

---

# How-to

*Release 3.3.9.p1*

**Link.it**

15 nov 2022



|          |  |            |
|----------|--|------------|
| <b>1</b> | <b>Le funzionalità di base</b>               | <b>1</b>   |
| 1.1      | Erogazione API REST . . . . .                | 1          |
| 1.2      | Erogazione API SOAP . . . . .                | 7          |
| 1.3      | Fruizione API . . . . .                      | 14         |
| <b>2</b> | <b>Le funzionalità avanzate</b>              | <b>21</b>  |
| 2.1      | Modalità Multi-Tenant . . . . .              | 21         |
| 2.2      | Gruppi di configurazioni . . . . .           | 25         |
| 2.3      | Gestione CORS . . . . .                      | 39         |
| 2.4      | Sospensione di una API . . . . .             | 42         |
| <b>3</b> | <b>OAuth</b>                                 | <b>53</b>  |
| 3.1      | Validazione tramite Introspection . . . . .  | 54         |
| 3.2      | Validazione JWT . . . . .                    | 62         |
| 3.3      | Autenticazione e OIDC UserInfo . . . . .     | 69         |
| 3.4      | Autorizzazione per Scope . . . . .           | 77         |
| 3.5      | Autorizzazione sui Claims . . . . .          | 89         |
| 3.6      | Autorizzazione XACML . . . . .               | 94         |
| 3.7      | Token Forward . . . . .                      | 100        |
| 3.8      | Registrazione Authorization Server . . . . . | 107        |
| <b>4</b> | <b>Autenticazione Https</b>                  | <b>111</b> |
| 4.1      | Identificazione dei Mittenti . . . . .       | 118        |



# CAPITOLO 1

---

## Le funzionalità di base

---

Raccolta di esempi relativi all'utilizzo delle funzionalità di base.

### 1.1 Erogazione API REST

In questa sezione vengono descritti i passi di configurazione necessari a registrare una API REST implementata da un applicativo interno al proprio dominio di gestione. Nello scenario si suppone che il servizio *PetStore*, disponibile online all'indirizzo <https://petstore.swagger.io/>, sia erogato all'interno del 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 Fig. 1.1. Prima di procedere con la configurazione effettuare il download dell'interfaccia OpenAPI 3.0 del servizio *PetStore* disponibile all'indirizzo "<https://raw.githubusercontent.com/link-it/govway/master/resources/openapi/3.0/openapi.yaml>".

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*: 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*.
- *Formato Specifica*: selezionare “*OpenAPI 3.0*” tra i formati supportati.
- *OpenAPI 3.0*: caricare l'interfaccia API scaricata dall'indirizzo “<https://raw.githubusercontent.com/link-it/govway/master/resources/openapi/3.0/openapi.yaml>”.

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 Fig. 1.3.

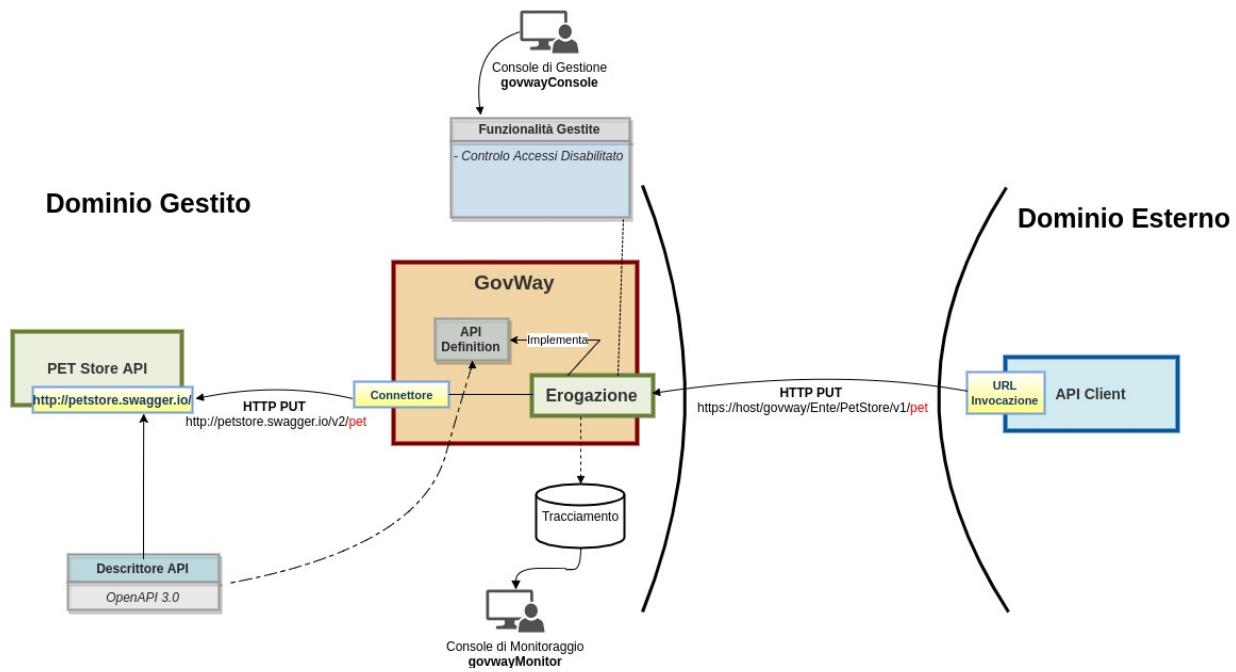


Fig. 1.1: Erogazione di una API Rest tramite GovWay

## 2. Registrazione Erogazione

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

- *Nome*: selezionare l’API precedentemente registrata “*PetStore v2*”.
- *Controllo degli Accessi - Accesso API*: per esporre l’API in modo che sia invocabile da qualunque client in forma anonima selezionare lo stato “*pubblico*”.
- *Connettore - Endpoint*: indicare la *base uri* dove viene erogata l’API nel dominio interno. Per il nostro esempio utilizzare la url:
  - *https://petstore.swagger.io/v2*

### Nota: Verifica del certificato server

Per validare il certificato ritornato dal server “*petstore.swagger.io*” deve essere effettuata una opportuna configurazione del *trustStore tls* come descritto nella sezione *avanzate\_connettori\_https*. Poichè non è obiettivo di questo scenario si suggerisce di disabilitare la validazione del certificato server.

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.

## 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/v1/<uri-risorsa>*

**Soggetto interno al dominio**

API > Aggiungi

Note: (\*) Campi obbligatori

**API**

|             |                                  |
|-------------|----------------------------------|
| Tipo        | Rest                             |
| Nome *      | PetStore                         |
| Descrizione | Servizio di Esempio per API REST |
| Versione    | 1                                |

**Specifiche delle interfacce**

|                   |  |
|-------------------|--|
| Formato Specifica | Open API 3.0   |
| Open API 3.0      | Choose File No file chosen<br>apimatic-converted-petstore.json |

**SALVA**

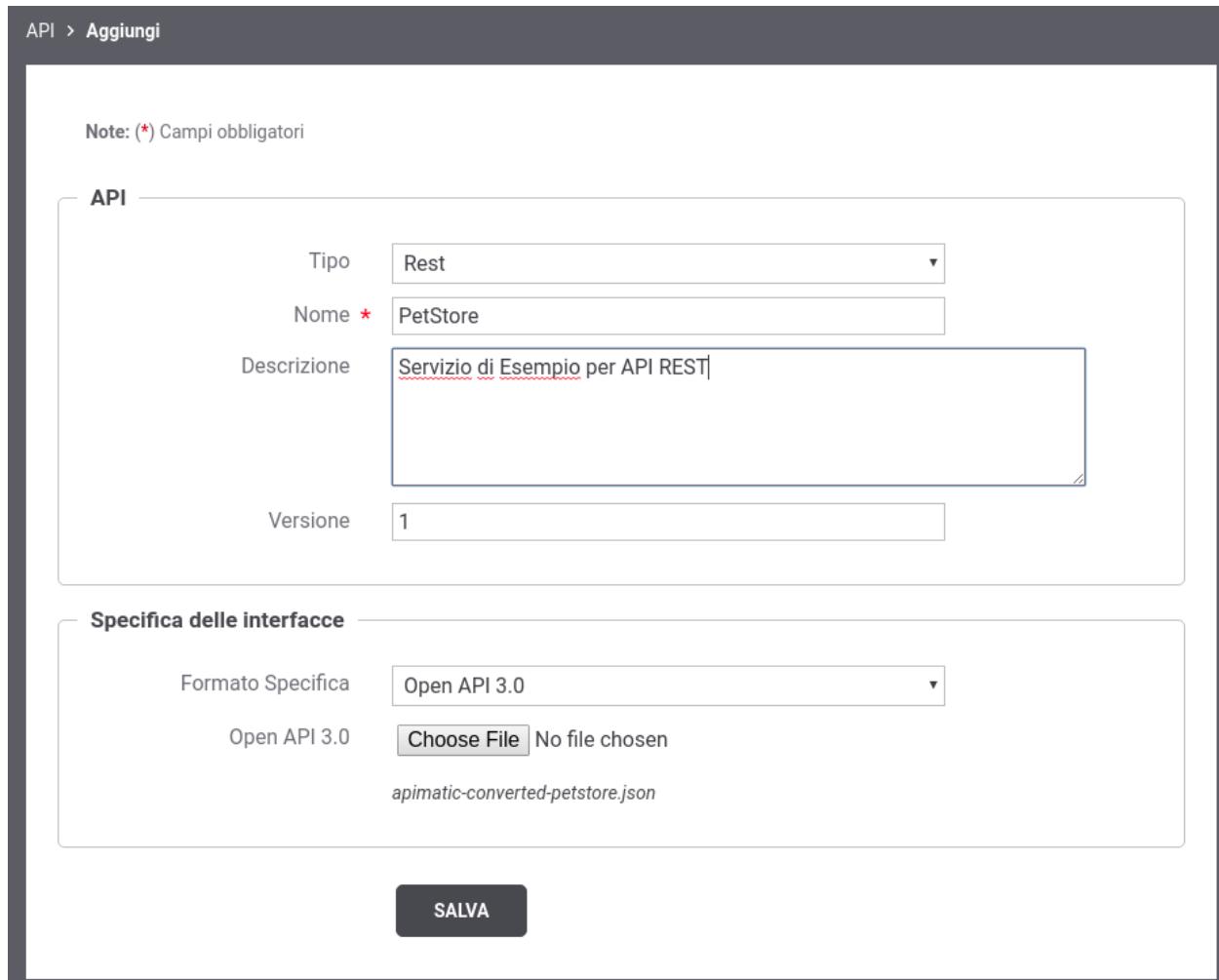
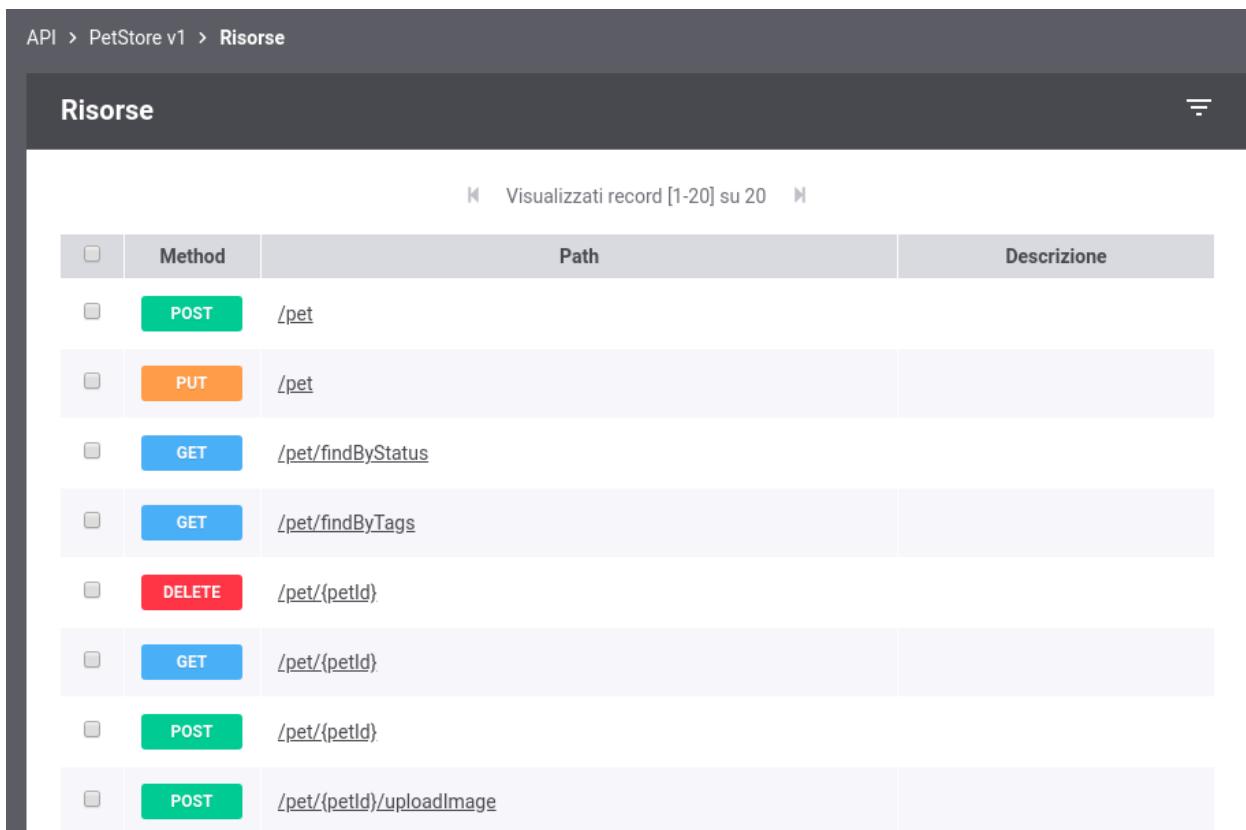


Fig. 1.2: Registrazione di una API



The screenshot shows a list of API resources for the PetStore v1 API. The list is titled 'Risorse' and includes the following entries:

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

Visualizzati record [1-20] su 20

Fig. 1.3: Risorse di una API

Erogazioni > Aggiungi

Note: (\*) Campi obbligatori

**Informazioni Generali**

**API**

|      |             |
|------|-------------|
| Nome | PetStore v1 |
| Tipo | Rest        |

**Controllo degli Accessi**

|             |          |
|-------------|----------|
| Accesso API | pubblico |
|-------------|----------|

**Connettore**

Endpoint \*

Autenticazione Http

Autenticazione Token

AutenticazioneHttps

Proxy

Ridefinisci Tempi Risposta

**Autenticazione Https**

Tipologia

Verifica Hostname

**Autenticazione Server**

Verifica

**Autenticazione Client**

Abilitato

**SALVA**

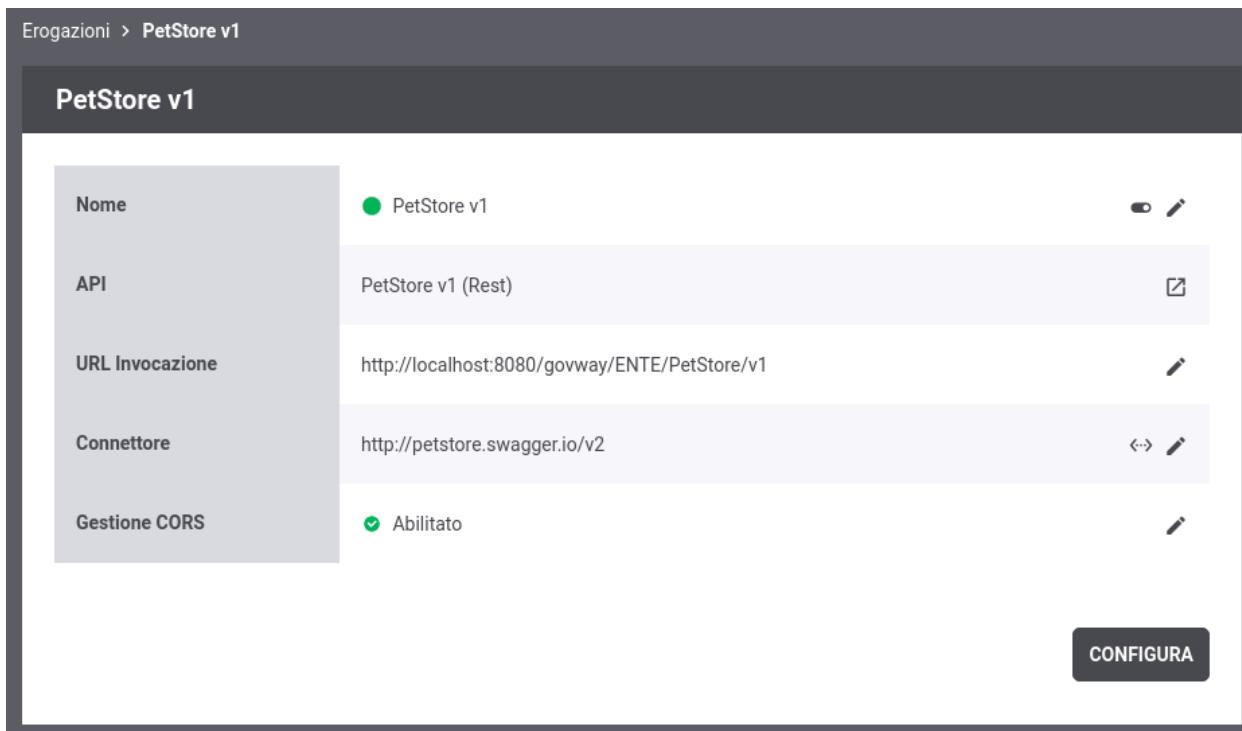


Fig. 1.5: URL di Invocazione dell'API erogata

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/v1/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
{
```

(continues on next page)

(continua dalla pagina precedente)

```

"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 (Fig. 1.6) e conoscere il dettaglio di una singola invocazione (Fig. 1.7).

| Lista Transazioni: record [1 - 5] |                                     |                         |            |       |          |           |             |         |
|-----------------------------------|-------------------------------------|-------------------------|------------|-------|----------|-----------|-------------|---------|
|                                   |                                     | Data Ingresso Richiesta | Tipologia  | Esito | Fruitore | Erogatore | API         | Azione  |
| <input type="checkbox"/>          | <input checked="" type="checkbox"/> | 2018-11-14 14:00:59     | Erogazione | Ok    |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <input checked="" type="checkbox"/> | 2018-11-14 14:00:58     | Erogazione | Ok    |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <input checked="" type="checkbox"/> | 2018-11-14 14:00:57     | Erogazione | Ok    |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <input checked="" type="checkbox"/> | 2018-11-14 14:00:55     | Erogazione | Ok    |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <input checked="" type="checkbox"/> | 2018-11-14 13:30:27     | Erogazione | Ok    |          | Ente      | PetStore v2 | PUT_pet |

Fig. 1.6: Tracce delle invocazioni transitate sul Gateway

## 1.2 Erogazione API SOAP

In questa sezione vengono descritti i passi di configurazione necessari a registrare una API SOAP implementata da un applicativo interno al proprio dominio di gestione. Nello scenario si suppone che il servizio *Credit Card Verification*, disponibile on line all'indirizzo <https://ws.cdyne.com/creditcardverify/luhnchecker.asmx>, sia erogato all'interno del 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 Fig. 1.8. Prima di procedere con la configurazione effettuare il download dell'interfaccia WSDL disponibile in <https://ws.cdyne.com/creditcardverify/luhnchecker.asmx?wsdl>.

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.

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            | <a href="#">Visualizza</a>   <a href="#">Esporta</a> |

**Dettagli Richiesta**

|                |                                      |
|----------------|--------------------------------------|
| 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**

|                |                         |
|----------------|-------------------------|
| 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**

|                        |                                      |
|------------------------|--------------------------------------|
| Metodo HTTP            | PUT                                  |
| URL Invocazione        | [in] /govway/in/Ente/PetStore/v2/pet |
| Indirizzo Client       | 127.0.0.1                            |
| Codice Risposta Client | 200                                  |

**Informazioni Avanzate**

|                       |   |
|-----------------------|---|
| ID Transazione        | ab361e6b-f41f-4a53-a194-60cb19f6b30f  |
| Dominio (ID)          | domain/gw/Ente  |
| Dominio (Soggetto)    | Ente  |
| Connettore            | <a href="http://petstore.swagger.io/v2/pet">http://petstore.swagger.io/v2/pet</a> |
| 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  |

Fig. 1.7: Dettaglio di una invocazione transitata sul Gateway

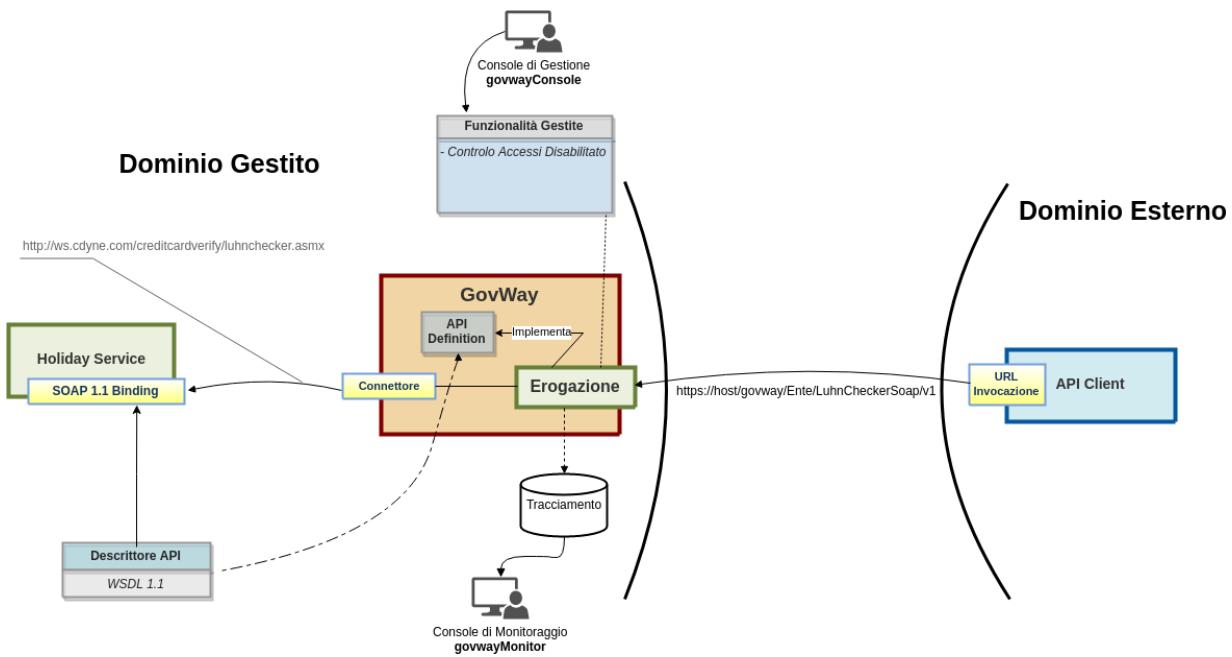


Fig. 1.8: Erogazione di una API SOAP tramite GovWay

- **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>.

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 Fig. 1.10 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*.

## 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*”.
- **Controllo degli Accessi - Accesso API:** per esporre l'API in modo che sia invocabile da qualunque client in forma anonima selezionare lo stato “*pubblico*”.
- **Connettore - Endpoint:** indicare l'endpoint dove viene erogata l'API nel dominio interno. Per il nostro esempio utilizzare la url:
  - <https://ws.cdyne.com/creditcardverify/luhnchecker.asmx>

### Nota: Verifica del certificato server

Per validare il certificato ritornato dal server “*ws.cdyne.com*” deve essere effettuata una opportuna configurazione del trustStore tls come descritto nella sezione *avanzate\_connatori\_https*. Poichè non è obiettivo di questo scenario si suggerisce di disabilitare la validazione del certificato server.

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**

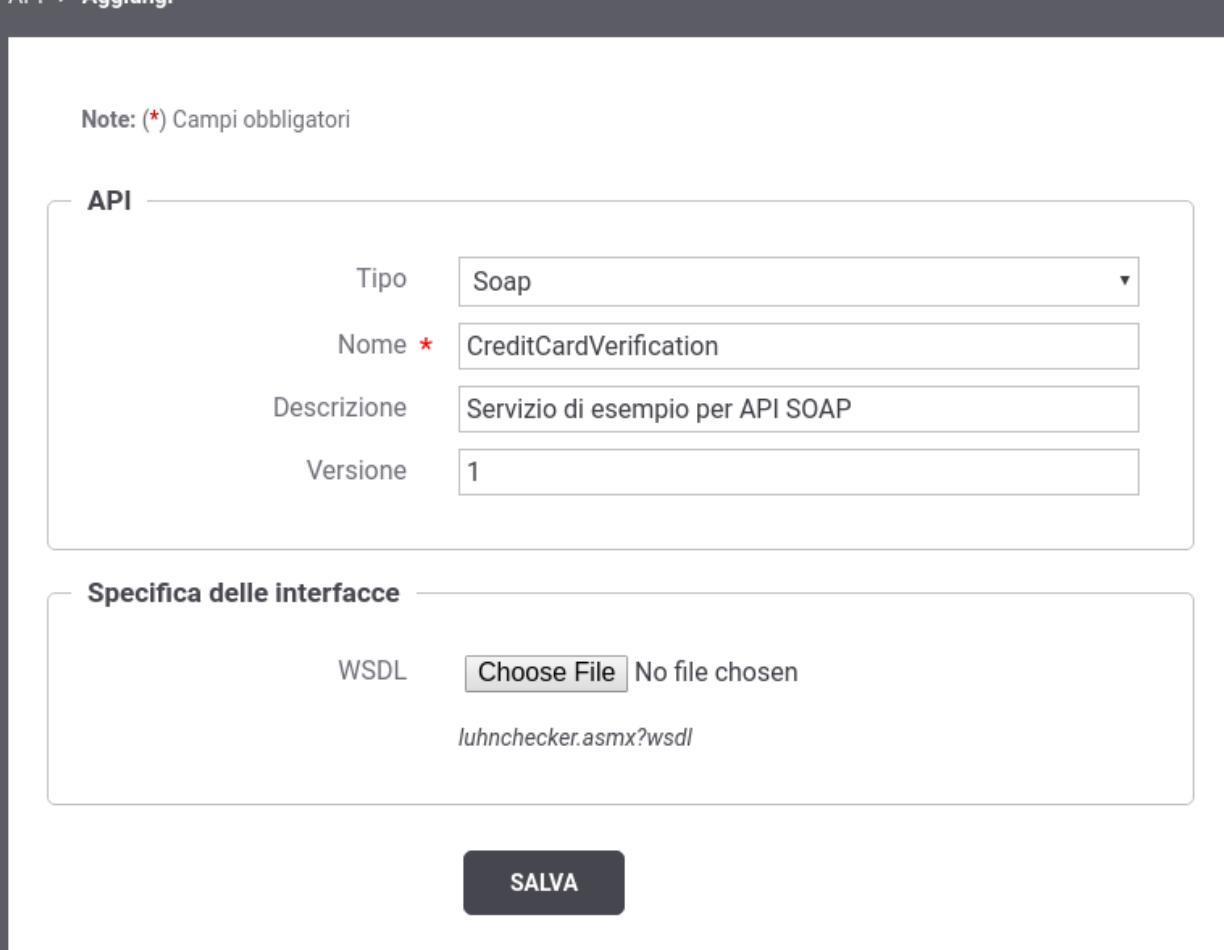


Fig. 1.9: Registrazione di una API SOAP

| API > CreditCardVerification v1 > Servizi |                                     |             |                               |
|---|-------------------------------------|-------------|-------------------------------|
| Servizi                                   |                                     |             |                               |
| Visualizzati record [1-3] su 3            |                                     |             |                               |
|   | Nome                                | Descrizione | Azioni                        |
| <input type="checkbox"/>                  | <a href="#">LuhnCheckerHttpGet</a>  |             | <a href="#">visualizza(1)</a> |
| <input type="checkbox"/>                  | <a href="#">LuhnCheckerHttpPost</a> |             | <a href="#">visualizza(1)</a> |
| <input type="checkbox"/>                  | <a href="#">LuhnCheckerSoap</a>     |             | <a href="#">visualizza(1)</a> |

ELIMINA AGGIUNGI

Fig. 1.10: Servizi di una API SOAP

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.

### 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
```

(continues on next page)

Erogazioni > Aggiungi

Note: (\*) Campi obbligatori

**Informazioni Generali**

**API**

Nome: CreditCardVerification v1

Tipo: Soap

Servizio (Soap) \*: LuhnCheckerSoap

**Controllo degli Accessi**

Accesso API: pubblico

**Connettore**

Endpoint \*: https://ws.cdyne.com/creditcardverify/luhnchecker.asmx

Autenticazione Http:

Autenticazione Token:

AutenticazioneHttps:

Proxy:

Ridefinisci Tempi Risposta:

**Autenticazione Https**

Tipologia: TLSv1.3

Verifica Hostname:

**Autenticazione Server**

Verifica:

**Autenticazione Client**

Abilitato:

**SALVA**

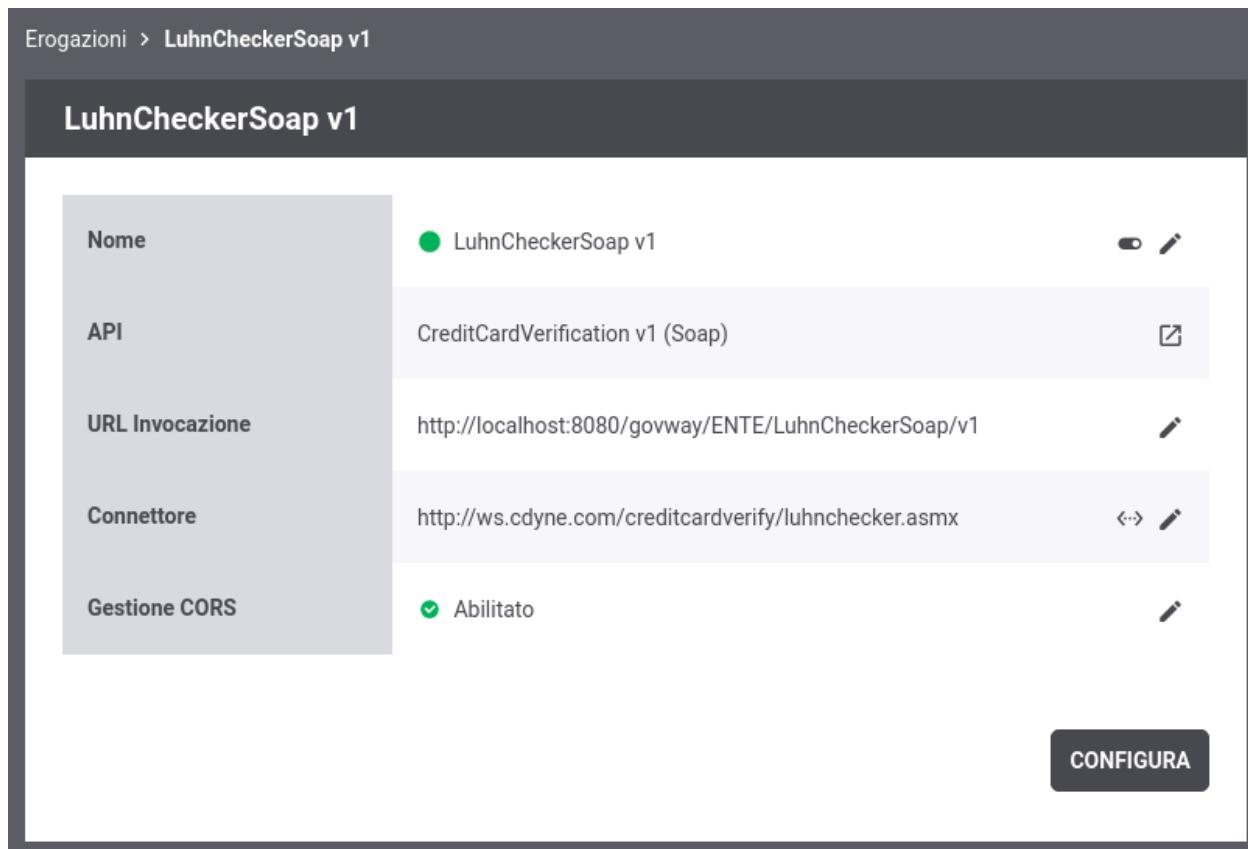


Fig. 1.12: URL di Invocazione dell'API SOAP erogata

(continua dalla pagina precedente)

```
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 *Erogazione API REST*.

### 1.3 Fruizione API

In questa sezione vengono descritti i 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* e, poiché si suppone che lo scenario descritto nella sezione *Erogazione API REST* 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 *Modalità Multi-Tenant* 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 Fig. 1.13.

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:

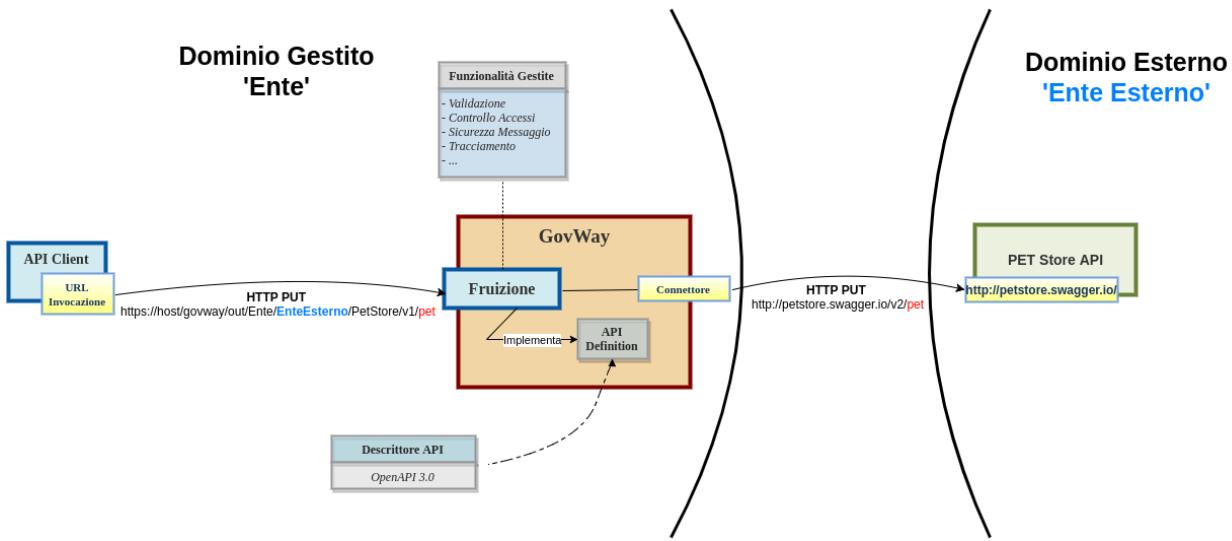


Fig. 1.13: Fruizione di una API tramite GovWay

- *Dominio*: selezionare la voce “*Esterno*”. Questa voce è presente solamente se il *Multitenat* è abilitato su GovWay (maggiori dettagli sono forniti nella sezione *Modalità Multi-Tenant*).
- *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.

## 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*”.
- *Controllo degli Accessi - Accesso API*: per esporre l’API in modo che sia invocabile da qualunque client in forma anonima selezionare lo stato “*pubblico*”.
- *Connettore - Endpoint*: indicare la *base uri* dove viene erogata l’API nel dominio esterno. Per il nostro esempio utilizzare la url:
  - *https://petstore.swagger.io/v2*

### Nota: Verifica del certificato server

Per validare il certificato ritornato dal server “*petstore.swagger.io*” deve essere effettuata una opportuna configurazione del *trustStore tls* come descritto nella sezione *avanzate\_connettori\_https*. Poichè non è obiettivo di questo scenario si suggerisce di disabilitare la validazione del certificato server.

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.

## 3. Invocazione API tramite GovWay

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

The screenshot shows a web interface for adding a new subject. At the top, it says 'Soggetti > Aggiungi'. Below that, a note says 'Note: (\*) Campi obbligatori'. The form is titled 'Soggetto' and contains the following fields:

- Domino: Esterno
- Nome \*: EnteEsterno
- Tipologia: Erogatore
- Descrizione: (empty)

At the bottom of the form is a 'SALVA' button.

Fig. 1.14: Registrazione nuovo Soggetto

- <http://host:port/govway/out/<soggetto-dominio-interno>/EnteEsterno/PetStore/v1/<uri-risorsa>>

#### Nota: 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/v1/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
```

(continues on next page)

Fruizioni > Aggiungi

Note: (\*) Campi obbligatori

**Informazioni Generali**

**API**

Nome: PetStore v1

Tipo: Rest

**Soggetto Erogatore**

Nome: EnteEsterno

**Controllo degli Accessi**

Accesso API: pubblico

**Connettore**

Endpoint \*: https://petstore.swagger.io/v2

Autenticazione Http:

Autenticazione Token:

AutenticazioneHttps:

Proxy:

Ridefinisci Tempi Risposta:

**AutenticazioneHttps**

Tipologia: TLSv1.3

Verifica Hostname:

**Autenticazione Server**

Verifica:

**Autenticazione Client**

Abilitato:

**SALVA**

Fig. 1.15: Registrazione di una fruizione di API

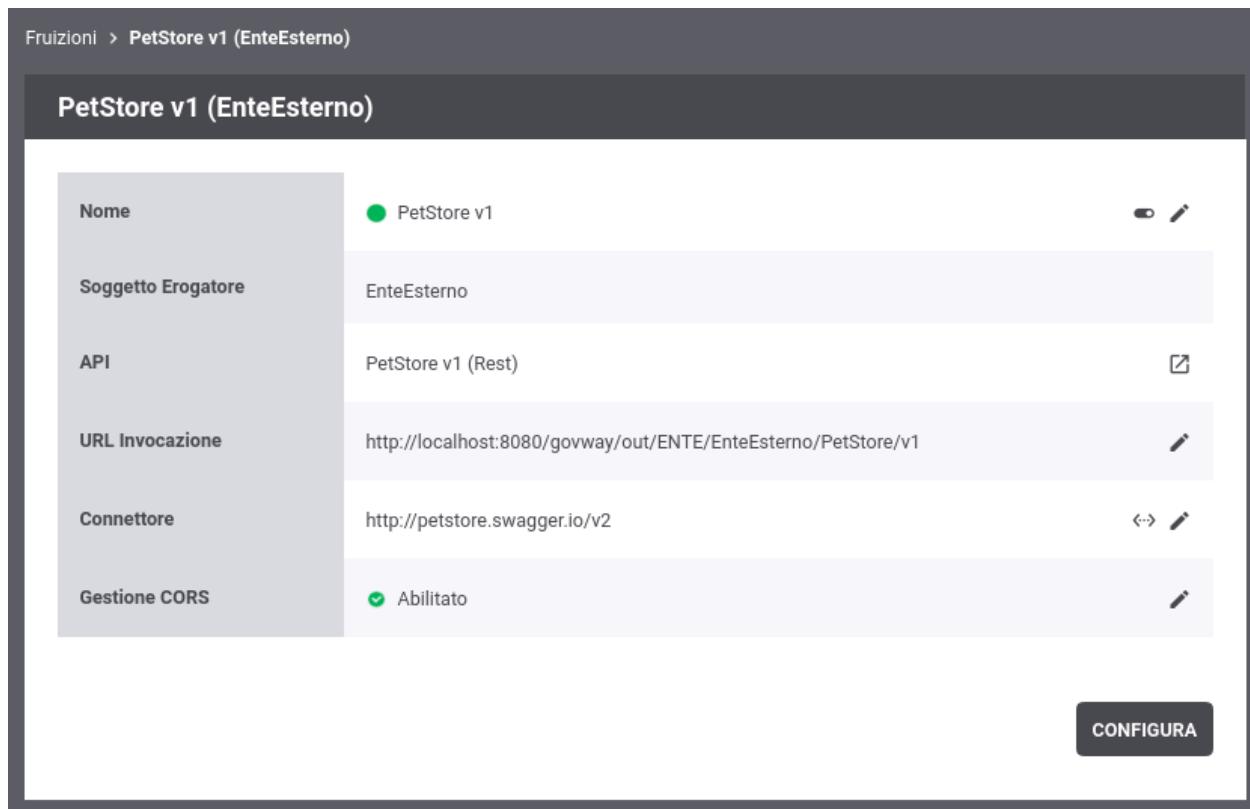


Fig. 1.16: URL di Invocazione dell'API fruita

(continua dalla pagina precedente)

```
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 *Erogazione API REST*.



# CAPITOLO 2

---

## Le funzionalità avanzate

---

Raccolta di esempi relativi alle funzionalità avanzate.

### 2.1 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 Fig. 2.1.

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 *Erogazione API REST* 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”.

#### 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*: optionalmente è possibile fornire una descrizione generica del 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.

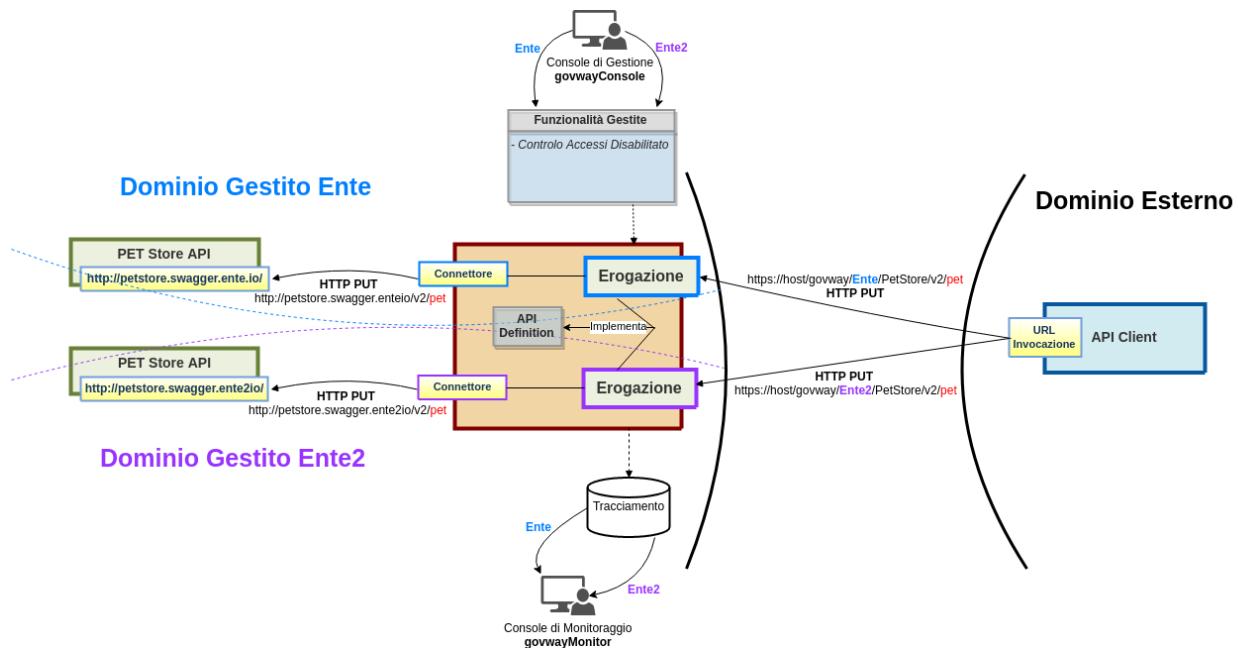


Fig. 2.1: Scenario Multi-Tenant

## Configurazione Generale

**Multi-Tenant**

|                    |  |
|--------------------|--|
| Stato              | abilitato  |
| <b>Fruizioni</b>   |  |
| Soggetto Erogatore | <input type="text" value="Solo Soggetti Esterni"/> ▼ |
| <b>Erogazioni</b>  |  |
| Soggetti Fruitori  | <input type="text" value="Solo Soggetti Esterni"/> ▼ |

Fig. 2.2: Configurazione Multi-Tenant Abilitato

Soggetti > Aggiungi

Note: (\*) Campi obbligatori

**Soggetto**

|             |         |
|-------------|---------|
| Dominio     | Interno |
| Nome *      | Ente2   |
| Descrizione |         |

**SALVA**

Fig. 2.3: Registrazione nuovo Soggetto

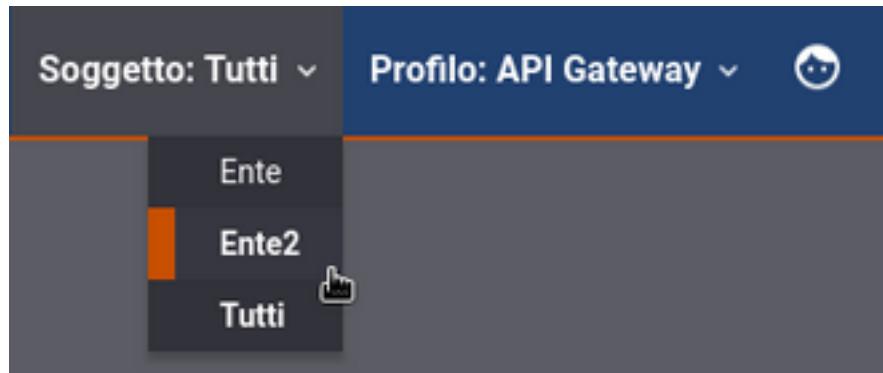


Fig. 2.4: Selezione del Soggetto

#### 4. Registrazione Erogazione

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

- *Nome*: selezionare l’API precedentemente registrata “*PetStore v2*”.
- *Controllo degli Accessi - Accesso API*: per esporre l’API in modo che sia invocabile da qualunque client in forma anonima selezionare lo stato “*pubblico*”.
- *Connettore - Endpoint*: indicare la *base uri* dove viene erogata l’API nel dominio interno. Per il nostro esempio utilizzare sempre la url:
  - *https://petstore.swagger.io/v2*

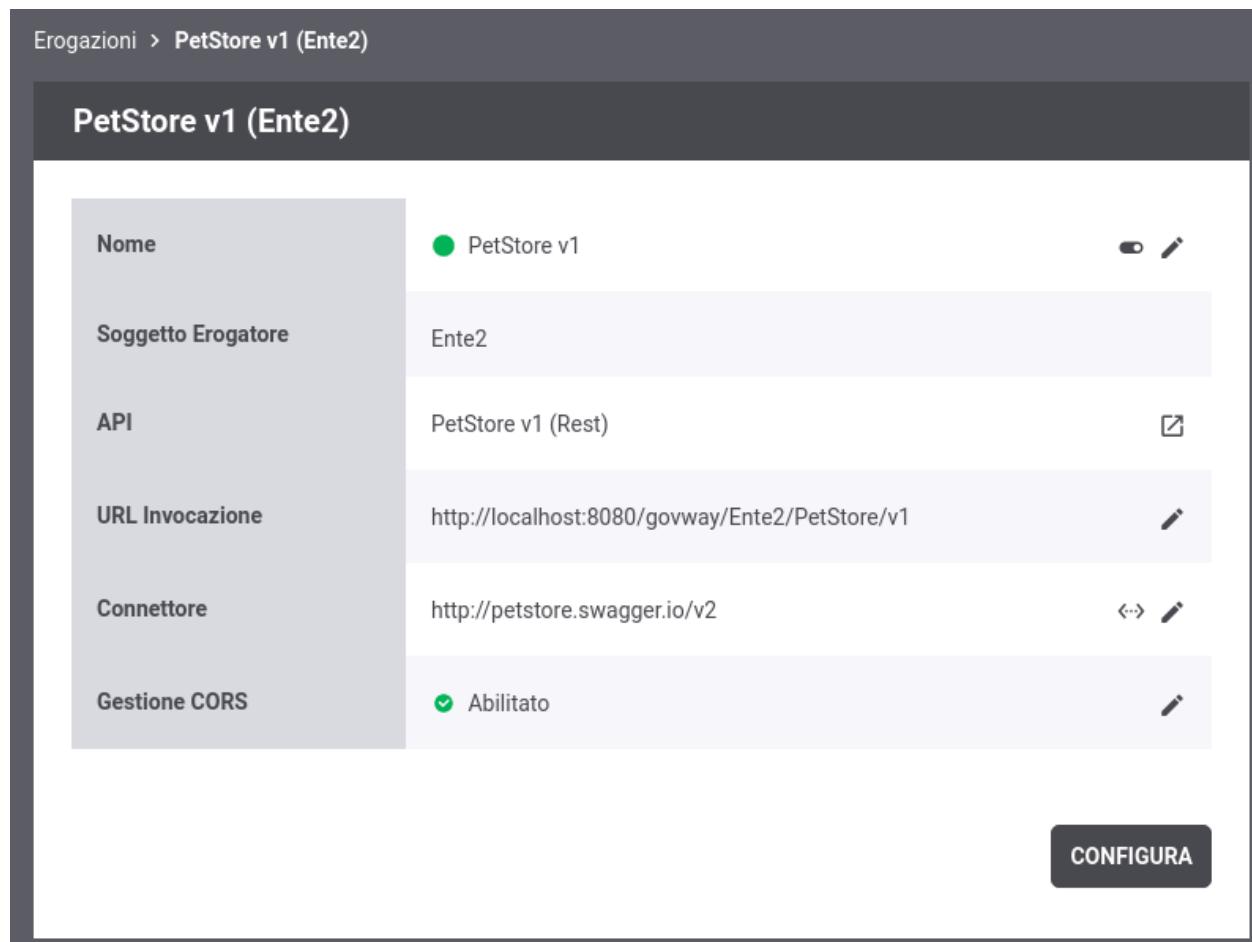
Effettuato il salvataggio, l’API erogata sarà consultabile all’interno dell’elenco delle erogazioni. Accedendo al dettaglio si potrà conoscere l’*url di invocazione* che deve essere comunicata ai client che desiderano invocare l’API.

---

**Nota: Nome del Soggetto presente nella url di invocazione**

Come si può vedere dalla Fig. 2.5 il soggetto *Ente2* compare nella url indicata.

---



| Nome               | PetStore v1                                    | • | • |
|--------------------|--|---|---|
| Soggetto Erogatore | Ente2  |   |   |
| API                | PetStore v1 (Rest)                             | ☒ |   |
| URL Invocazione    | http://localhost:8080/govway/Ente2/PetStore/v1 | • |   |
| Connettore         | http://petstore.swagger.io/v2                  | ↔ | • |
| Gestione CORS      | Abilitato                                      | ✓ | • |

**CONFIGURA**

Fig. 2.5: 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/v1/<uri-risorsa>>

```
curl -v -X PUT "http://127.0.0.1:8080/govway/Ente2/PetStore/v1/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 [Erogazione API REST](#), previa selezione del soggetto in gestione tramite il menù situato in alto a destra.

## 2.2 Gruppi di configurazioni

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 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* 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

---

### Nota: 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 Fig. 2.6.

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 v1*”. Dopodichè accedere, dal dettaglio dell'erozione, alla sezione di configurazione dell'API cliccando sul pulsante *Configura* posto in basso a destra. La successiva maschera consente di configurare l'API e di visualizzare i gruppi in cui sono state classificate le risorse. Per default non è presente alcun raggruppamento (Fig. 2.7).

Selezionare il pulsante “*Crea Nuova*” e fornire i seguenti dati:

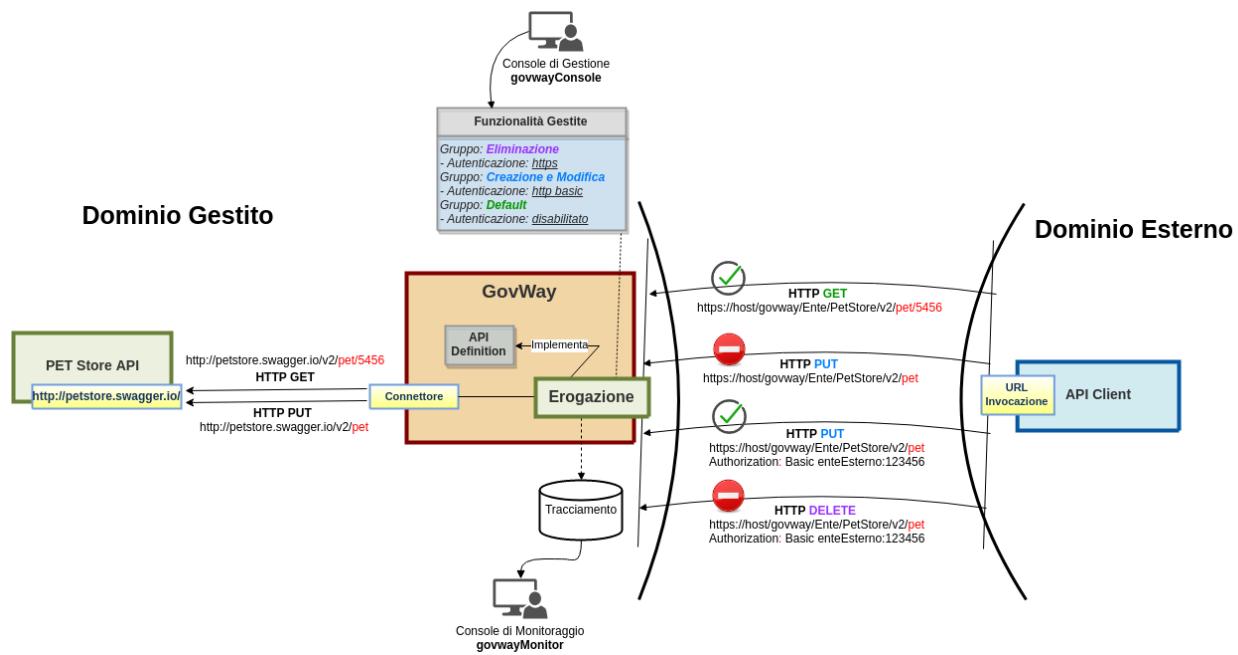


Fig. 2.6: Configurazioni differenti per gruppi di risorse di una API

Erogazioni > PetStore v1 (ENTE) > Configurazione

**Configurazione**

|                        |  |
|------------------------|--|
| Controllo Accessi      | <input checked="" type="checkbox"/> Disabilitato   |
| Rate Limiting          | <input checked="" type="checkbox"/> Disabilitato   |
| Validazione            | <input checked="" type="checkbox"/> Disabilitato   |
| Caching Risposta       | <input checked="" type="checkbox"/> Disabilitato   |
| Sicurezza Messaggio    | <input checked="" type="checkbox"/> Disabilitato   |
| Trasformazioni         | <input checked="" type="checkbox"/> Disabilitato   |
| Tracciamento           | <input checked="" type="checkbox"/> Transazioni <input checked="" type="checkbox"/> Diagnostic |
| Registrazione Messaggi | <input checked="" type="checkbox"/> Disabilitato   |

**CREA NUOVA**

Fig. 2.7: Situazione iniziale con unico gruppo “Predefinito”

- *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*.
- *Controllo degli Accessi - Accesso API*: per esporre l’API in modo che sia invocabile da client identificati tramite credenziali selezionare lo stato “*autenticato*”.

Erogazioni > PetStore v1 (ENTE) > Configurazione > Aggiungi

Note: (\*) Campi obbligatori

**Configurazione**

Nome Gruppo \* Creazione e Modifica

Risorse \*

- 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
- POST /store/order

Modalità Nuova

**Controllo degli Accessi**

Accesso API autenticato

**SALVA**

Fig. 2.8: Registrazione Gruppo “Creazione e Modifica”

Terminata la creazione, l’accesso alle risorse del gruppo “Creazione e Modifica”, richiede che il client presenti delle credenziali ssl come indicato nella sezione *Identificazione dei Mittenti* ( Fig. 2.9 ).

Per impostare una autenticazione “http-basic” accedere in modifica alla configurazione del Controllo degli Accessi indicando un’autenticazione “http-basic” e disabilitando l’autorizzazione come mostrato nella figura Fig. 2.10.

Una volta salvata la nuova configurazione per il Controllo degli Accessi, per accedere alle risorse associate al gruppo “Creazione e Modifiche” un client deve presentare delle credenziali http-basic ( Fig. 2.11 ) associate ad un soggetto o un applicativo registrato su GovWay. Al punto 7. verrà descritto come registrare un soggetto che possiede delle credenziali http-basic valide utilizzate in questo scenario di test.

| Nome Gruppo            | Creazione e Modifica   |                                     |
|------------------------|--|-------------------------------------|
| Elenco Risorse         | POST /pet, PUT /pet, POST /pet/{petId}, POST /pet/{petId}/uploadImage, POST /store/order, POST /user, POST /user/createWithArray, POST /user/create... | <input type="button" value="edit"/> |
| Controllo Accessi      | <input checked="" type="checkbox"/> Autenticazione Trasporto [ https ] <input type="checkbox"/> Autorizzazione [ Richiedente ]                         | <input type="button" value="edit"/> |
| Rate Limiting          | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value="edit"/> |
| Validazione            | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value="edit"/> |
| Caching Risposta       | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value="edit"/> |
| Sicurezza Messaggio    | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value="edit"/> |
| Trasformazioni         | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value="edit"/> |
| Tracciamento           | <input checked="" type="checkbox"/> Transazioni <input checked="" type="checkbox"/> Diagnostici  | <input type="button" value="edit"/> |
| Registrazione Messaggi | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value="edit"/> |

Fig. 2.9: Gruppo “Creazione e Modifica” configurato con autenticazione “https”

Erogazioni > PetStore v1 (ENTE) > Configurazione > **Controllo Accessi del gruppo 'Creazione e Modifica'**

### Controllo Accessi del gruppo 'Creazione e Modifica'

**Autenticazione Token**

Stato

**Autenticazione Trasporto**

Stato

Forward Authorization

Opzionale

**Autorizzazione**

Stato

**Autorizzazione Contenuti**

Stato

**SALVA**

Fig. 2.10: Gruppo “Creazione e Modifica”, Controllo degli Accessi configurazione con autenticazione “http-basic”

Erogazioni > PetStore v1 (ENTE) > Configurazione

## Configurazione

Creazione e Modifica Predefinito

|                        |  |                                 |                                 |
|------------------------|--|---------------------------------|---------------------------------|
| Nome Gruppo            | Creazione e Modifica   | <input type="button" value=""/> | <input type="button" value=""/> |
| Elenco Risorse         | POST /pet, PUT /pet, POST /pet/{petId}, POST /pet/{petId}/uploadImage, POST /store/order, POST /user, POST /user/createWithArray, POST /user/create... | <input type="button" value=""/> | <input type="button" value=""/> |
| Controllo Accessi      | <input checked="" type="checkbox"/> Autenticazione Trasporto [ http-basic ]  | <input type="button" value=""/> | <input type="button" value=""/> |
| Rate Limiting          | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value=""/> | <input type="button" value=""/> |
| Validazione            | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value=""/> | <input type="button" value=""/> |
| Caching Risposta       | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value=""/> | <input type="button" value=""/> |
| Sicurezza Messaggio    | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value=""/> | <input type="button" value=""/> |
| Trasformazioni         | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value=""/> | <input type="button" value=""/> |
| Tracciamento           | <input checked="" type="checkbox"/> Transazioni <input checked="" type="checkbox"/> Diagnostici  | <input type="button" value=""/> | <input type="button" value=""/> |
| Registrazione Messaggi | <input checked="" type="checkbox"/> Disabilitato   | <input type="button" value=""/> | <input type="button" value=""/> |

Fig. 2.11: Gruppo “Creazione e Modifica”, Controllo degli Accessi configurato con autenticazione “http-basic”

## 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*.
- *Controllo degli Accessi - Accesso API*: per esporre l’API in modo che sia invocabile da client identificati tramite credenziali selezionare lo stato “*autenticato*”.

Note: (\*) Campi obbligatori

**Configurazione**

Nome Gruppo \* Eliminazione

Risorse \* GET /pet/findByStatus  
GET /pet/findByTags  
**DELETE /pet/{petId}**  
GET /pet/{petId}  
GET /store/inventory  
**DELETE /store/order/{orderId}**  
GET /store/order/{orderId}  
GET /user/login  
GET /user/logout  
**DELETE /user/{username}**

Modalità Nuova

**Controllo degli Accessi**

Accesso API autenticato

**SALVA**

Fig. 2.12: Registrazione Gruppo “Eliminazione”

Come descritto precedentemente per il gruppo “Creazione e Modifica” modificare la configurazione relativa al Controllo degli Accessi per impostare un’autenticazione “http-basic”.

## 3. Verifica Gruppi Esistenti

Dal dettaglio dell’erogazione, accedendo alla sezione di configurazione dell’API cliccando sul pulsante *Configura* posto in basso a destra, è possibile visualizzati tre gruppi, i due gruppi creati in precedenza ed il gruppo predefinito che adesso contiene solamente le risorse con metodo http GET (Fig. 2.13).

Nella sezione di configurazione sarà possibile agire sui gruppi anche in un secondo momento aggiungendo o eliminando risorse da un gruppo o creandone di nuovi. Inoltre sarà possibile configurare per ogni gruppo le funzionalità disponibili con Govway quali Validazione dei Contenuti, Rate Limiting, Trasformazioni etc...

Erogazioni > PetStore v1 (ENTE) > Configurazione

## Configurazione

Creazione e Modifica   Eliminazione   Predefinito

| Nome Gruppo            | Predefinito  |  |
|------------------------|--|--|
| Elenco Risorse         | GET /pet/findByStatus, GET /pet/findByTags, GET /pet/{petId}, GET /store/inventory, GET /store/order/{orderId}, GET /user/login, GET /user/logout, ... |  |
| Controllo Accessi      | Disabilitato   |  |
| Rate Limiting          | Disabilitato   |  |
| Validazione            | Disabilitato   |  |
| Caching Risposta       | Disabilitato   |  |
| Sicurezza Messaggio    | Disabilitato   |  |
| Trasformazioni         | Disabilitato   |  |
| Tracciamento           | Transazioni   Diagnostici  |  |
| Registrazione Messaggi | Disabilitato   |  |

ELIMINA   CREA NUOVA

Fig. 2.13: Gruppi Registrati

Si può notare come i due gruppi creati per l'esempio possiedano un *Controllo Accessi* abilitato ( Fig. 2.11 ), mentre il gruppo *Predefinito* che contiene solo le risorse GET possiede tale funzionalità disabilitata ( Fig. 2.13 ).

#### 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”. (Fig. 2.14).

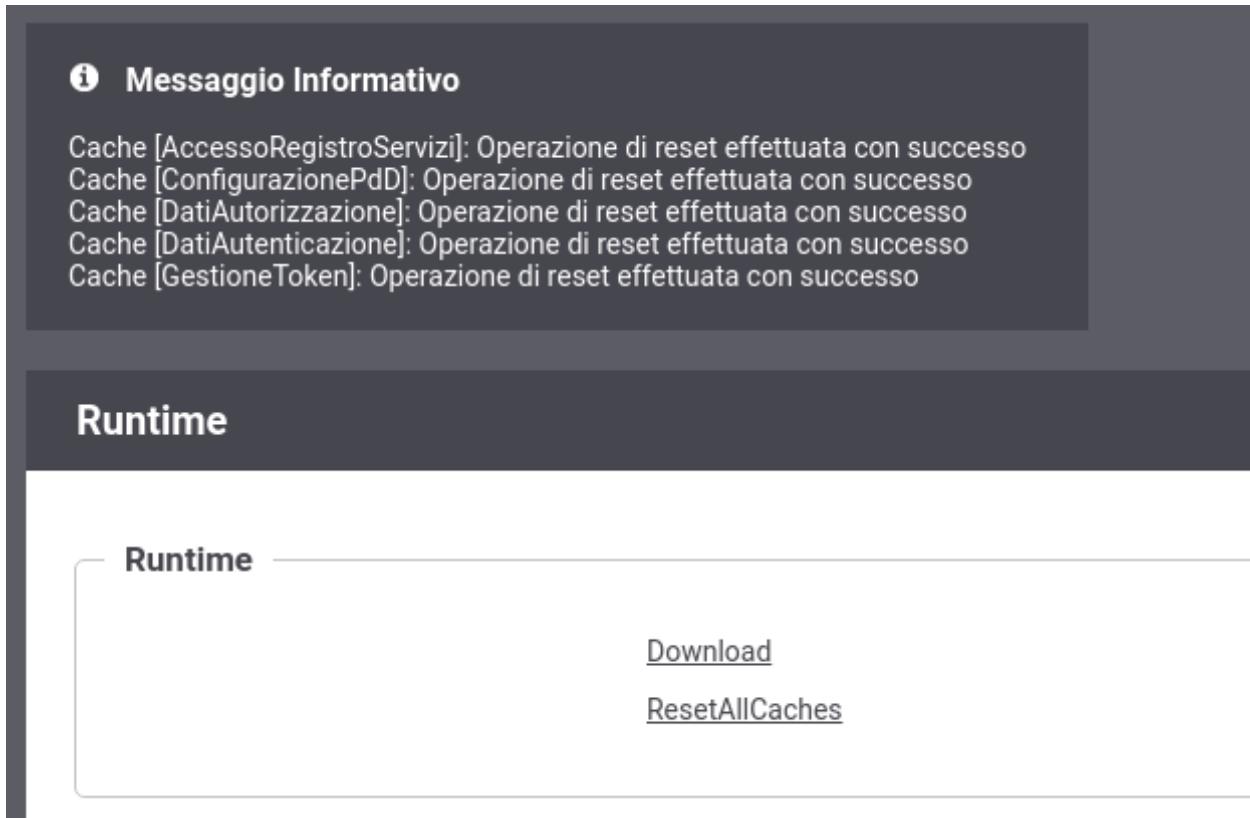


Fig. 2.14: Reset Cache delle Configurazioni di GovWay

#### 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/v1/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
```

(continues on next page)

(continua dalla pagina precedente)

```
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/v1/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 Fig. 2.15 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 (Fig. 2.16).

|                          | Data Ingresso Richiesta            | Tipologia  | Esito                  | Fruitore | Erogatore | API         | Azione        |
|--------------------------|------------------------------------|------------|------------------------|----------|-----------|-------------|---------------|
| <input type="checkbox"/> | <span>⚠ 2018-11-15 17:07:10</span> | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet       |
| <input type="checkbox"/> | <span>✓ 2018-11-15 17:03:43</span> | Erogazione | Ok                     |          | Ente      | PetStore v2 | GET_pet.petId |
| <input type="checkbox"/> | <span>⚠ 2018-11-15 17:03:09</span> | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet       |

Fig. 2.15: Tracce delle invocazioni transitate sul Gateway

| Storico > Intervallo Temporale > Dettagli Transazione > Messaggi Diagnostici |                  |                |   |
|--|------------------|----------------|---|
| Lista Diagnostici: record [1 - 5] su 5                                       |                  |                |   |
| Data   | Severita         | 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 [dbbd9cd8-9711-47b9-afce-2ab05e93b3df]  |
| 2018-11-15 17:07:10.921  | infoIntegration  | RicezioneBuste | Risposta<br>({"type":"https://httpstatuses.com/401","title":"Unauthorized","status":401,"detail":"Autenticazione fallita, credenziali non fornite","govway_status":401,"protocol:GOVWAY-109}) consegnata al mittente con codice di trasporto: 401 |

Fig. 2.16: 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.

Accedere al soggetto *EnteEsterno* creato in precedenza durante l’esempio descritto nella sezione *Fruizione API* e associargli delle credenziali “http basic” come ad esempio un username *enteEsterno* ed una password *123456* (Fig. 2.17).

Dopo aver associato le credenziali al soggetto effettuare il reset della cache delle configurazioni del Gateway come descritto in precedenza prima di procere 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/v1/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*:

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**

Fig. 2.17: Registrazione Gruppo “Eliminazione”

```
curl -v -X DELETE "http://127.0.0.1:8080/govway/Ente/PetStore/v1/pet/545646489" --  
-u 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 Fig. 2.18 si può vedere come le transazioni con metodo http *DEL* sono terminate con errore con esito *Autenticazione Fallita*.

| Lista Transazioni: record [1 - 6] |  |            |                        |             |           |             |                  |
|-----------------------------------|--|------------|------------------------|-------------|-----------|-------------|------------------|
|                                   | Data Ingresso Richiesta                                  | Tipologia  | Esito                  | Fruitore    | Erogatore | API         | Azione           |
| <input type="checkbox"/>          | <span style="color: red;">!</span> 2018-11-16 10:22:09   | Erogazione | Autenticazione Fallita |             | Ente      | PetStore v2 | DELETE_pet.petId |
| <input type="checkbox"/>          | <span style="color: red;">!</span> 2018-11-16 10:21:02   | Erogazione | Autenticazione Fallita |             | Ente      | PetStore v2 | DELETE_pet.petId |
| <input type="checkbox"/>          | <span style="color: green;">✓</span> 2018-11-16 10:20:56 | Erogazione | Ok                     | EnteEsterno | Ente      | PetStore v2 | PUT_pet          |
| <input type="checkbox"/>          | <span style="color: red;">!</span> 2018-11-16 10:20:44   | Erogazione | Autenticazione Fallita |             | Ente      | PetStore v2 | PUT_pet          |
| <input type="checkbox"/>          | <span style="color: green;">✓</span> 2018-11-16 10:18:41 | Erogazione | Ok                     |             | Ente      | PetStore v2 | GET_pet.petId    |
| <input type="checkbox"/>          | <span style="color: green;">✓</span> 2018-11-16 10:18:32 | Erogazione | Ok                     | EnteEsterno | Ente      | PetStore v2 | PUT_pet          |

Fig. 2.18: Tracce delle invocazioni transitate sul Gateway

### Nota: Ripristino Erogazione API con il solo gruppo predefinito per prosieguo degli scenari

Negli 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*”.

## 2.3 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 Fig. 2.19.

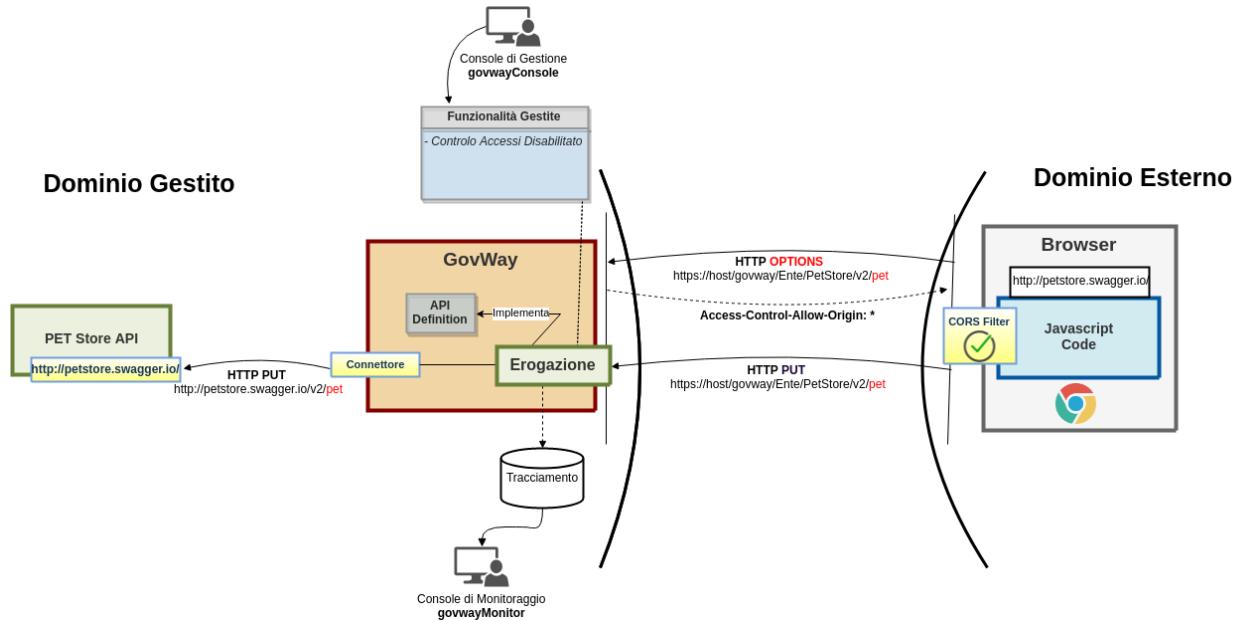


Fig. 2.19: Scenario cross-origin HTTP request (CORS)

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 Fig. 2.20 mostra la configurazione globale attiva per default.

Tramite il tool on-line disponibile all'indirizzo <https://www.test-cors.org/> è possibile verificare il funzionamento dello scenario descritto nella Fig. 2.19. Configurare il tool con i seguenti parametri per utilizzare il servizio descritto nella sezione *Erogazione API REST*:

- *HTTP Method*: PUT
- *Request Headers*:
  - accept: application/json
  - Content-Type: application/json
- *Request Content*:

**Gestione CORS**

|                       |   |
|-----------------------|---|
| Stato                 | abilitato                                   |
| Tipo                  | Gestito dal Gateway                         |
| <b>Access Control</b> |   |
| All Allow Origins     | <input checked="" type="checkbox"/>         |
| Allow Headers *       | Authorization x Content-Type x SOAPAction x |
| Allow Methods *       | GET x PUT x POST x DELETE x PATCH x         |
| Allow Credentials     | <input type="checkbox"/>                    |

Fig. 2.20: CORS - Configurazione di default

```
{
  "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/v1/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 Fig. 2.21 è possibile vedere come siano state effettuate due richieste http di cui la prima è stata iniziata dal browser (Initiator: corsclient.js).

La Fig. 2.22 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 “\*”.

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.

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.

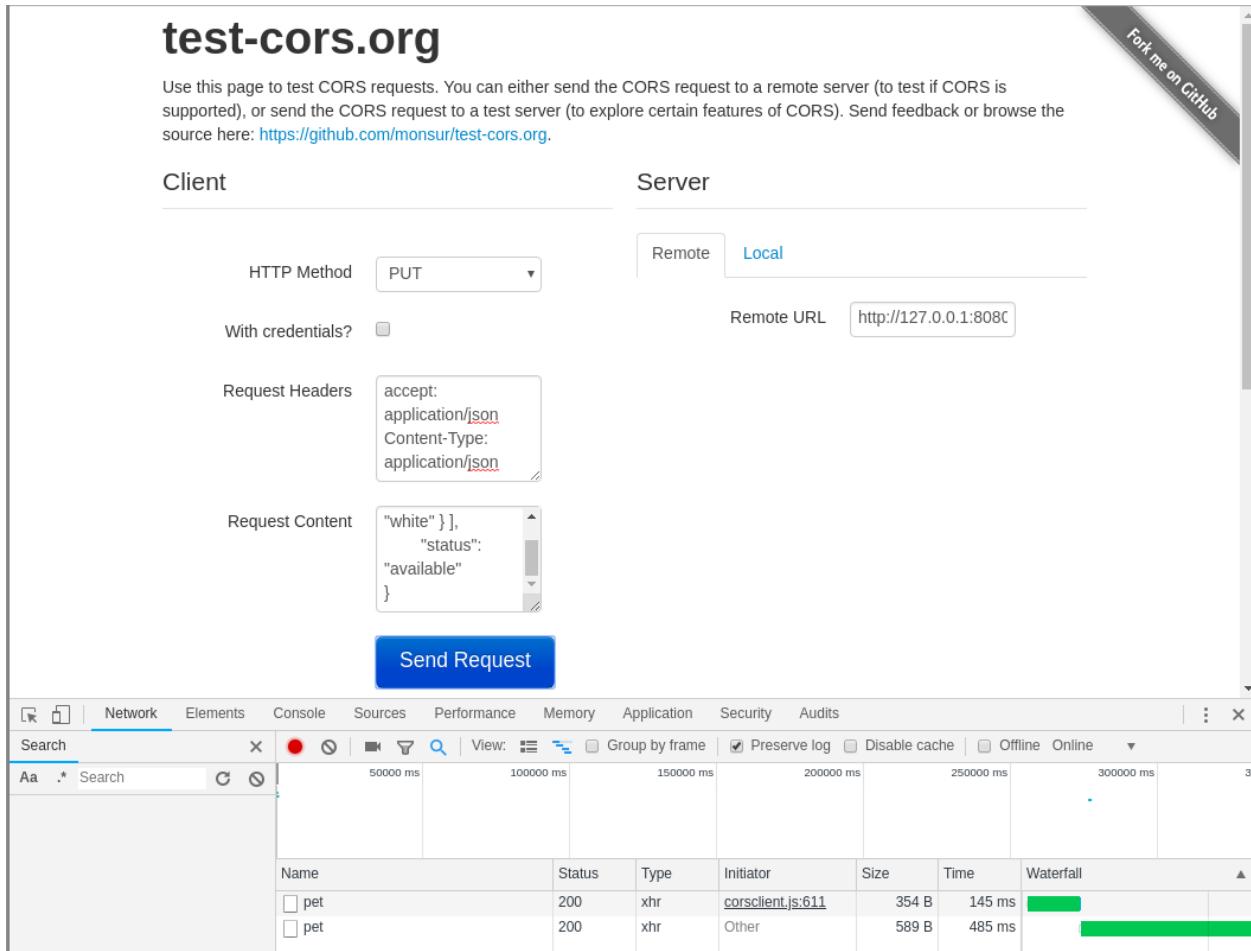


Fig. 2.21: Verifica CORS

| Name                         | Headers   | Preview | Response | Timing |
|------------------------------|---|---------|----------|--------|
| <input type="checkbox"/> pet |   |         |          |        |
| <input type="checkbox"/> pet | <p>▼ General</p> <p>Request URL: <a href="http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet">http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet</a></p> <p>Request Method: OPTIONS</p> <p>Status Code: <span style="color: green;">200</span> OK</p> <p>Remote Address: 127.0.0.1:8080</p> <p>Referrer Policy: no-referrer-when-downgrade</p>  |         |          |        |
|                              | <p>▼ Response Headers</p> <p>Access-Control-Allow-Headers: Authorization, Content-Type, SOAPAction</p> <p>Access-Control-Allow-Methods: GET, PUT, POST, DELETE, PATCH</p> <p>Access-Control-Allow-Origin: *</p> <p>Connection: keep-alive</p> <p>Date: Mon, 03 Dec 2018 10:02:28 GMT</p> <p>GovWay-Transaction-ID: 5da1832c-3708-4947-85dc-e585e58dc446</p> <p>Server: GovWay</p> <p>Transfer-Encoding: chunked</p> |         |          |        |
|                              | <p>▼ Request Headers</p> <p>⚠ Provisional headers are shown</p> <p>Access-Control-Request-Headers: content-type</p> <p>Access-Control-Request-Method: PUT</p> <p>Origin: <a href="https://www.test-cors.org">https://www.test-cors.org</a></p> <p>User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ch</p>   |         |          |        |

Fig. 2.22: Verifica CORS: richiesta OPTIONS

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.

---

**Nota: 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”.

---

## 2.4 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.

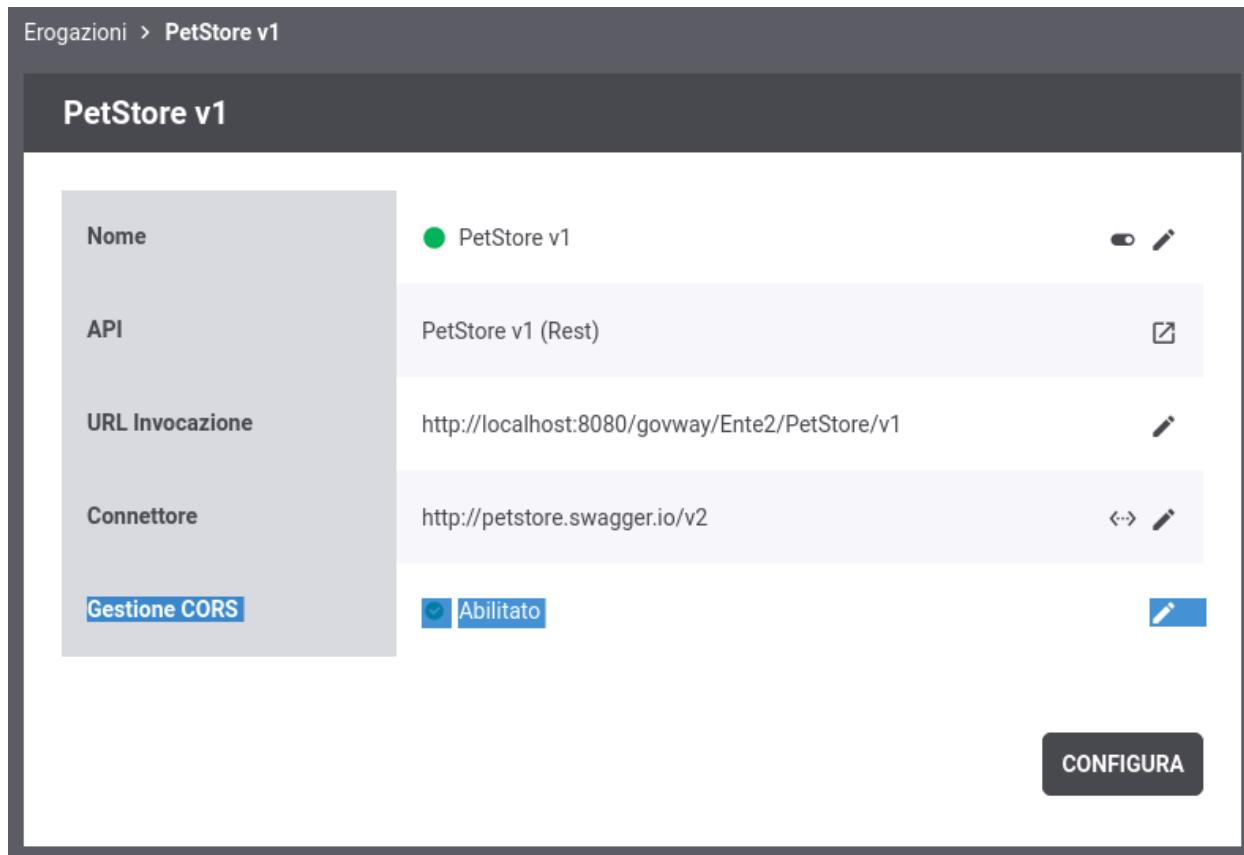


Fig. 2.23: Personalizzazione Gestione CORS di una erogazione

Erogazioni > PetStore v2 (Ente) > Gestione CORS

## Gestione CORS

Note: (\*) Campi obbligatori

**Gestione CORS**

|       |                     |
|-------|---------------------|
| Stato | ridefinito          |
|       | abilitato           |
| Tipo  | Gestito dal Gateway |

**Access Control**

All Allow Origins

Allow Origins \*

Allow Headers \*

Allow Methods \*

Allow Credentials

**SALVA**

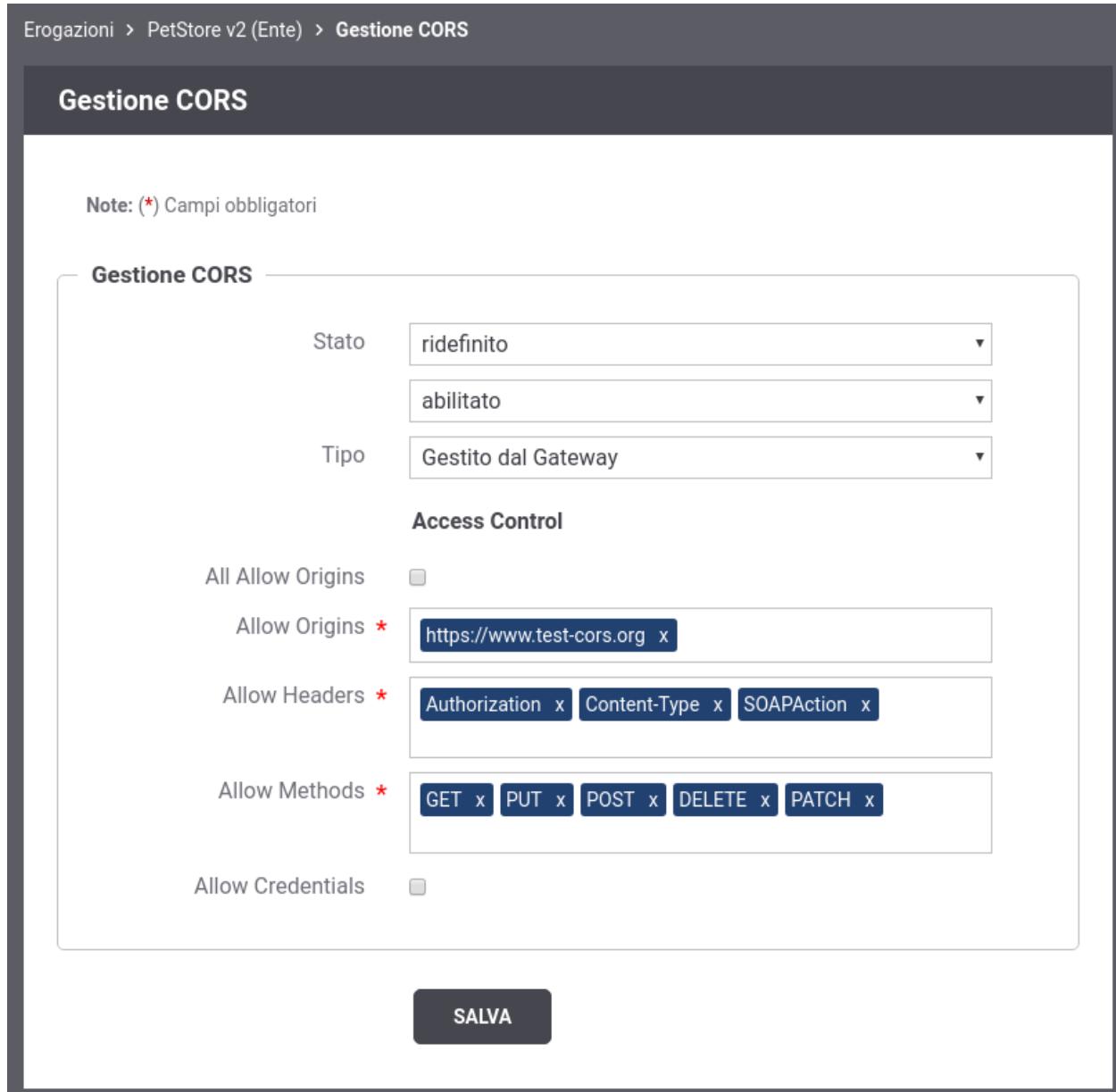


Fig. 2.24: Personalizzazione Gestione CORS: definizione di uno specifico “origin”

Name

pet

pet

Headers Preview Response Timing

General

Request URL: <http://127.0.0.1:8080/govway/Ente/PetStore/v2/pet>

Request Method: OPTIONS

Status Code: 200 OK

Remote Address: 127.0.0.1:8080

Referrer Policy: no-referrer-when-downgrade

Response Headers

Access-Control-Allow-Headers: Authorization, Content-Type, SOAPAction

Access-Control-Allow-Methods: GET, PUT, POST, DELETE, PATCH

Access-Control-Allow-Origin: <https://www.test-cors.org>

Connection: keep-alive

Date: Mon, 03 Dec 2018 10:42:34 GMT

GovWay-Transaction-ID: dfc8185a-f641-4bb0-a7b0-94c31b9d867f

Server: GovWay

Transfer-Encoding: chunked

Vary: Origin

Request Headers

⚠ Provisional headers are shown

Access-Control-Request-Headers: content-type

Access-Control-Request-Method: PUT

Origin: <https://www.test-cors.org>

Fig. 2.25: Verifica CORS: definizione di uno specifico “origin”

Una rappresentazione di questo scenario è mostrata nella Fig. 2.26.

Per sospendere una erogazione o fruizione di API, utilizzando la console *govwayConsole* dal dettaglio dell’erogazione o della fruizione cliccare sul “toggle” di abilitazione/disabilitazione presente nella prima riga dove viene presentato il nome dell’API. Comparirà una finestra di dialogo dove viene richiesto di confermare la sospensione. La Fig. 2.27 mostra una sospensione in corso dell’erogazione registrata nella sezione *Erogazione API REST*.

Procedendo con la conferma l’erogazione sarà a tutti gli effetti sospesa come mostra anche l’icona di stato rossa (Fig. 2.28).

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 Fig. 2.29.

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/v1/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"
}'
```

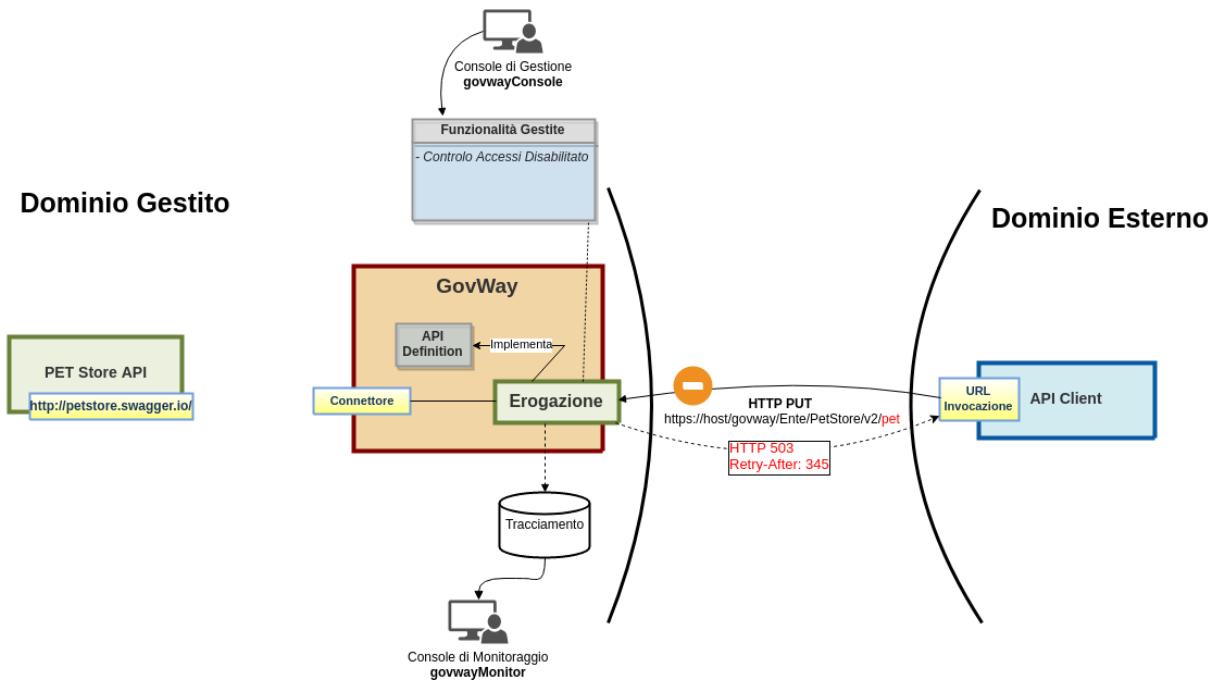


Fig. 2.26: Sospensione di una API

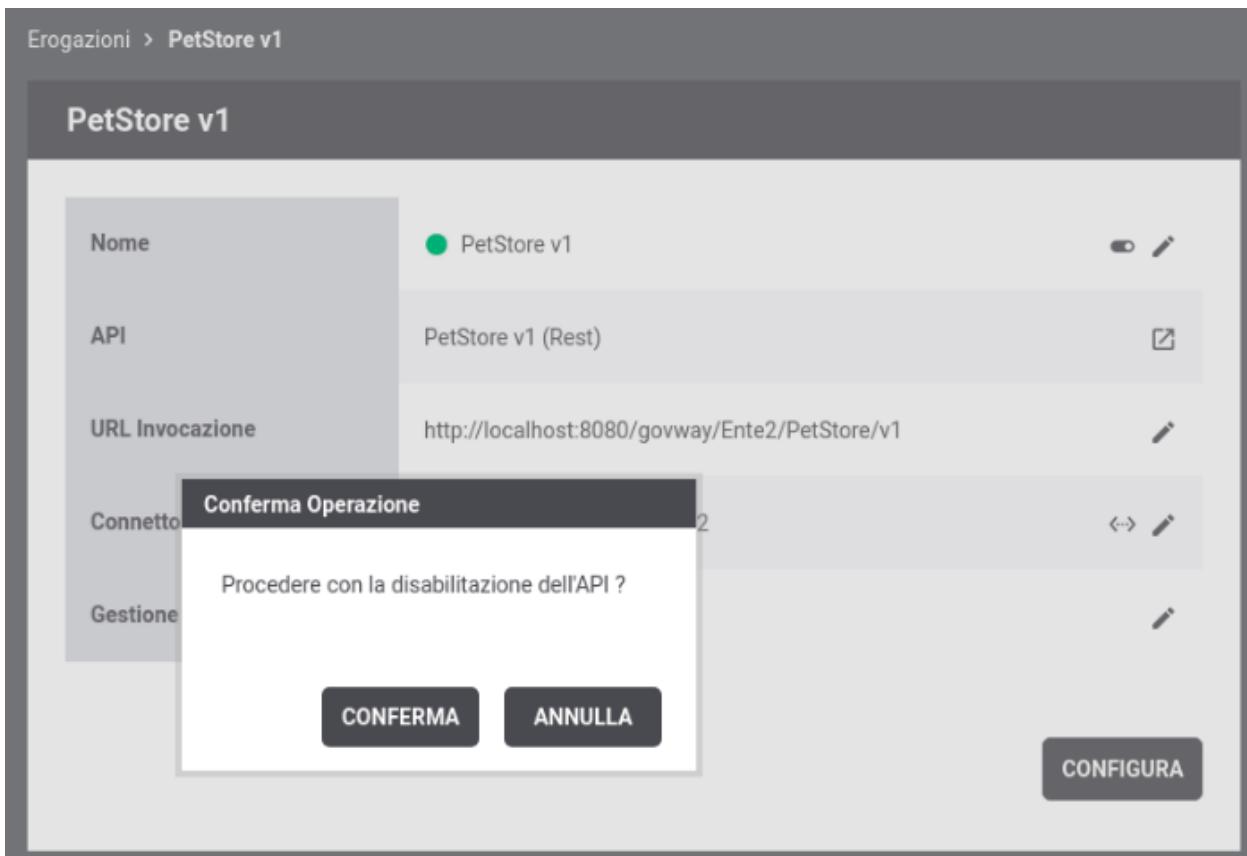


Fig. 2.27: Sospensione di una erogazione

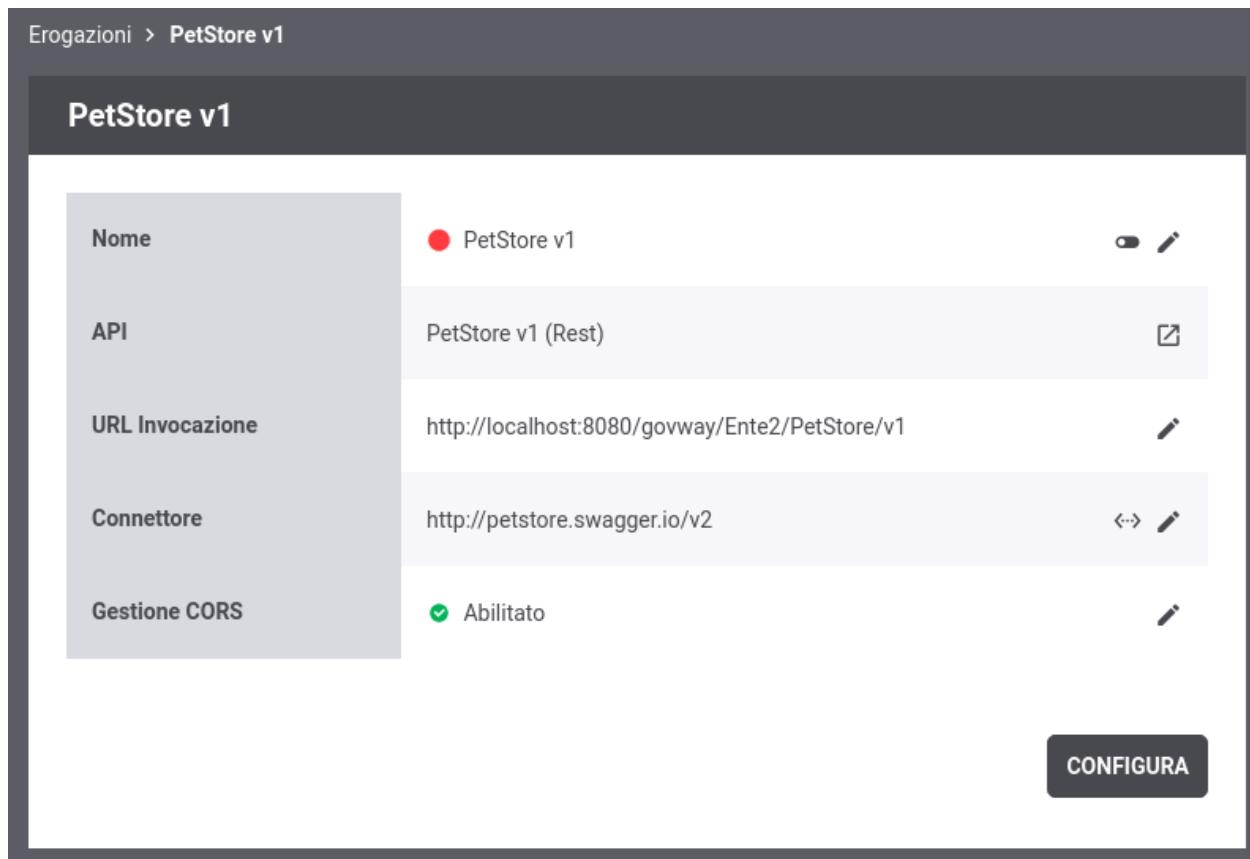


Fig. 2.28: Erogazione sospesa

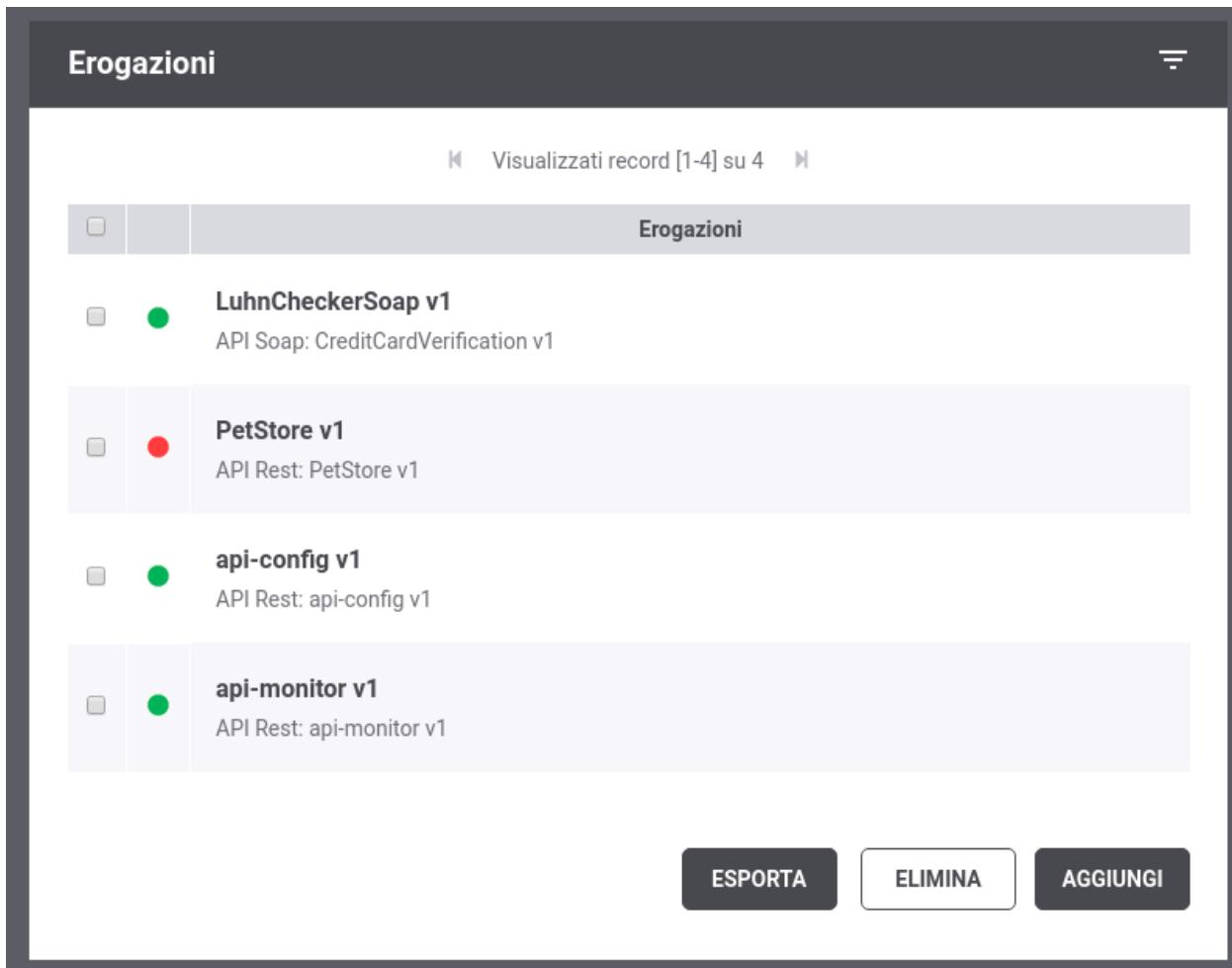


Fig. 2.29: Stato disabilitato riportato nell'elenco delle erogazioni

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 Fig. 2.30 si può vedere come le transazioni generate dopo la sospensione sono terminate con esito *API Sospesa*.

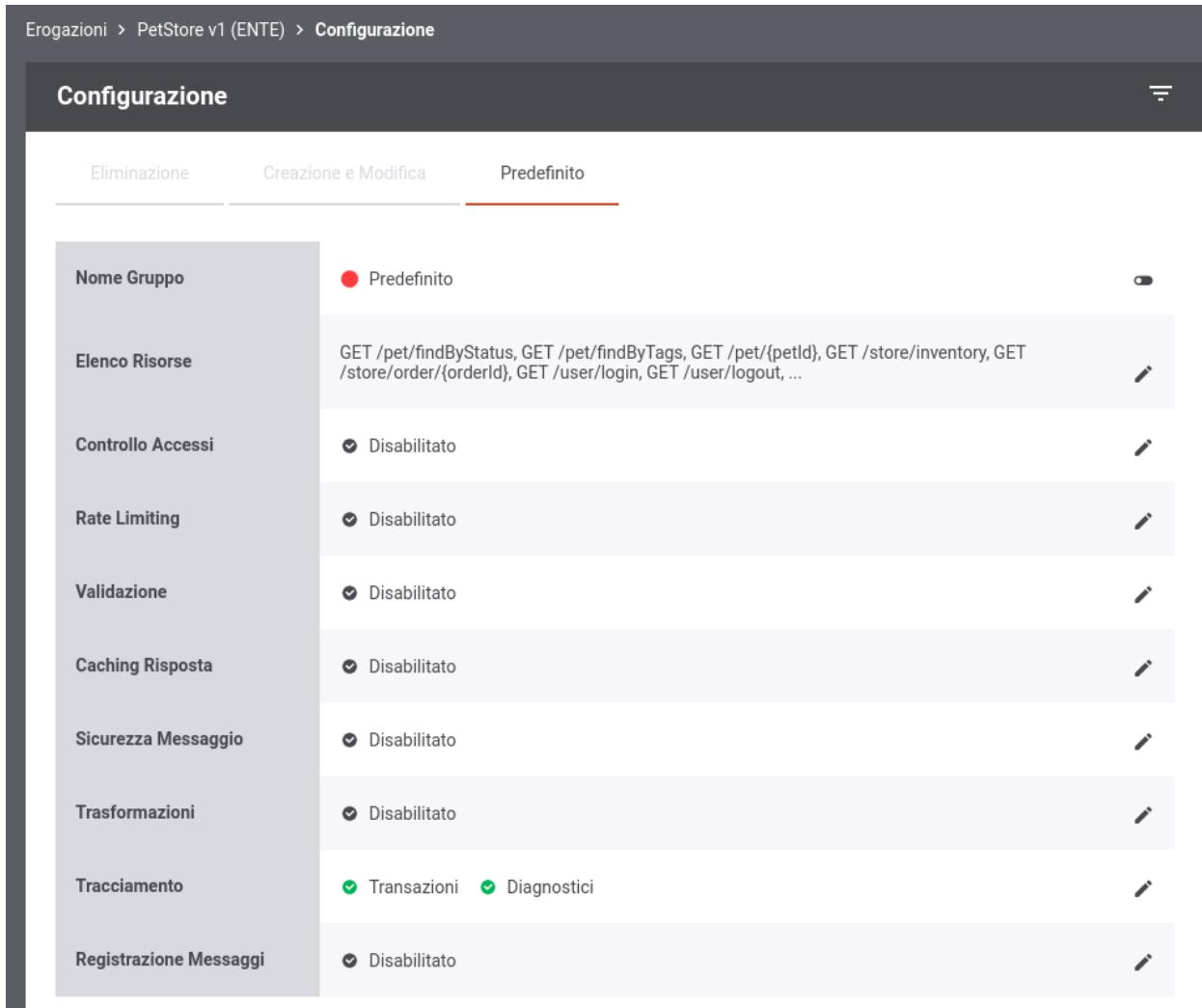
| Lista Fruizioni: record [1 - 2] |                                  |                          |                     |                    |            |          |                |                  |             |
|---------------------------------|----------------------------------|--------------------------|---------------------|--------------------|------------|----------|----------------|------------------|-------------|
|                                 |                                  |                          | Data Richiesta      | API                | Operazione | Mittente | Latenza Totale | Latenza Servizio | Esito       |
| <input type="checkbox"/>        | <input checked="" type="radio"/> | <input type="checkbox"/> | 2019-10-04 19:04:56 | PetStore v1 (ENTE) | PUT_pet    |          | 6 ms           | N.D.             | API Sospesa |
| <input type="checkbox"/>        | <input checked="" type="radio"/> | <input type="checkbox"/> | 2019-10-04 19:04:53 | PetStore v1 (ENTE) | PUT_pet    |          | 73 ms          | N.D.             | API Sospesa |

Fig. 2.30: 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 *Gruppi di configurazioni*, la sospensione può essere effettuata sul singolo gruppo.

La Fig. 2.31 mostra un esempio di sospensione, nello scenario sezione *Gruppi di configurazioni*, del solo gruppo *“Predefinito”*.

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 Fig. 2.32.



Erogazioni > PetStore v1 (ENTE) > Configurazione

## Configurazione

Predefinito

| Nome Gruppo            | Predefinito  |
|------------------------|--|
| Elenco Risorse         | GET /pet/findByStatus, GET /pet/findByTags, GET /pet/{petId}, GET /store/inventory, GET /store/order/{orderId}, GET /user/login, GET /user/logout, ... |
| Controllo Accessi      | Disabilitato   |
| Rate Limiting          | Disabilitato   |
| Validazione            | Disabilitato   |
| Caching Risposta       | Disabilitato   |
| Sicurezza Messaggio    | Disabilitato   |
| Trasformazioni         | Disabilitato   |
| Tracciamento           | Transazioni, Diagnostici   |
| Registrazione Messaggi | Disabilitato   |

Fig. 2.31: Gruppo di una erogazione sospeso

**Erogazioni**

Visualizzati record [1-4] su 4

|                          |   | Erogazioni   |
|--------------------------|---|--|
| <input type="checkbox"/> | ● | <b>LuhnCheckerSoap v1</b><br>API Soap: CreditCardVerification v1 |
| <input type="checkbox"/> | ● | <b>PetStore v1</b><br>API Rest: PetStore v1                      |
| <input type="checkbox"/> | ● | <b>api-config v1</b><br>API Rest: api-config v1                  |
| <input type="checkbox"/> | ● | <b>api-monitor v1</b><br>API Rest: api-monitor v1                |

ESPORTA    ELIMINA    AGGIUNGI

Fig. 2.32: Stato disabilitato di un gruppo riportato nell'elenco delle erogazioni



# CAPITOLO 3

---

## 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 Fig. 3.1 mette in evidenza tutte le comunicazioni e gli attori coinvolti per riuscire a porta a termine l'invocazione dello scenario descritto nella sezione ? 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 [Erogazione API REST](#). 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.

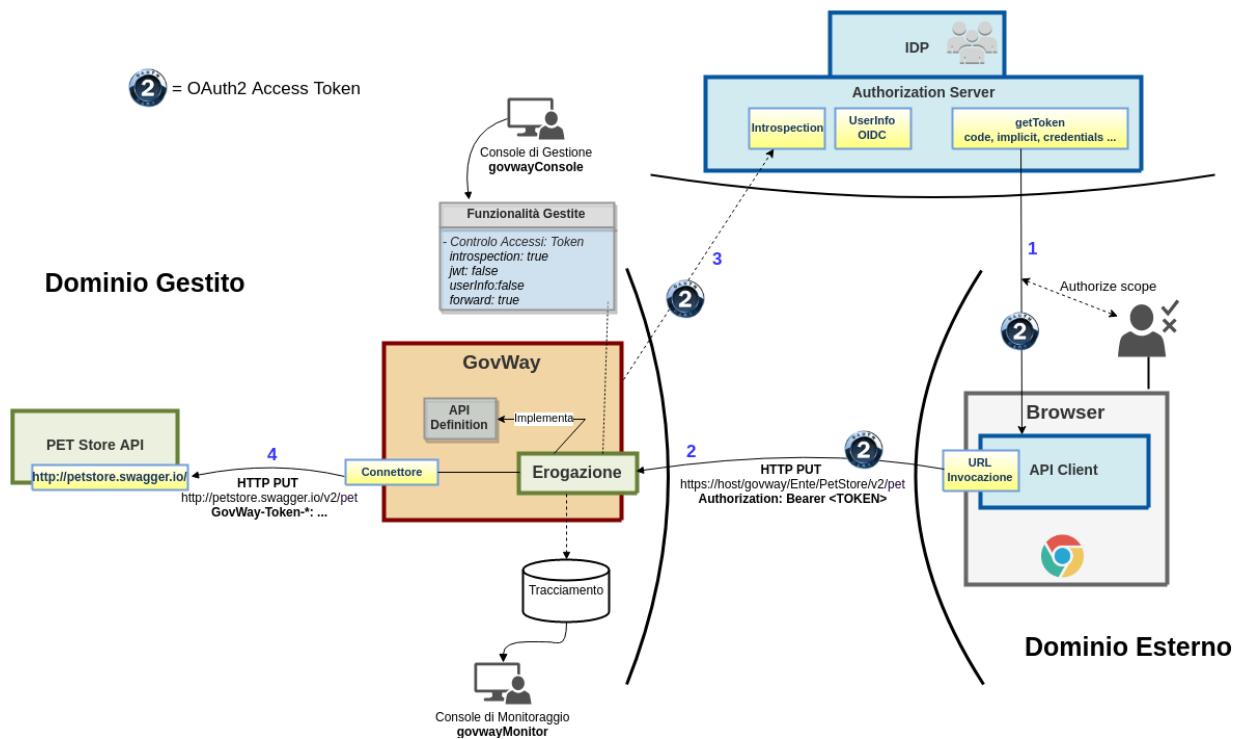


Fig. 3.1: Scenario OAuth

Come si evince dalla Fig. 3.1 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.

**Nota: Requisito account gmail**

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

### 3.1 Validazione tramite Introspection

In questa sezione viene descritto come realizzare lo scenario raffigurato nella Fig. 3.1 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 v1”. 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”.

- **Invocazione API senza un access token**

---

**Nota: 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 Fig. 3.3 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con errore con esito *Gestione Token Fallita*.

Erogazioni > PetStore v1 (ENTE) > Configurazione > **Controllo Accessi**

### Controllo Accessi

Note: (\*) Campi obbligatori

**Autenticazione Token**

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

**Required Claims**

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

**Autenticazione Trasporto**

|       |              |
|-------|--------------|
| Stato | disabilitato |
|-------|--------------|

Fig. 3.2: Configurazione OAuth2 per PetStore

|                          | Data Ingresso Richiesta                              | Tipologia   | Esito                  | Fruitore | Erogatore | API         | Azione  |
|--------------------------|--|-------------|------------------------|----------|-----------|-------------|---------|
| <input type="checkbox"/> | <input checked="" type="radio"/> 2018-12-04 12:13:37 | Erogazio... | Gestione Token Fallita |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/> | <input checked="" type="radio"/> 2018-12-04 12:13:36 | Erogazio... | Gestione Token Fallita |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/> | <input checked="" type="radio"/> 2018-12-04 12:08:37 | Erogazio... | Gestione Token Fallita |          | Ente      | PetStore v2 | PUT_pet |

Fig. 3.3: 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.

##### 2. Authorization Code

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

##### 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.

#### • Invocazione API con un access token

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

#### Nota: 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"
}'
```

The screenshot shows the Google Developers OAuth 2.0 Playground interface. At the top, there is a header with the Google Developers logo and the text "OAuth 2.0 Playground" in red, with a close button "X" to its right. Below the header, there is a section titled "Step 1 Select & authorize APIs" with a dropdown arrow. A sub-instruction reads: "Select the scope for the APIs you would like to access or input your own OAuth scopes below. Then click the "Authorize APIs" button." To the right of this section, the text "Request / Response" is displayed in red, followed by "No request." Below the main instruction, a list of API scopes is shown, each preceded by a small icon and a descriptive name. Some scopes have URLs listed under them. At the bottom of the list, there is a "Authorize APIs" button. A text input field at the bottom left is labeled "Input your own scopes".

| Scope  | Description   |
|--|---|
| Google Cloud Memorystore for Redis API v1                |   |
| Google Compute Engine Instance Group Updater API v1beta1 |   |
| Google Identity Toolkit API v3                           |   |
| Google Mirror API v1                                     |   |
| Google OAuth2 API v2                                     | <ul style="list-style-type: none"><li><a href="https://www.googleapis.com/auth/plus.login">https://www.googleapis.com/auth/plus.login</a></li><li><a href="https://www.googleapis.com/auth/plus.me">https://www.googleapis.com/auth/plus.me</a></li><li><a href="https://www.googleapis.com/auth/userinfo.email">https://www.googleapis.com/auth/userinfo.email</a></li><li><a href="https://www.googleapis.com/auth/userinfo.profile">https://www.googleapis.com/auth/userinfo.profile</a></li></ul> |
| Google Play Custom App Publishing API v1                 |   |
| Google Play Developer API v3                             |   |
| Google Play EMM API v1                                   |   |
| Google Play Game Services API v1                         |   |
| Google Play Game Services Management API v1management    |   |
| Google Play Game Services Publishing API v1configuration |   |
| Google Sheets API v4                                     |   |
| Google Site Verification API v1                          |   |
| Google Slides API v1                                     |   |
| Google+ API v1   |   |

Fig. 3.4: Ottenimento Token: Playground Google, Step 1

OAuth 2.0 Playground

Step 1 Select & authorize APIs

Step 2 Exchange authorization code for tokens

Once you got the Authorization Code from Step 1 click the **Exchange authorization code for tokens** button, you will get a refresh and an access token which is required to access OAuth protected resources.

Authorization code: 4/qQBy\_4oBfS\_xpe9VPpS63fqFkevLJzAs

Exchange authorization code for tokens

Refresh token: 1/d5PhXioSyaQmV-Nw5gllMfn82pEs

Access token: ya29.GltobRb4Ro\_2MeeQY3J37hm Refresh access token

Auto-refresh the token before it expires.

The access token will expire in **3565** seconds.

Note: The OAuth Playground will automatically revoke refresh tokens after 24h. You can avoid this by specifying your own application OAuth credentials using the Configuration panel.

Request / Response

HTTP/1.1 302 Found  
Location: <https://accounts.google.com/o/oauth2/v2/auth?re>

GET /oauthplayground/?code=4/qQBy\_4oBfS\_xpe9VPpS63fqFkevL  
Host: developers.google.com

Fig. 3.5: Ottenimento Token: Playground Google, Step 2

### 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%2FqQCTBFrJConLp2WBcKP40w0JqeAgPj56QlAuiKyn4Dz4dY9epFi7nfln-pxgyx0UkXlhxp_SC7rcQdqp8bZnE8&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: ESF
Content-Encoding: gzip
Cache-Control: private
Date: Tue, 04 Dec 2018 10:57:27 GMT
X-Frame-Options: SAMEORIGIN
Alt-Svc: quic=":443"; ma=2592000; v="44,43,39,35"
Content-Type: application/json; charset=utf-8

{
  "access_token": "ya29.GltobRb4Ro_2MeeQY3J37hm",
  "id_token": "eyJhbGciOiJSUzIiNiIsImtpZCI6IjQ2M2ZlNDgwYzNjNTgzOWJiYjE1ODYxZTA4YzMyZDE4N2ZhZjhNTYiLCJ0eXAiOiI",
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope": "https://www.googleapis.com/auth/plus.me",
  "refresh_token": "1/d5PhXioSyaQmV-Nw5gllMfn82pESCgIu3u0f7_ULYR0"
}
```

Fig. 3.6: Ottenimento Token: Playground Google, Step 3

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<br>access_token=ya29.Glt0BjchXoKagIEFXIOUxsN1UVVUW1ryp...kT2laD8ERHY<br>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 | <a href="#">Visualizza</a>              |

Fig. 3.7: 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.

- **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.

```

1  {
2    "valid" : true,
3    "sub" : "106235657592654397689",
4    "aud" : [ "407408718192.apps.googleusercontent.com" ],
5    "exp" : 1543925775000,
6    "clientId" : "407408718192.apps.googleusercontent.com",
7    "scopes" : [ "https://www.googleapis.com/auth/plus.me" ],
8    "userInfo" : { },
9    "claims" : {
10      "aud" : "407408718192.apps.googleusercontent.com",
11      "sub" : "106235657592654397689",
12      "access_type" : "offline",
13      "azp" : "407408718192.apps.googleusercontent.com",
14      "scope" : "https://www.googleapis.com/auth/plus.me",
15      "exp" : "1543925775",
16      "expires_in" : "3566"
17    },
18    "rawResponse" : "{\n      \"azp\": \"407408718192.apps.googleusercontent.com\", \"aud\":\n      \"407408718192.apps.googleusercontent.com\", \"sub\": \"106235657592654397689\", \"\n      \"scope\": \"https://www.googleapis.com/auth/plus.me\", \"exp\": \"1543925775\", \"\n      \"expires_in\": \"3566\", \"access_type\": \"offline\"\n    }",
19    "sourceType" : "INTROSPECTION"
20  }

```

DOWNLOAD

Fig. 3.8: Informazioni ottenute tramite Introspection del Token

```

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,
}

```

(continues on next page)

(continua dalla pagina precedente)

```

    "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 Fig. 3.2). 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 *Token Forward*.

Per vedere quali header vengono effettivamente prodotti possiamo utilizzare la funzionalità “*Registrazione Messaggi*”. Accedere alla sezione “*Erogazioni*” e selezionare l’API precedentemente registrata “*PetStore v1*”. 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*”.

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*” (Fig. 3.10) che permette di vedere gli header http prodotti da GovWay (Fig. 3.11).

## 3.2 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](#)).

Per simulare lo scenario utilizzeremo sempre il servizio *Playground* e l’*Authorization Server di Google* descritto nella precedente sezione *Validazione tramite Introspection* 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](#)).

### Nota: 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 v1*”. Dopodichè accedere, dal dettaglio dell’erogazione, alla sezione “*Configurazione*” dove vengono visualizzate le funzionalità

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**

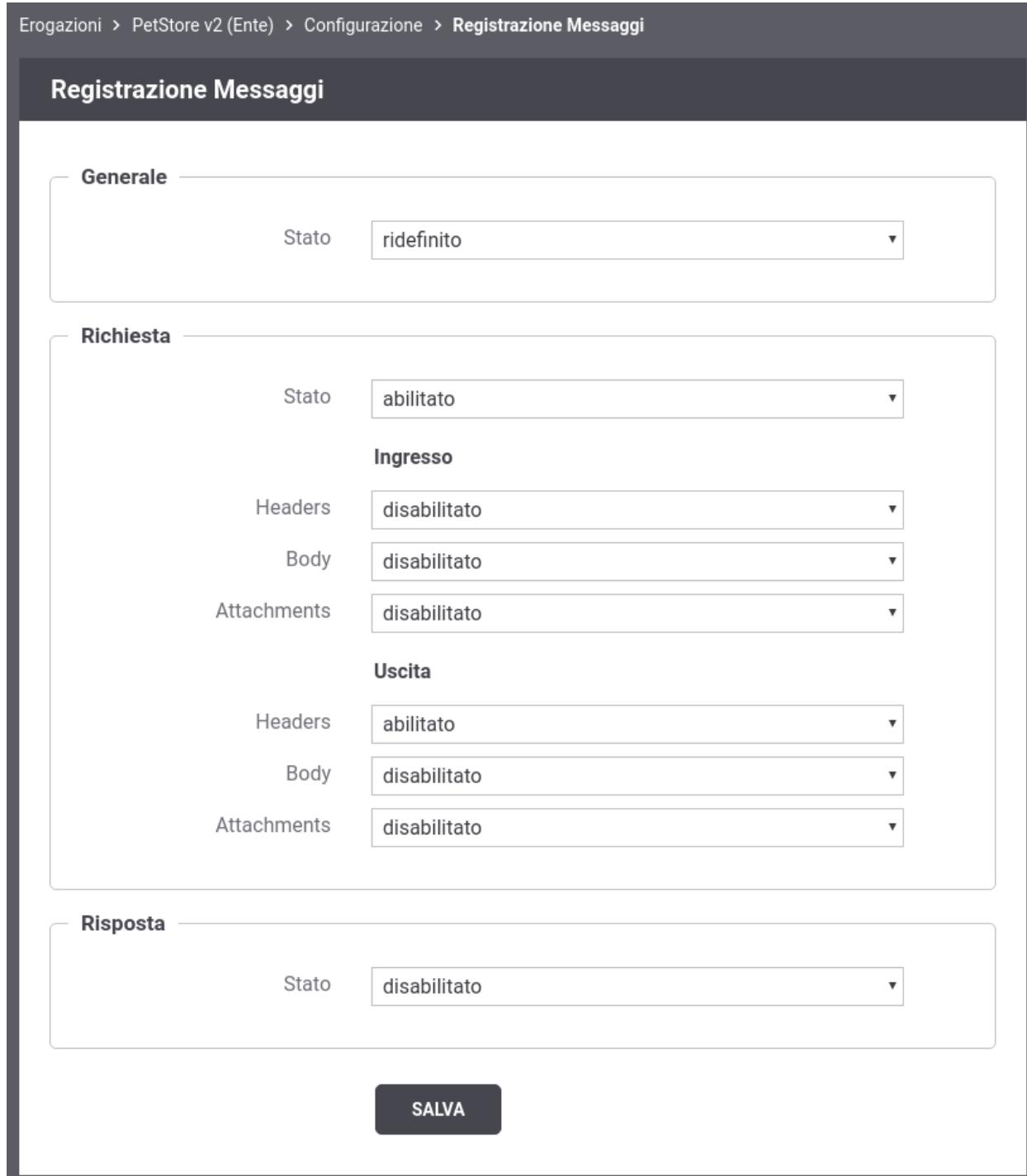


Fig. 3.9: Configurazione Registrazione Messaggi per visualizzare Header HTTP

The screenshot shows a user interface for transaction details. At the top, a breadcrumb navigation indicates: Storico > Intervallo Temporale > Dettaglio Transazione. The main title is 'Dettagli Transazione'. Below this, there are two expandable sections: 'Informazioni Generali' and 'Dettagli Richiesta'. The 'Informazioni Generali' section contains the following data:

|                        |  |
|------------------------|--|
| Tipologia              | Erogazione (API Gateway)                             |
| Erogatore              | Ente   |
| API                    | PetStore v2  |
| Azione                 | PUT_pet  |
| Profilo Collaborazione | Sincrono   |
| Esito                  | Ok   |
| Diagnostici            | <a href="#">Visualizza</a>   <a href="#">Esporta</a> |

The 'Dettagli Richiesta' section contains the following data:

|                  |  |
|------------------|--|
| 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 | <a href="#">Visualizza</a>   <a href="#">Esporta</a> |

Fig. 3.10: Dettaglio della transazione con contenuti

Storico > Intervallo Temporale > Dettagli Transazione > **Messaggio di Richiesta - Contenuti Uscita**

### Messaggio di Richiesta - Contenuti Uscita

**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                                       |

Fig. 3.11: Header HTTP prodotti da GovWay contenenti le informazioni sul Token

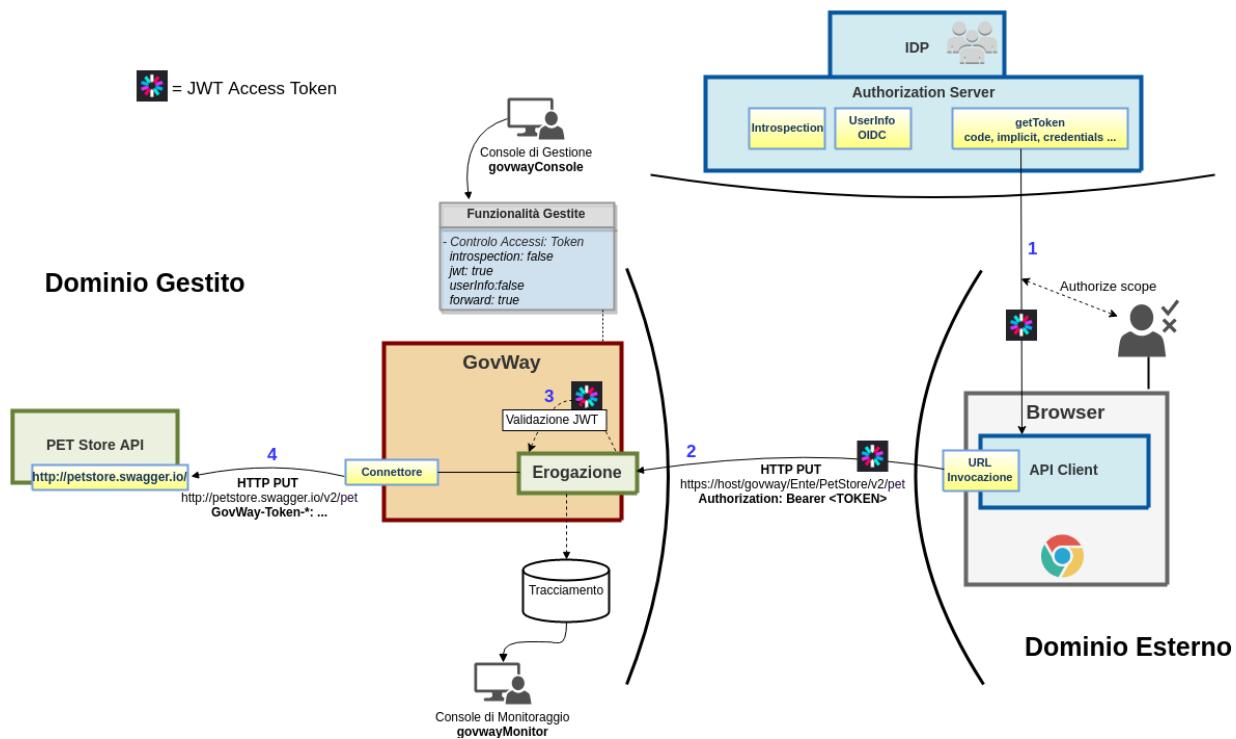


Fig. 3.12: Scenario OAuth con validazione JWT

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 salvare cliccando sul pulsante “Salva”.

#### • Acquisizione Access Token

Per simulare l’acquisizione di un token in formato JWT utilizzare l’applicazione *Playground* come descritto nella precedente sezione [Validazione tramite Introspection](#). 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*.

#### • Invocazione API con un access token

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

#### Nota: Bearer Token Usage

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

Erogazioni > PetStore v1 (ENTE) > Configurazione > **Controllo Accessi**

### Controllo Accessi

Note: (\*) Campi obbligatori

**Autenticazione Token**

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

**Required Claims**

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

**Autenticazione Trasporto**

|       |              |
|-------|--------------|
| Stato | disabilitato |
|-------|--------------|

Fig. 3.13: Configurazione OAuth2 - Validazione JWT

### 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%2FqQctBFrJConLp2VWBcKP40w0JqeAgPj56QlAuiKyn4Dz4dY9epFi7nfln-pxgyx0UkXlhxp_SC7rc0dqp8bZnE8&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: ESF
Content-encoding: gzip
Cache-control: private
Date: Tue, 04 Dec 2018 10:57:27 GMT
X-frame-options: SAMEORIGIN
Alt-svc: quic=":443"; ma=2592000; v="44,43,39,35"
Content-type: application/json; charset=utf-8

{
  "access_token": "ya29.Glt0BufJc390CX50k-ea3aZ0zGW29RGhUhZ0Se3TU46gp9IdbBpJLB3Ygo27RYGYmeF7sibN3rNb1r8BbBX",
  "id_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjQ2M2ZlNDgwYzNjNTg0WjYjE10DYxZTA4YzMyZDE4N2ZhZjlhNTYiLCJ0eXAiOiI",
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope": "https://www.googleapis.com/auth/plus.me",
  "refresh_token": "1/d5PhXioSyaQmV-Nw5gllMfn82pESCglU3u0f7_ULYR0"
}
```

Fig. 3.14: Ottenimento Token: Playground Google, Step 3

```
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,
```

(continues on next page)

(continua dalla pagina precedente)

```

"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        | <pre>[in] /govway/in/Ente/PetStore/v2/pet? access_token=ya29.Glt0BjchXoKaglEFXIOUxsN1UVVUW1ryp...kT2laD8ERHY 1ZyE-Af2sMPrL-cOWzZx_R</pre> |
| Indirizzo Client       | 127.0.0.1   |
| Codice Risposta Client | 200   |
| <b>Token Info</b>      |   |
| Client ID              | 407408718192.apps.googleusercontent.com   |
| Subject                | 106235657592654397689   |
| Token Info             | <a href="#">Visualizza</a>  |

Fig. 3.15: Traccia di una invocazione terminata con successo

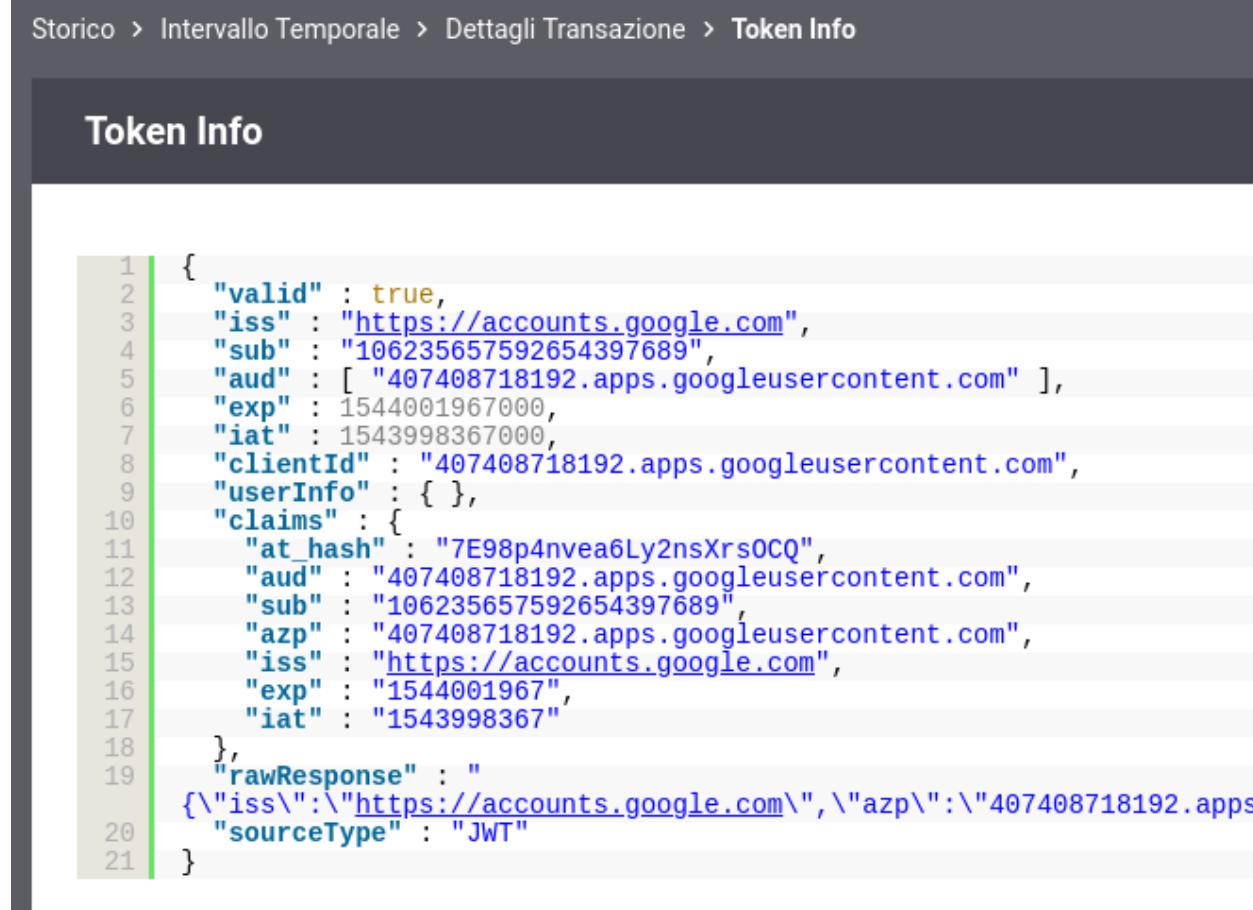
Cliccando sul link “Visualizza” della voce “*Token Info*” è possibile vedere tutti i claims presenti nel token, tra cui è possibile constatare la presenza del claim *scope* valorizzato con quanto richiesto tramite l’applicazione Playground.

### 3.3 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 opaco tramite il servizio di Introspection (descritto nella sezione *Validazione tramite Introspection*), altrimenti tramite la validazione JWT (sezione *Validazione JWT*) 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 Fig. 3.8 e Fig. 3.16.

GovWay può essere configurato per verificare che un *access token* presenti 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).



The screenshot shows a web interface with a dark header bar. The header contains the text: "Storico > Intervallo Temporale > Dettagli Transazione > Token Info". Below the header, the main content area has a dark header with the text "Token Info". The main content area contains a code block with line numbers (1 to 21) and JSON-like text. The text describes a JWT token with various fields and their values.

```
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" : "7E98p4nvea6Ly2nsXrsOCQ",
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","sourceType": "JWT"
21 }
```

Fig. 3.16: Informazioni presenti in un Token JWT

- *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](#).

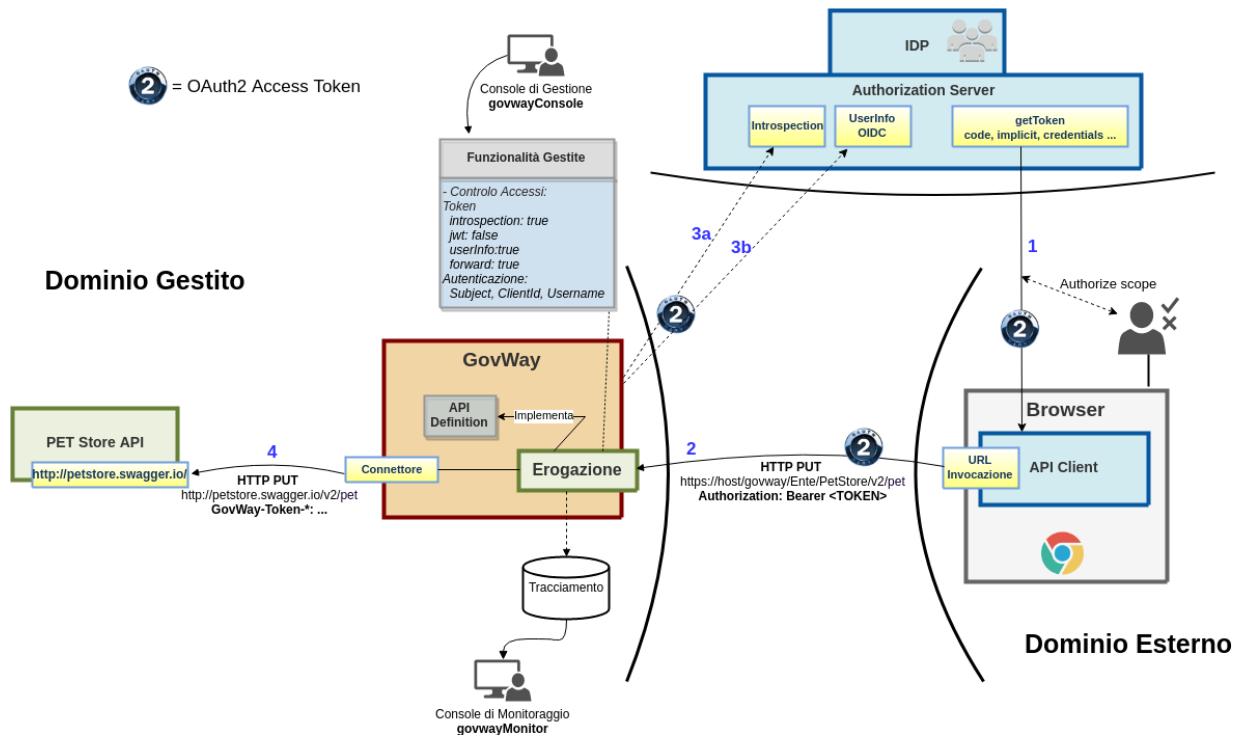


Fig. 3.17: Scenario OAuth con accesso servizio UserInfo

Per simulare lo scenario utilizzeremo sempre il servizio *Playground* e l’*Authorization Server* di *Google* descritto nella precedente sezione [Validazione tramite Introspection](#). 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 v1*”. Dopo di che 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 abilitando i seguenti “Required Claims”:

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

Effettuata la configurazione salvarla cliccando sul pulsante “Salva”.

### • Invocazione API

---

#### Nota: 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 *Validazione tramite Introspection* 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
{
```

(continues on next page)

Erogazioni > PetStore v1 (ENTE) > Configurazione > **Controllo Accessi**

## Controllo Accessi

Note: (\*) Campi obbligatori

**Autenticazione Token**

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

**Required Claims**

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

**Autenticazione Trasporto**

|       |              |
|-------|--------------|
| Stato | disabilitato |
|-------|--------------|

Fig. 3.18: Configurazione OAuth2 - Autenticazione

(continua dalla pagina precedente)

```

"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 Fig. 3.19 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con errore con esito *Autenticazione Fallita*.

| Lista Transazioni: record [1 - 4] |  |            |                        |          |           |             |         |
|-----------------------------------|--|------------|------------------------|----------|-----------|-------------|---------|
|                                   | Data Ingresso Richiesta                              | Tipologia  | Esito                  | Fruitore | Erogatore | API         | Azione  |
| <input type="checkbox"/>          | <input checked="" type="radio"/> 2018-12-05 15:31:42 | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <input checked="" type="radio"/> 2018-12-05 15:31:42 | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <input checked="" type="radio"/> 2018-12-05 15:31:41 | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <input checked="" type="radio"/> 2018-12-05 15:29:46 | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet |

Fig. 3.19: 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

| 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 [b6fdbdd4-051a-4a3f-87da-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 |

ESPORTA

Fig. 3.20: 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”.

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

Storico > Intervallo Temporale > Dettagli Transazione > Token Info

Token Info

```
1  {
2      "valid" : true,
3      "sub" : "106235657592654397689",
4      "aud" : [ "407408718192.apps.googleusercontent.com" ],
5      "exp" : 1544023764000,
6      "clientId" : "407408718192.apps.googleusercontent.com",
7      "scopes" : [ "https://www.googleapis.com/auth/plus.me" ],
8      "userInfo" : { },
9      "claims" : {
10         "aud" : "407408718192.apps.googleusercontent.com",
11         "sub" : "106235657592654397689",
12         "access_type" : "offline",
13         "azp" : "407408718192.apps.googleusercontent.com",
14         "scope" : "https://www.googleapis.com/auth/plus.me",
15         "exp" : "1544023764",
16         "expires_in" : "3578"
17     },
18     "rawResponse" : "[\n    \"azp\": \"407408718192.apps.googleusercontent.com\",\n    \"aud\": \"407408718192.apps.googleusercontent.com\",\n    \"sub\": \"106235657592654397689\",\n    \"scope\": \"https://www.googleapis.com/auth/plus.me\",\n    \"exp\": \"1544023764\",\n    \"expires_in\": \"3578\",\n    \"access_type\": \"offline\"\n]\n",
19     "sourceType" : "INTROSPECTION"
20 }
```

DOWNLOAD

Fig. 3.21: Informazioni presenti nel Token

Tramite la *govwayConsole* accedere nuovamente alla maschera di configurazione “*Controllo Accessi*” dell’API *“PetStore v1”* ed abilitare stavolta anche il servizio *“User Info”*.

- Nuova invocazione API

**Nota: 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 *Validazione tramite Introspection* 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.

Erogazioni > PetStore v1 (ENTE) > Configurazione > **Controllo Accessi**

### Controllo Accessi

Note: (\*) Campi obbligatori

**Autenticazione Token**

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

**Required Claims**

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

**Autenticazione Trasporto**

|       |              |
|-------|--------------|
| Stato | disabilitato |
|-------|--------------|

Fig. 3.22: Configurazione OAuth2 - Autenticazione

- **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?<br>access_token=ya29.GltpBpHYHBNDdKRNNP1_fedzujBFAe5Jr39tukpYdzhvne<br>9g97sAeoFAUeJA6QOMX2IovSYDa5JICzVLH5qkl0cD2SGw5rfzmlvRED3Ej0v0<br>jxe7wRBlfRhGojWS |
| Indirizzo Client       | 127.0.0.1   |
| Codice Risposta Client | 200   |
| <b>Token Info</b>      |   |
| Client ID              | 407408718192.apps.googleusercontent.com   |
| Subject                | 106235657592654397689   |
| Username               | Andrea Poli   |
| Token Info             | <a href="#">Visualizza</a>  |

Fig. 3.23: 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*”.

## 3.4 Autorizzazione per Scope

La verifica di un *access token*, se opaco tramite il servizio di Introspection (descritto nella sezione *Validazione tramite Introspection*), altrimenti tramite la validazione JWT (sezione *Validazione JWT*), 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 *Validazione tramite Introspection* 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*.

- **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*.

Storico > Intervallo Temporale > Dettagli Transazione > Token Info

### Token Info

```

1  {
2      "valid" : true,
3      "sub" : "106235657592654397689",
4      "username" : "Andrea Poli",
5      "aud" : [ "407408718192.apps.googleusercontent.com" ],
6      "exp" : 1544023764000,
7      "clientId" : "407408718192.apps.googleusercontent.com",
8      "scopes" : [ "https://www.googleapis.com/auth/plus.me" ],
9      "userInfo" : {
10         "fullName" : "Andrea Poli",
11         "firstName" : "Andrea",
12         "familyName" : "Poli"
13     },
14     "claims" : {
15         "aud" : "407408718192.apps.googleusercontent.com",
16         "sub" : "106235657592654397689",
17         "access_type" : "offline",
18         "azp" : "407408718192.apps.googleusercontent.com",
19         "scope" : "https://www.googleapis.com/auth/plus.me",
20         "profile" : "https://plus.google.com/106235657592654397689",
21         "name" : "Andrea Poli",
22         "exp" : "1544023764",
23         "given_name" : "Andrea"
24     }
25 }
```

[DOWNLOAD](#)

Fig. 3.24: Informazioni presenti in un Token JWT

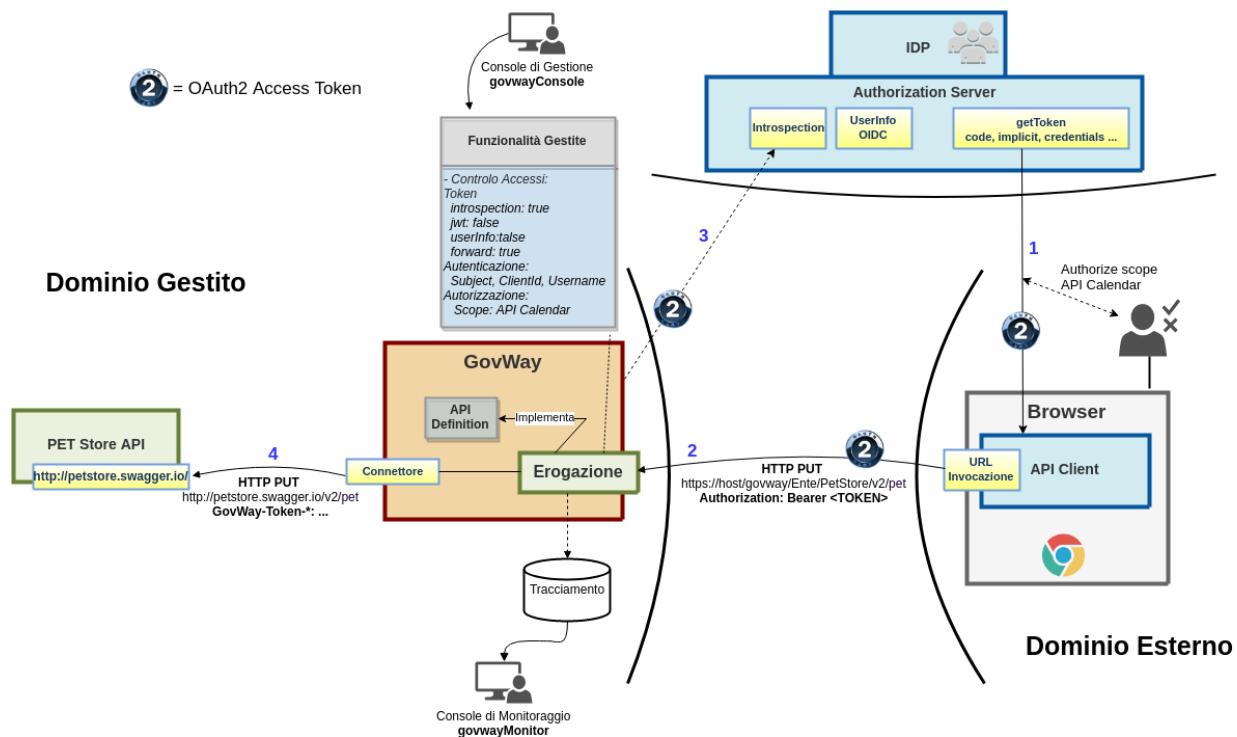


Fig. 3.25: Scenario OAuth con autorizzazione per Scope

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.

### 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 *Playground* all'utilizzo degli 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.

#### • 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.

Tabella 3.1: Registrazione Scope

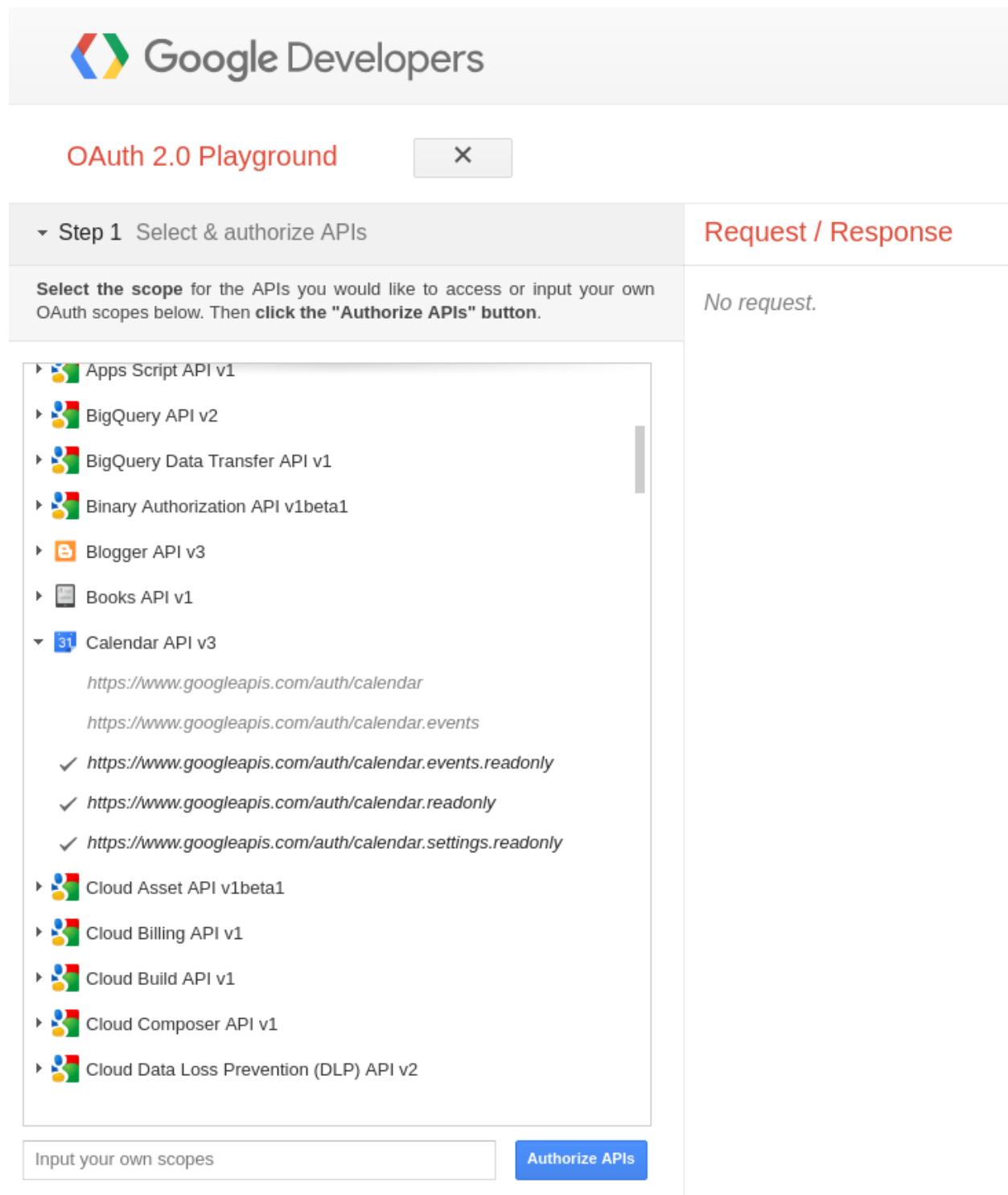
| Nome                              | Identificativo Esterno  | Contesto  |
|-----------------------------------|---|-----------|
| google.calendar.events.readonly   | <a href="https://www.googleapis.com/auth/calendar.events.readonly">https://www.googleapis.com/auth/calendar.events.readonly</a>     | Qualsiasi |
| google.calendar.readonly          | <a href="https://www.googleapis.com/auth/calendar.readonly">https://www.googleapis.com/auth/calendar.readonly</a>                   | Qualsiasi |
| google.calendar.settings.readonly | <a href="https://www.googleapis.com/auth/calendar.settings.readonly">https://www.googleapis.com/auth/calendar.settings.readonly</a> | Qualsiasi |
| google.drive                      | <a href="https://www.googleapis.com/auth/drive">https://www.googleapis.com/auth/drive</a>   | Qualsiasi |

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

#### • Configurazione Controllo degli Accessi

Accedere alla sezione “*Erogazioni*” e selezionare l'API precedentemente registrata “*PetStore v1*”. 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



Google Developers

OAuth 2.0 Playground X

| Step 1 Select & authorize APIs   | Request / Response  |
|--|---|
| <p>Select the <b>scope</b> for the APIs you would like to access or input your own OAuth scopes below. Then <b>click the "Authorize APIs" button</b>.</p> <ul style="list-style-type: none"><li>▶  Apps Script API v1</li><li>▶  BigQuery API v2</li><li>▶  BigQuery Data Transfer API v1</li><li>▶  Binary Authorization API v1beta1</li><li>▶  Blogger API v3</li><li>▶  Books API v1</li><li>▼  <b>Calendar API v3</b><ul style="list-style-type: none"><li><a href="https://www.googleapis.com/auth/calendar">https://www.googleapis.com/auth/calendar</a></li><li><a href="https://www.googleapis.com/auth/calendar.events">https://www.googleapis.com/auth/calendar.events</a></li><li>✓ <a href="https://www.googleapis.com/auth/calendar.events.readonly">https://www.googleapis.com/auth/calendar.events.readonly</a></li><li>✓ <a href="https://www.googleapis.com/auth/calendar.readonly">https://www.googleapis.com/auth/calendar.readonly</a></li><li>✓ <a href="https://www.googleapis.com/auth/calendar.settings.readonly">https://www.googleapis.com/auth/calendar.settings.readonly</a></li></ul></li><li>▶  Cloud Asset API v1beta1</li><li>▶  Cloud Billing API v1</li><li>▶  Cloud Build API v1</li><li>▶  Cloud Composer API v1</li><li>▶  Cloud Data Loss Prevention (DLP) API v2</li></ul> | <p>No request.</p>  |
| <input type="text" value="Input your own scopes"/>   | <span style="border: 1px solid #00796b; padding: 2px 10px; border-radius: 5px; color: #00796b; text-decoration: none; font-weight: bold;">Authorize APIs</span> |

Fig. 3.26: Ottenimento Token: Playground Google, scelta scope API Calendar

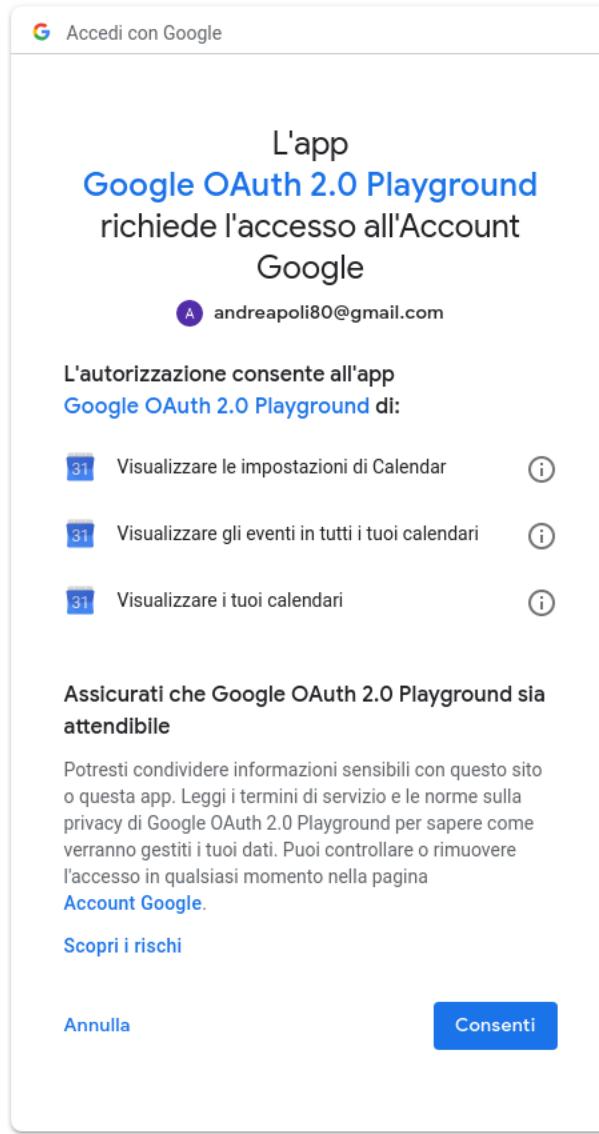


Fig. 3.27: Ottenimento Token: Playground Google, autorizzazione scope API Calendar

### Request / Response

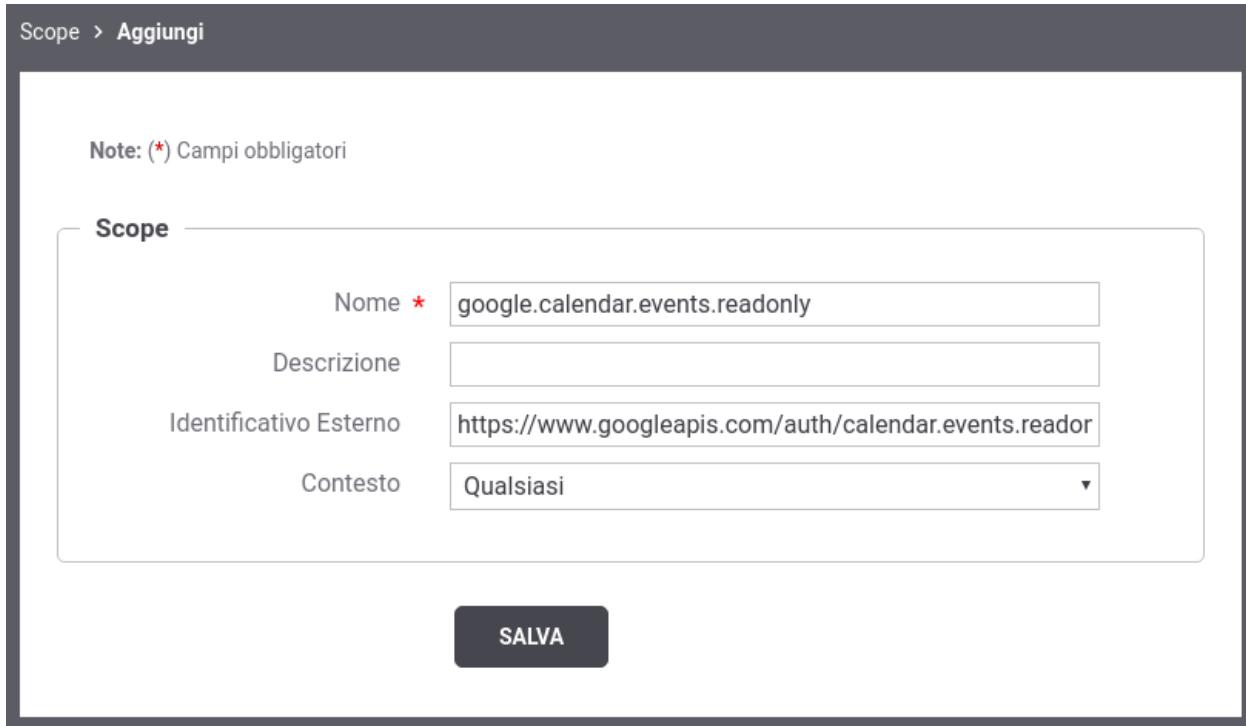
```
POST /oauth2/v4/token HTTP/1.1
Host: www.googleapis.com
Content-length: 277
Content-type: application/x-www-form-urlencoded
User-Agent: google-oauth-playground

code=4%2FqgBqFLXZ1DEB5EU9Lb01oA3xr1fo4HgNCqplozTtw5h7a_0f2mq05PNEkafedo5GpYX6mW5PH0btAi030CY9ih0&redirect_uri=https%3A%2F%2Fplayground.google.com%2Fcallback

HTTP/1.1 200 OK
Content-length: 449
Content-type: application/json; charset=utf-8
Date: Wed, 05 Dec 2018 15:32:41 GMT
Server: ESF
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Request-Id: f2mTR8P80LR3lMhqJDZuelsi4Re_G0A
X-XSS-Protection: 1; mode=block
Transfer-Encoding: chunked
Vary: Origin, X-Origin, Referer
Alt-Svc: quic=":443"; ma=2592000; v="44,43,39,35"
Cache-Control: private
Content-Encoding: gzip
Content-Type: application/json; charset=utf-8

{
  "access_token": "ya29.GltBkJy5djr9V6z4nPgl0PfnUJghEasRTrm10een04rBAke0ou2Gj4PId-bGls-f2mTR8P80LR3lMhqJDZuelsi4Re_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/MdYRAcdcEl6auXXbwq003nHarKVfEPv9mncfIZEmQs8"
}
```

Fig. 3.28: Ottenimento Token: Playground Google, Step 3



Scope > Aggiungi

Note: (\*) Campi obbligatori

**Scope**

|                        |  |
|------------------------|--|
| Nome *                 | google.calendar.events.readonly                          |
| Descrizione            |  |
| Identificativo Esterno | https://www.googleapis.com/auth/calendar.events.readonly |
| Contesto               | Qualsiasi  |

**SALVA**

Fig. 3.29: Configurazione OAuth2 - Registrazione Scope

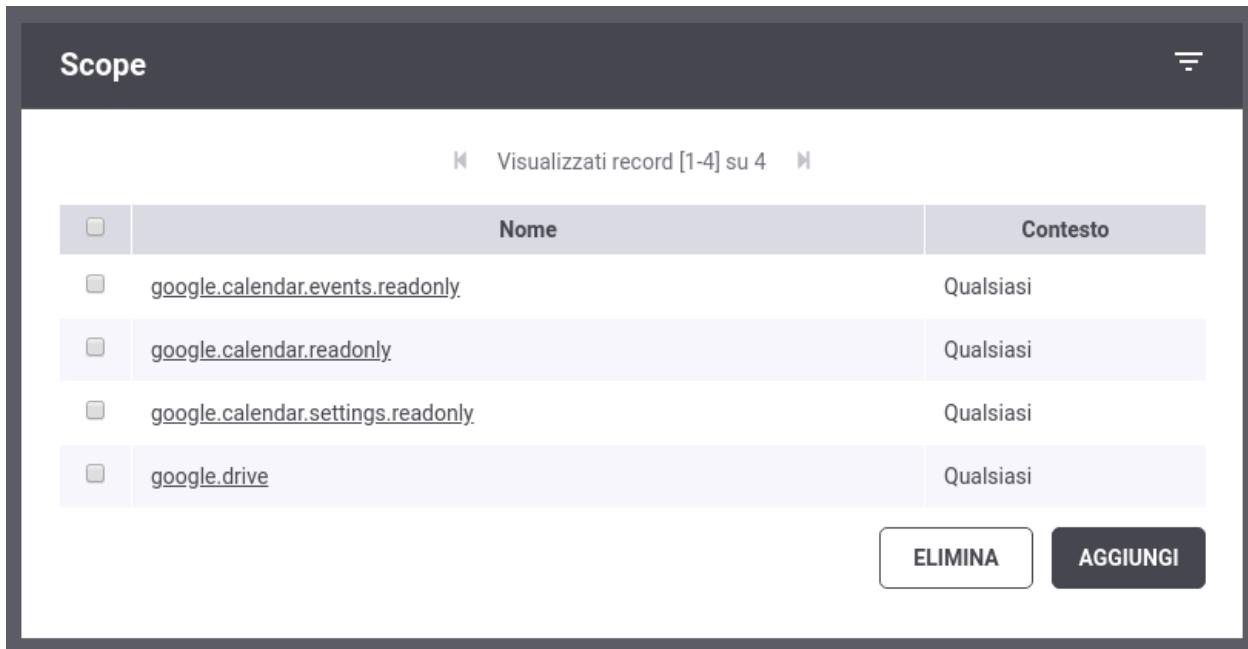


Fig. 3.30: Configurazione OAuth2 - Lista degli Scope registrati

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”.

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.

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

- **Invocazione API**

**Nota: 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": "kai",
    "petType": "dog",
    "age": 3,
    "size": "small",
    "status": "available"
}'
```

(continues on next page)

**Autenticazione Token**

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

**Required Claims**

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

**Autenticazione Trasporto**

|       |              |
|-------|--------------|
| Stato | disabilitato |
|-------|--------------|

**Autorizzazione**

|  |                                     |
|--|-------------------------------------|
| Stato                                  | abilitato                           |
| <b>Autorizzazione per Ruoli</b>        |                                     |
| Abilitato                              | <input type="checkbox"/>            |
| <b>Autorizzazione per Scope</b>        |                                     |
| Abilitato                              | <input checked="" type="checkbox"/> |
| Scope Richiesti                        | tutti                               |
| <b>Autorizzazione per Token Claims</b> |                                     |
| Abilitato                              | <input type="checkbox"/>            |

Fig. 3.31: Configurazione OAuth2 - Autorizzazione

**Autorizzazione**

|       |           |
|-------|-----------|
| Stato | abilitato |
|-------|-----------|

**Autorizzazione per Ruoli**

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

**Autorizzazione per Scope**

|                           |                                     |
|---------------------------|-------------------------------------|
| Abilitato                 | <input checked="" type="checkbox"/> |
| Scope Richiesti           | tutti                               |
| <a href="#">Scope (0)</a> |                                     |

**Autorizzazione per Token Claims**

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

Fig. 3.32: Configurazione OAuth2 - Autorizzazione - Scope

Erogazioni > PetStore v2 (Ente) > Configurazione > Controllo Accessi > **Scope**

| Scope                          |  |
|--------------------------------|--|
| Visualizzati record [1-4] su 4 |  |
| <input type="checkbox"/>       | Nome   |
| <input type="checkbox"/>       | google.calendar.events.readonly                  |
| <input type="checkbox"/>       | google.calendar.readonly                         |
| <input type="checkbox"/>       | google.calendar.settings.readonly                |
| <input type="checkbox"/>       | google.drive                                     |
|                                | <a href="#">ELIMINA</a> <a href="#">AGGIUNGI</a> |

Fig. 3.33: Configurazione OAuth2 - Autorizzazione - Elenco Scope

(continua dalla pagina precedente)

```

    "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 Fig. 3.34 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con esito *Autorizzazione Negata*.

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

Fig. 3.34: 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.

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.

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

| 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) completata con successo  |  |
| 2018-12-05 17:20:12.264                | infoProtocol     | RicezioneBuste | Ricevuto messaggio di cooperazione con identificativo [cc5a58d7-2131-4c79-9028-d6b235bb084]  |  |
| 2018-12-05 17:20:12.264                | infoIntegration  | RicezioneBuste | Verifica autorizzazione [token] messaggio con identificativo [cc5a58d7-2131-4c79-9028-d6b235bb084] 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 [cc5a58d7-2131-4c79-9028-d6b235bb084] 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   |  |

Fig. 3.35: Diagnostici di una invocazione terminata con errore

Storico > Intervallo Temporale > Dettagli Transazione > Token Info

## Token Info

```
1  {
2    "valid" : true,
3    "aud" : [ "407408718192.apps.googleusercontent.com" ],
4    "exp" : 1544027561000,
5    "clientId" : "407408718192.apps.googleusercontent.com",
6    "scopes" : [ "https://www.googleapis.com/auth/calendar.events.readonly", "https://www.googleapis.com/auth/calendar.readonly" ],
7    "userInfo" : { },
8    "claims" : {
9      "aud" : "407408718192.apps.googleusercontent.com",
10     "access_type" : "offline",
11     "azp" : "407408718192.apps.googleusercontent.com",
12     "scope" : "https://www.googleapis.com/auth/calendar.events.readonly https://www.googleapis.com/auth/calendar.readonly https://www.googleapis.com/auth/calendar.settings.readonly",
13     "exp" : "1544027561",
14     "expires_in" : "956",
15   },
16   "rawResponse" : "{\n  \"azp\": \"407408718192.apps.googleusercontent.com\", \n  \"aud\": \"407408718192.apps.googleusercontent.com\", \n  \"scope\": \"https://www.googleapis.com/auth/calendar.events.readonly https://www.googleapis.com/auth/calendar.readonly https://www.googleapis.com/auth/calendar.settings.readonly\", \n  \"exp\": \"1544027561\", \n  \"expires_in\": \"956\", \n  \"access_type\": \"offline\"\n}\n",
17   "sourceType" : "INTROSPECTION"
18 }
```

**DOWNLOAD**

Fig. 3.36: Scope presenti nel Token

Tramite la *govwayConsole* accedere nuovamente alla maschera di configurazione “*Controllo Accessi*” dell’API “*PetStore v1*”; all’interno della sezione “*Autorizzare*” modificare il tipo di controllo “*Scope Richiesti*” dal valore “*tutti*” al valore “*almeno uno*”.

The screenshot shows the 'Autorizzazione' (Authorization) configuration screen. At the top, there is a dropdown labeled 'Stato' (Status) with the value 'abilitato' (enabled). Below this, there are two sections: 'Autorizzazione per Ruoli' (Authorization for Roles) and 'Autorizzazione per Scope' (Authorization for Scopes). In the 'Autorizzazione per Scope' section, the 'Abilitato' (Enabled) checkbox is checked. The 'Scope Richiesti' dropdown is highlighted with a red border and contains the value 'almeno uno' (at least one). Below the dropdown, the text 'Scope (4)' is visible. The 'Autorizzazione per Token Claims' section is also present but is not highlighted.

Fig. 3.37: Configurazione OAuth2 - Autorizzazione degli scope con opzione “Almeno uno”

- **Nuova invocazione API**

---

**Nota: 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.

## 3.5 Autorizzazione sui Claims

Oltre ad una autorizzazione sugli scope, descritta nello scenario della sezione *Autorizzazione per Scope*, 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 validazioni più complesse si rimanda all'utilizzo di una policy XACML descritta nello scenario della sezione *Autorizzazione XACML*.

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

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.

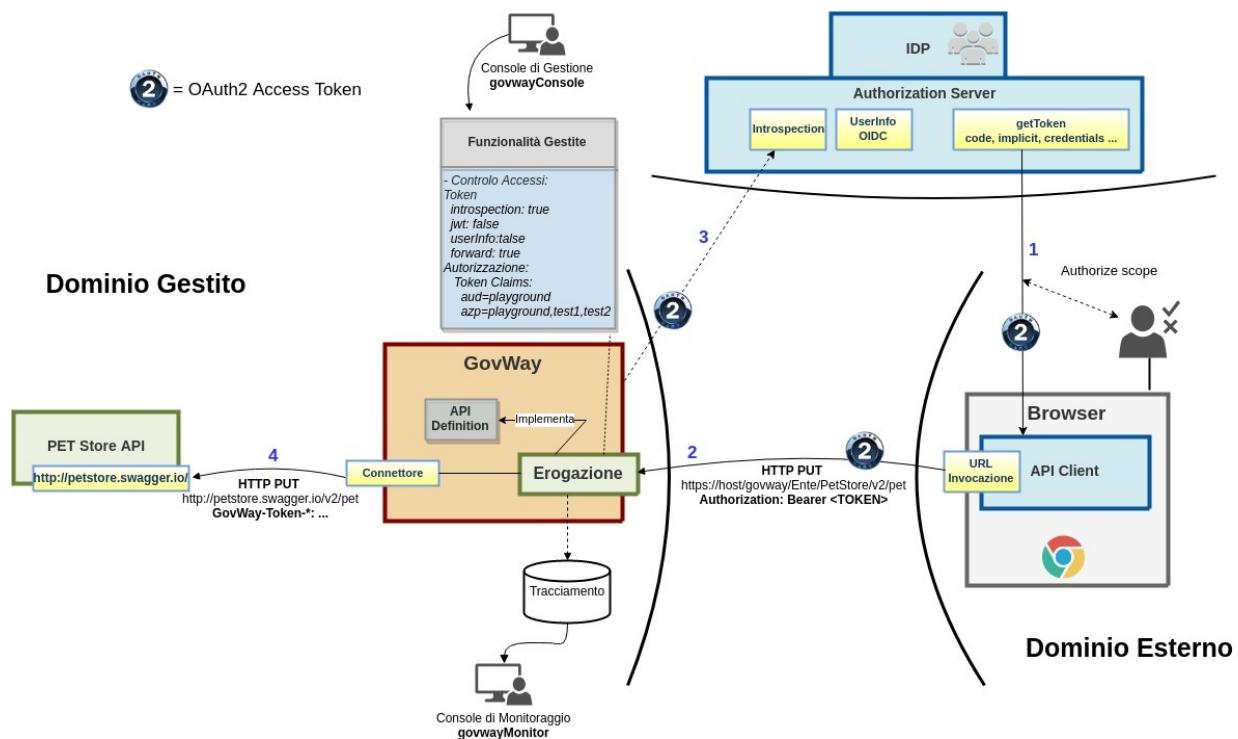


Fig. 3.38: Scenario OAuth con autorizzazione sui Claims

### • Configurazione Controllo degli Accessi

Accedere alla sezione “Erogazioni” e selezionare l'API precedentemente registrata “PetStore v1”. 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*”:

---

**Nota: Per conoscere l'identificativo dell'applicazione Playground**

È possibile vedere una precedente transazione terminata con successo per conoscere l'esatto valore associato all'applicazione *Playground* (es. Fig. 3.8).

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

Effettuata la configurazione salvarla cliccando sul pulsante “*Salva*”.

**Autorizzazione**

Stato: abilitato

**Autorizzazione per Ruoli**

Abilitato:

**Autorizzazione per Scope**

Abilitato:

**Autorizzazione per Token Claims**

Abilitato:

Claims: aud=407408718192.apps.googleusercontent.com  
azp=client1, client2

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

Fig. 3.39: Configurazione OAuth2 - Autorizzazione

- **Invocazione API**

---

**Nota: 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 *Validazione tramite Introspection*.

```
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 Fig. 3.40 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con esito *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”.

- **Registrazione ClientId corretto in Controllo degli Accessi**

Tramite la *govwayConsole* accedere nuovamente alla maschera di configurazione “Controllo Accessi” dell’API “PetStore v1”; all’interno della sezione “Autorizzare” modificare il valore del claim “azp” aggiungendo l’applicazione *Playground*:

- aud=407408718192.apps.googleusercontent.com

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

Fig. 3.40: Tracce delle invocazioni terminate con errore “Autorizzazione Negata”

| Lista Diagnostici: record [1 - 8] su 8 |                  |                |  |
|--|------------------|----------------|--|
| Data                                   | Severità         | 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 [b9fcefc-5e6a-4bf0-b84c-84250c009c2a]  |
| 2018-12-11 16:37:20.279                | infoIntegration  | RicezioneBuste | Verifica autorizzazione [token] messaggio con identificativo [b9fcefc-5e6a-4bf0-b84c-84250c009c2a] servizio [gw/Ente:gw/PetStore:2:PUT_pet] in corso ...   |
| 2018-12-11 16:37:20.286                | errorIntegration | RicezioneBuste | Verifica autorizzazione [token] messaggio con identificativo [b9fcefc-5e6a-4bf0-b84c-84250c009c2a] servizio [gw/Ente:gw/PetStore:2: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”:”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                           |

Fig. 3.41: Diagnostici di una invocazione terminata con errore

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

**Autorizzazione**

|  |  |
|--|--|
| Stato  | abilitato  |
| <b>Autorizzazione per Ruoli</b>  |  |
| Abilitato  | <input type="checkbox"/>   |
| <b>Autorizzazione per Scope</b>  |  |
| Abilitato  | <input type="checkbox"/>   |
| <b>Autorizzazione per Token Claims</b>   |  |
| Abilitato  | <input checked="" type="checkbox"/>  |
| Claims   | <pre>aud=407408718192.apps.googleusercontent.com azp=client1, client2, 407408718192.apps.googleusercontent.com</pre> |
| <small>Indicare per riga i claims richiesti (nome=valore); visualizzare 'info' per maggiori dettagli</small> |  |

Fig. 3.42: Configurazione OAuth2 - Autorizzazione dei claims corretta

- **Nuova invocazione API**

---

**Nota: 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.

## 3.6 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 *Validazione tramite Introspection*.

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.

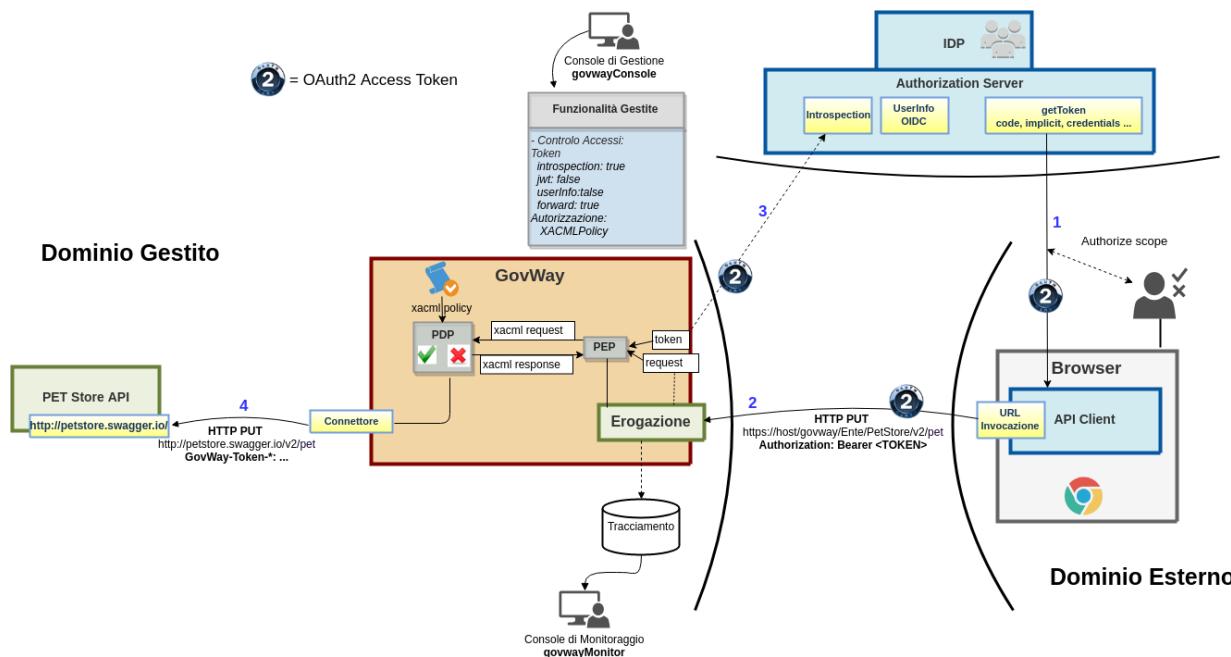


Fig. 3.43: Scenario OAuth con autorizzazione XACMLPolicy

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.

Tabella 3.2: Parametri XACML

| 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  |
| 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. spcoop)  |
| <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 *Autorizzazione sui Claims*. 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.

---

### Nota: Per conoscere l'identificativo dell'applicazione Playground

È possibile vedere una precedente transazione terminata con successo per conoscere l'esatto valore associato all'applicazione *Playground* (es. Fig. 3.8).

---

```
<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">
                    <
                        AttributeId=""
                        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"></AttributeValue>
                    </Apply>
                </Apply>
                <Apply
                    FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-
                    ↪one-member-of">
                    <
                        AttributeId=""
                        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"></AttributeValue>
                        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema
                        ↪#string"></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 v1*”. 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”.

Fig. 3.44: Configurazione OAuth2 - Autorizzazione XACML Policy

#### • Invocazione API

##### **Nota: 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 *Validazione tramite Introspection*.

```
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",
    "version": 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 Fig. 3.45 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"/>          | <input checked="" type="radio"/> 2018-12-05 17:20:12 | Erogazione | Autorizzazione Negata |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <input checked="" type="radio"/> 2018-12-05 17:20:11 | Erogazione | Autorizzazione Negata |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <input checked="" type="radio"/> 2018-12-05 17:20:11 | Erogazione | Autorizzazione Negata |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <input checked="" type="radio"/> 2018-12-05 17:16:45 | Erogazione | Autorizzazione Negata |          | Ente      | PetStore v2 | PUT_pet |

Fig. 3.45: 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)”.

- **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>
```

(continues on next page)

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]   |
| 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:2:PUT_pet] in corso ...  |
| 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:2: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 [b89b8c42-04d8-42d6-9451-aa9a669dccfa]   |
| 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); govway_status": "protocol:GOVWAY-1352"}) consegnata al mittente con codice di trasporto: 403 |

**ESPORTA**

Fig. 3.46: Diagnostici di una invocazione terminata con errore

(continua dalla pagina precedente)

```

<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">
        <
            AttributeId=""
            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"></AttributeValue>
            </Apply>
        </Apply>

        <Apply
            FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
            ↵least-one-member-of">
            <
                AttributeId=""
                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"></AttributeValue>
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema
        ↵#string"></AttributeValue>
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema
        ↵#string"></AttributeValue>
            </Apply>
        </Apply>

```

(continues on next page)

(continua dalla pagina precedente)

```
        </Apply>
    </Condition>
</Rule>
<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 v1*”; all’interno della sezione “*Autorizzare*” caricare la policy aggiornata.

- Nuova invocazione API

**Nota: 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.

### 3.7 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. [Fig. 3.2](#)). 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*”. Accedere alla sezione “*Erogazioni*” e selezionare l’API precedentemente registrata “*PetStore v1*”. 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”.

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” (Fig. 3.48) che permette di vedere gli header http prodotti da GovWay (Fig. 3.49).

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 (Fig. 3.50) e inoltre delle informazioni raccolte durante la validazione del token (Fig. 3.51).

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:

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**

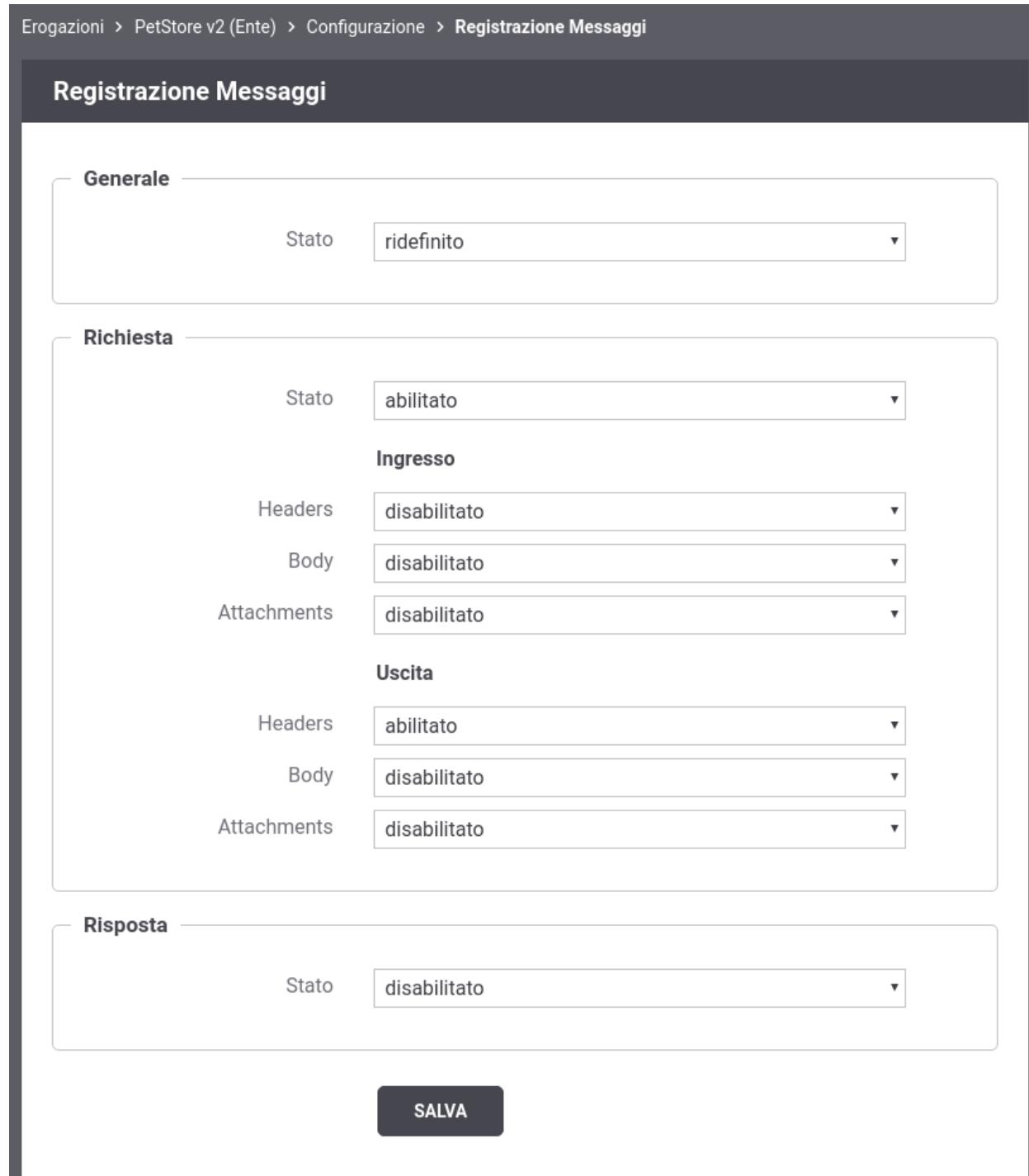


Fig. 3.47: Configurazione Registrazione Messaggi per visualizzare Header HTTP

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            | <a href="#">Visualizza</a>   <a href="#">Esporta</a> |

**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 | <a href="#">Visualizza</a>   <a href="#">Esporta</a> |

Fig. 3.48: Dettaglio della transazione con contenuti

Storico > Intervallo Temporale > Dettagli Transazione > **Messaggio di Richiesta - Contenuti Uscita**

### Messaggio di Richiesta - Contenuti Uscita

**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                                       |

Fig. 3.49: Header HTTP prodotti da GovWay contenenti le informazioni sul Token



Fig. 3.50: Modalità di Forward delle Informazioni Raccolte

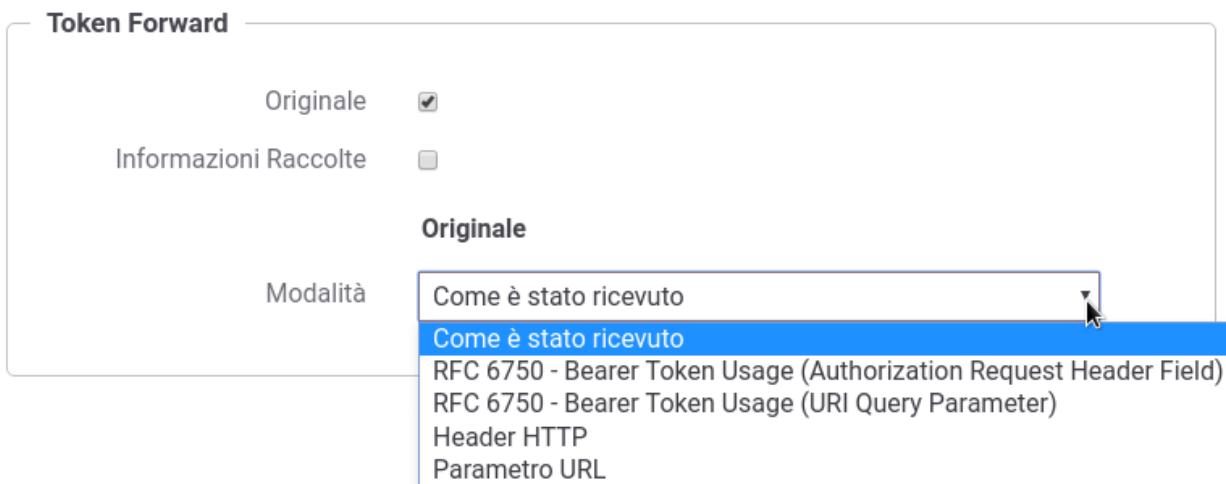


Fig. 3.51: Modalità di Forward del Token Originale

```
{  
    "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.

## 3.8 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. [Fig. 3.2](#)).

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 [Token Forward](#)

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 type="text"/> |

**Informazioni Generali**

**Token**

|           |  |
|-----------|--|
| Tipo      | <input type="text" value="JWS"/>                           |
| Posizione | <input type="text" value="RFC 6750 - Bearer Token Usage"/> |

**Elaborazione Token**

|                     |                                     |
|---------------------|-------------------------------------|
| Validazione JWT     | <input checked="" type="checkbox"/> |
| Token Introspection | <input checked="" type="checkbox"/> |
| OIDC - UserInfo     | <input checked="" type="checkbox"/> |
| Token Forward       | <input checked="" type="checkbox"/> |

**Endpoint Token**

|                      |                                     |
|----------------------|-------------------------------------|
| Connection Timeout * | <input type="text" value="10000"/>  |
| Read Timeout *       | <input type="text" value="120000"/> |

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

Fig. 3.52: Token Policy di esempio: Google (1/2)

**Validazione JWT**

Claims Parser

**Token Introspection**

Tipo

URL \*

**OIDC - UserInfo**

Tipo

URL \*

**Https**

Tipologia

Hostname Verifier

**Autenticazione Server**

Tipo

File \*

Password \*

Algoritmo \*

**Token Forward**

Originale

Informazioni Raccolte

**Informazioni Raccolte**

Modalità

**SALVA**



# CAPITOLO 4

## 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).

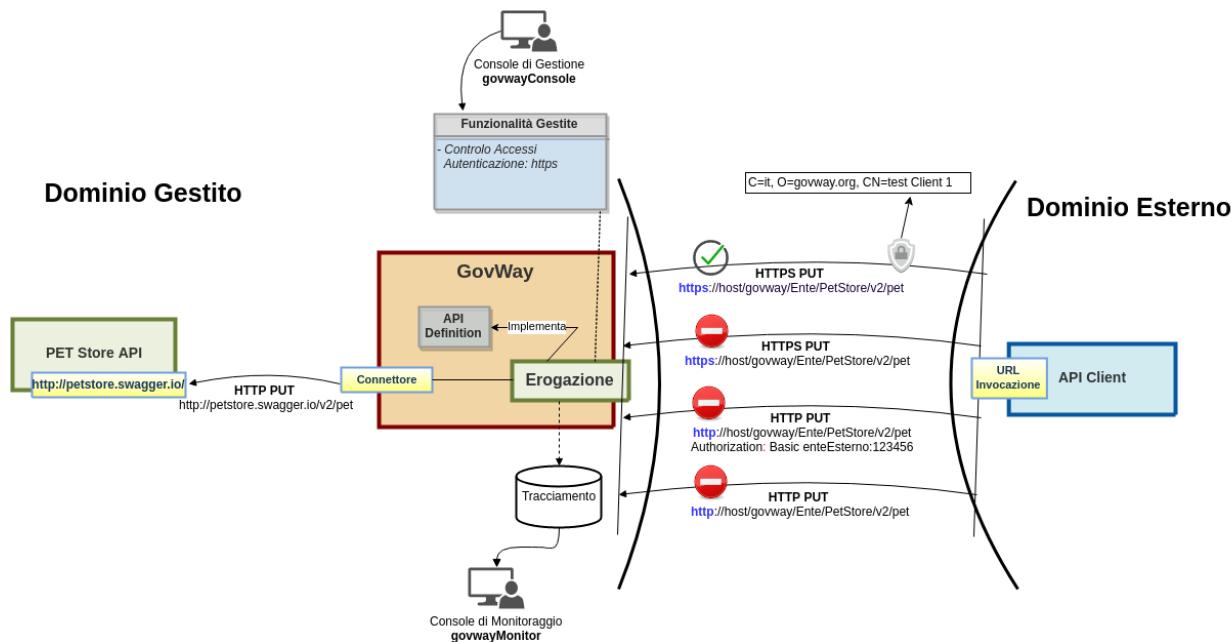
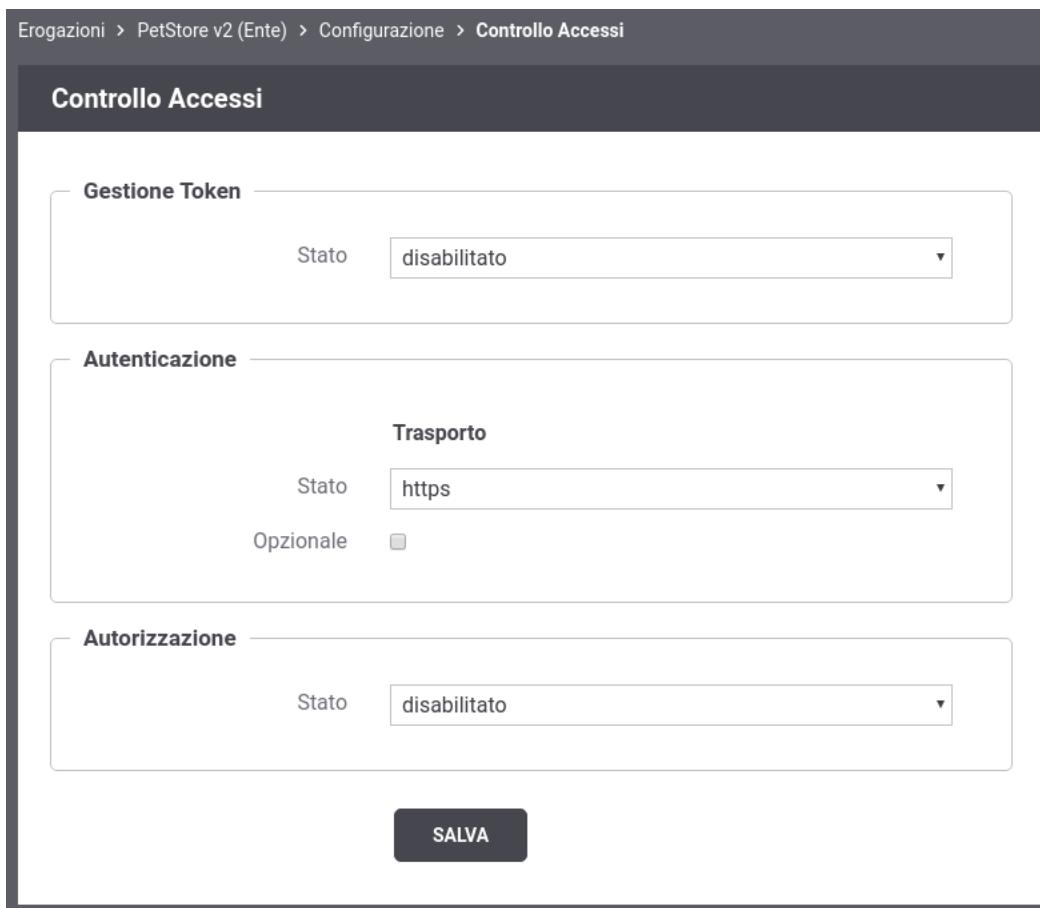


Fig. 4.1: Scenario con autenticazione Https

### • Configurazione Controllo degli Accessi

Per abilitare l'autenticazione “ssl” accedere alla sezione “Erogazioni” e selezionare l'API precedentemente registrata “PetStore v1”. 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*”.



The screenshot shows the 'Controllo Accessi' configuration page. The 'Autenticazione' section is expanded, showing the 'Trasporto' (Transport) configuration. The 'State' dropdown is set to 'https'. The 'Opzionale' (Optional) checkbox is unchecked. The 'Gestione Token' and 'Autorizzazione' sections are collapsed. A 'SALVA' (Save) button is at the bottom.

Fig. 4.2: Configurazione Autenticazione Https

---

**Nota: 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 di Fig. 4.1 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:

---

**Nota: 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 Fig. 4.3 si può vedere come il subject del certificato client utilizzato dal chiamante sia stato associato alla traccia.

### Informazioni Mittente

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

Fig. 4.3: 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 Fig. 4.4 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)
        ...
        ..
```

I criteri di ricerca descritti nella Fig. 4.5 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”.

- *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",
```

(continues on next page)

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=goway.org, CN=test Client 1

**Filtro Dati Transazione**

Esito: [Qualsiasi]

Dettaglio Esito: [Qualsiasi]

Evento: [ ]

**NUOVA RICERCA** **FILTRA RISULTATI** **RIPULISCI**

Fig. 4.4: Ricerca di transazioni con mittente identificato fornendo l'intero subject del certificato client

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 \*: test Client 1

Filtro Dati Transazione

Esito: [Qualsiasi]

Dettaglio Esito: [Qualsiasi]

Evento:

**NUOVA RICERCA** **FILTRA RISULTATI** **RIPULISCI**

Fig. 4.5: Ricerca di transazioni con mittente identificato fornendo una parte del subject del certificato client

(continua dalla pagina precedente)

```

    "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 Fig. 4.6 si può vedere come le transazioni generate dopo la configurazione sopra indicata sono terminate con errore con esito *Autenticazione Fallita*.

| Lista Transazioni: record [1 - 5] |                                    |            |                        |          |           |             |         |
|-----------------------------------|------------------------------------|------------|------------------------|----------|-----------|-------------|---------|
|                                   | Data Ingresso Richiesta            | Tipologia  | Esito                  | Fruitore | Erogatore | API         | Azione  |
| <input type="checkbox"/>          | <span>●</span> 2018-12-14 11:25:23 | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <span>●</span> 2018-12-14 11:25:22 | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <span>●</span> 2018-12-14 11:25:21 | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <span>●</span> 2018-12-14 11:25:20 | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet |
| <input type="checkbox"/>          | <span>●</span> 2018-12-14 11:08:21 | Erogazione | Autenticazione Fallita |          | Ente      | PetStore v2 | PUT_pet |

Fig. 4.6: 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.

| Lista 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 |

ESPORTA

Fig. 4.7: 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_u
˓→test:123456 \
-H "accept: application/json" \
-H "Content-Type: application/json" \

```

(continues on next page)

(continua dalla pagina precedente)

```
-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"
}
```

## 4.1 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 XXX analisiStatistica).

Nella Fig. 4.8 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.

Di seguito viene descritto come realizzare lo scenario di Fig. 4.8:

- **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”.

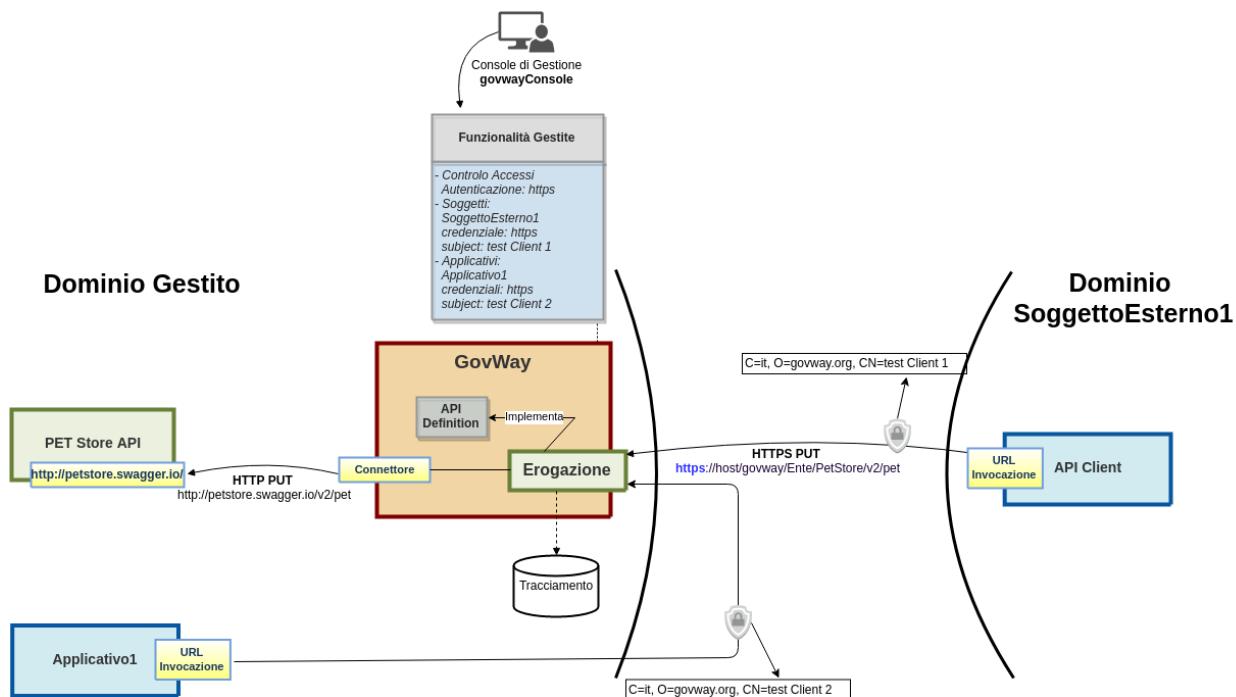


Fig. 4.8: Scenario con autenticazione Https e identificazione dei mittenti

- *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 “*DOCK-ER\_DIR/pki/esempi/test\_Client\_1/ee\_test\_Client\_1.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_1/ee_test_Client_1.cert.
              -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)
    ...
    
```

Soggetti > Aggiungi

Note: (\*) Campi obbligatori

**Soggetto**

Nome \* SoggettoEsterno1

Tipologia Fruitore

Descrizione

**Modalità di Accesso**

Tipo https

Subject \* C=it, O=govway.org, CN=test Client 1

**SALVA**

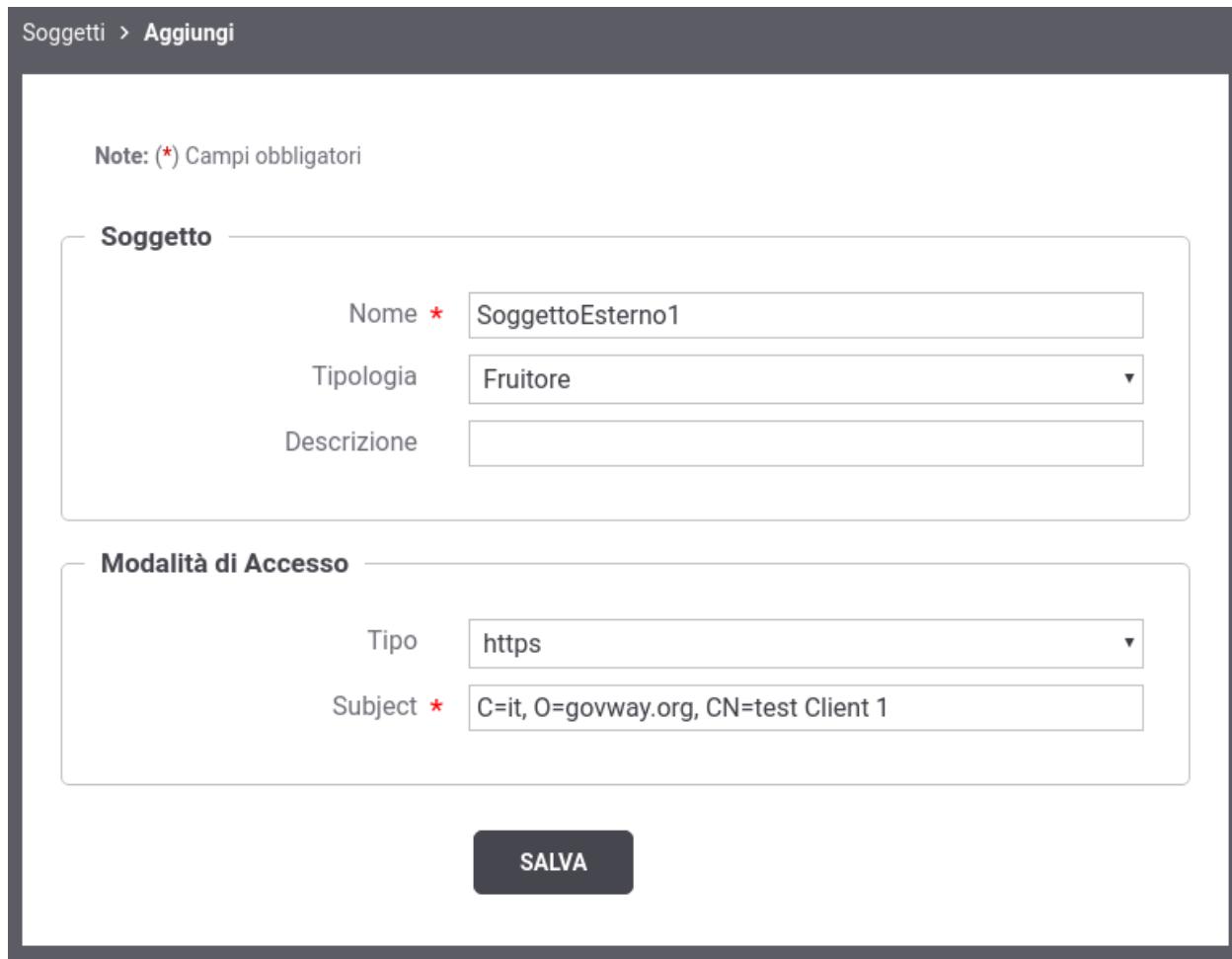


Fig. 4.9: Registrazione nuovo Soggetto

- **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 “*DOCK-ER\_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)
        ....
```

- *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"
}'
```

Applicativi > Aggiungi

Note: (\*) Campi obbligatori

**Applicativo**

Nome \* Applicativo1

**Modalità di Accesso**

Tipo https

Subject \* C=it, O=govway.org, CN=test Client 2

**SALVA**

Fig. 4.10: Registrazione nuovo Applicativo

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 Fig. 4.11 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 the "Dettaglio Transazione" (Transaction Detail) page in the govwayMonitor interface. The top navigation bar includes "Storico", "Intervallo Temporale", and "Dettaglio Transazione". The main content is organized into four sections:

- Informazioni Generali** (General Information):
 

|                        |  |
|------------------------|--|
| Tipologia              | Erogazione (API Gateway)                             |
| Fruitore               | <a href="#">SoggettoEsterno1</a>                     |
| Erogatore              | Ente   |
| API                    | PetStore v2  |
| Azione                 | PUT_pet  |
| Profilo Collaborazione | Sincrono   |
| Esito                  | Ok   |
| Diagnostici            | <a href="#">Visualizza</a>   <a href="#">Esporta</a> |
- 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  |

Fig. 4.11: 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”.

- *Invocazione con certificato ssl “test Client 2”.*

Storico > Intervallo Temporale

### Intervallo Temporale

**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]

Evento: [ ]

**NUOVA RICERCA** **FILTRA RISULTATI** **RIPULISCI**

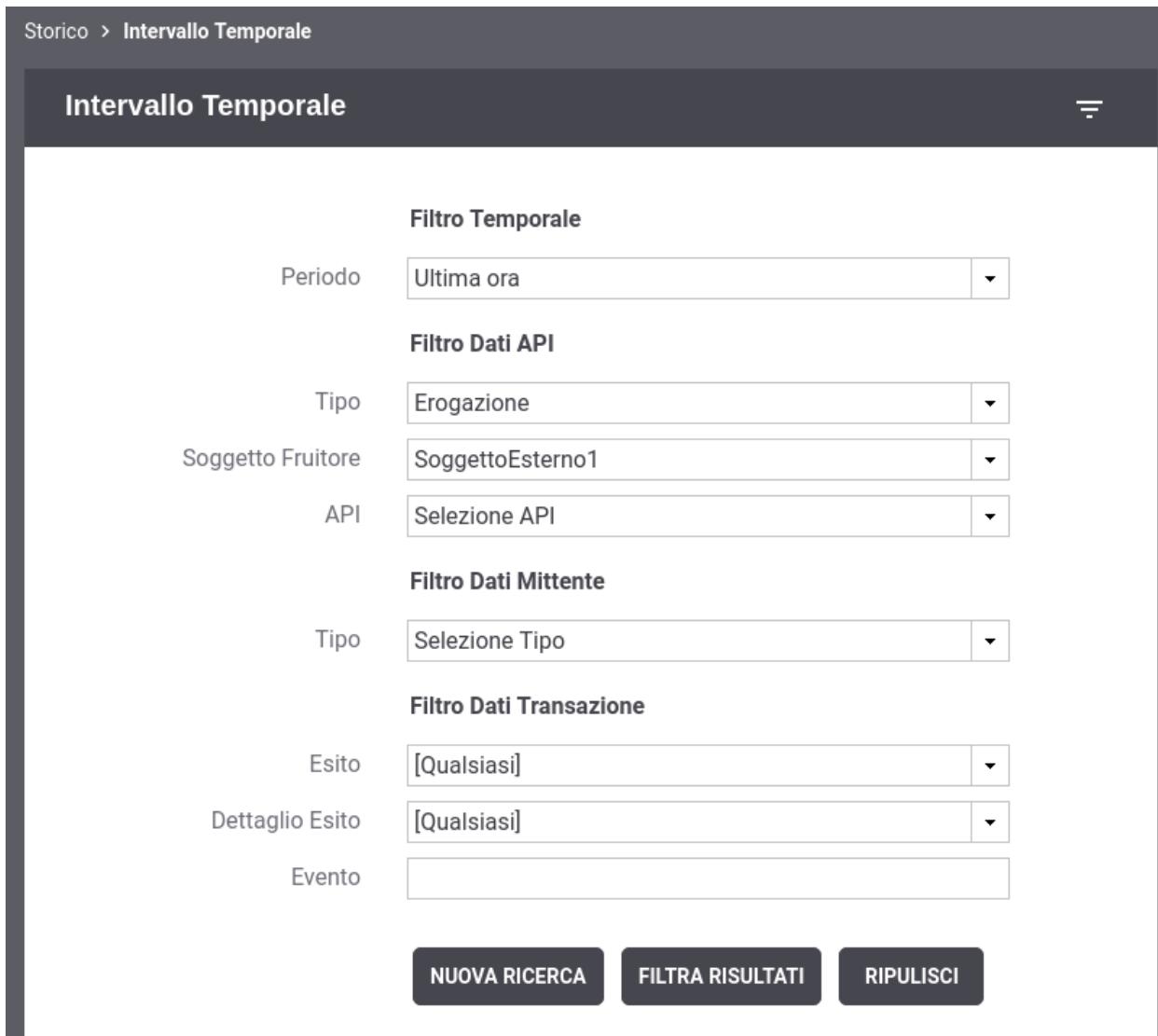


Fig. 4.12: Ricerca di transazioni di un soggetto mittente

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 Fig. 4.13 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 Mittente

|                        |  |
|------------------------|--|
| 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  |

Fig. 4.13: 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

Storico > Intervallo Temporale

## Intervallo Temporale

Filtro Temporale

Periodo: Ultima ora

Filtro Dati API

Tipo: Erogazione

Soggetto Fruitore: Ente

API: Selezione API

Filtro Dati Mittente

Tipo: Applicativo

Applicativo \*: Applicativo1

Filtro Dati Transazione

Esito: [Qualsiasi]

Dettaglio Esito: [Qualsiasi]

Evento:

**NUOVA RICERCA** **FILTRA RISULTATI** **RIPULISCI**

Fig. 4.14: Ricerca di transazioni di un applicativo mittente