



Indagor Service

- ST006 •

ACQUISIZIONE PUNTUALE DATI DI STATO DI FAMIGLIA

Specifiche di Integrazione

INDICE

Descrizione.....	1
Contenuto della Richiesta.....	1
Contenuto delle Eccezioni.....	4
Contenuto della Risposta.....	4
Esempi di Integrazione.....	7
API Java.....	7
API .Net (C#).....	11
API Javascript.....	15
Tabella Uffici Comunali.....	19

ST006

Dati di STATO DI FAMIGLIA

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)
- Legame Soggetto

E per ogni eventuale componente del Foglio Famiglia i seguenti gruppi di dati

- ID ANPR
- Generalità
- Legame Soggetto

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***

RespST006 ST006(String idOperatore, int idUfficio, String comunelstat, String codiceFiscale, String motivoRichiesta)

- **ST006:** 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**
- **codiceFiscale:** Codice Fiscale del cittadino
 - Ambiente di test: Codice Fiscale (Esempio **RSSLGU90C24H501L**)
- **motivoRichiesta:** Testo libero (ad esempio "Numero Pratica" a cui si riferisce l'acquisizione dati)

Richiesta per IDANPR

RespST006 ST006_IdANPR(String idOperatore, int idUfficio, String comunelstat, String idANPR, String motivoRichiesta)

- **ST006_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)
- **comunelstat:** 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 **TR76929XO**)
- **motivoRichiesta:** Testo libero (ad esempio "Numero Pratica" a cui si riferisce l'acquisizione dati)

Richiesta per Generalità

RespST006 ST006(String idOperatore, int idUfficio, String comunelstat, Generalita generalita, String motivoRichiesta)

- **ST006:** 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 **ROSSI**
 - **nome:** Nome del cittadino
 - Ambiente di test: Nome **LUIGI**
 - **Sesso:** Sesso del cittadino
 - Ambiente di test: Sesso **M**
 - **dataNascita:** Data di Nascita del cittadino
 - Ambiente di test: Data di Nascita **1990-03-24**
 - **comuneNascita:** Comune di Nascita del cittadino
 - Ambiente di test: Comune di Nascita **ROMA**
 - **siglaProvinciaNascita:** Sigla della Provincia del Comune di Nascita del cittadino
 - Ambiente di test: Sigla della Provincia del Comune di Nascita **RM**
- **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 ST006; 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
Legame Soggetto			
Tipo Legame Soggetto Descrizione			
Data Decorrenza Famiglia/Convivenza Soggetto			Formato aaaa-mm-gg
Codice Legame Soggetto			
Codice Legame Soggetto Descrizione			
Progressivo Componente Soggetto			
Data Decorrenza Legame Soggetto			Formato aaaa-mm-gg

Se presente il secondo componente seguiranno i seguenti gruppi di dati:

ID ANPR (secondo componente famiglia)

Vedi ID ANPR sopra

Generalità (secondo componente famiglia)

Vedi Generalità sopra

Legame Soggetto (secondo componente famiglia)

Vedi Legame Soggetto sopra

Se presente il terzo componente seguiranno i seguenti gruppi di dati:

ID ANPR (terzo componente famiglia)

Vedi ID ANPR sopra

Generalità (terzo componente famiglia)

Vedi Generalità sopra

Legame Soggetto (terzo componente famiglia)

Vedi Legame Soggetto sopra

E così via per eventuali altri componenti della Famiglia

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){
        TestST006();
    }
    public static void TestST006(){
        Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
        RespST006 resp=null;
        try{
            String idOperatore = "user00001";
            int idUfficio = 20;

            //Ricerca per Codice Fiscale
            resp = cl.ST006(idOperatore, idUfficio, "040007", "RSSLGU90C24H501L", "Pratica N. 987654321");

            //Ricerca per IDANPR
            resp = cl.ST006_IdANPR(idOperatore, idUfficio, "040007", "TR76929XO", "Pratica N. 987654321");

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

```
//Esempio Cittadino nato in Italia
generalita.nominativo.cognome = "ROSSI";
generalita.nominativo.nome = "LUIGI";
generalita.sesso = "M";
generalita.dataNascita.dataEvento = "1990-03-24";
generalita.luogoNascita.comune.nomeComune = "ROMA";
generalita.luogoNascita.comune.siglaProvincia = "RM";

//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.ST006(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

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 comuneNascitaEestero=resp.getNascitaComuneEestero();
    String statoNascita=resp.getNascitaStatoEestero();
    String nascitaEsteriIstat=resp.getNascitaStatoEesteroCodiceIstat();
    String nascitaProvinciaEsteri=resp.getNascitaProvinciaEsteri();
    String soggettoAIRE=resp.getSoggettoAIRE();
    String annoEspatrio=resp.getAnnoEspatrio();
    String idSoggettoComuneIstat=resp.getIdSchedaSoggettoComuneIstat();
    String idSoggettoComune=resp.getIdSchedaSoggettoComune();
    String idSchedaSoggetto=resp.getIdSchedaSoggetto();
    String tipoLegameSoggetto = resp.getTipoLegameSoggetto();
    String tipoLegameSoggettoDescr = resp.getTipoLegameSoggettoDescr();
    String dataDecorrenzaFamigliaConvivenzaSoggetto = resp.getDataDecorrenzaFamigliaConvivenzaSoggetto();
    LocalDate dataDecorrenzaFamigliaConvivenzaSoggettoData =
    resp.getDataDecorrenzaFamigliaConvivenzaSoggettoAsDate();
    String dataDecesso = resp.getDataDecesso();
    LocalDate dataDecessoData = resp.getDataDecessoAsDate();
    String codiceLegameSoggetto = resp.getCodiceLegameSoggetto();
    String codiceLegameSoggettoDescr = resp.getCodiceLegameSoggettoDescr();
}
```

```
String progressivoComponenteSoggetto = resp.getProgressivoComponenteSoggetto();
String dataDecorrenzaLegameSoggetto = resp.getDataDecorrenzaLegameSoggetto();
LocalDate dataDecorrenzaLegameSoggettoData = resp.getDataDecorrenzaLegameSoggettoAsDate();

for (int i=0; i< resp.getSoggetto().length; i++)
{
    String idANPRSoggetto = resp.getSoggetto()[i].getIdANPR();
    String codicefiscaleSoggetto = resp.getSoggetto()[i].getCodiceFiscale();
    String flagCodFiscValSoggetto = resp.getSoggetto()[i].getCodiceFiscaleValiditaFlag();
    String flagCodFiscDescrSoggetto = resp.getSoggetto()[i].getCodiceFiscaleValiditaFlagDescrizione();
    String codiceFiscaleValiditaSoggetto = resp.getSoggetto()[i].getCodiceFiscaleValidita();
    LocalDate codiceFiscaleValiditaSoggettoData = resp.getSoggetto()[i].getCodiceFiscaleValiditaAsDate();
    String cognomeSoggetto = resp.getSoggetto()[i].getCognome();
    String flagSenzaCognomeSoggetto = resp.getSoggetto()[i].getSenzaCognomeFlag();
    String nomeSoggetto = resp.getSoggetto()[i].getNome();
    String flagSenzaNomeSoggetto = resp.getSoggetto()[i].getSenzaNomeFlag();
    String sessoSoggetto = resp.getSoggetto()[i].getSesso();
    String dataNascitaSoggetto = resp.getSoggetto()[i].getNascitaData();
    LocalDate dataNascitaSoggettoData = resp.getSoggetto()[i].getNascitaDataAsDate();
    String flagSenzaGiornoSoggetto = resp.getSoggetto()[i].getNascitaDataSenzaGiorno();
    String flagSenzaGiornoMeseSoggetto = resp.getSoggetto()[i].getNascitaDataSenzaGiornoMese();
    String nascitaLuogoEccezionaleSoggetto = resp.getSoggetto()[i].getNascitaLuogoEccezionale();
    String comuneNascitaSoggetto = resp.getSoggetto()[i].getNascitaComuneItaliano();
    String nascitaIT_IstatSoggetto = resp.getSoggetto()[i].getNascitaComuneItalianoCodiceIstat();
    String nascitaProvincialeItalianaSoggetto = resp.getSoggetto()[i].getNascitaProvincialeItaliana();
    String localitaSoggetto = resp.getSoggetto()[i].getNascitaComuneItalianoLocalita();
    String comuneNascitaEsteroSoggetto = resp.getSoggetto()[i].getNascitaComuneEstero();
    String statoNascitaSoggetto = resp.getSoggetto()[i].getNascitaStatoEstero();
    String nascitaEsteroIstatSoggetto = resp.getSoggetto()[i].getNascitaStatoEsteroCodiceIstat();
    String nascitaProvinciaEsteraSoggetto = resp.getSoggetto()[i].getNascitaProvinciaEstera();
    String soggettoAIRESoggetto = resp.getSoggetto()[i].getSoggettoAIRE();
    String annoEspatrioSoggetto = resp.getSoggetto()[i].getAnnoEspatrio();
    String idSoggettoComuneIstatSoggetto = resp.getSoggetto()[i].getIdSchedaSoggettoComuneIstat();
    String idSoggettoComuneSoggetto = resp.getSoggetto()[i].getIdSchedaSoggettoComune();
    String idSchedaSoggettoSoggetto = resp.getSoggetto()[i].getIdSchedaSoggetto();
    String tipoLegameSoggetto = resp.getSoggetto()[i].getTipoLegameSoggetto();
    String tipoLegameSoggettoDescr = resp.getSoggetto()[i].getTipoLegameSoggettoDescr();
    String dataDecorrenzaFamigliaConvivenzaSoggetto =
resp.getSoggetto()[i].getDataDecorrenzaFamigliaConvivenzaSoggetto();
    LocalDate dataDecorrenzaFamigliaConvivenzaSoggettoData =
resp.getSoggetto()[i].getDataDecorrenzaFamigliaConvivenzaSoggettoAsDate();
    String codiceLegameSoggetto = resp.getSoggetto()[i].getCodiceLegameSoggetto();
    String codiceLegameSoggettoDescr = resp.getSoggetto()[i].getCodiceLegameSoggettoDescr();
    String progressivoComponenteSoggetto = resp.getSoggetto()[i].getProgressivoComponenteSoggetto();
    String dataDecorrenzaLegameSoggetto = resp.getSoggetto()[i].getDataDecorrenzaLegameSoggetto();
    LocalDate dataDecorrenzaLegameSoggettoData =
resp.getSoggetto()[i].getDataDecorrenzaLegameSoggettoAsDate();
}
}
}
```

API .Net (C#)

Si aggiunge al progetto la libreria:

- **SteseiGateway.dll**

Si richiama nel seguente modo:

```
{
    class Program{
        static void Main(string[] args){
            TestST006();
        }
        static void TestST006(){
            Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
            RespST006 resp = null;
            try{
                string idOperatore = "user00001";
                int idUfficio = 20;

                //Ricerca per Codice Fiscale
                resp = cl.ST006(idOperatore, idUfficio, "040007", "RSSLGU90C24H501L", "Pratica N. 987654321");

                //Ricerca per IDANPR
                resp = cl.ST006_IdANPR(idOperatore, idUfficio, "040007", "TR76929XO", "Pratica N. 987654321");

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

                //Esempio Cittadino nato in Italia
                generalita.nominativo.cognome = "ROSSI";
                generalita.nominativo.nome = "LUIGI";
                generalita.sesso = "M";
                generalita.dataNascita.dataEvento = "1990-03-24";
                generalita.luogoNascita.comune.nomeComune = "ROMA";
                generalita.luogoNascita.comune.siglaProvincia = "RM";

                //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.ST006(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 tipoLegameSoggetto = resp.tipoLegameSoggetto;
string tipoLegameSoggettoDescr = resp.tipoLegameSoggettoDescr;
string dataDecorrenzaFamigliaConvivenzaSoggetto = resp.dataDecorrenzaFamigliaConvivenzaSoggetto;
DateTime dataDecorrenzaFamigliaConvivenzaSoggettoData =
    resp.dataDecorrenzaFamigliaConvivenzaSoggettoAsDate();

string dataDecesso = resp.dataDecesso;
DateTime dataDecessoData = resp.dataDecessoAsDate();
string codiceLegameSoggetto = resp.codiceLegameSoggetto;
string codiceLegameSoggettoDescr = resp.codiceLegameSoggettoDescr;
string progressivoComponenteSoggetto = resp.progressivoComponenteSoggetto;
string dataDecorrenzaLegameSoggetto = resp.dataDecorrenzaLegameSoggetto;
DateTime dataDecorrenzaLegameSoggettoData = resp.dataDecorrenzaLegameSoggettoAsDate();

for (int i = 0; i < resp.soggetto.Length; i++){
    string idANPRSoggetto = resp.soggetto[i].idANPR;
    string codicefiscaleSoggetto = resp.soggetto[i].codiceFiscale;
    string flagCodFiscValSoggetto = resp.soggetto[i].codiceFiscaleValiditaFlag;
    string flagCodFiscDescrSoggetto = resp.soggetto[i].codiceFiscaleValiditaFlagDescrizione;
    string codiceFiscaleValiditaSoggetto = resp.soggetto[i].codiceFiscaleValidita;
    DateTime codiceFiscaleValiditaSoggettoData = resp.soggetto[i].codiceFiscaleValiditaAsDate();
    string cognomeSoggetto = resp.soggetto[i].cognome;
```

```
string flagSenzaCognomeSoggetton = resp.soggetto[i].senzaCognomeFlag;
string nomeSoggetton = resp.soggetto[i].nome;
string flagSenzaNomeSoggetton = resp.soggetto[i].senzaNomeFlag;
string sessoSoggetton = resp.soggetto[i].sesso;
string dataNascitaSoggetton = resp.soggetto[i].nascitaData;
DateTime dataNascitaSoggettonData = resp.soggetto[i].nascitaDataAsDate();
string flagSenzaGiornoSoggetton = resp.soggetto[i].nascitaDataSenzaGiorno;
string flagSenzaGiornoMeseSoggetton = resp.soggetto[i].nascitaDataSenzaGiornoMese;
string nascitaLuogoEccezionaleSoggetton = resp.soggetto[i].nascitaLuogoEccezionale;
string comuneNascitaSoggetton = resp.soggetto[i].nascitaComuneItaliano;
string nascitaT_IstatSoggetton = resp.soggetto[i].nascitaComuneItalianoCodiceIstat;
string nascitaProvincialeItalianaSoggetton = resp.soggetto[i].nascitaProvincialeItaliana;
string localitaSoggetton = resp.soggetto[i].nascitaComuneItalianoLocalita;
string comuneNascitaEsteroSoggetton = resp.soggetto[i].nascitaComuneEstero;
string statoNascitaSoggetton = resp.soggetto[i].nascitaStatoEstero;
string nascitaEsteroIstatSoggetton = resp.soggetto[i].nascitaStatoEsteroCodiceIstat;
string nascitaProvinciaEsteraSoggetton = resp.soggetto[i].nascitaProvinciaEstera;
string soggettoAIRESoggetton = resp.soggetto[i].soggettoAIRE;
string annoEspatrioSoggetton = resp.soggetto[i].annoEspatrio;
string idSoggettoComuneIstatSoggetton = resp.soggetto[i].idSchedaSoggettoComuneIstat;
string idSoggettoComuneSoggetton = resp.soggetto[i].idSchedaSoggettoComune;
string idSchedaSoggettoSoggetton = resp.soggetto[i].idSchedaSoggetto;
string tipoLegameSoggetton = resp.soggetto[i].tipoLegameSoggetto;
string tipoLegameSoggettonDescr = resp.soggetto[i].tipoLegameSoggettoDescr;
string dataDecorrenzaFamigliaConvivenzaSoggetton =
    resp.soggetto[i].dataDecorrenzaFamigliaConvivenzaSoggetto;
DateTime dataDecorrenzaFamigliaConvivenzaSoggettonData =
    resp.soggetto[i].dataDecorrenzaFamigliaConvivenzaSoggettoAsDate();
string codiceLegameSoggetton = resp.soggetto[i].codiceLegameSoggetto;
string codiceLegameSoggettonDescr = resp.soggetto[i].codiceLegameSoggettoDescr;
string progressivoComponenteSoggetton = resp.soggetto[i].progressivoComponenteSoggetto;
string dataDecorrenzaLegameSoggetton = resp.soggetto[i].dataDecorrenzaLegameSoggetto;
DateTime dataDecorrenzaLegameSoggettonData =
    resp.soggetto[i].dataDecorrenzaLegameSoggettoAsDate();
    }
}
}
```

API Javascript

```
let endpoint= "http://84.240.190.236/api/ST006";
```

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

```
var criteri={comunelstat: "040007", codiceFiscale: "RSSLGU90C24H501L"};
```

```
//Ricerca per IDANPR
```

```
var criteri={comunelstat: "040007", idANPR: "TR76929XO"};
```

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

```
var generalita = {  
  "nominativo": {  
    "nome": "LUIGI",  
    "cognome": "ROSSI",  
  },  
  "sesso": "M",  
  "dataNascita": {  
    "dataEvento": "1990-03-24",  
  },  
  "luogoNascita": {  
    "comune": {  
      "nomeComune": "ROMA",  
      "siglaProvincialstat": "RM",  
    },  
  },  
};
```

```
//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": "TR76929XO",  
    "codiceFiscale": "RSSLGU90C24H501L",  
    "codiceFiscaleValiditaFlag": "1",  
    "codiceFiscaleValiditaFlagDescrizione": "VALIDATO",  
    "codiceFiscaleValidita": "2022-02-04",  
    "cognome": "ROSSI",  
    "senzaCognomeFlag": "",  
    "nome": "LUIGI",  
    "senzaNomeFlag": "",  
    " Sesso": "M",  
    "nascitaData": "1990-03-24",  
    "nascitaDataSenzaGiorno": "",  
    "nascitaDataSenzaGiornoMese": "",  
    "nascitaLuogoEccezionale": "",  
    "nascitaComuneItaliano": "ROMA",  
    "nascitaComuneItalianoCodiceIstat": "058091",  
    "nascitaProvincialeItaliana": "RM",  
    "nascitaComuneItalianoLocalita": "",  
    "nascitaComuneEstero": "",  
    "nascitaStatoEstero": "",  
    "nascitaStatoEsteroCodiceIstat": "",  
    "nascitaProvinciaEstera": "",  
    "soggettoAIRE": "N",  
    "annoEspotrio": ""  
  }  
};
```

```
"idSchedaSoggettoComuneIstat": "",
"idSchedaSoggettoComune": "",
"idSchedaSoggetto": "93106839",
"dataDecesso": "",
"codiceLegameSoggetto": "1",
"codiceLegameSoggettoDescr": "Intestario Scheda",
"progressivoComponenteSoggetto": "",
"dataDecorrenzaLegameSoggetto": "",
"soggetto": [{
  "idANPR": "",
  "codiceFiscale": "NRECLD90R50H501G",
  "codiceFiscaleValiditaFlag": "1",
  "codiceFiscaleValiditaFlagDescrizione": "VALIDATO",
  "codiceFiscaleValidita": "2022-02-04",
  "cognome": "NERI",
  "senzaCognomeFlag": "",
  "nome": "CLAUDIA",
  "senzaNomeFlag": "",
  "sesso": "F",
  "nascitaData": "1990-10-10",
  "nascitaDataSenzaGiorno": "",
  "nascitaDataSenzaGiornoMese": "",
  "nascitaLuogoEccezionale": "",
  "nascitaComuneItaliano": "ROMA",
  "nascitaComuneItalianoCodiceIstat": "058091",
  "nascitaProvincialeItaliana": "RM",
  "nascitaComuneItalianoLocalita": "",
  "nascitaComuneEstero": "",
  "nascitaStatoEstero": "",
  "nascitaStatoEsteroCodiceIstat": "",
  "nascitaProvinciaEstera": "",
  "soggettoAIRE": "N",
  "annoEspatrio": "",
  "idSchedaSoggettoComuneIstat": "",
  "idSchedaSoggettoComune": "",
  "idSchedaSoggetto": "93106840",
  "tipoLegameSoggetto": "3",
  "tipoLegameSoggettoDescr": "Famiglia",
  "dataDecorrenzaFamigliaConvivenzaSoggetto": "2022-02-04",
  "codiceLegameSoggetto": "23",
  "codiceLegameSoggettoDescr": "Convivente (con vincoli di adozione o affettivi)",
  "progressivoComponenteSoggetto": "",
  "dataDecorrenzaLegameSoggetto": ""
}], {
  "idANPR": "",
  "codiceFiscale": "RSSGLC20T24H501W",
  "codiceFiscaleValiditaFlag": "1",
  "codiceFiscaleValiditaFlagDescrizione": "VALIDATO",
  "codiceFiscaleValidita": "2022-02-04",
```

```
"cognome": "ROSSI",  
"senzaCognomeFlag": "",  
"nome": "GIANLUCA",  
"senzaNomeFlag": "",  
"sesso": "M",  
"nascitaData": "2020-12-24",  
"nascitaDataSenzaGiorno": "",  
"nascitaDataSenzaGiornoMese": "",  
"nascitaLuogoEccezionale": "",  
"nascitaComuneItaliano": "ROMA",  
"nascitaComuneItalianoCodiceIstat": "058091",  
"nascitaProvinciaItaliana": "RM",  
"nascitaComuneItalianoLocalita": "",  
"nascitaComuneEstero": "",  
"nascitaStatoEstero": "",  
"nascitaStatoEsteroCodiceIstat": "",  
"nascitaProvinciaEstera": "",  
"soggettoAIRE": "N",  
"annoEspatrio": "",  
"idSchedaSoggettoComuneIstat": "",  
"idSchedaSoggettoComune": "",  
"idSchedaSoggetto": "93106841",  
"tipoLegameSoggetto": "3",  
"tipoLegameSoggettoDescr": "Famiglia",  
"dataDecorrenzaFamigliaConvivenzaSoggetto": "2022-02-04",  
"codiceLegameSoggetto": "3",  
"codiceLegameSoggettoDescr": "Figlio / Figlia",  
"progressivoComponenteSoggetto": "",  
"dataDecorrenzaLegameSoggetto": ""  
}  
}  
}
```

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 ST006 - Dati di Stato di Famiglia**

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

1	Scuola e Pubblica Istruzione
2	Sociali/Sociosanitari – Welfare
5	Tributi
6	Cimiteri
9	Avvocatura e Uffici competenti per Ordinanze – Ingiunzioni
10	Notifiche e Spedizioni
11	Personale – Risorse Umane
13	Espropri
14	Igiene e Sanità Pubblica – Animali d'affezione - Inquinamento
16	Stazione Unica Appaltante
20	Polizia Locale – Attività di Polizia Giudiziaria e Servizi della Centrale Operativa
21	Polizia Locale – Attività Amministrative