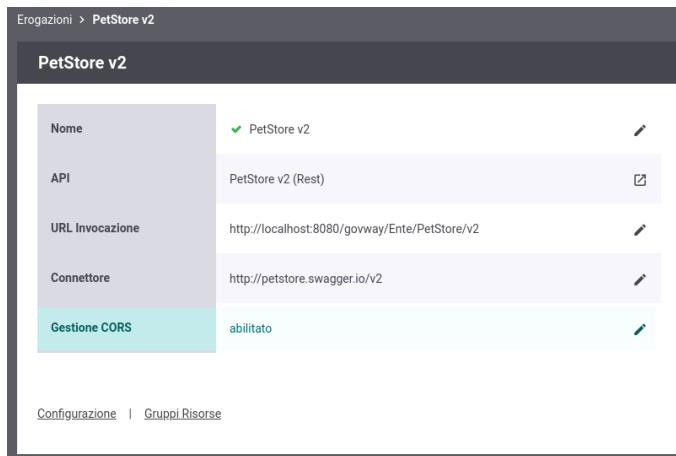


Figura 45: Personalizzazione Gestione CORS di una erogazione

Impostare il campo *Stato* al valore *Ridefinito*. La maschera di configurazione si aggiornerà presentando i dati relativi alla configurazione globale di default. Deselezionare a questo punto la voce '*All Allow Origins*' ed impostare un'origine specifica nel campo '*Allow Origins*'. Ad esempio utilizzare il valore '<https://www.test-cors.org>' relativo al tool di test descritto in precedenza.

The screenshot shows the 'Gestione CORS' configuration form. The 'Access Control' section is expanded, showing the following settings:

- All Allow Origins:** Unchecked
- Allow Origins:** https://www.test-cors.org
- Allow Headers:** Authorization, Content-Type, SOAPAction
- Allow Methods:** GET, PUT, POST, DELETE, PATCH
- Allow Credentials:** Unchecked

A note at the top states: "Note: (*) Campi obbligatori". A 'SALVA' button is at the bottom.

Figura 46: Personalizzazione Gestione CORS: definizione di uno specifico 'origin'

Effettuando un nuovo test tramite il tool on-line *test-cors* è possibile vedere nella prima richiesta OPTIONS, che tra gli header della risposta non vi è più l'header 'Access-Control-Allow-Origin' impostato al valore '*' ma bensì con il nuovo valore configurato.

Reset Cache delle Configurazioni prima di un nuovo test

Le configurazioni accedute da GovWay vengono mantenute in una cache dopo il primo accesso per 2 ore, è quindi necessario forzare un reset della cache. Per farlo accedere alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Figura 47: Verifica CORS: definizione di uno specifico 'origin'

5 Controllo degli Accessi

5.1 OAuth

GovWay permette di proteggere le erogazioni e/o fruizioni di API tramite il protocollo *OAuth2*. Una API può essere configurata in modo che ogni sua invocazione debba essere accompagnata da un *access token* valido rilasciato da uno degli *Authorization Server* censiti.

La figura Figura 48 mette in evidenza tutte le comunicazioni e gli attori coinvolti per riuscire a porta a termine l'invocazione dello scenario descritto nella sezione Sezione 1.1 dove però l'api viene protetta tramite *OAuth*.

1. Acquisizione Access Token

Un client deve richiedere un *access token* direttamente all'*Authorization Server* secondo le modalità supportate. In OAuth esistono diverse modalità alcune delle quali richiedono anche il coinvolgimento dell'utente al quale verrà richiesto di autenticarsi e poi di autorizzare le operazioni che il client intende eseguire. ([RFC 6749](#))

2. Richiesta di servizio con Access Token

Un client ottenuto l'*access token* deve spenderlo all'interno della richiesta inoltrata a GovWay già descritta nella sezione Sezione 1.1. Un *access token* può essere incluso nella richiesta tramite diverse modalità definite dalla specifica [RFC 6750](#). Nello scenario di esempio è stato utilizzato l'header *http Authorization* utilizzando la modalità *Bearer*.

3. Validazione Access Token

GovWay verifica che la richiesta contenga un *access token* valido. Per effettuare tale validazione GovWay supporta differenti modalità:

- *Servizio di Introspection*: se l'*access token* è 'opaco' l'unica maniera per validarla è accedere al servizio di introspection che deve essere disponibile sull'*Authorization Server*. Tale servizio viene definito dalla specifica [RFC 7662](#)
- *Validazione JWT*: se l'*access token* è un token 'JWT' ([RFC 7519](#)) GovWay può essere configurato per validarla secondo la specifica JWS ([RFC 7515](#)) o JWE ([RFC 7516](#)). direttamente sul gateway senza accedere ad alcun servizio remoto.

4. Forward Claims dell'Access Token

Effettuata la validazione dell'*access token* GovWay può fornire all'applicativo erogatore le varie informazioni acquisite durante la validazione del token, ad esempio sotto forma di header *http*.

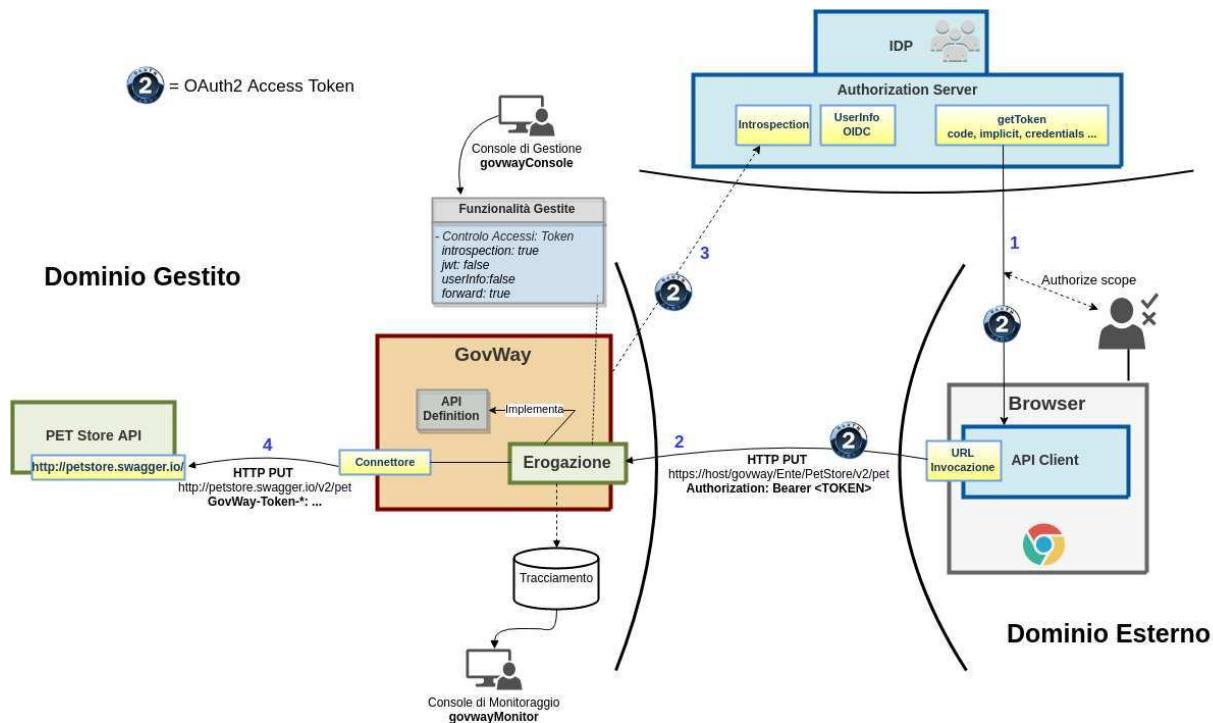


Figura 48: Scenario OAuth

Come si evince dalla figura Figura 48 la creazione del token non è gestita da GovWay, ma da un qualunque Authorization Server esterno. GovWay è preconfigurato per poter utilizzare Google come Authorization Server nell'installazione di base e quindi un applicativo può ottenere il token da Google e poi spenderlo all'interno delle richieste applicative spedite verso GovWay.

Lo scenario istanziato su Google sarà utilizzato in tutte le successive sotto-sezioni per descrivere tutte le funzionalità inerenti *OAuth2* attivabili su GovWay.

Requisito: account gmail

Per provare gli scenari descritti nelle successive sotto-sezioni è necessario avere un account su gmail.

5.1.1 Validazione tramite Introspection

In questa sezione viene descritto come realizzare lo scenario raffigurato nella figura Figura 48 dove GovWay utilizza il servizio Introspection dell'*Authorization Server di Google* per validare l'*access token* ricevuto.

- **Configurazione Controllo degli Accessi**

Accedere alla sezione '*Erogazioni*' e selezionare l'API precedentemente registrata '*PetStore v2*'. Dopodichè accedere, dal dettaglio dell'erogazione, alla sezione '*Configurazione*' dove vengono visualizzate le funzionalità attive. Per abilitare una protezione dell'api basata su *OAuth* cliccare sulla voce presente nella colonna '*Controllo Accessi*' e procedere con la seguente configurazione all'interno della sezione '*Gestione Token*':

- *Stato*: abilitato
 - *Policy*: Google
 - *Validazione JWT*: disabilitato
 - *Introspection*: abilitato
 - *User Info*: disabilitato
 - *Token Forward*: abilitato

Effettuata la configurazione salvarla cliccando sul pulsante 'Salva'.

The screenshot shows the 'Controllo Accessi' (Access Control) configuration page for the PetStore v2 service. It includes three main sections: 'Gestione Token' (Token Management), 'Autenticazione' (Authentication), and 'Autorizzazione' (Authorization). In the 'Gestione Token' section, the 'Policy' is set to 'Google'. The 'Autenticazione' section contains fields for 'Trasporto' (Transport) and 'Token' (Issuer, ClientId, Subject, Username, eMail). The 'Autorizzazione' section has a single 'State' dropdown set to 'disabilitato'. At the bottom is a 'SALVA' (Save) button.

Figura 49: Configurazione OAuth2 per PetStore

- **Invocazione API senza un access token**

Reset Cache delle Configurazioni prima di un nuovo test

Le configurazioni accedute da GovWay vengono mantenute in una cache dopo il primo accesso per 2 ore, è quindi necessario forzare un reset della cache. Per farlo accedere alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Al termine di questi passi di configurazione il servizio REST sarà invocabile solamente se viene fornito un *access token*. Con il seguente comando è possibile constatare come una richiesta che non possieda l'*access token* viene rifiutata da GovWay.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento termina con un codice di errore http 400 e una risposta problem+json che riporta la motivazione:

```

HTTP/1.1 400 Bad Request
WWW-Authenticate: Bearer realm="Google", error="invalid_request", error_description="The ←
    request is missing a required token parameter"
Content-Type: application/problem+json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
  "type": "https://httpstatuses.com/400",
  "title": "Bad Request",
  "status": 400,
  "detail": "Token non presente",
  "govway_status": "protocol:GOVWAY-1366"
}

```

• Consultazione Tracce in errore

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Dalla figura Figura 50 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con errore con esito *Gestione Token Fallita*.

Lista Transazioni: record [1 - 3]							
	Data Ingresso Richiesta	Tipologia	Esito	Fruitore	Erogatore	API	Azione
<input type="checkbox"/>	2018-12-04 12:13:37	Erogazio...	Gestione Token Fallita	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-04 12:13:36	Erogazio...	Gestione Token Fallita	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-04 12:08:37	Erogazio...	Gestione Token Fallita	Ente	PetStore v2	PUT_pet	

Figura 50: Tracce delle invocazioni terminate con errore 'Gestione Token Fallita'

• Acquisizione Access Token

Per simulare l'acquisizione di un token è possibile utilizzare l'applicazione *Playground*, disponibile all'indirizzo <https://developers.google.com/oauthplayground/>, che consente di richiedere un *access token* all'*Authorization Server di Google*.

L'applicazione *Playground* consente agevolmente di ottenere l'*access token*:

1. Selezione scope

Devono essere selezionati gli *scope* che un'applicazione client necessita per invocare poi effettivamente le API di Google. Ad esempio selezioniamo lo scope '<https://www.googleapis.com/auth/plus.me>' che permette all'applicazione di conoscere l'identità di un utente su google. Cliccando infine sul pulsante '*Authorize APIs*' si verrà rediretti alla pagina di autenticazione in google dove si dovrà procedere ad autenticarsi.

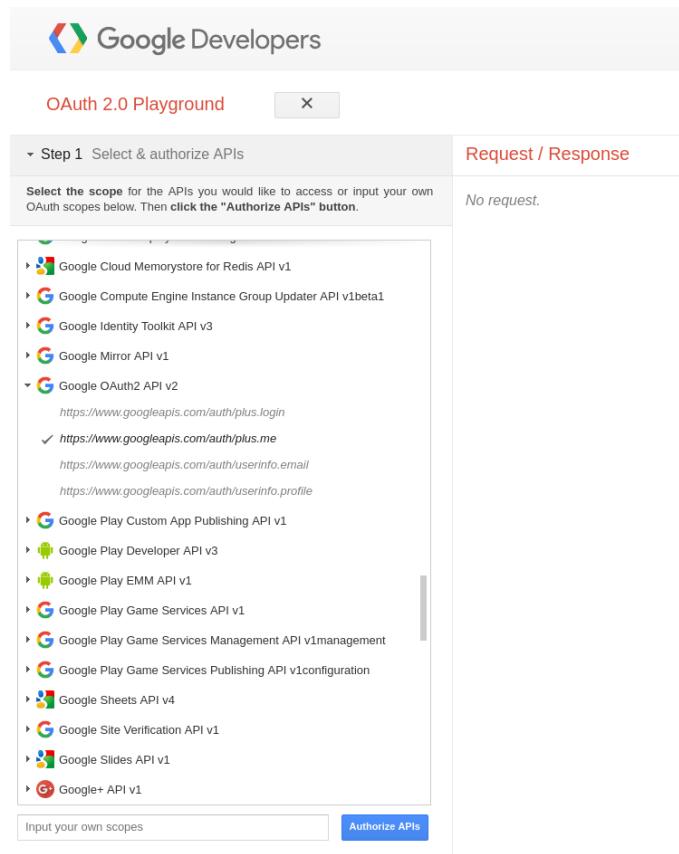


Figura 51: Ottenimento Token: Playground Google, Step 1

2. Authorization Code

Effettuata l'autenticazione in Google si viene rediretti alla seconda fase prevista dall'applicazione *Playground* denominata '*Exchange authorization code for tokens*'.

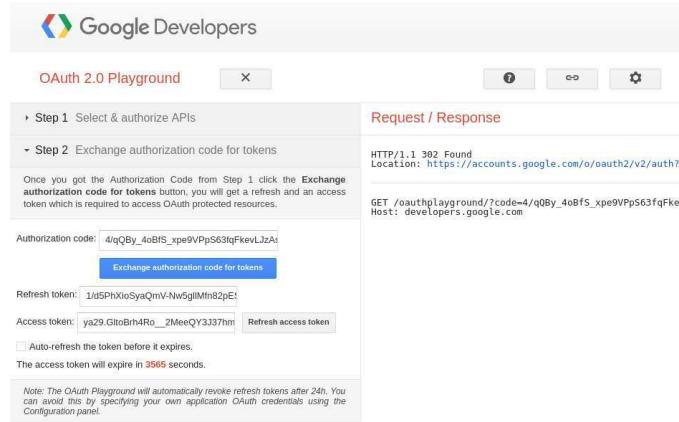


Figura 52: Ottenimento Token: Playground Google, Step 2

3. Access Token

Cliccando sul pulsante '*Exchange authorization code for tokens*' si ottiene infine un *access token* da estrarre nella risposta http visualizzata sulla destra dell'applicazione.

Request / Response

```

POST /oauth2/v4/token HTTP/1.1
Host: www.googleapis.com
Content-length: 277
content-type: application/x-www-form-urlencoded
user-agent: google-oauth-playground
code=4%2Fq0ctBFrJConLp2VmBckP40w0JqeAgPj56QlAuiKyn4Dz4dY9epFi7nfIn-pxgyx0ukXlhxp_SC7rc0dq8bZnE8&redirect_uri

HTTP/1.1 200 OK
Content-length: 1097
X-xss-protection: 1; mode=block
X-content-type-options: nosniff
Transfer-encoding: chunked
Vary: Origin, X-Origin, Referer
Server: ES
Content-encoding: gzip
Cache-control: private
Date: Tue, 04 Dec 2018 10:57:27 GMT
X-Content-Type-Options: nosniff
Alt-svc: quic=:443; ma=2592000; v="44,43,39,35"
Content-type: application/json; charset=utf-8

{
  "access_token": "ya29.GltBuFj390CX50k-ea3aZoZgH29RHqUhZ06e3IU46gp9IdbbpJLB83Ygo27RYGYmef7sibN9rnNb1r88b8X
  id_token": "eyJhbGciOiJSUzI1Ni1SImtpZC10IJ02pZz1NDgwY2NjNTgzonJ1YjE10DYXzTA4Y2MzDE4M2ZhZj1hNTY1LCJ0eXA101
  expires_in": 3600,
  "token_type": "Bearer",
  "scope": "https://www.googleapis.com/auth/plus.me",
  "refresh_token": "1/d9PhXloSya0mV-Nw5glMfn82pEScg[usu0f7_ULYR0"
}

```

Figura 53: Ottenimento Token: Playground Google, Step 3

- Invocazione API con un access token**

Con il seguente comando è possibile effettuare una richiesta che possiede l'*access token* ottenuto nella precedente fase.

Bearer Token Usage

Un *access token* può essere incluso nella richiesta tramite una delle modalità definite dalla specifica [RFC 6750](#).

```

curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token= ←
  ACCESS_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'

```

L'esito dell'aggiornamento viene confermato con un codice http 200 e una risposta json equivalente alla richiesta:

```

HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Content-Type: application/json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Message-ID: 84e1d9a4-c181-436f-b7f0-4cabf55c370d
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
  "id":3,
  "category":{ "id":22, "name": "dog" },
  "name": "doggie",
  "photoUrls": [ "http://image/dog.jpg" ],
  "tags": [ { "id": 23, "name": "white" } ],
  "status": "available"
}

```

- Consultazione Tracce**

Attraverso la console *govwayMonitor* è possibile adesso vedere che le richieste transitano con successo sul gateway. Accedendo al dettaglio di una transazione, tra le varie informazioni presenti nella sezione '*Informazioni Mittente*', sono presenti le informazioni principali estratte dal token (es. Subject presente nel claim 'sub').

The screenshot shows a 'Token Info' section with the following details:

- Client ID: 407408718192.apps.googleusercontent.com
- Subject: 106235657592654397689
- Token Info: [Visualizza](#)

Figura 54: Traccia di una invocazione terminata con successo

Cliccando sul link '*Visualizza*' della voce '*Token Info*' è possibile vedere tutti i claims presenti nel token, tra cui è possibile constatare la presenza del claim *scope* valorizzato con quanto richiesto tramite l'applicazione Playground.

The JSON output of the token is as follows:

```
{
  "valid": true,
  "aud": "106235657592654397689",
  "sub": "106235657592654397689",
  "exp": 1543925775889,
  "client_id": "407408718192.apps.googleusercontent.com",
  "scopes": ["https://www.googleapis.com/auth/plus.me"],
  "access_type": "offline",
  "token_type": "Bearer",
  "claims": {
    "aud": "407408718192.apps.googleusercontent.com",
    "sub": "106235657592654397689",
    "access_type": "offline",
    "token_type": "Bearer",
    "scope": "https://www.googleapis.com/auth/plus.me",
    "exp": 1543925775,
    "expires_in": 3566
  },
  "rawResponse": "\n  \"azp\": \"407408718192.apps.googleusercontent.com\", \"aud\": \"407408718192.apps.googleusercontent.com\", \"sub\": \"106235657592654397689\", \"exp\": 1543925775889, \"scope\": \"https://www.googleapis.com/auth/plus.me\", \"access_type\": \"offline\", \"token_type\": \"Bearer\", \"claims\": { \"aud\": \"407408718192.apps.googleusercontent.com\", \"sub\": \"106235657592654397689\", \"exp\": 1543925775, \"scope\": \"https://www.googleapis.com/auth/plus.me\", \"access_type\": \"offline\" }, \"sourceType\": \"INTROSPECTION\"\n"
}
```

[DOWNLOAD](#)

Figura 55: Informazioni ottenute tramite Introspection del Token

• Invocazione API con un access token non valido

GovWay utilizza il servizio Introspection di Google per validatore l'*access token* ricevuto. E' possibile ottenere un errore di validazione attendendo che l'access token scada o falsificandolo modificando ad esempio i primi caratteri.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token= ←
ERR_ACCESS_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
  "id": 3,
  "category": { "id": 22, "name": "dog" },
  "name": "doggie",
  "photoUrls": [ "http://image/dog.jpg" ],
  "tags": [ { "id": 23, "name": "white" } ],
  "status": "available"
}'
```

L'esito dell'aggiornamento termina con un codice di errore http 401 e una risposta problem+json che riporta la motivazione:

```
HTTP/1.1 401 Unauthorized
```

```
WWW-Authenticate: Bearer realm="Google", error="invalid_token", error_description="Token ←  
invalid"  
Content-Type: application/problem+json  
Transfer-Encoding: chunked  
Server: GovWay  
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824  
  
{  
  "type": "https://httpstatuses.com/401",  
  "title": "Unauthorized",  
  "status": 401,  
  "detail": "Token non valido",  
  "govway_status": "protocol:GOVWAY-1367"  
}
```

• Forward Token Info all'Applicativo

La configurazione descritta precedentemente indicava di abilitare la funzionalità '*Token Forward*' all'interno della sezione '*Gestione Token*' (vedi Figura 49). Tale configurazione fa sì che GovWay inoltri all'applicativo interno al dominio (nel nostro esempio il servizio *PetStore*) le informazioni inerenti il token ricevuto sotto forma di header http. Differenti modalità di consegna di tali informazioni vengono descritte nella sezione Sezione 5.1.7.

Per vedere quali header vengono effettivamente prodotti possiamo utilizzare la funzionalità '*Registrazione Messaggi*' descritta nel dettaglio nella sezione Sezione 10. Accedere alla sezione '*Erogazioni*' e selezionare l'API precedentemente registrata '*PetStore v2*'. Dopodichè accedere, dal dettaglio dell'erogazione, alla sezione '*Configurazione*' dove vengono visualizzate le funzionalità attive. Per abilitare la registrazione degli header cliccare sulla voce presente nella colonna '*Registrazione Messaggi*' e procedere con la seguente configurazione.

- '*Generale - Stato*': ridefinito
- '*Richiesta - Stato*': abilitato
- '*Richiesta - Ingresso*': disabilitare tutte le voci
- '*Richiesta - Uscita*': abilitare solo la voce relativa agli header
- '*Risposta - Stato*': disabilitato

Effettuata la configurazione salvarla cliccando sul pulsante 'Salva'.

Erogazioni > PetStore v2 (Ente) > Configurazione > Registrazione Messaggi

Registrazione Messaggi

Generale

Stato: ridefinito

Richiesta

Stato: abilitato

Ingresso

Headers: disabilitato
Body: disabilitato
Attachments: disabilitato

Uscita

Headers: abilitato
Body: disabilitato
Attachments: disabilitato

Risposta

Stato: disabilitato

SALVA

Figura 56: Configurazione Registrazione Messaggi per visualizzare Header HTTP

Prima di procedere con una nuova richiesta effettuare il reset della cache delle configurazioni accedendo alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Effettuare quindi una nuova invocazione contenente un *access token* valido e successivamente consultare il dettaglio della transazione tramite la *govWayMonitor*. Nel dettaglio sarà adesso disponibile la voce '*Contenuti Uscita*' (Figura 57) che permette di vedere gli header http prodotti da GovWay (Figura 58).

Storico > Intervallo Temporale > Dettaglio Transazione

Dettagli Transazione

Informazioni Generali

Tipologia	Erogazione (API Gateway)
Erogatore	Ente
API	PetStore v2
Azione	PUT_pet
Profilo Collaborazione	Sincrono
Esito	Ok
Diagnostici	Visualizza Esporta

Dettagli Richiesta

ID Messaggio	6f6c1374-8744-4345-81ba-534ca8ca0793
Data Ingresso	2018-12-04 12:40:16.371
Data Uscita	2018-12-04 12:40:16.602
Bytes Ingresso	225 B
Bytes Uscita	225 B
Contenuti Uscita	Visualizza Esporta

Figura 57: Dettaglio della transazione con contenuti

Headers	
Nome	Valore
GovWay-Provider	Ente
GovWay-Token-Expire	2018-12-04_13:16:15.000
GovWay-Service-Type	gw
GovWay-Token-Scopes	https://www.googleapis.com/auth/plus.me
GovWay-Token-ClientId	407408718192.apps.googleusercontent.com
GovWay-Token-Subject	106235657592654397689
accept	application/json
User-Agent	GovWay
GovWay-Message-ID	6f6c1374-8744-4345-81ba-534ca8ca0793
GovWay-Service	PetStore
GovWay-Token-ProcessTime	2018-12-04_12:40:16.582
GovWay-Token-Audience	407408718192.apps.googleusercontent.com
GovWay-Action	PUT_pet
GovWay-Provider-Type	gw
GovWay-Transaction-ID	9319b9d7-0458-4599-84e1-09a583d0bcd4
GovWay-Service-Version	2

Figura 58: Header HTTP prodotti da GovWay contenenti le informazioni sul Token

5.1.2 Validazione JWT

In questa sezione viene descritto uno scenario in cui GovWay non interagisce con un servizio di Introspection per validare l'access token ricevuto ma lo valida direttamente secondo la specifica JWS ([RFC 7515](#)).

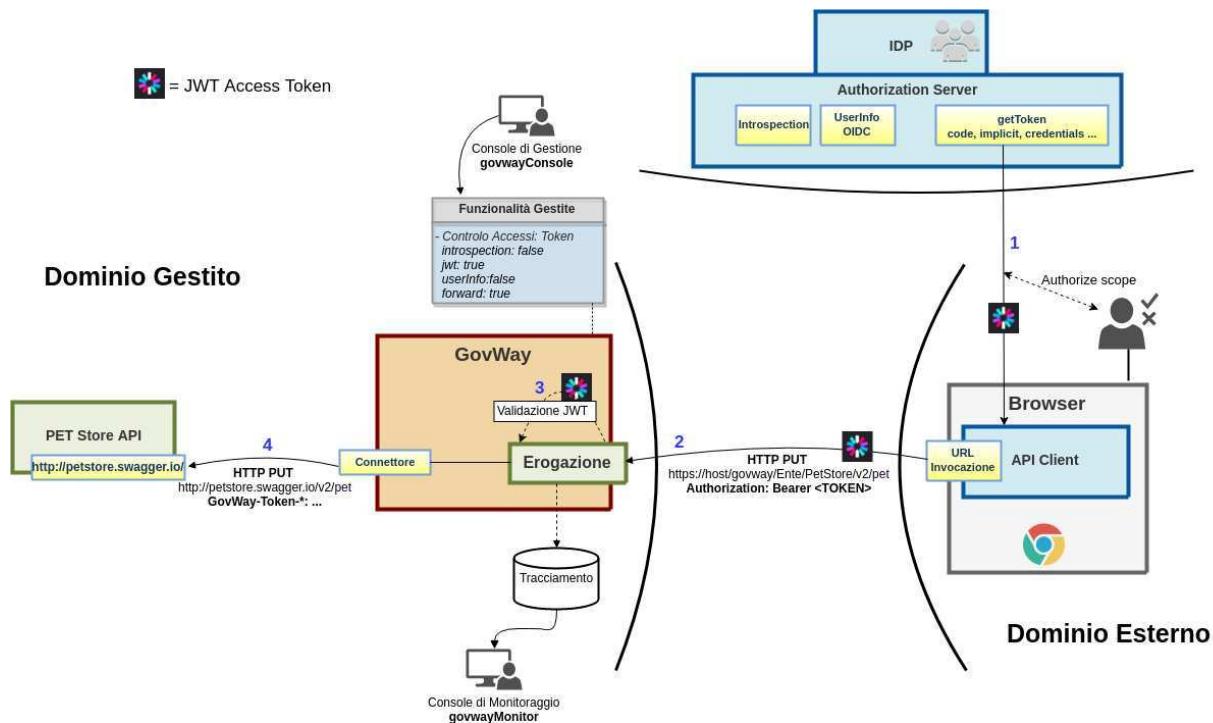


Figura 59: Scenario OAuth con validazione JWT

Per simulare lo scenario utilizzeremo sempre il servizio *Playground* e l'*Authorization Server di Google* descritto nella precedente sezione Sezione 5.1.1 utilizzando però impropriamente come *access token* l'*'id token'* ottenuto insieme all'*access token*. L'*id token* contiene le informazioni sull'*utente* strutturate all'interno di un *JWT* (per ulteriori dettagli si rimanda [OIDC Connect - IDToken](#)).

Utilizzo improprio dell'*id token*

L'utilizzo dell'*'id token'* come *access token* è da considerarsi solo a titolo di esempio per mostrare la funzionalità di validazione di un token *JWT* disponibile su GovWay che potrebbe essere utilizzata negli scenari reali quando effettivamente l'*access token* non è opaco ma possieda una struttura *JWT*.

• Configurazione Controllo degli Accessi

Accedere alla sezione '*Erogazioni*' e selezionare l'API precedentemente registrata '*PetStore v2*'. Dopodichè accedere, dal dettaglio dell'erogazione, alla sezione '*Configurazione*' dove vengono visualizzate le funzionalità attive. Cliccare sulla voce presente nella colonna '*Controllo Accessi*' e procedere con la seguente configurazione all'interno della sezione '*Gestione Token*':

- *Stato*: abilitato
- *Policy*: Google
- *Validazione JWT*: abilitato
- *Introspection*: disabilitato
- *User Info*: disabilitato
- *Token Forward*: abilitato

Effettuata la configurazione salvarla cliccando sul pulsante '*Salva*'.

The screenshot shows the 'Controllo Accessi' configuration interface for a PetStore application. The 'Validazione JWT' section is active, with the 'Stato' dropdown set to 'abilitato'. Other settings include 'Policy' set to 'Google', and various optional token-related options like Introspection and User Info disabled.

Figura 60: Configurazione OAuth2 - Validazione JWT

• Acquisizione Access Token

Per simulare l'acquisizione di un token in formato JWT utilizzare l'applicazione *Playground* come descritto nella precedente sezione Sezione 5.1.1. In fondo alla procedura, dopo aver cliccato sul pulsante '*Exchange authorization code for tokens*', estrarre dalla risposta http visualizzata sulla destra dell'applicazione l'*id token*.

```

Request / Response
POST /oauth2/v4/token HTTP/1.1
Host: www.googleapis.com
Content-length: 277
Content-type: application/x-www-form-urlencoded
User-Agent: google-oauth-playground
code=4%FqQ0CtBFRJConlp2VmBckP4w0JqeAgPj56QlAuiKyn4Dz4dY9epF17nfln-pxgyx0UkXlhxp_SC7rc0dqpbZnE8&redirect_uri

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1007
Date: Tue, 06 Dec 2016 18:57:27 GMT
Server: ESF
X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Vary: Origin, X-Origin, Referer
Cache-Control: private
Alt-Svc: quic=:443; ma=2592000; v="44,43,39,35"
Content-Type: application/json; charset=UTF-8

{
  "access_token": "ya29_GIteBufj3c390CX58k_oa3a70zGh29RGHqUhZ0Se3IU46gp0TdbRnJLB83Yg027RYGYm9F7c1bN3rNb1r8BhB8
  "id_token": "eyJhbGciOiJSUzI1NiIiImtpZC16Ij02M2Z1NDgwY2NjNTgz0Wj1YE100YxZTA4YzMzDE4N22hZjhNTYiLCJ0eXAiO
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope": "https://www.googleapis.com/auth/plus.me",
  "refresh_token": "1/d5PhXloSya0mV-Nw5glMfn82pESGqU3u0f7_ULYR0"
}

```

Figura 61: Ottenimento Token: Playground Google, Step 3

• Invocazione API con un access token

Con il seguente comando è possibile effettuare una richiesta che possiede l'*id token* ottenuto nella precedente fase.

Bearer Token Usage

Un *access token* può essere incluso nella richiesta tramite una delle modalità definite dalla specifica [RFC 6750](#).

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token=ID_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento viene confermato con un codice http 200 e una risposta json equivalente alla richiesta:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Content-Type: application/json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Message-ID: 84e1d9a4-c181-436f-b7f0-4cabf55c370d
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
    "id":3,
    "category": {"id":22, "name": "dog" },
    "name": "doggie",
    "photoUrls": ["http://image/dog.jpg"],
    "tags": [{"id":23, "name": "white"}],
    "status": "available"
}
```

• Consultazione Tracce

Attraverso la console *govwayMonitor* è possibile adesso vedere che le richieste transitano con successo sul gateway. Accedendo al dettaglio di una transazione, tra le varie informazioni presenti nella sezione '*Informazioni Mittente*', sono presenti le informazioni principali estratte dal token (es. Subject presente nel claim 'sub').

Informazioni Mittente	
Metodo HTTP	PUT
URL Invocazione	[in] /govway/in/Ente/PetStore/v2/pet? access_token=y29.GitoBjCHXoKaglEFXI0UxsN1UVVUW1ryp..kt2laD8ERHY 1ZyE-Af2sMPrL-cOWzZx_R
Indirizzo Client	127.0.0.1
Codice Risposta Client	200
Token Info	
Client ID	407408718192.apps.googleusercontent.com
Subject	106235657592654397689
Token Info	Visualizza

Figura 62: Traccia di una invocazione terminata con successo

Cliccando sul link '*Visualizza*' della voce '*Token Info*' è possibile vedere tutti i claims presenti nel token, tra cui è possibile constatare la presenza del claim *scope* valorizzato con quanto richiesto tramite l'applicazione Playground.

```

1  {
2   "valid" : true,
3   "iss" : "https://accounts.google.com",
4   "sub" : "106235657592654397689",
5   "aud" : [ "407408718192.apps.googleusercontent.com" ],
6   "exp" : 1544001967000,
7   "iat" : 1543998367000,
8   "clientId" : "407408718192.apps.googleusercontent.com",
9   "userInfo" : { },
10  "claims" : {
11    "at_hash" : "7E98p4nvea6Ly2nsXrs0CQ",
12    "aud" : "407408718192.apps.googleusercontent.com",
13    "sub" : "106235657592654397689",
14    "azp" : "407408718192.apps.googleusercontent.com",
15    "iss" : "https://accounts.google.com",
16    "exp" : "1544001967",
17    "iat" : "1543998367"
18  },
19  "rawResponse" : "
20  {"iss":"https://accounts.google.com","azp":"407408718192.apps.googleusercontent.com","sub":"106235657592654397689","aud":["407408718192.apps.googleusercontent.com"],"exp":1544001967,"iat":1543998367,"azp":407408718192,"at_hash":"7E98p4nvea6Ly2nsXrs0CQ","claims":{"sub":106235657592654397689,"iss":https://accounts.google.com,"aud":407408718192,"exp":1544001967,"iat":1543998367,"azp":407408718192,"azp":407408718192,"at_hash":7E98p4nvea6Ly2nsXrs0CQ}}"
21 }

```

Figura 63: Informazioni presenti in un Token JWT

5.1.3 Autenticazione e OIDC UserInfo

Nelle precedenti sezioni è stato mostrato come proteggere un'api in modo che ogni richiesta debba possedere un *access token* valido rilasciato da un *Authorization Server* censito su GovWay, nell'esempio Google. La verifica di un *access token*, se opposto tramite il servizio di Introspection (descritto nella sezione Sezione 5.1.1), altrimenti tramite la validazione JWT (sezione Sezione 5.1.2) permette a GovWay di conoscere i claims associati al token come ad esempio il subject ('sub'), l'issuer ('iss') etc e salvarli nella traccia come è stato mostrato nelle figura 55 e Figura 63.

GovWay può essere configurato per verificare che un *access token* presente al suo interno alcuni claims che identificano i seguenti attori principali nello scenario OAuth:

- *Issuer* (claim 'iss'): identifica l'Authorization Server che ha generato il token (es. <https://accounts.google.com>).
- *ClientId* (claim 'client_id' o 'azp'): rappresenta l'applicazione, censita sull'Authorization Server, a cui è stato rilasciato il token (es. client Playground).
- *Subject* (claim 'sub'): identifica l'utente, censito sull'Authorization Server (o IDP associato), che ha confermato le informazioni richiesti dall'applicazione e presenti nel token. Il Subject è presente se il rilascio di un token viene effettuato tramite dei flussi che prevedono l'interazione con l'utente il quale dovrà autenticarsi ed eventualmente autorizzare gli scope richiesti dall'applicazione. Il Subject è una informazione codificata (stringa o URI) che identifica univocamente l'utente nel dominio dell'Authorization Server (Issuer).
- *Username* (claim 'username', 'preferred_username' o 'name'): fornisce una rappresentazione 'human-readable' dell'utente.
- *eMail* (claim 'email'): identifica l'indirizzo e-mail dell'utente.

Se viene abilitato un controllo e GovWay non rileva il claim dopo la verifica dell'access token, la transazione termina con errore.

Le informazioni riguardanti l'*Username* e l'*eMail* potrebbero non essere disponibili dopo la semplice validazione dell'access token (sia introspection che jwt), e per ottenerle potrebbe essere necessario richiedere maggiori informazioni sull'utente tramite il servizio *OIDC UserInfo* dell'*Authorization Server*. Per maggiori informazioni a riguardo si rimanda alla specifica [OIDC Connect - UserInfo](#).

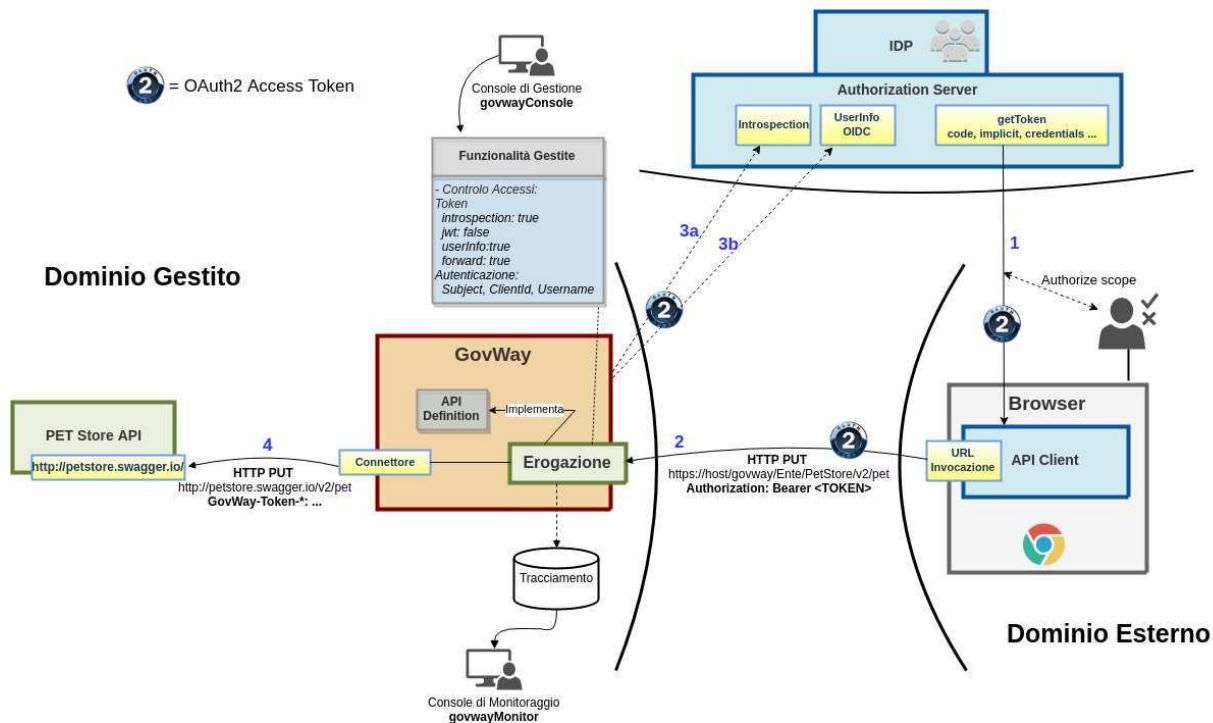


Figura 64: Scenario OAuth con accesso servizio UserInfo

Per simulare lo scenario utilizzeremo sempre il servizio *Playground* e l'*Authorization Server di Google* descritto nella precedente sezione Sezione 5.1.1. Faremo un primo test in cui il Gateway non accede al servizio *User Info* e vedremo come non è disponibile l'informazione sull'utente sotto forma 'human-readable' che invece verrà recuperata abilitando l'interazione con tale servizio.

• Configurazione Controllo degli Accessi

Accedere alla sezione '*Erogazioni*' e selezionare l'API precedentemente registrata '*PetStore v2*'. Dopodichè accedere, dal dettaglio dell'erogazione, alla sezione '*Configurazione*' dove vengono visualizzate le funzionalità attive. Cliccare sulla voce presente nella colonna '*Controllo Accessi*' e procedere con la seguente configurazione all'interno della sezione '*Gestione Token*':

- *Stato*: abilitato
- *Policy*: Google
- *Validazione JWT*: disabilitato
- *Introspection*: abilitato
- *User Info*: disabilitato
- *Token Forward*: abilitato

Procedere inoltre con la seguente configurazione all'interno della sezione '*Autenticazione*':

- *Trasporto - Stato*: disabilitato
- *Token - Issuer*: disabilitato
- *Token - ClientId*: abilitato
- *Token - Subject*: abilitato
- *Token - Username*: abilitato
- *Token - eMail*: disabilitato

Effettuata la configurazione salvarla cliccando sul pulsante 'Salva'.

The screenshot shows the 'Controllo Accessi' configuration page. In the 'Gestione Token' section, the 'Policy' field is set to 'Google'. In the 'Autenticazione' section, the 'Trasporto' dropdown is set to 'disabilitato'. Under 'Token', the 'ClientId' and 'Subject' checkboxes are checked, while 'Issuer' and 'Username' are unchecked.

Figura 65: Configurazione OAuth2 - Autenticazione

- **Invocazione API**

Reset Cache delle Configurazioni prima di un nuovo test

Le configurazioni accedute da GovWay vengono mantenute in una cache dopo il primo accesso per 2 ore, è quindi necessario forzare un reset della cache. Per farlo accedere alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Per effettuare il test acquisire un token utilizzando l'applicazione *Playground* come descritto nella precedente sezione Sezione 5.1.1 e procedere con il seguente comando.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token= ←
    ACCESS_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento termina con un codice di errore http 401 e una risposta problem+json che riporta la motivazione:

```
HTTP/1.1 401 Unauthorized
```

```

WWW-Authenticate: Bearer realm="Google", error="insufficient_scope", error_description=" ←
    The request requires higher privileges than provided by the access token"
Content-Type: application/problem+json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
  "type": "https://httpstatuses.com/401",
  "title": "Unauthorized",
  "status": 401,
  "detail": "La richiesta presenta un token non sufficiente per fruire del servizio ←
    richiesto",
  "govway_status": "protocol:GOVWAY-1368"
}

```

• Consultazione Tracce in errore

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Dalla figura Figura 66 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con errore con esito *Autenticazione Fallita*.

		Data Ingresso Richiesta	Tipologia	Esito	Fruitore	Erogatore	API	Azione
<input type="checkbox"/>	●	2018-12-05 15:31:42	Erogazione	Autenticazione Fallita	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	●	2018-12-05 15:31:42	Erogazione	Autenticazione Fallita	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	●	2018-12-05 15:31:41	Erogazione	Autenticazione Fallita	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	●	2018-12-05 15:29:46	Erogazione	Autenticazione Fallita	Ente	PetStore v2	PUT_pet	

Figura 66: Tracce delle invocazioni terminate con errore 'Autenticazione Fallita'

Accedendo al dettaglio di una transazione terminata in errore, e visualizzandone i diagnostici è possibile comprendere l'errore che come atteso risulta essere la mancanza dell'informazione *Username* richiesta obbligatoriamente tramite la sezione '*Autenticazione*' precedentemente configurata

Storico > Intervallo Temporale > Dettagli Transazione > Messaggi Diagnostici				
Lista Diagnostici: record [1 - 7] su 7				
Data	Severità	Funzione	Messaggio	
2018-12-05 15:31:42.875	infoIntegration	RicezioneBuste	Ricevuta richiesta applicativa	
2018-12-05 15:31:42.878	infoIntegration	RicezioneBuste	Gestione Token [Google] (Validazione Introspection) in corso ...	
2018-12-05 15:31:42.879	infoIntegration	RicezioneBuste	Gestione Token [Google] (Validazione Introspection) completata con successo	
2018-12-05 15:31:42.879	infoIntegration	RicezioneBuste	Autenticazione token (ClientId,Subject,Username) in corso ...	
2018-12-05 15:31:42.879	errorIntegration	RicezioneBuste	Autenticazione token (ClientId,Subject,Username) fallita: Token without username claim	
2018-12-05 15:31:42.881	errorProtocol	RicezioneBuste	Generato messaggio di cooperazione di Errore con identificativo [b6fbdd4-051a-4a3f-97da-18c7f0dd9755]	
2018-12-05 15:31:42.884	infoIntegration	RicezioneBuste	Risposta ({"type": "https://httpstatuses.com/401", "title": "Unauthorized", "status": 401, "detail": "La richiesta presenta un token non sufficiente per fruire del servizio richiesto", "govway_status": "protocol:GOVWAY-1368"}) consegnata al mittente con codice di trasporto: 401	

Figura 67: Diagnostici di una invocazione terminata con errore

Cliccando sul link '*Visualizza*' della voce '*Token Info*' è possibile comunque vedere tutti i claims presenti nel token, dove si denota come non sia presente uno dei claim che rappresenta l'informazione '*Username*'.

The screenshot shows a JSON representation of a JWT token. The token is valid and issued by '407408718192.apps.googleusercontent.com'. It has an audience of '407408718192.apps.googleusercontent.com', an expiration date of 1544623764, and a client ID of '407408718192.apps.googleusercontent.com'. The scopes are 'https://www.googleapis.com/auth/plus.me'. The user info is present. The claims include 'azp' (407408718192.apps.googleusercontent.com), 'sub' (106235657592654397689), and 'aud' (407408718192.apps.googleusercontent.com). The raw response is also shown.

```

1 { "valid": true,
2   "sub": "106235657592654397689",
3   "aud": ["407408718192.apps.googleusercontent.com"],
4   "exp": 1544623764,
5   "client_id": "407408718192.apps.googleusercontent.com",
6   "scopes": ["https://www.googleapis.com/auth/plus.me"],
7   "user_info": {},
8   "claims": {
9     "azp": "407408718192.apps.googleusercontent.com",
10    "sub": "106235657592654397689",
11    "aud": "407408718192.apps.googleusercontent.com",
12    "access_type": "offline",
13    "azp": "407408718192.apps.googleusercontent.com",
14    "scope": "https://www.googleapis.com/auth/plus.me",
15    "exp": 1544623764,
16    "expires_in": "3578"
17  },
18  "raw_response": "(\n  \"azp\": \"407408718192.apps.googleusercontent.com\",\n  \"aud\":\n    \"407408718192.apps.googleusercontent.com\",\n  \"sub\": \"106235657592654397689\",\n  \"exp\": 1544623764,\n  \"scope\": \"https://www.googleapis.com/auth/plus.me\"\n)\n"
19 }
  
```

DOWNLOAD

Figura 68: Informazioni presenti nel Token

- **Abilitazione UserInfo in Configurazione Controllo degli Accessi**

Tramite la *govwayConsole* accedere nuovamente alla maschera di configurazione '*Controllo Accessi*' dell'API '*PetStore v2*' ed abilitare stavolta anche il servizio '*User Info*'.

The screenshot shows the 'Controllo Accessi' configuration page. Under 'Gestione Token', the 'User Info' setting is highlighted with a red border. Under 'Autenticazione', the 'Trasporto' section shows 'Issuer' and 'Clientid' selected. The 'Token' section shows 'Subject', 'Username', and 'eMail' selected.

Figura 69: Configurazione OAuth2 - Autenticazione

- **Nuova invocazione API**

Reset Cache delle Configurazioni prima di un nuovo test

Effettuare il reset della cache accedendo alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Per effettuare il test acquisire un token utilizzando l'applicazione *Playground* come descritto nella precedente sezione Sezione 5.1.1 e procedere con il seguente comando.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token=←
    ACCESS_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
        "id": 3,
        "category": { "id": 22, "name": "dog" },
        "name": "doggie",
        "photoUrls": [ "http://image/dog.jpg" ],
        "tags": [ { "id": 23, "name": "white" } ],
        "status": "available"
}'
```

L'esito dell'aggiornamento termina stavolta con successo con un codice http 200 e una risposta json equivalente alla richiesta.

• Consultazione Tracce

Attraverso la console *govwayMonitor* è possibile adesso vedere che le richieste transitano con successo sul gateway. Accedendo al dettaglio di una transazione, tra le varie informazioni presenti nella sezione '*Informazioni Mittente*', sono presenti tutte e tre le informazioni principali attese: ClientId, Subject e Username.

Informazioni Mittente	
Metodo HTTP	PUT
URL Invocazione	[in] /govway/in/Ente/PetStore/v2/pet access_token=ya29.GltBpHYHBNDdkRNNP1_fedzujBFaE5Jr39tukpYdzhvne9g97sAeoFAUeJA6QOMX2IovSYDa5JCz2VLH5qkI0cD2SGw5rfzmlvRED3Ej0vJxe7wRBfRhGojWS
Indirizzo Client	127.0.0.1
Codice Risposta Client	200
Token Info	
Client ID	407408718192.apps.googleusercontent.com
Subject	106235657592654397689
Username	Andrea Poli
Token Info	Visualizza

Figura 70: Traccia di una invocazione terminata con successo

Cliccando sul link '*Visualizza*' della voce '*Token Info*' è possibile vedere tutti i claims presenti nel token, tra cui è possibile constatare la presenza dei claims estratti grazie all'invocazione del servizio '*User Info*'.

```
{
  "valid": true,
  "sub": "106235657592654397689",
  "username": "Andrea Poli",
  "aud": "407408718192.apps.googleusercontent.com",
  "exp": 1544023764688,
  "client_id": "407408718192.apps.googleusercontent.com",
  "scopes": ["https://www.googleapis.com/auth/plus.me"],
  "user_info": {
    "full_name": "Andrea Poli",
    "first_name": "Andrea",
    "family_name": "Poli"
  },
  "claims": {
    "aud": "407408718192.apps.googleusercontent.com",
    "sub": "106235657592654397689",
    "access_type": "offline",
    "azp": "407408718192.apps.googleusercontent.com",
    "scope": "https://www.googleapis.com/auth/plus.me",
    "profile": "https://www.googleapis.com/oauth2/v1/userinfo",
    "name": "Andrea Poli",
    "exp": 1544023764,
    "given_name": "Andrea"
  }
}
```

Figura 71: Informazioni presenti in un Token JWT

5.1.4 Autorizzazione per Scope

La verifica di un *access token*, se opaco tramite il servizio di Introspection (descritto nella sezione Sezione 5.1.1), altrimenti tramite la validazione JWT (sezione Sezione 5.1.2), permette a GovWay di conoscere i claims associati al token ed in particolare quali sono gli scope autorizzati dall'utente.

Gli scope permettono di definire delle "funzioni applicative" il cui utilizzo da parte di un'applicazione deve essere autorizzato da un utente.

Per simulare lo scenario utilizzeremo sempre il servizio *Playground* e l'*Authorization Server di Google* descritto nella precedente sezione Sezione 5.1.1 dove però verranno richiesti altri scope rispetto a quello utilizzato nel precedente scenario. Simuleremo di aver bisogno di accedere alle API Calendar di Google e quindi dovremo richiedere tali scope che devono essere autorizzati una volta che ci siamo autenticati su Google.

Su GovWay è possibile registrare gli scope disponibili su di un *Authorization Server* ed utilizzarli per definire politiche di autorizzazione rispetto agli scope presenti nell'*access token*. Lo scenario descritto in questa sezione mostra un esempio di registrazione degli scope '*API Calendar*' di Google dove si configura a titolo esemplificativo che tali scope sono necessari per poter invocare il servizio *PetStore*.

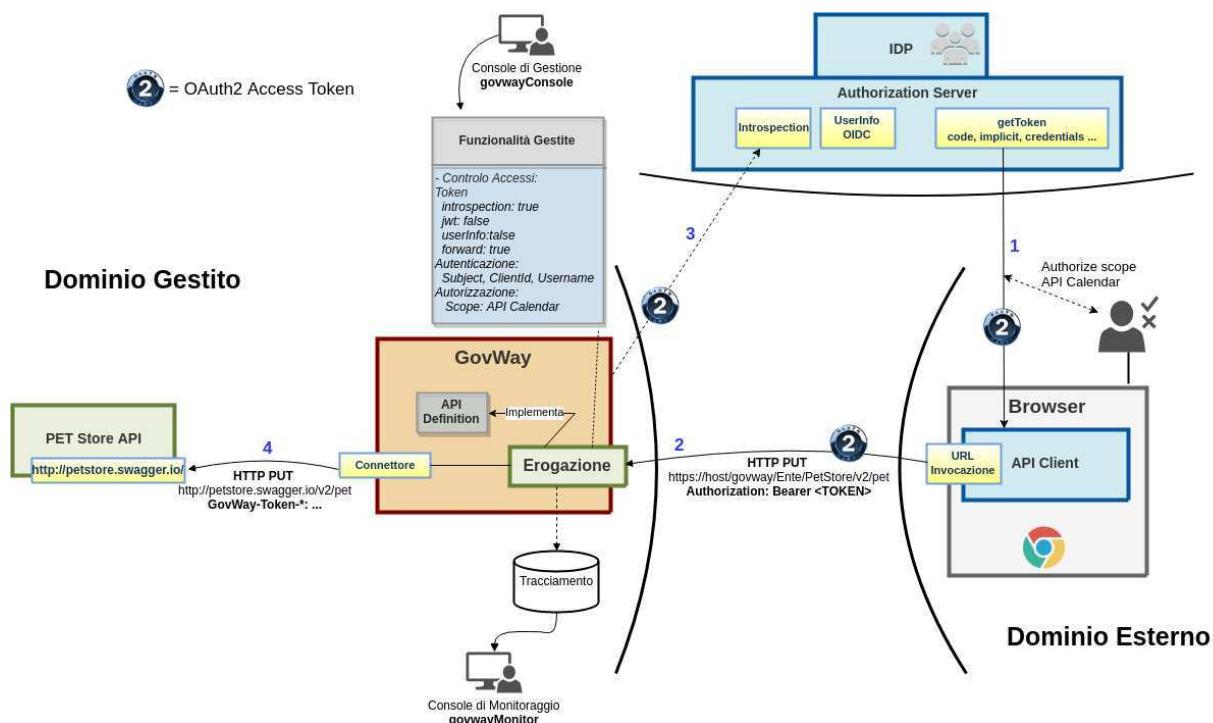


Figura 72: Scenario OAuth con autorizzazione per Scope

• Acquisizione Access Token con scope API Calendar

Per simulare l'acquisizione di un token è possibile utilizzare l'applicazione *Playground*, disponibile all'indirizzo <https://developers.google.com/oauthplayground/>, che consente di richiedere un *access token* all'*Authorization Server di Google*.

L'applicazione *Playground* consente agevolmente di ottenere l'*access token* con gli scope richiesti dall'esempio:

1. Selezione scope

Devono essere selezionati gli *scope*:

- <https://www.googleapis.com/auth/calendar.events.readonly>
- <https://www.googleapis.com/auth/calendar.readonly>
- <https://www.googleapis.com/auth/calendar.settings.readonly>

Cliccando infine sul pulsante '*Authorize APIs*' si verrà rediretti alla pagina di autenticazione in google dove si dovrà procedere ad autenticarsi.

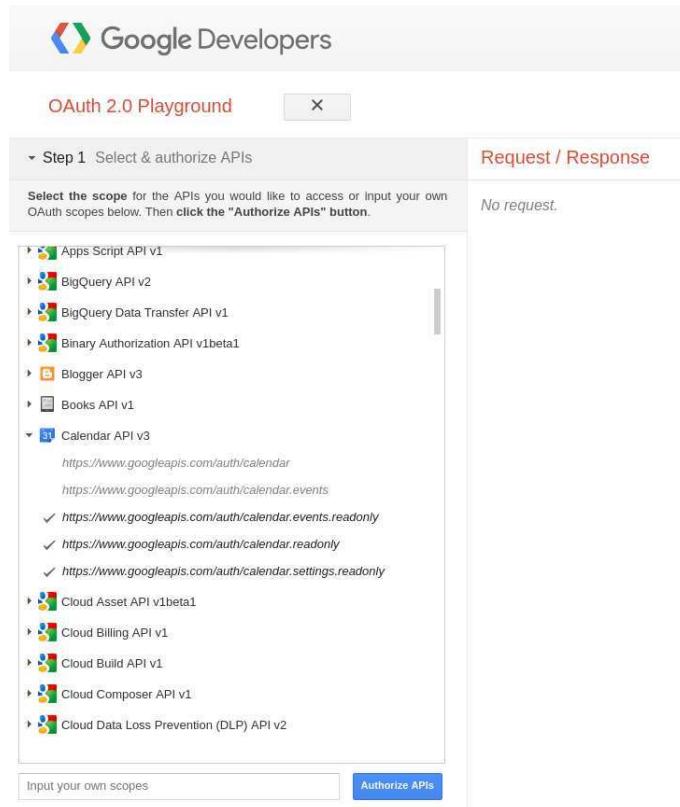


Figura 73: Ottenimento Token: Playground Google, scelta scope API Calendar

2. Autorizzazione scope API Calendar

Effettuata l'autenticazione in Google si viene rediretti ad una pagina dove è richiesto all'utente di autorizzare l'applicazione Playgroud all'utilizzo degli scope API Calendar.

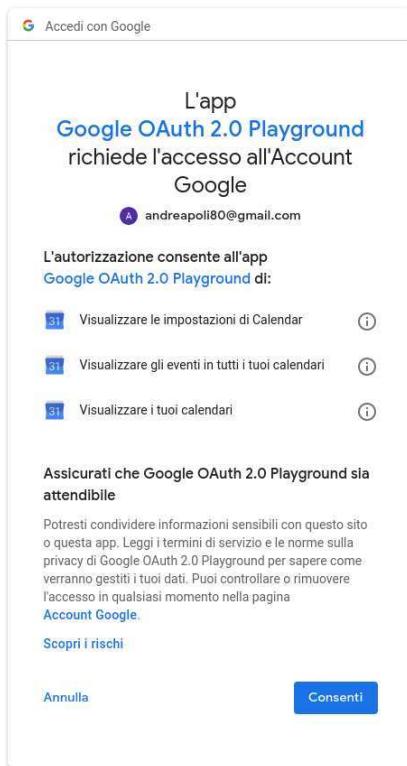


Figura 74: Ottenimento Token: Playground Google, autorizzazione scope API Calendar

3. Access Token

Autorizzati gli scope si viene rediretti alla seconda fase prevista dall'applicazione *Playground* denominata '*Exchange authorization code for tokens*'. Cliccando sul pulsante '*Exchange authorization code for tokens*' si ottiene infine un *access token* da estrarre nella risposta http visualizzata sulla destra dell'applicazione.

```
Request / Response

POST /oauth2/v4/token HTTP/1.1
Host: www.googleapis.com
Content-length: 277
Content-type: application/x-www-form-urlencoded
User-Agent: google-auth-playground
Code=4%2FqgBqFLXZ1DE5EU9Lb01oA3xr1fo4HgNCp1ozTtw5h7a_0f2mq05PNEkafedo5GpYX6mW5PH0btA103CY9ih0&redirect_uri=https%3A%2F%2Fwww.google.com%2Fauth%2Ftoken%2Fcallback

HTTP/1.1 200 OK
Content-length: 449
X-xss-protection: 1; mode=block
X-content-type-options: nosniff
Transfer-Encoding: chunked
Vary: Origin, X-Origin, Referer
Server: ESF
Content-Encoding: gzip
Cache-control: private
Date: Wed, 05 Dec 2018 15:32:41 GMT
X-Content-Type-Options: nosniff
Alt-Svc: quic=":443"; ma=2592000; v="44,43,39,35"
Content-type: application/json; charset=utf-8

{
  "access_token": "ya29.GltPBkjy5djr9V6z4nPglOPfnUjhEasRTm10een04rBAke0u2Gj4PiD-bG1S-f2mTR8P80LR3lMhqJDZue1Sl4Re_G0A",
  "scope": "https://www.googleapis.com/auth/calendar.events.readonly https://www.googleapis.com/auth/calendar.readonly",
  "token_type": "Bearer",
  "expires_in": 3600,
  "refresh_token": "1/MdYRACdcEl6auXXbwq0B3niHarKVFEPv9mnncfIZEmQsB"
}
```

Figura 75: Ottenimento Token: Playground Google, Step 3

• Registrazione degli scope su GovWay

Accedere alla sezione '*Scope*' della *govwayConsole* per registrare gli scope relativi ad *API Calendar*. Per registrare un nuovo scope cliccare sul pulsante '*Aggiungi*'. Effettuare la registrazione degli scopes richiesti precedentemente tramite *Playground* ed anche un ulteriore scope (*API Google Driver*), non richiesto durante l'acquisizione del token, che verrà utilizzato nei test descritti in questa sezione.

Nome	Identificativo Esterno	Contesto
google.calendar.events.readonly	https://www.googleapis.com/auth/calendar.events.readonly	Qualsiasi
google.calendar.readonly	https://www.googleapis.com/auth/calendar.readonly	Qualsiasi
google.calendar.settings.readonly	https://www.googleapis.com/auth/calendar.settings.readonly	Qualsiasi
google.drive	https://www.googleapis.com/auth/drive	Qualsiasi

Tabella 1: Registrazione scope

The screenshot shows a web-based configuration interface for adding a new OAuth scope. The form has a header 'Scope > Aggiungi'. Below it, there's a note: 'Note: (*) Campi obbligatori'. The main section is titled 'Scope' and contains the following fields:

- Nome: google.calendar.events.readonly (marked with a red asterisk)
- Descrizione: (empty)
- Identificativo Esterno: https://www.googleapis.com/auth/calendar.events.readonly
- Contesto: Qualsiasi

At the bottom right is a large 'SALVA' button.

Figura 76: Configurazione OAuth2 - Registrazione Scope

Terminata la registrazione gli scope è possibile specificarli all'interno del Controllo degli Accessi di una API.

The screenshot shows a list of registered OAuth scopes. The title bar says 'Scope'. The list area shows the following entries:

	Nome	Contesto
google.calendar.events.readonly	Qualsiasi	
google.calendar.readonly	Qualsiasi	
google.calendar.settings.readonly	Qualsiasi	
google.drive	Qualsiasi	

At the bottom right are 'ELIMINA' and 'AGGIUNGI' buttons.

Figura 77: Configurazione OAuth2 - Lista degli Scope registrati

• Configurazione Controllo degli Accessi

Accedere alla sezione '*Erogazioni*' e selezionare l'API precedentemente registrata '*PetStore v2*'. Dopodichè accedere, dal dettaglio dell'erogazione, alla sezione '*Configurazione*' dove vengono visualizzate le funzionalità attive. Cliccare sulla voce presente nella colonna '*Controllo Accessi*' e procedere con la seguente configurazione all'interno della sezione '*Gestione Token*':

- *Stato*: abilitato
- *Policy*: Google
- *Validazione JWT*: disabilitato
- *Introspection*: abilitato
- *User Info*: disabilitato
- *Token Forward*: abilitato

Procedere inoltre con la seguente configurazione all'interno della sezione '*Autorizzazione*':

- *Autorizzazione - Stato*: abilitato
- *Autorizzazione per Scope - Stato*: abilitato
- *Autorizzazione per Scope - Scope Richiesti*: tutti

Effettuata la configurazione salvarla cliccando sul pulsante 'Salva'.

The screenshot shows the 'Controllo Accessi' configuration page for the PetStore v2 (Ente) service. The 'Note: (*) Campi obbligatori' section is visible at the top. The 'Gestione Token' section contains fields for 'Stato' (abilitato), 'Policy' (Google), and optional token validation (Validazione JWT, Introspection, User Info, Token Forward). The 'Autenticazione' section includes 'Trasporto' (disabilitato) and 'Token' fields for Issuer, ClientId, Subject, Username, and eMail. The 'Autorizzazione' section has 'Stato' (abilitato), 'Abilitato' (checkbox checked), 'Scope Richiesti' (tutti selected), and 'Abilitato' (checkbox checked) for Token Claims. A 'SALVA' button is at the bottom.

Figura 78: Configurazione OAuth2 - Autorizzazione

Salvata la configurazione si deve nuovamente accedere al *Controllo Accessi* dove nella sezione '*Autorizzazione*' è adesso disponibile un link '*Scope (0)*' che permette di registrare gli scope che un token deve possedere quando invoca l'api PetStore.

The screenshot shows the 'Autorizzazione' (Authorization) section of the OAuth2 configuration. It includes fields for 'Stato' (Status), 'Abilitato' (Enabled), and dropdown menus for 'Scope Richiesti' (Requested Scopes) and 'Scope (U)' (selected 'tutti').

Figura 79: Configurazione OAuth2 - Autorizzazione - Scope

Tramite il pulsante '*Aggiungi*' aggiungere tutti e 4 gli scope precedentemente registrati su GovWay.

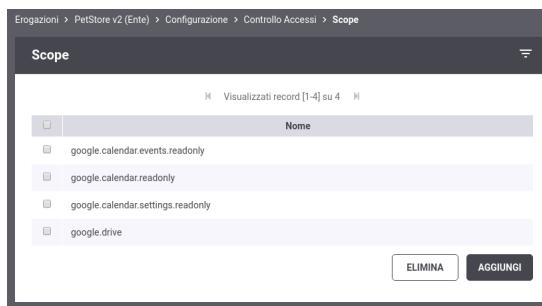


Figura 80: Configurazione OAuth2 - Autorizzazione - Elenco Scope

• Invocazione API

Reset Cache delle Configurazioni prima di un nuovo test

Le configurazioni accedute da GovWay vengono mantenute in una cache dopo il primo accesso per 2 ore, è quindi necessario forzare un reset della cache. Per farlo accedere alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Per effettuare il test utilizzare il token, contenente gli scope API Calendar, precedentemente ottenuto.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token= ←
    ACCESS_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
        "id": 3,
        "category": { "id": 22, "name": "dog" },
        "name": "doggie",
        "photoUrls": [ "http://image/dog.jpg" ],
        "tags": [ { "id": 23, "name": "white" } ],
        "status": "available"
}'
```

L'esito dell'aggiornamento termina con un codice di errore http 403 e una risposta problem+json che riporta la motivazione:

```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer realm="Google", error="insufficient_scope", error_description=" ←
The request requires higher privileges than provided by the access token", scope=" ←
https://www.googleapis.com/auth/calendar.events.readonly,https://www.googleapis.com/ ←
auth/calendar.readonly,https://www.googleapis.com/auth/calendar.settings.readonly, ←
https://www.googleapis.com/auth/drive"
```

```

Content-Type: application/problem+json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
  "type": "https://httpstatuses.com/403",
  "title": "Forbidden",
  "status": 403,
  "detail": "La richiesta presenta un token non sufficiente per fruire del servizio richiesto",
  "govway_status": "protocol:GOVWAY-1368"
}

```

• Consultazione Tracce in errore

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Dalla figura Figura 81 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con errore con esito *Autorizzazione Negata*.

	Data Ingresso Richiesta	Tipologia	Esito	Fruitore	Erogatore	API	Azione
<input type="checkbox"/>	2018-12-05 17:20:12	Erogazione	Autorizzazione Negata	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-05 17:20:11	Erogazione	Autorizzazione Negata	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-05 17:20:11	Erogazione	Autorizzazione Negata	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-05 17:16:45	Erogazione	Autorizzazione Negata	Ente	PetStore v2	PUT_pet	

Figura 81: Tracce delle invocazioni terminate con errore 'Autorizzazione Negata'

Accedendo al dettaglio di una transazione terminata in errore, e visualizzandone i diagnostici è possibile comprendere l'errore che come atteso risulta essere la mancanza dello scope *https://www.googleapis.com/auth/drive* richiesto poichè nella sezione 'Autorizzazione' è stato indicato che gli scope registrati devono essere tutti presenti nell'access token.

Storico > Intervallo Temporale > Dettagli Transazione > Messaggi Diagnostici			
Lista Diagnostici: record [1 - 8] su 8			
Data	Severità	Funzione	Messaggio
2018-12-05 17:20:12.259	infoIntegration	RicezioneBuste	Ricevuta richiesta applicativa
2018-12-05 17:20:12.263	infoIntegration	RicezioneBuste	Gestione Token [Google] (Validazione Introspection) in corso ...
2018-12-05 17:20:12.264	infoIntegration	RicezioneBuste	Gestione Token [Google] (Validazione Introspection) completato con successo
2018-12-05 17:20:12.264	infoProtocol	RicezioneBuste	Ricevuto messaggio di cooperazione con identificativo [cc5a5bd7-2131-4c79-9028-d6b235bb0d084]
2018-12-05 17:20:12.264	infoIntegration	RicezioneBuste	Verifica autorizzazione [token] messaggio con identificativo [cc5a5bd7-2131-4c79-9028-d6b235bb0d084] servizio [gw/Ente/gw/PetStore:2.PUT_pet] in corso ...
2018-12-05 17:20:12.265	errorIntegration	RicezioneBuste	Verifica autorizzazione [token] messaggio con identificativo [cc5a5bd7-2131-4c79-9028-d6b235bb0d084] servizio [gw/Ente/gw/PetStore:2.PUT_pet] fallita (codice: GOVWAY-1368) (Scope https://www.googleapis.com/auth/drive not found) La richiesta presenta un token non sufficiente per fruire del servizio richiesto
2018-12-05 17:20:12.266	errorProtocol	RicezioneBuste	Generato messaggio di cooperazione di Errore con identificativo [4d4fd07b-ab3c-4ad9-bf91-6459781b726b]
2018-12-05 17:20:12.270	infoIntegration	RicezioneBuste	Risposta (Type: "https://httpstatuses.com/403" title: "Forbidden", status: "403", detail: "La richiesta presenta un token non sufficiente per fruire del servizio richiesto", "govway_status": "protocol:GOVWAY-1368") consegnata al mittente con codice di trasporto: 403

Figura 82: Diagnostici di una invocazione terminata con errore

Cliccando sul link 'Visualizza' della voce 'Token Info' è possibile vedere tutti i claims presenti nel token, dove si possono vedere gli scope richiesti tramite Playground.

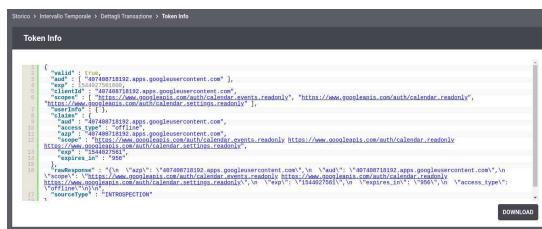


Figura 83: Scope presenti nel Token

- **Modifica controllo degli scope (Almeno uno) in Configurazione Controllo degli Accessi**

Tramite la *govwayConsole* accedere nuovamente alla maschera di configurazione '*Controllo Accessi*' dell'API '*PetStore v2*'; all'interno della sezione '*Autorizzare*' modificare il tipo di controllo '*Scope Richiesti*' dal valore '*tutti*' al valore '*almeno uno*'.

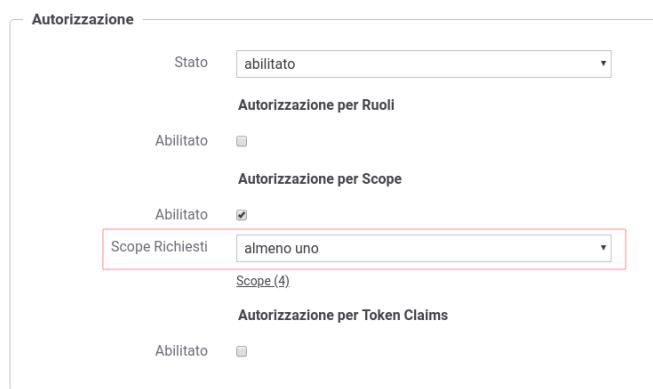


Figura 84: Configurazione OAuth2 - Autorizzazione degli scope con opzione 'Almeno uno'

- **Nuova invocazione API**

Reset Cache delle Configurazioni prima di un nuovo test

Effettuare il reset della cache accedendo alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Effettuare una nuova invocazione del test.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token=←
    ACCESS_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
        "id": 3,
        "category": { "id": 22, "name": "dog" },
        "name": "doggie",
        "photoUrls": [ "http://image/dog.jpg" ],
        "tags": [ { "id": 23, "name": "white" } ],
        "status": "available"
}'
```

L'esito dell'aggiornamento termina stavolta con successo con un codice http 200 e una risposta json equivalente alla richiesta.

5.1.5 Autorizzazione sui Claims

Oltre ad una autorizzazione sugli scope, descritta nello scenario Sezione [5.1.4](#), GovWay può essere configurato per verificare ulteriori claims ottenuti tramite la validazione dell'access token. La validazione che verrà descritta in questa sezione consiste in

una validazione semplice la cui logica si basa sulla semplice constatazione che uno o più claim siano stati riscontrati all'interno del token e possiedano il valore atteso. Per validazione più complesse si rimanda all'utilizzo di una policy XACML descritta nello scenario Sezione 5.1.6.

Per simulare lo scenario utilizzeremo sempre il servizio *Playground* e l'*Authorization Server* di *Google* descritto nella precedente sezione Sezione 5.1.1.

Verrà configurato GovWay al fine di effettuare le seguenti verifiche all'interno del token:

- *Audience* (claim 'aud'): contenga l'identificativo dell'applicazione *Playground* come destinatario del token
- *Applicazione Client* (claim 'azp'): controlleremo che il client appartenga ad uno delle applicazioni conosciute. Nell'elenco, inseriremo l'identificativo di *Playground* in modo da completare con successo la verifica.

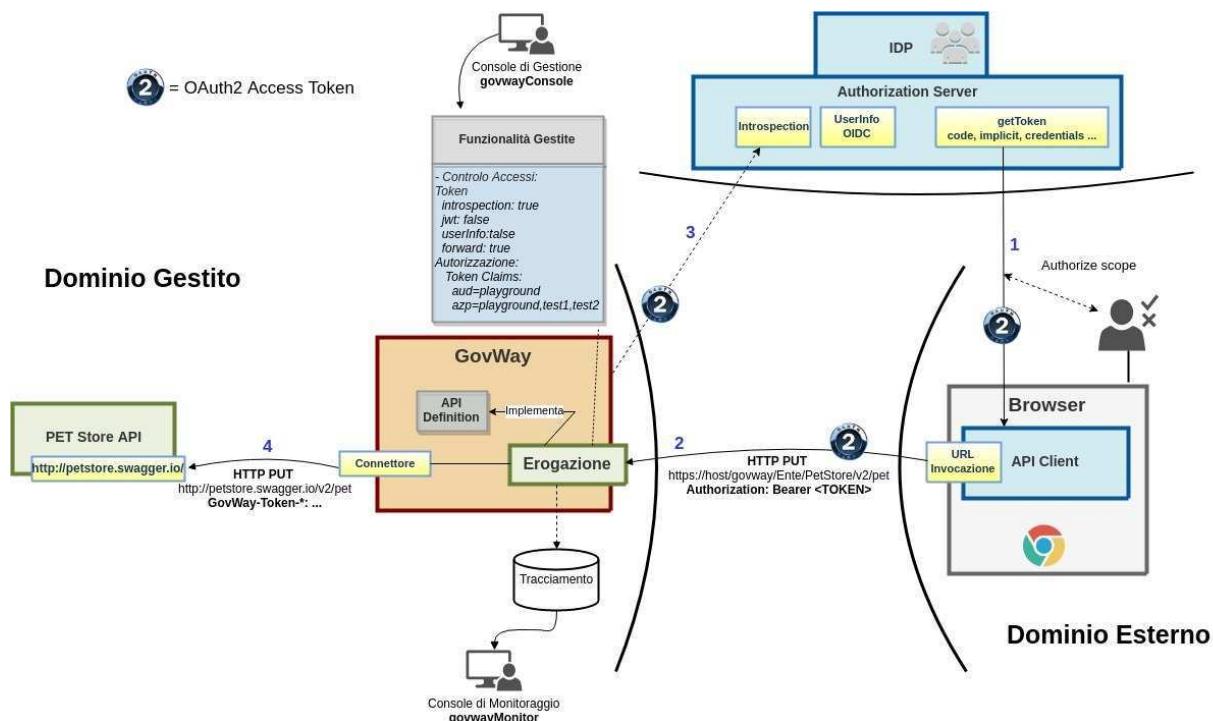


Figura 85: Scenario OAuth con autorizzazione sui Claims

• Configurazione Controllo degli Accessi

Accedere alla sezione '*Erogazioni*' e selezionare l'API precedentemente registrata '*PetStore v2*'. Dopodichè accedere, dal dettaglio dell'erogazione, alla sezione '*Configurazione*' dove vengono visualizzate le funzionalità attive. Cliccare sulla voce presente nella colonna '*Controllo Accessi*' e procedere con la seguente configurazione all'interno della sezione '*Gestione Token*':

- *Stato*: abilitato
- *Policy*: Google
- *Validazione JWT*: disabilitato
- *Introspection*: abilitato
- *User Info*: disabilitato
- *Token Forward*: abilitato

Procedere inoltre con la seguente configurazione all'interno della sezione '*Autorizzazione*':

- *Autorizzazione - Stato*: abilitato

- Autorizzazione per Token Claims - Stato: abilitato
- Claims, configuriamo l'identificativo dell'applicazione Playground come valore atteso per il claim 'aud', mentre forniamo una lista di valori tra i quali non è presente l'applicazione Playground per il claim 'azp':

Per conoscere l'identificativo dell'applicazione Playground

E' possibile vedere una precedente transazione terminata con successo per conoscere l'esatto valore associato all'applicazione *Playground* (es. Figura 55).

- * aud=407408718192.apps.googleusercontent.com
- * azp=client1, client2

Effettuata la configurazione salvarla cliccando sul pulsante 'Salva'.

The screenshot shows the 'Controllo Accessi' configuration page for OAuth2. In the 'Autorizzazione' section, under the 'Claims' configuration, the 'Value' field contains the string 'aud=407408718192.apps.googleusercontent.com' and 'azp=client1, client2'. This indicates that the application is requesting the 'aud' claim to have the value '407408718192.apps.googleusercontent.com' and the 'azp' claim to have the values 'client1' and 'client2'.

Figura 86: Configurazione OAuth2 - Autorizzazione

• Invocazione API

Reset Cache delle Configurazioni prima di un nuovo test

Le configurazioni accedute da GovWay vengono mantenute in una cache dopo il primo accesso per 2 ore, è quindi necessario forzare un reset della cache. Per farlo accedere alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Per effettuare il test utilizzare il token ottenuto come descritto nella sezione Sezione 5.1.1.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token=←
ACCESS_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
```

```

    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}

```

L'esito dell'aggiornamento termina con un codice di errore http 403 e una risposta problem+json che riporta la motivazione:

```

HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer realm="Google", error="insufficient_scope", error_description="←
The request requires higher privileges than provided by the access token"
Content-Type: application/problem+json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
  "type": "https://httpstatuses.com/403",
  "title": "Forbidden",
  "status": 403,
  "detail": "La richiesta presenta un token non sufficiente per fruire del servizio ←
             richiesto",
  "govway_status": "protocol:GOVWAY-1368"
}

```

• Consultazione Tracce in errore

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Dalla figura Figura 87 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con errore con esito *Autorizzazione Negata*.

Lista Transazioni: record [1 - 4]							
	Data Ingresso Richiesta	Tipologia	Esito	Fruitore	Erogatore	API	Azione
<input type="checkbox"/>	2018-12-05 17:20:12	Erogazione	Autorizzazione Negata	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-05 17:20:11	Erogazione	Autorizzazione Negata	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-05 17:20:11	Erogazione	Autorizzazione Negata	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-05 17:16:45	Erogazione	Autorizzazione Negata	Ente	PetStore v2	PUT_pet	

Figura 87: Tracce delle invocazioni terminate con errore 'Autorizzazione Negata'

Accedendo al dettaglio di una transazione terminata in errore, e visualizzandone i diagnostici è possibile comprendere l'errore che come atteso risulta essere un valore sconosciuto per quanto concerne il claim 'azp'.

Lista Diagnostici: record [1 - 8] su 8				
Data	Serività	Funzione	Messaggio	
2018-12-11 16:37:20.135	infoIntegration	RicezioneBuste	Ricevuta richiesta applicativa	
2018-12-11 16:37:20.138	infoIntegration	RicezioneBuste	Gestione Token [Google] (Validazione Introspection) in corso ...	
2018-12-11 16:37:20.273	infoIntegration	RicezioneBuste	Gestione Token [Google] (Validazione Introspection) completata con successo	
2018-12-11 16:37:20.278	infoProtocol	RicezioneBuste	Ricevuto messaggio di cooperazione con identificativo [b9cefcc-5e6a-4bf0-b84c-84250c009c2a]	
2018-12-11 16:37:20.279	infoIntegration	RicezioneBuste	Verifica autorizzazione [token] messaggio con identificativo [b9cefcc-5e6a-4bf0-b84c-84250c009c2a] servizio [gw/Ente gw/PetStore.v2.PUT_pet] in corso ...	
2018-12-11 16:37:20.286	errorIntegration	RicezioneBuste	Verifica autorizzazione [token] messaggio con identificativo [b9cefcc-5e6a-4bf0-b84c-84250c009c2a] servizio [gw/Ente gw/PetStore.v2.PUT_pet] fallita (codice: GOVWAY-1368) (Token claim 'azp' with unexpected value) La richiesta presenta un token non sufficiente per fruire del servizio richiesto	
2018-12-11 16:37:20.287	errorProtocol	RicezioneBuste	Generato messaggio di cooperazione di Errore con identificativo [d63ad91f-0269-4f31-8928-000a82950d41]	
2018-12-11 16:37:20.288	infoIntegration	RicezioneBuste	Risposta ({type:"https://httpstatuses.com/403","title":"403","status":403,"detail":"La richiesta presenta un token non sufficiente per fruire del servizio richiesto","govway_status":"protocol:GOVWAY-1368"}) consegnata al mittente con codice di trasporto: 403	<button>ESPORTA</button>

Figura 88: Diagnostici di una invocazione terminata con errore

• Registrazione ClientId corretto in Controllo degli Accessi

Tramite la *govwayConsole* accedere nuovamente alla maschera di configurazione 'Controllo Accessi' dell'API 'PetStore v2'; all'interno della sezione 'Autorizzare' modificare il valore del claim 'azp' aggiungendo l'applicazione *Playground*:

- aud=407408718192.apps.googleusercontent.com
- azp=client1, client2, 407408718192.apps.googleusercontent.com

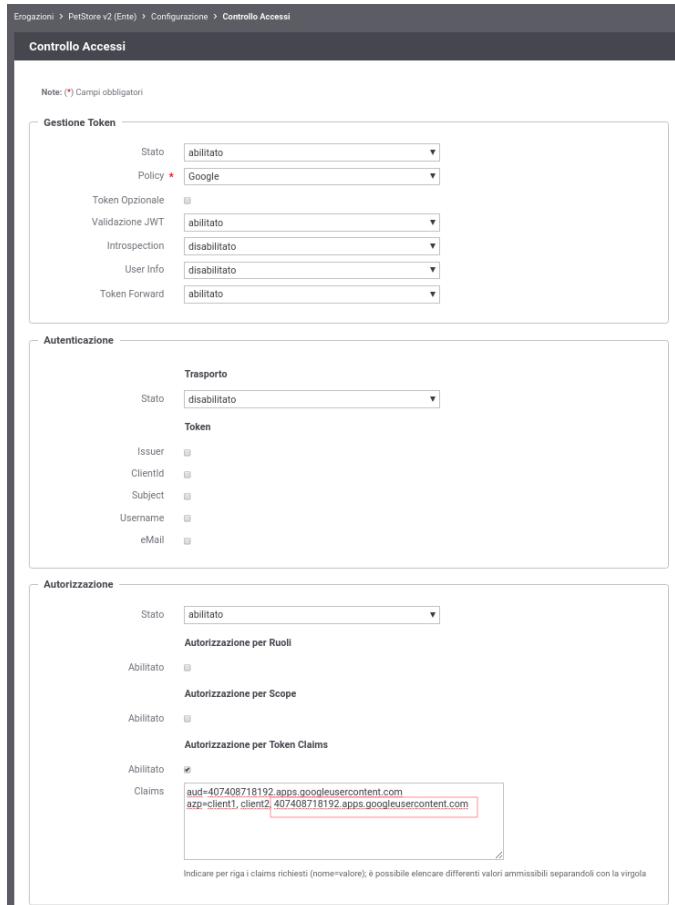


Figura 89: Configurazione OAuth2 - Autorizzazione dei claims corretta

• Nuova invocazione API

Reset Cache delle Configurazioni prima di un nuovo test

Effettuare il reset della cache accedendo alla sezione 'Strumenti' - 'Runtime' e selezionare la voce 'ResetAllCaches'.

Effettuare una nuova invocazione del test.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token= ←
    ACCESS_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
        "id": 3,
        "category": { "id": 22, "name": "dog" },
        "name": "doggie",
        "photoUrls": [ "http://image/dog.jpg" ],
```

```

    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}

```

L'esito dell'aggiornamento termina stavolta con successo con un codice http 200 e una risposta json equivalente alla richiesta.

5.1.6 Autorizzazione XACML

GovWay può essere configurato per effettuare verifiche, dei claims ottenuti tramite la validazione dell'access token, più complesse rispetto a quelle descritte nei precedenti paragrafi. Per farlo si deve utilizzare una policy XACML.

Per simulare lo scenario utilizzeremo sempre il servizio *Playground* e l'*Authorization Server di Google* descritto nella precedente sezione Sezione 5.1.1.

Per l'autorizzazione verrà caricata su GovWay una XACML Policy, di seguito descritta, che non possiede una vera logica autorizzativa ma serve solo a titolo di esempio per descrivere la funzionalità.

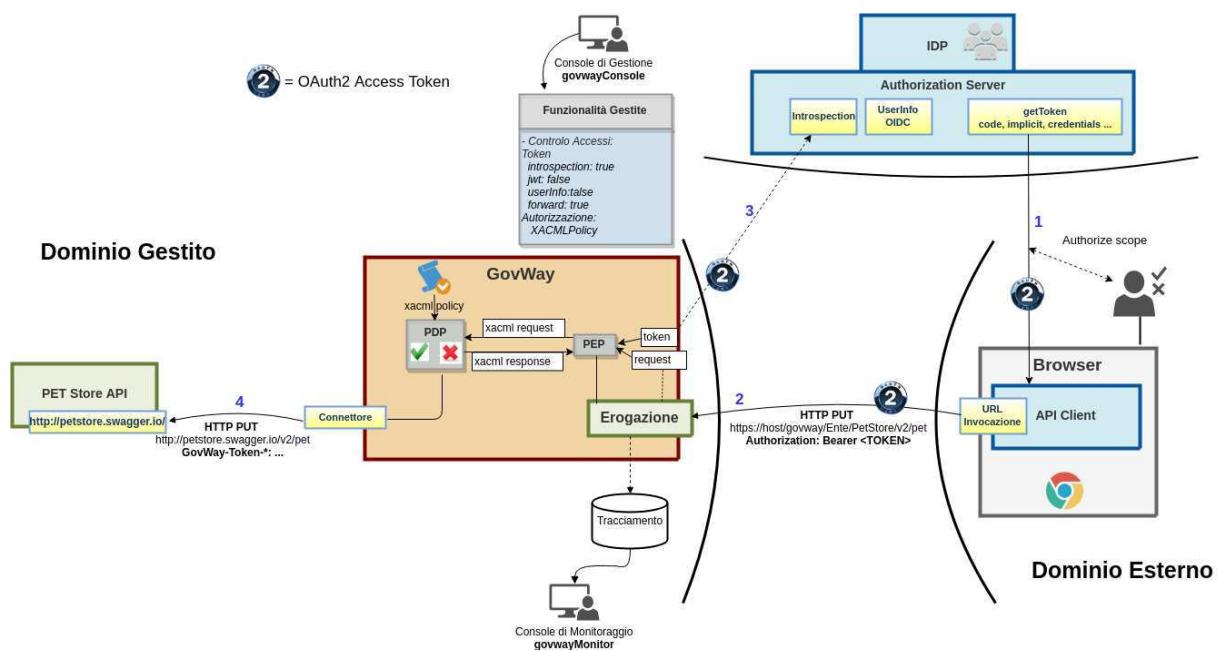


Figura 90: Scenario OAuth con autorizzazione XACMLPolicy

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 le informazioni presenti nel token. Nella tabella seguente vengono forniti i dettagli sui nomi dei parametri.

Nome	Descrizione
<i>Sezione 'Action'</i>	
org:govway:action:token:audience	Destinatario del token
org:govway:action:token:scope	Lista di scopes
org:govway:action:token:jwt:claim:<nome>=<valore>	Tutti i claims presenti nel jwt validato
org:govway:action:token:introspection:claim:<nome>=<valore>	Tutti i claims presenti nella risposta del servizio di introspection
org:govway:action:provider	Indica il soggetto erogatore del servizio
org:govway:action:service	Indica il servizio nel formato tipo/nome
org:govway:action:action	Nome dell'operazione del servizio invocata
org:govway:action:url	Url di invocazione utilizzata dal mittente

Nome	Descrizione
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	Profilo di utilizzo associata al servizio richiesto (es. scoop)
<i>Sezione 'Subject'</i>	
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: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>=<valore>	Tutti i claims presenti nella risposta del servizio di UserInfo
org:govway:subject:organization	Indica il soggetto fruitore
org:govway:subject:client	Identificativo del servizio 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

Di seguito un esempio di XACMLPolicy che traduce in policy l'esempio descritto nella precedente sezione Sezione 5.1.5. La verifica che andiamo a definire è la seguente:

- *Audience* (claim 'aud'): contenga l'identificativo dell'applicazione *Playground* come destinatario del token
- *Applicazione Client* (claim 'azp'): controlleremo che il client appartenga ad uno delle applicazioni conosciute. Nell'elenco, non inseriremo immediatamente l'identificativo di *Playground* in modo che l'autorizzazione fallisca in un primo test.

Per conoscere l'identificativo dell'applicazione *Playground*

E' possibile vedere una precedente transazione terminata con successo per conoscere l'esatto valore associato all'applicazione *Playground* (es. Figura 55).

```
<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:and">
      <Apply
        FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <ActionAttributeDesignator
            AttributeId="org:govway:action:token:audience"
```

```
  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"> ←
    407408718192.apps.googleusercontent.com</AttributeValue>
</Apply>
</Apply>

<Apply
  FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
  <SubjectAttributeDesignator
    AttributeId="org:govway:subject:token:clientId"
    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">client1</ →
      AttributeValue>
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">client2</ →
      AttributeValue>
  </Apply>
</Apply>
</Condition>
</Rule>
<Rule Effect="Deny" RuleId="ko" />
</Policy>
```

• Configurazione Controllo degli Accessi

Accedere alla sezione '*Erogazioni*' e selezionare l'API precedentemente registrata '*PetStore v2*'. Dopodichè accedere, dal dettaglio dell'erogazione, alla sezione '*Configurazione*' dove vengono visualizzate le funzionalità attive. Cliccare sulla voce presente nella colonna '*Controllo Accessi*' e procedere con la seguente configurazione all'interno della sezione '*Gestione Token*':

- *Stato*: abilitato
- *Policy*: Google
- *Validazione JWT*: disabilitato
- *Introspection*: abilitato
- *User Info*: disabilitato
- *Token Forward*: abilitato

Procedere inoltre con la seguente configurazione all'interno della sezione '*Autorizzazione*':

- *Autorizzazione - Stato*: xacml-Policy
- *Policy*: caricare la xacml policy descritta precedentemente

Effettuata la configurazione salvarla cliccando sul pulsante 'Salva'.

Note: (*) Campi obbligatori

Gestione Token

- Stato: abilitato
- Policy *: Google
- Token Opzionale:
- Validazione JWT: disabilitato
- Introspection: abilitato
- User Info: disabilitato
- Token Forward: abilitato

Autenticazione

- Trasporto: disabilitato
- Token**
- Issuer:
- ClientId:
- Subject:
- Username:
- eMail:

Autorizzazione

- Stato: xacml-Policy
- Fonte Ruoli: Qualsiasi
- Policy: Choose File No file chosen
xacmlPolicyTest.xml

SALVA

Figura 91: Configurazione OAuth2 - Autorizzazione XACML Policy

• Invocazione API

Reset Cache delle Configurazioni prima di un nuovo test

Le configurazioni accedute da GovWay vengono mantenute in una cache dopo il primo accesso per 2 ore, è quindi necessario forzare un reset della cache. Per farlo accedere alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Per effettuare il test utilizzare il token ottenuto come descritto nella sezione Sezione 5.1.1.

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token= ACCESS_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento termina con un codice di errore http 403 e una risposta problem+json che riporta la motivazione:

```
HTTP/1.1 403 Forbidden
Content-Type: application/problem+json
```

```

Transfer-Encoding: chunked
Server: GovWay
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
  "type": "https://httpstatuses.com/403",
  "title": "Forbidden",
  "status": 403,
  "detail": "Il mittente non è autorizzato ad invocare il servizio gw/PetStore (versione:2) ← erogato da gw/Ente (result-1 DENY code:urn:oasis:names:tc:xacml:1.0:status:ok)",
  "govway_status": "protocol:GOVWAY-1352"
}

```

• Consultazione Tracce in errore

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Dalla figura Figura 92 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con errore con esito *Autorizzazione Negata*.

Lista Transazioni: record [1 - 4]							
	Data Ingresso Richiesta ▾	Tipologia	Esito	Fruitore	Erogatore	API	Azione
<input type="checkbox"/>	2018-12-05 17:20:12	Erogazione	Autorizzazione Negata		Ente	PetStore v2	PUT_pet
<input type="checkbox"/>	2018-12-05 17:20:11	Erogazione	Autorizzazione Negata		Ente	PetStore v2	PUT_pet
<input type="checkbox"/>	2018-12-05 17:20:11	Erogazione	Autorizzazione Negata		Ente	PetStore v2	PUT_pet
<input type="checkbox"/>	2018-12-05 17:16:45	Erogazione	Autorizzazione Negata		Ente	PetStore v2	PUT_pet

Figura 92: Tracce delle invocazioni terminate con errore 'Autorizzazione Negata'

Accedendo al dettaglio di una transazione terminata in errore, e visualizzandone i diagnostici è possibile comprendere che l'errore è dovuto ad una decisione 'deny' ottenuta dopo la valutazione della policy: '(result-1 DENY code:urn:oasis:names:tc:xacml:1.0:status:ok)

Storico > Intervallo Temporale > Dettagli Transazione > Messaggi Diagnostici				
Lista Diagnostici: record [1 - 8] su 8				
Data	Severità	Funzione	Messaggio	
2018-12-12 09:08:35,401	infoIntegration	RicezioneBuste	Ricevuta richiesta applicativa	
2018-12-12 09:08:35,403	infoIntegration	RicezioneBuste	Gestione Token [Google] (Validazione Introspection) in corso ...	
2018-12-12 09:08:35,532	infoIntegration	RicezioneBuste	Gestione Token [Google] (Validazione Introspection) completata con successo	
2018-12-12 09:08:35,537	infoProtocol	RicezioneBuste	Ricevuto messaggio di cooperazione con identificativo [1660df45-758d-4cd0-9fa1-bb10c7ba739c] servizio [gw/Ente gw/PetStore v2.PUT_pet] in corso	
2018-12-12 09:08:35,537	infoIntegration	RicezioneBuste	Verifica autorizzazione [xacmlPolicy] messaggio con identificativo [1660df45-758d-4cd0-9fa1-bb10c7ba739c] servizio [gw/Ente gw/PetStore v2.PUT_pet] fallita (codice: GOVWAY-1352) Il mittente non è autorizzato ad invocare il servizio gw/PetStore (versione:2) erogato da gw/Ente (result-1 DENY code:urn:oasis:names:tc:xacml:1.0:status:ok)	
2018-12-12 09:08:35,902	errorIntegration	RicezioneBuste	Verifica autorizzazione [xacmlPolicy] messaggio con identificativo [1660df45-758d-4cd0-9fa1-bb10c7ba739c] servizio [gw/Ente gw/PetStore v2.PUT_pet] fallita (codice: GOVWAY-1352) Il mittente non è autorizzato ad invocare il servizio gw/PetStore (versione:2) erogato da gw/Ente (result-1 DENY code:urn:oasis:names:tc:xacml:1.0:status:ok)	
2018-12-12 09:08:35,903	errorProtocol	RicezioneBuste	Generato messaggio di cooperazione di Errore con identificativo [b898b42-0468-42d6-9451-aa9e9a669dcdfa]	
2018-12-12 09:08:35,904	infoIntegration	RicezioneBuste	Risposta (Type: "https://httpstatuses.com/403" title: "Forbidden" status: 403 detail: "Il mittente non è autorizzato ad invocare il servizio gw/PetStore (versione:2) erogato da gw/Ente (result-1 DENY code:urn:oasis:names:tc:xacml:1.0:status:ok)" protocol:GOVWAY-1352) consegnata al mittente con codice di trasporto: 403	

Figura 93: Diagnostici di una invocazione terminata con errore

• Registrazione ClientId corretto nella XACMLPolicy

Di seguito un esempio di XACMLPolicy nella quale tra i valori consentiti per l'applicazione client viene aggiunto l'identificativo di *Playground* in modo che l'autorizzazione termini con successo.

```

<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:and">

      <Apply
        FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of" >
        <ActionAttributeDesignator
          AttributeId="org:govway:action:token:audience"
          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"> ←
            407408718192.apps.googleusercontent.com</AttributeValue>
        </Apply>
      </Apply>
    </Condition>
  <Rule Effect="Deny" RuleId="ko" />
</Policy>

```

- **Aggiornamento XACMLPolicy in Controllo degli Accessi**

Tramite la *govwayConsole* accedere nuovamente alla maschera di configurazione '*Controllo Accessi*' dell'API '*PetStore v2*'; all'interno della sezione '*Autorizzare*' caricare la policy aggiornata.

- **Nuova invocazione API**

Reset Cache delle Configurazioni prima di un nuovo test

Effettuare il reset della cache accedendo alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Effettuare una nuova invocazione del test.

```

curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet?access_token= ←
  ACCESS_TOKEN" \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
  "id": 3,

```

```
"category": { "id": 22, "name": "dog" },
"name": "doggie",
"photoUrls": [ "http://image/dog.jpg" ],
"tags": [ { "id": 23, "name": "white" } ],
"status": "available"
}'
```

L'esito dell'aggiornamento termina stavolta con successo con un codice http 200 e una risposta json equivalente alla richiesta.

5.1.7 Token Forward

Tutte le configurazioni descritte nei precedente paragrafi indicavano di abilitare la funzionalità '*Token Forward*' all'interno della sezione '*Gestione Token*' (vedi ad es. Figura 49). Tale configurazione fa sì che GovWay inoltri all'applicativo interno al dominio (nel nostro esempio il servizio *PetStore*) le informazioni inerenti il token ricevuto sotto forma di header http.

Per vedere quali header vengono effettivamente prodotti possiamo utilizzare la funzionalità '*Registrazione Messaggi*' descritta nel dettaglio nella sezione Sezione 10. Accedere alla sezione '*Erogazioni*' e selezionare l'API precedentemente registrata '*PetStore v2*'. Dopodichè accedere, dal dettaglio dell'erogazione, alla sezione '*Configurazione*' dove vengono visualizzate le funzionalità attive. Per abilitare la registrazione degli header cliccare sulla voce presente nella colonna '*Registrazione Messaggi*' e procedere con la seguente configurazione.

- '*Generale - Stato*': ridefinito
- '*Richiesta - Stato*': abilitato
- '*Richiesta - Ingresso*': disabilitare tutte le voci
- '*Richiesta - Uscita*': abilitare solo la voce relativa agli header
- '*Risposta - Stato*': disabilitato

Effettuata la configurazione salvarla cliccando sul pulsante '*Salva*'.

The screenshot shows the 'Registrazione Messaggi' configuration page. It has three main sections: 'Generale', 'Richiesta', and 'Risposta'. In the 'Generale' section, the 'Stato' dropdown is set to 'ridefinito'. In the 'Richiesta' section, the 'Stato' dropdown is set to 'abilitato'. Under the 'Ingresso' heading, the 'Headers', 'Body', and 'Attachments' dropdowns are all set to 'disabilitato'. Under the 'Uscita' heading, the 'Headers' dropdown is set to 'abilitato', while 'Body' and 'Attachments' are set to 'disabilitato'. In the 'Risposta' section, the 'Stato' dropdown is set to 'disabilitato'. At the bottom of the page is a large 'SALVA' button.

Figura 94: Configurazione Registrazione Messaggi per visualizzare Header HTTP

Prima di procedere con una nuova richiesta effettuare il reset della cache delle configurazioni accedendo alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Effettuare quindi una nuova invocazione contenente un *access token* valido e successivamente consultare il dettaglio della transazione tramite la *govWayMonitor*. Nel dettaglio sarà adesso disponibile la voce '*Contenuti Uscita*' (Figura 95) che permette di vedere gli header http prodotti da GovWay (Figura 96).

Nome	Valore
Tipologia	Erogazione (API Gateway)
Erogatore	Ente
API	PetStore v2
Azione	PUT_pet
Profilo Collaborazione	Sincrono
Esito	Ok
Diagnostici	Visualizza Esporta

ID Messaggio	6f6c1374-8744-4345-81ba-534ca8ca0793
Data Ingresso	2018-12-04 12:40:16.371
Data Uscita	2018-12-04 12:40:16.602
Bytes Ingresso	225 B
Bytes Uscita	225 B
Contenuti Uscita	Visualizza Esporta

Figura 95: Dettaglio della transazione con contenuti

Nome	Valore
GovWay-Provider	Ente
GovWay-Token-Expires	2018-12-04_13:16:15.000
GovWay-Service-Type	gw
GovWay-Token-Scopes	https://www.googleapis.com/auth/plus.me
GovWay-Token-ClientId	407408718192.apps.googleusercontent.com
GovWay-Token-Subject	106235657592654397689
accept	application/json
User-Agent	GovWay
GovWay-Message-ID	6f6c1374-8744-4345-81ba-534ca8ca0793
GovWay-Service	PetStore
GovWay-Token-ProcessTime	2018-12-04_12:40:16.582
GovWay-Token-Audience	407408718192.apps.googleusercontent.com
GovWay-Action	PUT_pet
GovWay-Provider-Type	gw
GovWay-Transaction-ID	9319b9d7-0458-4599-84e1-09a583d0bcd4
GovWay-Service-Version	2

Figura 96: Header HTTP prodotti da GovWay contenenti le informazioni sul Token

Le informazioni, inerenti il token ricevuto, trasmesse sotto forma di header http all'applicativo dietro il Gateway, rappresenta la modalità di default di GovWay per quanto concerne la Token Policy 'Google'. GovWay supporta anche differenti modalità di consegna di tali informazioni che possono essere attivate accendendo alla voce del menù 'Configurazione - Token Policy', selezionando una policy (es. Google) e accedendo alla sezione 'Token Forward'. Le modalità si suddividono tra inoltro del token originale (Figura 98) e inoltre delle informazioni raccolte durante la validazione del token (Figura 97).



Figura 97: Modalità di Forward delle Informazioni Raccolte

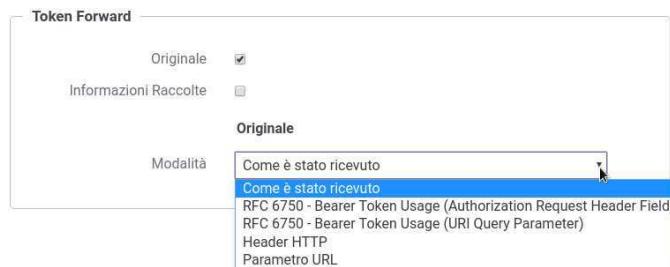


Figura 98: Modalità di Forward del Token Originale

Di seguito vengono descritte le varie modalità di consegna supportate:

- *Inoltro del token originale*: il token originale dopo essere stato validato dal gateway viene comunque inoltrato all'applicativo. È possibile configurare la modalità di inoltro 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.
 - *Header HTTP*: Il token viene inoltrato al destinatario utilizzando un header HTTP il cui nome deve essere specificato.
 - *Parametro URL*: Il token viene inoltrato al destinatario utilizzando un parametro della Query String il cui nome deve essere specificato.
- *Inoltro delle Informazioni Raccolte*: consente di veicolare i dati inerenti il token ricevuto tramite una delle seguenti modalità:
 - *GovWay Headers* (utilizzato nella token policy 'Google' delle sezioni precedenti): I dati raccolti dal token vengono inseriti nei seguenti header HTTP:

GovWay-Token-Issuer
 GovWay-Token-Subject
 GovWay-Token-Username

```

GovWay-Token-Audience
GovWay-Token-ClientId
GovWay-Token-IssuedAt
GovWay-Token-Expire
GovWay-Token-NotToBeUsedBefore
GovWay-Token-Scopes
GovWay-Token-FullName
GovWay-Token-FirstName
GovWay-Token-MiddleName
GovWay-Token-FamilyName
GovWay-Token-EMail

```

- *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": "string"},
    "clientId": {"type": "string"},
    "iat": {
      "type": "string",
      "format": "date-time"
    },
    "expire": {
      "type": "string",
      "format": "date-time"
    },
    "expire": {
      "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"},
        "familyName": {"type": "string"},
        "email": {"type": "string"}
      },
      "additionalProperties": false
    }
  },
  "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 proprietà delle 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 delle url, vengono inseriti dei JWT firmati (caso JWS) o cifrati (caso JWE) contenenti al loro interno il JSON.

5.1.8 Registrazione Authorization Server

Per poter definire politiche di controllo degli accessi basate sui Token è necessario creare delle Token Policy da riferire nel ‘Controllo degli Accessi’ delle specifiche erogazioni e fruizioni come è stato descritto nei precedenti paragrafi (vedi ad es. Figura 49).

Ogni Token Policy definisce la configurazione necessaria al Gateway per interagire con uno specifico Authorization Server. All’interno di una Token Policy vengono definite:

- *Posizione Token*: indica dove il gateway si attende di ricevere il token.
- *Validazione JWT*: indica se la validazione di un token ‘JWT’ ([RFC 7519](#)) è utilizzabile e nel caso tutti i parametri (es. keystore, claim parser) necessari a validarlo secondo la specifica JWS ([RFC 7515](#)) o JWE ([RFC 7516](#)).
- *Token Introspection*: indica se la validazione di un token tramite il servizio Introspection (definito dalla specifica [RFC 7662](#)) è utilizzabile. Poichè tale servizio deve essere disponibile sull’Authorization Server devono essere forniti i parametri necessari all’invocazione (endpoint, configurazione ssl ...).
- *OIDC - UserInfo*: le informazioni riguardanti ad esempio l’*Username* e l’*eMail* potrebbero non essere disponibili dopo la semplice validazione dell’access token (sia introspection che jwt), e per ottenerle è necessario richiedere maggiori informazioni sull’utente tramite il servizio *OIDC UserInfo* (definito dalla specifica [OIDC Connect - UserInfo](#)). Anche per questo servizio, che deve essere disponibile sull’Authorization Server, devono essere forniti i parametri necessari alla sua invocazione (endpoint, configurazione ssl ...).
- *Token Forward*: definisce come le informazioni raccolte durante la validazione del token e/o il token originale vengono inoltrate all’applicativo. Per maggiori dettagli vedere la sezione [Sezione 5.1.7](#)

Per modificare una Token Policy esistente (es. Google), o crearne di nuove, cliccare sul menù nella voce ‘Configurazione - Token Policy’ della govwayConsole. Per creare una nuova policy si utilizza il pulsante *Aggiungi* mentre per modificarne una esistente si deve cliccare sul nome della Policy.

Token Policy > Google

Google

Note: (*) Campi obbligatori

Token Policy

Nome	Google
Descrizione	[Input Field]

Informazioni Generali

Token

Tipo	JWS
Posizione	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 checked="" type="checkbox"/>

Endpoint Token

Connection Timeout *	10000
Read Timeout *	120000

Https	<input checked="" type="checkbox"/>
Proxy	<input type="checkbox"/>

Figura 99: Token Policy di esempio: Google (1/2)

The screenshot shows the configuration interface for a 'Token Policy' named 'Google'. The interface is divided into several sections:

- Validazione JWT**: Set 'Claims Parser' to 'Google - ID Token'.
- Token Introspection**: Set 'Tipo' to 'Google - TokenInfo' and 'URL' to 'https://www.googleapis.com/oauth2/v3/tokeninfo'.
- OIDC - UserInfo**: Set 'Tipo' to 'Google - UserInfo' and 'URL' to 'https://www.googleapis.com/oauth2/v3/userinfo'.
- Https**: Set 'Tipologia' to 'TLSv1.2' and 'Hostname Verifier' to checked. Under 'Autenticazione Server', set 'Tipo' to 'JKS', 'File' to '/Token.jks', 'Password' to '123456', and 'Algoritmo' to 'PKIX'.
- Token Forward**: Set 'Originale' to unchecked and 'Informazioni Raccolte' to checked. Under 'Informazioni Raccolte', set 'Modalità' to 'GovWay Headers'.

A large 'SALVA' button is located at the bottom center of the form.

Figura 100: Token Policy di esempio: Google (2/2)

5.2 Autenticazione

GovWay può essere configurata per autenticare i mittenti che invocano una erogazione o fruizione di API attraverso una delle seguenti modalità:

- **https**: l'invocazione del client deve essere avvenuta su canale ssl e deve aver inviato un proprio certificato client validato dal front-end https. La terminazione ssl può essere gestita direttamente sull'application server (es. wildfly, tomcat) o può essere gestita da un frontend web (es. apache) il quale deve però inoltrare le informazioni ssl all'application server (es. via mod_jk). Un esempio viene descritto nella sezione Sezione 5.2.1.
- **http-basic**: il client deve inoltrare a GovWay delle credenziali di tipo *BASIC* (vedi specifica [RFC 7617](#)). L'username e la password fornita deve corrispondere ad un applicativo o ad un soggetto registrato. Un esempio viene descritto nella sezione Sezione 5.2.2.
- **principal**: questa configurazione richiede che l'autenticazione sia delegata al container via jaas in modo da permettere a GovWay di accedere al principal tramite la api `HttpServletRequest.getUserPrincipal()`. Un esempio viene descritto nella sezione Sezione 5.2.3.

5.2.1 Autenticazione Https

Per tutte le richieste verso una erogazione o fruizione è possibile abilitare l'autenticazione 'ssl' del client in modo da accettare solamente richieste in cui il client ha inviato il proprio certificato.

La terminazione ssl, con la configurazione dei certificati trusted, può essere gestita direttamente sull'application server (es. wildfly, tomcat) o può essere gestita da un frontend web (es. apache) il quale deve però inoltrare le informazioni sui certificati client validati all'application server (es. via mod_jk).

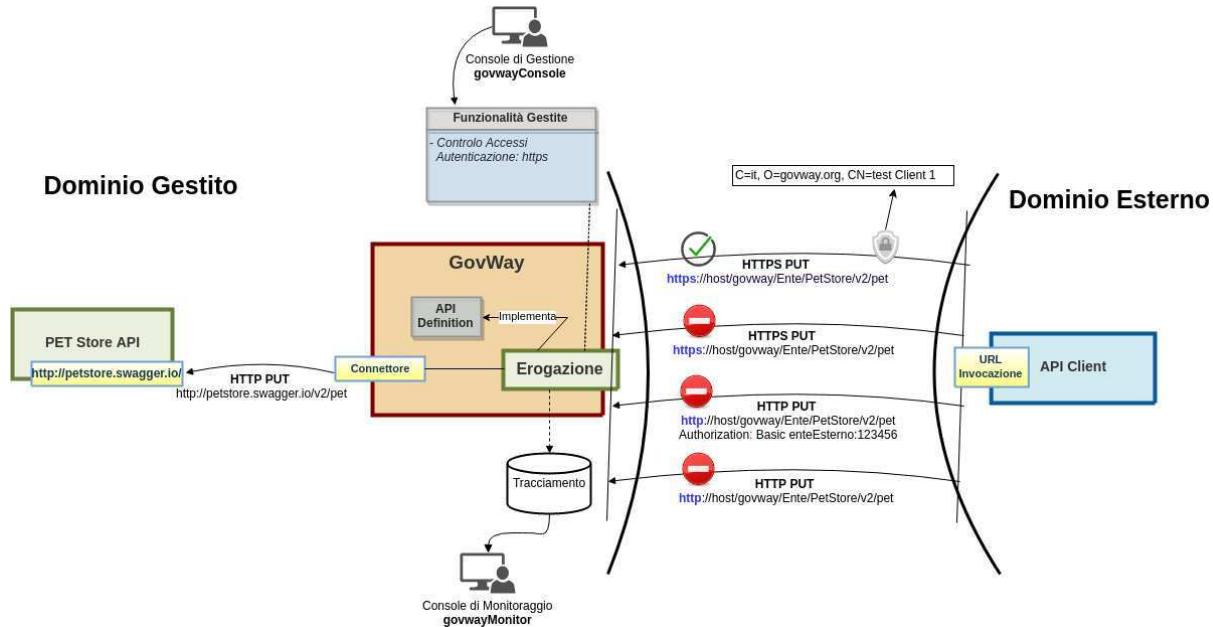


Figura 101: Scenario con autenticazione Https

• Configurazione Controllo degli Accessi

Per abilitare l'autenticazione 'ssl' accedere alla sezione '*Erogazioni*' e selezionare l'API precedentemente registrata '*PetStore v2*'. Dopodichè accedere, dal dettaglio dell'erogazione, alla sezione '*Configurazione*' dove vengono visualizzate le funzionalità attive. Cliccare sulla voce presente nella colonna '*Controllo Accessi*' e procedere con la modifica dello stato relativo all'*'Autenticazione'* con il valore '*'https'*'. Effettuata la configurazione salvarla cliccando sul pulsante '*Salva*'.

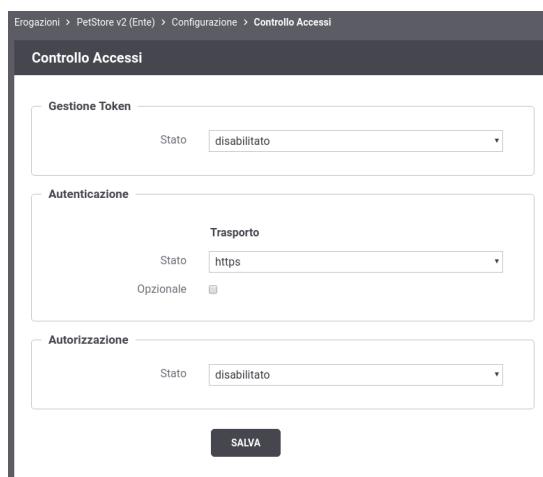


Figura 102: Configurazione Autenticazione Https

Reset Cache delle Configurazioni prima di un nuovo test

Le configurazioni accedute da GovWay vengono mantenute in una cache dopo il primo accesso per 2 ore, è quindi necessario forzare un reset della cache. Per farlo accedere alla sezione '*Strumenti*' - '*Runtime*' e selezionare la voce '*ResetAllCaches*'.

Di seguito replichiamo le invocazioni descritte nello scenario Figura 101 e contestualmente vengono mostrate le funzionalità specifiche fornite da GovWay.

• Invocazione con certificato client ssl

Per effettuare una invocazione fornendo un certificato client è possibile utilizzare il seguente comando:

Docker

Nell'esempio si suppone di utilizzare l'installazione di GovWay realizzata tramite '*govway-docker*' disponibile su github all'indirizzo <https://github.com/link-it/govway-docker>.

La directory indicata nei comandi '*DOCKER_DIR*' corrisponde a quella indicata nel comando utilizzato per avviare il docker come descritto nel README del progetto.

La password '*PASSWORD_CHIAVE_PRIVATA*' utilizzata nel comando deve corrispondere a quella presente nel file '*DOCKER_DIR/pki/esempi/test_Client_1/ee_test_Client_1 README.txt*'

```
curl --resolve test.govway.org:8443:127.0.0.1 -v -X PUT "https://test.govway.org:8443/ ←
govway/Ente/PetStore/v2/pet" \
--cacert DOCKER_DIR/pki/esempi/test_Client_1/ca_test.cert.pem \
--pass 'PASSWORD_CHIAVE_PRIVATA' \
--key DOCKER_DIR/pki/esempi/test_Client_1/ee_test_Client_1.key.pem \
--key-type PEM \
--cert DOCKER_DIR/pki/esempi/test_Client_1/ee_test_Client_1.cert.pem \
--cert-type PEM \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento termina con successo con un codice http 200 e una risposta json equivalente alla richiesta.

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Accedendo al dettaglio di una transazione, come mostrato dalla figura Figura 103 si può vedere come il subject del certificato client utilizzato dal chiamante sia stato associato alla traccia.



Figura 103: Traccia dell'invocazione contenente il subject del certificato client

Sempre attraverso la console *govwayMonitor* è possibile ricercare tutte le transazioni che sono transitate sul gateway relative ad uno specifico certificato client. Per farlo si deve modificare i parametri relativi alla sezione '*Filtro Dati Mittente*' presenti nel filtro di ricerca dello storico delle transazioni indicando:

- *Tipo*: selezionare l'opzione 'Identificativo Autenticato'
- *Autenticazione*: selezionare l'opzione 'https'
- *Ricerca Esatta*: se la ricerca la si vuole effettuare fornendo l'intero subject indicare 'si', se invece si fornisce una informazione parziale del subject indicare 'no'.
- *Case Sensitive*: indica se la ricerca deve essere effettuata considerando le maiuscole e minuscole.
- *Identificativo*: subject complessivo o porzione del subject da cercare

I criteri di ricerca descritti nella figura Figura 104 ricercano le transazioni che contengono il subject utilizzato nell'esempio precedente. Per estrarre il subject dal certificato client è possibile utilizzare ad esempio il seguente comando:

```
openssl x509 -in DOCKER_DIR/pki/esempi/test_Client_1/ee_test_Client_1.cert.pem -text - ←
noout
```

e si ottiene un output simile al seguente dove è possibile recuperare il subject '*C=it, O=govway.org, CN=test Client 1*':

```
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 203 (0xcb)
Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=it, O=govway.org, CN=GovWay CA
Validity
    Not Before: Dec 14 09:07:37 2018 GMT
    Not After : Dec 3 09:07:37 2020 GMT
Subject: C=it, O=govway.org, CN=test Client 1
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
    ....
```

Storico > Intervallo Temporale

Intervallo Temporale

Filtro Temporale

Periodo: Ultima ora

Filtro Dati API

Tipo: Erogazione
Soggetto Fruitore: Selezione Soggetto Fruitore
API: Selezione API

Filtro Dati Mittente

Tipo: Identificativo Autenticato
Autenticazione *: https
Ricerca Esatta: Si No
Case Sensitive: Si No
Identificativo *: C=it, O=govway.org, CN=test Client 1

Filtro Dati Transazione

Esito: [Qualsiasi]
Dettaglio Esito: [Qualsiasi]
Evento:

Bottoni

- NUOVA RICERCA
- FILTRA RISULTATI
- RIPULISCI

Figura 104: Ricerca di transazioni con mittente identificato fornendo l'intero subject del certificato client

I criteri di ricerca descritti nella figura Figura 105 effettuano invece una ricerca che consente di ottenere le transazioni relative al subject utilizzato nell'esempio precedente, fornendo come criterio solamente il valore del 'CN'.

Figura 105: Ricerca di transazioni con mittente identificato fornendo una parte del subject del certificato client

- Invocazione senza certificato ssl.*

Con il seguente comando invochiamo sempre in https senza però fornire un certificato client e si otterrà un errore di autenticazione:

```
curl --resolve test.govway.org:8443:127.0.0.1 -v -X PUT "https://test.govway.org:8443/ →
govway/Ente/PetStore/v2/pet" \
--cacert DOCKER_DIR/pki/esempi/test_Client_1/ca_test.cert.pem \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento termina con un codice di errore http 401 e una risposta problem+json che riporta la motivazione:

```
HTTP/1.1 401 Unauthorized
Content-Type: application/problem+json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824
{
```

```

"type": "https://httpstatuses.com/401",
"title": "Unauthorized",
"status": 401,
"detail": "Autenticazione fallita, credenziali non fornite",
"govway_status": "protocol:GOVWAY-109"
}

```

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Dalla figura Figura 106 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con errore con esito *Autenticazione Fallita*.

	Data Ingresso Richiesta	Tipologia	Esito	Fruitore	Erogatore	API	Azione
<input type="checkbox"/>	2018-12-14 11:25:23	Erogazione	Autenticazione Fallita	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-14 11:25:22	Erogazione	Autenticazione Fallita	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-14 11:25:21	Erogazione	Autenticazione Fallita	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-14 11:25:20	Erogazione	Autenticazione Fallita	Ente	PetStore v2	PUT_pet	
<input type="checkbox"/>	2018-12-14 11:08:21	Erogazione	Autenticazione Fallita	Ente	PetStore v2	PUT_pet	

Figura 106: Tracce delle invocazioni terminate con errore 'Autenticazione Fallita'

Accedendo al dettaglio di una transazione terminata in errore, e visualizzandone i diagnostici è possibile comprendere l'errore che come atteso risulta essere riconducibile al fatto che non sono disponibili le credenziali del client.

Storico > Intervallo Temporale > Dettagli Transazione > Messaggi Diagnostici			
Messaggi Diagnostici: record [1 - 5] su 5			
Data	Severità	Funzione	Messaggio
2018-12-14 11:25:23.429	infoIntegration	RicezioneBuste	Ricevuta richiesta applicativa
2018-12-14 11:25:23.431	infoIntegration	RicezioneBuste	Autenticazione [ssl] in corso ...
2018-12-14 11:25:23.432	errorIntegration	RicezioneBuste	Autenticazione [ssl] fallita : Autenticazione fallita, credenziali non fornite
2018-12-14 11:25:23.433	errorProtocol	RicezioneBuste	Generato messaggio di cooperazione di Errore con identificativo [ef0a8046-7b51-4348-ba38-9b6a48065491]
2018-12-14 11:25:23.434	infoIntegration	RicezioneBuste	Risposta ("{"type":"https://httpstatuses.com/401","title":"Unauthorized","status":401,"detail":"Autenticazione fallita, credenziali non fornite","govway_status":"protocol:GOVWAY-109"})) consegnata al mittente con codice di trasporto: 401

Figura 107: Diagnostici di una invocazione terminata con errore

- *Invocazione in http.*

Con il seguente comando invochiamo il servizio utilizzando http invece che https e si ottiene comunque un errore di autenticazione (sia che vengano generate o meno credenziali basic):

```

curl -v -X PUT "http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet" --basic --user test ←
    :123456 \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
        "id": 3,
        "category": { "id": 22, "name": "dog" },
        "name": "doggie",
        "photoUrls": [ "http://image/dog.jpg" ],
        "tags": [ { "id": 23, "name": "white" } ],
        "status": "available"
}'

```

L'esito dell'aggiornamento termina con un codice di errore http 401 e una risposta problem+json che riporta la motivazione:

```

HTTP/1.1 401 Unauthorized
Content-Type: application/problem+json
Transfer-Encoding: chunked
Server: GovWay
GovWay-Transaction-ID: 6c13b9ac-3d60-45a6-9130-297a4d832824

{
  "type": "https://httpstatuses.com/401",
  "title": "Unauthorized",
  "status": 401,
  "detail": "Autenticazione fallita, credenziali non fornite",
  "govway_status": "protocol:GOVWAY-109"
}

```

5.2.1.1 Identificazione dei Mittenti

Il subject ottenuto grazie all'autenticazione 'https' può essere utilizzato da GovWay per identificare un soggetto (client esterno al dominio di gestione) o un applicativo (client interno al dominio di gestione) registrato tramite la '*govwayConsole*'. Al momento della registrazione, ad un soggetto o ad un applicativo gli viene associato il subject.

L'identificazione puntuale di un mittente su GovWay permette di beneficiare delle seguenti funzionalità:

- *Tracciamento*: accedendo al dettaglio di una transazione, oltre alle credenziali utilizzate dal client verrà riportato l'identificativo con cui è stato registrato su GovWay.
- *Ricerca*: nello storico delle transazioni è possibile cercare tutte le transazioni che possiedono il soggetto o l'applicativo mittente registrato su GovWay.
- *Informazioni Statistiche*: sarà possibile ottenere distribuzioni temporali e reports statistici relativi ai soggetti o applicativi registrati (per maggiori dettagli vedi sezione Sezione 15).

Nella figura Figura 108 viene mostrato un esempio di registrazione sia di un soggetto, che rappresenta un client esterno al dominio di gestione, sia di un applicativo interno al dominio gestito.

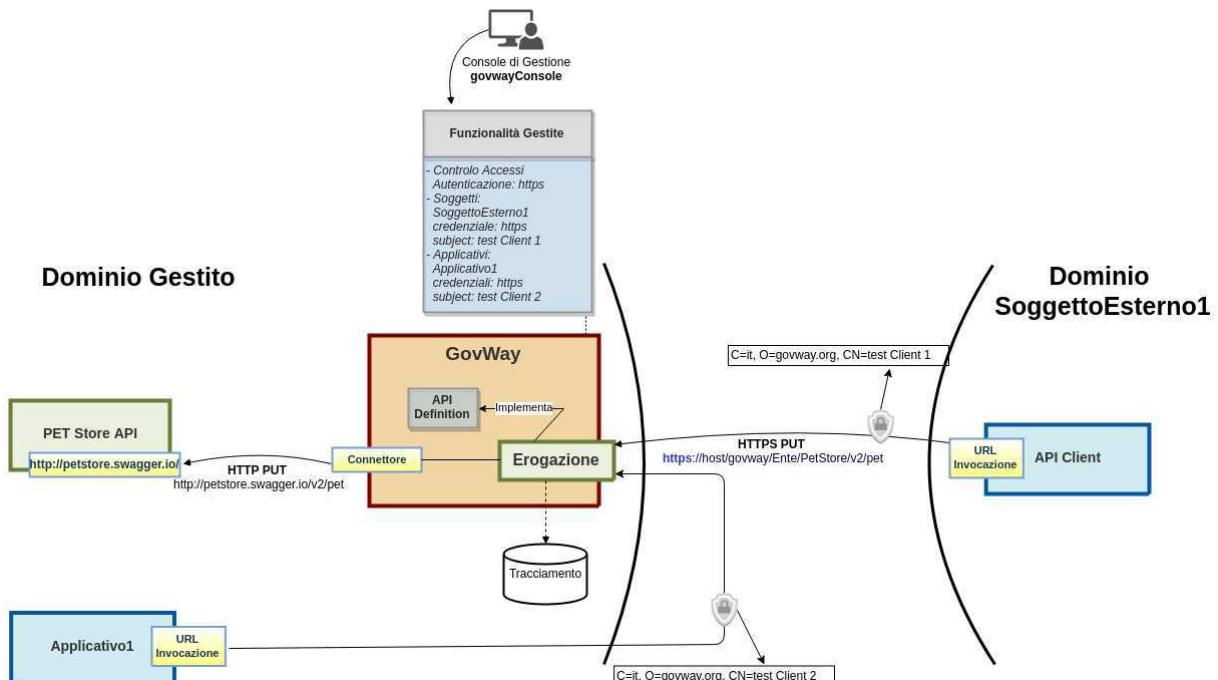


Figura 108: Scenario con autenticazione Https e identificazione dei mittenti

Di seguito viene descritto come realizzare lo Figura 108:

- **Registrazione nuovo Soggetto del dominio esterno**

Accedere alla sezione 'Soggetti' e selezionare il pulsante 'Aggiungi'. Fornire i seguenti dati:

- *Dominio*: selezionare la voce 'Esterno'.
- *Nome*: indicare il nome del Soggetto che rappresenta il nuovo dominio esterno, ad esempio 'SoggettoEsterno1'.
- *Tipologia*: selezionare la voce 'Fruitore'.
- *Descrizione*: opzionalmente è possibile fornire una descrizione generica del soggetto.
- *Modalità Accesso - tipo*: indicare 'https'.
- *Modalità Accesso - subject*: deve essere indicato il Subject del certificato che il client esterno al dominio utilizzerà per invocare GovWay.

Nel nostro esempio si suppone di utilizzare il certificato disponibile in 'DOCKER_DIR/pki/esempi/test_Client_1/ee_test_Client_1.cert'. Per estrarre il subject dal certificato client è possibile utilizzare ad esempio il seguente comando:

```
openssl x509 -in DOCKER_DIR/pki/esempi/test_Client_1/ee_test_Client_1.cert.pem -text -noout
```

e si ottiene un output simile al seguente dove è possibile recuperare il subject 'C=it, O=govway.org, CN=test Client 1':

```
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 203 (0xcb)
Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=it, O=govway.org, CN=GovWay CA
Validity
    Not Before: Dec 14 09:07:37 2018 GMT
    Not After : Dec 3 09:07:37 2020 GMT
Subject: C=it, O=govway.org, CN=test Client 1
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
        ...
        .....
```

Figura 109: Registrazione nuovo Soggetto

- **Registrazione Applicativo interno al dominio**

Accedere alla sezione '*Applicativi*' e selezionare il pulsante '*Aggiungi*'. Fornire i seguenti dati:

- *Nome*: indicare il nome dell'applicativo che rappresenta l'applicazione client interna al dominio di gestione, ad esempio '*Applicativo1*'.
- *Modalità Accesso - tipo*: indicare '*https*'.
- *Modalità Accesso - subject*: deve essere indicato il Subject del certificato che il client interno al dominio utilizzerà per invocare GovWay.

Nel nostro esempio si suppone di utilizzare il certificato disponibile in '*DOCKER_DIR/pki/esempi/test_Client_2/ee_test_Client_2.cert.pem*'. Per estrarre il subject dal certificato client è possibile utilizzare ad esempio il seguente comando:

```
openssl x509 -in DOCKER_DIR/pki/esempi/test_Client_2/ee_test_Client_2.cert.pem -text -noout
```

e si ottiene un output simile al seguente dove è possibile recuperare il subject '*C=it, O=govway.org, CN=test Client 2*':

```
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 203 (0xcb)
Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=it, O=govway.org, CN=GovWay CA
Validity
    Not Before: Dec 14 09:07:37 2018 GMT
    Not After : Dec  3 09:07:37 2020 GMT
Subject: C=it, O=govway.org, CN=test Client 2
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
    ....
```

Figura 110: Registrazione nuovo Soggetto

- *Invocazione con certificato ssl 'test Client 1'*.

Simuliamo l'invocazione dell'api *PetStore* protetta da GovWay tramite autenticazione '*https*' tramite il seguente comando:

```
curl --resolve test.govway.org:8443:127.0.0.1 -v -X PUT "https://test.govway.org:8443/ \
govway/Ente/PetStore/v2/pet" \
--cacert DOCKER_DIR/pki/esempi/test_Client_1/ca_test.cert.pem \
--pass 'PASSWORD_CHIAVE_PRIVATA' \
--key DOCKER_DIR/pki/esempi/test_Client_1/ee_test_Client_1.key.pem \
```

```
--key-type PEM \
--cert DOCKER_DIR/pki/esempi/test_Client_1/ee_test_Client_1.cert.pem \
--cert-type PEM \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

L'esito dell'aggiornamento termina con successo con un codice http 200 e una risposta json equivalente alla richiesta.

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Accedendo al dettaglio di una transazione, come mostrato dalla figura Figura 111 si può vedere come oltre al subject del certificato client utilizzato dal chiamante, alla traccia sia stato associato come mittente il soggetto identificato 'SoggettoEsterno1'.

The screenshot shows a transaction trace interface with the following sections:

- Informazioni Generali** (General Information):

Tipologia	Erogazione (API Gateway)
Fruitore	SoggettoEsterno1
Erogatore	Ente
API	PetStore v2
Azione	PUT_pet
Profilo Collaborazione	Sincrono
Esito	Ok
Diagnostici	Visualizza Esporta
- Dettagli Richiesta** (Request Details):

ID Messaggio	abd0edf8-7e44-4075-97a3-7efbd0bbc696
Data Ingresso	2018-12-14 12:37:04.652
Data Uscita	2018-12-14 12:37:04.771
Bytes Ingresso	225 B
Bytes Uscita	225 B
- Dettagli Risposta** (Response Details):

Data Ingresso	2018-12-14 12:37:05.170
Data Uscita	2018-12-14 12:37:05.181
Bytes Ingresso	150 B
Bytes Uscita	150 B
- Informazioni Mittente** (Sender Information):

ID Autenticato	/o=govway.org/c=it/cn=test Client 1/
Metodo HTTP	PUT
URL Invocazione	[in] /govway/in/Ente/PetStore/v2/pet
Credenziali	(SSL-Subject 'CN=test Client 1, O=govway.org, C=it')
Indirizzo Client	172.17.0.1
Codice Risposta Client	200

Figura 111: Traccia dell'invocazione contenente il soggetto mittente

Sempre attraverso la console *govwayMonitor* è possibile ricercare tutte le transazioni che sono transitate sul gateway relative ad uno specifico soggetto mittente. Per farlo si deve modificare i parametri relativi alla sezione 'Filtro Dati API' presenti nel filtro di ricerca dello storico delle transazioni indicando come soggetto mittente il soggetto 'SoggettoEsterno1'.

The screenshot shows the 'Intervallo Temporale' search interface. It includes several filter sections: 'Filtro Temporale' (Periodo: Ultima ora), 'Filtro Dati API' (Tipo: Erogazione, Soggetto Fruitore: SoggettoEsterno1, API: Selezione API), 'Filtro Dati Mittente' (Tipo: Selezione Tipo), 'Filtro Dati Transazione' (Esito: [Qualsiasi], Dettaglio Esito: [Qualsiasi]), and an 'Evento' field. At the bottom are three buttons: 'NUOVA RICERCA', 'FILTRA RISULTATI', and 'RIPULISCI'.

Figura 112: Ricerca di transazioni di un soggetto mittente

- Invocazione con certificato ssl 'test Client 2'.*

Simuliamo l'invocazione dell'api *PetStore* protetta da GovWay tramite autenticazione '*https*' tramite il seguente comando:

```
curl --resolve test.govway.org:8443:127.0.0.1 -v -X PUT "https://test.govway.org:8443/ →
govway/Ente/PetStore/v2/pet" \
--cacert DOCKER_DIR/pki/esempi/test_Client_2/ca_test.cert.pem \
--pass 'PASSWORD_CHIAVE_PRIVATA' \
--key DOCKER_DIR/pki/esempi/test_Client_2/ee_test_Client_2.key.pem \
--key-type PEM \
--cert DOCKER_DIR/pki/esempi/test_Client_2/ee_test_Client_2.cert.pem \
--cert-type PEM \
-H "accept: application/json" \
-H "Content-Type: application/json" \
-d '{
    "id": 3,
    "category": { "id": 22, "name": "dog" },
    "name": "doggie",
    "photoUrls": [ "http://image/dog.jpg" ],
    "tags": [ { "id": 23, "name": "white" } ],
    "status": "available"
}'
```

La password '*PASSWORD_CHIAVE_PRIVATA*' utilizzata nel comando deve corrispondere a quella presente nel file '*DOCKER_DIR/pki/esempi/test_Client_2/ee_test_Client_2 README.txt*'

L'esito dell'aggiornamento termina con successo con un codice http 200 e una risposta json equivalente alla richiesta.

Attraverso la console *govwayMonitor* è possibile consultare lo storico delle transazioni che sono transitate nel gateway. Accedendo al dettaglio di una transazione, come mostrato dalla figura Figura 113 si può vedere come oltre al subject del certificato client utilizzato dal chiamante, alla traccia sia stato associato l'applicativo mittente identificato come 'Applicativo1'.

Informazioni Fruitore	
Applicativo Fruitore	Applicativo1
ID Autenticato	/o=govway.org/c=it/cn=test Client 2/
Metodo HTTP	PUT
URL Invocazione	[in] /govway/in/Ente/PetStore/v2/pet
Credenziali	(SSL-Subject 'CN=test Client 2, O=govway.org, C=it')
Indirizzo Client	172.17.0.1
Codice Risposta Client	200

Figura 113: Traccia dell'invocazione contenente l'applicativo mittente

Sempre attraverso la console *govwayMonitor* è possibile ricercare tutte le transazioni che sono transitate sul gateway relative ad uno specifico applicativo mittente. Per farlo si deve modificare i parametri relativi alla sezione 'Filtro Dati Mittente' presenti nel filtro di ricerca dello storico delle transazioni indicando:

- *Tipo*: selezionare l'opzione 'Applicativo'
- *Soggetto Fruitore* (sezione 'Filtro Dati API'): selezionare il soggetto del dominio gestito
- *Applicativo*: selezionare l'applicativo mittente delle transazioni che si desidera ricercare

The screenshot shows the 'Intervallo Temporale' search interface. It includes several filter sections:

- Filtro Temporale:** Periodo set to 'Ultima ora'.
- Filtro Dati API:** Tipo set to 'Erogazione', Soggetto Fruitore set to 'Ente', and API set to 'Selezione API'.
- Filtro Dati Mittente:** Tipo set to 'Applicativo', and Applicativo set to 'Applicativo1'.
- Filtro Dati Transazione:** Esito set to '[Qualsiasi]', Dettaglio Esito set to '[Qualsiasi]', and Evento is empty.
- Buttons at the bottom: NUOVA RICERCA, FILTRA RISULTATI, and RIPULISCI.

Figura 114: Ricerca di transazioni di un applicativo mittente

5.2.2 Autenticazione Http Basic

5.2.3 Autenticazione Container

TODO

5.3 Autorizzazione

TODO: Descrizione generica scenario

5.3.1 Autorizzazione Puntuale

TODO

5.3.2 Autorizzazione per Ruoli

TODO

5.3.3 XACML

TODO

6 Rate Limiting

TODO: Descrizione generica scenario

6.1 Numero massimo di Richieste

TODO

6.2 Numero massimo di Richieste Concorrenti

TODO

6.3 Massima Banda Occupabile

TODO

6.4 Tempo Medio di Risposta

TODO

6.5 Numero massimo di Fault Applicativi

TODO

7 Validazione Messaggi

TODO: Descrizione generica scenario

7.1 Validazione API REST

TODO

7.2 Validazione API SOAP

TODO

8 Caching Risposte

TODO: Descrizione generica scenario

9 Sicurezza Messaggi

TODO: Descrizione generica scenario dove si crea sia un mittente che un destinatario a scopi di test.

9.1 WSSecurity Signature

TODO

9.2 WSSecurity Encrypt

TODO

9.3 WSSecurity SAML

TODO

9.4 JWT Signature

TODO

9.5 JWT Encrypt

TODO

10 Registrazione Messaggi

TODO: Descrizione generica scenario

11 Tracciamento

TODO: Descrizione generica scenario

11.1 Correlazione Applicativa su API REST

TODO

11.2 Correlazione Applicativa su API SOAP

TODO

11.3 Disattivazione

TODO

11.4 Livello di Log

TODO

12 MTOM

TODO: Descrizione generica scenario dove viene simulato l'invio e la ricezione.

12.1 Packaging

TODO

12.2 Unpackaging

TODO

12.3 Validazione

TODO

12.4 Verifica

TODO

13 Profilo FatturaPA

TODO: Descrizione generica

13.1 Fatturazione Attiva

TODO

13.2 Fatturazione Passiva

TODO

14 Profilo SPCoop

TODO: Descrizione generica

14.1 Profilo Oneway

TODO

14.2 Profilo Sincrono

TODO

14.3 Profilo Asincrono Simmetrico

TODO

14.4 Profilo Asincrono Asimmetrico

TODO

15 Analisi Statistica

TODO: Descrizione generica

15.1 Distribuzione Temporale

TODO

15.2 Distribuzione per Esiti

TODO
