Indagor Service
• ST027 •

INVIO COORDINATE DEGLI ACCESSI AD ANNCSU

Specifiche di Integrazione

INDICE

ST027
Invio Coordinate degli Accessi ad ANNCSU

DESCRIZIONE

Il servizio consente di inoltrare le coordinate degli accessi ad ANNCSU

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

ST027(String idOperatore, String comuneIstat, String progressivoCivico, String coordinataX, String coordinataY, String coordinataZ, String metodoRilevazione)

- **ST027**: Nome dell'Indagor Service richiamato
- **idOperatore**: Codice ID assegnato dall'applicativo all'operatore del Comune che inoltra la richiesta
- **comuneIstat**: Codice Istat del Comune fruitore dell'applicativo che inoltra la chiamata
  - Ambiente di test: Codice Istat 040007
- **progressivoCivico**: Progressivo Civico: codice che identifica univocamente la terna (Comune, odonimo, numero civico) a livello nazionale
  - Ambiente di test: (Esempio 31040739)
- **coordinataX**: Longitudine. Valori ammessi 6.0 ≤ x ≤ 18.0
  - Ambiente di test: Longitudine (Esempio 12.24721)
- **coordinataY**: Latitudine. Valori ammessi 36.0 ≤ y ≤ 47.0
  - Ambiente di test: Latitudine (Esempio 44.1333)
- **coordinataZ**: Altitudine espressa in metri
  - Ambiente di test: Altitudine (Esempio 44)
- **metodoRilevazione**: Metodo di Rilevazione. Numero intero da 1 a 4
  - Ambiente di test: Metodo di Rilevazione (Esempio 4)
    Valori ammessi:
    1. Rilevazione strumentale sul campo, con accuratezza < 5 m
    2. Rilevazione strumentale sul campo, con accuratezza ≥ 5 m
    3. Derivazione indiretta da base dati territoriale, con accuratezza stimabile < 5 m
    4. Derivazione indiretta da base dati territoriale, con accuratezza stimabile ≥ 5 m

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"
    - "Progressivo civico deve essere composto da massimo 15 cifre"
    - "Il campo X deve essere un valore di longitudine ammesso in Italia 6.0 ≤ x ≤ 18.0"
    - "Il campo Y deve essere un valore di latitudine ammesso in Italia 36.0 ≤ y ≤ 47.0"
    - "L'altitudine deve essere un numero"
    - "Il campo Metodo deve essere un numero da 1 a 4"
- **ADEException**: errori e anomalie restituite da Agenzia delle Entrate fronte della ricerca effettuata

CONTENUTO DELLA RISPOSTA

Di seguito sono riportati i dati che compongono la risposta resa dal servizio ST027; 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)

Dati	progressivoCivico
	codiceCivicoComunale
	numeroCivico
	esponente
	specificita
	metrico
	sezioneCensimento
	dataInizio
	dataFine: ,
	progressivoNazionale
	codiceCatastale
	coordinataX
	coordinataY
	coordinataZ
	metodoRilevazione
	dataInizioValiditaAmministrativa
	dataFineValiditaAmministrativa
	isolato

ESEMPI DI INTEGRAZIONE

API Java

Si aggiunge al progetto la libreria:

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

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){
		TestST027();
	}
	public static void  TestST027()
	{
		Client cl = new Client("http://84.240.190.236", "040007", "INIZIALE");
		RespST027 resp=null;
		Try
		{
			String idOperatore = "user00001";
			resp = cl.ST027(idOperatore, idUfficio, "040007", "31040739", "12.24721", "44.1333", "44", "4");
		}
		catch (ConnectionException econn)
		{
			String msg=econn.getMessage();  //errore di connessione
		}
		catch (ServerException eserver)
		{
			String msg=eserver.getMessage();  //errore interno del server
		}
		catch (AuthenticationException eauth)
		{
			String msg=eauth.getMessage();  //utente e/o password errati
		}
```

```
            catch (PermissionDeniedException edenied)
            {
                  String msg=edenied.getMessage();  //operazione non permessa (es: comune non disponibile)
            }
            catch (RequestException erequest)
            {
                  String msg=erequest.getMessage();  //dati mancanti nella richiesta o richiesta vuota)
            }
            catch (ADEException eade)
            {
                  String msg=eanpr.getMessage();  //errori, anomalie restituiti da Agenzia delle Entrate
            }



            // risposta
            if (resp != null)
            {
                  for (int i = 0; i< resp.Dati.length; i++)
                  {
                        String progressivoCivico = resp.Dati[i].getProgressivoCivico();
                        String codiceCivicoComunale = resp.Dati[i].getCodiceCivicoComunale();
                        String numeroCivico = resp.Dati[i].getNumeroCivico();
                        String esponente = resp.Dati[i].getEsponente();
                        String specificita = resp.Dati[i].getSpecificita();
                        String metrico = resp.Dati[i].getMetrico();
                        String sezioneCensimento = resp.Dati[i].getSezioneCensimento();
                        String dataInizio = resp.Dati[i].getDataInizio();
                        LocalDateTime dataInizioData = resp.Dati[i].getDataInizioAsDate();
                        String dataFine = resp.Dati[i].getDataFine();
                        LocalDateTime dataFineData = resp.Dati[i].getDataFineAsDate();
                        String progressivoNazionale = resp.Dati[i].getProgressivoNazionale();
                        String codiceCatastale = resp.Dati[i].getCodiceCatastale();
                        String coordinataX = resp.Dati[i].getCoordinataX();
                        String coordinataY = resp.Dati[i].getCoordinataY();
                        String coordinataZ = resp.Dati[i].getCoordinataZ();
                        String metodoRilevazione = resp.Dati[i].getMetodoRilevazione();
                        String dataInizioValiditaAmministrativa = resp.Dati[i].getDataInizioValiditaAmministrativa();
                        LocalDate dataInizioValiditaAmministrativaData = resp.Dati[i].getDataInizioValiditaAmministrativaAsDate();
                        String dataFineValiditaAmministrativa = resp.Dati[i].getDataFineValiditaAmministrativa();
                        LocalDate dataFineValiditaAmministrativaData = resp.Dati[i].getDataFineValiditaAmministrativaAsDate();
                        String isolato = resp.Dati[i].getIsolato();
                  }
            }
      }
}
```

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)
		{
			TestST027();
		}
		static void TestST027()
		{
			Client cl = new Client(""http://localhost", "040007", "INIZIALE");
			RespST027 resp = null;
			Try
			{
				string idOperatore = "user00001";
				resp = cl.ST027(idOperatore, "040007", "31040739", "12.24721", "44.1333", "44", "4");
			}
			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 (ADEException eade)
			{
				string msg = eade.Message;  //errori, anomalie restituiti da Agenzia delle Entrate
			}
```

```
                    //risposta
                    if (resp != null)
                    {
                        for (int i = 0; i< resp.Dati.Length; i++)
                        {
                            string progressivoCivico = resp.Dati[i].progressivoCivico;
                            string codiceCivicoComunale = resp.Dati[i].codiceCivicoComunale;
                            string numeroCivico = resp.Dati[i].numeroCivico;
                            string esponente = resp.Dati[i].esponente;
                            string specificita = resp.Dati[i].specificita;
                            string metrico = resp.Dati[i].metrico;
                            string sezioneCensimento = resp.Dati[i].sezioneCensimento;
                            string dataInizio = resp.Dati[i].dataInizio;
                            DateTime dataInizioData = resp.Dati[i].DataInizioAsDate();
                            string dataFine = resp.Dati[i].dataFine;
                            DateTime dataFineData = resp.Dati[i].DataFineAsDate();
                            string progressivoNazionale = resp.Dati[i].progressivoNazionale;
                            string codiceCatastale = resp.Dati[i].codiceCatastale;
                            string coordinataX = resp.Dati[i].coordinataX;
                            string coordinataY = resp.Dati[i].coordinataY;
                            string coordinataZ = resp.Dati[i].coordinataZ;
                            string metodoRilevazione = resp.Dati[i].metodoRilevazione;
                            string dataInizioValiditaAmministrativa = resp.Dati[i].dataInizioValiditaAmministrativa;
                            DateTime dataInizioValiditaAmministrativaData =
                            resp.Dati[i].DataInizioValiditaAmministrativaAsDate();
                            string dataFineValiditaAmministrativa = resp.Dati[i].dataFineValiditaAmministrativa;
                            DateTime dataFineValiditaAmministrativaData =
                            resp.Dati[i].DataFineValiditaAmministrativaAsDate();
                            string isolato = resp.Dati[i].isolato;
                        }
                    }
                }
            }
        }
```

API Javascript

```
let endpoint=  "http://84.240.190.236/api/ST027";

var criteri = { comuneIstat: "040007", progressivoCivico: "31040739", coordinataX: "12.24721", coordinataY: "44.1333", coordinataZ:
"44", metodoRilevazione: "4"};
var account = { utente: "040007", password: "INIZIALE" };
var restrizioni = { idOperatore: "user00001" };
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)
        {
                if (!risposta.errore.errore)
                {
                        leggiRisposta(risposta.response);
                }
        },
        error: function (qXHR, textStatus, errorThrown)
        {
                alert("ERR");
        }
});
```

```
function leggiRisposta(resp)
{
	if (resp != null)
	{
		for (let i = 0; i < resp.Dati.length; i++)
		{
			let progressivoCivico = resp.Dati[i].progressivoCivico;
			let codiceCivicoComunale = resp.Dati[i].codiceCivicoComunale;
			let numeroCivico = resp.Dati[i].numeroCivico;
			let esponente = resp.Dati[i].esponente;
			let specificita = resp.Dati[i].specificita;
			let metrico = resp.Dati[i].metrico;
			let sezioneCensimento = resp.Dati[i].sezioneCensimento;
			let dataInizio = resp.Dati[i].dataInizio;
			let dataFine = resp.Dati[i].dataFine;
			let progressivoNazionale = resp.Dati[i].progressivoNazionale;
			let codiceCatastale = resp.Dati[i].codiceCatastale;
			let coordinataX = resp.Dati[i].coordinataX;
			let coordinataY = resp.Dati[i].coordinataY;
			let coordinataZ = resp.Dati[i].coordinataZ;
			let metodoRilevazione = resp.Dati[i].metodoRilevazione;
			let dataInizioValiditaAmministrativa = resp.Dati[i].dataInizioValiditaAmministrativa;
			let dataFineValiditaAmministrativa = resp.Dati[i].dataFineValiditaAmministrativa;
			let isolato = resp.Dati[i].isolato;
		}
	}
}
```

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"
    - "Progressivo civico deve essere composto da massimo 15 cifre"
    - "Il campo X deve essere un valore di longitudine ammesso in Italia 6.0 ≤ x ≤ 18.0"
    - "Il campo Y deve essere un valore di latitudine ammesso in Italia 36.0 ≤ y ≤ 47.0"
    - "L'altitudine deve essere un numero"
    - "Il campo Metodo deve essere un numero da 1 a 4"
- **"ADE Error"**: errori e anomalie restituite da ANPR a fronte della ricerca effettuata