

Manuale Utilizzo WebServices Anagrafe delle Prestazioni 2.0

Aggiornato al 04/07/2022

Sommario

| | |
|---|----------|
| <i>1 Prima fase di test</i> | 3 |
| 2 Specifiche Ambiente di Test | 3 |
| <i>3 Flusso di Utilizzo dei Servizi</i> | 3 |
| 4 Get Configurazione Incarico Consulente | 4 |
| <i>5 Get Configurazione Incarico Dipendente</i> | 5 |
| <i>6 Login Webservice</i> | 6 |
| <i>7 Inserimento Incarico Consulente</i> | 6 |
| <i>8 Inserimento Incarico Dipendente</i> | 6 |
| <i>9 Modifica Incarico Consulente</i> | 7 |
| <i>10 Modifica Incarico Dipendente</i> | 7 |
| <i>11 Cancellazione Incarico Consulente</i> | 7 |
| <i>12 Cancellazione Incarico Dipendente</i> | 7 |
| <i>13 Revoca Incarico Consulente</i> | 7 |
| <i>14 Revoca Incarico Dipendente</i> | 8 |
| 15 Recupero nuovo id Incarico (solo per incarichi migrati) | 8 |
| 16 Tipologia Errori | 8 |

1 Prima fase di test

In questa prima fase saranno autorizzate alcune PA all'utilizzo del web services. Le credenziali saranno fornite dal DFP previa abilitazione della PA all'utilizzo del web services.

Come dettagliato in seguito, le credenziali fornite consisteranno in un AppId e una Secret da utilizzare nel momento in cui si intende operare tramite Web Service.

Il sistema di autenticazione è basato su token (**Bearer Token**) che sarà rilasciato dal sistema al termine della procedura di login sul web services. Tale token dovrà essere utilizzato per operare sugli incarichi e avrà una durata temporanea. Alla scadenza andrà rinnovato con una nuova procedura di login.

2 Specifiche Ambiente di Test

L'ambiente di esecuzione dei test è quello di *collaudo*.

Il webservice è di tipo *REST*.

Base Url: <https://adp-api-coll.dfp.gov.it>

Indirizzo Swagger servizi: <https://adp-api-coll.dfp.gov.it/swagger/index.html>

I test saranno effettuati su base dati non popolate e su sistemi di test.

L'utente dovrà inserire un incarico e poi operare sull'incarico inserito tramite il suo id.

Il sistema è in **beta** ed è soggetto a possibili variazioni nei servizi. Tali modifiche saranno riportate su Swagger.

3 Flusso di Utilizzo dei Servizi

Per effettuare test sul *WebServices* che gestisce gli incarichi di *Adp2.0* occorre seguire i seguenti passi:

1. **Recuperare configurazione incarico:** invocare il servizio "Get Configurazione Incarico (*dipendente o consulente*)" che restituisce l'elenco degli id dei parametri da inserire negli appositi campi delle varie request. La chiamata non prevede autenticazione;

2. **Effettuare login**: invocare il servizio “*Login Web Services*” che restituisce un token da inserire nell’header delle chiamate successive di gestione degli incarichi, effettuando una chiamata post e passando il **secret** e l’**appld** fornito;
3. **Inserimento incarico**: invocare il servizio di inserimento incarico (dipendente o consulente). Viene restituito l’id incarico appena inserito;
4. **Modifica incarico**: invocare il servizio di modifica incarico (dipendente o consulente) passando l’id incarico;
5. **Cancellazione incarico**: invocare il servizio di cancellazione incarico (dipendente o consulente) passando l’id incarico;
6. **Revoca incarico**: invocare il servizio di revoca incarico (dipendente o consulente) passando l’id incarico.

In generale gli **endpoint** da chiamare saranno composti da **<base url> + <indirizzo servizio>**

4 Get Configurazione Incarico Consulente

url: /api/ws/Incarico/Consulente/Configurazione

tipo: GET

parametri: -

autenticato: no

response: lista di oggetti “*chiave - valore*” che rappresentano i vari parametri da poter inserire negli incarichi

Nella response di questa chiamata si ottengono gli elementi da inserire nelle successive chiamate di inserimento degli incarichi.

Nel caso in cui il campo fosse già presente sul vecchio sistema Perla, alla voce “codice” si otterrà l’id corrispondente nel vecchio sistema.

Di seguito un esempio

```
[
  {
    "descrizione": "oggettoIncarico",
    "elementi": [
      {
        "id": "a45de0a1-23b3-4e7f-9cfa-703eb650f7ba",
        "descrizione": "Amministratore di condominio",
        "codice": "72"
      },
      {
        "id": "75e60e6f-469f-45f8-8941-c5c20f14671c",
        "descrizione": "Amministratore di sistemi informatici",
        "codice": "28"
      },
      {
        "id": "96f8c3e1-685b-4f5f-82d7-90f34cefca68",
        "descrizione": "Arbitrato",
        "codice": "3"
      },
      {

```

5 Get Configurazione Incarico Dipendente

url: /api/ws/Incarico/Consulente/Dipendente

tipo: GET

parametri: -

autenticato: no

response: lista di oggetti "chiave - valore" che rappresentano i vari parametri da poter inserire negli incarichi

Nella response di questa chiamata si ottengono gli elementi da inserire nelle successive chiamate di inserimento degli incarichi.

Nel caso in cui il campo fosse già presente sul vecchio sistema Perla, alla voce "codice" si otterrà l'id corrispondente nel vecchio sistema.

```
[
  {
    "descrizione": "oggettoIncarico",
    "elementi": [
      {
        "id": "15cb72ae-2cbb-44f7-b8d5-f0a7004da2b2",
        "descrizione": "Amministratore di condominio",
        "codice": "72"
      },
      {
        "id": "cd691149-0858-464f-ad4a-ac60f552312c",
        "descrizione": "Animazione, Assistenza",
        "codice": "25"
      },
      {
        "id": "d52bb0d7-0cd6-4a46-8d43-7b2f0a1a9bad",
        "descrizione": "Arbitrato",
        "codice": "3"
      },
      {
        "id": "bae66196-69a0-4d91-a313-408c8ee7d50a",
        "descrizione": "Associazioni dilettantistiche",
        "codice": "12"
      }
    ]
  }
],
```

6 Login Webservice

url: /api/ws/auth/login

tipo: POST

parametri: appld (string), secret (string)

response: token da usare per le chiamate autenticate come bearer token

7 Inserimento Incarico Consulente

url: /api/ws/Incarico/Consulente

tipo: POST

request: da swagger

response: id incarico inserito

autenticato: si (Bearer token nell'header)

8 Inserimento Incarico Dipendente

url: /api/ws/Incarico/Dipendente

tipo: POST

request: da swagger

response: id incarico inserito

autenticato: si (Bearer token nell'header)

9 Modifica Incarico Consulente

url: /api/ws/Incarico/Consulente/{incaricoid}

tipo: PUT

parametri: -

request: da swagger

autenticato: si (Bearer token nell'header)

10 Modifica Incarico Dipendente

url: /api/ws/Incarico/Dipendente/{incaricoid}

tipo: PUT

request: da swagger

autenticato: si (Bearer token nell'header)

11 Cancellazione Incarico Consulente

url: /api/ws/Incarico/Consulente/{incaricoid}

tipo: DELETE

parametri: appld (string), secret (string)

request: da swagger

autenticato: si (Bearer token nell'header)

12 Cancellazione Incarico Dipendente

url: /api/ws/Incarico/Dipendente/{incaricoid}

tipo: DELETE

parametri: appld (string), secret (string)

request: da swagger

autenticato: si (Bearer token nell'header)

13 Revoca Incarico Consulente

url: /api/ws/Incarico/Consulente/Revoca/{incaricoid}

tipo: POST

request: da swagger

autenticato: si (Bearer token nell'header)

14 Revoca Incarico Dipendente

url: /api/ws/Incarico/Dipendente/Revoca/{incaricold}

tipo: POST

request: da swagger

autenticato: si (Bearer token nell'header)

Le richieste di esempio sono documentate all'interno dello swagger condiviso con il quale è possibile effettuare delle chiamate di test.

15 Recupero nuovo id Incarico (solo per incarichi migrati)

url: /api/ws/Incarico/Dipendente/{idOldIncarico}

tipo: GET

request: da swagger

autenticato: si (Bearer token nell'header)

url: /api/ws/Incarico/Consulente/{idOldIncarico}

tipo: GET

request: da swagger

autenticato: si (Bearer token nell'header)

Il metodo restituisce l'id incarico del nuovo sistema PerlaPA a partire dall'id del vecchio incarico. Questo endpoint funzionerà solo per gli incarichi che sono stati migrati dal vecchio sistema. In restituzione si avrà una stringa di tipo GUID (es. 4b157db5-9fe0-45c8-9532-a77e0358cf73).

Le richieste di esempio sono documentate all'interno dello swagger condiviso con il quale è possibile effettuare delle chiamate di test.

16 Tipologia Errori

Il sistema genera errori ben precisi a seconda del livello di errore che si ottiene.

Il codice http restituito dalla chiamata Rest indica l'esito dell'operazione rispettando gli standard di tale tecnologia.

Errori formali di scrittura della request

In questo caso si otterrà un codice 400 con un oggetto indicante il tipo di errori che si ottengono nella request effettuata (gli errori sono presenti nel campo "errors").

Di seguito un esempio di errore formale nella request

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "00-fd6e145588aea240a1e192541aea3162-47fb35cd1c96ba45-00",
  "errors": {
    "$oggettoIncaricoDipendenteId": [
      "The JSON value could not be converted to System.Guid. Path: $.oggettoIncaricoDipendenteId | LineNumber: 13 |
      BytePositionInLine: 72."
    ]
  }
}
```

Errori applicativi

In questo caso si otterrà un codice 400 o 500 con in risposta la lista degli errori
Di seguito un esempio

```
{
  "instance": "/api/ws/Incarico/Dipendente",
  "status": 400,
  "typeException": "BusinessLogicException",
  "errorCodes": [
    "TipologiaNormaNotFound"
  ]
}
```

Nell'elenco seguente riportiamo i possibili errori

- ApplIdNotValid
- SecretNotValid
- AmministrazioneNotFound
- UnitaOrganizzativeNotFound
- AreeOrganizzativeOmogeneeNotFound
- AmministrazioneCessata
- Unauthorized
- DataFineNotValid
- PersonaFisicaNotFound
- PersonaGiuridicaNotFound
- FileCvNotValid
- FileDichiarazioneSvolgimentoAltriIncarichiNotValid
- ServizioIstituzionePubblicaNotFound

OggettoIncaricoNotFound
AmbitoTematicoNotFound
PercettoreNotValid
PercettorePersonaFisicaNotValid
TipologiaNormaNotFound
IncaricoNotFound