



FITPLACE

Grossi Manuel
Grifoni Christian
Cardinali Marco



Introduzione

L'applicazione permette, previa autenticazione dell'utente, di localizzarsi e cercare i punti di interesse per quanto riguarda il fitness in un range di area prestabilito. Una volta localizzata la zona nella quale ci troviamo, possiamo visualizzare sulla mappa le palestre, i parchi, le spa, ed infine gli stadi.

Inoltre è disponibile una sezione Contattaci, con le varie informazioni riguardanti gli sviluppatori, una chat ed infine un form che ci da la possibilità di calcolare, una volta inserite le informazioni richieste (età, altezza, peso e sesso) il metabolismo basale, la massa magra e la massa grassa.

Avvio

Per installare le dipendenze eseguire 'npm install', verranno lette dal file `*package.json*` e installate.

Per avviare il server eseguire il comando 'node server.js' sul terminale.

WebSocket (porta 8080) deve essere in esecuzione su `_localhost_`.

Collegandosi a <http://localhost:8888/api-docs> è possibile visualizzare la documentazione

REST API

Maps JavaScript API: consente di personalizzare le mappe con i propri contenuti e immagini per la visualizzazione su pagine web e dispositivi mobili.

Geocode: tramite una get, passando l'indirizzo del luogo richiesto ci ritorna le sue informazioni dalle quali verranno prese latitudine e longitudine.

NearbySearch: si effettua una richiesta passando latitudine, longitudine, il tipo di ricerca che si sta effettuando ed il raggio d'interesse per trovare un insieme di luoghi di quella tipologia. La ricerca può essere fatta su: palestre, spa, parchi, stadi da calcio.

OAuth

Con OAuth abbiamo la possibilità di chiamare in modo sicuro ed autorizzato API messe a disposizione da un servizio web.

Google Login: Implementato con Passport, richiede l'autenticazione e restituisce le informazioni base del proprio profilo (Id, nome, email).

Funzionalità

- WebSocket: In 'server.js' viene inizializzato il server sulla porta 8080, e si connette al proprio server. Prende il messaggio facendo il parsing JSON e lo inoltra, tramite la funzione 'broadcast', a tutti i client connessi. In 'chat.ejs' viene gestito l'invio del messaggio alla WebSocket tramite la funzione 'send' e lo visualizza tramite la funzione 'onmessage'.