Interface Agreement Ditte Registrazione Consegna

Regione CAMPANIA

Documento di Specifiche Funzionali

Titolo Documento: Interface Agreement ditte Registrazione Consegna

Versione: 1.10

Data: 10-10-2022

Enterprise Services Italia S.r.l. (DXC Technology)
KPMG Advisory S.p.A.
Exprivia S.p.A.
Dedagroup Public Services S.r.l.
Dedagroup S.p.A.
Data Management PA S.p.A.
SIAV S.p.A.

Tabella Revisioni Documento

Versione	Data	Elabora	Descrizione
1.0	28-12-2021	DXC	Versione iniziale - nascita del documento
1.1	02-02-2022	DXC	Modifiche della versione effettuate: • Paragrafo “Registrazione evento di consegna”; • request.
1.2	24-02-2022	DXC	Modifiche della versione effettuate: • Paragrafo 3.1.2 “registrazione evento di consegna ossigeno liquido”; • swagger.
1.3	10-03-2022	DXC	Modifiche della versione effettuate: • aggiunto un messaggio di errore; • resa obbligatoria la bolla di consegna.
1.4	20-04-2022	DXC	Specificati gli errori nelle sezioni “Scenario alternativo n. 1”, “Scenario alternativo n. 2” e “Scenario alternativo n. 3” del paragrafo 3.1.2.
1.5	16-05-2022	DXC	Inserito nuovo paragrafo 3.1.3 “registrazione evento di consegna concentratore”.
1.6	10-06-2022	DXC	Aggiornato paragrafo 3.1.3 “registrazione evento di consegna concentratore”.
1.7	20-06-2022	DXC	Modifiche paragrafo 3.1.3 “registrazione evento di consegna concentratore”: campo minsan non obbligatorio
1.8	26-07-2022	DXC	Modifiche paragrafo 3.1.2 “registrazione evento di consegna ossigeno liquido”: aggiunto errore bloccante “DATA CONSEGNA SUCCESSIVA ALLA DATA FINE PT” Modifiche paragrafo 3.1.3 “registrazione evento di consegna concentratore”: aggiunto errore bloccante “DATA CONSEGNA SUCCESSIVA ALLA DATA FINE PT”
1.9	12-09-2022	DXC	Modifiche paragrafo 3.1.2 “registrazione evento di consegna ossigeno liquido”: aggiunto errore bloccante “IMPOSSIBILE CARICARE UN DDT GIA' PRESENTE SUL SISTEMA” Modifiche paragrafo 3.1.3 “registrazione evento di consegna concentratore”: aggiunto errore bloccante “IMPOSSIBILE CARICARE UN DDT GIA' PRESENTE SUL SISTEMA”
1.10	10-10-2022	DXC	Modificato endpoint “Registrazione evento di consegna ossigeno liquido”

1 Sommario

2 Generalità

2.1 Scopo e ambito del documento

Il seguente documento definisce e specifica i casi d'uso ed i relativi endpoint di integrazione per la registrazione di un evento di consegna di ossigeno liquido e di concentratori da parte della ditta fornitrice assegnata al piano terapeutico di ossigenoterapia domiciliare.

2.2 Definizioni e acronimi

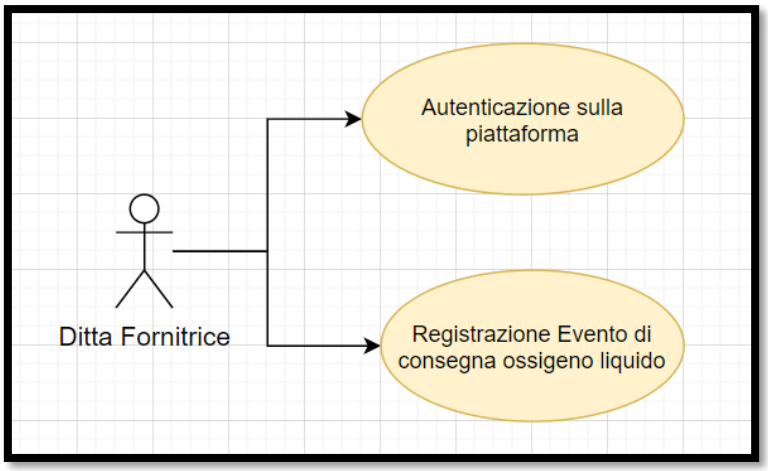
Definizione/Acronimo	Descrizione
PT	Piano Terapeutico
JWT	Json Web Token

3 Specifiche funzionali

3.1 Descrizione degli Use Case

Nella tabella che segue viene riportato l'elenco degli use case della componente applicativa. In particolare, per ciascun caso d'uso, viene riportata la denominazione ed una descrizione dello scopo. Per i casi d'uso complessi viene riportato il flusso delle operazioni effettuate.

Rif.	Caso d'uso	Descrizione
1	Autenticazione sulla piattaforma	La ditta si autentica sulla piattaforma tramite nome utente e password
2	Registrazione evento di consegna ossigeno liquido	Registra un evento di consegna di ossigeno liquido sulla piattaforma, e restituisce il codice autorizzazione distretto, contestualmente ad eventuali messaggi di errore o di warning.
3	Registrazione evento di consegna concentratore	Registra un evento di consegna di concentratore sulla piattaforma, e restituisce il codice autorizzazione distretto, contestualmente ad eventuali messaggi di errore o di warning.



3.1.1 Autenticazione sulla Piattaforma

Servizio di autenticazione che prendendo in input username, password e nome dell'applicazione (**OSSITER**) restituisce un token JWT di autenticazione.

Il token deve poi essere utilizzato successivamente inserendolo nell'header Authorization con il prefisso "Bearer" per tutti gli scenari di integrazione proposti nel documento.

Metodo: POST

Endpoint: <base-path>/sec-auth/auth

Input:

```
{
        "username":"Esempio Username",
        "password":"Esempio Password",
        "application":"OSSITER "
}
```

Output:

{"data":{"access_token":"***contenuto del token jwt***"}}

Scenario alternativo n. 1

Il sistema restituisce un errore di sistema per nome utente o password errati

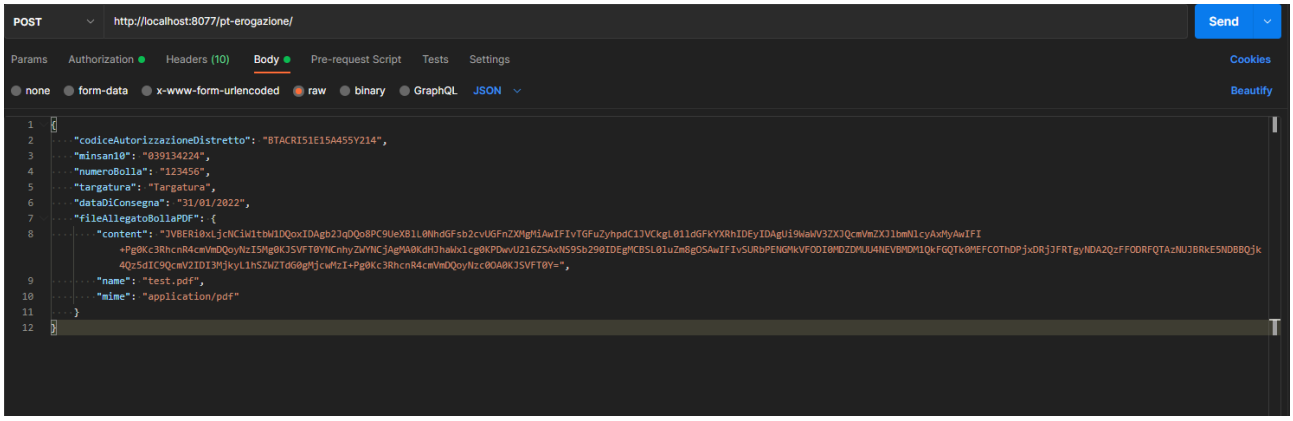
```
{"apierror":{
        "statusCode":"401 UNAUTHORIZED",
        "message":"Credenziali non valide",
        "timestamp":"22-12-2021 10:17:05",
}}
```

3.1.2 Registrazione evento di consegna ossigeno liquido

Servizio di scrittura che ha lo scopo di registrare un evento di consegna ossigenoterapia liquido effettuato dalla ditta assegnata al PT. E' possibile anche allegare il file della bolla in formato PDF. Come specificato nel punto 3.1.1 la richiesta viene autenticata dal bearer token inserito nell'header "authorization".

Si tratta inoltre di un endpoint che in input consuma "application/json" e produce come output "application/json".

La request deve essere specificata inserendo il Json di input di cui i dettagli sono indicati in calce. Si allega screenshot di esempio per il test della chiamata fatta da postman.



Metodo: POST

Endpoint: <base-path>/pt-erogazione/

Input:

- **codiceAutorizzazioneDistretto(obbligatorio)**: Codice autorizzazione distretto del piano terapeutico (che viene comunicato tramite canale mail alla ditta nel momento in cui viene assegnato dal distretto il piano terapeutico alla ditta fornitrice).
- **minsan10 (obbligatorio):** Codice del ministero della salute identificativo del farmaco consegnato; lunghezza massima di 9 caratteri.
- **numeroBolla(obbligatorio)**: Numero della bolla della consegna; lunghezza massima di 100 caratteri.
- **targatura(opzionale)**: targatura della consegna, ossia codice univoco riportato sul bollino adesivo, deve essere trasmesso come stringa composta da esattamente 10 caratteri.
- **dataDiConsegna(obbligatorio)**: data di avvenuta consegna del farmaco in formato dd/MM/yyyy

- **fileAllegatoBollaPDF (obbligatorio):** file allegato bolla PDF, che a sua volta è una struttura dati così composta:
    - **content (obbligatorio):** stringa rappresentante il contenuto del file in formato Base64
    - **name (obbligatorio):** nome del file comprensivo di estensione (es. “testFile.pdf”)
    - **mime (obbligatorio):** mime-type del file (sempre valorizzato con “application/pdf”)

Output:

- **codiceAutorizzazioneDistretto**: lo stesso codice autorizzazione distretto fornito in input
- **protocollo**: stringa che contiene due progressivi separati da un underscore. Il primo è l’identificativo della ditta che ha registrato la consegna, il secondo è l’identificativo univoco della consegna. Potrebbe contenere eventuali messaggi di warning, il cui valore è specificato nella sezione scenari alternativi n.1.

Esempio Request:

```
{
      "codiceAutorizzazioneDistretto": "M2220536A014",
      "minsan10": "039134224",
      "numeroBolla": "123456",
      "targatura": "Targatura",
      "dataDiConsegna": "22/03/2022",
      "fileAllegatoBollaPDF": {,
            "content ": "……stringa formato base 64……",
            "name": "testfile.pdf",
            "mime": "application/pdf",
      }
}
```

Esempio di Response :

```
{
      "codiceAutorizzazioneDistretto": "M2220536A014",
```

```
        "protocollo": "12_1245"
}
```

Scenario alternativo n. 1

Response con protocollo bloccanti per le quali non vengono inserite le erogazioni sul sistema:
- ERR - COD. AUTORIZZAZIONE DISTRETTO NON PRESENTE IN PT
- ERR - FABBISOGNO ESAURITO
- ERR - DATA CONSEGNA SUCCESSIVA ALLA DATA FINE PT
- ERR - COD AUTORIZZAZIONE DISTRETTO NON CONGRUENTE CON LA DATA CONSEGNA
- ERR - DATA CONSEGNA PRECEDENTE ALLA DATA INIZIO PT
- ERR - GIA' EFFETTUATA CONSEGNA IN DATA ODIERNA
- ERR - IMPOSSIBILE CARICARE UN DDT GIA' PRESENTE SUL SISTEMA

Response con protocollo non bloccanti per le quali vengono inserite le erogazioni sul sistema:
- ERR - CONSEGNA REGISTRATA - CONSEGNA URGENTE
- ERR - CONSEGNA REGISTRATA - PENULTIMA CONSEGNA POSSIBILE
- ERR - CONSEGNA REGISTRATA - ULTIMA CONSEGNA POSSIBILE

Formato:

```
{
   "codiceAutorizzazioneDistretto": "string",
   "protocollo": "string"
}
```

Scenario alternativo n. 2

Il sistema ritorna un errore con codice http 400 se qualcuno dei campi obbligatori non è inserito oppure se l'erogazione degli ordini non è abilitata.

Lista errori client/server side:
- HTTP 400 Bad Request
- HTTP 409 Business Exception:
    - Elenco businessErrorCode:
        - 404 NOT_FOUND
        - ERROR_DITTA
        - ERROR_RIATTIVAZIONE
        - SERVER_ERROR
        - ERROR_NOT_FOUND_CAD

Tutti gli errori Client/Server Side sono nel seguente formato:

```
{
        "businessErrorCode": "string",
        "codRichiesta": "string",
        "debugMessage": "string",
```

```
        "message": "string",
        "spanId": "string",
        "statusCode": "string",
        "subErrors": [
                {}
        ],
        "timestamp": "dd-MM-yyyy HH:mm:ss",
        "traceId": "string"
}
```


Nel Message è definito il motivo dell'errore;
Il debugMessage ed i subErrors possono essere valorizzati con il dettaglio dell'errore.

Esempio di accesso non autorizzato:


```
{
   "apierror": {
      "businessErrorCode": null,
      "statusCode": "403 FORBIDDEN",
      "message": "Access is denied",
      "timestamp": "15-04-2022 08:28:16",
      "debugMessage": "AccessDeniedException: Access is denied",
      "subErrors": null,
      "traceId": "3e28985b3faadd0f",
      "spanId": "3e28985b3faadd0f",
      "codRichiesta": null
   }
}
```


Esempio di Business Exception 409 CONFLICT:


```
{
   "apierror": {
      "businessErrorCode": "ERROR_DITTA",
      "statusCode": "409 CONFLICT",
      "message": "Utente non autorizzato ad effettuare l'erogazione dell'ordine. La ditta associata al piano terapeutico non corrisponde alla ditta dell'utente che sta operando. Contattare il supporto tecnico.",
      "timestamp": "15-04-2022 08:30:27",
      "debugMessage": null,
      "subErrors": null,
      "traceId": "c4a4eda6c333c13a",
      "spanId": "c4a4eda6c333c13a",
      "codRichiesta": null
   }
}
```


Esempio errore validazione campi in input 400 BAD_REQUEST:
errore inserimento campo targatura:


```
{
   "apierror": {
```

```
        "businessErrorCode": "500",
        "statusCode": "400 BAD_REQUEST",
        "message": "Errore di Validazione",
        "timestamp": "15-04-2022 08:39:23",
        "debugMessage": "Validation failed for argument [0] in public
org.springframework.http.ResponseEntity<com.dxc.sgisanita.coreservices.apiexternal.controller.dto.Dett
aglioErogazioneResponseDTO>
com.dxc.sgisanita.coreservices.apiexternal.controller.PtErogazioneController.savePtErogazione(com.dxc.s
gisanita.coreservices.apiexternal.controller.dto.DettaglioErogazioneRequestDTO): [Field error in object
'dettaglioErogazioneRequestDTO' on field 'targatura': rejected value [Tureg888]; codes
[Size.dettaglioErogazioneRequestDTO.targatura,Size.targatura,Size.java.lang.String,Size]; arguments
[org.springframework.context.support.DefaultMessageSourceResolvable: codes
[dettaglioErogazioneRequestDTO.targatura,targatura]; arguments []; default message [targatura],10,10];
default message [size must be between 10 and 10]] ",
        "subErrors": [
          {
            "object": "dettaglioErogazioneRequestDTO",
            "field": "targatura",
            "rejectedValue": "Tureg888",
            "message": "size must be between 10 and 10"
          }
        ],
        "traceId": "7e629d6b62fb2817",
        "spanId": "7e629d6b62fb2817",
        "codRichiesta": null
    }
}
```


Nel message è definito il motivo dell'errore: "Errore di Validazione".
Nei subError sono presenti i dettagli dell'errore:

- subErrors.field: specifica il campo che è non ha passato la validazione;
- subErrors.rejectedValue: specifica il valore del campo che non ha passato la validazione;
- subErrors.message: specifica il dettaglio dell'errore.

Scenario alternativo n. 3

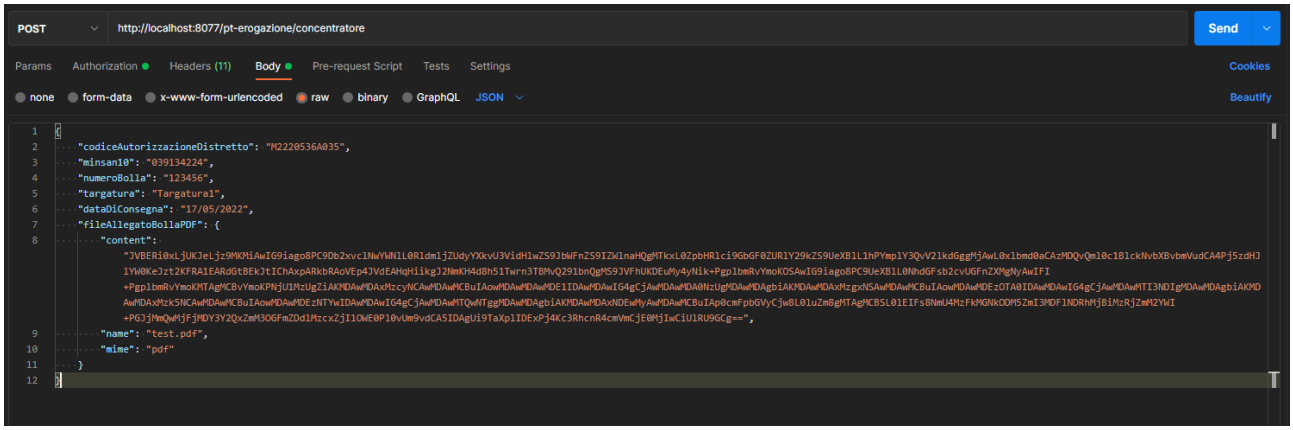
Il sistema ritorna un errore con codice http 500 in caso di errore non gestito.

3.1.3 Registrazione evento di consegna concentratore

Servizio di scrittura che ha lo scopo di registrare un evento di consegna ossigenoterapia concentratore effettuato dalla ditta assegnata al PT. E' possibile anche allegare il file della bolla in formato PDF. Come specificato nel punto 3.1.1 la richiesta viene autenticata dal bearer token inserito nell'header "authorization".

Si tratta inoltre di un endpoint che in input consuma "application/json" e produce come output "application/json".

La request deve essere specificata inserendo il Json di input di cui i dettagli sono indicati in calce. Si allega screenshot di esempio per il test della chiamata fatta da postman.



Metodo: POST

Endpoint: <base-path>/pt-erogazione/concentratore

Input:

- **codiceAutorizzazioneDistretto(obbligatorio)**: Codice autorizzazione distretto del piano terapeutico (che viene comunicato tramite canale mail alla ditta nel momento in cui viene assegnato dal distretto il piano terapeutico alla ditta fornitrice).
- **minsan10 (<u>non obbligatorio</u>):** Codice del ministero della salute identificativo del farmaco consegnato; lunghezza massima di 9 caratteri.
- **numeroBolla(obbligatorio)**: Numero della bolla della consegna; lunghezza massima di 100 caratteri.
- **targatura(opzionale)**: targatura della consegna, ossia codice univoco riportato sul bollino adesivo, deve essere trasmesso come stringa composta da esattamente 10 caratteri.
- **dataDiConsegna(obbligatorio)**: data di avvenuta consegna del farmaco in formato dd/MM/yyyy

- **fileAllegatoBollaPDF (obbligatorio):** file allegato bolla PDF, che a sua volta è una struttura dati così composta:
    - **content (obbligatorio):** stringa rappresentante il contenuto del file in formato Base64
    - **name (obbligatorio):** nome del file comprensivo di estensione (es. “testFile.pdf”)
    - **mime (obbligatorio):** mime-type del file (sempre valorizzato con “application/pdf”)

Output:

- **codiceAutorizzazioneDistretto**: lo stesso codice autorizzazione distretto fornito in input
- **protocollo**: stringa che contiene due progressivi separati da un underscore. Il primo è l’identificativo della ditta che ha registrato la consegna, il secondo è l’identificativo univoco della consegna. Potrebbe contenere eventuali messaggi di warning, il cui valore è specificato nella sezione scenari alternativi n.1.

Esempio Request:

```
{
      "codiceAutorizzazioneDistretto": "M2220536A014",
      "minsan10": "039134224",
      "numeroBolla": "123456",
      "targatura": "Targatura",
      "dataDiConsegna": "22/03/2022",
      "fileAllegatoBollaPDF": {,
            "content ": "……stringa formato base 64……",
            "name": "testfile.pdf",
            "mime": "application/pdf",
      }
}
```

Esempio di Response :

```
{
      "codiceAutorizzazioneDistretto": "M2220536A014",
```

```
        "protocollo": "12_1245"
}
```

Scenario alternativo n. 1

Response con protocollo bloccanti per le quali non vengono inserite le erogazioni sul sistema:
- ERR - COD. AUTORIZZAZIONE DISTRETTO NON PRESENTE IN PT
- ERR - FABBISOGNO ESAURITO
- ERR - DATA CONSEGNA SUCCESSIVA ALLA DATA FINE PT
- ERR - COD AUTORIZZAZIONE DISTRETTO NON CONGRUENTE CON LA DATA CONSEGNA
- ERR - DATA CONSEGNA PRECEDENTE ALLA DATA INIZIO PT
- ERR - IMPOSSIBILE CARICARE UN DDT GIA' PRESENTE SUL SISTEMA

Formato:

```
{
   "codiceAutorizzazioneDistretto": "string",
   "protocollo": "string"
}
```

Scenario alternativo n. 2

Il sistema ritorna un errore con codice http 400 se qualcuno dei campi obbligatori non è inserito oppure se l'erogazione degli ordini non è abilitata.

Lista errori client/server side:
- HTTP 400 Bad Request
- HTTP 409 Business Exception:
    - Elenco businessErrorCode:
        - 404 NOT_FOUND
        - ERROR_DITTA
        - ERROR_RIATTIVAZIONE
        - SERVER_ERROR
        - ERROR_NOT_FOUND_CAD

Tutti gli errori Client/Server Side sono nel seguente formato:

```
{
        "businessErrorCode": "string",
        "codRichiesta": "string",
        "debugMessage": "string",
        "message": "string",
        "spanId": "string",
        "statusCode": "string",
        "subErrors": [
                {}
        ],
        "timestamp": "dd-MM-yyyy HH:mm:ss",
```

```
        "traceId": "string"
}
```


Nel Message è definito il motivo dell'errore;
Il debugMessage ed i subErrors possono essere valorizzati con il dettaglio dell'errore.

Esempio di accesso non autorizzato:


```
{
   "apierror": {
      "businessErrorCode": null,
      "statusCode": "403 FORBIDDEN",
      "message": "Access is denied",
      "timestamp": "15-04-2022 08:28:16",
      "debugMessage": "AccessDeniedException: Access is denied",
      "subErrors": null,
      "traceId": "3e28985b3faadd0f",
      "spanId": "3e28985b3faadd0f",
      "codRichiesta": null
   }
}
```


Esempio di Business Exception 409 CONFLICT:


```
{
   "apierror": {
      "businessErrorCode": "ERROR_DITTA",
      "statusCode": "409 CONFLICT",
      "message": "Utente non autorizzato ad effettuare l'erogazione dell'ordine. La ditta associata al piano terapeutico non corrisponde alla ditta dell'utente che sta operando. Contattare il supporto tecnico.",
      "timestamp": "15-04-2022 08:30:27",
      "debugMessage": null,
      "subErrors": null,
      "traceId": "c4a4eda6c333c13a",
      "spanId": "c4a4eda6c333c13a",
      "codRichiesta": null
   }
}
```


Esempio errore validazione campi in input 400 BAD_REQUEST:
errore inserimento campo targatura:


```
{
   "apierror": {
      "businessErrorCode": "500",
      "statusCode": "400 BAD_REQUEST",
      "message": "Errore di Validazione",
      "timestamp": "15-04-2022 08:39:23",
      "debugMessage": "Validation failed for argument [0] in public org.springframework.http.ResponseEntity<com.dxc.sgisanita.coreservices.apiexternal.controller.dto.DettaglioErogazioneResponseDTO>
```

com.dxc.sgisanita.coreservices.apiexternal.controller.PtErogazioneController.savePtErogazioneConcentratore(com.dxc.sgisanita.coreservices.apiexternal.controller.dto.DettaglioErogazioneRequestDTO): [Field error in object 'dettaglioErogazioneRequestDTO' on field 'targatura': rejected value [Tureg888]; codes [Size.dettaglioErogazioneRequestDTO.targatura,Size.targatura,Size.java.lang.String,Size]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [dettaglioErogazioneRequestDTO.targatura,targatura]; arguments []; default message [targatura],10,10]; default message [size must be between 10 and 10]] ",

```
      "subErrors": [
        {
          "object": "dettaglioErogazioneRequestDTO",
          "field": "targatura",
          "rejectedValue": "Tureg888",
          "message": "size must be between 10 and 10"
        }
      ],
      "traceId": "7e629d6b62fb2817",
      "spanId": "7e629d6b62fb2817",
      "codRichiesta": null
   }
}
```


Nel message è definito il motivo dell'errore: "Errore di Validazione".
Nei subError sono presenti i dettagli dell'errore:

- subErrors.field: specifica il campo che è non ha passato la validazione;
- subErrors.rejectedValue: specifica il valore del campo che non ha passato la validazione;
- subErrors.message: specifica il dettaglio dell'errore.

Scenario alternativo n. 3

Il sistema ritorna un errore con codice http 500 in caso di errore non gestito.

3.2 Swagger dei servizi esposti

Si riporta lo swagger dei servizi esposti.

swagger.json