Longrun Planner and Tracker

Installation

To install this software: `$ git clone https://git.code.tecnalia.com/inakio/longrun.git`
To run it: `$ cd longrun ~/longrun$ docker-compose up -d`

Defining a new Map

The map to be used by the routing service OSRM should be of type OSM. This type of collaboratively generated maps cann be obtained in a number of different ways. Extracts for different countries can be obtained from geofabrik which can be further reduce by means of tools like osmconvert. An example to obtain a smaller rectangular map is provided: `osmconvert spain-latest.osm.pbf -b=-3.504037,42.509484,-1.702279,43.493699 -o=basque.osm`

The result needs to be renamed to `mapa.osm` and copied to the `osrm_server` directory.

Using the Planner

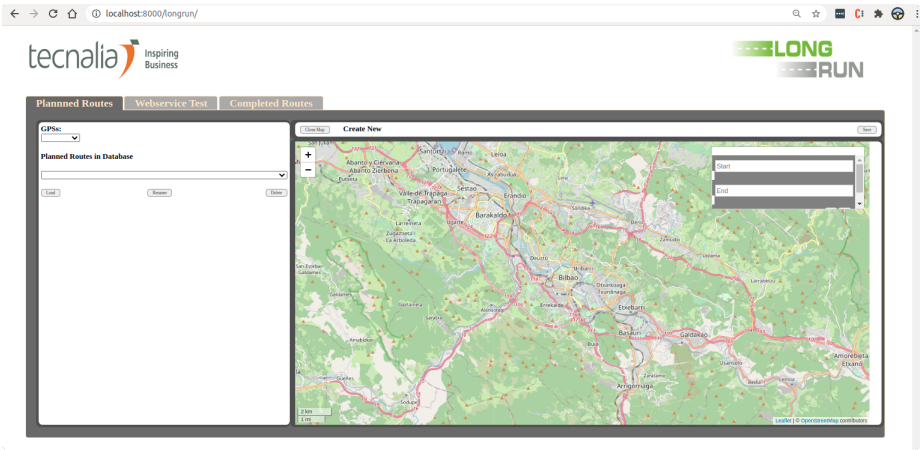

Figure 1: Beginning Screen

Database

In order to store trip planifications and actual map-matched trips there is a MySQL database with the following table structure. Note that some of the fields are not completely used for the time being.

planificaciones Table:

Field	Type	Null	Key	Default	Extra
id_user	int(11)	YES		NULL	
id_plan	int(11)	NO	PRI	NULL	auto_increment
polyline	text	YES		NULL	
nombre	varchar(200)	YES		NULL	
waypoints	text	YES		NULL	

routes Table:

Field	Type	Null	Key	Default	Extra
id_trip	int(11)	NO		NULL	
id_plan	int(11)	YES		NULL	
differences	json	YES		NULL	
timestamp	bigint(20)	YES		NULL	
diffs_polyline	text	YES		NULL	
plan_polyline	text	YES		NULL	
diffIndexes	text	YES		NULL	
planDiffIndexes	text	YES		NULL	

configuration Table:

Field	Type	Null	Key	Default	Extra
type	varchar(5)	NO		NULL	
params	varchar(255)	YES		NULL	

users Table:

Field	Type	Null	Key	Default	Extra
id_user	int(11)	NO		NULL	
nombre	varchar(20)	NO		NULL	
email	varchar(40)	NO		NULL	

points Table:

Field	Type	Null	Key	Default	Extra
id_trip	int(11)	NO		NULL	
timestamp	bigint(20)	YES		NULL	
lat	double	YES		NULL	
lon	double	YES		NULL	
polyline	text	YES		NULL	
mlat	double	YES		NULL	
mlon	double	YES		NULL	
mpolyline	text	YES		NULL	
mtimes	text	YES		NULL	

mapped_pnts:

Field	Type	Null	Key	Default	Extra
id_trip	mediumtext	YES		NULL	
timestamp	bigint(20)	YES		NULL	
lat	double	YES		NULL	
lon	double	YES		NULL	

API Definition

Device Creation

This method is used in order to add a new vehicle to the database. A unique id_user of type string needs to be provided, a plate number is advised to be used.

- **path:** http://longrun:8000/longrun/webresources/changeDB/createDevice
- **type:** PUT
- **params:**

```
{ "id_user": "id_user","name": "name"}
```

- **example:**

```
{ "id_user": "2008ZGZ", "name": "Red Truck" }
```

- **response:**

```
{"code":200,"id_user":"2008ZGZ"}
```

Start Sending Trip

This method needs to be called before sending GPS data. The service responds with a id_trip that needs to be included every time GPS data is sent to the system. A valid id_plan needs to be associated to the trip to be done.

- **path:** http://longrun:8000/longrun/webresources/track/iniTrip
- **type:** PUT
- **params:**

```
{ "id_user": "id_user","id_plan": id_plan, "t":timestamp}
```

- **example:**

```
{"id_user":"10001", "id_plan":9, "t":1626880173335}
```

- **response:**

```
{"code":200,"message":"ok","id_trip":17558}
```

End Sending Trip

This method needs to be called after sending GPS data for a given trip. It closes the current trip, after closure the id_trip cannot be longer be used in order to send data to platform.

- **path:** http://longrun:8000/longrun/webresources/track/endTrip
- **type:** PUT
- **params:**

```
{"id_trip": id_trip, "t":timestamp}
```

- **example:**

```
{"id_trip":17558, "t":1626880173335}
```

- **response:**

```
{"code":200,"message":"ok"}
```

Send GPS Data

Send GPS data in order to know the route to destination even if a deviation from plan is taken.

- **path:** http://longrun:8000/longrun/webresources/track/match
- **type:** PUT
- **params:**

  {"id_trip": id_trip, "t": timestamp, "lat": latitude, "lon": longitude }
- **example:**

  {"id_trip":17558, "t":1416672171716, "lat":43.293304, "lon":-2.863138 }
- **response:**


```
{
                "mlat": 43.293287,
                "mlon": -2.863277,
                "route": {
                    "nodes": [
                        {   "lat": 43.29329,
                            "lon": -2.86328,
                            "elevation": 48,
                            "distance": 0,
                            "prct": 0,
                            "maxspeed": null,
                            "speed": 24.12,
                            "oneway": "yes",
                            "duration": 0,
                            "R": 105.82674336742907,
                            "lanes": "-",
                            "id": 311752043,
                            "bridge": "-",
                            "highway": "residential",
                            "lat_c": 43.29340003094893,
                            "lon_c": -2.8619811874628622
                        },
                        ...,
                        {   "lat": 43.32918,
                            "lon": -2.87189,
                            "elevation": 81,
                            "distance": 34.01784,
                            "prct": 0,
                            "maxspeed": null,
                            "speed": 15.12,
                            "oneway": "-",
                            "duration": 8.2,
                            "R": 375.390923,
                            "lanes": "-",
                            "id": 3794041384,
```

```
                    "bridge": "-",
                    "highway": "service",
                    "lat_c": 43.325907276142715,
                    "lon_c": -2.8730280108392776
                }
            ],
            "code": 200,
            "geometry": "avfgGnfnPVE\\Mf@YLILITUPOFGT]Ta@Re@L]Rc@DKXq@HQLQFDHBH
        },
        "code": 200,
        "message": "{\"code\":\"Ok\",\"waypoints\":[{\"nodes\":[311752044,31175
}
```

User Manual

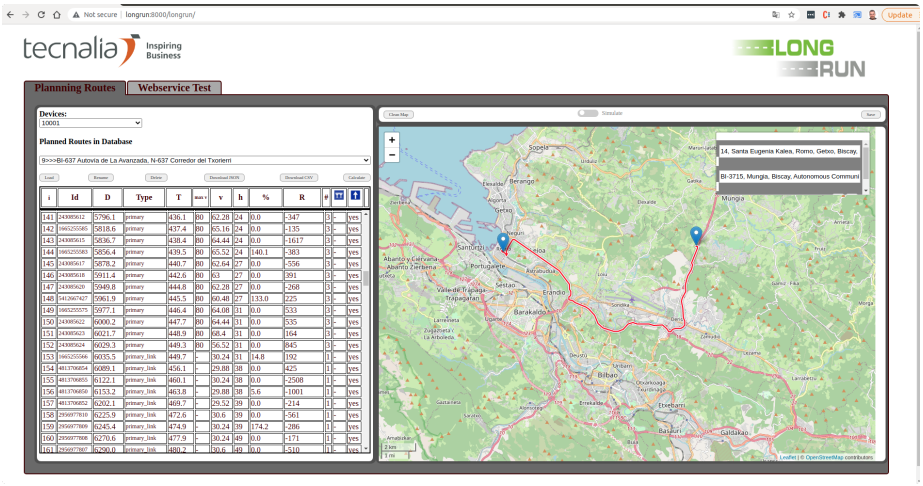

Figure 2: Screen with results

The web application has two different tabs: the main tab (Planning Routes) and the testing tab (Webservice Test). The main tab is divided in the vertical sides: on the left side there are two menus, six buttons and a table and on the right a two buttoms a switch and a map. The menus allows the user the following:

- **Devices:** This menu allows to select one of the devices stored within the database.
- **Planned Routes in Database:** Allows to select the trajectories planned for the choosen device.

The buttoms allows the user to perform several functionalities:

- On the Left:
- **Load:** loads the selected trajectory for the chosen device into the map.
- **Rename:** renames the selected trajectory in the platform database.
- **Delete:** deletes the selected trajectory from the platform database
- **Download JSON:** downloads json file with the information of the trajectory shown in the table below.
- **Download CSV:** downloads csv file with the information of the trajectory shown in the table below.
- **Calculate:** This computes the advanced information and fills the table on the left side.
- On the Right:
- **Clean Map:** To clean the map from painted trajectories.
- **Save:** To save the trajectory into the database.

The table with advanced information has the following columns:

- **i:** index of the point within the trajectory
- **Id:** identification of the OSM node associated with the point.
- **D:** distance along the trajectory
- **T:** time along the trajectory
- **max v:** maximum speed on the point, if it is avaiable.
- **v:** speed provided by the router (OSRM) this speed depends on the maximum speed and the type of the road.
- **h:** elevation of the point
- **%:** percentange of the climb.
- **R:** radious of curvature in meters, the sign indicates the position with respect to the directotion of the trajectory (negative on the right, positive on the left).
- **#:** number of lanes of the road that point (if it is avaiable)
- **bridge:** yes or '-' (if avaiable). This value is used in order to correct the elevation.
- **one way:** yes or '-' (if avaiable)

The switch on top of the map turns on/off the simulation tab:

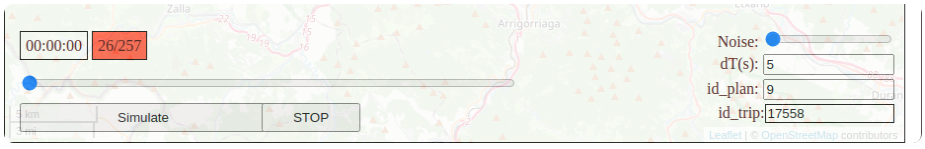

Figure 3: Detail of the Simulation tab

This tab allows the user of the web to simulate a trip with the loaded plan in the map. In order for the platform to perform the simulation the following needs to be chosen:

- **Noise:** the gaussian noise added to the GPS measurement. (Keep to low values for the time being)
- **dT:** the time period between consecutive GPS measurements.
- **id_plan:** the id_plan to associate the trip with.

Once this parameters have been chosen, in order to simulate the trip the only thing to do is to press the "Simulate" buttom. During the simulation the index box is coloured red to know the simulation is being performed. This process can be interrupted in any moment pressing the "STOP" buttom. Also the simulated trip gets and identifcation number associated with it, which value is shown on the right hand side. Finally once the simulation process is finished the result can be visualized moving the slider bar to choose different points in the trayectory. Note that the plannification simulated does not need to agree with the id_plan parameter used within the simulation.