# 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".

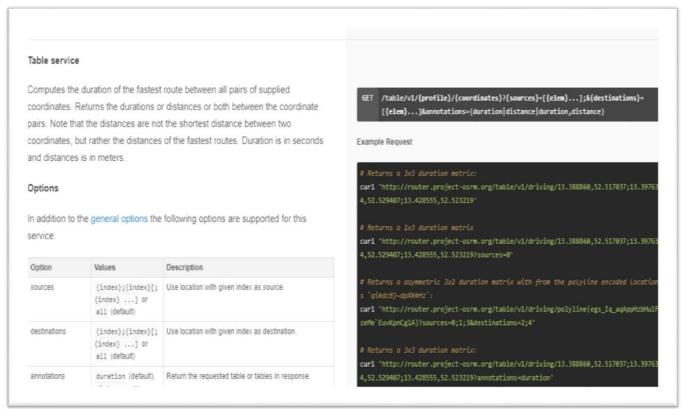
## (struttura di distanze\_poi)



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)



#### **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 <a href="http://router.project-osrm.org">http://router.project-osrm.org</a>), 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 <a href="http://router.project-osrm.org">http://router.project-osrm.org</a>)

```
nclude '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);
```

### **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:



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 <a href="matrix">matrix</a>(\$k,\$k1,array \$lon arr,array \$lon arr,array \$lon arr,stype,\$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/