

OSMAND

L'App dà la possibilità di impostare un percorso personalizzato, può essere utilizzata offline e presenta un sintetizzatore vocale

IL CODICE <https://github.com/osmandapp/Osmand>

L'App è Open Source e il codice si può trovare su GitHub

OsmAnd rende possibile l'utilizzo di un API utilizzabile per aggiungere mappe, creare audio, navigare tra diversi luoghi e altro, una feature che potrebbe essere utilizzata anche dalla nostra App.

OsmAnd permette anche di costruire un'applicazione a partire dal proprio core, provvede una libreria java e molte funzioni.

Nella licenza dell'app ci sono cose che potrebbero interessarci. OsmAnd usa un codice fatto da terze parti con licenze permissive del genere LGPL, MIT, Apache. Il codice ha un copyright da OsmAnd BV ed è distribuita sotto GPLv2 con un'eccezione per la distribuzione sul play store senza avere il permesso. Ciò dà la possibilità di creare un'applicazione per uso personale o per creare un'applicazione open source con la licenza GPLv2. Se si vuole creare un'applicazione per uso interno bisogna contattare l'azienda alla loro mail di business (business@osmand.net). L'azienda rende anche possibile la modifica del loro routing engine, utilizzato per calcolare la strada più veloce da un punto all'altro; l'algoritmo funziona con questa equazione: $(= \text{distance} / (\text{speed} * \text{priority}) + \text{penalties})$. Esso è suddiviso in 7 sezioni diverse che sono:

- Accesso (1 autorizzato ; - 1 non autorizzato)
- Velocità (KmH)
- Priorità (tra 0 e 1) - un multiplo della velocità e collegato alla velocità massima
- Senso Unico (1, -1, 0) - utilizzato per stabilire come si può accedere ad una strada
- Penalizzazione - utilizzata per definire la penalità quando si cambia il tipo di strada (es: se si passa da autostrada a strada civile)
- Ostacoli - definisce se ci sono ostacoli durante il tragitto
- Tempo di ostacoli - definisce quanto tempo si impiegherebbe per oltrepassare un ostacolo (l'engine troverà una strada più veloce in base a questo)

Per testare l'engine di routing (routing.xml) si può utilizzare [OsmAndMapCreator](#).

TUTTO QUESTO PUÒ ESSERE UTILIZZATO PER CREARE PERCORSI PER PERSONE CON DISABILITÀ O SIMILI.

L'applicazione è stata creata utilizzando un'implementazione simile a OsmDroid <https://github.com/osmdroid/osmdroid>

QUA SOTTO VERRÀ LASCIATA LA TRASCRIZIONE IN INGLESE:

OsmAnd API allows you to control the installed OsmAnd app. Starting development via the API is easy and straightforward and can be embedded in any application, without licensing requirements of the target app. The only drawback is that the user should have OsmAnd installed

OsmAnd API features:

- Adding favorites and markers to the map
- Navigation between locations
- Creating audio, video and photo notes
- Starting and stopping GPX track recording
- Importing GPX tracks into OsmAnd and navigating along them
- Many other features are already present or can be implemented by request

Start your project

You can build your own project in any way you like. Integration with OsmAnd API is done using two types of intents: silent or visible. A silent intent doesn't keep OsmAnd open, whereas a visible intent brings OsmAnd to a specific screen. There are plans to add Android Interprocess Communication in future. Please take a look at source code of the OsmAnd API project.

<https://github.com/osmandapp/osmand-api-demo>

License

Since there is no direct code usage from the core OsmAnd project, the License is different for the OsmAnd API and for the OsmAnd Core project. Most likely application using OsmAnd API will be written from scratch and this application provided as an example won't be used at all. For OsmAnd API the least restrictive license is used, MIT license.

<https://github.com/osmand-api-demo/Osmand/blob/master/LICENSE.md>

OsmAnd Sample

OsmAnd Sample represents a possibility to build an application on top of OsmAnd core. OsmAnd provides a Java library with lots of functions included. Please take a look at [the application](#) and at [the source code](#). OsmAnd Sample is not an API use case and the LICENSE of OsmAnd Sample

is the same as the OsmAnd application itself. It provides a possibility to package and distribute it independently from OsmAnd.

License

The license <https://github.com/osmandapp/Osmand/blob/master/LICENSE> for the OsmAnd application is rather big. There are 2 important things: OsmAnd uses third-party code only with permissive licenses like (LGPL, MIT, Apache) and OsmAnd code itself copyrighted by OsmAnd BV is distributed under GPLv2 license with exception of distributing it on Google Play markets without permission. That provides a possibility to build any application for personal usage or build an opensource application under GPLv2 license (cause OsmAnd is not a platform and all the code built on top of the core must be also licensed as GPL).

If you want to build an application for internal usage, please contact business@osmand.net, in some cases exemption from GPLv2 code copyrighted by OsmAnd BV are provided. For the rest of the could please double check the list from the [LICENSE](#) which consists only of permissive licenses (LGPL, MIT, Apache).

In case of any question, please don't hesitate to contact business@osmand.net. We also will be happy to provide help from experienced OsmAnd developers to build a special purpose app.

Create new rendering style

Creating new rendering style is possible without any change in the program. Style definitions could be difficult if you want to write it from scratch for example [Nautical style](#) or [Default style](#) (15K lines of code). It is advised to create a dependent style so you could change start changing style by small pieces. For example [UniRS](#) style contains mostly color redefinition and icon appearance.

To test rendering style you could use [OsmAndMapCreator](#). In case you would like to use in the OsmAnd on device. You need to put the xml definition of the style into folder on sdcard ('rendering') next to other rendering styles.

All rights on the style is created belongs to the author. Though these rights could not be applied to the OsmAnd binaries and .

Customize routing engine

In some cases you might want to extend or adjust routing experience of OsmAnd, so you could add extra parameters, add extra barriers or change some penalties. Please take a look at [routing.xml](#).

OsmAnd routing is using 2 directional A* algorithm based on fastest time ($= \text{distance} / (\text{speed} * \text{priority}) + \text{penalties}$). It is a pretty easy format. It is divided into 7 sections (evaluation groups):

- access (1 allow, -1 not allow)
- speed (number that presents speed in kmh)
- priority (number between 0 and 1) - a multiplier to the speed and it is still bound to maxSpeed, so $\text{minimum}(\text{speed} * \text{multiplier}, \text{maxSpeed})$ will be used for A*
- oneway (1, -1 or 0) - used to clarify access based on direction of the movement
- penalty_transition (penalty in seconds) - used to define penalty when user goes from high class road to low class road. For example, motorway - 10, trunk - 15, if user goes from trunk to motorway penalty will be 5 seconds ($= 15 - 10$). This penalty will be used by A* algorithm. There is no penalty if user goes from trunk to motorway and there is no penalty if route continues from motorway to motorway.
- obstacle (penalty in seconds) - defines penalty which is to the routing time
- obstacle_time (penalty in seconds) - defines penalty which is displayed to the user but it is not considered by routing engine i.e. obstacle_time - 2 hours, obstacle - 1 minute, on the shortest route engine will find a route using 1 minute obstacle but user will see that route time is 2 hours.

In order to test routing.xml you could use [OsmAndMapCreator](#).

Projects using their own implementation similar to osmdroid:

- OsmAnd