



Indagor Service

• ST032 •

ACQUISIZIONE PUNTUALE DATI DI TITOLI DI STUDIO VIA CODICE FISCALE CITTADINO

Specifiche di Integrazione

INDICE

Descrizione.....	1
Contenuto della Richiesta.....	2
Contenuto delle Eccezioni.....	3
Contenuto della Risposta.....	4
Esempi di Integrazione.....	5
API Java.....	5
API .Net (C#).....	7
API Javascript.....	9

ST032

ACQUISIZIONE PUNTUALE DATI DI TITOLI DI STUDIO VIA CODICE FISCALE CITTADINO

DESCRIZIONE

Il servizio consente, dato il Codice Fiscale o l'ID ANPR di un soggetto, di acquisire da ANIST in maniera puntuale il dettaglio dei titoli di studio conseguiti.

In generale

- Denominazione titolo
- Dati Istituto
- Anno conseguimento titolo
- Voto finale

Qui di seguito è riportato il contenuto

- Della Richiesta
- Delle Eccezioni
- Della Risposta

Segue un esempio di integrazione completo per:

- Ambiente di sviluppo JAVA
- Ambiente di sviluppo NET
- Javascript

CONTENUTO DELLA RICHIESTA

Classe Client:

Costruttore:

Client(String hostport, String utente, String password)

- **hostport:** radice dell'end point
 - Ambiente di test: <http://84.240.190.236>
 - Ambiente di produzione: <https://gateway1.steseiservizi.it>
- **utente:** Identificativo Gateway assegnato da Stesei all'applicativo per l'autenticazione delle chiamate effettuate
 - Ambiente di test: 040007
- **password:** password Gateway associata all'identificativo: la password è gestita dal produttore dell'applicativo attraverso il portale INDAGOR (<http://www.steseiservizi.it/indagor-pdnd>)
 - Ambiente di test: "INIZIALE"

Metodi:

Richiesta per Codice Fiscale

RespST032 ST032(string comunelstat, string codiceFiscale, string idANPR)

- **ST032:** Nome dell'Indagor Service richiamato
- **comunelstat:** Codice Istat del Comune fruitore dell'applicativo che inoltra la chiamata
 - Ambiente di test: Codice Istat 040007
- **codiceFiscale:** Codice Fiscale del Cittadino
 - Ambiente di test: Codice Fiscale (Esempio [PRFTST80A11H501U](#))
- **idANPR:** Id ANPR del Cittadino (in alternativa al Codice Fiscale Cittadino)

CONTENUTO DELLE ECCEZIONI

Eccezioni restituite da tutti i metodi del Gateway INDAGOR:

- **ConnectionException**: errore di connessione o di server non attivo
- **ServerException**: errori interni del server
- **AuthenticationException**: errore di autenticazione (utente e/o password errati)
- **PermissionDeniedException**:
 - operazione non permessa (es: comune non censito nel Gateway o Codice Istat del Comune vuoto)
- **RequestException**:
 - Se i parametri della ricerca inseriti non sono corretti:
 - "Il parametro Codice Istat ha un formato non valido"
 - "Il parametro Codice Fiscale del Cittadino ha un formato non valido"
 - Se manca il criterio di ricerca
 - "Criteri di ricerca non validi"
- **ANISTException**: errori e anomalie restituite da ANIST a fronte della ricerca effettuata

CONTENUTO DELLA RISPOSTA

Di seguito sono riportati i dati che compongono la risposta resa dal servizio ST032; se i valori della risposta sono pari a:

- **null**
- **stringa vuota**
- **LocalDate.MIN**

significa che tali dati, per la posizione anagrafica richiesta, non sono presenti in ANIST causa:

- Mancato inserimento del dato da parte del Comune (errore o dato non disponibile)
- Evento non avvenuto (quindi ovviamente privo dei dati)

Descrizione	Note
Presenza Titoli	Booleano (true o false)
Denominazione Titolo	String
Codice Titolo	String
Codice Istituto Principale	String
Denominazione Istituto Principale	String
Codice Meccanografico	Numero intero (ad es. 4)
Denominazione Plesso	String
Anno scolastico	Numero intero
Anno conseguimento Titolo	Numero intero
Voto finale	Numero intero
Flag lode	String

ESEMPI DI INTEGRAZIONE

API Java

Si aggiunge al progetto la libreria:

- **steseigateway-1.0.0-jar-with-dependencies.jar**

O in alternativa le seguenti librerie:

- **steseigateway-1.0.0.jar**
- **jackson-core-2.12.1.jar**
- **jackson-databind-2.12.1.jar**
- **jackson-annotations-2.1.2.jar**

Si richiama nel seguente modo:

```
package main;
import stesei.gateway.*;
import java.io.IOException;
import java.time.LocalDate;
public class Test{
    public static void main(String[] args){
        TestST032();
    }
    static void TestST032()
    {
        Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
        RespST032 resp=null;
        Try
        {
            resp = cl.ST032("040007", "PRFTST80A11H501U", "");
        }
        catch (ConnectionException econn)
        {
            String msg=econn.getMessage(); //errore di connessione al Gateway
            System.out.println(msg);
        }
        catch (ServerException eserver)
        {
            String msg=eserver.getMessage(); //errore interno del server
            System.out.println(msg);
        }
        catch (AuthenticationException eauth)
        {
            String msg=eauth.getMessage(); //utente e/o password errati
            System.out.println(msg);
        }
    }
}
```

```
catch (PermissionDeniedException edenied)
{
    String msg=edenied.getMessage(); //operazione non permessa (es: comune non disponibile)
    System.out.println(msg);
}
catch (RequestException erequest)
{
    String msg=erequest.getMessage(); //dati mancanti nella richiesta o richiesta vuota
    System.out.println(msg);
}
catch (ANISTException eanist)
{
    String msg=eanist.getMessage(); //errori, anomalie restituiti da ANIST
    System.out.println(msg);
}

// risposta
if (resp != null)
{
    boolean presenzaTitoli = resp.isPresenzaTitoli();
    for(TitoloStudenteDettaglio titolo : resp.getListaTitoli())
    {
        String denominazioneTitolo = titolo.getDenominazioneTitolo();
        String codiceTitolo = titolo.getCodiceTitolo();
        String codicelstitutoPrincipale = titolo.getCodicelstitutoPrincipale();
        String denolstitutoPrincipale = titolo.getDenolstitutoPrincipale();
        String codiceMeccanografico = titolo.getCodiceMeccanografico();
        String denominazionePlesso = titolo.getDenominazionePlesso();
        int annoScolastico = titolo.getAnnoScolastico();
        int annoConseguimentoTitolo = titolo.getAnnoConseguimentoTitolo();
        int votoFinale = titolo.getVotoFinale();
        String flagLode = titolo.getFlagLode();
    }
}
}
```

API .Net (C#)

Si aggiunge al progetto la libreria:

- **SteseiGateway.dll**

Si richiama nel seguente modo:

```
static void TestST032()
{
    Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
    RespST032 resp = null;
    Try
    {
        resp = cl.ST032("040007", "PRFTST80A11H501U", "");
    }
    catch (ConnectionException econn)
    {
        string msg = econn.Message; //errore di connessione
    }
    catch (ServerException eserver)
    {
        string msg = eserver.Message; //errore interno del server
    }
    catch (AuthenticationException eauth)
    {
        string msg = eauth.Message; //utente e/o password errati
    }
    catch (PermissionDeniedException edenied)
    {
        string msg = edenied.Message; //operazione non permessa (es: comune non disponibile)
    }
    catch (RequestException erequest)
    {
        string msg = erequest.Message; //dati mancanti nella richiesta o richiesta vuota
    }
    catch (ANISTException eanist)
    {
        string msg = eanist.Message; //errori, anomalie restituiti da ANIST
    }
}
```

```
//risposta
if (resp != null)
{
    bool presenzaTitoli = resp.presenzaTitoli;
    foreach(TitoloStudianteDettaglio titolo in resp.listaTitoli)
    {
        string denominazioneTitolo = titolo.denominazioneTitolo;
        string codiceTitolo = titolo.codiceTitolo;
        string codiceIstitutoPrincipale = titolo.codiceIstitutoPrincipale;
        string denoIstitutoPrincipale = titolo.denoIstitutoPrincipale;
        string codiceMeccanografico = titolo.codiceMeccanografico;
        string denominazionePlesso = titolo.denominazionePlesso;
        int annoScolastico = titolo.annoScolastico;
        int annoConseguimentoTitolo = titolo.annoConseguimentoTitolo;
        int votoFinale = titolo.votoFinale;
        string flagLode = titolo.flagLode;
    }
}
}
```

API Javascript

```
let endpoint= "http://84.240.190.236/api/ST032";
```

```
var criteri={comunelstat: "040007", codiceFiscale: "PRFTST80A11H501U", idANPR:""};
```

```
var account={utente: "040007", password: "INIZIALE"};
```

```
var richiesta={credenzialiAccesso: account, criteriRicerca: criteri};
```

```
$.ajax  
(  
  type: "POST",  
  url: endpoint,  
  data: JSON.stringify(richiesta),  
  contentType: "application/json; charset=utf-8",  
  dataType: "json",  
  success: function (risposta)  
  {  
    showJson(risposta);  
  },  
  error: function (qXHR, textStatus, errorThrown)  
  {  
    alert("ERR");  
  }  
);
```

```
let risposta=  
{  
  "errore":  
  {  
    "errore": false,  
    "tipoErrore": "",  
    "messaggio": ""  
  },  
  "response":  
  {  
    "presenzaTitoli": true,  
    "listaTitoli":  
    [  
      {  
        "denominazioneTitolo": "ISTITUTO TECNICO SETTORE ECONOMICO INDIRIZZO TURISMO",  
        "codiceTitolo": "IT",  
        "codiceIstituitoPrincipale": "BSTD110006",  
        "denominazioneIstituitoPrincipale": "ASTOLFO LUNARDI",  
        "codiceMeccanografico": "BSTD024018",  
        "denominazionePlesso": "LUNARDI - BRESCIA",  
        "annoScolastico": 202526,  
        "annoConseguimentoTitolo": 2026,  
        "votoFinale": 69,  
        "flagLode": "N"  
      },  
    ],  
  },  
}
```

```
{
  "denominazioneTitolo": "Diploma conclusivo del primo ciclo di istruzione",
  "codiceTitolo": " ",
  "codiceIstitutoPrincipale": "VTIC82500A",
  "denoIstitutoPrincipale": "I.C. CAPRANICA",
  "codiceMeccanografico": "VTMM82502C",
  "denominazionePlesso": "SCUOLA SEC. I VEIANO",
  "annoScolastico": 202021,
  "annoConseguimentoTitolo": 2021,
  "votoFinale": 6,
  "flagLode": "N"
}
]
"codiceIstitutoPrincipale": "BSIS024002",
"denoIstitutoPrincipale": "ASTOLFO LUNARDI",
"denominazionePlesso": "LUNARDI - BRESCIA",
"codiceMeccanografico": "BSTD024018",
"annoCorso": 5,
"percorsoStudi": "SCIENTIFICO",
"esitoFrequenza": 1
}
```

Valori tipoErrore

- **"Server Error"**: errori interni del server
- **"Authentication Error"**: errore di autenticazione (utente e/o password errati)
- **"Permission Error"**: operazione non permessa (esempio: Comune non censito nel Gateway o Codice ISTAT del Comune vuoto)
- **"Request Error"**:
 - Se i parametri della ricerca inseriti non sono corretti:
 - "Il parametro Codice Istat ha un formato non valido"
 - "Il parametro Codice Fiscale del Cittadino ha un formato non valido"
 - "Il parametro Id ANPR ha un formato non valido"
- **"ANIST Error"**: errori e anomalie restituite da ANIST a fronte della ricerca effettuata