

API REST Gestione dispositivi Personal Safety

API per l'utilizzo e la gestione dei dispositivi Personal Safety.

TESTO ESERCIZIO:

Al tempo del coronavirus, si presenta il problema del distanziamento sociale. Recentemente la proposta del Ministero dell'innovazione e della Regione Lombardia è quella di installare un'app sul cellulare delle persone sullo stile di quanto fatto in Corea ([clicca qui](#) o [qui](#))

Il problema più grosso in questa situazione è quello della protezione dei dati e della centralizzazione della memorizzazione che da una parte semplifica la gestione, ma dall'altra introduce concreti problemi di sicurezza degli accessi (si veda solo quello che è successo ieri al sito dell'INPS) e del controllo centralizzato dei cittadini

Proporre l'analisi della gestione di un sistema alternativo di gestione della posizione delle persone da attuarsi solo nei luoghi affollati (per esempio supermercati, ristoranti, cinema, ...) basato su una scheda che viene data ad ogni persona all'ingresso e restituita all'uscita, che collegata al proprio smartphone faccia partire un cicalino/suoneria se la distanza da un'altra scheda scende sotto il fatidico 1,5 metri.

La scheda è in grado di determinare la distanza di un'altra scheda qualsiasi e di identificare le sue coordinate all'interno del supermercato.

Le schede vengono consegnate e restituite presso il punto di controllo posizionato vicino alle casse. La scheda viene abilitata al momento della consegna segnalando al sistema la consegna con la lettura di un QR code e disabilitata alla fine.

Nel punto di controllo è possibile monitorare in tempo reale la situazione relativa alla posizione dei clienti in modo da identificare le zone/corsie in cui si verificano concentrazioni troppo alte di clienti.

Il sistema memorizza ad intervalli fissi la situazione del supermercato, in modo da avere una sequenza di istantanee che permettano a posteriori di effettuare analisi statistiche e ottimizzazioni dei percorsi.

Version 1.0

Autore: Orsenigo Giacomo

Paths

/app/me/devices/

GET /app/me/devices/{idScheda}

Summary

Informazione della scheda

Description

Consente di avere l'elenco dei clienti che hanno utilizzato una determinata scheda

Assicurarsi che i valori relativi al content-type siano conformi a quanto previsto:

- produces: - application/xml

Parameters

Name	Located in	Description	Required	Schema
idScheda	Path	Codice identificativo della scheda	yes	integer

Responses

Code	Description	Schema
200	Elenco delle persone	<pre><xs:element name="clienti"> <xs:complexType> <xs:sequence> <xs:element name="cliente" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="id" type="xs:integer" /> <xs:element name="nome" type="xs:string" /> <xs:element name="cognome" type="xs:string" /> <xs:element name="dataOraInizio" type="xs:dateTime" /> <xs:element name="dataOraFine" type="xs:dateTime" /> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element></pre>
403	Non autorizzato alla richiesta	
404	Scheda non trovata	
default	Errore non previsto	

Query:

```
SELECT Cliente.id, nome, cognome, dataOraInizio, dataOraFine
FROM Cliente
      INNER JOIN Utilizza ON (Cliente.id = Utilizza.idCliente)
WHERE Utilizza.idScheda = [idScheda]
```

/app/me/contatti/

POST /app/me/contatti/

Summary

Memorizza un contatto avvenuto tra due persone

Description

Consente di aggiungere un contatto avvenuto tra due persone, nel caso si fossero trovate troppo vicine

Assicurarsi che i valori relativi al content-type siano conformi a quanto previsto:

- produces: - application/xml
- consumes: - application/xml

Parameters

Name	Located in	Description	Required	Schema
contatto	body	Informazioni sul contatto da inserire	yes	<pre><xs:element name="contatto"> <xs:complexType> <xs:sequence> <xs:element name="idScheda" type="xs:int" /> <xs:element name="idEstraneo" type="xs:int" /> <xs:element name="dataOra" type="xs:dateTime" /> <xs:element name="latitudine" type="xs:decimal" /> <xs:element name="longitudine" type="xs:decimal" /> </xs:sequence> </xs:complexType> </xs:element></pre>

Responses

Code	Description	Schema
200	Contatto aggiunto correttamente	<pre><xs:element name="idContatto" type="xs:integer" /></pre>
403	Non autorizzato alla richiesta	
406	Parametri di input non validi	
415	Il valore del content-type non è quello atteso	
default	Errore non previsto	

Query:

```
INSERT INTO Collisione (idScheda, idEstraneo, dataOra, latitudine, longitudine)
VALUES ([idScheda], [idEstraneo], [dataOra], [latitudine], [longitudine]);

SELECT LAST_INSERT_ID();
```

GET /app/me/contatti/

Summary

Informazione sui contatti avvenuti tra i clienti

Description

Consente di avere l'elenco di tutti i contatti che sono avvenuti tra i clienti

Assicurarsi che i valori relativi al content-type siano conformi a quanto previsto:

- produces: - application/xml

Responses

Code	Description	Schema
200	Elenco dei contatti	<pre><xs:element name="contatti"> <xs:complexType> <xs:sequence> <xs:element name="contatto" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="cliente1"> <xs:complexType> <xs:sequence> <xs:element name="id" type="xs:int"/> <xs:element name="nome" type="xs:string" /> <xs:element name="cognome" type="xs:string" /> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="cliente2"> <xs:complexType> <xs:sequence> <xs:element name="id" type="xs:int"/> <xs:element name="nome" type="xs:string" /> <xs:element name="cognome" type="xs:string" /> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="dataOra" type="xs:dateTime" /> <xs:element name="latitudine" type="xs:integer" /> <xs:element name="longitudine" type="xs:integer" /> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element></pre>
403	Non autorizzato alla richiesta	
default	Errore non previsto	

Query:

```
SELECT cliente1.id, cliente1.nome, cliente1.cognome,
       cliente2.id, cliente2.nome, cliente2.cognome, collisione.*
FROM collisione
  INNER JOIN utilizza AS utilizza1 ON (utilizza1.idScheda = collisione.idScheda)
  INNER JOIN utilizza AS utilizza2 ON (utilizza2.idScheda = collisione.idEstraneo)
  INNER JOIN cliente AS cliente1 ON (utilizza1.idCliente = cliente1.id)
  INNER JOIN cliente AS cliente2 ON (utilizza2.idCliente = cliente2.id)
WHERE (collisione.dataOra > utilizza1.dataOraInizio AND
       (collisione.dataOra < utilizza1.dataOraFine OR utilizza1.dataOraFine IS NULL))
AND (collisione.dataOra > utilizza2.dataOraInizio AND
     (collisione.dataOra < utilizza2.dataOraFine OR utilizza2.dataOraFine IS NULL))
```