

## PROGETTO TURISMO - TIROCINIO INTERNO



Matteo D'Eramo

MATRICOLA: 290480

Lo scopo di questa semplice applicazione riguarda la possibilità di poter calcolare le distanze e le durate in minuti tra coppie di un insieme di “poi”, ovvero località situate a Perugia e dintorni e presenti nel database del progetto turismo, utilizzando le coordinate di ogni località e poi inserendo i dati ottenuti nella tabella “distanze\_poi”.

(struttura di distanze\_poi)

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito	Commenti	Extra	Azione
<input type="checkbox"/> 1	idpoi_da	int(11)			No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 2	idpoi_a	int(11)			No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 3	idmezzo	int(11)			No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 4	distanza_minuti	int(11)			No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 5	distanza_metri	int(11)			No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 6	data_import	datetime			No	CURRENT_TIMESTAMP			Modifica Elimina Più

Per svolgere questa operazione è stato utilizzato:

- Il linguaggio di scripting **PHP**, con l’ausilio di codice **HTML** e, in minima parte, anche di codice **javascript**;
- **Open street map routing service**, un server offerto da OpenStreetMap per cui è stata creata un’istanza locale in una macchina virtuale linux, in modo da utilizzarne appieno le funzionalità;
- L’API di OSRM relativa al “table service”, che consente di calcolare una matrice di distanze a partire da un set di coordinate; la richiesta effettuata è di tipo “GET”.

(documentazione OSRM)

#### Table service

Computes the duration of the fastest route between all pairs of supplied coordinates. Returns the durations or distances or both between the coordinate pairs. Note that the distances are not the shortest distance between two coordinates, but rather the distances of the fastest routes. Duration is in seconds and distances is in meters.

#### Options

In addition to the [general options](#) the following options are supported for this service:

Option	Values	Description
sources	{index};{index}[]; {index} ...] or all (default)	Use location with given index as source.
destinations	{index};{index}[]; {index} ...] or all (default)	Use location with given index as destination.
annotations	duration (default), ... ..	Return the requested table or tables in response.

```
GET /table/v1/{profile}/{coordinates}?{sources}={elem}...];&{destinations}={elem}...]&annotations={duration|distance|duration,distance}
```

#### Example Request

```
# Returns a 3x3 duration matrix:
curl 'http://router.project-osrm.org/table/v1/driving/13.388860,52.517037;13.397634,52.529407;13.428555,52.523219'

# Returns a 1x3 duration matrix
curl 'http://router.project-osrm.org/table/v1/driving/13.388860,52.517037;13.397634,52.529407;13.428555,52.523219?sources=0'

# Returns a asymmetric 3x2 duration matrix with from the polyline encoded location s "qldcB~-dpXkHHz":
curl 'http://router.project-osrm.org/table/v1/driving/polyline(egs_Iq_aqAppHzbHulFzeMe"EuVKnCg1A)?sources=0;1;3&destinations=2;4'

# Returns a 3x3 duration matrix:
curl 'http://router.project-osrm.org/table/v1/driving/13.388860,52.517037;13.397634,52.529407;13.428555,52.523219?annotations=duration'
```

## PREMESSE

Nella fase preliminare di questo tirocinio è stata molto importante, oltre allo studio dei linguaggi utilizzati, anche la valutazione dell'effettiva fattibilità del progetto: posto come punto di partenza l'utilizzo di apposite API per facilitare il calcolo, sono state valutate anche le API di Google e MapQuest; queste due piste sono state successivamente scartate perché entrambe le case produttrici imponevano gratuitamente solo un utilizzo limitato dei propri servizi in base ad un certo numero di transazioni effettuate, le quali non sarebbero mai state sufficienti per il grande numero di operazioni richieste.

L'unica alternativa rappresentava OSRM, il quale però per poter essere utilizzato, con tutte le sue funzionalità (e non nella versione demo utilizzabile da <http://router.project-osrm.org>), necessitava di essere installato localmente.

Per andare nello specifico, l'API può essere utilizzata tramite una richiesta GET (e quindi specificando i parametri da passare direttamente nell'url) e restituisce una risposta in formato **json**, in cui sono contenuti anche i dati d'interesse; per estrarre i valori desiderati è necessario convertire il file json in array, e poi individuare la corretta locazione dei dati d'interesse.

(esempio di richiesta e risposta da <http://router.project-osrm.org>)

```
include 'functions.php';
curlSES=curl_init();
url="http://router.project-osrm.org/table/v1/car/12.33607568786624,43.09845527677389;12.38962471877130,43.11750966323680";

curl_setopt($curlSES,CURLOPT_URL,$url);
curl_setopt($curlSES,CURLOPT_RETURNTRANSFER,true);
curl_setopt($curlSES,CURLOPT_HEADER, false);
$result=curl_exec($curlSES);

//il risultato json viene convertito in array
echo $result;
curl_close($curlSES);
```

```
{
  "durations":
  [[0,1139.6],[1076.3,0]],
  "destinations": [{"hint": "2oLEg92CxIMAAAAAFQAAAAAADfAQAAAAAALbuzkAAAAA_nwSqWAAAAVAAAAAAN8BAABypwAA7ki8ANWkkQLMO7wAV6GRagAAPwHGscpf",
    "distance": 291.2703721281812,
    "name": "",
    "location": [12.339438, 43.099349]},
    {"hint": "vuphgI3qYYAAAAALAAAAJkBAABIAAAAAAQAUAJUKjf7VD7I3BQgAAAAaAAAAmQEAGUAAABypwAA7ki8ANWkkQLMO7wAV6GRagAAPwHGscpf",
    "distance": 57.32384678277592, "name": "Via Fra' Andrea da Perugia",
    "location": [12.389474, 43.118014]}],
  "sources": [{"hint": "2oLEg92CxIMAAAAAFQAAAAAADfAQAAAAAALbuzkAAAAA_nwSqWAAAAVAAAAAAN8BAABypwAA7ki8ANWkkQLMO7wAV6GRagAAPwHGscpf",
    "distance": 291.2703721281812, "name": "", "location": [12.339438, 43.099349]},
    {"hint": "vuphgI3qYYAAAAALAAAAJkBAABIAAAAAAQAUAJUKjf7VD7I3BQgAAAAaAAAAmQEAGUAAABypwAA7ki8ANWkkQLMO7wAV6GRagAAPwHGscpf",
    "distance": 57.32384678277592, "name": "Via Fra' Andrea da Perugia", "location": [12.389474, 43.118014]}],
  "code": "Ok"}
```

## PROCEDIMENTO

Inizialmente viene presentata un'interfaccia grafica in cui è brevemente spiegato il procedimento e che presenta un tasto “trasferisci” per avviare la procedura di inserimento e un altro tasto “ripulisci tabella” per svuotare la tabella *distanze\_poi* preventivamente:

### INSERIMENTO

questa semplice interfaccia e' pensata con lo scopo di fornire un semplice input al fine di inserire distanze e durate all'interno della tabella **distanze\_poi** tramite le api di OSRM.

(per maggiori informazioni visita [OSRM](#)).

#### procedura

- composizione di un array di id e di coppie latitudine-longitudine dalla tabella **poi** tramite le coordinate
- utilizzo delle coordinate per fare richieste in coppia al **server OSRM** di tipo "table" ottenendo una risposta in formato **json** che presenta due sottocampi **distances** e **duration** il quale verra' successivamente convertita in array, dacui si potranno ricavare i valori dei sottocampi gia' citati
- dalle mini-matrici 2x2 vengono prelevati i valori di interesse
- i valori vengono immediatamente posti dentro la tabella temporanea **distanze\_poitemp**
- a seconda della scelta fatta nella pagina finale, si possono inserire i dati nella tabella ufficiale oppure, in caso di errori nello svolgimento dell'operazione, si puo' svuotare la tabella temporanea e ricominciare da capo oppure ripartire

trasferisci

ripulisci tabella

Avviando la procedura di trasferimento viene attivata una action sul file “*script.php*”, in cui vengono attuate delle query SELECT in modo da prelevare i valori *id*, *lat* e *lon* dalla tabella *poi*; Questi valori vengono salvati in memoria e ci saranno utili per l’operazione d’inserimento.

Nella pagina corrente viene infine richiamata la funzione `matrix($k,$k1,array $lon_arr,array $lat_arr,array $id_arr,$type,$link,$id mezzo)` che rappresenta il cuore della intera applicazione; la funzione, salvata nel file *functions.php*, fa sì che venga inizializzata una sessione curl e costruito un url passando come parametri le latitudini e longitudini, fatte scorrere ciclicamente tramite due cicli for. Ad ogni ciclo non vengono solamente calcolate distanze e durate, ma anche inserite in una tabella temporanea *distanze\_poitemp*. Per rendere più robusta l’operazione, viene inizializzata una transazione in cui, per ogni 10 tuple inserite, viene creato un SAVEPOINT e, in caso di errore, viene effettuato un ROLLBACK TO SAVEPOINT in modo che vengano cancellate le precedenti operazioni.

Una volta completata l’operazione, verrà aperta un’interfaccia in cui viene mostrata la percentuale di completamento della tabella, in base al numero di elementi inseriti e al numero di operazioni che teoricamente devono essere fatte; a questo punto si ha la possibilità di inserire effettivamente i dati in *distanze\_poi* (e ripulendo *distanze\_poitemp*) oppure ripetere l’operazione (nel caso in cui la percentuale di completamento non sia 100%) e effettuare di nuovo l’inserimento, ignorando le tuple già inserite attraverso l’opzione ON DUPLICATE KEY UPDATE.

la tabella temporanea e' stata riempita non completamente (NAN%)-- errore nella procedura.  
inserire tutti i valori ottenuti?

no e correggi

si

Una volta risposto “si” i dati verranno trasferiti in *distanze\_poi* e l’operazione sarà completata.

# CARICAMENTO COMPLETATO!

torna alla pagina iniziale

## **FONTI**

<https://stackoverflow.com/>

<http://project-osrm.org/>

<https://www.html.it/guide/guida-php-di-base/>

<https://www.w3schools.com/php/>