



Indagor Service

- ST008 •

ACQUISIZIONE PUNTUALE DATI DI MATRIMONIO

Specifiche di Integrazione

INDICE

Descrizione.....	1
Contenuto della Richiesta.....	1
Contenuto delle Eccezioni.....	4
Contenuto della Risposta.....	4
Esempi di Integrazione.....	8
API Java.....	8
API .Net (C#).....	13
API Javascript.....	17
Tabella Uffici Comunali.....	21

ST008

Dati di MATRIMONIO

DESCRIZIONE

Il servizio consente, dato un ID ANPR o un Codice Fiscale o le Generalità di un soggetto, di acquisire in ANPR - in sincrono - i seguenti gruppi di dati:

- ID ANPR
- Generalità
- Data di Decesso (valorizzato solo nel caso in cui il Soggetto risulti essere Deceduto)
- Stato Civile

Nel caso in cui il soggetto abbia contratto Matrimonio i seguenti gruppi di dati:

- IDANPR Coniuge
- Generalità Coniuge
- Cittadinanza Coniuge
- Dati Matrimonio
- Dati Cessazione/Annullamento Matrimonio

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*

RespST008 ST008(String idOperatore, int idUfficio, String comunelstat, String codiceFiscale, String motivoRichiesta)

- **ST008:** Nome dell'Indagor Service richiamato
- **idOperatore:** Codice ID assegnato dall'applicativo all'operatore del Comune che inoltra la richiesta
- **idUfficio:** Codice Ufficio di appartenenza dell'operatore che inoltra la Richiesta; valore compreso tra 1 e 21 (vedi tabella "Uffici Comunali" più oltre)
- **comuneIstat:** 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 **SGGNNA90R50F839T**)
- **motivoRichiesta:** Testo libero (ad esempio "Numero Pratica" a cui si riferisce l'acquisizione dati)

Richiesta per IDANPR

RespST008 ST008_IdANPR(String idOperatore, int idUfficio, String comunelstat, String idANPR, String motivoRichiesta)

- **ST008_IdANPR:** Nome dell'Indagor Service richiamato
- **idOperatore:** Codice ID assegnato dall'applicativo all'operatore del Comune che inoltra la richiesta
- **idUfficio:** Codice Ufficio di appartenenza dell'operatore che inoltra la Richiesta; valore compreso tra 1 e 21 (vedi tabella "Uffici Comunali" più oltre)
- **comuneIstat:** Codice Istat del Comune fruitore dell'applicativo che inoltra la chiamata
 - Ambiente di test: Codice Istat **040007**
- **IDANPR:** ID ANPR del cittadino
 - Ambiente di test: IDANPR (Esempio **EV62727VC**)
- **motivoRichiesta:** Testo libero (ad esempio "Numero Pratica" a cui si riferisce l'acquisizione dati)

Richiesta per Generalità

RespST008 ST008(String idOperatore, int idUfficio, String comunelstat, Generalita generalita, String motivoRichiesta)

- **ST008:** Nome dell'Indagor Service richiamato
- **idOperatore:** Codice ID assegnato dall'applicativo all'operatore del Comune che inoltra la richiesta
- **idUfficio:** Codice Ufficio di appartenenza dell'operatore che inoltra la Richiesta; valore compreso tra 1 e 21 (vedi tabella "Uffici Comunali" più oltre)
- **comunelstat:** Codice Istat del Comune fruitore dell'applicativo che inoltra la chiamata
 - Ambiente di test: Codice Istat **040007**
- Generalità (esempio cittadino nato in Italia)
 - **cognome:** Cognome del cittadino
 - Ambiente di test: Cognome **SOGGETTA**
 - **nome:** Nome del cittadino
 - Ambiente di test: Nome **ANNA**
 - **Sesso:** Sesso del cittadino
 - Ambiente di test: Sesso **F**
 - **dataNascita:** Data di Nascita del cittadino
 - Ambiente di test: Data di Nascita **1990-10-10**
 - **comuneNascita:** Comune di Nascita del cittadino
 - Ambiente di test: Comune di Nascita **NAPOLI**
 - **siglaProvinciaNascita:** Sigla della Provincia del Comune di Nascita del cittadino
 - Ambiente di test: Sigla della Provincia del Comune di Nascita **NA**
- Generalità (esempio cittadino nato all'estero)
 - **cognome:** Cognome del cittadino
 - Ambiente di test: Cognome **DELGADO**
 - **nome:** Nome del cittadino
 - Ambiente di test: Nome **BLANCA**
 - **Sesso:** Sesso del cittadino
 - Ambiente di test: Sesso **F**
 - **dataNascita:** Data di Nascita del cittadino
 - Ambiente di test: Data di Nascita **1970-05-20**
 - **localitaNascita:** Località Estera di Nascita del cittadino
 - Ambiente di test: Località Estera di Nascita **CITTÀ DEL GUATEMALA**
 - **statoEsteroNascita:** Stato della Località Estera di Nascita del cittadino
 - Ambiente di test: Stato della Località Estera di Nascita **GUATEMALA**
- **motivoRichiesta:** Testo libero (ad esempio "Numero Pratica" a cui si riferisce l'acquisizione dati)

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

CONTENUTO DELLA RISPOSTA

Di seguito sono riportati i dati che compongono la risposta resa dal servizio ST008; 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 ANPR 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 Operazione ANPR			
ID Operazione ANPR			Numero assegnato da ANPR alla Richiesta
ID ANPR			
ID ANPR			Codice Identificativo Univoco del Soggetto assegnato da ANPR
Generalità			
Codice Fiscale	Codice Fiscale		
	Flag Validità Codice Fiscale		
	Descrizione Validità Codice Fiscale		Descrizione del Flag di Validità del Codice Fiscale
	Validità Codice Fiscale		Data Validità Codice Fiscale - Formato aaaa-mm-gg
Cognome			
Flag Senza Cognome			In caso di Valore "1" il Soggetto è anagraficamente privo di Cognome
Nome			
Flag Senza Nome			In caso di Valore "1" il Soggetto è anagraficamente privo di Nome
Sesso			
Data Nascita	Data di Nascita		Formato aaaa-mm-gg
	Data di Nascita Senza Giorno		In caso di Valore "1" il Soggetto è anagraficamente privo del Giorno di Nascita (valore 00)
	Data di Nascita Senza Giorno e Mese		In caso di Valore "1" il Soggetto è anagraficamente privo del Giorno e del Mese di Nascita (valori 00)
Luogo Nascita	Luogo Eccezionale di Nascita		
	Comune	Comune Italiano di Nascita	
		Codice ISTAT Comune Italiano di Nascita	
		Sigla Provincia Comune Italiano di Nascita	
		Località Comune Italiano di Nascita	
	Località Estera	Località Estera di Nascita	
		Stato Estero di Nascita	
		Codice ISTAT Stato Estero di Nascita	
		Provincia/Contea Estera di Nascita	
Flag Soggetto AIRE			Valori S=Si o N=No; in caso di Valore null si intende Soggetto NON AIRE
Anno di Espatrio			Formato aaaa
Scheda Soggetto Comune			
	Codice ISTAT Comune		Codice ISTAT Comune assegnatario dell'ID Scheda Soggetto Comune
	ID Scheda Soggetto Comune		Codice Identificativo Univoco della Scheda Individuale del Soggetto assegnato dal Comune nel proprio archivio locale
ID Scheda Soggetto ANPR			Numero assegnato da ANPR alla Scheda Individuale del Soggetto
Data Decesso			
Data di Decesso			Formato aaaa-mm-gg – Presente solo se il Soggetto è Deceduto

Stato Civile		
Stato Civile		
Stato Civile Descrizione		
Note sullo Stato Civile		
Stato Civile NON Disponibile o Sconosciuto		
ID ANPR (Coniuge)		
Vedi ID ANPR sopra		
Generalità (Coniuge)		
Vedi Generalità sopra		
Cittadinanza (Coniuge)		
Stato Estero della Cittadinanza coniuge		
Codice ISTAT Stato Estero della Cittadinanza coniuge		
Note sulla Cittadinanza coniuge		
Data Validità della Cittadinanza coniuge		Formato aaaa-mm-gg
Dati Matrimonio		
Ordine Matrimonio		Numero Ordine di Matrimonio
Data	Data del Matrimonio	Formato aaaa-mm-gg
	Flag Senza Giorno del Matrimonio	In caso di Valore "1" il Matrimonio del Soggetto è privo del Giorno (valore 00)
	Flag Senza Giorno e Mese del Matrimonio	In caso di Valore "1" il Matrimonio del Soggetto è privo del Giorno e del Mese (valori 00)
Luogo	Luogo Eccezionale del Matrimonio	
	Comune	Comune Italiano del Matrimonio
		Codice ISTAT Comune Italiano del Matrimonio
		Sigla Provincia Comune Italiano del Matrimonio
		Località Comune Italiano del Matrimonio
	Dati Località Estera	
		Località Estera del Matrimonio
		Stato Estero del Matrimonio
		Codice ISTAT Stato Estero del Matrimonio
		Provincia/Contea Estera del Matrimonio
	Atto	Comune
Codice ISTAT Comune Italiano Atto di Matrimonio		
Sigla Provincia Comune Italiano Atto di Matrimonio		
Località Comune Italiano Atto di Matrimonio		
Ufficio Atto di Matrimonio		
Anno Atto di Matrimonio		Formato aaaa
Parte Atto di Matrimonio		
Serie Atto di Matrimonio		
Numero Atto di Matrimonio		
Volume Atto di Matrimonio		
Data Atto di Matrimonio		Formato aaaa-mm-gg
Trascrizione Atto di Matrimonio		

Dati Annullamento/Cessazione Matrimonio

Tipo Cessazione/Annullamento Matrimonio			
Cessazione/Annullamento Matrimonio Descrizione			
Sentenza	Data Sentenza Cessazione/Annullamento Matrimonio		Formato aaaa-mm-gg
	Numero Sentenza Cessazione/Annullamento Matrimonio		
	Tipo Tribunale Sentenza Cessazione/Annullamento Matrimonio		
	Tribunale Sentenza Cessazione/Annullamento Matrimonio Descrizione		Formato aaaa-mm-gg
	Autorità Sentenza Cessazione/Annullamento Matrimonio		
	Data Validità Sentenza Cessazione/Annullamento Matrimonio		
Atto	Comune	Comune Italiano Atto di Cessazione/Annullamento Matrimonio	
		Codice ISTAT Comune Italiano Atto di Cessazione/Annullamento Matrimonio	
		Sigla Provincia Comune Italiano Atto di Cessazione/Annullamento Matrimonio	
		Località Comune Italiano Atto di Cessazione/Annullamento Matrimonio	
	Ufficio Atto di Cessazione/Annullamento Matrimonio		
	Anno Atto di Cessazione/Annullamento Matrimonio		Formato aaaa
	Parte Atto di Cessazione/Annullamento Matrimonio		
	Serie Atto di Cessazione/Annullamento Matrimonio		
	Numero Atto di Cessazione/Annullamento Matrimonio		
	Volume Atto di Cessazione/Annullamento Matrimonio		
	Data Atto di Cessazione/Annullamento Matrimonio		Formato aaaa-mm-gg
	Trascrizione Atto di Cessazione/Annullamento Matrimonio		

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){
        TestST008();
    }
    public static void TestST008(){
        Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
        RespST008 resp=null;
        try{
            String idOperatore = "user00001";
            int idUfficio = 20;

            //Ricerca per Codice Fiscale
            resp = cl.ST008(idOperatore, idUfficio, "040007", "SGGNNA90R50F839T", "Pratica N. 987654321");

            //Ricerca per IDANPR
            resp = cl.ST008_IdANPR(idOperatore, idUfficio, "040007", "EV62727VC", "Pratica N. 987654321");

            //Ricerca per Generalità
            Generalita generalita = new Generalita();
```

```
//Esempio Cittadino nato in Italia
generalita.nominativo.cognome = "SOGGETTA";
generalita.nominativo.nome = "ANNA";
generalita.sesso = "F";
generalita.dataNascita.dataEvento = "1990-10-10";
generalita.luogoNascita.comune.nomeComune = "NAPOLI";
generalita.luogoNascita.comune.siglaProvincia = "NA";

//Esempio Cittadino nato all'Estero
generalita.nominativo.cognome = "DELGADO";
generalita.nominativo.nome = "BLANCA";
generalita.sesso = "F";
generalita.dataNascita.dataEvento = "1970-05-20";
generalita.luogoNascita.localita.descrizioneStato = "GUATEMALA";
generalita.luogoNascita.localita.descrizioneLocalita = "CITTÀ DEL GUATEMALA";

resp = cl.ST008(idOperatore, idUfficio, "040007", generalita, "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 (ANPRException eanpr){
    String msg=eanpr.getMessage(); //errori, anomalie restituiti da ANPR
    System.out.println(msg);
}

// risposta
if (resp != null){
    String idOperazioneANPR=resp.getIdOperazioneANPR();
    String idANPR=resp.getIdANPR();
    String codiceFiscale=resp.getCodiceFiscale();
    String flagCodFiscVal=resp.getCodiceFiscaleValiditaFlag();
    String flagCodFiscDescr=resp.getCodiceFiscaleValiditaFlagDescrizione();
    String codiceFiscaleValidita=resp.getCodiceFiscaleValidita();
    LocalDate codiceFiscaleValiditaData=resp.getCodiceFiscaleValiditaAsDate();
    String cognome=resp.getCognome();
    String flagSenzaCognome=resp.getSenzaCognomeFlag();
    String nome=resp.getNome();
    String flagSenzaNome=resp.getSenzaNomeFlag();
    String sesso=resp.getSesso();
    String dataNascita=resp.getNascitaData();
    LocalDate dataNascitaData=resp.getNascitaDataAsDate();
    String flagSenzaGiorno=resp.getNascitaDataSenzaGiorno();
    String flagSenzaGiornoMese=resp.getNascitaDataSenzaGiornoMese();
    String nascitaLuogoEccezionale=resp.getNascitaLuogoEccezionale();
    String comuneNascita=resp.getNascitaComuneItaliano();
    String nascitaIT_Istat=resp.getNascitaComuneItalianoCodiceIstat();
    String nascitaProvincialeItaliana=resp.getNascitaProvincialeItaliana();
    String localita=resp.getNascitaComuneItalianoLocalita();
    String comuneNascitaEstero=resp.getNascitaComuneEstero();
    String statoNascita=resp.getNascitaStatoEstero();
    String nascitaEsteroIstat=resp.getNascitaStatoEsteroCodiceIstat();
    String nascitaProvinciaEstera=resp.getNascitaProvinciaEstera();
    String soggettoAIRE=resp.getSoggettoAIRE();
    String annoEspatrio=resp.getAnnoEspatrio();
    String idSoggettoComuneIstat=resp.getIdSchedaSoggettoComuneIstat();
    String idSoggettoComune=resp.getIdSchedaSoggettoComune();
    String idSchedaSoggetto=resp.getIdSchedaSoggetto();
    String dataDecesso=resp.getDataDecesso();
    LocalDate dataDecessoData=resp.getDataDecessoAsDate();
    String statoCivile=resp.getStatoCivile();
    String statoCivileDescr=resp.getStatoCivileDescr();
    String noteStatoCivile=resp.getNoteStatoCivile();
    String statoCivileNonDisponibile=resp.getStatoCivileNonDisponibile();
    String dataDecesso=resp.getDataDecesso();
}
```

```
LocalDate dataDecessoData=resp.getDataDecessoAsDate();
String idANPRConiuge=resp.getIdANPRConiuge();
String codicefiscaleConiuge=resp.getCodiceFiscaleConiuge();
String flagCodFiscValConiuge=resp.getCodiceFiscaleValiditaFlagConiuge();
String flagCodFiscDescrConiuge=resp.getCodiceFiscaleValiditaFlagDescrizioneConiuge();
String codiceFiscaleValiditaConiuge=resp.getCodiceFiscaleValiditaConiuge();
LocalDate codiceFiscaleValiditaConiugeData=resp.getCodiceFiscaleValiditaConiugeAsDate();
String cognomeConiuge=resp.getCognomeConiuge();
String flagSenzaCognomeConiuge=resp.getSenzaCognomeFlagConiuge();
String nomeConiuge=resp.getNomeConiuge();
String flagSenzaNomeConiuge=resp.getSenzaNomeFlagConiuge();
String sessoConiuge=resp.getSessoConiuge();
String dataNascitaConiuge=resp.getNascitaDataConiuge();
LocalDate dataNascitaConiugeData=resp.getNascitaDataConiugeAsDate();
String flagSenzaGiornoConiuge=resp.getNascitaDataSenzaGiornoConiuge();
String flagSenzaGiornoMeseConiuge=resp.getNascitaDataSenzaGiornoMeseConiuge();
String nascitaLuogoEccezionaleConiuge=resp.getNascitaLuogoEccezionaleConiuge();
String comuneNascitaConiuge=resp.getNascitaComuneItalianoConiuge();
String nascitaIT_IstatConiuge=resp.getNascitaComuneItalianoCodiceIstatConiuge();
String nascitaProvincialeItalianaConiuge=resp.getNascitaProvincialeItalianaConiuge();
String localitaConiuge=resp.getNascitaComuneItalianoLocalitaConiuge();
String comuneNascitaEsteroConiuge=resp.getNascitaComuneEsteroConiuge();
String statoNascitaConiuge=resp.getNascitaStatoEsteroConiuge();
String nascitaEsterolstatConiuge=resp.getNascitaStatoEsteroCodiceIstatConiuge();
String nascitaProvinciaEsteraConiuge=resp.getNascitaProvinciaEsteraConiuge();
String soggettoAIREConiuge=resp.getSoggettoAIREConiuge();
String annoEspatrioConiuge=resp.getAnnoEspatrioConiuge();
String idSoggettoComuneIstatConiuge=resp.getIdSchedaSoggettoComuneIstatConiuge();
String idSoggettoComuneConiuge=resp.getIdSchedaSoggettoComuneConiuge();
String idSchedaSoggettoConiuge=resp.getIdSchedaSoggettoConiuge();
String statoCittadinanzaConiuge=resp.getCittadinanzaStatoConiuge();
String istatCittadinanzaConiuge=resp.getCittadinanzaIstatConiuge();
String noteCittadinanzaConiuge=resp.getCittadinanzaNoteConiuge();
String dataValiditaCittadinanzaConiuge=resp.getCittadinanzaDataValiditaConiuge();
LocalDate dataValiditaCittadinanzaConiugeData=resp.getCittadinanzaDataValiditaConiugeAsDate();
String ordineMatrimonio=resp.getOrdineMatrimonio();
String dataMatrimonio=resp.getMatrimonioData();
LocalDate dataMatrimonioData=resp.getMatrimonioDataAsDate();
String flagSenzaGiornoMatrimonio=resp.getMatrimonioDataSenzaGiorno();
String flagSenzaGiornoMeseMatrimonio=resp.getMatrimonioDataSenzaGiornoMese();
String matrimonioLuogoEccezionale=resp.getMatrimonioLuogoEccezionale();
String comuneMatrimonio=resp.getMatrimonioComuneItaliano();
String matrimonioIT_Istat=resp.getMatrimonioComuneItalianoCodiceIstat();
String matrimonioProvincialeItaliana=resp.getMatrimonioProvincialeItaliana();
String localitaMatrimonio=resp.getMatrimonioComuneItalianoLocalita();
String comuneMatrimonioEstero=resp.getMatrimonioComuneEstero();
String statoMatrimonio=resp.getMatrimonioStatoEstero();
String matrimonioEsterolstat=resp.getMatrimonioStatoEsteroCodiceIstat();
String matrimonioProvinciaEstera=resp.getMatrimonioProvinciaEstera();
String comuneAttoMatrimonio=resp.getAttoMatrimonioComuneItaliano();
String attoMatrimonioIT_Istat=resp.getAttoMatrimonioComuneItalianoCodiceIstat();
String attoMatrimonioProvincialeItaliana=resp.getAttoMatrimonioProvincialeItaliana();
```

```
String localitaAttoMatrimonio=resp.getAttoMatrimonioComuneItalianoLocalita();
String ufficioAttoMatrimonio=resp.getAttoMatrimonioUfficio();
String annoAttoMatrimonio=resp.getAttoMatrimonioAnno();
String parteAttoMatrimonio=resp.getAttoMatrimonioParte();
String serieAttoMatrimonio=resp.getAttoMatrimonioSerie();
String numeroAttoMatrimonio=resp.getAttoMatrimonioNumero();
String volumeAttoMatrimonio=resp.getAttoMatrimonioVolume();
String dataAttoMatrimonio=resp.getAttoMatrimonioData();
LocalDate dataAttoMatrimonioData=resp.getAttoMatrimonioDataAsDate();
String trascrittoAttoMatrimonio=resp.getAttoMatrimonioTrascritto();
String tipoCessazioneAnnullamentoMatrimonio=resp.getTipoCessazioneAnnullamentoMatrimonio();
String CessazioneAnnullamentoMatrimonioDescr=resp.getCessazioneAnnullamentoMatrimonioDescr();
String dataSentenzaCessazioneAnnullamentoMatrimonio=
    resp.getDataSentenzaCessazioneAnnullamentoMatrimonio();
LocalDate dataSentenzaCessazioneAnnullamentoMatrimonioData=
    resp.getDataSentenzaCessazioneAnnullamentoMatrimonioAsDate();
String numeroSentenzaCessazioneAnnullamentoMatrimonio=
    resp.getNumeroSentenzaCessazioneAnnullamentoMatrimonio();
String tipoTribunaleSentenzaCessazioneAnnullamentoMatrimonio=
    resp.getTipoTribunaleSentenzaCessazioneAnnullamentoMatrimonio();
String TribunaleSentenzaCessazioneAnnullamentoMatrimonioDescr=
    resp.getTribunaleSentenzaCessazioneAnnullamentoMatrimonioDescr();
String autoritaSentenzaCessazioneAnnullamentoMatrimonio=
    resp.getAutoritaSentenzaCessazioneAnnullamentoMatrimonio();
String dataValiditaSentenzaCessazioneAnnullamentoMatrimonio=
    resp.getDataValiditaSentenzaCessazioneAnnullamentoMatrimonio();
LocalDate dataValiditaSentenzaCessazioneAnnullamentoMatrimonioData=
    resp.getDataValiditaSentenzaCessazioneAnnullamentoMatrimonioAsDate();
String comuneAttoCessazioneAnnullamentoMatrimonio=
    resp.getAttoCessazioneAnnullamentoMatrimonioComuneItaliano();
String attoCessazioneAnnullamentoMatrimonioT_Istat=
    resp.getAttoCessazioneAnnullamentoMatrimonioComuneItalianoCodiceIstat();
String attoCessazioneAnnullamentoMatrimonioProvincialeItaliana=
    resp.getAttoCessazioneAnnullamentoMatrimonioProvincialeItaliana();
String localitaAttoCessazioneAnnullamentoMatrimonio=
    resp.getAttoCessazioneAnnullamentoMatrimonioComuneItalianoLocalita();
String ufficioAttoCessazioneAnnullamentoMatrimonio=resp.getAttoCessazioneAnnullamentoMatrimonioUfficio();
String annoAttoCessazioneAnnullamentoMatrimonio=resp.getAttoCessazioneAnnullamentoMatrimonioAnno();
String parteAttoCessazioneAnnullamentoMatrimonio=resp.getAttoCessazioneAnnullamentoMatrimonioParte();
String serieAttoCessazioneAnnullamentoMatrimonio=resp.getAttoCessazioneAnnullamentoMatrimonioSerie();
String numeroAttoCessazioneAnnullamentoMatrimonio=
    resp.getAttoCessazioneAnnullamentoMatrimonioNumero();
String volumeAttoCessazioneAnnullamentoMatrimonio=
    resp.getAttoCessazioneAnnullamentoMatrimonioVolume();
String dataAttoCessazioneAnnullamentoMatrimonio=resp.getAttoCessazioneAnnullamentoMatrimonioData();
LocalDate dataAttoCessazioneAnnullamentoMatrimonioData=
    resp.getAttoCessazioneAnnullamentoMatrimonioDataAsDate();
String trascrittoAttoCessazioneAnnullamentoMatrimonio=
    resp.getAttoCessazioneAnnullamentoMatrimonioTrascritto();
    }
}
```

API .Net (C#)

Si aggiunge al progetto la libreria:

- **SteseiGateway.dll**

Si richiama nel seguente modo:

```
{
    class Program{
        static void Main(string[] args){
            TestST008();
        }
        static void TestST008(){
            Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
            RespST008 resp = null;
            try{
                string idOperatore = "user00001";
                int idUfficio = 20;

                //Ricerca per Codice Fiscale
                resp = cl.ST008(idOperatore, idUfficio, "040007", "SGGNNA90R50F839T", "Pratica N. 987654321");

                //Ricerca per IDANPR
                resp = cl.ST008_IdANPR(idOperatore, idUfficio, "040007", "EV62727VC", "Pratica N. 987654321");

                //Ricerca per Generalità
                Generalita generalita = new Generalita();

                //Esempio Cittadino nato in Italia
                generalita.nominativo.cognome = "SOGGETTA";
                generalita.nominativo.nome = "ANNA";
                generalita.sesso = "F";
                generalita.dataNascita.dataEvento = "1990-10-10";
                generalita.luogoNascita.comune.nomeComune = "NAPOLI";
                generalita.luogoNascita.comune.siglaProvincia = "NA";

                //Esempio Cittadino nato all'Estero
                generalita.nominativo.cognome = "DELGADO";
                generalita.nominativo.nome = "BLANCA";
                generalita.sesso = "F";
                generalita.dataNascita.dataEvento = "1970-05-20";
                generalita.luogoNascita.localita.descrizioneStato = "GUATEMALA";
                generalita.luogoNascita.localita.descrizioneLocalita = "CITTÀ DEL GUATEMALA";
            }
        }
    }
}
```

```
resp = cl.ST008(idOperatore, idUfficio, "040007", generalita, "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 (ANPRException eanpr){
    string msg = eanpr.Message; //errori, anomalie restituiti da ANPR
}

//risposta
if (resp != null){
    string idOperazioneANPR = resp.idOperazioneANPR;
    string idANPR = resp.idANPR;
    string codiceFiscale = resp.codiceFiscale;
    string flagCodFiscVal = resp.codiceFiscaleValiditaFlag;
    string flagCodFiscDescr = resp.codiceFiscaleValiditaFlagDescrizione;
    string codiceFiscaleValidita = resp.codiceFiscaleValidita;
    DateTime codiceFiscaleValiditaData = resp.codiceFiscaleValiditaAsDate();
}
```

```
string cognome = resp.cognome;  
string flagSenzaCognome = resp.senzaCognomeFlag;  
string nome = resp.nome;  
string flagSenzaNome = resp.senzaNomeFlag;  
string sesso = resp.sesso;  
string dataNascita = resp.nascitaData;  
DateTime dataNascitaData = resp.nascitaDataAsDate();  
string flagSenzaGiorno = resp.nascitaDataSenzaGiorno;  
string flagSenzaGiornoMese = resp.nascitaDataSenzaGiornoMese;  
string nascitaLuogoEccezionale = resp.nascitaLuogoEccezionale;  
string comuneNascita = resp.nascitaComuneItaliano;  
string nascitaIT_Istat = resp.nascitaComuneItalianoCodiceIstat;  
string nascitaProvincialeItaliana = resp.nascitaProvincialeItaliana;  
string localita = resp.nascitaComuneItalianoLocalita;  
string comuneNascitaEstero = resp.nascitaComuneEstero;  
string statoNascita = resp.nascitaStatoEstero;  
string nascitaEsteroIstat = resp.nascitaStatoEsteroCodiceIstat;  
string nascitaProvinciaEstera = resp.nascitaProvinciaEstera;  
string soggettoAIRE = resp.soggettoAIRE;  
string annoEspatrio = resp.annoEspatrio;  
string idSoggettoComuneIstat = resp.idSchedaSoggettoComuneIstat;  
string idSoggettoComune = resp.idSchedaSoggettoComune;  
string idSchedaSoggetto = resp.idSchedaSoggetto;  
string dataDecesso = resp.dataDecesso;  
DateTime dataDecessoData = resp.dataDecessoAsDate();  
string statoCittadinanzaConiuge = resp.cittadinanzaStatoConiuge;  
string istatCittadinanzaConiuge = resp.cittadinanzaIstatConiuge;  
string noteCittadinanzaConiuge = resp.cittadinanzaNoteConiuge;  
string dataValiditaCittadinanzaConiuge = resp.cittadinanzaDataValiditaConiuge;  
DateTime dataValiditaCittadinanzaConiugeData = resp.cittadinanzaDataValiditaConiugeAsDate();  
string ordineMatrimonio = resp.ordineMatrimonio;  
string dataMatrimonio = resp.matrimonioData;  
DateTime dataMatrimonioData = resp.matrimonioDataAsDate();  
string flagSenzaGiornoMatrimonio = resp.matrimonioDataSenzaGiorno;  
string flagSenzaGiornoMeseMatrimonio = resp.matrimonioDataSenzaGiornoMese;  
string matrimonioLuogoEccezionale = resp.matrimonioLuogoEccezionale;  
string comuneAttoMatrimonio = resp.attoMatrimonioComuneItaliano;  
string attoMatrimonioIT_Istat = resp.attoMatrimonioComuneItalianoCodiceIstat;  
string attoMatrimonioProvincialeItaliana = resp.attoMatrimonioProvincialeItaliana;  
string localitaAttoMatrimonio = resp.attoMatrimonioComuneItalianoLocalita;  
string ufficioAttoMatrimonio = resp.attoMatrimonioUfficio;  
string annoAttoMatrimonio = resp.attoMatrimonioAnno;  
string parteAttoMatrimonio = resp.attoMatrimonioParte;  
string serieAttoMatrimonio = resp.attoMatrimonioSerie;  
string numeroAttoMatrimonio = resp.attoMatrimonioNumero;  
string volumeAttoMatrimonio = resp.attoMatrimonioVolume;
```

```
string dataAttoMatrimonio = resp.attoMatrimonioData;  
DateTime dataAttoMatrimonioData = resp.attoMatrimonioDataAsDate();  
string trascittoAttoMatrimonio = resp.attoMatrimonioTrascritto;  
string tipoCessazioneAnnullamentoMatrimonio = resp.tipoCessazioneAnnullamentoMatrimonio;  
string CessazioneAnnullamentoMatrimonioDescr = resp.cessazioneAnnullamentoMatrimonioDescr;  
string dataSentenzaCessazioneAnnullamentoMatrimonio =  
    resp.dataSentenzaCessazioneAnnullamentoMatrimonio;  
DateTime dataSentenzaCessazioneAnnullamentoMatrimonioData =  
    resp.dataSentenzaCessazioneAnnullamentoMatrimonioAsDate();  
string numeroSentenzaCessazioneAnnullamentoMatrimonio =  
    resp.numeroSentenzaCessazioneAnnullamentoMatrimonio;  
string tipoTribunaleSentenzaCessazioneAnnullamentoMatrimonio =  
    resp.tipoTribunaleSentenzaCessazioneAnnullamentoMatrimonio;  
string TribunaleSentenzaCessazioneAnnullamentoMatrimonioDescr =  
    resp.tribunaleSentenzaCessazioneAnnullamentoMatrimonioDescr;  
string autoritaSentenzaCessazioneAnnullamentoMatrimonio =  
    resp.autoritaSentenzaCessazioneAnnullamentoMatrimonio;  
string dataValiditaSentenzaCessazioneAnnullamentoMatrimonio =  
    resp.dataValiditaSentenzaCessazioneAnnullamentoMatrimonio;  
DateTime dataValiditaSentenzaCessazioneAnnullamentoMatrimonioData =  
    resp.dataValiditaSentenzaCessazioneAnnullamentoMatrimonioAsDate();  
string comuneAttoCessazioneAnnullamentoMatrimonio =  
    resp.attoCessazioneAnnullamentoMatrimonioComunelItaliano;  
string attoCessazioneAnnullamentoMatrimonioIT_Istat =  
    resp.attoCessazioneAnnullamentoMatrimonioComunelItalianoCodiceIstat;  
string attoCessazioneAnnullamentoMatrimonioProvincialeItaliana =  
    resp.attoCessazioneAnnullamentoMatrimonioProvincialeItaliana;  
string localitaAttoCessazioneAnnullamentoMatrimonio =  
    resp.attoCessazioneAnnullamentoMatrimonioComunelItalianoLocalita;  
string ufficioAttoCessazioneAnnullamentoMatrimonio =  
    resp.attoCessazioneAnnullamentoMatrimonioUfficio;  
string annoAttoCessazioneAnnullamentoMatrimonio = resp.attoCessazioneAnnullamentoMatrimonioAnno;  
string parteAttoCessazioneAnnullamentoMatrimonio = resp.attoCessazioneAnnullamentoMatrimonioParte;  
string serieAttoCessazioneAnnullamentoMatrimonio = resp.attoCessazioneAnnullamentoMatrimonioSerie;  
string numeroAttoCessazioneAnnullamentoMatrimonio =  
    resp.attoCessazioneAnnullamentoMatrimonioNumero;  
string volumeAttoCessazioneAnnullamentoMatrimonio =  
    resp.attoCessazioneAnnullamentoMatrimonioVolume;  
string dataAttoCessazioneAnnullamentoMatrimonio = resp.attoCessazioneAnnullamentoMatrimonioData;  
DateTime dataAttoCessazioneAnnullamentoMatrimonioData =  
    resp.attoCessazioneAnnullamentoMatrimonioDataAsDate();  
string trascittoAttoCessazioneAnnullamentoMatrimonio =  
    resp.attoCessazioneAnnullamentoMatrimonioTrascritto;
```

}

}

}

}

API Javascript

```
let endpoint= "http://84.240.190.236/api/ST008";
```

```
//Ricerca per Codice Fiscale
```

```
var criteri={comunelstat: "040007", codiceFiscale: "SGGNNA90R50F839T"};
```

```
//Ricerca per IDANPR
```

```
var criteri={comunelstat: "040007", idANPR: "EV62727VC"};
```

```
//Ricerca per Generalità - Esempio Cittadino nato in Italia
```

```
var generalita = {  
  "nominativo": {  
    "nome": "ANNA",  
    "cognome": "SOGGETTA",  
  },  
  "sesso": "F",  
  "dataNascita": {  
    "dataEvento": "1990-10-10",  
  },  
  "luogoNascita": {  
    "comune": {  
      "nomeComune": "NAPOLI",  
      "siglaProvincialstat": "NA",  
    }  
  }  
};
```

```
//Ricerca per Generalità - Esempio Cittadino nato all'Estero
```

```
var generalita = {  
  "nominativo": {  
    "nome": "BLANCA",  
    "cognome": "DELGADO",  
  },  
  "sesso": "F",  
  "dataNascita": {  
    "dataEvento": "1970-05-20",  
  },  
  "luogoNascita": {  
    "localita": {  
      "descrizioneLocalita": "CITTÀ DEL GUATEMALA",  
      "descrizioneStato": "GUATEMALA",  
    }  
  }  
};
```

```
var criteri={comuneIstat: "040007", generalita: generalita};  
var account={utente: "040007", password: "INIZIALE"};  
var restrizioni={idOperatore: "user00001", idUfficio: 20, 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": {  
    "idOperazioneANPR": "58758039",  
    "idANPR": "EV62727VC",  
    "codiceFiscale": "SGGNNA90R50F839T",  
    "codiceFiscaleValiditaFlag": "1",  
    "codiceFiscaleValiditaFlagDescrizione": "VALIDATO",  
    "codiceFiscaleValidita": "2022-02-04",  
    "cognome": "SOGGETTA",  
    "senzaCognomeFlag": "",  
    "nome": "ANNA",  
    "senzaNomeFlag": "",  
    " Sesso": "F",  
    "nascitaData": "1990-10-10",  
    "nascitaDataSenzaGiorno": "",  
    "nascitaDataSenzaGiornoMese": "",  
    "nascitaLuogoEccezionale": "",  
    "nascitaComuneItaliano": "NAPOLI",  
    "nascitaComuneItalianoCodiceIstat": "063049",  
    "nascitaProvincialeItaliana": "NA",  
    "nascitaComuneItalianoLocalita": "",  
    "nascitaComuneEstero": "",  
    "nascitaStatoEstero": "",  
    "nascitaStatoEsteroCodiceIstat": "",  
    "nascitaProvinciaEstera": "",  
    "soggettoAIRE": "N",  
    "annoEspatrio": "",  
    "idSchedaSoggettoComuneIstat": "",  
    "idSchedaSoggettoComune": ""  
  }  
};
```

"idSchedaSoggetto": "87938835",
"dataDecesso": "",
"statoCivile": "2",
"statoCivileDescr": "Coniugato/a",
"noteStatoCivile": "",
"statoCivileNonDisponibile": "N",
"idANPRConiuge": "",
"codiceFiscaleConiuge": "QRTSGT90A01H501B",
"codiceFiscaleValiditaFlagConiuge": "1",
"codiceFiscaleValiditaFlagDescrizioneConiuge": "VALIDATO",
"codiceFiscaleValiditaConiuge": "",
"cognomeConiuge": "QUARTO",
"senzaCognomeFlagConiuge": "",
"nomeConiuge": "SOGGETTO",
"senzaNomeFlagConiuge": "",
"sessoConiuge": "M",
"nascitaDataConiuge": "1990-01-01",
"nascitaDataSenzaGiornoConiuge": "",
"nascitaDataSenzaGiornoMeseConiuge": "",
"nascitaLuogoEccezionaleConiuge": "",
"nascitaComuneItalianoConiuge": "ROMA",
"nascitaComuneItalianoCodiceIstatConiuge": "058091",
"nascitaProvincialeItalianaConiuge": "RM",
"nascitaComuneItalianoLocalitaConiuge": "",
"nascitaComuneEsteroConiuge": "",
"nascitaStatoEsteroConiuge": "",
"nascitaStatoEsteroCodiceIstatConiuge": "",
"nascitaProvinciaEsteraConiuge": "",
"soggettoAIREConiuge": "N",
"annoEspatrioConiuge": "",
"idSchedaSoggettoComuneIstatConiuge": "",
"idSchedaSoggettoComuneConiuge": "",
"idSchedaSoggettoConiuge": "2760295",
"cittadinanzaStatoConiuge": "",
"cittadinanzaIstatConiuge": "",
"cittadinanzaNoteConiuge": "",
"cittadinanzaDataValiditaConiuge": "",
"ordineMatrimonio": "",
"matrimonioData": "2015-03-20",
"matrimonioDataSenzaGiorno": "",
"matrimonioDataSenzaGiornoMese": "",
"matrimonioLuogoEccezionale": "",
"matrimonioComuneItaliano": "",
"matrimonioComuneItalianoCodiceIstat": "",
"matrimonioProvincialeItaliana": "",
"matrimonioComuneItalianoLocalita": "",
"matrimonioComuneEstero": "",
"matrimonioStatoEstero": "",
"matrimonioStatoEsteroCodiceIstat": "",
"matrimonioProvinciaEstera": "",
"attoMatrimonioComuneItaliano": "PAVULLO NEL FRIGNANO",

```
"attoMatrimonioComuneltalianoCodiceIstat": "036030",  
"attoMatrimonioProvincialeItaliana": "MO",  
"attoMatrimonioComuneltalianoLocalita": "",  
"attoMatrimonioUfficio": "",  
"attoMatrimonioAnno": "2021",  
"attoMatrimonioParte": "2",  
"attoMatrimonioSerie": "2",  
"attoMatrimonioNumero": "44556677",  
"attoMatrimonioVolume": "44",  
"attoMatrimonioData": "2021-10-11",  
"attoMatrimonioTrascritto": "",  
"tipoCessazioneAnnullamentoMatrimonio": "7",  
"cessazioneAnnullamentoMatrimonioDescr": "Scioglimento D.L. 12 settembre 2014, n. 132",  
"dataSentenzaCessazioneAnnullamentoMatrimonio": "",  
"numeroSentenzaCessazioneAnnullamentoMatrimonio": "",  
"tipoTribunaleSentenzaCessazioneAnnullamentoMatrimonio": "",  
"tribunaleSentenzaCessazioneAnnullamentoMatrimonioDescr": "",  
"autoritaSentenzaCessazioneAnnullamentoMatrimonio": "",  
"dataValiditaSentenzaCessazioneAnnullamentoMatrimonio": "",  
"attoCessazioneAnnullamentoMatrimonioComuneltaliano": "PAVULLO NEL FRIGNANO",  
"attoCessazioneAnnullamentoMatrimonioComuneltalianoCodiceIstat": "036030",  
"attoCessazioneAnnullamentoMatrimonioProvincialeItaliana": "MO",  
"attoCessazioneAnnullamentoMatrimonioComuneltalianoLocalita": "",  
"attoCessazioneAnnullamentoMatrimonioUfficio": "",  
"attoCessazioneAnnullamentoMatrimonioAnno": "2021",  
"attoCessazioneAnnullamentoMatrimonioParte": "4",  
"attoCessazioneAnnullamentoMatrimonioSerie": "4",  
"attoCessazioneAnnullamentoMatrimonioNumero": "5678",  
"attoCessazioneAnnullamentoMatrimonioVolume": "444",  
"attoCessazioneAnnullamentoMatrimonioData": "2021-11-15",  
"attoCessazioneAnnullamentoMatrimonioTrascritto": ""
```

```
}
```

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"
- **"ANPR Error"**: errori e anomalie restituite da ANPR a fronte della ricerca effettuata

TABELLA UFFICI COMUNALI

Qui di seguito è riportata la tabella degli Uffici autorizzati all'utilizzo del

- **Indagor Service ST008 - Dati di Matrimonio**

Nella richiesta è necessario riportare il numero corrispondente all'Ufficio che la inoltra.

2	Sociali/Sociosanitari – Welfare
5	Tributi
6	Cimiteri
11	Personale – Risorse Umane
13	Espropri
20	Polizia Locale – Attività di Polizia Giudiziaria e Servizi della Centrale Operativa