



Indagor Service

• ST034 •

ACQUISIZIONE MASSIVA MANCATA FREQUENZA

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#).....	9
API Javascript.....	12

ST034

ACQUISIZIONE MASSIVA MANCATA FREQUENZA

DESCRIZIONE

Il servizio consente, dato un elenco di Codici Fiscali o ID ANPR, di verificare se i soggetti – nell'anno in corso - frequentano la scuola oppure NON la frequentano.

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

RespST034 ST034_Richiesta(string comunelstat, string idClient, string[] codiciFiscali)

- **ST034:** Nome dell'Indagor Service richiamato
- **comunelstat:** Codice Istat del Comune fruitore dell'applicativo che inoltra la chiamata
 - Ambiente di test: Codice Istat **040007**
- **idClient:** Identificativo univoco attribuito all'operazione dall'ente
 - Ambiente di test: Codice Fiscale (Esempio **test-client**)
- **codiciFiscali:** Elenco codici fiscali

Annullamento Richiesta

RespST034 ST034_AnnullaRichiesta(string comunelstat, string idRichiesta)

- **ST034:** Nome dell'Indagor Service richiamato
- **comunelstat:** Codice Istat del Comune fruitore dell'applicativo che inoltra la chiamata
 - Ambiente di test: Codice Istat **040007**
- **idRichiesta:** Id ottenuto da ANIST a fronte della richiesta per Codice Fiscale

Recupero Dati

RespST034 ST034_RecuperoDati(string comunelstat, string idRichiesta)

- **ST034:** Nome dell'Indagor Service richiamato
- **comunelstat:** Codice Istat del Comune fruitore dell'applicativo che inoltra la chiamata
 - Ambiente di test: Codice Istat **040007**
- **idRichiesta:** Id della Richiesta ottenuto da ANIST

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 ST034; 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
ID Richiesta	String
Codice Fiscale	String
Verifica Frequenza	"F" = Frequentante - "NF" = Non Frequentante

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){
        String idRichiesta;

        idRichiesta=TestST034_Richiesta();
        TestST034_AnnullamentoRichiesta(idRichiesta);

        idRichiesta = TestST034_Richiesta();
        TestST034_RecuperoDati(idRichiesta);
    }
```

RICHIESTA

```
Static String TestST034 Richiesta()
{
    Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
    RespST034 resp=null;
    Try
    {
        resp = cl.ST034_Richiesta("040007", "test-client", new String[] { "PRFTST80A11H501U",
            "PRFTST80A12H501Z", "PRFTST80A18H501N"});
    }
    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)
{
    String idRichiesta = resp.getIdRichiesta();
    return idRichiesta;
}
return null;
}
```

ANNULLAMENTO RICHIESTA

```
Static void String TestST034 AnnullamentoRichiesta ()
{
    Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
    RespST034 resp=null;
    Try
    {
        resp = cl.ST034_AnnullaRichiesta("040007", idRichiesta);
    }
    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);
}
```

RECUPERO DATI

```
static void TestST034_RecuperoDati(String idRichiesta)
{
    Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
    RespST034 resp=null;
    Try
    {
        resp = cl.ST034_RecuperoDati("040007", idRichiesta);
    }
    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)
{
    String idRichiesta2 = resp.getIdRichiesta();
    for(RispostaMassivaFrequenzaMancataSoggetto soggetto : resp.getRisposteMassive())
    {
        String msg=eanist.getMessage(); //errori, anomalie restituiti da ANIST
        System.out.println(msg);
    }
}
}
```

API .Net (C#)

Si aggiunge al progetto la libreria:

- **SteseiGateway.dll**

Si richiama nel seguente modo:

```
namespace Test
{
    class Program
    {
        static void Main(string[] args)
        {
            string idRichiesta;
            idRichiesta=TestST034_Richiesta();
            TestST034_AnnullamentoRichiesta(idRichiesta);
            idRichiesta = TestST034_Richiesta();
            TestST034_RecuperoDati(idRichiesta);
        }
    }
}
```

RICHIESTA

```
static string TestST034_Richiesta()
{
    Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
    RespST034 resp = null;
    try
    {
        resp = cl.ST034_Richiesta("040007", "test-client", new string[] { "PRFTST80A11H501U",
            "PRFTST80A12H501Z", "PRFTST80A18H501N"});
    }
    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)
{
    string idRichiesta = resp.idRichiesta;
    return idRichiesta;
}
return null;
}
```

ANNULLAMENTO RICHIESTA

```
static void TestST034_AnnullamentoRichiesta(string idRichiesta)
{
    Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
    RespST034 resp = null;
    try
    {
        resp = cl.ST034_AnnullaRichiesta("040007", idRichiesta);
    }
    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)
    {
    }
}
```

RECUPERO DATI

```
static void TestST034_RecuperoDati(string idRichiesta)
{
    Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
    RespST034 resp = null;
    try
    {
        resp = cl.ST034_RecuperoDati("040007", idRichiesta);
    }
    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)
    {
        string idRichiesta2 = resp.idRichiesta;
        foreach (RispostaMassivaFrequenzaMancataSoggetto soggetto in resp.risposteMassive)
        {
            string cf = soggetto.cf;
            string verificaFrequenza = soggetto.verificaFrequenza;
        }
    }
}
}
```

API Javascript

RICHIESTA

```
function callST034_Richiesta()
{
    let endpoint= "http://84.240.190.236/api/ST034";
    var criteri={comuneIstat: "040007", idClient:"test-client", codiciFiscali:["PRFTST80A11H501U", "PRFTST80A12H501Z",
"PRFTST80A18H501N"]};
    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)
        {
            idRichiesta=""; if (!risposta.error.error)
            {
                let idRichiesta = resp.idRichiesta; idRichiesta=risposta.response.idRichiesta;
            }
        },
        error: function (qXHR, textStatus, errorThrown)
        {
            alert("ERR");
        }
    });
}

let risposta =
{
    "error":
    {
        "error": false, "tipoErrore": "", "messaggio": ""
    },
    "response":
    {
        "idRichiesta": "20260609-6f76470f-078c-40d4-ab85-0124c342f229", "risposteMassive": null
    }
};
```

ANNULLAMENTO RICHIESTA

```
function callST034_AnnullamentoRichiesta()
{
    let endpoint= "http://84.240.190.236/api/ST034";
    var criteri={comunelstat: "040007", idRichiesta: "20260609-6f76470f-078c-40d4-ab85-0124c342f229", annullaRichiesta: true};
    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)
        {
            if (!risposta.erroro.erroro)
            {
            }
            Else
            {
            }
        },
        error: function (qXHR, textStatus, errorThrown)
        {
            alert("ERR");
        }
    });
}

let risposta =
{
    "errore":
    {
        "errore": false,
        "tipoErrore": "",
        "messaggio": ""
    },
    "response":
    {
        "idRichiesta": "20260609-6f76470f-078c-40d4-ab85-0124c342f229", "risposteMassive": null
    }
}
```

RECUPERO DATI

```
function callST034_RecuperoDati()
{
    let endpoint= "http://84.240.190.236/api/ST034";
    var criteri={comunelstat: "040007", idRichiesta: "20260609-6f76470f-078c-40d4-ab85-0124c342f229", recuperoDati: false};
    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)
        {
            if (!risposta.errore.errore)
            {
                let idRichiesta2 = resp.idRichiesta;
                for (soggetto of resp.risposteMassive)
                {
                    let cf = soggetto.cf;
                    let verificaFrequenza = soggetto.verificaFrequenza;
                }
            }
        },
        error: function (qXHR, textStatus, errorThrown)
        {
            alert("ERR");
        }
    });
}

let risposta =
{
    "errore":
    {
        "errore": false,
        "tipoErrore": "",
        "messaggio": ""
    },
    "response":
    {
        "idRichiesta": "20260609-6f76470f-078c-40d4-ab85-0124c342f229", "risposteMassive":
        [
            {"cf": "PRFTST80A11H501U", "verificaFrequenza": "F"},
            {"cf": "PRFTST80A12H501Z", "verificaFrequenza": "F"},
            {"cf": "PRFTST80A18H501N", "verificaFrequenza": "F"}
        ]
    }
}
```

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"
 - "Il Codice Meccanografico ha un formato non valido"
 - "Il Codice Fiscale Istituto ha un formato non valido"
- **"ANIST Error"**: errori e anomalie restituite da ANIST a fronte della ricerca effettuata