



**Indagor Service**  
**• ST029 •**

---

---

**ACQUISIZIONE MASSIVA  
DOMICILI DIGITALI  
DA INAD**

---

---

**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.....	8

## ST029

# Dati di Domicili Digitali da INAD

## DESCRIZIONE

Il servizio consente, dato un file contenente un elenco di Codici Fiscali, di acquisire in INAD - in asincrono - i seguenti gruppi di dati:

- Domicili Digitali

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"**

**Metodo:**

**RespST029 ST029\_Richiesta(String comunelstat, String[] codiciFiscali, String motivoRichiesta)**

- **ST029:** Nome dell'Indagor Service richiamato
- **comunelstat:** Codice Istat del Comune fruitore dell'applicativo che inoltra la chiamata
  - Ambiente di test: Codice Istat **040007**
- **codiciFiscali:** Elenco Codici Fiscali dei cittadini
  - Ambiente di test: Elenco Codici Fiscali (Esempio **"TRVVCN73H02L259I"**, **"FSTRCR56T18C227H"**, **"GLLL5587505G838N"**)
  - N.B.** In ambiente di produzione è consentito dalle specifiche INAD inserire massimo 1000 Codici Fiscali per richiesta
- **motivoRichiesta:** Testo libero (ad esempio "Numero Pratica" a cui si riferisce l'acquisizione dati)

**RespST029 ST029\_RecuperoDati(String comunelstat, String idRichiesta)**

- **ST029:** Nome dell'Indagor Service richiamato
- **comunelstat:** Codice Istat del Comune fruitore dell'applicativo che inoltra la chiamata
  - Ambiente di test: Codice Istat **040007**
- **idrichiesta:** identificativo della richiesta ottenuto dal metodo precedente
  - si tratta di un UUID/GUID valido 48 ore, trascorse le quali i risultati della ricerca sono cancellati. Es: **"6e10f078-e453-4890-87a2-69776a6ca5cd"**

## 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 Codici Fiscali ha un codice non valido 710FBA80B29D810R"
    - "L'ID Richiesta ha un formato non valido"
  - Se manca il criterio di ricerca
    - "Parametri di input non validi"
- **"INADException"**: errori e anomalie restituite da INAD a fronte della ricerca effettuata

## CONTENUTO DELLA RISPOSTA

Di seguito sono riportati i dati che compongono la risposta resa dal servizio ST029; se i valori della risposta sono pari a:

- null
- stringa vuota
- LocalDate.MIN

significa che tali dati non sono presenti in INAD

Descrizione		Note	
Id Richiesta			
<b>Soggetto 1</b>			
Soggetto 1	Codice Fiscale		
	Data Estrazione	Formato aaaa-mm-gg	
	<b>Domicilio Digitale 1</b>		
	Indirizzo Digitale 1	Indirizzo Digitale ID1	
		Professione ID1	
		Data Fine Validità ID1	Formato aaaa-mm-gg
		Motivo Cessazione ID1	
	<b>Domicilio Digitale 2</b>		
	Vedi Domicilio Digitale sopra		
	<b>Soggetto 2</b>		
Vedi Soggetto sopra			

## 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){
        TestST029_Richiesta();
        TestST029_RecuperoDati();
    }
    public static void TestST029_Richiesta() {
        Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");

        RespST029 resp=null;
        String[] codiciFiscali = { "TRVVCN73H02L259I", "FSTRCR56T18C227H", "GLLLSS87S05G838N" };
        try {
            resp = cl.ST029_Richiesta("040007", codiciFiscali, "Pratica N. 987654321");
        }
        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 (INADException einad) {  
    String msg=einad.getMessage(); //errori, anomalie restituiti da INAD  
    System.out.println(msg);  
}  
  
// risposta  
if (resp != null) {  
    String idRichiesta = resp.getIdRichiesta();  
    System.out.println(idRichiesta);  
}  
  
}  
  
public static void TestST029_RecuperoDati() {  
    Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");  
  
    RespST029 resp=null;  
  
    try {  
        resp = cl.ST029_RecuperoDati("040007", "6e10f078-e453-4890-87a2-69776a6ca5cd");  
    }  
    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 (INADException einad) {  
        String msg=einad.getMessage(); //errori, anomalie restituiti da INAD  
        System.out.println(msg);  
    }  
  
    // risposta  
    if (resp != null) {  
        String idRichiesta = resp.getIdRichiesta();  
        for(RespSoggettoINAD sogg : resp.getSoggetti()) {  
            String codiceFiscale = sogg.getCodiceFiscale();  
            String dataEstrazione = sogg.getDataEstrazione();  
            LocalDateTime dataEstrazioneData = sogg.getDataEstrazioneAsDate();  
            for (int i = 0; i < sogg.getIndirizziDigitali().length; i++)  
            {  
                String indirizzoDigitale = sogg.getIndirizziDigitali()[i].getIndirizzoDigitale();  
                String professione = sogg.getIndirizziDigitali()[i].getProfessione();  
                String dataFineValidita=sogg.getIndirizziDigitali()[i].getDataFineValidita();
```

```
LocalDateTime dataFineValiditaData=sogg.getIndirizziDigitali()[i].getDataFineValiditaAsDate();  
String motivazioneCessazione = sogg.getIndirizziDigitali()[i].getMotivazioneCessazione();
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

## 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) {
            TestST029_Richiesta();
            TestST029_RecuperoDati();
        }
        static void TestST029_Richiesta()
        {
            Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
            RespST029 resp = null;

            try
            {
                string[] codiciFiscali = { "TRVVCN73H02L259I", "FSTRCR56T18C227H", "GLLLSS87S05G838N" };
                resp = cl.ST029_Richiesta("040007", codiciFiscali, "Pratica N. 987654321");
            }
            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 (INADException einad)
            {
                string msg = einad.Message; //errori, anomalie restituiti da INAD
            }

            //risposta
            if (resp != null)
            {
                string idRichiesta = resp.idRichiesta;
            }
        }
    }
}
```

```
static void TestST029_RecuperoDati()
{
    Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
    RespST029 resp = null;
    try
    {
        resp = cl.ST029_RecuperoDati("040007", "6e10f078-e453-4890-87a2-69776a6ca5cd");
    }
    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 (INADException einad)
    {
        string msg = einad.Message; //errori, anomalie restituiti da INAD
    }

    //risposta
    if (resp != null)
    {
        string idRichiesta = resp.idRichiesta;

        foreach (RespSoggettoINAD sogg in resp.soggetti)
        {
            string codiceFiscale = sogg.codiceFiscale;
            string dataEstrazione = sogg.dataEstrazione;
            DateTime dataEstrazioneData = sogg.dataEstrazioneAsDate();
            for (int i = 0; i < sogg.indirizziDigitali.Length; i++)
            {
                string indirizzoDigitale = sogg.indirizziDigitali[i].indirizzoDigitale;
                string professione = sogg.indirizziDigitali[i].professione;
                string dataFineValidita = sogg.indirizziDigitali[i].dataFineValidita;
                DateTime dataFineValiditaData = sogg.indirizziDigitali[i].dataFineValiditaAsDate();
                string motivazioneCessazione = sogg.indirizziDigitali[i].motivazioneCessazione;
            }
        }
    }
}
```

## API Javascript

```
let endpoint = "http://84.240.190.236/api/ST029";
var criteri = { comunelstat: "040007", codiciFiscali: ["TRVVCN73H02L259I", "FSTRCR56T18C227H", "GLLLSS87S05G838N"] };
var account = { utente: "040007", password: "INIZIALE" };
var restrizioni = { motivoRichiesta: "test pratica" };
var richiesta = { credenzialiAccesso: account, restrizioniAccesso: restrizioni, 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": {
    "idRichiesta": "6e10f078-e453-4890-87a2-69776a6ca5cd",
    "soggetti": null
  }
}
```

```
let endpoint = "http://84.240.190.236/api/ST029";
var criteri = { comunelstat: "040007", idRichiesta: "6e10f078-e453-4890-87a2-69776a6ca5cd" };
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": {
    "idRichiesta": "6e10f078-e453-4890-87a2-69776a6ca5cd",
    "soggetti": [{
      "codiceFiscale": "TRVVCN73H02L259",
      "dataEstrazione": "2026-05-05T11:55:00Z",
      "indirizziDigitali": [{
        "indirizzoDigitale": "test@pec.it",
        "professione": "NON AVVOCATO",
        "motivazioneCessazione": "CESSAZIONE_UFFICIO",
        "dataFineValidita": "2024-09-01T00:00:00Z"
      }],
      {
        "indirizzoDigitale": "test@pec.it",
        "professione": "",
        "motivazioneCessazione": "CESSAZIONE_UFFICIO",
        "dataFineValidita": "2024-09-01T00:00:00Z"
      },
      {
        "indirizzoDigitale": "test@pec.it",
        "professione": "MAI AVVOCATO",
        "motivazioneCessazione": "CESSAZIONE_UFFICIO",
        "dataFineValidita": "2024-09-01T00:00:00Z"
      }
    ]
  },
  {
    "codiceFiscale": "FSTR56T18C227H",
    "dataEstrazione": "2026-05-05T11:55:00Z",
    "indirizziDigitali": [{
      "indirizzoDigitale": "test@pec.it",
      "professione": "",
      "motivazioneCessazione": "CESSAZIONE_UFFICIO",
      "dataFineValidita": "2025-06-08T00:00:00Z"
    }
  ],
  {
    "codiceFiscale": "GLLLSS87S05G838N",
    "dataEstrazione": "2026-05-05T11:55:00Z",
    "indirizziDigitali": []
  }
}
}
```

## 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 ha un formato non valido"
  - Se manca il criterio di ricerca
    - "Criteri di ricerca non validi"
- **"INAD Error"**: errori e anomalie restituite da INAD a fronte della ricerca effettuata