percorso.Cerca(string token, string indirizzo_partenza, string indirizzo_arrivo, dict opzioni, string orario, string lang)

Versione corrente: 2

Versioni deprecate: 1

Descrizione

Questo metodo esegue il geocoding degli indirizzi di partenza e destinazione, e il calcolo di un percorso; restituisce il risultato sia in formato testuale che come mappa.

Formati input/output

Input

- `string token`: Token di autenticazione dell'utente
- `string indirizzo_partenza`: Punto di partenza: può essere un indirizzo, numero civico ed eventuale città di partenza (default: Roma); oppure una palina (es.: fermata:70100); oppure le coordinate WGS84 di un punto (es.: punto:(41.90,12.47))
- `string indirizzo_arrivo`: Punto di arrivo
- `dictionary opzioni`: Opzioni per il calcolo del percorso. Attualmente sono disponibili le seguenti opzioni:
  - `int mezzo`: Mezzo di trasporto, tra: `0` (mezzo privato), `1` (mezzo pubblico), `2` (park and ride), `3` (bike and ride), `4` (car sharing)
  - `int piedi`: Propensione spostamenti a piedi, tra: `0` (bassa), `1` (media), `2` (alta)
  - `bool bus`: Inclusione di linee bus nel percorso calcolato
  - `bool metro`: Inclusione di linee di metropolitana nel percorso calcolato
  - `bool ferro`: Inclusione di linee di ferrovie urbane (Atac e Trenitalia) nel percorso calcolato
  - `bool carpooling`: Inclusione di tratti in car pooling nel percorso calcolato (opzionale, solo per utenti abilitati). Valori possibili: `0` (car pooling disabilitato), `1` (offerta di passaggio), `2` (ricerca di passaggio)
  - `float max_distanza_bici`: Massima distanza percorribile con la bicicletta
  - `list linee_escluse`: Linee o luoghi da escludere dalla ricerca del percorso; è un dizionario, le cui chiavi sono gli identificatori delle linee o dei luoghi esclusi, mentre i valori sono i nomi di tali linee o luoghi (arbitrariamente scelti). Ad esempio: `{'MEA': 'Metro A', '542': '542', 'MEB': 'Metro B e B1'}`. Se non è definito, il sistema carica le linee e i luoghi esclusi di default dal database.
  - `int quando`: Istante di ricerca del percorso. Può essere `0` (adesso), `1` (tra 5 minuti), `2` (partenza nell'orario passato tramite il parametro `orario`), `3` (arrivo nell'orario passato tramite parametro). Opzionale, valore di default `2`
  - `list tipi_ris`: Lista di identificatori di tipi di risorse da cercare lungo il percorso (opzionale)
  - `list ztl`: Lista degli identificatori delle ZTL per le quali l'utente possiede il permesso di accesso (opzionale)
  - `bool parcheggi_scambio, bool parcheggi_autorimesse`: Se `mezzo==2` (park and ride), i tipi di parcheggio presso cui può avvenire il cambio modale. Parametri opzionali, di default solo i parcheggi di scambio sono attivi
- `string orario`: Data e ora di inizio dello spostamento, in formato `YYYY-MM-DD HH:MM:SS`
- `string lang`: Codice della lingua in cui si vogliono esprimere le indicazioni

Output

Il servizio restituisce un dizionario i cui elementi dipendono dalla correttezza o meno del geocoding degli indirizzi.

Se il geocoding non ha avuto successo, il servizio restituisce uno o entrambi gli elementi `errore-partenza`, `errore-arrivo`. Essi sono, a loto volta, dizionari con i seguenti elementi:

- `string stato`: Error (indirizzo non trovato) o Ambiguous (molti indirizzi trovati)

- `dict mappa`: Descrizione degli oggetti (marker e polilinee) da far comparire sulla mappa; il formato sarà descritto nella sezione successiva
- `list indicazioni`: Indicazioni testuali del percorso calcolato. Le indicazioni sono un'alternanza di indicazioni relative a nodi (punti del percorso) e tratti. Ogni indicazione è un dizionario con un unico elemento: `nodo`, se l'indicazione è relativa al nodo, o `tratto`, se l'indicazione è relativa a un tratto. Gli elementi sono, a loro volta, dizionari con la seguente struttura:
    - `dict nodo`:
        - `int numero`: indice del nodo corrente. Il primo nodo ha indice 0.
        - `datetime t`: istante di tempo in cui si prevede che l'utente si troverà presso il nodo corrente
        - `string tipo`: tipo del nodo, tra `F` (Fermata), `I` (Indirizzo), `L` (Luogo)
        - `string id`: se il nodo è di tipo `F` o `L`, identificatore della fermata (id_palina) o del luogo
        - `string nome`: nome del nodo, ossia nome della fermata, oppure indirizzo
        - `dict punto`: coordinate del nodo in formato WGS84, se disponibili; stringa vuota, altrimenti. E' un dizionario con gli elementi `x` e `y`
    - `dict tratto`:
        - `int numero`: indice del tratto corrente. Il primo tratto ha indice 0.
        - `string mezzo`: mezzo usato per percorrere il tratto, tra `P` (Piedi), `I` (Tratto a piedi all'interno di un nodo di scambio), `B` (Autobus o tram), `M` (Metropolitana), `T` (Treno), `CP` (Car pooling), `A` (Automobile privata), `CS` (Car sharing)
        - `string linea`: nome della linea
        - `string dest`: capolinea di destinazione
        - `string id`: identificatore del tratto corrente
        - `string tipo_attesa`: metodologia che ha usato l'algoritmo per valutare il tempo di attesa del mezzo, tra `P` (tempo previsto tramite tracking del veicolo in arrivo) ("In arrivo fra/dopo..."), `S` (tempo stimato attraverso statistiche o frequenze di passaggi) ("Attesa prevista: ..."), `O` (tempo calcolato in base all'orario ufficiale dei passaggi) ("Orario: ..."), `E` (istante esatto di passaggio, es. in seguito a un appuntamento)
        - `string tempo_attesa`: tempo di attesa formattato, se tipo_attesa vale `P` o `S`; oppure orario di passaggio del mezzo, se tipo_attesa vale `O` o `E`
        - `string info_tratto`: informazioni sintetiche sul tratto di percorso
        - `float dist`: lunghezza del tratto di percorso, in metri
        - `string info_tratto_exp`: ulteriori informazioni sul tratto di percorso.
- `dict stat`: statistiche globali sul percorso:
    - `float distanza_piedi`: distanza percorsa a piedi, in metri
    - `string distanza_piedi_format`: distanza percorsa a piedi, arrotondata e formattata
    - `float distanza_totale`: lunghezza totale dello spostamento, in metri
    - `distanza_totale_format`: lunghezza totale dello spostamento, arrotondata e formattata
    - `tempo_totale`: durata dello spostamento, in secondi
    - `tempo_totale_format`: durata dello spostamento, arrotondata e formattata

Mappe

Quando occorre disegnare una mappa, un metodo restituisce un dizionario con la descrizione degli oggetti che il client dovrà disegnare come overlay. Gli oggetti attualmente sono di due tipi: marker e polilinee.

La descrizione è fornita attraverso un dizionario, avente due chiavi:

- `list markers`: lista dei marker
- `list polylines`: lista delle polilinee

Ogni marker è rappresentato attraverso un dizionario con i seguenti elementi:

- `list point`: [lat, long] - posizione del marker
- `string icon`: path della risorsa dell'icona (all'interno del namespace http://www.muovi.roma.it/)
- `list iconSize`: [width, height] - dimensioni del marker
- `list anchor`: [anchor_x, anchor_y] - pixel dell'icona che va ancorato al punto `point` sulla mappa
- `string infobox`: testo che deve apparire nel fumetto qualora l'utente clicchi sul marker
- `string name`: se diverso da None, il testo del fumetto dovrà essere richiesto invocando un apposito metodo [TODO: METODO DA DOCUMENTARE]. In tal caso, `name` rappresenta l'identificatore del marker da passare al metodo.
- `bool open`: True se e solo se il fumetto deve essere aperto al caricamento del marker
- `string label`: etichetta del marker, oppure None
- `string drop_callback`: eventuale descrizione dell'azione da compiere se il marker viene trascinato (es.: usa le coordinate del marker come punto di inizio/fine di un nuovo cerca percorso). Se è una stringa vuota, il marker non è trascinabile [TODO: PRECISARE QUALI AZIONI SONO DISPONIBILI]

- `string color`: Colore della polilinea in formato web (#RRGGBB)
- `float opacity`: opacità della polilinea, tra 0 (trasparente) e 1 (opaca)
- `list points`: punti che costituiscono la polilinea. Ogni punto, a sua volta, è rappresentato tramite una lista [lat, lon] che esprime le sue coordinate
- `float thickness`: spessore della polilinea
- `int zIndex`: intero che esprime l'ordine con cui disegnare le polilinee. Polilinee aventi zIndex più basso sono disegnate sotto quelle aventi zIndex più alto

Eventuali altri elementi del dizionario sono da ignorare. st

Updated 2015-02-09