



ROMA SERVIZI PER LA MOBILITA' S.r.l.

Capitolato Speciale di Appalto

Accordo quadro con unico operatore, a norma dell'art. 59 del D.lgs. 36/2023, relativo alla fornitura di un sistema per la rilevazione degli accessi alle ZZTL

Allegato:
Specifica interfacciamento web-service

Direzione Tecnica Mobilità e Ingegneria

ROMA





P@SRI 4.0

Specifica interfacciamento web-service

© Project Automation S.p.A. 2024 - 24

Tutti i diritti sono riservati.

La riproduzione totale o parziale è proibita senza l'autorizzazione scritta del proprietario del copyright.

All rights strictly reserved.

Reproduction or issue to third parties in any form whatever is not permitted without written permission from copyright owner.



ELENCO DELLE REVISIONI

Rev.	Data	Descrizione delle modifiche
0	06/05/2021	Prima emissione
1	07/09/2021	Nel paragrafo 3.3.1 è stato rimarcato che il campo UniqueId è un identificativo unico della autorizzazione fornito dal chiamante. Modificata la signature del metodo delete nel paragrafo 3.3.3 con l'aggiunta del parametro "EntrySourceId". A fattore chiarezza per i clienti, nel paragrafo 0 è stato specificato lo stato in cui può essere un transito. Aggiunto paragrafo 3.4.2 per l'esportazione dei transiti accertati. per l'esportazione delle immagini
2	17/09/2021	Corretto paragrafo 3.3.4
3	30/09/2021	Corretta la stringa relativa al country code ITALIA nel body della richiesta json dei metodi descritti in 3.3.1 e 3.3.2 e 3.3.4. Da IT a ITA. Aggiunta precisazione nei paragrafi 3.2.2Aggiunti paragrafi 3.5.1, 3.5.2 e 3.5.2.1
4	22/10/2021	Modificato il nome di un parametro di input della struttura json nel paragrafo 3.5.2 da UniqueId a TransitId. Nei paragrafi 3.5.1 e 3.5.2 sono stati dettagliati i formati delle strutture json restituite. Inserita nota aggiuntiva al paragrafo 3.1 'WS Note generali'
5	02/12/2021	Modificato il paragrafo 3.5.2 relativo all'invio delle risorse
6	21/03/2022	Modificata la signature del metodo 3.5.1, sempre nel paragrafo 3.5.1 eliminata l'obbligatorietà dei campi plate e platecountry per i sistemi Rosso e Velocità, modificata la signature del metodo 3.5.2, aggiunto paragrafo 3.6.1 'Diagnostica di dettaglio'
7	25/03/2022	Aggiunto paragrafo 3.3.5 ''
8	21/04/2022	Rettifica paragrafo 3.1 togliendo ogni riferimento all'auth token bearer
9	28/06/2022	Aggiunte note al paragrafo 3.4.2 ''
10	06/09/2022	Aggiunto nuovo metodo 3.4.2.3 "
11	23/09/2022	Modifica url metodo
12	12/10/2022	Aggiornato paragrafo 3.2 'WS Anagrafiche' aggiunti nuovi metodi
13	23/11/2022	Aggiornato paragrafo 3.3.6 '', paragrafo 3.3.1 '' inserito elenco codici campo 'Result'
14	21/04/2023	Corretto errore chiamata del metodo <a href="https://<server>/api/authorization/periods/all">https://<server>/api/authorization/periods/all
15	05/05/2023	Rettifica alle segnature dei metodi 3.2.6, 3.2.7, 3.2.8, 3.6, 3.6.1 ed inserito nuovo metodo 3.6.2 'Preleva diagnostica di dettaglio'
16	08/06/2023	Aggiornato messaggio json ottenuto con l'invocazione del metodo 3.4.2.2 ''
17	05/09/2023	Aggiunta nota 'controllo sequenzialità' al metodo 3.4.2.2 ''



18	20/09/2023	Aggiunta nota al paragrafo 3.1 'WS Note generali', Modificato oggetto di ritorno del metodo Elenco Giorni Festivi definita al paragrafo 3.2.7
19	19/10/2023	Aggiornata specifica paragrafo 3.5.2
20	14/02/2024	Aggiornato paragrafo 3.3.6 ', parametro 'EntrySourceId'

Attività	Funzione/Reparto/Ente	N. Cognome	Firma
Emissione/Verifica		G.Fabrizi	<i>G.Fabrizi</i>
Approvazione		F.Zanzottera	<i>F.Zanzottera</i>

Archiviazione elettronica del documento

Nome del File	Q7450-ZF-0005-SF-P@SRI 4.0 Specifica interfacciamento WS.doc
Data di creazione	06/05/2021



SOMMARIO

1.	SCOPO E CAMPO DI APPLICAZIONE	2
2.	ABBREVIAZIONI E DEFINIZIONI	2
3.	DESCRIZIONE FUNZIONALE	2
3.1	WS NOTE GENERALI	2
3.2	WS ANAGRAFICHE	2
3.2.1	<i>ELENCO VARCHI</i>	2
3.2.2	<i>ELENCO ZONE</i>	3
3.2.3	<i>ELENCO PERIODI</i>	3
3.2.4	<i>ELENCO AUTORIZZAZIONI</i>	4
3.2.5	<i>PROFILO ORARIO AUTORIZZAZIONE</i>	5
3.2.6	<i>PROFILO ORARIO ATTIVAZIONE VARCO</i>	5
3.2.7	<i>ELENCO GIORNI FESTIVI</i>	6
3.2.8	<i>ELENCO ECCEZIONI</i>	7
3.5	WS INTEGRAZIONE SOTTOSISTEMI TERZI	8
3.5.1	<i>INSERIMENTO DI UN TRANSITO</i>	8
3.5.2	<i>INSERIMENTO DI UNA RISORSA</i>	10
3.6	WS DIAGNOSTICA STATO OPERATIVO	12
3.6.1	<i>DIAGNOSTICA DI DETTAGLIO</i>	13
3.6.2	<i>PRELEVA DIAGNOSTICA DI DETTAGLIO</i>	13
4.	WS DATI DI TRAFFICO (TRAFFIC DATA API)	15
4.1	INTRODUZIONE	16
4.2	METODO TRAFFICDATA	17
4.2.1	<i>PARAMETRI</i>	18
4.2.2	<i>RISPOSTA</i>	18
4.2.3	<i>ESEMPIO DI INVOCAZIONE E RISPOSTA</i>	20



1. SCOPO E CAMPO DI APPLICAZIONE

Lo scopo del presente documento è la descrizione di dettaglio per la progettazione e lo sviluppo di una nuova funzione software da applicare al sistema P@SRI 4.0.

2. ABBREVIAZIONI E DEFINIZIONI

P@SRI 4.0 Sistema Rilevamento Infrazioni realizzato da Project Automation
SRI P@SRI 4.0

3. DESCRIZIONE FUNZIONALE

Il presente documento descrive le modalità di interfacciamento tra il sistema unico per il sanzionamento denominato **P@SRI 4.0** (Sistema Rilevamento Infrazioni), che nel seguito del documento identificheremo con SRI e sistemi terzi, per lo scambio di dati, utilizzando chiamate web-service.

3.1 *WS Note generali*

Nel presente paragrafo sono elencate le note generali applicate a tutte le web-service elencate:

- Saranno utilizzati servizi web di tipo REST su protocollo https e dove previsto i dati saranno in formato json;
- L'accesso ai metodi è consentito previo utilizzo di apposita 'api key' che deve essere fornita al chiamante e passata nell'header della richiesta http con la chiave 'x-api-key';
- **Si raccomanda che nello sviluppo del servizio di interfacciamento si preveda, quando necessario, un sistema che bufferizzi le richieste qualora per motivi tecnici, es. connettività di rete, i servizi web non dovessero risultare raggiungibili;**
- **Si evidenzia che con il rilascio di nuove revisioni della specifica, potranno essere aggiunti nuovi parametri nella chiamata dei metodi ed anche nuovi chiavi nei dati JSON.**

3.2 *WS Anagrafiche*

Le chiamate ai seguenti metodi permettono di acquisire le anagrafiche contenute nel sistema centrale ed i corrispondenti codici identificativi.

3.2.1 **Elenco varchi**

Il metodo consente di recuperare l'elenco dei varchi di uno specifico sottosistema. Il campo SubSystem indica il sottosistema nel quale effettuare la ricerca (ZTL = 0, Velocità =1, Rosso= 2). Il metodo restituisce una lista con l'elenco dei varchi come riportato in esempio.

Verbo	Url
GET	<a href="https://<server>/api/gates/all?subSystem=<id> sottosistema">https://<server>/api/gates/all?subSystem=<id> sottosistema

[



```
{
  "Lanes": [
    {
      "Id": 1,
      "Direction": 0,
      "Metadata": {}
    }
  ],
  "SubSystem": 0,
  "Metadata": {},
  "IsAuthorizedByDefault": false,
  "Key": "GATE-KEY-028",
  "Name": "Barbarasa"
}
```

3.2.2 Elenco zone

Il metodo consente di recuperare l'elenco delle 'zone', cioè i raggruppamenti logici di varchi identificati con un nome, definiti dall'ordinanza che regola l'accesso alla ZTL o area equivalente. Si rimarca che le zone esistono solo nell'ambito del sistema ZTL, per cui il metodo in oggetto non presenta come parametro il campo "SubSystem" in quanto è automaticamente riferito al sistema ZTL.

Verbo	Url
GET	<a href="https://<server>/api/authorization/zones/all">https://<server>/api/authorization/zones/all

```
[
  {
    "Key": "CORSO-VECCHIO",
    "Name": "Corso Vecchio",
    "Description": "Zona per i solo varco di Corso Vecchio"
  }
]
```

3.2.3 Elenco periodi

Il metodo consente di recuperare l'elenco dei periodi, dove un periodo è identificato da una data iniziale ed eventualmente una data finale e rappresenta il contesto temporale rispetto al quale vengono definite autorizzazioni ed attivazioni nel sistema. Una tipologia di autorizzazione, quindi, potrebbe essere definita su periodi differenti, con diversi profili orari. Si rimarca che i profili di attivazione/autorizzazione non hanno una valenza assoluta, ma relativa ad un determinato periodo temporale.

Verbo	Url
GET	<a href="https://<server>/api/authorization/periods/all">https://<server>/api/authorization/periods/all



```
[
  {
    "Key": "YEAR-2020",
    "Name": "YEAR 2020",
    "ValidFrom": "2015-01-01T00:00:00.000Z",
    "ValidTo": "2015-12-31T00:00:00.000Z"
  },
  ...
]
```

Il campo ValidTo nella response potrebbe essere vuoto, nel caso in cui un periodo non abbia termine: solitamente dato un elenco di periodi ordinati cronologicamente, l'ultimo non ha una data di fine validità (è concettualmente infinito a partire dalla ValidFrom).

3.2.4 Elenco autorizzazioni

Il metodo consente di recuperare l'elenco delle 'fasce orarie', cioè l'elenco delle 'categorie' degli aventi diritto ad accedere alla ZTL e relative fasce orarie nelle quali sono abilitate al transito, definite dall'ordinanza che regola l'accesso alla ZTL o area equivalente.

In analogia al metodo precedente, anche per questo si rimarca che le fasce orarie esistono solo nell'ambito del sistema ZTL, per cui il metodo in oggetto non presenta come parametro il campo "SubSystem" in quanto è automaticamente riferito al sistema ZTL.

Verbo	Url
GET	https://<server>/api/authorization/driver-categories/all

Il metodo restituisce una lista di zone come sotto riportato.

```
[
  {
    "Id": 78,
    "Key": "H24",
    "Name": "H24",
    "Description": "Autorizzato tutti i giorni 24 ore su 24",
    "IsAuthorizedByDefault": true
  }
]
```




3.2.5 Profilo orario autorizzazione

Il metodo consente di reperire le fasce orarie giornaliere nelle quali la 'categoria' di utente è autorizzato, **dato un periodo**. Si rimarca l'importanza del parametro di input che specifica la chiave del periodo, poiché il profilo di attivazione di una categoria è definito non in assoluto ma relativamente ad un determinato periodo temporale:

Verbo	Url
GET	https://<server>/api/authorization/driver-categories/profile? CategoryKey=<ckey>&PeriodKey=<pkey>

La CategoryKey è la chiave della categoria di fascia oraria e la PeriodKey è la chiave che identifica il periodo.

La response del servizio avrà il formato seguente:

```
[  
  {  
    "Day": "LU",  
    "Profile": "00000....11...00"  
  },  
  ...  
]
```

Abbiamo quindi un array composto da 8 elementi: uno per ogni giorno della settimana dal lunedì alla domenica più uno per i giorni festivi. Ogni elemento dell'array sarà composto un campo che identifica la tipologia di giorno (LU/MA/ME/GI/VE/SA/DO e FE per i festivi) ed un altro campo che definisce il profilo di attivazione per l'autorizzazione su quel determinato tipo di giorno (sul periodo specificato). Il profilo di attivazione in particolare è una stringa di 288 caratteri di 0 e 1: è composta in maniera tale da poter specificare se l'autorizzazione è attiva (carattere 1) o meno (0) su intervalli di 5 minuti (si noti che $288 \times 5 = 1440$, ovvero il numero di minuti sulle 24h). Ad esempio un profilo siffatto "010...." specificherà che dalle 00:00 alle 00:05 l'autorizzazione non è attiva (0), mentre lo è dalle 00:05 alle 00:10 (1) e tornerà a non essere più attiva dalle 00:10 alle 00:15 ecc..

3.2.6 Profilo orario attivazione varco

Il metodo consente di reperire le fasce orarie giornaliere, nelle quali, il varco specificato, è attivo, **dato uno periodo**.

Verbo	Url
GET	https://<server>/api/gates/profile? GateKey=<gkey>&LaneId=<lkey>&PeriodKey=<pkey>&subSystem=<skey>

```
[  
  {
```



```
"Day": "LU",  
"Profile": "00000....11...00"  
},  
...  
]
```

Abbiamo quindi un array composto da 8 elementi: uno per ogni giorno della settimana dal lunedì alla domenica più uno per i giorni festivi. Ogni elemento dell'array sarà composto un campo che identifica il giorno (LU/MA/ME/GI/VE/SA/DO e FE per i festivi) ed un altro campo che definisce il profilo di attivazione per il varco, sul periodo specificato, su quel determinato tipo di giorno.

Il profilo di attivazione in particolare è una stringa di 288 caratteri di 0 e 1: è composta in maniera tale da poter specificare se il varco è attivo (carattere 1) o meno (0, dunque non sanziona) su intervalli di 5 minuti (si noti che $288 \times 5 = 1440$, ovvero il numero di minuti sulle 24h). Ad esempio un profilo siffatto "010...." specificherà che dalle 00:00 alle 00:05 il varco non è attivo (0), mentre lo è dalle 00:05 alle 00:10 (1, quindi sanziona) e tornerà a non essere attivo dalle 00:10 alle 00:15 ecc..

3.2.7 Elenco giorni festivi

Il metodo consente di reperire l'elenco dei giorni festivi configurati all'interno del sistema compresi tra le date specificate nei parametri di input:

Verbo	Url
GET	<code>https://<server>/api/authorization/holidays/all?From=<yyyy-MM-dd>&To=<yyyy-MM-dd></code>

```
{  
  "Holidays": [  
    {  
      "Date": "2023-01-01T00:00:00",  
      "Name": "Capodanno 2023",  
      "Type": "SystemHoliday"  
    },  
    {  
      "Date": "2023-07-07T00:00:00",  
      "Name": "PATRONO 2023",  
      "Type": "CustomHoliday",  
      "IsRecurrent": true  
    },  
    {  
      "Date": "2023-07-08T00:00:00",  
      "Name": "FESTIVITA NON RICORRENTE 2023",  
      "Type": "CustomHoliday",  
      "IsRecurrent": false  
    }  
  ]  
}
```



La response sarà una struttura contenente un array di elementi chiamato Holidays, ogni elemento contiene la data della festività, il nome della festività, il tipo che può essere 'SystemHoliday' per le festività preconfigurate nel sistema o 'CustomHoliday' per le festività configurate da operatore e la proprietà 'IsRecurrent' che è presente solo per le Festività di tipo 'CustomHoliday'. Se 'IsRecurrent' ha valore true specifica che la festività ricorre ogni anno nel mese e giorno specificato dal campo Date, se ha valore false la festività sarà limitata unicamente alla data specificata.

3.2.8 Elenco eccezioni

Il metodo consente di reperire l'elenco delle eccezioni di attivazione (e relativo profilo orario) su un dato varco, all'interno di un range di date specificato.

Il significato delle eccezioni è quello di definire un comportamento differente rispetto al profilo di attivazione del varco, definito nei periodi, su determinate giornate, considerate "eccezionali":

Verbo	Url
GET	https://<server>/api/gates/exceptions/list?GateKey=<gateKey>&LaneId=<IKey>&From=<yyyy-MM-dd>&To=<yyyy-MM-dd>&subSystem=<sKey>

```
[  
  {  
    "Date": "2022-09-21",  
    "Profile": "000101010...01010"  
  }  
  ...  
]
```

Il risultato sarà un array nel quale ogni elemento contiene una data (la data oggetto dell'eccezione) ed un profilo di attivazione.

Il profilo di attivazione in particolare è una stringa di 288 caratteri di 0 e 1: è composta in maniera tale da poter specificare se il varco è attivo (carattere 1) o meno (0, dunque non sanziona) su intervalli di 5 minuti (si noti che $288 \times 5 = 1440$, ovvero il numero di minuti sulle 24h). Ad esempio un profilo siffatto "010...." specificherà che dalle 00:00 alle 00:05 il varco non è attivo (0), mentre lo è dalle 00:05 alle 00:10 (1, quindi sanziona) e tornerà a non essere attivo dalle 00:10 alle 00:15 ecc..



3.5 *WS integrazione sottosistemi terzi*

I seguenti metodi consentono l'inserimento dei transiti, delle relative risorse e dei transiti statistici. Si rimarca che questa ultima funzione è disponibile solo per il sistema ZTL in quanto per il controllo di rosso e di velocità, non essendo previste liste di autorizzazione né orari di spegnimento degli apparati, ogni transito rilevato è una potenziale sanzione.

3.5.1 **Inserimento di un transito**

L'inserimento di un transito avviene in due fasi, nella prima viene richiamato un metodo per l'inserimento dei dati del transito, nella seconda vengono inviati con un altro metodo le relative risorse.

Per l'inserimento dei dati del transito viene messo a disposizione il seguente metodo che prende in input la struttura dati json sotto riportata e restituisce lo unique identifier del transito inserito o null se il transito viene autorizzato. L'autorizzazione di un transito è una funzionalità presente solo per il sottosistema ZTL nel quale, all'atto dell'inserimento, vengono controllati sia gli orari di attivazione del varco che eventuali autorizzazioni su base lista bianca.

Negli altri due sottosistemi i transiti vengono tutti inseriti a meno ovviamente di errori di validazione dei dati di input.

Verbo	Url
POST	https://<server>/api/transits/add

La struttura di input è un json costituito dai seguenti campi:

```
{
  "SubSystem": 0,
  "GateKey": "GATE-009",
  "Lane": "1",
  "TimeStamp": "2021-04-25T22:48:00.880",
  "Plate": "AA123AA",
  "PlateCountry": "ITA",
  "OCRConfidence": "100",
  "VehicleType": "AUTOVEICOLO",
  "Direction": "1",
  "ADRCODE": "60-1710",
  "ADRScore": "90",
  "SpeedLimit": "70",
  "DetectedSpeed": "107.12",
  "SecondsAfterRedLight": "2.46",
  "SecondsAfterYellowLight": "5.01",
  "ConfiguredYellowLightTime": "5.00",
  "SerialNumber": "XY123ZZ-2021/09",
  "TrailerPlate": "Z1234",
  "TrailerConfidence": "90"
}
```



La tabella seguente riporta per ciascun campo il significato, gli eventuali valori ammissibili e l'indicazione della obbligatorietà o meno per ciascun sistema.

		Ztl	Rosso	Velocità
SubSystem	Indicazione del tipo di sistema	X	X	X
GateKey	Chiave del varco	X	X	X
Lane	Corsia del varco	X	X	X
TimeStamp	Data e ora del transito	X	X	X
Plate	Targa	X		
PlateCountry	Nazionalità della targa Codice ISO 3166-1 alpha3	X		
OCRConfidence	Grado in percentuale di affidabilità dell'OCR relativo alla targa.	X	X	X
VehicleType	Tipo del veicolo	X	X	X
Direction	Direzione di marcia. I valori possibili sono: <ul style="list-style-type: none">0: non specificato;1: la targa rilevata si riferisce al posteriore del veicolo;2: la targa rilevata si riferisce all'anteriore del veicolo	X		
ADRCODE	Codice KEMLER delle merci pericolose			
ADRScore	Grado in percentuale di affidabilità dell'OCR relativo al codice ADR			
SpeedLimit	Limite di velocità			X
DetectedSpeed	Velocità rilevata in Km/h. Massimo due decimali			X
SecondsAfterRedLight	Numero di secondi del transito dopo l'attivazione della luce rossa. Massimo due decimali		X	
SecondsAfterYellowLight	Numero di secondi del transito dopo l'attivazione della luce giallo. Massimo due decimali		X	
ConfiguredYellowLightTime	Numero di secondi configurati per il tempo di giallo. Massimo due decimali		X	
SerialNumber	Numero seriale dell'apparato.		X	X
TrailerPlate	Targa del rimorchio			
TrailerConfidence	Grado in percentuale di affidabilità dell'OCR relativo alla targa del rimorchio			

In caso di inserimento con esito positivo viene restituito il codice HTTP 201 (Created) e nel body della risposta viene restituito il transito inserito, ovvero i dati passati in input, a cui vengono



aggiunti due campi che sono lo unique identifier del transito e lo stato. Lo stato può assumere tre valori distinti che sono:

- 0 = Suspect. Il transito è a disposizione degli operatori per la verifica e l'accertamento.
- 1 = Moved. A seguito di verifiche sui dati del transito, il medesimo è stato inserito ma con uno speciale stato che indica agli operatori che il transito è attenzionato. Alcuni esempi sono i transiti doppi (molteplici transiti con la stessa targa in un intervallo di tempo ristretto), transiti con data futura o troppo vecchia etc.
- 3 = Authorized. Il transito è stato autorizzato a seguito dei controlli sulle autorizzazioni di attivazione dei varchi e lista bianca. Questo stato è possibile solo nell'ambito del sistema ZTL.

3.5.2 Inserimento di una risorsa

Lo unique identifier restituito dal metodo precedente deve poi essere utilizzato per l'invio delle risorse del transito, con il seguente metodo.

Verbo	Url
POST	https://<server>/api/resources/add

La struttura di input è un json costituito dai seguenti campi:

```
{
  "SubSystem": 0,
  "TransitId": "f54ef5c8-f620-4a70-9a82-b011e5b916fa",
  "Type": "0",
  "Scene": "1",
  "TimeLine": "0",
  "Focal": "0",
  "MimeType": "image/jpeg",
  "RawData": "0xVGt6/%&iii991000xx00.....",
  "ExtraData": {
    "Key1": "Value1",...
  }
}
```

I valori ammissibili dei campi Type, Scene, Timeline e Focal sono i seguenti, seguono le combinazioni possibili in riferimento ai tre sottosistemi:

- "Type" 0 foto, 1 video, 2 file;
- "Scene" 0 non specificata, 1 dettaglio, 2 contesto;
- "TimeLine" per alcuni apparati (rosso) indica l'istante antecedente o successivo all'infrazione. Può assumere i valori da -4 a -1 per i quattro fotogrammi antecedenti e da 1 a 4 per i quattro successivi. Il valore 0 indica timeline non specificata;
- "Focal" per alcuni apparati (robot) indica la focale aperta 1 o chiusa 2. Il valore 0 indica focale non specificata;

Per il sottosistema ZTL sono previste due risorse che si riferiscono alla immagine di dettaglio e a quella di contesto. La tabella seguente riporta i valori da specificare per i vari parametri.



	Type	Scene	Timeline	Focal
Immagine di dettaglio	0	1	0	0
Immagine di contesto	0	2	0	0

Per il sottosistema di velocità sono previste due possibilità:

- due immagini come per ZTL;
- quattro immagini di cui due di contesto e due di dettaglio ma nell'ambito di ciascuna coppia vengono fornite due immagini perfettamente sovrapponibili ma scattate con apertura della focale dell'obiettivo differente.

	Type	Scene	Timeline	Focal
Immagine di dettaglio a focale aperta	0	1	0	1
Immagine di dettaglio a focale chiusa	0	1	0	2
Immagine di contesto a focale aperta	0	2	0	1
Immagine di contesto a focale chiusa	0	2	0	2

Per il sottosistema di rosso sono previste quattro possibilità:

- due immagini come per ZTL;
- quattro immagini come per velocità;
- otto immagini più un video. Le otto immagini sono organizzate in due di dettaglio eseguite un istante immediatamente prima e dopo lo scatto del rosso e le altre sei sono di contesto di cui tre scattate in tre istanti precedenti lo scatto del rosso e le altre tre nei tre istanti successivi;

	Type	Scene	Timeline	Focal
Immagine di dettaglio antecedente il rosso	0	1	-1	0
Immagine di dettaglio successiva al rosso	0	1	1	0
Immagine di contesto antecedente il rosso	0	2	-2	0
Immagine di contesto antecedente il rosso	0	2	-3	0
Immagine di contesto antecedente il rosso	0	2	-4	0
Immagine di contesto successiva al rosso	0	2	2	0
Immagine di contesto successiva al rosso	0	2	3	0
Immagine di contesto successiva al rosso	0	2	4	0
Video	1	0	0	0

In caso di inserimento con esito positivo viene restituito il codice HTTP 201 (Created) e nel body della risposta vengono restituiti i dati della risorsa inserita con il suo unique identifier e ovviamente senza i dati binari del campo RawData.

Nel caso del sistema ZTL, l'inserimento di una risorsa collegata ad un transito che è stato inserito come autorizzato non è consentito e viene restituito il codice http 405 Method Not Allowed.

Indipendentemente dal sottosistema considerato, in aggiunta ai valori di Type indicati in 3.4.3 è possibile specificare il valore 3 per inserire risorse che rappresentano il/i file originali che hanno generato il transito. In questo gli attributi scene, timeline e focal verranno ignorati. In ExtraData ci si aspetta di trovare le seguenti chiavi:

- FileName, nome del file originale
- FileExtension, estensione del file



- FileInfo_CreationTime, FileInfo_LastAccessTime, FileInfo_LastWriteTime che rappresentano rispettivamente la data di creazione, ultimo accesso e ultima scrittura sul file. Qualora non presenti, verrà utilizzata la data e ora corrente per settare tali campi.
- FileHeader, sezione di header del file originale dal quale sono stati estrapolati i dati del transito

	Type	Scene	Timeline	Focal
File originale	3	0	0	0

3.6 *WS Diagnostica stato operativo*

La chiamata ai seguenti metodi consente di inserire informazioni di diagnostica relative alla raggiungibilità via rete (ping) dei dispositivi di rilevamento ed al sincronismo orario.

Verbo	Url
POST	https://<server>/api/diagnostic/

La struttura di input è un json costituito dai seguenti campi:

```
{
  "SubSystem": 0,
  "GateKey": "GATE-009",
  "Lane": "1",
  "Network": "OK",
  "TimeSynchronism": "FAIL",
  "Status": "OK"
}
```

I campi Network e TimeSynchronism possono assumere i valori OK e FAIL ed indicano rispettivamente la condizione di raggiungibilità via rete del dispositivo di rilevamento ed il suo corretto sincronismo orario con un time server di riferimento. Il campo Status può assumere i valori OK, Not-Operating e Degraded ad indicare rispettivamente lo stato di corretto funzionamento dell'apparato, di indisponibilità e di funzionamento degradato.

Il metodo deve essere chiamato ad intervalli regolari per ogni dispositivo in modo da tenere sempre aggiornati i dati di diagnostica. Lo SRI4, una volta superato un intervallo di tempo prefissato e configurabile, se i dati non sono aggiornati li considera non attendibili.



3.6.1 Diagnostica di dettaglio

La chiamata al seguente metodo permette di inviare lo stato diagnostico di dettaglio degli apparati. Per convenzione un valore diagnostico = 1 identifica uno stato di funzionamento regolare, diversamente uno stato diagnostico = 0 identifica uno stato di funzionamento anomalo.

Verbo	Url
POST	https://<server>/api/diagnostic/details/

```
{
  "SubSystem": 0,
  "GateKey": "GATE-009",
  "Lane": "1",
  "doorOpen": 1,
  "timeSyncError": 1,
  "storageAlarm": 1,
  "cu1Detection": 1,
  "cu2Detection": 1,
  "cmeterDetection": 1,
  "cabinetHeater": 1,
  "cabinetFan": 1,
  "camera1Detection": 1,
  "camera2Detection": 1,
  "camera3Detection": 1,
  "camera4Detection": 1,
  "flir1Detection": 1,
  "flir2Detection": 1,
  "redKeyValidation": 1,
  "laneLoopDetection": 1,
  "plateDetection": 1,
  "eventiNonProdotti": 1,
  "timestamp": "2021-04-25T22:48:00.88"
}
```

3.6.2 Preleva diagnostica di dettaglio

La chiamata al seguente metodo permette di ricevere lo stato diagnostico di dettaglio degli apparati. Per convenzione un valore diagnostico = 1 identifica uno stato di funzionamento regolare, diversamente uno stato diagnostico = 0 identifica uno stato di funzionamento anomalo.

Verbo	Url
GET	https://<server>/api/diagnostic/details/?GateKey=<gKey>&LaneId=<IKey>&subSystem=<skey>



Il metodo restituisce nella response un oggetto a contenuto variabile la cui struttura è riportata di seguito.

```
{
  "SubSystem": 0,
  "GateKey": "GATE-009",
  "Lane": "1",
  "doorOpen": 1,
  "timeSyncError": 1,
  "storageAlarm": 1,
  "cu1Detection": 1,
  "cu2Detection": 1,
  "cmeterDetection": 1,
  "cabinetHeater": 1,
  "cabinetFan": 1,
  "camera1Detection": 1,
  "camera2Detection": 1,
  "camera3Detection": 1,
  "camera4Detection": 1,
  "flir1Detection": 1,
  "flir2Detection": 1,
  "redKeyValidation": 1,
  "laneLoopDetection": 1,
  "plateDetection": 1,
  "eventiNonProdotti": 1,
  "timestamp": "2021-04-25T22:48:00.88"
}
```



4. WS DATI DI TRAFFICO (TRAFFIC DATA API)

La collezione di servizi **Traffic Data API** è definita a supporto della condivisione con sistemi di terze parti delle misure di traffico acquisite ed elaborate dal sistema **Misure Traffico**. Il servizio è in grado di restituire, in specifiche strutture JSON, sia le misure complessive aggregate dal sistema rispetto al periodo di aggregazione di riferimento, sia le misure di traffico distinte rispetto allo schema di classificazione adottato dal sistema (9+1 classi, come riportato in Tabella 1). A seconda della tipologia di sensore con cui la postazione di rilevamento dei dati di traffico è equipaggiata, è possibile rilevare differenti proprietà fisiche del veicolo in transito (lunghezza, numero di assi, profilo tridimensionale, ecc.) e di conseguenza assegnare il transito alla classe veicolare più opportuna. Per alcune tipologie di postazione di rilevamento, l'assegnazione non avviene su tutte le classi dello schema di classificazione di riferimento, ma solamente su quelle rilevabili in base alla sensoristica in dotazione.

Sulla base dei transiti rilevati dalle postazioni di rilievo dei flussi veicolari, il sistema elabora le seguenti grandezze di traffico:

- **Conteggio**: numero di veicoli rilevati nel periodo di aggregazione (complessivi, oppure distinti in classi veicolari).
- **Velocità media**: valore medio della velocità rilevata per tutti i transiti nel periodo di aggregazione (complessivi, oppure distinti in classi veicolari), espressa in Km/h.
- **Tasso di occupazione**: indicatore della densità del traffico veicolare, calcolato come percentuale del tempo di occupazione dell'area di rilevamento, rispetto alla durata complessiva del periodo di aggregazione.












ID		Descrizione
0		<i>Not classified</i>
1		<i>Motorcycle</i>
2		<i>Car</i>
3		<i>Van</i>
4		<i>Bus</i>
5		<i>Coach</i>
6		<i>Truck</i>
7		<i>Articulated truck</i>
8		<i>Semi-truck</i>
9		<i>Pedestrian</i>

Tabella 1: Scema di classificazione su 9+1 classi

4.1 Introduzione

I servizi contenuti nella collezione **Traffic Data API** sono di tipo HTTP REST (*REpresentation State Transfer*) con i seguenti parametri di configurazione:

- Method: GET o POST
- Content-type: application/json
- Authorization: oauth2
- Body
 - Struttura in formato JSON, per data payload (dove richiesto)

I codici di errore ritornati dal web service sono conformi agli standard HTTP:



Error code	Descrizione
200	OK. Interrogazione completata secondo i parametri indicati.
400	Bad request , e.g. mancanza di un parametro di interrogazione obbligatorio, o un parametro non è nel formato atteso.
401	Unauthorized , e.g. l'invocazione avviene con credenziali non valide.
404	Not found
500	Internal server error. Errore inatteso durante l'esecuzione dell'interrogazione.

Tabella 2: Codici di errore

L'elemento Body viene utilizzato per descrivere eventuali parametri di invocazione del Web service all'interno di una struttura JSON.

4.2 **Metodo trafficData**

Il metodo **trafficData**, di tipo GET, viene invocato con i parametri in formato JSON riportati in Tabella 3, e restituisce l'insieme delle misure di traffico presenti nel sistema che corrispondono ai criteri di interrogazioni impostati.



4.2.1 Parametri

Parametro	Tipo	Descrizione
fromDate	String (date-time)	(Required) La data iniziale del periodo di interesse (es. "2024-02-01 00:00:00").
toDate	string	(Required) La data finale del periodo di interesse (es. "2024-02-01 00:00:00").
devices	array	Lista dei sensori di interesse, ciascuno indicato con il proprio identificativo numerico univoco, di cui si richiedono le misure di traffico. Se il parametro non è indicato, il sistema restituisce le misure di traffico relative a tutti i sensori.
Page	integer	Il numero della pagina di dati richiesta (se il set di dati risultante dall'interrogazione supera la lunghezza massima consentita in una pagina, il sistema provvede alla paginazione dei dati, per cui i dati in eccesso devono essere recuperati con interrogazioni successive. Se il parametro non è indicato, il metodo restituisce sempre la prima pagina).

Tabella 3: Parametri di invocazione del metodo **trafficData**

4.2.2 Risposta

- Response Content-type: application/json

Il metodo restituisce una struttura **TrafficMeasureSet**, lista di entità **TrafficMeasure** associate ad uno specifico sensore di rilievo dei flussi veicolari, corrispondente all'insieme delle misure di traffico risultanti secondo i parametri di interrogazione utilizzati.

Attributo	Tipo	Descrizione
device	integer	Identificativo numerico univoco del sensore di rilievo dei flussi veicolari (es. "201")
description	string	Descrizione testuale del sensore di rilievo dei flussi veicolari (es. "Varco Viale Roma")
lane	integer	Identificativo numerico della corsia monitorata dal sensore (es "1" per corsia di marcia, "2" per corsia di sorpasso).
period	integer	Il periodo di aggregazione delle misure di traffico, espresse in secondi (es. "300" per misure aggregate ai 5 minuti).



timestamp	string (date-time)	Il timestamp finale del periodo di aggregazione della misura di traffico ("2024-02-07 15:00:00").
transmissionTime	string (date-time)	Il timestamp di trasmissione della misura di traffico ("2024-02-07 15:02:13.000").
aggregatedTrafficMeasure	ref	La struttura JSON AggregatedTrafficMeasure delle misure di traffico complessive nel periodo di aggregazione indicato.
aggregatedTrafficMeasure	array	Lista delle misure di traffico, distinte secondo lo schema di classificazione previsto. Per ogni classe veicolare, la misura è riportata secondo la struttura ClassifiedTrafficMeasures .

Tabella 4: Proprietà dell'entità TrafficMeasureSet

Attributo	Tipo	Descrizione
count	integer	Valore numerico intero, corrispondente al numero complessivo di veicoli rilevati nel periodo di aggregazione.
countUnit	string	Unità di misura del conteggio riportato dalla proprietà count (es. "num").
speed	float	Valore numerico decimale, corrispondente alla velocità media di tutti i veicoli rilevati nel periodo di aggregazione.
speedUnit	string	Unità di misura della velocità media riportata dalla proprietà speed (es. "Km/h").
occupancyRate	float	Valore numerico decimale, corrispondente al tasso di occupazione determinato da tutti i veicoli rilevati nel periodo di aggregazione.
occupancyRateUnit	string	Unità di misura del tasso di occupazione riportata dalla proprietà occupancyRate (es. "num").

Tabella 5: Struttura AggregatedTrafficMeasure

Attributo	Tipo	Descrizione
classId	string	Identificativo alfanumerico della classe veicolare interessata dalla misura di traffico (es. "CAR").
count	integer	Valore numerico intero, corrispondente al numero di veicoli rilevati per la specifica classe veicolare.
countUnit	string	Unità di misura del conteggio riportato dalla proprietà count (es. "num").
speed	float	Valore numerico decimale, corrispondente alla velocità media dei veicoli rilevati nel periodo di aggregazione per la specifica classe veicolare.



speedUnit	string	Unità di misura della velocità media riportata dalla proprietà speed (es. "Km/h").
occupancyRate	float	Valore numerico decimale, corrispondente al tasso di occupazione determinato dai veicoli rilevati nel periodo di aggregazione per la specifica classe veicolare.
occupancyRateUnit	string	Unità di misura del tasso di occupazione riportata dalla proprietà occupancyRate (es. "%").

Tabella 6: Struttura ClassifiedTrafficMeasures

Valore	Descrizione
NOT_CLASSIFIED	Non classificato
MOTORCYCLE	Motociclo, Ciclomotore
CAR	Autoveicolo
VAN	Furgone
BUS	Autobus
COACH	Corriera, Pullman
TRUCK	Autocarro, Camion
ARTICULATED_TRUCK	Autoarticolato, Autotreno
SEMI_TRUCK	Semirimorchio
PEDESTRIAN	Pedone

Tabella 7: Enumerazione per classe veicolare

Oltre alla struttura JSON delle misure di traffico corrispondenti ai criteri di interrogazione indicati, il metodo restituisce una sezione di informazioni utili alle successive interrogazioni del metodo stesso, contenenti link per la gestione della paginazione dei dati restituiti:

- **first**: URL di invocazione del metodo automaticamente impostata per richiedere l'acquisizione della prima pagina dei dati.
- **prev**: URL di invocazione del metodo automaticamente impostata per richiedere l'acquisizione della precedente pagina dei dati (NULL se la precedente interrogazione del metodo riguardava la prima pagina di dati).
- **next**: URL di invocazione del metodo automaticamente impostata per richiedere l'acquisizione della successiva pagina dei dati (NULL se la precedente interrogazione del metodo riguardava l'ultima pagina di dati).
- **last**: URL di invocazione del metodo automaticamente impostata per richiedere l'acquisizione dell'ultima pagina dei dati.

4.2.3 Esempio di invocazione e risposta

Metodo HTTP	GET
URL	<a href="https://<HostNameOrIPAndPort>/<pathTranneNomeMetodo>/trafficData">https://<HostNameOrIPAndPort>/<pathTranneNomeMetodo>/trafficData
Header	Authorization: Auth 2.0 Bearer Token Content-Type: application/json
Body	



Response	<pre>{ "trafficData": [{ "aggregatedMeasure": { "count": 2, "countUnit": "num", "speed": 63.0, "speedUnit": "Km/h", "occupancyRateUnit": "%", "occupancyRate": 0.2080 }, "classifiedMeasures": [{ "classId": "CAR", "count": 2, "countUnit": "num", "speed": 63.0, "speedUnit": "Km/h", "occupancyRate": 0.2080, "occupancyRateUnit": "%" }] }, { "timestamp": "2023-07-11T09:05:00.000+00:00", "period": 300, "device": 2, "description": "Fake sensor", "lane": 1, "transmissionTime": "2023-07-11T15:19:07.139+00:00" }], { "aggregatedMeasure": { "count": 2, "countUnit": "num", "speed": 72.0, "speedUnit": "Km/h", "occupancyRateUnit": "%", "occupancyRate": 0.2786 }, "classifiedMeasures": [{ "classId": "CAR", "count": 2, "countUnit": "num", "speed": 72.0, "speedUnit": "Km/h", "occupancyRate": 0.2786, "occupancyRateUnit": "%" }] }, { "timestamp": "2023-07-11T09:15:00.000+00:00", "period": 300, "device": 2, "description": " Fake sensor", "lane": 1, "transmissionTime": "2023-07-11T15:19:07.562+00:00" } }</pre>
----------	---



	<pre>"aggregatedMeasure": { "count": 2, "countUnit": "num", "speed": 27.0, "speedUnit": "Km/h", "occupancyRateUnit": "%", "occupancyRate": 0.412 }, "classifiedMeasures": [{ "classId": "CAR", "count": 2, "countUnit": "num", "speed": 27.0, "speedUnit": "Km/h", "occupancyRate": 0.412, "occupancyRateUnit": "%" }], "timestamp": "2023-07-11T09:25:00.000+00:00", "period": 300, "device": 2, "description": " Fake sensor ", "lane": 1, "transmissionTime": "2023-07-11T15:19:07.997+00:00" }], "links": { "first": "http://localhost:9500/traffic-data-services/trafficData?fromDate=2023-07-11 08:00&toDate=2023-07-11 20:00&page=1", "next": null, "prev": null, "last": "http://localhost:9500/traffic-data-services/trafficData?fromDate=2023-07-11 08:00&toDate=2023-07-11 20:00&page=1" } }</pre>
--	--

Tabella 8: Esempio di invocazione e risposta del metodo trafficData