# Mireo Android API

## Android API specification

Document date: September 22, 2021

Mireo d.d.
Buzinski prilaz 32, 10000 Zagreb, HR-Croatia
WEB: www.mireo.hr      mail: info@mireo.hr

# 0.1 Main Page

**Author**

Mireo d.d.

**Version**

3.1.0

## 0.1.1 Introduction

This document describes Mireo Android API.

Communication with Mireo navigation is performed using this API.

## 0.1.2 Installation

Mireo API is distributed as a single jar archive **api.jar**. Client has to add library in its android project. The library itself has no additional dependancies.

Mireo API consists of:

- hr.mireo.arthur.api.EasyAPI - the main entry point for the API.
- hr.mireo.arthur.api.INotificationListener3, hr.mireo.arthur.api.INotificationListener2, hr.mireo.arthur.api.INotificationListener - handler for notifications (location changes, route changes, advices)
- various data types and enumerations: hr.mireo.arthur.api.Advice, hr.mireo.arthur.api.Route, hr.mireo.arthur.api.GeoAddress, hr.mireo.arthur.api.PositionData, hr.mireo.arthur.api.TrafficStatus

## 0.1.3 Example usage

```
// create API instance
EasyAPI mAPI =  new EasyAPI("gm", this);
// get application version
mAPI.getVersionEx((status, version) -> {
    if (status == API.RESULT_OK) {
        Log.d(TAG, "Application version: " + version.appVersion);
        Log.d(TAG, "Build number: " + version.buildNumber);
        Log.d(TAG, "Map version: " + version.mapVersion);
        Log.d(TAG, "Device serial number: " + version.serial);
    }
    else {
        Log.e(TAG, "getVersionEx ERROR: " + EasyAPI.statusMessage(status));
    }
});
// get current address
mAPI.getCurrentAddress((status, address) -> {
    if (status == API.RESULT_OK) {
        Log.d(TAG, "Current address:");
        Log.d(TAG, address.formatted[0]);
        Log.d(TAG, address.formatted[1]);
    }
    else {
        Log.e(TAG, "getCurrentAddress ERROR: " + EasyAPI.statusMessage(status));
    }
});
```

## 0.2 Appendices

### 0.2.1 APPENDIX A – Extended ISO codes

As a rule, it is a regular ISO 3166 code with 2 more digits for region

- Canada
  - 12401 Alberta
  - 12402 British Columbia
  - 12403 Manitoba
  - 12404 New Brunswick
  - 12405 Newfoundland and Labrador
  - 12406 Northwest Territories
  - 12407 Nova Scotia
  - 12408 Nunavut
  - 12409 Ontario
  - 12410 Prince Edward Island
  - 12411 Québec
  - 12412 Saskatchewan
  - 12413 Yukon Territory
  - 12413 Yukon
- USA
  - 84001 Alabama
  - 84002 Alaska
  - 84003 Arizona
  - 84004 Arkansas
  - 84005 California
  - 84006 Colorado
  - 84007 Connecticut
  - 84008 Delaware
  - 84009 District of Columbia
  - 84010 Florida
  - 84011 Georgia
  - 84012 Hawaii
  - 84013 Idaho
  - 84014 Illinois
  - 84015 Indiana
  - 84016 Iowa
  - 84017 Kansas

  - 84018 Kentucky
  - 84019 Louisiana
  - 84020 Maine
  - 84021 Maryland
  - 84022 Massachusetts
  - 84023 Michigan
  - 84024 Minnesota
  - 84025 Mississippi
  - 84026 Missouri
  - 84027 Montana
  - 84028 Nebraska
  - 84029 Nevada
  - 84030 New Hampshire
  - 84031 New Jersey
  - 84032 New Mexico
  - 84033 New York
  - 84034 North Carolina
  - 84035 North Dakota
  - 84036 Ohio
  - 84037 Oklahoma
  - 84038 Oregon
  - 84039 Pennsylvania
  - 84040 Rhode Island
  - 84041 South Carolina
  - 84042 South Dakota
  - 84043 Tennessee
  - 84044 Texas
  - 84080 USA AK
  - 84081 USA HI
  - 84072 USA Middle
  - 84076 USA North East NT
  - 84077 USA South East NT
  - 84071 USA West
  - 84045 Utah
  - 84046 Vermont
  - 84047 Virginia
  - 84048 Washington
  - 84049 West Virginia
  - 84050 Wisconsin
  - 84051 Wyoming

- India
  - 35635 Andaman & Nicobar Islands
  - 35628 Andhra Pradesh
  - 35612 Arunachal Pradesh

  - 35618 Assam
  - 35610 Bihar
  - 35604 Chandigarh Metro
  - 35622 Chhattisgarh
  - 35626 Dadra and Nagar Haveli
  - 35625 Daman and Diu
  - 35607 Delhi NCR
  - 35630 Goa
  - 35624 Gujarat
  - 35606 Haryana
  - 35602 Himachal Pradesh
  - 35601 Jammu & Kashmir
  - 35620 Jharkhand
  - 35629 Karnataka
  - 35632 Kerala
  - 35631 Lakshadweep
  - 35623 Madhya Pradesh
  - 35627 Maharashtra
  - 35614 Manipur
  - 35617 Meghalaya
  - 35615 Mizoram
  - 35613 Nagaland
  - 35621 Odisha
  - 35621 Orissa
  - 35634 Puducherry
  - 35603 Punjab
  - 35608 Rajasthan
  - 35611 Sikkim
  - 35633 Tamil Nadu
  - 35636 Telangana
  - 35616 Tripura
  - 35609 Uttar Pradesh
  - 35605 Uttaranchal
  - 35619 West Bengal

- Russia
  - 64301 Far Eastern Federal District
  - 64302 Siberian Federal District
  - 64303 Ural Federal District
  - 64304 North Caucasian Federal District
  - 64305 Southern Federal District
  - 64306 Central Federal District
  - 64307 Northwestern Federal District
  - 64308 Volga Federal District

### 0.2.2 APPENDIX B – Address Type Codes

- 1 Country Capital
- 2 State Capital
- 10 Generic City
- 11 Metropolis
- 12 Large City
- 13 Town
- 14 MediumTown
- 15 Small Town
- 16 Urban Area
- 17 Village
- 18 City Part
- 19 Remaining City Centers
- 30 Town District
- 80 Square
- 170 Toll Booth
- 171 Toll Trigger
- 180 Car Train Terminal
- 201 Bridge Viaduct
- 202 Aqueduct
- 203 Tunnel
- 300 Border Crossing
- 301 Restorans – American Food
- 302 Restoran - Californian Food
- 303 Restoran - Chinese Food
- 304 Restoran – Continental Food
- 305 Restoran – French Food
- 306 Restoran – German Food
- 307 Restoran – Greek Food,
- 308 Restoran – Indian Food
- 309 Restoran – Italian Food
- 310 Restoran – Japanese Food
- 311 Restoran – Mexican Food
- 312 Restoran – Seafood
- 313 Restoran – Thai Food

- 314 Restoran – Vegetarian Food
- 315 Restoran – Vietnamese Food
- 316 Restoran – Austrian Food
- 317 Restoran – Sothern Barbecue
- 318 Restoran – Belgian Food
- 319 Bistro
- 320 Brewpub
- 321 Restoran - British Isles Food
- 322 Restoran – Dutch Food
- 323 Restoran – East European Food
- 325 Restoran – Grill
- 326 Restoran – Hawaiian and Polynesian Food
- 327 Restoran – Hungarian Food
- 328 Restoran – Indonesian and Malaysian Food
- 329 Restoran – Jewish Kosher Food
- 330 Restoran – Korean Food
- 331 Restoran – Latin American Food
- 332 Restoran – Maltese Food
- 333 Restoran – Middle Eastern Food
- 334 Restoran – Filipino Food
- 335 Restoran – Polish Food
- 336 Restoran – Portuguese Food
- 337 Restoran – Russian Food
- 338 Sandwich Bar
- 339 Restoran – Scandinavian Food
- 340 Restoran – South American Food
- 341 Restoran – Southeast Asian Food
- 342 Restoran – Southwestern Food
- 343 Restoran – Surinamese Food
- 344 Restoran – Spanish Food
- 345 Steak House
- 346 Restoran – Swiss Food
- 347 Restoran – Turkish Food
- 348 Restoran – African Food
- 349 Restoran – Canadian Food
- 350 Restoran – International Food

- 351 Restoran – Bohemian Food
- 352 Restoran – Balkan Food
- 353 Restoran – Finnish Food
- 354 Restoran – Australian Food
- 355 Pizzeria
- 356 Snacks and Beverages
- 357 Restoran – Breakfast
- 358 Restoran – Chicken
- 359 Ice Cream
- 360 Tapas
- 361 Restoran – Irish Food
- 362 Restoran – Caribbean Food
- 363 Restoran – Malaysian Food
- 364 Restoran – Moroccan Food
- 365 Restoran – Fusion
- 366 Restoran – Brazilian Food
- 367 Restoran – Creole Food
- 368 Restoran – Burgers
- 369 Restoran – Creperie
- 370 Restoran – Pastries
- 371 Restoran – Sushi
- 372 Restoran – Vegan Food
- 373 Restoran – Cajun Food
- 374 Restoran – Indonesian Food
- 375 Restoran – Fondue
- 376 Restoran – Argentinean Food
- 377 Restoran – Chilean Food
- 378 Restoran – Azerbaijan Food
- 379 Restoran – Baltic Food
- 380 Restoran – Caucasian Food
- 381 Restoran – Ukrainian Food
- 382 Restoran – Venezuelan Food
- 383 Restoran – Bruneian Food
- 400 Tourist Information
- 401 Tourist Office
- 402 Museum

- 403 Historical Monument
- 404 Castle to Visit
- 405 Natural Attraction
- 406 Tourist Attraction
- 407 RecreationFacility
- 408 Amusement Park
- 409 Castle
- 410 School
- 411 University or College
- 412 Library
- 413 Kindergarten
- 414 Institute
- 415 Campus
- 416 Educational Religion Institute
- 420 Culture Site
- 421 Theatre
- 422 Casino
- 423 Night Club
- 424 Other Entertainment Facilty
- 425 Zoo
- 425 Cinema
- 427 Opera
- 428 Leisure Site
- 429 Gallery
- 430 Business Facility
- 431 Company
- 433 Unspecified Services
- 434 Laundry and Dry Cleaning
- 435 Hair and Beauty Salon
- 436 Attorney
- 440 Exhibition or Conference
- 441 Unspecified Events
- 442 Organization And Societies
- 450 Hotel or Motel
- 451 Hotel
- 452 Motel

- 453 Camping Ground
- 454 Convention Centre
- 459 Other Accomodation
- 460 Generic Restaurant
- 461 Cafe or Pub
- 462 Banquet or Marriage Hall
- 463 Fast Food
- 464 Coffee Shop
- 465 Bar
- 470 Shopping Center
- 471 Grocery Store
- 472 Winery
- 473 Market
- 474 Florist
- 475 Clothing Store
- 476 Home Specialty Store
- 477 Specialty Store
- 478 Sporting Goods Store
- 479 Home Improvement Hardware Store
- 480 Stadium
- 481 Sports Activity
- 482 Ski Resort
- 483 Golf Course
- 484 Bowling Center
- 485 Ice Skating Rink
- 486 Sports Center
- 487 Sports Hall
- 488 Swimming Pool
- 489 Tennis Court
- 490 Nature Attraction
- 491 Mountain Pass
- 492 Mountain Peak
- 493 Beach
- 494 Bay
- 500 Government Office

- 501 City Hall
- 502 PoliceStation
- 504 Court House
- 506 Embassy
- 507 Fire Station
- 508 Neighbour District
- 509 Public Restroom
- 510 Recycling Centre
- 511 Prison
- 530 City Feature
- 531 Community Centre
- 532 Cemetary
- 533 Park
- 534 City Centre
- 535 Sport Club
- 536 Retirement Home
- 537 Practical Information
- 538 Funeral Place
- 539 Ritual Bathes
- 540 Church
- 541 Mosque
- 542 Temple
- 543 Sinagogue
- 544 Buddhist
- 545 Hindu
- 546 Sikh
- 547 Jain
- 548 Holy Site
- 560 Hospital
- 561 Doctor
- 562 Pharmacy
- 563 Dentist
- 564 Veterinarian
- 565 Ambulance
- 566 Blood Bank
- 567 Familiy General Practice

- 568 Clinic Medical Centre
- 569 Optician
- 570 Post Office
- 571 Main Post Office
- 572 Minor Post Office
- 580 Bank
- 581 AT M
- 582 Money Exchange
- 590 Industry Facility
- 600 Vehicle Repair Facility
- 601 Vehicle Tehnical Check Station
- 602 Car Wash
- 603 Auto Parts
- 604 Car Repair
- 605 Truck Repair
- 606 Tire Repair
- 610 Petrol Station
- 611 LPG Station
- 612 Vehicle Charging Station
- 620 RentA Car
- 630 Parking
- 631 Garage Parking
- 632 Garage Parking PR
- 633 Open Parking
- 634 Open Parking PR
- 635 Ren-a-Car Parking
- 636 Park And Ride
- 637 Commuter Rail Station
- 638 Bicycle Parking
- 639 Bicycle Sharing Location
- 640 Railway Station
- 641 Main Railway Station
- 642 Minor Railway Station
- 643 Metro Station
- 644 Bus Station
- 645 Suburban

- 646 Tram Stop
- 647 Bus Stop
- 650 Airport
- 651 Civil Airport
- 652 Military Airport
- 653 Public Sport Airport
- 654 Airport Terminal
- 660 Rest Area
- 670 Car Dealer
- 671 Car Dealer Sales
- 672 Car Dealer Repair
- 673 Car Dealer Sales and Repair
- 674 Motoring Organization Office
- 675 Motorcycle Dealer
- 676 Truck Dealer
- 680 Ferry Terminal
- 681 Marina
- 682 Harbour
- 683 Quay
- 690 Lighthouse
- 692 Traffic Sign
- 693 Black Spot End
- 694 Traffic Light
- 695 Speed Camera
- 696 Speed Camera Mobile
- 697 Red Light Camera
- 698 Speed Camera Average Speed
- 699 Generic POI
- 700 Entry Point
- 701 Main Entry Point
- 702 Minor Entry Point
- 710 Airline Access
- 711 Domestic Airline Access
- 712 International Airline Access
- 713 Domestic and Internation Airline Access

- 720 Consumer Electronics Store

- 721 Office Supply Services Store
- 722 Furniture Shop
- 723 Book Store
- 724 Kiosk
- 725 Fitness Health Club

- 726 Soccer Club
- 727 Ski Lift
- 728 Archaeological Site

- 729 Big Grocery Store
- 730 Adult Entertainment
- 810 House Number
- 811 Display House Number
- 812 Search House Number
- 813 HouseNumberAsPOI
- 820 Postal Code
- 831 Apartment
- 832 Hill Station
- 833 Restaurant - Continental
- 834 Restaurant - Fast Food
- 835 Restaurant - Indian
- 836 Restaurant - Oriental
- 837 Shopping Others
- 838 Shopping Retail Chain
- 839 Shopping Retail Shop
- 840 Desert POI
- 850 Junction Point
- 871 Public Transport Stop
- 872 Public Transit Access
- 873 Taxi Station
- 880 Browse Freeway Intersection
- 881 Browse Freeway Exit
- 882 Browse Freeway Interchange
- 885 Browse Main Junction

- 886 Browse Toll Booth
- 887 Browse Mileage Start
- 888 Browse Mileage End
- 900 Intersection
- 910 Complex Street
- 920 Freeway Intersection
- 921 Freeway Exit
- 922 Freeway Interchange
- 925 Main Junction
- 930 Railway Crossing
- 1001 Route
- 1005 Pedestrian Zone
- 1007 Walkway
- 1010 Highway
- 1020 Fast Road
- 1030 Regional Road
- 1040 Main Road
- 1050 Local Road
- 1060 Connecting Road
- 1070 Slow Road
- 1080 Minor Road
- 1090 Service Road
- 1100 Railway
- 1201 Railway Long Distance
- 1202 Public Bus Line
- 1203 Tourist Bus Line
- 1204 Metro Line
- 1205 Tram Line
- 1206 Water Taxi
- 1210 Virtual Connector
- 1400 Ferry
- 1420 Ferry Train
- 1500 River
- 1501 Canal

### 0.2.3 APPENDIX C – Road Types

- 1010 Highway
- 1016 Highway Unpaved
- 1018 Highway Private
- 1020 Fast Road
- 1026 Fast Road Unpaved
- 1028 Fast Road Private
- 1030 Regional Road
- 1036 Regional Road Unpaved
- 1038 Regional Road Private
- 1040 Main Road
- 1046 Main Road Unpaved
- 1048 Main Road Private
- 1050 Local Road
- 1056 Local Road Unpaved
- 1058 Local Road Private
- 1060 Connecting Road
- 1066 Connecting Road Unpaved
- 1068 Connecting Road Private
- 1070 Slow Road
- 1076 Slow Road Unpaved
- 1078 Slow Road Private
- 1080 Minor Road
- 1086 Minor Road Unpaved
- 1088 Minor Road Private
- 1090 Service Road
- 1096 Service Road Unpaved
- 1098 Service Road Private

## 0.3 Deprecated List

**Member EasyAPI.getActiveRoute (final AddressListResult rr)**

use the overload with RouteResult callback

**Member EasyAPI.requestNotifications (final int mask, final INotificationListener notificationListener, final Result resultListener)**

Use version with BoolResult listener instead

**Class Enums.TypeFamily**

## 0.4 Hierarchical Index

### 0.4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

## 0.5 Class Index

### 0.5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

# 0.6 Class Documentation

## 0.6.1 EasyAPI.AddressListResult Interface Reference

### 0.6.1.1 Description

Callback interface for EasyAPI calls that return list of addresses.

**Public Member Functions**

- void onAddressList (int status, List< GeoAddress > addresses)

  *Called after application has executed the call.*

### 0.6.1.2 Member Function Documentation

#### 0.6.1.2.1 onAddressList()

```
void onAddressList (
    int status,
    List< GeoAddress > addresses )
```

Called after application has executed the call.

Address list will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *addresses* | Result of the call. |

## 0.6.2 EasyAPI.AddressResult Interface Reference

### 0.6.2.1 Description

Callback interface for EasyAPI calls that return single address.

#### Public Member Functions

- void onAddress (int status, GeoAddress address)

  *Called after application has executed the call.*

### 0.6.2.2 Member Function Documentation

#### 0.6.2.2.1 onAddress()

```
void onAddress (
    int status,
    GeoAddress address )
```

Called after application has executed the call.

Address will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *address* | Result of the call. |

## 0.6.3 Advice Class Reference

### 0.6.3.1 Description

Class that represents navigation instructions.

Active navigation advice is sent to application using APINotifications mechanism.

#### Classes

- class I18

  *Localized and formatted advice values.*

### Public Attributes

- final int blowupRadius

  *Radius of advice validity.*
- final int entryCourseDeg

  *Entry road course in degrees.*
- final boolean isDestination

  *If true, advice is destination advice.*
- final boolean isViaPoint

  *If true, advice is waypoint advice.*
- final int metersToAdvice

  *Distance to advice in meters.*
- final int secondsToAdvice

  *Time to advice in seconds.*
- final int adviceType

  *Type of advice.*
- final String currentStreet

  *Formatted street name of the current positions.*
- final String nextStreet

  *Formatted street name of the street where advice is leading us.*
- final String adviceText

  *Localized and formatted advice text.*
- final int roundaboudArcAngle

  *Roundabout arc angle in degrees.*
- final int roundaboutExit

  *If advice is one of the roundabout advices this is exit number starting from 1.*
- final boolean needsAttention

  *Does current instruction needs attention.*
- final List< Enums.Lane > laneInfo

  *List of lane info enums.*
- final Route route

  *Active route object.*
- final I18 i18

  *Advice data formatted using current language and units.*
- final Advice next

  *next advice*
- final double longitude

  *Advice longitude.*
- final double latitude

  *Advice latitude.*

### 0.6.3.2 Member Data Documentation

#### 0.6.3.2.1 adviceText `final String adviceText`

Localized and formatted advice text.

Only simple formatting is used: <b> for bold and <i> for italic.

#### 0.6.3.2.2 adviceType `final int adviceType`

Type of advice.

#### 0.6.3.2.3 blowupRadius `final int blowupRadius`

Radius of advice validity.

When current position is within this radius advice is shown.

#### 0.6.3.2.4 currentStreet `final String currentStreet`

Formatted street name of the current positions.

#### 0.6.3.2.5 entryCourseDeg `final int entryCourseDeg`

Entry road course in degrees.

#### 0.6.3.2.6 i18 `final I18 i18`

Advice data formatted using current language and units.

#### 0.6.3.2.7 isDestination `final boolean isDestination`

If true, advice is destination advice.

#### 0.6.3.2.8 isViaPoint `final boolean isViaPoint`

If true, advice is waypoint advice.

#### 0.6.3.2.9 laneInfo `final List<Enums.Lane> laneInfo`

List of lane info enums.

#### 0.6.3.2.10 latitude `final double latitude`

Advice latitude.

#### 0.6.3.2.11 longitude `final double longitude`

Advice longitude.

#### 0.6.3.2.12 metersToAdvice `final int metersToAdvice`

Distance to advice in meters.

#### 0.6.3.2.13 needsAttention `final boolean needsAttention`

Does current instruction needs attention.

This flag is set roughly at the times when audible instruction is about to be played. However, this flag is set even if guidance sound is muted.

#### 0.6.3.2.14 next `final Advice next`

next advice

#### 0.6.3.2.15 nextStreet `final String nextStreet`

Formatted street name of the street where advice is leading us.

#### 0.6.3.2.16 roundaboudArcAngle `final int roundaboudArcAngle`

Roundabout arc angle in degrees.

#### 0.6.3.2.17 roundaboutExit `final int roundaboutExit`

If advice is one of the roundabout advices this is exit number starting from 1.

#### 0.6.3.2.18 route `final Route route`

Active route object.

#### 0.6.3.2.19 secondsToAdvice `final int secondsToAdvice`

Time to advice in seconds.

## 0.6.4 EasyAPI.AdviceListResult Interface Reference

### 0.6.4.1 Description

Callback interface for getAdviceList EasyAPI call.

**Public Member Functions**

- void onAdviceList (int status, List< Advice > advices)

  *Called after application has executed the call.*

### 0.6.4.2 Member Function Documentation

#### 0.6.4.2.1 onAdviceList()

```
void onAdviceList (
        int status,
        List< Advice > advices )
```

Called after application has executed the call.

List of advices will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *advices* | List of advices. |

## 0.6.5 EasyAPI.AdviceResult Interface Reference

### 0.6.5.1 Description

Callback interface for getCurrentAdvice EasyAPI call.

**Public Member Functions**

- void onAdvice (int status, Advice advice)

  *Get current advice.*

#### 0.6.5.2 Member Function Documentation

##### 0.6.5.2.1 onAdvice()

```
void onAdvice (
    int status,
    Advice advice )
```

Get current advice.

Result will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *advice* | Current of advices. |

### 0.6.6 Enums.AdviceType Enum Reference

#### 0.6.6.1 Description

Advice types.

### 0.6.7 API Class Reference

#### 0.6.7.1 Description

Various constant and global methods.

**Static Public Member Functions**

- static void setUrlScheme (String url_scheme)

  *Set URL sheme for API intents.*

### Static Public Attributes

- static final int SAVED_DEST_RECENT = 1

  *Recently saved destinations.*
- static final int SAVED_DEST_FAVORITE = 2

  *Favorite saved destinations.*
- static final int HOME = 4

  *Home address.*
- static final int WORK = 8

  *Work address.*
- static final int SAVED_DEST_ALL = 0xf

  *All saved destinations.*
- static final int DISPLAY_MODE_AUTOMATIC = 0

  *Automatic display mode.*
- static final int DISPLAY_MODE_DAY = 1

  *Day display.*
- static final int DISPLAY_MODE_NIGHT = 2

  *Night display.*
- static final int UNIT_METRIC = 0

  *Use metric units*
- static final int UNIT_IMPERIAL = 1

  *Use imperial units*
- static final int ALERT_SPEED_LIMIT = 1

  *Speed limit sound alert*
- static final int ALERT_SAFETY_CAMERA = 2

  *Safety camera sound alert.*
- static final int ALERT_PETROL_STATIONS = 3

  *Next petrol station sound information.*
- static final int ALERT_OFF = 0

  *Turn off sound alert*
- static final int ALERT_TTS = 1

  *Play sound alerts as short sentences using text-to-speech.*
- static final int ALERT_CHIME = 2

  *Play sound alerts as short sounds.*
- static final int SHOW_TRAFFIC_FLOW = 1

  *Show traffic flow data.*
- static final int SHOW_TRAFFIC_EVENTS = 2

  *Show traffic events.*
- static final int SHOW_TRAFFIC_ALL = 3

  *Show both traffic flow and events.*
- static final int RESULT_OK = 0

  *Request was executed successfully.*
- static final int RESULT_FAIL = -1

  *Request failed from unknown reason.*
- static final int RESULT_NOT_IMPLEMENTED = -2

  *Request failed - unimplemented API*
- static final int RESULT_INVALID_REQUEST = -3

*Request failed - invalid request*

- static final int RESULT_INVALID_SEARCH_COUNTRY = -4

  *Search request failed - invalid country or country not covered with map specified.*
- static final int RESULT_NO_ACTIVE_ROUTE = -5

  *Routing related request is executed with no active route present.*
- static final int RESULT_VOICE_NOT_AVAILABLE = -6

  *Specified voice is not available.*
- static final int RESULT_LANGUAGE_NOT_AVAILABLE = -7

  *Specified UI language is not available.*
- static final int RESULT_ACTIVE_ROUTE_LOCKED = -8

  *Active route cannot be modified.*
- static final int RESULT_NO_SCREEN = -9

  *Request requires that application UI is in foreground.*
- static final int RESULT_CANCELED = -10

  *request is canceled*
- static final int RESULT_NOT_FOUND = -11

  *Requested item (place, route, route candidate etc..) is not found.*
- static final int GEOFENCE_ENTER = 1

  *The transition type indicating that the user enters the geofence(s).*
- static final int GEOFENCE_EXIT = 2

  *The transition type indicating that the user exits the geofence(s).*
- static final int NOTIFY_ROUTING = 0x1

  *Notify about navigation changes: navigation started, navigation stopped, route changed.*
- static final int NOTIFY_ADVICES = 0x2

  *Notify about driving instruction changes.*
- static final int NOTIFY_SPEED_LIMITS = 0x4

  *Notify about speed limit changes.*
- static final int NOTIFY_SPEED_CAMERAS = 0x8

  *Notify about safety cameras.*
- static final int NOTIFY_POSITION_DATA = 0x10

  *Notify about current position.*
- static final int NOTIFY_SPEED_VIOLATIONS = 0x20

  *Notify about speed violations.*
- static final int NOTIFY_GEO_FENCES = 0x40

  *Notify about geofence events.*
- static final int NOTIFY_FAVORITES = 0x80

  *Notify about saved places change.*
- static final int NOTIFY_ALL = 0xff

  *Notify about everything.*

### 0.6.7.2 Member Function Documentation

#### 0.6.7.2.1 setUrlScheme()

```
static void setUrlScheme (
        String url_scheme )  [static]
```

Set URL sheme for API intents.

**Parameters**

| | |
|---|---|
| *url_scheme* | url scheme for API intents - provided by Mireo (default is "gm" for Mireo GeniusMaps) |

### 0.6.7.3 Member Data Documentation

#### 0.6.7.3.1 ALERT_CHIME `final int ALERT_CHIME = 2 [static]`

Play sound alerts as short sounds.

#### 0.6.7.3.2 ALERT_OFF `final int ALERT_OFF = 0 [static]`

Turn off sound alert

#### 0.6.7.3.3 ALERT_PETROL_STATIONS `final int ALERT_PETROL_STATIONS = 3 [static]`

Next petrol station sound information.

#### 0.6.7.3.4 ALERT_SAFETY_CAMERA `final int ALERT_SAFETY_CAMERA = 2 [static]`

Safety camera sound alert.

#### 0.6.7.3.5 ALERT_SPEED_LIMIT `final int ALERT_SPEED_LIMIT = 1 [static]`

Speed limit sound alert

#### 0.6.7.3.6 ALERT_TTS `final int ALERT_TTS = 1 [static]`

Play sound alerts as short sentences using text-to-speech.

#### 0.6.7.3.7 DISPLAY_MODE_AUTOMATIC `final int DISPLAY_MODE_AUTOMATIC = 0 [static]`

Automatic display mode.

#### 0.6.7.3.8 DISPLAY_MODE_DAY `final int DISPLAY_MODE_DAY = 1 [static]`

Day display.

#### 0.6.7.3.9 DISPLAY_MODE_NIGHT `final int DISPLAY_MODE_NIGHT = 2 [static]`

Night display.

#### 0.6.7.3.10 GEOFENCE_ENTER `final int GEOFENCE_ENTER = 1 [static]`

The transition type indicating that the user enters the geofence(s).

#### 0.6.7.3.11 GEOFENCE_EXIT `final int GEOFENCE_EXIT = 2 [static]`

The transition type indicating that the user exits the geofence(s).

#### 0.6.7.3.12 HOME `final int HOME = 4 [static]`

Home address.

#### 0.6.7.3.13 NOTIFY_ADVICES `final int NOTIFY_ADVICES = 0x2 [static]`

Notify about driving instruction changes.

#### 0.6.7.3.14 NOTIFY_ALL `final int NOTIFY_ALL = 0xff [static]`

Notify about everything.

#### 0.6.7.3.15 NOTIFY_FAVORITES `final int NOTIFY_FAVORITES = 0x80 [static]`

Notify about saved places change.

#### 0.6.7.3.16 NOTIFY_GEO_FENCES `final int NOTIFY_GEO_FENCES = 0x40 [static]`

Notify about geofence events.

#### 0.6.7.3.17 NOTIFY_POSITION_DATA `final int NOTIFY_POSITION_DATA = 0x10 [static]`

Notify about current position.

#### 0.6.7.3.18 NOTIFY_ROUTING `final int NOTIFY_ROUTING = 0x1 [static]`

Notify about navigation changes: navigation started, navigation stopped, route changed.

#### 0.6.7.3.19 NOTIFY_SPEED_CAMERAS `final int NOTIFY_SPEED_CAMERAS = 0x8 [static]`

Notify about safety cameras.

#### 0.6.7.3.20 NOTIFY_SPEED_LIMITS `final int NOTIFY_SPEED_LIMITS = 0x4 [static]`

Notify about speed limit changes.

#### 0.6.7.3.21 NOTIFY_SPEED_VIOLATIONS `final int NOTIFY_SPEED_VIOLATIONS = 0x20 [static]`

Notify about speed violations.

#### 0.6.7.3.22 RESULT_ACTIVE_ROUTE_LOCKED `final int RESULT_ACTIVE_ROUTE_LOCKED = -8 [static]`

Active route cannot be modified.

#### 0.6.7.3.23 RESULT_CANCELED `final int RESULT_CANCELED = -10 [static]`

request is canceled

#### 0.6.7.3.24 RESULT_FAIL `final int RESULT_FAIL = -1 [static]`

Request failed from unknown reason.

#### 0.6.7.3.25 RESULT_INVALID_REQUEST `final int RESULT_INVALID_REQUEST = -3 [static]`

Request failed - invalid request

#### 0.6.7.3.26 RESULT_INVALID_SEARCH_COUNTRY `final int RESULT_INVALID_SEARCH_COUNTRY = -4 [static]`

Search request failed - invalid country or country not covered with map specified.

#### 0.6.7.3.27 RESULT_LANGUAGE_NOT_AVAILABLE `final int RESULT_LANGUAGE_NOT_AVAILABLE = -7 [static]`

Specified UI language is not available.

#### 0.6.7.3.28 RESULT_NO_ACTIVE_ROUTE `final int RESULT_NO_ACTIVE_ROUTE = -5 [static]`

Routing related request is executed with no active route present.

#### 0.6.7.3.29 RESULT_NO_SCREEN `final int RESULT_NO_SCREEN = -9 [static]`

Request requires that application UI is in foreground.

#### 0.6.7.3.30 RESULT_NOT_FOUND `final int RESULT_NOT_FOUND = -11 [static]`

Requested item (place, route, route candidate etc..) is not found.

#### 0.6.7.3.31 RESULT_NOT_IMPLEMENTED `final int RESULT_NOT_IMPLEMENTED = -2 [static]`

Request failed - unimplemented API

#### 0.6.7.3.32 RESULT_OK `final int RESULT_OK = 0 [static]`

Request was executed successfully.

#### 0.6.7.3.33 RESULT_VOICE_NOT_AVAILABLE `final int RESULT_VOICE_NOT_AVAILABLE = -6 [static]`

Specified voice is not available.

#### 0.6.7.3.34 SAVED_DEST_ALL `final int SAVED_DEST_ALL = 0xf [static]`

All saved destinations.

#### 0.6.7.3.35 SAVED_DEST_FAVORITE `final int SAVED_DEST_FAVORITE = 2 [static]`

Favorite saved destinations.

#### 0.6.7.3.36 SAVED_DEST_RECENT `final int SAVED_DEST_RECENT = 1 [static]`

Recently saved destinations.

#### 0.6.7.3.37 SHOW_TRAFFIC_ALL `final int SHOW_TRAFFIC_ALL = 3 [static]`

Show both traffic flow and events.

#### 0.6.7.3.38 SHOW_TRAFFIC_EVENTS `final int SHOW_TRAFFIC_EVENTS = 2 [static]`

Show traffic events.

#### 0.6.7.3.39 SHOW_TRAFFIC_FLOW `final int SHOW_TRAFFIC_FLOW = 1 [static]`

Show traffic flow data.

#### 0.6.7.3.40 UNIT_IMPERIAL `final int UNIT_IMPERIAL = 1 [static]`

Use imperial units

#### 0.6.7.3.41 UNIT_METRIC `final int UNIT_METRIC = 0 [static]`

Use metric units

#### 0.6.7.3.42 WORK `final int WORK = 8 [static]`

Work address.

## 0.6.8 APIAsyncRequest Class Reference

### 0.6.8.1 Description

Asynchronous API request.

Results can be returned to caller using callback interface or client application can wait for the request to be executed and retrieve response. Both request data and response data are encoded as JSON strings.

**Classes**

- interface IResultListener

  *Callback interface for request.*

### Public Member Functions

- String getAction ()

  *Action ("show_map", etc..).*
- long getRequestId ()

  *Request ID is used to uniquely identify request.*
- JSONObject getRequest ()

  *Request JSON data.*
- synchronized JSONObject getResult ()

  *Retrieve response JSON data.*
- synchronized void setResult (JSONObject r)

  *Internal use only!*
- boolean waitForResult (long timeout_ms)

  *Wait for the response to be ready.*
- void setResultListener (IResultListener listener)

  *Set response callback.*
- int toStatus ()

  *Get response status.*
- List< GeoAddress > toAddressList ()

  *Utility method to extract address list from response.*
- List< Advice > getAdviceList (String field_name)

  *Utility method to extract advice list from response.*
- Advice getAdvice (String field_name)

  *Utility method to extract advic from response.*
- List< Voice > getVoiceList ()

  *Utility method to extract the list of voice names from response.*
- Integer getIntValue (String name)

  *Utility method to extract int value from response.*
- VersionInfo getVersionInfo ()

  *Utility method to extract version info from response.*
- Boolean getBooleanValue (String name)

  *Utility method to extract boolean value from response.*
- String getStringValue (String name)

  *Utility method to extract string value from response.*
- String getStringValue (String name, boolean ignoreStatus)

  *Utility method to extract string value from response.*
- double getDoubleValue (String name)

  *Utility method to extract double value from response.*
- JSONObject getJsonObject (String name)

  *Utility method to extract JSON object from response.*
- GeoAddress getAddressValue (String name)

  *Utility method to extract GeoAddress value from response.*
- String[ ] getStringArray (String name)

  *Utility method to extract string array from response.*
- Byte[ ] getByteArray (String name)

  *Utility method to extract byte array from response.*
- Route getRouteValue (String name)

  *Utility method to extract route result from response.*
- List< Route > getRouteList (String name)

  *Utility method to extract routes list from response.*
- synchronized boolean hasMoreData ()

  *Check if application is done with this request.*
- synchronized void cancel ()

  *Cancel the request.*

### 0.6.8.2 Member Function Documentation

#### 0.6.8.2.1 cancel() `synchronized void cancel ( )`

Cancel the request.

#### 0.6.8.2.2 getAction() `String getAction ( )`

Action ("show_map", etc..).

**Returns**

Action string

#### 0.6.8.2.3 getAddressValue() `GeoAddress getAddressValue ( String name )`

Utility method to extract GeoAddress value from response.

**Parameters**

| | |
|---|---|
| *name* | JSON field name |

**Returns**

value of the field

#### 0.6.8.2.4 getAdvice() `Advice getAdvice ( String field_name )`

Utility method to extract advic from response.

Applicable to

**See also**

API::get_advices API response.

**Parameters**

|  |  |
|---|---|
| *field_name* | the name of JSON field that contains advice list |

**Returns**

the list of advices

### 0.6.8.2.5 getAdviceList()

```
List<Advice> getAdviceList (
        String field_name )
```

Utility method to extract advice list from response.

Applicable to

**See also**

API::get_advices API response.

**Parameters**

|  |  |
|---|---|
| *field_name* | the name of JSON field that contains advice list |

**Returns**

the list of advices

### 0.6.8.2.6 getBooleanValue()

```
Boolean getBooleanValue (
        String name )
```

Utility method to extract boolean value from response.

**Parameters**

|  |  |
|---|---|
| *name* | JSON field name |

**Returns**

value of the field

#### 0.6.8.2.7 getByteArray() `Byte [] getByteArray ( String name )`

Utility method to extract byte array from response.

**Parameters**

| | |
|---|---|
| *name* | JSON field name |

**Returns**

value of the field

#### 0.6.8.2.8 getDoubleValue() `double getDoubleValue ( String name )`

Utility method to extract double value from response.

**Parameters**

| | |
|---|---|
| *name* | JSON field name |

**Returns**

value of the field

#### 0.6.8.2.9 getIntValue() `Integer getIntValue ( String name )`

Utility method to extract int value from response.

**Parameters**

| | |
|---|---|
| *name* | JSON field name |

**Returns**

value of the field

#### 0.6.8.2.10 getJsonObject() `JSONObject getJsonObject ( String name )`

Utility method to extract JSON object from response.

**Parameters**

| | |
|---|---|
| *name* | JSON field name |

**Returns**

value of the field

#### 0.6.8.2.11 getRequest() `JSONObject getRequest ( )`

Request JSON data.

**Returns**

JSON data

#### 0.6.8.2.12 getRequestId() `long getRequestId ( )`

Request ID is used to uniquely identify request.

**Returns**

request ID

#### 0.6.8.2.13 getResult() `synchronized JSONObject getResult ( )`

Retrieve response JSON data.

**Returns**

Response JSON data

#### 0.6.8.2.14 getRouteList() `List<Route> getRouteList ( String name )`

Utility method to extract routes list from response.

**Parameters**

| | |
|---|---|
| *name* | JSON field name |

**Returns**

value of the field

### 0.6.8.2.15 getRouteValue() `Route getRouteValue ( String name )`

Utility method to extract route result from response.

**Parameters**

| | |
|---|---|
| *name* | JSON field name |

**Returns**

value of the field

### 0.6.8.2.16 getStringArray() `String [] getStringArray ( String name )`

Utility method to extract string array from response.

**Parameters**

| | |
|---|---|
| *name* | JSON field name |

**Returns**

value of the field

### 0.6.8.2.17 getStringValue() [1/2] `String getStringValue ( String name )`

Utility method to extract string value from response.

**Parameters**

| | |
|---|---|
| *name* | JSON field name |

**Returns**

value of the field

#### 0.6.8.2.18 getStringValue() [2/2]

```
String getStringValue (
        String name,
        boolean ignoreStatus )
```

Utility method to extract string value from response.

**Parameters**

| | |
|---|---|
| *name* | JSON field name |
| *ignoreStatus* | flag indicating whether to ignore status |

**Returns**

value of the field

#### 0.6.8.2.19 getVersionInfo()

```
VersionInfo getVersionInfo ( )
```

Utility method to extract version info from response.

**Returns**

value of the field

#### 0.6.8.2.20 getVoiceList()

```
List<Voice> getVoiceList ( )
```

Utility method to extract the list of voice names from response.

**Returns**

the list of voice names.

#### 0.6.8.2.21 hasMoreData()

```
synchronized boolean hasMoreData ( )
```

Check if application is done with this request.

If not, more data is expected to come from application and the callback will be called each time new data arrives. This enables streaming interface to the application.

**Returns**

false if request is finished

#### 0.6.8.2.22 setResult()

```
synchronized void setResult (
        JSONObject r )
```

Internal use only!

#### 0.6.8.2.23 setResultListener()

```
void setResultListener (
        IResultListener listener )
```

Set response callback.

**Parameters**

|  |  |
|---|---|
| *listener* | callback instance |

#### 0.6.8.2.24 toAddressList() `List<GeoAddress> toAddressList ( )`

Utility method to extract address list from response.

Applicable to responses for search API queries.

**Returns**

the list of addresses

#### 0.6.8.2.25 toStatus() `int toStatus ( )`

Get response status.

**Returns**

response status

#### 0.6.8.2.26 waitForResult()

```
boolean waitForResult (
        long timeout_ms )
```

Wait for the response to be ready.

**Parameters**

|  |  |
|---|---|
| *timeout_ms* | maximum wait time in milliseconds, -1 means wait infinitely |

**Returns**

true is response is available

## 0.6.9 Enums.Arrow Enum Reference

### 0.6.9.1 Description

Enumeration for lane arrows.

## 0.6.10 EasyAPI.AudioAlertResult Interface Reference

### 0.6.10.1 Description

Callback interface for getAudioAlertSettings.

**See also**

EasyAPI::getAudioAlertSettings(AudioAlertResult)

**Public Member Functions**

- void onAlertStatus (int status, @API.AlertKind int speed, @API.AlertKind int camera, @API.AlertKind int petrol_stations)

  *Called after application has executed the call.*

### 0.6.10.2 Member Function Documentation

#### 0.6.10.2.1 onAlertStatus()

```
void onAlertStatus (
        int status,
        @API.AlertKind int speed,
        @API.AlertKind int camera,
        @API.AlertKind int petrol_stations )
```

Called after application has executed the call.

Arguments will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *speed* | status of speeding alert. |
| *camera* | status of saftery camera alert. |
| *petrol_stations* | status of nearby petrol stations alert. |

## 0.6.11 EasyAPI.BoolResult Interface Reference

### 0.6.11.1 Description

Callback interface for EasyAPI calls that return boolean values.

**Public Member Functions**

- void onResult (int status, boolean value)

  *Called after application has executed the call.*

#### 0.6.11.2 Member Function Documentation

##### 0.6.11.2.1 onResult()

```
void onResult (
        int status,
        boolean value )
```

Called after application has executed the call.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *value* | Result value. It is only valid of status is API::RESULT_OK. |

### 0.6.12 RouteCandidates.Candidate Class Reference

#### 0.6.12.1 Description

Candidate route.

**Public Attributes**

- final int totalMeters

  *Route length in meters.*
- final int totalSeconds

  *Calculated route duration in seconds.*
- final int brokenAvoids

  *If route cannot be calculated with requested option, this field will contain features (EAvoidMask) that had to be used.*
- final String brokenAvoidsText

  *If route cannot be calculated with requested option, this field will contain the list of features that had to be used.*

#### 0.6.12.2 Member Data Documentation

##### 0.6.12.2.1 brokenAvoids

```
final int brokenAvoids
```

If route cannot be calculated with requested option, this field will contain features (EAvoidMask) that had to be used.

##### 0.6.12.2.2 brokenAvoidsText

```
final String brokenAvoidsText
```

If route cannot be calculated with requested option, this field will contain the list of features that had to be used.

For example, if toll roads cannot be avoided on the route, this field will contain the string "Route uses: toll roads". The text is localized to current language.

#### 0.6.12.2.3 totalMeters `final int totalMeters`

Route length in meters.

#### 0.6.12.2.4 totalSeconds `final int totalSeconds`

Calculated route duration in seconds.

## 0.6.13 EasyAPI Class Reference

### 0.6.13.1 Description

Easy asynchronous interface for API.

Each method returns instance of APIAsyncRequest so that asynchronous calls can be converted to synchronous with `APIAsyncRequest.waitForResult(timeout_ms)`.

EasyAPI may use Android bound service to communicate with the application. The consequence of that is that each API call may throw RuntimeException if service has died. The client has to make sure to handle the exception appropriately.

Example usage:

```
 // Our EasyAPI instance
 private EasyAPI mAPI;
 // Our notification listener implementation
 private class Notifications implements INotificationListener3 {
    @Override
    public void OnRoute(Route route) {
        Log.d(TAG, " route status changed ");
        Log.d(TAG, " navigation is " + (!route.isActive ? " not ": "") + "active");
    }
    @Override
    public void OnAdvice(Advice advice) {
        Log.d(TAG, " advice: " + advice.adviceText);
    }
    @Override
    public void OnSpeedLimit(int limit_kmh) {
        Log.d(TAG, " speed limit is " + limit_kmh + " km/h");
    }
    @Override
    public void OnSpeedCamera(int limit_kmh, int distance_to_camera_in_meters) {
        Log.d(TAG, " speed camera at : " distance_to_camera_in_meters + " meters, limit is " + limit_kmh + "
       km/h");
    }
    @Override
  void OnGeoFenceEvent(String geoFenceName, int geoFenceEvent,  PositionData pd) {
        Log.d(TAG, " geofence event : "  + geoFenceName + (geoFenceEvent == API.GEOFENCE_ENTER ? " entered
       ": " exited "));
    }
}
 ...
 // Create API instance
 mAPI = new EasyAPI("gm", this);
 mNotifications = new Notifications();
 // register for notifications
 mAPI.requestNotifications("MyClientID",
    API.NOTIFY_ADVICES + API.NOTIFY_ROUTING,
    mNotifications,
    (status , navigationActive) -> {
        Log.d(TAG, " requestNotifications " + (status == API.RESULT_OK ? "succeeded": "failed"));
        Log.d(TAG, " navigation is " + (!navigationActive ? " not ": "") + "active");
    });
 ...
 // call some methods
 mAPI.showMap(null);
 // search
 mAPI.search("Redd", "United Kingdom", 30, false,
    (status, addresses) -> {
        if (status == API.RESULT_OK)
```

```
            Log.d(TAG, " search returnd " + addresses.size() + "addressess");
    });
...
// stop listening for notification
mAPI.removeNotifications("MyClientID",
    API.NOTIFY_ADVICES + API.NOTIFY_ROUTING,
    mNotifications,
    status -> {
        Log.d(TAG, " removeNotifications " + (status == API.RESULT_OK ? "succeeded": "failed"));
    });
```

## Classes

- interface AddressListResult

  *Callback interface for EasyAPI calls that return list of addresses.*
- interface AddressResult

  *Callback interface for EasyAPI calls that return single address.*
- interface AdviceListResult

  *Callback interface for getAdviceList EasyAPI call.*
- interface AdviceResult

  *Callback interface for getCurrentAdvice EasyAPI call.*
- interface AudioAlertResult

  *Callback interface for getAudioAlertSettings.*
- interface BoolResult

  *Callback interface for EasyAPI calls that return boolean values.*
- interface GpsResult

  *Callback interface for EasyAPI getGpsPosition call.*
- interface IntResult

  *Callback interface for EasyAPI calls that return integer values.*
- interface Result

  *Callback interface for EasyAPI calls.*
- interface RouteCandidatesResult

  *Callback interface for route candidates results.*
- interface RouteListResult

  *Callback interface for list of route results.*
- interface RouteResult

  *Callback interface for route results.*
- interface RouteSettingsResult

  *Callback interface for getRouteSettings.*
- interface ShowTrafficResult

  *Callback interface for showTrafficSettings.*
- interface StringArrayResult

  *Callback interface for EasyAPI calls that return string array.*
- interface StringResult

  *Callback interface for EasyAPI calls that return string values.*
- interface TrafficStatusResult

  *Callback interface for trafficStatus.*
- interface VersionResult

  *Callback interface for getVersionEx.*
- interface VoiceListResult

  *Callback interface for getInstalledVoices EasyAPI call.*

### Public Member Functions

- EasyAPI (String urlScheme, Context context)

  *Create remote - out of process EasyAPI client.*
- EasyAPI (String urlScheme, Context context, @NonNull ComponentName svcName)

  *Create remote - out of process EasyAPI client that is connected to specified service.*
- EasyAPI (String urlScheme, APILink link)

  *Create API client with given link.*
- void connect ()

  *Connect to the application.*
- void setOnLinkDied (LinkDied ld)

  *Set callback to be called when link to the application has died.*
- void destroy ()

  *Destroy must be called when we are done with the API.*
- boolean isConnected ()

  *Check if API is connected to the application.*
- void setScreenFlags (int flags)

  *Set target screen flags.*
- int getScreenFlags ()

  *Get target screen flags.*

### Display Control

- APIAsyncRequest showMap (final Result listener)

  *Show map screen in navigation.*
- APIAsyncRequest isIdle (boolean or_navigating, final BoolResult listener)

  *Returns true if application is idle i.e.*
- APIAsyncRequest showActiveRoute (final Result listener)

  *Show active route in navigation.*
- APIAsyncRequest setMapSafeArea (int x, int y, int width, int height, int duration_ms, Result callback)

  *Set map safe area - the part of the screen that is unobscured.*
- APIAsyncRequest callUIAction (String actionName, String[ ] parameters, final Result listener)

  *Call application UI action.*
- APIAsyncRequest closeWindow (final Result listener)

  *Close navigation window.*
- void hibernate (boolean on)

  *Send the application to hibernation or wakes it up from hibernation.*
- APIAsyncRequest mapZoom (final long display_num, final double factor, final Result listener)

  *Increase or decrease zoom on map display.*
- APIAsyncRequest mapSmartZoom (final double factor, final boolean enable_smart_zoom, final Result listener)

  *Increase or decrease zoom on map display.*
- APIAsyncRequest startTracking (final boolean keep_north, final Result listener)

  *Start tracking the vehicle , ie set camera to follow current GPS position.*
- APIAsyncRequest showOnMap (GeoAddress address, final Result listener)

  *Show place on map.*
- APIAsyncRequest zoomToPlace (GeoAddress address, final Result listener)

  *Zoom to place on map.*

### Sound Control

- APIAsyncRequest setMute (final boolean mute, final Result listener)

  *Mute/unmute navigation sounds.*
- APIAsyncRequest muteGuidance (final boolean mute, final Result listener)

  *Mute/unmute only guidance sounds.*

- APIAsyncRequest repeatGuidance (final Result rr)

  *Repeat current voice guidance instruction.*
- APIAsyncRequest muteCurrentGuidance (final Result rr)

  *Mute current voice guidance instruction.*

### Reverse Geocoding

- APIAsyncRequest getCurrentAddress (@NonNull final AddressResult listener)

  *Get current address.*
- APIAsyncRequest getAddressAtLonLat (final float longitude, final float latitude, @NonNull final AddressResult listener)

  *Get address at position.*

### Routing and Navigation

- APIAsyncRequest getGpsPosition (final GpsResult listener)

  *Get current GPS position.*
- APIAsyncRequest navigateTo (final GeoAddress address, final boolean noUI, final AddressResult listener)

  *Navigate to destination.*
- APIAsyncRequest startRoute (final GeoAddress address, final AddressResult listener)

  *Start new route planning workflow.*
- APIAsyncRequest addDestination (final GeoAddress address, final AddressResult listener)

  *Add destination at the end of planned route.*
- APIAsyncRequest replaceDestination (final GeoAddress address, final AddressResult listener)

  *Replace last destination in planned route.*
- APIAsyncRequest addWaypoint (final GeoAddress address, final boolean addBeforeEnd, final AddressResult listener)

  *Add waypoint to the route.*
- APIAsyncRequest clearRoute (final Result listener)

  *Clear active planned route.*
- APIAsyncRequest startDemo (boolean start, final Result listener)

  *Start or stop current route in demonstration mode.*
- APIAsyncRequest isNavigationActive (@NonNull final BoolResult listener)

  *Check if navigation is active.*
- APIAsyncRequest setRoute (final List< GeoAddress > address, @NonNull final RouteResult listener)

  *Set complete route with all waypoints.*
- APIAsyncRequest startNavigation (final Result listener)

  *Start navigating planned route.*
- APIAsyncRequest endNavigation (final Result listener)

  *End navigation.*
- APIAsyncRequest getNextDestination (final AddressResult listener)

  *Get next destination on active navigation route ahead of current position.*
- APIAsyncRequest getFinalDestination (final AddressResult listener)

  *Get the last destination on active navigation route.*
- APIAsyncRequest getActiveRoute (final AddressListResult rr)

  *Get active navigation route.*
- APIAsyncRequest getActiveRoute (final RouteResult rr)

  *Get active navigation or planned route.*
- APIAsyncRequest getDtg (final IntResult ir)

  *Get distance to go to the next destination in meters.*
- APIAsyncRequest getTtg (final IntResult ir)

  *Get time to go to the next destination in seconds.*
- APIAsyncRequest getWholeDtg (final IntResult ir)

  *Get distance to go to the final destination in meters.*
- APIAsyncRequest getWholeTtg (final IntResult ir)

  *Get time to go to the final destination in seconds.*
- APIAsyncRequest getAdviceList (int maxItems, @NonNull final AdviceListResult alr)

  *Get the list of advices on the active route.*

- APIAsyncRequest getCurrentAdvice (@NonNull final AdviceResult alr)

  *Get the current advice.*

### Settings

- APIAsyncRequest getVersion (final IntResult listener)

  *Get navigation content version.*
- APIAsyncRequest getVersionEx (final VersionResult listener)

  *Get version information.*
- APIAsyncRequest getInstalledVoices (final VoiceListResult vlr)

  *Get the list of installed guidance voices.*
- APIAsyncRequest setVoice (final String voiceName, final Result listener)

  *Set the voice for voice guidance.*
- APIAsyncRequest enableSpeedAlerts (final boolean value, final Result listener)

  *Enable/disable speed alerts.*
- APIAsyncRequest enableSafetyCameraAlert (final boolean value, final Result listener)

  *Enable/disable safety camera alerts .*
- APIAsyncRequest enablePoiAlongRoute (final boolean value, final Result listener)

  *Enable/disable POI along the route.*
- APIAsyncRequest enableTraffic (final boolean value, final Result listener)

  *Enable/disable Live Traffic.*
- APIAsyncRequest enableAutomaticAvoid (final boolean value, final Result listener)

  *Enable/disable automatic traffic avoidance.*
- APIAsyncRequest getInstalledLanguages (@NonNull final StringArrayResult sar)

  *Get the list of installed languages.*
- APIAsyncRequest setLanguage (final String language_id, final Result listener)

  *Set application language.*
- APIAsyncRequest setToDeviceLanguage (final StringResult listener)

  *Set application language.*
- APIAsyncRequest setUnits (final boolean metric, final Result listener)

  *Set measurements units.*
- APIAsyncRequest setTimeAMPM (final boolean value, final Result listener)

  *Set time format.*
- APIAsyncRequest setDisplayMode (@API.DisplayMode final int mode, final Result listener)

  *Set application display mode.*
- APIAsyncRequest setExternalDisplayMode (@API.DisplayMode final int mode, final Result listener)

  *Set external display mode (from light sensor, for example).*
- APIAsyncRequest restorePurchases (final Result listener)

  *Restore in-app purchases.*
- APIAsyncRequest resetToDefaults (final Result listener)

  *Reset all settings to defaults.*
- APIAsyncRequest setRouteSettings (final int routeType, final int avoidMask, final Result listener)

  *Set default route settings.*
- APIAsyncRequest getRouteSettings (final RouteSettingsResult rsr)

  *Get current route settings.*
- APIAsyncRequest registerActivationCode (final String activationCode, final Result listener)

  *Register activation code for paid content.*
- APIAsyncRequest enableSignposts (boolean enable, final Result listener)

  *Enable / disable signpost display.*
- APIAsyncRequest enableJunctionViews (boolean enable, final Result listener)

  *Enable / disable junction view display.*
- APIAsyncRequest enableUiSounds (boolean enable, final Result listener)

  *Enable / disable UI sounds.*
- APIAsyncRequest setAudioAlerts (@API.AlertSource int alert_src, @API.AlertKind int alert_kind, final Result listener)

  *Set audio alert behavior.*
- APIAsyncRequest showTraffic (@API.TrafficKind int kind, boolean enable, final Result listener)

  *Show traffic.*

- APIAsyncRequest getBooleanSetting (@BooleanSettingType String name, @NonNull BoolResult listener)

  *Get boolean setting value.*
- APIAsyncRequest getAudioAlertSettings (@NonNull AudioAlertResult listener)

  *Get settings for Audio Alerts.*
- APIAsyncRequest showTrafficSettings (@NonNull ShowTrafficResult listener)

  *Get Settings for display of traffic.*
- APIAsyncRequest getDisplayMode (boolean external, @NonNull IntResult listener)

  *Get display mode.*
- APIAsyncRequest trafficStatus (@NonNull TrafficStatusResult listener)

  *Get Live traffic status.*

### Notifications

- APIAsyncRequest requestNotifications (String client_id, int mask, INotificationListener notificationListener, BoolResult resultListener)

  *Request notifications from the application.*
- APIAsyncRequest requestNotifications (final int mask, final INotificationListener notificationListener, final Result resultListener)
- APIAsyncRequest removeNotifications (final String client_id, final INotificationListener notificationListener, final Result resultListener)

  *Unsubscribe from all notifications.*
- APIAsyncRequest removeNotifications (final String client_id, final int mask, final INotificationListener notificationListener, final Result resultListener)

  *Unsubscribe from specific notifications.*

### Customization Points

- APIAsyncRequest customApi (String api_name, boolean bring_to_foreground, final StringResult result↩ Listener, Object... args)

  *Call custom API method.*
- APIAsyncRequest customApi (String api_name, final StringResult resultListener, Object... args)

  *Call custom API method in background.*
- APIAsyncRequest customApi (String api_name, Object... args)

  *Call custom API method in background .*

### Favorite routes

- APIAsyncRequest saveRoute (List< GeoAddress > routePoints, String metadata, @NonNull final RouteResult rr)

  *Save route to favorite routes.*
- APIAsyncRequest saveRoute (List< GeoAddress > routePoints, String routeName, String metadata, @NonNull final RouteResult rr)

  *Save route to favorite routes.*
- APIAsyncRequest saveCurrentRoute (String metadata, @NonNull final RouteResult rr)

  *Save current route to favorite routes.*
- APIAsyncRequest saveCurrentRoute (String name, String metadata, @NonNull final RouteResult rr)

  *Save current route to favorite routes.*
- APIAsyncRequest loadRoute (int routeId, @NonNull final RouteResult rr)

  *Load route from the list of favorite routes.*
- APIAsyncRequest removeRoute (int routeId, final Result r)

  *Remove route from favorites.*
- APIAsyncRequest getSavedRoutes ( @NonNull final RouteListResult rr)

  *List favorite routes .*

### Route Alternatives

- APIAsyncRequest calculateAlternativeRoutes (GeoAddress destination, boolean replaceCurrent, @NonNull RouteCandidatesResult callback)

  *Calculate alternative routes.*
- APIAsyncRequest showAlternativeRoute (RouteCandidates routeCandidates, int index, Result callback)

  *Show alternative route.*
- APIAsyncRequest selectAlternativeRoute (RouteCandidates routeCandidates, int index, RouteResult callback)

  *Select alternative route and start navigating.*
- APIAsyncRequest cancelAlternativeRoutes (RouteCandidates routeCandidates, Result callback)

  *Release all resources used by routeCandidates object (calculated routes etc...)*

## Static Public Member Functions

- static String statusMessage (int status)

  *Status code description.*

## Search

- APIAsyncRequest search (String query, String country, int maxResults, boolean showOnMap, @NonNull final AddressListResult alr)

  *Free-form search.*
- APIAsyncRequest tokenSearch (String tokens, int maxResults, boolean showOnMap, final AddressListResult alr)

  *Token search.*
- APIAsyncRequest exactSearch (String city, String street, String houseNumber, String country, boolean showOnMap, final AddressListResult alr)

  *"Exact" search.*
- APIAsyncRequest getPlaces (final Enums.SavedPlaceType kind, final AddressListResult alr)

  *Get saved places.*
- APIAsyncRequest setAsFavorite (GeoAddress address, boolean add, Result callback)

  *Add / remove the place from favorites.*
- APIAsyncRequest setAsHome (GeoAddress address, boolean set, Result callback)

  *Set or remove place as home.*
- APIAsyncRequest setAsWork (GeoAddress address, boolean set, Result callback)

  *Set or remove place as work.*
- APIAsyncRequest findClosestPoi (GeoAddress origin, int[ ] poiCategories, int[ ] extraPoiTypes, int maxResults, boolean showOnMap, final AddressListResult alr)

  *Find POIs closest to some point.*
- APIAsyncRequest showSearchResultsPage (String query, final Result listener)

  *Shows a list of search results in application's search view.*
- APIAsyncRequest clearSearchResults (final Result rr)

  *Clear search results from map.*

## Geofencing

- static int rgba (int red, int green, int blue, int alpha)

  *Encode color from components.*
- APIAsyncRequest addGeoFence (@NonNull String name, @NonNull List< Location > geoArea, int rgbaColorOrStyleNum, @NonNull final Result rr)

  *Add geofence area to the application.*
- APIAsyncRequest showGeoFence (@NonNull String name, boolean show, @NonNull final Result rr)

  *Show geofence area in the application.*
- APIAsyncRequest removeGeoFence (@NonNull String name, @NonNull final Result rr)

  *Remove geofence area from the application.*
- APIAsyncRequest listGeoFences (@NonNull final StringArrayResult rr)

  *List geofence areas in the application.*

### 0.6.13.2 Constructor & Destructor Documentation

#### 0.6.13.2.1 EasyAPI() [1/3]

```
EasyAPI (
    String urlScheme,
    Context context )
```

Create remote - out of process EasyAPI client.

URL scheme is supplied by Mireo, "gm" is value used with GeniusMaps

**Parameters**

| | |
|---|---|
| *urlScheme* | URL scheme to use |
| *context* | Parent context |

#### 0.6.13.2.2 EasyAPI() [2/3]

```
EasyAPI (
    String urlScheme,
    Context context,
    @NonNull ComponentName svcName )
```

Create remote - out of process EasyAPI client that is connected to specified service.

Component name is supplied by Mireo.

**Parameters**

| | |
|---|---|
| *urlScheme* | URL scheme to use ("gm" for GeniusMaps) |
| *context* | Parent context |
| *svcName* | Component name of the remote service |

#### 0.6.13.2.3 EasyAPI() [3/3]

```
EasyAPI (
    String urlScheme,
    APILink link )
```

Create API client with given link.

@exclude

**Parameters**

| | |
|---|---|
| *urlScheme* | URL scheme to use (gm:// for GeniusMaps) |
| *link* | API transport link |

### 0.6.13.3 Member Function Documentation

#### 0.6.13.3.1 addDestination()

```
APIAsyncRequest addDestination (
        final GeoAddress address,
        final AddressResult listener )
```

Add destination at the end of planned route.

Part of route planning API.

**Parameters**

| | |
|---|---|
| *address* | destination address |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.2 addGeoFence()

```
APIAsyncRequest addGeoFence (
        @NonNull String name,
        @NonNull List< Location > geoArea,
        int rgbaColorOrStyleNum,
        @NonNull final Result rr )
```

Add geofence area to the application.

**Parameters**

| | |
|---|---|
| *name* | unique geofence name |
| *geoArea* | list of geofence points. It is assumed that polygon is closed, so there is no need to repeat starting point at the end |
| *rgbaColorOrStyleNum* | optional encoded color (from EasyAPI::rgba) or style. If value is less than 29, one of 30 predefined styles will be used |
| *rr* | result callback |

#### 0.6.13.3.3 addWaypoint()

```
APIAsyncRequest addWaypoint (
        final GeoAddress address,
        final boolean addBeforeEnd,
        final AddressResult listener )
```

Add waypoint to the route.

Waypoint is added just before the last destination if `addBeforeEnd` is `true`, and just after the start if `addBeforeEnd` is `false`.

**Parameters**

| | |
|---|---|
| *address* | waypoint address |
| *addBeforeEnd* | add waypoint before end |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.4 calculateAlternativeRoutes()

```
APIAsyncRequest calculateAlternativeRoutes (
        GeoAddress destination,
        boolean replaceCurrent,
        @NonNull RouteCandidatesResult callback )
```

Calculate alternative routes.

Calculated RouteCandidates object will be returned in callback.

**Parameters**

| | |
|---|---|
| *destination* | destination address |
| *replaceCurrent* | replace current route |
| *callback* | callback for result |

#### 0.6.13.3.5 callUIAction()

```
APIAsyncRequest callUIAction (
        String actionName,
        String[ ] parameters,
        final Result listener )
```

Call application UI action.

**Parameters**

| | |
|---|---|
| *actionName* | one of SHOW_HOME, SHOW_SETTINGS, SEARCH, SEARCH_NEARBY, SHOW_MAPS |
| *parameters* | parameters of screen SHOW_HOME , SHOW_SETTINGS, SEARCH, SHOW_MAPS: nothing SEARCH_NEARBY: [search origin (JSON string of GeoAddress) or null, search string] |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.6 cancelAlternativeRoutes()

```
APIAsyncRequest cancelAlternativeRoutes (
        RouteCandidates routeCandidates,
        Result callback )
```

Release all resources used by routeCandidates object (calculated routes etc...)

**Parameters**

| | |
|---|---|
| *routeCandidates* | object to clear |
| *callback* | result callback. |

#### 0.6.13.3.7 clearRoute()

```
APIAsyncRequest clearRoute (
final Result listener )
```

Clear active planned route.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may be null) |

#### 0.6.13.3.8 clearSearchResults()

```
APIAsyncRequest clearSearchResults (
        final Result rr )
```

Clear search results from map.

#### 0.6.13.3.9 closeWindow()

```
APIAsyncRequest closeWindow (
        final Result listener )
```

Close navigation window.

Call this to put navigation to background. This method is usually called from background service or broadcast receiver. Since navigation is configured as SINGLE_TOP activity you may need to bring your UI activity to from manually, ie. Android default activity stack may not do what is desired.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may be null) |

#### 0.6.13.3.10 connect()

```
void connect ( )
```

Connect to the application.

Normally, this method should not be used since API is connected to the application by default. This method can be used when connection is broken for whatever reason: application crash etc.

#### 0.6.13.3.11 customApi() [1/3]

```
APIAsyncRequest customApi (
        String api_name,
        boolean bring_to_foreground,
        final StringResult resultListener,
        Object... args )
```

Call custom API method.

This method enables additional customization of application for additional customer requirements not covered with other methods. Details of the custom API protocol are determined between customer and Mireo.

**Parameters**

| | |
|---|---|
| *api_name* | name of the API |
| *bring_to_foreground* | Bring application to foreground |
| *resultListener* | Result callback (may be null) |
| *args* | List of API arguments. |

#### 0.6.13.3.12 customApi() [2/3]

```
APIAsyncRequest customApi (
        String api_name,
        final StringResult resultListener,
        Object... args )
```

Call custom API method in background.

**Parameters**

| | |
|---|---|
| *api_name* | name of the API |
| *resultListener* | Result callback (may be null) |
| *args* | List of API arguments. |

#### 0.6.13.3.13 customApi() [3/3]

```
APIAsyncRequest customApi (
        String api_name,
        Object... args )
```

Call custom API method in background .

Overload without callback parameter.

**Parameters**

| | |
|---|---|
| *api_name* | name of the API |
| *args* | List of api arguments |

#### 0.6.13.3.14 destroy()

```
void destroy ( )
```

Destroy must be called when we are done with the API.

#### 0.6.13.3.15 enableAutomaticAvoid()

```
APIAsyncRequest enableAutomaticAvoid (
        final boolean value,
        final Result listener )
```

Enable/disable automatic traffic avoidance.

**Parameters**

| | |
|---|---|
| *value* | true to enable |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.16 enableJunctionViews()

```
APIAsyncRequest enableJunctionViews (
        boolean enable,
        final Result listener )
```

Enable / disable junction view display.

**Parameters**

| | |
|---|---|
| *enable* | enable junction view if true |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.17 enablePoiAlongRoute()

```
APIAsyncRequest enablePoiAlongRoute (
        final boolean value,
        final Result listener )
```

Enable/disable POI along the route.

**Parameters**

| | |
|---|---|
| *value* | true to enable |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.18 enableSafetyCameraAlert()

```
APIAsyncRequest enableSafetyCameraAlert (
        final boolean value,
        final Result listener )
```

Enable/disable safety camera alerts .

**Parameters**

| | |
|---|---|
| *value* | true to enable |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.19 enableSignposts()

```
APIAsyncRequest enableSignposts (
boolean enable,
final Result listener )
```

Enable / disable signpost display.

**Parameters**

| | |
|---|---|
| *enable* | enable signposts if true |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.20 enableSpeedAlerts()

```
APIAsyncRequest enableSpeedAlerts (
    final boolean value,
    final Result listener )
```

Enable/disable speed alerts.

**Parameters**

| | |
|---|---|
| *value* | enable/dissable speed alerts |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.21 enableTraffic()

```
APIAsyncRequest enableTraffic (
    final boolean value,
    final Result listener )
```

Enable/disable Live Traffic.

**Parameters**

| | |
|---|---|
| *value* | true to enable |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.22 enableUiSounds()

```
APIAsyncRequest enableUiSounds (
    boolean enable,
    final Result listener )
```

Enable / disable UI sounds.

**Parameters**

| | |
|---|---|
| *enable* | enable UI sounds (click and/or TTS) |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.23 endNavigation()

```
APIAsyncRequest endNavigation (
        final Result listener )
```

End navigation.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may be null) |

#### 0.6.13.3.24 exactSearch()

```
APIAsyncRequest exactSearch (
        String city,
        String street,
        String houseNumber,
        String country,
        boolean showOnMap,
        final AddressListResult alr )
```

"Exact" search.

Arguments in this method are in fact just hints to the search engine. Search engine will try to narrow the search using the arguments provided. If the value specified is empty string or null, that field will be ignored

**Parameters**

| | |
|---|---|
| *city* | city |
| *street* | street |
| *houseNumber* | house number |
| *country* | country, empty value means all countries |
| *showOnMap* | show results on map |

#### 0.6.13.3.25 findClosestPoi()

```
APIAsyncRequest findClosestPoi (
        GeoAddress origin,
        int[] poiCategories,
        int[] extraPoiTypes,
        int maxResults,
        boolean showOnMap,
        final AddressListResult alr )
```

Find POIs closest to some point.

**Parameters**

| | |
|---|---|
| *origin* | origin of search |
| *poiCategories* | list of poi categories |

**See also**

Enums.TypeFamily

**Parameters**

| | |
|---|---|
| *extraPoiTypes* | extra POI types, outside listed categories |
| *maxResults* | maximum number of results to return |
| *showOnMap* | show results on map |

#### 0.6.13.3.26 getActiveRoute() [1/2]

```
APIAsyncRequest getActiveRoute (
        final AddressListResult rr )
```

Get active navigation route.

The list of destinations is reported to callback.

**Deprecated** use the overload with RouteResult callback

#### 0.6.13.3.27 getActiveRoute() [2/2]

```
APIAsyncRequest getActiveRoute (
        final RouteResult rr )
```

Get active navigation or planned route.

Resulting Route is reported to callback.

#### 0.6.13.3.28 getAddressAtLonLat()

```
APIAsyncRequest getAddressAtLonLat (
        final float longitude,
        final float latitude,
        @NonNull final AddressResult listener )
```

Get address at position.

Resulting GeoAddress is reported in callback.

**Parameters**

| | |
|---|---|
| *longitude* | longitude degrees |
| *latitude* | latitude degrees |
| *listener* | Result callback (may NOT be null) |

#### 0.6.13.3.29 getAdviceList()

```
APIAsyncRequest getAdviceList (
              int maxItems,
              @NonNull final AdviceListResult alr )
```

Get the list of advices on the active route.

**Parameters**

| | |
|---|---|
| *maxItems* | maximum number of advices to return |

#### 0.6.13.3.30 getAudioAlertSettings() APIAsyncRequest getAudioAlertSettings ( @NonNull AudioAlertResult *listener* )

Get settings for Audio Alerts.

**Parameters**

| | |
|---|---|
| *listener* | result callback - cannot be null |

**Returns**

request

#### 0.6.13.3.31 getBooleanSetting() APIAsyncRequest getBooleanSetting ( @BooleanSettingType String *name*, @NonNull BoolResult *listener* )

Get boolean setting value.

**Parameters**

| | |
|---|---|
| *name* | setting name (one of the BooleanSettingsType values) |
| *listener* | result listener (must not be null) |

**Returns**

request

#### 0.6.13.3.32 getCurrentAddress() APIAsyncRequest getCurrentAddress ( @NonNull final AddressResult *listener* )

Get current address.

Resulting GeoAddress is reported in callback.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may NOT be null) |

### 0.6.13.3.33 getCurrentAdvice() `APIAsyncRequest getCurrentAdvice ( @NonNull final AdviceResult alr )`

Get the current advice.

### 0.6.13.3.34 getDisplayMode()

```
APIAsyncRequest getDisplayMode (
    boolean external,
    @NonNull IntResult listener )
```

Get display mode.

**Parameters**

| | |
|---|---|
| *external* | external display mode (set by device) or internal display mode (set by user) |
| *listener* | result listener (must not be null) |

**Returns**

request

### 0.6.13.3.35 getDtg() `APIAsyncRequest getDtg ( final IntResult ir )`

Get distance to go to the next destination in meters.

### 0.6.13.3.36 getFinalDestination() `APIAsyncRequest getFinalDestination ( final AddressResult listener )`

Get the last destination on active navigation route.

Resulting GeoAddress is reported in callback.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may NOT be null) |

#### 0.6.13.3.37 getGpsPosition()

```
APIAsyncRequest getGpsPosition (
        final GpsResult listener )
```

Get current GPS position.

Result of this method will be delivered in callback.

**Parameters**

| | |
|---|---|
| *listener* | callback where result will be delivered (may NOT be null) |

#### 0.6.13.3.38 getInstalledLanguages()

```
APIAsyncRequest getInstalledLanguages (
        @NonNull final StringArrayResult sar )
```

Get the list of installed languages.

**Parameters**

| | |
|---|---|
| *sar* | Result callback (MUST NOT be null) |

#### 0.6.13.3.39 getInstalledVoices()

```
APIAsyncRequest getInstalledVoices (
        final VoiceListResult vlr )
```

Get the list of installed guidance voices.

#### 0.6.13.3.40 getNextDestination()

```
APIAsyncRequest getNextDestination (
        final AddressResult listener )
```

Get next destination on active navigation route ahead of current position.

Resulting GeoAddress is reported in callback.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may NOT be null) |

#### 0.6.13.3.41 getPlaces()

```
APIAsyncRequest getPlaces (
        final Enums.SavedPlaceType kind,
        final AddressListResult alr )
```

Get saved places.

**Parameters**

| | |
|---|---|
| *kind* | SavedPlaceType.favourites, SavedPlaceType.recents, SavedPlaceType.both |

#### 0.6.13.3.42 getRouteSettings() `APIAsyncRequest getRouteSettings ( final RouteSettingsResult rsr )`

Get current route settings.

Upon completion, current route settings will be delivered to RouteSettingsResult.onRouteSettings method of the callback.

Route type is one of Enums.ERouteType and avoid type is bitmask of Enum.EAvoidMask values

**Parameters**

| | |
|---|---|
| *rsr* | callback (may NOT be null) |

#### 0.6.13.3.43 getSavedRoutes() `APIAsyncRequest getSavedRoutes ( @NonNull final RouteListResult rr )`

List favorite routes .

**Parameters**

| | |
|---|---|
| *rr* | callback |

**Returns**

callback will be called with the list of saved routes

#### 0.6.13.3.44 getScreenFlags() `int getScreenFlags ( )`

Get target screen flags.

#### 0.6.13.3.45 getTtg() `APIAsyncRequest getTtg ( final IntResult ir )`

Get time to go to the next destination in seconds.

#### 0.6.13.3.46 getVersion() `APIAsyncRequest getVersion ( final IntResult listener )`

Get navigation content version.

Result will be returned in callback as 8 digit number yyyymmvv where y is year, m is month and v is build (ie 20171101).

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may be null) |

#### 0.6.13.3.47 getVersionEx() `APIAsyncRequest getVersionEx ( final VersionResult listener )`

Get version information.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may be null) |

#### 0.6.13.3.48 getWholeDtg() `APIAsyncRequest getWholeDtg ( final IntResult ir )`

Get distance to go to the final destination in meters.

#### 0.6.13.3.49 getWholeTtg() `APIAsyncRequest getWholeTtg ( final IntResult ir )`

Get time to go to the final destination in seconds.

#### 0.6.13.3.50 hibernate() `void hibernate ( boolean on )`

Send the application to hibernation or wakes it up from hibernation.

Call this to put navigation into hibernation or wake it from hibernation. When application is sent to hibernation, application writes its state on permanent storage and go to deep sleep. Application uses minimum amount of resources during hibernation. Most notably, any further API calls will be ignored until `hibernate(false)` is called. This method is usually used in automotive installations during ignition off/on cycles.

**Parameters**

| | |
|---|---|
| *on* | hibernate or wake up the application |

### 0.6.13.3.51 isConnected() `boolean isConnected ( )`

Check if API is connected to the application.

**Returns**

### 0.6.13.3.52 isIdle()

```
APIAsyncRequest isIdle (
        boolean or_navigating,
        final BoolResult listener )
```

Returns true if application is idle i.e.

there's no ongoing user interaction: menus, dialogs...) in the application.

**Parameters**

| | |
|---|---|
| *or_navigating* | if true, the method will return true even if navigation is active (but no menus, dialogs...) |
| *listener* | Result callback (may be null) |

### 0.6.13.3.53 isNavigationActive()

```
APIAsyncRequest isNavigationActive (
        @NonNull final BoolResult listener )
```

Check if navigation is active.

**Parameters**

| | |
|---|---|
| *listener* | Result callback |

### 0.6.13.3.54 listGeoFences()

```
APIAsyncRequest listGeoFences (
        @NonNull final StringArrayResult rr )
```

List geofence areas in the application.

**Parameters**

| | |
|---|---|
| *rr* | result callback. |

#### 0.6.13.3.55 loadRoute()

```
APIAsyncRequest loadRoute (
        int routeId,
        @NonNull final RouteResult rr )
```

Load route from the list of favorite routes.

This method sets saved route as active route, possibly cancelling active navigation, and starts new navigation.

**Parameters**

| | |
|---|---|
| *route↩ Id* | saved route ID |
| *rr* | callback |

**Returns**

if the call is successful, callback parameter will be saved route

#### 0.6.13.3.56 mapSmartZoom()

```
APIAsyncRequest mapSmartZoom (
        final double factor,
        final boolean enable_smart_zoom,
        final Result listener )
```

Increase or decrease zoom on map display.

If factor is greater than 0 map zoom will increase (zoom out) and it will decrease otherwise. If enable smart zoom is true, camera will move according to the situation: the speed, distance to the next maneuver...

**Parameters**

| | |
|---|---|
| *factor* | zoom factor |
| *enable_smart_zoom* | enable smart zoom |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.57 mapZoom()

```
APIAsyncRequest mapZoom (
        final long display_num,
        final double factor,
        final Result listener )
```

Increase or decrease zoom on map display.

Factor is multiplied with current map zoom level which means zoom out if factor is greater than 1 and zoom in otherwise. If factor is 1, than automatic ("smart") zoom feature is also selected.

**Parameters**

| | |
|---|---|
| *display_num* | display number. Main screen number is 0. |
| *factor* | zoom factor |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.58 muteCurrentGuidance()

```
APIAsyncRequest muteCurrentGuidance (
        final Result rr )
```

Mute current voice guidance instruction.

#### 0.6.13.3.59 muteGuidance()

```
APIAsyncRequest muteGuidance (
        final boolean mute,
        final Result listener )
```

Mute/unmute only guidance sounds.

**Parameters**

| | |
|---|---|
| *mute* | true means mute, false unmute |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.60 navigateTo()

```
APIAsyncRequest navigateTo (
        final GeoAddress address,
        final boolean noUI,
        final AddressResult listener )
```

Navigate to destination.

**Parameters**

| | |
|---|---|
| *address* | destination address |
| *noUI* | do not show UI ie. start navigation in background |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.61 registerActivationCode()

```
APIAsyncRequest registerActivationCode (
        final String activationCode,
        final Result listener )
```

Register activation code for paid content.

Code type and value is agreed upon between Mireo and client/OEM manufacturer/ODM. It can be Vehiche Identification Number (VIN) in case of in-car navigation or activation code from package, in case of PND device.

**Parameters**

| | |
|---|---|
| *activationCode* | Activation code |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.62 removeGeoFence()

```
APIAsyncRequest removeGeoFence (
        @NonNull String name,
        @NonNull final Result rr )
```

Remove geofence area from the application.

**Parameters**

| | |
|---|---|
| *name* | unique geofence name |
| *rr* | result callback |

#### 0.6.13.3.63 removeNotifications() [1/2]

```
APIAsyncRequest removeNotifications (
        final String client_id,
        final INotificationListener notificationListener,
        final Result resultListener )
```

Unsubscribe from all notifications.

**Parameters**

| | |
|---|---|
| *client_id* | Unique client id |
| *notificationListener* | Notification callback |
| *resultListener* | Result callback (may be null) |

#### 0.6.13.3.64 removeNotifications() [2/2]

```
APIAsyncRequest removeNotifications (
        final String client_id,
        final int mask,
        final INotificationListener notificationListener,
        final Result resultListener )
```

Unsubscribe from specific notifications.

**Parameters**

| | |
|---|---|
| *client_id* | Notification client ID |
| *mask* | Notification mask (combination of API.NOTIFY_∗ constants) |
| *notificationListener* | Notification callback |
| *resultListener* | Result callback (may be null) |

### 0.6.13.3.65 removeRoute()

```
APIAsyncRequest removeRoute (
        int routeId,
        final Result r )
```

Remove route from favorites.

**Parameters**

| | |
|---|---|
| *route↩ Id* | saved route ID |
| *r* | callback |

### 0.6.13.3.66 repeatGuidance()

```
APIAsyncRequest repeatGuidance (
        final Result rr )
```

Repeat current voice guidance instruction.

### 0.6.13.3.67 replaceDestination()

```
APIAsyncRequest replaceDestination (
        final GeoAddress address,
        final AddressResult listener )
```

Replace last destination in planned route.

Part of route planning API.

**Parameters**

| | |
|---|---|
| *address* | new destination address |
| *listener* | Result callback (may be null) |

### 0.6.13.3.68 requestNotifications() [1/2]

```
APIAsyncRequest requestNotifications (
        final int mask,
        final INotificationListener notificationListener,
        final Result resultListener )
```

**Deprecated** Use version with BoolResult listener instead

#### 0.6.13.3.69 requestNotifications() [2/2]

```
APIAsyncRequest requestNotifications (
        String client_id,
        int mask,
        INotificationListener notificationListener,
        BoolResult resultListener )
```

Request notifications from the application.

Setting mask to zero has the same effect as calling removeNotifications

**Parameters**

| | |
|---|---|
| *client_id* | Unique client id |
| *mask* | Notification mask (combination of API.NOTIFY_* constants) |
| *notificationListener* | Notification callback |
| *resultListener* | Result callback (may be null) |

#### 0.6.13.3.70 resetToDefaults()

```
APIAsyncRequest resetToDefaults (
        final Result listener )
```

Reset all settings to defaults.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may be null) |

#### 0.6.13.3.71 restorePurchases()

```
APIAsyncRequest restorePurchases (
        final Result listener )
```

Restore in-app purchases.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may be null) |

#### 0.6.13.3.72 rgba()

```
static int rgba (
        int red,
        int green,
        int blue,
        int alpha )  [static]
```

Encode color from components.

**Parameters**

| | |
|---|---|
| *red* | red component (0-255) |
| *green* | green component (0-255) |
| *blue* | blue component (0-255) |
| *alpha* | alpha component (0-255) |

**Returns**

encoded color value

### 0.6.13.3.73 saveCurrentRoute() [1/2]

```
APIAsyncRequest saveCurrentRoute (
              String metadata,
              @NonNull final RouteResult rr )
```

Save current route to favorite routes.

If the call is successful, callback will receive with saved route information. Saved route will contain route_id (Route::routeId) that can be used in EasyAPI::loadRoute.

**Parameters**

| | |
|---|---|
| *metadata* | save route with metadata (#Route.metadata) |
| *rr* | callback |

### 0.6.13.3.74 saveCurrentRoute() [2/2]

```
APIAsyncRequest saveCurrentRoute (
              String name,
              String metadata,
              @NonNull final RouteResult rr )
```

Save current route to favorite routes.

If the call is successful, callback will receive with saved route information. Saved route will contain route_id (Route::routeId) that can be used in EasyAPI::loadRoute.

**Parameters**

| | |
|---|---|
| *name* | save route with name |
| *metadata* | save route with metadata (#Route.metadata) |
| *rr* | callback |

#### 0.6.13.3.75 saveRoute() [1/2]

```
APIAsyncRequest saveRoute (
        List< GeoAddress > routePoints,
        String metadata,
        @NonNull final RouteResult rr )
```

Save route to favorite routes.

If the call is successful, callback will receive saved route information. Saved route will contain route_↩ id (Route::routeId) that can be used in EasyAPI::loadRoute.

**Parameters**

| | |
|---|---|
| *routePoints* | route destinations |
| *metadata* | save route with metadata (Route::metadata) |
| *rr* | callback |

#### 0.6.13.3.76 saveRoute() [2/2]

```
APIAsyncRequest saveRoute (
        List< GeoAddress > routePoints,
        String routeName,
        String metadata,
        @NonNull final RouteResult rr )
```

Save route to favorite routes.

If the call is successful, callback will receive saved route information. Saved route will contain route_↩ id (Route::routeId) that can be used in EasyAPI::loadRoute.

**Parameters**

| | |
|---|---|
| *routePoints* | route destinations |
| *routeName* | route name |
| *metadata* | save route with metadata (Route::metadata) |
| *rr* | callback |

#### 0.6.13.3.77 search()

```
APIAsyncRequest search (
        String query,
        String country,
        int maxResults,
        boolean showOnMap,
        @NonNull final AddressListResult alr )
```

Free-form search.

**Parameters**

| | |
|---|---|
| *query* | search query |
| *country* | search in country, empty value means all countries |
| *maxResults* | maximum number of results to return |
| *showOnMap* | show results on map |

### 0.6.13.3.78 selectAlternativeRoute()

```
APIAsyncRequest selectAlternativeRoute (
        RouteCandidates routeCandidates,
        int index,
        RouteResult callback )
```

Select alternative route and start navigating.

**Parameters**

| | |
|---|---|
| *routeCandidates* | route candidates |
| *index* | selected route |
| *callback* | callback |

### 0.6.13.3.79 setAsFavorite()

```
APIAsyncRequest setAsFavorite (
        GeoAddress address,
        boolean add,
        Result callback )
```

Add / remove the place from favorites.

If parameter add is true, the place is added to favorites, otherwise, it is removed from favorites.

**Parameters**

| | |
|---|---|
| *address* | place address |
| *add* | add or remove place to favorites |

### 0.6.13.3.80 setAsHome()

```
APIAsyncRequest setAsHome (
        GeoAddress address,
        boolean set,
        Result callback )
```

Set or remove place as home.

**Parameters**

| | |
|---|---|
| *address* | place address |
| *set* | true to set the place as home, false to remove it |

#### 0.6.13.3.81 setAsWork()

```
APIAsyncRequest setAsWork (
        GeoAddress address,
        boolean set,
        Result callback )
```

Set or remove place as work.

**Parameters**

| | |
|---|---|
| *address* | place address |
| *set* | true to set the place as work, false to remove it |

#### 0.6.13.3.82 setAudioAlerts()

```
APIAsyncRequest setAudioAlerts (
        @API.AlertSource int alert_src,
        @API.AlertKind int alert_kind,
        final Result listener )
```

Set audio alert behavior.

**Parameters**

| | |
|---|---|
| *alert_src* | alert source: one of API.ALERT_SPEED_LIMIT, API.ALERT_SAFETY_CAMERA, API.ALERT_PETROL_STATIONS |
| *alert_kind* | alert kind : one of API.ALERT_OFF, API.ALERT_TTS, API.ALERT_CHIME |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.83 setDisplayMode()

```
APIAsyncRequest setDisplayMode (
        @API.DisplayMode final int mode,
        final Result listener )
```

Set application display mode.

If `mode` is `API.DISPLAY_MODE_AUTOMATIC`, display fmode is calculated from time of day and year, calculated form GPS data

**Parameters**

| | |
|---|---|
| *mode* | API.DISPLAY_MODE_AUTOMATIC, API.DISPLAY_MODE_DAY, API.DISPLAY_MODE_NIGHT |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.84 setExternalDisplayMode()

```
APIAsyncRequest setExternalDisplayMode (
@API.DisplayMode final int mode,
final Result listener )
```

Set external display mode (from light sensor, for example).

If `mode` is different from `API.DISPLAY_MODE_AUTOMATIC`, this value overrides display mode calculated from current GPS data.

It comes into effect when user's selected display mode is API.DISPLAY_MODE_AUTOMATIC.

**Parameters**

| | |
|---|---|
| *mode* | API.DISPLAY_MODE_AUTOMATIC, API.DISPLAY_MODE_DAY, API.DISPLAY_MODE_NIGHT |
| *listener* | Result callback (may be null) |

### 0.6.13.3.85 setLanguage()

```
APIAsyncRequest setLanguage (
    final String language_id,
    final Result listener )
```

Set application language.

**Parameters**

| | |
|---|---|
| *language←↩ _id* | language id from the list of installed languages |
| *listener* | Result callback (may be null) |

### 0.6.13.3.86 setMapSafeArea()

```
APIAsyncRequest setMapSafeArea (
    int x,
    int y,
    int width,
    int height,
    int duration_ms,
    Result callback )
```

Set map safe area - the part of the screen that is unobscured.

**Parameters**

| | |
|---|---|
| *x* | leftmost coordinate |
| *y* | topmost coordinate |
| *width* | area width |
| *height* | area height |
| *duration_ms* | transition duration in millseconds |
| *callback* | callback |

#### 0.6.13.3.87 setMute()

```
APIAsyncRequest setMute (
    final boolean mute,
    final Result listener )
```

Mute/unmute navigation sounds.

**Parameters**

| | |
|---|---|
| *mute* | true means mute, false unmute |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.88 setOnLinkDied()

```
void setOnLinkDied (
    LinkDied ld )
```

Set callback to be called when link to the application has died.

**Parameters**

| | |
|---|---|
| *ld* | callback |

#### 0.6.13.3.89 setRoute()

```
APIAsyncRequest setRoute (
    final List< GeoAddress > address,
    @NonNull final RouteResult listener )
```

Set complete route with all waypoints.

This call replaces full route planning workflow as if the following sequence of calls is executed:

```
List<GeoAddress> addresses = ...;
setRoute(addresses, ....)
```

is equivalent to:

```
List<GeoAddress> addresses = ...;
startRoute(addresses.get(0))
addDestination(addresses.get(1))
addDestination(addresses.get(2))
...
```

**Parameters**

| | |
|---|---|
| *address* | list of destinations. If the fist element is null, planning will start from current GPS position |
| *listener* | Result callback. Must not be null. |

#### 0.6.13.3.90 setRouteSettings()

```
APIAsyncRequest setRouteSettings (
    final int routeType,
    final int avoidMask,
    final Result listener )
```

Set default route settings.

**Parameters**

| | |
|---|---|
| *routeType* | default route type: currently only Enums.ERouteType.quickest (1) is supported |
| *avoidMask* | or-ed values of EAvoidMask enum |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.91 setScreenFlags()

```
void setScreenFlags (
    int flags )
```

Set target screen flags.

This method is used when there are multiple screens and API method target only specific screen(s).

**Parameters**

| | |
|---|---|
| *flags* | screen flags |

#### 0.6.13.3.92 setTimeAMPM()

```
APIAsyncRequest setTimeAMPM (
    final boolean value,
    final Result listener )
```

Set time format.

**Parameters**

| | |
|---|---|
| *value* | if true use 12 hour (AM/PM) time format, use 24 hour format otherwise |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.93 setToDeviceLanguage()

```
APIAsyncRequest setToDeviceLanguage (
    final StringResult listener )
```

Set application language.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may not be null) |

#### 0.6.13.3.94 setUnits()

```
APIAsyncRequest setUnits (
    final boolean metric,
    final Result listener )
```

Set measurements units.

**Parameters**

| | |
|---|---|
| *metric* | if true, use metric units, use imperial units otherwise |
| *listener* | Result callback (may be null) |

### 0.6.13.3.95 setVoice()

```
APIAsyncRequest setVoice (
        final String voiceName,
        final Result listener )
```

Set the voice for voice guidance.

**Parameters**

| | |
|---|---|
| *voiceName* | voice name. Has to be one of the voices returned with getInstalledVoices() |

### 0.6.13.3.96 showActiveRoute()

```
APIAsyncRequest showActiveRoute (
        final Result listener )
```

Show active route in navigation.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may be null) |

### 0.6.13.3.97 showAlternativeRoute()

```
APIAsyncRequest showAlternativeRoute (
        RouteCandidates routeCandidates,
        int index,
        Result callback )
```

Show alternative route.

**Parameters**

| | |
|---|---|
| *routeCandidates* | route candidates |
| *index* | selected route |
| *callback* | callback - may be null |

### 0.6.13.3.98 showGeoFence()

```
APIAsyncRequest showGeoFence (
@NonNull String name,
boolean show,
@NonNull final Result rr )
```

Show geofence area in the application.

**Parameters**

| | |
|---|---|
| *name* | unique geofence name |
| *show* | show (true) or hide (false) geofence polygon |
| *rr* | result callback |

#### 0.6.13.3.99 showMap() `APIAsyncRequest showMap ( final Result listener )`

Show map screen in navigation.

Puts current activity in background an start main activity of the navigation.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may be null) |

#### 0.6.13.3.100 showOnMap() `APIAsyncRequest showOnMap ( GeoAddress address, final Result listener )`

Show place on map.

**Parameters**

| | |
|---|---|
| *address* | place to show |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.101 showSearchResultsPage() `APIAsyncRequest showSearchResultsPage ( String query, final Result listener )`

Shows a list of search results in application's search view.

Input to this method is JSON string. Following keys are supported:

- `poi_family` - integer value of constant from `Enums.ESuperFamily` or `Enums.EFamily`

Example: `{ "poi_family": Enums.EFamily.fmGasStations.value }`

**Parameters**

| | |
|---|---|
| *query* | search query JSON string |

#### 0.6.13.3.102 showTraffic()

```
APIAsyncRequest showTraffic (
    @API.TrafficKind int kind,
    boolean enable,
    final Result listener )
```

Show traffic.

**Parameters**

| | |
|---|---|
| *kind* | SHOW_TRAFFIC_FLOW, SHOW_TRAFFIC_EVENTS, SHOW_TRAFFIC_ALL |
| *enable* | show or hide |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.103 showTrafficSettings()

```
APIAsyncRequest showTrafficSettings (
    @NonNull ShowTrafficResult listener )
```

Get Settings for display of traffic.

**Parameters**

| | |
|---|---|
| *listener* | result callback - must not be null |

**Returns**

request

#### 0.6.13.3.104 startDemo()

```
APIAsyncRequest startDemo (
    boolean start,
    final Result listener )
```

Start or stop current route in demonstration mode.

**Parameters**

| | |
|---|---|
| *start* | true to start demo mode, false to stop it |
| *listener* | callback, may be null |

**Returns**

#### 0.6.13.3.105 startNavigation()

```
APIAsyncRequest startNavigation (
        final Result listener )
```

Start navigating planned route.

**Parameters**

| | |
|---|---|
| *listener* | Result callback (may be null) |

#### 0.6.13.3.106 startRoute()

```
APIAsyncRequest startRoute (
        final GeoAddress address,
        final AddressResult listener )
```

Start new route planning workflow.

The workflow for route planning may look like: ` startRoute(...) addDestination(...) addDestination(...) add↩ Waypoint(...) ... startNavigation(...) `

**Parameters**

| | |
|---|---|
| *address* | route start address. If address is null planning will start from current GPS position |
| *listener* | Result callback (may be null). |

#### 0.6.13.3.107 startTracking()

```
APIAsyncRequest startTracking (
        final boolean keep_north,
        final Result listener )
```

Start tracking the vehicle , ie set camera to follow current GPS position.

**Parameters**

| | |
|---|---|
| *keep_north* | Keep map oriented to north |
| *listener* | Result callback (may be null) |

#### 0.6.13.3.108 statusMessage()

```
static String statusMessage (
        int status )  [static]
```

Status code description.

**Parameters**

| | |
|---|---|
| *status* | Status code |

**Returns**

Status message

#### 0.6.13.3.109 tokenSearch()

```
APIAsyncRequest tokenSearch (
        String tokens,
        int maxResults,
        boolean showOnMap,
        final AddressListResult alr )
```

Token search.

Input to this method is JSON string with search tokens.

Following tokens are supported:

- query - free form text, represents unknown token
- street - street name
- city - city
- postal_code - postal code
- house_no - house number
- poi - Point of Interest,
- street_or_poi - search engine will try to find either street or POI with given token value
- category - localized search category name (eg - "restaurant"), or #NNN where NNN is Mireo geo type identifier (for the list, see Appendix B)
- country - search country

Example: ` { "POI" : "McDonald's" "city" : "Zagreb" "country" : "Croatia" "category" : "fast food" } `

**Parameters**

| | |
|---|---|
| *tokens* | search tokens JSON string |
| *maxResults* | maximum number of results to return |
| *showOnMap* | show results on map |

#### 0.6.13.3.110 trafficStatus()

```
APIAsyncRequest trafficStatus (
        @NonNull TrafficStatusResult listener )
```

Get Live traffic status.

**Parameters**

| | |
|---|---|
| *listener* | result callback - must not be null |

**Returns**

request

#### 0.6.13.3.111 zoomToPlace()

```
APIAsyncRequest zoomToPlace (
    GeoAddress address,
    final Result listener )
```

Zoom to place on map.

Differs from EasyAPI::showOnMap because it will not create marker.

**Parameters**

| | |
|---|---|
| *address* | place to show |
| *listener* | callback (may be null) |

## 0.6.14 Enums.EAvoidMask Enum Reference

### 0.6.14.1 Description

Routing flags.

**Public Attributes**

- avoid_highway =(1)

  *Avoid highways.*
- avoid_ferries =(2)

  *Avoid ferries.*
- avoid_toll_roads =(4)

  *Avoid toll roads.*
- avoid_unpaved =(8)

  *Avoid unpaved roads.*
- use_palestine =(16)

  *(Israel only) if this flag is set USE the roads in palestinian areas*
- even_ring_roads =(32)

  *(Greece only) Apply rules in Athens for vehicles with even plate number*
- odd_ring_roads =(64)

  *(Greece only) Apply rules in Athens for vehicles with odd plate number*
- even_iran_roads =(128)

  *(Iran only) Apply rules in Tehran for vehicles with even plate number*
- odd_iran_roads =(256)

  *(Iran only) Apply rules in Tehran for vehicles with odd plate number*
- use_special =(512)

  *(Iran only) Apply rules in Tehran for special vehicles*

### 0.6.14.2 Member Data Documentation

#### 0.6.14.2.1 avoid_ferries `avoid_ferries =(2)`

Avoid ferries.

#### 0.6.14.2.2 avoid_highway `avoid_highway =(1)`

Avoid highways.

#### 0.6.14.2.3 avoid_toll_roads `avoid_toll_roads =(4)`

Avoid toll roads.

#### 0.6.14.2.4 avoid_unpaved `avoid_unpaved =(8)`

Avoid unpaved roads.

#### 0.6.14.2.5 even_iran_roads `even_iran_roads =(128)`

(Iran only) Apply rules in Tehran for vehicles with even plate number

#### 0.6.14.2.6 even_ring_roads `even_ring_roads =(32)`

(Greece only) Apply rules in Athens for vehicles with even plate number

#### 0.6.14.2.7 odd_iran_roads `odd_iran_roads =(256)`

(Iran only) Apply rules in Tehran for vehicles with odd plate number

#### 0.6.14.2.8 odd_ring_roads `odd_ring_roads =(64)`

(Greece only) Apply rules in Athens for vehicles with odd plate number

#### 0.6.14.2.9 use_palestine `use_palestine =(16)`

(Israel only) if this flag is set USE the roads in palestinian areas

#### 0.6.14.2.10 use_special `use_special =(512)`

(Iran only) Apply rules in Tehran for special vehicles

## 0.6.15 Enums.ECargoType Enum Reference

### 0.6.15.1 Description

Cargo types for truck routing calculations.

**Public Attributes**

- notdangerous =(0)

  *Normal cargo.*
- explosives =(1)

  *Potentially explosive material.*
- waterharmful =(2)

  *Material harmful to water.*
- hazardous =(3)

  *Hazardous material.*

### 0.6.15.2 Member Data Documentation

#### 0.6.15.2.1 explosives `explosives =(1)`

Potentially explosive material.

#### 0.6.15.2.2 hazardous `hazardous =(3)`

Hazardous material.

#### 0.6.15.2.3 notdangerous `notdangerous =(0)`

Normal cargo.

#### 0.6.15.2.4 waterharmful `waterharmful =(2)`

Material harmful to water.

## 0.6.16 Enums.EFamily Enum Reference

### 0.6.16.1 Description

New POI families.

## 0.6.17 Enums Class Reference

### 0.6.17.1 Description

Enumerations.

**Classes**

- enum AdviceType

  *Advice types.*
- enum Arrow

  *Enumeration for lane arrows.*
- enum EAvoidMask

  *Routing flags.*
- enum ECargoType

  *Cargo types for truck routing calculations.*
- enum EFamily

  *New POI families.*
- enum ERouteType

  *Route types for routing calculations.*
- enum ESuperFamily

  *New POI super families.*
- enum ETruckType

  *Truck types for routing calculations.*
- enum EVehicleType

  *Vehicle types for routing calculations.*
- enum Lane

  *Enumeration for lane info.*
- enum SavedPlaceType

  *Enumeration for saved places.*
- enum TypeFamily

  *POI types.*

## 0.6.18 Enums.ERouteType Enum Reference

### 0.6.18.1 Description

Route types for routing calculations.

Only
`ERouteType#quickest`

is supported right now.

**Public Attributes**

- quickest =(1)

  *Calculate quickest route.*

### 0.6.18.2 Member Data Documentation

#### 0.6.18.2.1 quickest `quickest =(1)`

Calculate quickest route.

## 0.6.19 Enums.ESuperFamily Enum Reference

### 0.6.19.1 Description

New POI super families.

## 0.6.20 Enums.ETruckType Enum Reference

### 0.6.20.1 Description

Truck types for routing calculations.

Only important when vehicle type is
`EVehicleType#truck`

**Public Attributes**

- transport =(0)

  *Transport truck.*
- delivery =(1)

  *Delivery truck.*
- trailer =(2)

  *Truck with trailer.*

#### 0.6.20.2 Member Data Documentation

##### 0.6.20.2.1 delivery `delivery =(1)`

Delivery truck.

##### 0.6.20.2.2 trailer `trailer =(2)`

Truck with trailer.

##### 0.6.20.2.3 transport `transport =(0)`

Transport truck.

### 0.6.21 Enums.EVehicleType Enum Reference

#### 0.6.21.1 Description

Vehicle types for routing calculations.

**Public Attributes**

- passenger =(0)

  *Calculate "normal", passenger vehicle route.*
- taxi =(1)

  *Calculate route for Taxis.*
- pedestrian =(2)

  *Calculate pedestrian route.*
- truck =(3)

  *Calculate route for trucks.*

#### 0.6.21.2 Member Data Documentation

##### 0.6.21.2.1 passenger `passenger =(0)`

Calculate "normal", passenger vehicle route.

#### 0.6.21.2.2 pedestrian `pedestrian =(2)`

Calculate pedestrian route.

#### 0.6.21.2.3 taxi `taxi =(1)`

Calculate route for Taxis.

#### 0.6.21.2.4 truck `truck =(3)`

Calculate route for trucks.

## 0.6.22 GeoAddress Class Reference

### 0.6.22.1 Description

Address structure.

**Classes**

- class SphereUtils

  *Utilities for conversion between internal coordinate representation and WGS84 coordinates.*

**Public Member Functions**

- void setLonLat (double lon, double lat)

  *Set address' longitude and latitude.*
- String getLabel ()

  *Get localized address label.*
- String getDescription ()

  *Get localized address description.*
- boolean isHome ()

  *Check if address is home address.*
- boolean isWork ()

  *Check if address is work address.*
- boolean isFavorite ()

  *Check if address is in favorites.*
- boolean isRecent ()

  *Check if address is in recent places.*

**Static Public Member Functions**

- static GeoAddress fromLonLat (double lon, double lat)

  *construct GeoAddress from longitude and latitude*

### Public Attributes

- double longitude

  *Longitude.*
- double latitude

  *Latitude.*
- String city = ""

  *City.*
- String postal = ""

  *Postal code.*
- String street = ""

  *Street.*
- String houseNumber = ""

  *House number.*
- String name = ""

  *Some extra address description, for example, contact name in case it's a contact's address.*
- String POI = ""

  *POI name, if address is of some POI.*
- String country = ""

  *Country name.*
- String phone = ""

  *Phone number.*
- String area = ""

  *Geographic area: province, state or similar.*
- String type = ""

  *Address type.*
- int x = 0

  *Internal x coordinate.*
- int y = 0

  *Internal y coordinate.*
- int distance

  *Distance to current location.*
- int duration

  *(Optional) Trip duration to the address*
- int iso

  *Country numeric iso code.*
- double confidence = 0.0

  *Search result confidence from 0 - 1.*
- String[ ] formatted

  *Formatted address lines.*

### 0.6.22.2 Member Function Documentation

#### 0.6.22.2.1 fromLonLat()

```
static GeoAddress fromLonLat (
        double lon,
        double lat )  [static]
```

construct GeoAddress from longitude and latitude

**Parameters**

| | |
|---|---|
| *lon* | degrees longitude |
| *lat* | degrees latitude |

**Returns**

GeoAddress

#### 0.6.22.2.2 getDescription() `String getDescription ( )`

Get localized address description.

**Returns**

address label

#### 0.6.22.2.3 getLabel() `String getLabel ( )`

Get localized address label.

**Returns**

address label

#### 0.6.22.2.4 isFavorite() `boolean isFavorite ( )`

Check if address is in favorites.

#### 0.6.22.2.5 isHome() `boolean isHome ( )`

Check if address is home address.

#### 0.6.22.2.6 isRecent() `boolean isRecent ( )`

Check if address is in recent places.

#### 0.6.22.2.7 isWork() `boolean isWork ( )`

Check if address is work address.

#### 0.6.22.2.8 setLonLat()

```
void setLonLat (
        double lon,
        double lat )
```

Set address' longitude and latitude.

**Parameters**

| | |
|---|---|
| *lon* | degrees longitude |
| *lat* | degrees latitude |

### 0.6.22.3 Member Data Documentation

#### 0.6.22.3.1 area `String area = ""`

Geographic area: province, state or similar.

#### 0.6.22.3.2 city `String city = ""`

City.

#### 0.6.22.3.3 confidence `double confidence = 0.0`

Search result confidence from 0 - 1.

Only available if the address is result of search query

#### 0.6.22.3.4 country `String country = ""`

Country name.

#### 0.6.22.3.5 distance `int distance`

Distance to current location.

Only available if the address is result of search nearby queries

#### 0.6.22.3.6 duration `int duration`

(Optional) Trip duration to the address

#### 0.6.22.3.7 formatted `String [] formatted`

Formatted address lines.

#### 0.6.22.3.8 houseNumber `String houseNumber = ""`

House number.

#### 0.6.22.3.9 iso `int iso`

Country numeric iso code.

#### 0.6.22.3.10 latitude `double latitude`

Latitude.

#### 0.6.22.3.11 longitude `double longitude`

Longitude.

#### 0.6.22.3.12 name `String name = ""`

Some extra address description, for example, contact name in case it's a contact's address.

#### 0.6.22.3.13 phone `String phone = ""`

Phone number.

#### 0.6.22.3.14 POI `String POI = ""`

POI name, if address is of some POI.

#### 0.6.22.3.15 postal `String postal = ""`

Postal code.

#### 0.6.22.3.16 street `String street = ""`

Street.

#### 0.6.22.3.17 type `String type = ""`

Address type.

It may be POI type if address is POI, or road type. Possible values for this field are listed in apendices

#### 0.6.22.3.18 x `int x = 0`

Internal x coordinate.

#### 0.6.22.3.19 y `int y = 0`

Internal y coordinate.

## 0.6.23 EasyAPI.GpsResult Interface Reference

### 0.6.23.1 Description

Callback interface for EasyAPI getGpsPosition call.

**Public Member Functions**

- void onGps (int status, boolean connected, boolean valid, double longitude, double latitude)
  *Called with latest GPS position or error code.*

### 0.6.23.2 Member Function Documentation

#### 0.6.23.2.1 onGps()

```
void onGps (
        int status,
        boolean connected,
        boolean valid,
        double longitude,
        double latitude )
```

Called with latest GPS position or error code.

Arguments will have valid values only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Statuus of the operation |
| *connected* | Is GPS source connected? |
| *valid* | Is current GPS valid? |
| *longitude* | Geographic longitude |
| *latitude* | Geographic latitude |

## 0.6.24 Advice.I18 Class Reference

### 0.6.24.1 Description

Localized and formatted advice values.

Values are localized with respect to current unit type and time display format. Also, numeric values are formatted in user friendly manner.

**Public Attributes**

- final String distanceToAdvice

  *localized distance to advice*
- final String timeToAdvice

  *localized time to advice*
- final String distanceToDestination

  *localized time string*
- final String timeToDestination

  *localized time to destination*
- final String routeETA

  *localized estimated time of arrival*
- .UnitType final int units

  *units used : 0 metric, 1 imperial*

### 0.6.24.2 Member Data Documentation

#### 0.6.24.2.1 distanceToAdvice `final String distanceToAdvice`

localized distance to advice

#### 0.6.24.2.2 distanceToDestination `final String distanceToDestination`

localized time string

#### 0.6.24.2.3 routeETA `final String routeETA`

localized estimated time of arrival

#### 0.6.24.2.4 timeToAdvice `final String timeToAdvice`

localized time to advice

#### 0.6.24.2.5 timeToDestination `final String timeToDestination`

localized time to destination

#### 0.6.24.2.6 units `.UnitType final int units`

units used : 0 metric, 1 imperial

## 0.6.25 INotificationListener Interface Reference

### 0.6.25.1 Description

Notification callbacks interface.

When client registers itself to certain kinds of events using `EasyAPI::requestNotifications` the events are sent to this callback.

Inherited by INotificationListener2.

**Public Member Functions**

- void OnRoute (@NonNull Route route)

  *Called when active route changes.*
- void OnAdvice (@NonNull Advice advice)

  *Called when advice changes.*
- void OnSpeedLimit (int limit_kmh)

  *Called when speed limit is changed.*
- void OnSpeedCamera (int limit_kmh, int distance_to_camera_in_meters)

  *Called to notify about speed camera is changed.*

### 0.6.25.2 Member Function Documentation

#### 0.6.25.2.1 OnAdvice()

```
void OnAdvice (
        @NonNull Advice advice )
```

Called when advice changes.

Usually called every second to update distance and time to advice.

**Parameters**

| | |
|---|---|
| *advice* | Current advice |

#### 0.6.25.2.2 OnRoute()

```
void OnRoute (
        @NonNull Route route )
```

Called when active route changes.

**Parameters**

| | |
|---|---|
| *route* | active route |

#### 0.6.25.2.3 OnSpeedCamera()

```
void OnSpeedCamera (
        int limit_kmh,
        int distance_to_camera_in_meters )
```

Called to notify about speed camera is changed.

**Parameters**

| | |
|---|---|
| *limit_kmh* | Speed limit in km/h. |
| *distance_to_camera_in_meters* | Distance to speed camera in meters. |

#### 0.6.25.2.4 OnSpeedLimit()

```
void OnSpeedLimit (
        int limit_kmh )
```

Called when speed limit is changed.

**Parameters**

| | |
|---|---|
| *limit_kmh* | Current speed limit in km/h |

## 0.6.26 INotificationListener2 Interface Reference

### 0.6.26.1 Description

Extended notification callbacks interface.

`EasyAPI::requestNotifications` the events are sent to this callback.

Inherits INotificationListener.

Inherited by INotificationListener3.

### Public Member Functions

- void OnPositionData (@NonNull PositionData pd)

  *Called to notify about position data change.*
- void OnSpeedViolations (int limit_kmh, int current_speed_kmh)

  *Called on speed limit violation .*

### 0.6.26.2 Member Function Documentation

#### 0.6.26.2.1 OnPositionData() `void OnPositionData ( @NonNull PositionData pd )`

Called to notify about position data change.

**Parameters**

| | |
|---|---|
| *pd* | Position data. |

#### 0.6.26.2.2 OnSpeedViolations() `void OnSpeedViolations ( int limit_kmh, int current_speed_kmh )`

Called on speed limit violation .

**Parameters**

| | |
|---|---|
| *limit_kmh* | Current speed limit in km/h |
| *current_speed_kmh* | Current speed in km/h |

## 0.6.27 INotificationListener3 Interface Reference

### 0.6.27.1 Description

Extended notification callbacks interface.

This is extension of the INotificationListener2 with geofencing event `EasyAPI::requestNotifications` the events are sent to this callback.

Inherits INotificationListener2.

### Public Member Functions

- void OnGeoFenceEvent (@NonNull String geoFenceName, int geoFenceEvent, PositionData pd)

  *Called to notify about geo fence events.*
- default void OnPlaceChanged (@NonNull List< PlaceChange > changes)

  *Called to notify about changes to places.*
- default void OnSavedRouteChanged (@NonNull List< SavedRouteChange > changes)

  *Called to notify about changes to favorite routes.*

### 0.6.27.2 Member Function Documentation

#### 0.6.27.2.1 OnGeoFenceEvent()

```
void OnGeoFenceEvent (
        @NonNull String geoFenceName,
        int geoFenceEvent,
        PositionData pd )
```

Called to notify about geo fence events.

**Parameters**

| | |
|---|---|
| *geoFenceName* | Name of the geofence. |
| *geoFenceEvent* | geofence event - (One of `API::GEOFENCE_ENTER` and `API::GEOFENCE_EXIT`) . |
| *pd* | Position that triggered the event. |

#### 0.6.27.2.2 OnPlaceChanged()

```
default void OnPlaceChanged (
        @NonNull List< PlaceChange > changes )
```

Called to notify about changes to places.

The following changes are monitored : favorite place added or removed, list of recently visited places changed, home or work place selected.

**Parameters**

| | |
|---|---|
| *changes* | list of changed places |

#### 0.6.27.2.3 OnSavedRouteChanged()

```
default void OnSavedRouteChanged (
        @NonNull List< SavedRouteChange > changes )
```

Called to notify about changes to favorite routes.

**Parameters**

| | |
|---|---|
| *changes* | list of changed routes |

## 0.6.28 EasyAPI.IntResult Interface Reference

### 0.6.28.1 Description

Callback interface for EasyAPI calls that return integer values.

**Public Member Functions**

- void onResult (int status, int value)

  *Called after application has executed the call.*

### 0.6.28.2 Member Function Documentation

#### 0.6.28.2.1 onResult()

```
void onResult (
    int status,
    int value )
```

Called after application has executed the call.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *value* | Result value. It is only valid of status is API::RESULT_OK |

## 0.6.29 APIAsyncRequest.IResultListener Interface Reference

### 0.6.29.1 Description

Callback interface for request.

**Public Member Functions**

- void onResult (APIAsyncRequest response)

  *This method is called when the request is executed.*

### 0.6.29.2 Member Function Documentation

#### 0.6.29.2.1 onResult()

```
void onResult (
    APIAsyncRequest response )
```

This method is called when the request is executed.

**Parameters**

| | |
|---|---|
| *response* | response JSON data can be reterived using `getResult` method |

## 0.6.30 Enums.Lane Enum Reference

### 0.6.30.1 Description

Enumeration for lane info.

Naming:

- XXX: Lane with direction XXX; driver should go XXX
- XXX__YYY: Lane with directions XXX and YYY; driver should go XXX
- __XXX: Lane with direction XXX; driver should not use this lane
- __XXX_YYY: Lane with directions XXX and YYY, driver should not use this lane.

## 0.6.31 LaneInfo Class Reference

### 0.6.31.1 Description

Class that represents lane information.

**Static Public Member Functions**

- static List< Enums.Lane > fromJson (JSONArray laneInfo, JSONArray maskedLaneInfo)

  *Converts json arrays of integers to list of Lanes.*

### 0.6.31.2 Member Function Documentation

#### 0.6.31.2.1 fromJson()

```
static List<Enums.Lane> fromJson (
    JSONArray laneInfo,
    JSONArray maskedLaneInfo )  [static]
```

Converts json arrays of integers to list of Lanes.

**Parameters**

| | |
|---|---|
| *laneInfo* | json array of integers; nine least significant bits of each integer describe (unhighlighted/all) arrows of lane info; RLL\|LLR\|RRS where R=Right, L=Left, S=Straight and if any of {L\|R\|S} equals 1 this integer represents {L\|R\|S} arrow |
| *maskedLaneInfo* | json array of integers; nine least significant bits of each integer describe highlighted arrows of lane info in the same way as laneInfo parameter; i-th integer in maskedLaneInfo array (a_i) refers to j-th integer in laneInfo array (b_j) iff (a_i >> 9) == (j) |

**Returns**

list of Lanes

## 0.6.32 LinkDied Interface Reference

### 0.6.32.1 Description

Callback to be called when link to the application has died.

## 0.6.33 PlaceChange Class Reference

### 0.6.33.1 Description

Encapsulate change of saved place.

See INotificationListener3::OnSavedPlacesChanged.

**Public Attributes**

- final GeoAddress place

  *Address of changed place.*
- final int changeType

  *Change type.*

**Static Public Attributes**

- static final int SET_HOME = API.HOME

  *the place is marked as home place*
- static final int REMOVE_HOME = ∼API.HOME

  *the place is no longer marked as home place*
- static final int SET_WORK = API.WORK

  *the place is marked as work place*
- static final int REMOVE_WORK = ∼API.WORK

  *the place is no longer marked as work place*
- static final int SET_AS_FAVORITE = API.SAVED_DEST_FAVORITE

  *the place is added to favorite places*
- static final int REMOVE_FAVORITE = ∼API.SAVED_DEST_FAVORITE

  *the place is removed from favorite places*
- static final int SET_AS_RECENT = API.SAVED_DEST_RECENT

  *the place is added to recent destinations*
- static final int REMOVE_RECENT = ∼API.SAVED_DEST_RECENT

  *the place is removed from recent destinations*

### 0.6.33.2 Member Data Documentation

#### 0.6.33.2.1 changeType `final int changeType`

Change type.

#### 0.6.33.2.2 place `final GeoAddress place`

Address of changed place.

#### 0.6.33.2.3 REMOVE_FAVORITE `final int REMOVE_FAVORITE = ~API.SAVED_DEST_FAVORITE [static]`

the place is removed from favorite places

#### 0.6.33.2.4 REMOVE_HOME `final int REMOVE_HOME = ~API.HOME [static]`

the place is no longer marked as home place

#### 0.6.33.2.5 REMOVE_RECENT `final int REMOVE_RECENT = ~API.SAVED_DEST_RECENT [static]`

the place is removed from recent destinations

#### 0.6.33.2.6 REMOVE_WORK `final int REMOVE_WORK = ~API.WORK [static]`

the place is no longer marked as work place

#### 0.6.33.2.7 SET_AS_FAVORITE `final int SET_AS_FAVORITE = API.SAVED_DEST_FAVORITE [static]`

the place is added to favorite places

#### 0.6.33.2.8 SET_AS_RECENT `final int SET_AS_RECENT = API.SAVED_DEST_RECENT [static]`

the place is added to recent destinations

#### 0.6.33.2.9 SET_HOME `final int SET_HOME = API.HOME [static]`

the place is marked as home place

#### 0.6.33.2.10 SET_WORK `final int SET_WORK = API.WORK [static]`

the place is marked as work place

## 0.6.34 PositionData Class Reference

### 0.6.34.1 Description

PostionData is used in notifications.

**Static Public Member Functions**

- static PositionData fromJSON (JSONObject o)

  *Convert from JSON.*

**Public Attributes**

- final double longitude

  *Longitude.*
- final double latitude

  *Latitude.*
- final int speed

  *Speed in km/h.*
- final int course

  *Course in degrees from north eastward.*
- final int utc

  *UTC timestamp.*
- final int quality

  *HDOP.*
- final int altitude

  *Altitude in meters.*
- final int speed_limit

  *Speed limit in km/h.*
- final int camera_limit

  *Camera limit in km/h.*
- final int camera_distance

  *Distance to camera in meters.*
- final int road_type

  *Optional road type.*
- final int country_code

  *Optional country ISO code.*
- final String current_street_name

  *Current street name.*

### 0.6.34.2 Member Function Documentation

#### 0.6.34.2.1 fromJSON() `static PositionData fromJSON ( JSONObject o ) [static]`

Convert from JSON.

**Parameters**

| | |
|---|---|
| *o* | JSONObject |

**Returns**

PositionData instance

### 0.6.34.3 Member Data Documentation

#### 0.6.34.3.1 altitude `final int altitude`

Altitude in meters.

#### 0.6.34.3.2 camera_distance `final int camera_distance`

Distance to camera in meters.

#### 0.6.34.3.3 camera_limit `final int camera_limit`

Camera limit in km/h.

#### 0.6.34.3.4 country_code `final int country_code`

Optional country ISO code.

#### 0.6.34.3.5 course `final int course`

Course in degrees from north eastward.

#### 0.6.34.3.6 current_street_name `final String current_street_name`

Current street name.

#### 0.6.34.3.7 latitude `final double latitude`

Latitude.

#### 0.6.34.3.8 longitude `final double longitude`

Longitude.

#### 0.6.34.3.9 quality `final int quality`

HDOP.

#### 0.6.34.3.10 road_type `final int road_type`

Optional road type.

#### 0.6.34.3.11 speed `final int speed`

Speed in km/h.

#### 0.6.34.3.12 speed_limit `final int speed_limit`

Speed limit in km/h.

#### 0.6.34.3.13 utc `final int utc`

UTC timestamp.

## 0.6.35 RemoteLink Class Reference

### 0.6.35.1 Description

RemoteLink is used to send API request using BroadcastReceiver.

Responses are also received with local Broadcast receiver

Inherits APILink.

**Public Member Functions**

- void close ()

  *Unregisters receiver and close the link.*
- void open ()

  *Registers BroadcastReceiver to listen for API responses.*

### 0.6.35.2 Member Function Documentation

#### 0.6.35.2.1 close() `void close ( )`

Unregisters receiver and close the link.

#### 0.6.35.2.2 open() `void open ( )`

Registers BroadcastReceiver to listen for API responses.

## 0.6.36 EasyAPI.Result Interface Reference

### 0.6.36.1 Description

Callback interface for EasyAPI calls.

When API call is processed by application the callback method onResult is called with the status of the call.

**Public Member Functions**

- void onResult (int status)

  *Called after application has executed the call.*

### 0.6.36.2 Member Function Documentation

#### 0.6.36.2.1 onResult()

```
void onResult (
        int status )
```

Called after application has executed the call.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |

## 0.6.37 Route Class Reference

### 0.6.37.1 Description

Class that represents route.

**Classes**

- class Subtrip

  *Route subtrip ("leg") data.*

**Static Public Member Functions**

- static Route fromJSON (JSONObject o)

  *Convert form JSON object.*

**Public Attributes**

- final ArrayList< GeoAddress > routePoints

  *List of route points.*
- final boolean isNavigation

  *True if route is navigation route (as opposed to route plan)*
- final boolean isActive

  *True if route is active.*
- final boolean isCompleted

  *True if route is completed.*
- final int totalMeters

  *Route length in meters.*
- final int totalSeconds

  *Calculated route duration in seconds.*
- final int requestedFeatures

  *Requested avoidance features.*
- final int usedFeatures

  *Used avoidance features.*
- final int completedMeters

  *Distance in meters completed so far.*
- final int elapsedSeconds

  *Duration in seconds spent on route so far.*
- final int remainingMeters

  *Remaining distance to final destination in meters (DTG)*
- final int remainingSeconds

  *Remaining time to final destination (TTG)*
- final int dtgToNext

*Remaining distance to the next waypoint.*

- final int ttgToNext

    *Remaining time to the next waypoint.*

- final String metadata

    *Comma separated list of additional, application specific, key-value pairs, for example, "kind=business" or "kind=personal".*

- final String name

    *optional route name*

- final int routeId

    *Saved route id, -1 if route is not saved to favorite routes list.*

- final ArrayList< Subtrip > subtrips = new ArrayList<>()

    *Array of subtrips (route legs)*

### 0.6.37.2 Member Function Documentation

#### 0.6.37.2.1 fromJSON() `static Route fromJSON ( JSONObject o ) [static]`

Convert form JSON object.

**Parameters**

| | |
|---|---|
| *o* | JSON object |

**Returns**

route object

### 0.6.37.3 Member Data Documentation

#### 0.6.37.3.1 completedMeters `final int completedMeters`

Distance in meters completed so far.

#### 0.6.37.3.2 dtgToNext `final int dtgToNext`

Remaining distance to the next waypoint.

#### 0.6.37.3.3 elapsedSeconds `final int elapsedSeconds`

Duration in seconds spent on route so far.

#### 0.6.37.3.4 isActive `final boolean isActive`

True if route is active.

#### 0.6.37.3.5 isCompleted `final boolean isCompleted`

True if route is completed.

#### 0.6.37.3.6 isNavigation `final boolean isNavigation`

True if route is navigation route (as opposed to route plan)

#### 0.6.37.3.7 metadata `final String metadata`

Comma separated list of additional, application specific, key-value pairs, for example, "kind=business" or "kind=personal".

#### 0.6.37.3.8 name `final String name`

optional route name

#### 0.6.37.3.9 remainingMeters `final int remainingMeters`

Remaining distance to final destination in meters (DTG)

#### 0.6.37.3.10 remainingSeconds `final int remainingSeconds`

Remaining time to final destination (TTG)

#### 0.6.37.3.11 requestedFeatures `final int requestedFeatures`

Requested avoidance features.

Mask of Enums.EAvoidMask values.

#### 0.6.37.3.12 routeId `final int routeId`

Saved route id, -1 if route is not saved to favorite routes list.

#### 0.6.37.3.13 routePoints `final ArrayList<GeoAddress> routePoints`

List of route points.

This member is null when used in notifications, since it's expensive to pass the array of addresses to often

#### 0.6.37.3.14 subtrips `final ArrayList<Subtrip> subtrips = new ArrayList<>()`

Array of subtrips (route legs)

#### 0.6.37.3.15 totalMeters `final int totalMeters`

Route length in meters.

#### 0.6.37.3.16 totalSeconds `final int totalSeconds`

Calculated route duration in seconds.

#### 0.6.37.3.17 ttgToNext `final int ttgToNext`

Remaining time to the next waypoint.

#### 0.6.37.3.18 usedFeatures `final int usedFeatures`

Used avoidance features.

May be different form requested features if route cannot be calculated to honor all of the requested features.

## 0.6.38 RouteCandidates Class Reference

### 0.6.38.1 Description

Class that keeps up to 3 alternative routes.

Route with index 0 is default.

Object of this class must be released with EasyAPI::cancelAlternativeRoutes call.

#### Classes

- class Candidate

  *Candidate route.*

#### Public Attributes

- final ArrayList< Candidate > candidates = new ArrayList<>()

  *List of all route candidates.*
- final GeoAddress start

  *starting address*
- final GeoAddress end

  *destination address*
- final String errorTitle

  *if routes cannot be calculated this field will have error title*
- final String errorDescription

  *if routes cannot be calculated this field will have error description*

### 0.6.38.2 Member Data Documentation

#### 0.6.38.2.1 candidates `final ArrayList<Candidate> candidates = new ArrayList<>()`

List of all route candidates.

Best candidate is first on the list

#### 0.6.38.2.2 end `final GeoAddress end`

destination address

#### 0.6.38.2.3 errorDescription `final String errorDescription`

if routes cannot be calculated this field will have error description

#### 0.6.38.2.4 errorTitle `final String errorTitle`

if routes cannot be calculated this field will have error title

#### 0.6.38.2.5 start `final GeoAddress start`

starting address

## 0.6.39 EasyAPI.RouteCandidatesResult Interface Reference

### 0.6.39.1 Description

Callback interface for route candidates results.

**Public Member Functions**

- void onRouteCandidatesResult (int status, RouteCandidates candidates)

  *Called to return route candidates back to the application.*

### 0.6.39.2 Member Function Documentation

#### 0.6.39.2.1 onRouteCandidatesResult()

```
void onRouteCandidatesResult (
        int status,
        RouteCandidates candidates )
```

Called to return route candidates back to the application.

Candidates will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *candidates* | Calculated route candidates. |

## 0.6.40 EasyAPI.RouteListResult Interface Reference

### 0.6.40.1 Description

Callback interface for list of route results.

**Public Member Functions**

- void onRoutes (int status, List< Route > route)

  *Called to return route list to the application.*

#### 0.6.40.2 Member Function Documentation

##### 0.6.40.2.1 onRoutes()

```
void onRoutes (
        int status,
        List< Route > route )
```

Called to return route list to the application.

List will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *route* | route list. |

### 0.6.41 EasyAPI.RouteResult Interface Reference

#### 0.6.41.1 Description

Callback interface for route results.

**Public Member Functions**

- void onRouteResult (int status, Route route)

  *Called to return route back to the application.*

#### 0.6.41.2 Member Function Documentation

##### 0.6.41.2.1 onRouteResult()

```
void onRouteResult (
        int status,
        Route route )
```

Called to return route back to the application.

Route will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *route* | Calculated route. |

## 0.6.42 EasyAPI.RouteSettingsResult Interface Reference

### 0.6.42.1 Description

Callback interface for getRouteSettings.

**Public Member Functions**

- void onRouteSettings (int status, int route_type, int avoid_mask)

  *Called after application has executed the call.*

### 0.6.42.2 Member Function Documentation

#### 0.6.42.2.1 onRouteSettings()

```
void onRouteSettings (
        int status,
        int route_type,
        int avoid_mask )
```

Called after application has executed the call.

Arguments will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *route_type* | Route type (currently, only quickest is supported). |
| *avoid_mask* | Mask of route avoidance flags ( |

**See also**

Enums::EAvoidMask).

## 0.6.43 Enums.SavedPlaceType Enum Reference

### 0.6.43.1 Description

Enumeration for saved places.

**Public Attributes**

- favorites =(API.SAVED_DEST_FAVORITE)

  *Favorite places.*
- recents =(API.SAVED_DEST_RECENT)

  *Recently visited places.*
- home =(API.HOME)

  *Home.*
- work =(API.WORK)

  *Home.*
- both =(API.SAVED_DEST_ALL)

  *All saved places.*

### 0.6.43.2 Member Data Documentation

#### 0.6.43.2.1 both `both =(API.SAVED_DEST_ALL)`

All saved places.

#### 0.6.43.2.2 favorites `favorites =(API.SAVED_DEST_FAVORITE)`

Favorite places.

#### 0.6.43.2.3 home `home =(API.HOME)`

Home.

#### 0.6.43.2.4 recents `recents =(API.SAVED_DEST_RECENT)`

Recently visited places.

#### 0.6.43.2.5 work `work =(API.WORK)`

Home.

## 0.6.44 SavedRouteChange Class Reference

### 0.6.44.1 Description

Encapsulate change of saved place.

See INotificationListener3::OnSavedPlacesChanged.

#### Public Attributes

- final int itemId

  *Saved route identifier.*
- final int changeType

  *Change type.*

#### Static Public Attributes

- static final int SAVE_ROUTE = 1

  *the route is saved to favorite routes*
- static final int REMOVE_ROUTE = ∼1

  *the route is removed from favorite routes*

### 0.6.44.2 Member Data Documentation

#### 0.6.44.2.1 changeType `final int changeType`

Change type.

#### 0.6.44.2.2 itemId `final int itemId`

Saved route identifier.

#### 0.6.44.2.3 REMOVE_ROUTE `final int REMOVE_ROUTE = ∼1 [static]`

the route is removed from favorite routes

#### 0.6.44.2.4 SAVE_ROUTE `final int SAVE_ROUTE = 1 [static]`

the route is saved to favorite routes

## 0.6.45 ServiceLink Class Reference

### 0.6.45.1 Description

ServiceLink is used to send API request using Messenger Service.

Inherits APILink.

#### Public Member Functions

- void close ()

  *Unbind from remote server.*
- void open ()

  *Bind to remote service.*

### 0.6.45.2 Member Function Documentation

#### 0.6.45.2.1 close() `void close ( )`

Unbind from remote server.

#### 0.6.45.2.2 open() `void open ( )`

Bind to remote service.

## 0.6.46 EasyAPI.ShowTrafficResult Interface Reference

### 0.6.46.1 Description

Callback interface for showTrafficSettings.

**See also**

EasyAPI::showTrafficSettings(ShowTrafficResult)

#### Public Member Functions

- void onShowTrafficStatus (int status, boolean flow, boolean events)

  *Called after application has executed the call.*

### 0.6.46.2 Member Function Documentation

#### 0.6.46.2.1 onShowTrafficStatus()

```
void onShowTrafficStatus (
        int status,
        boolean flow,
        boolean events )
```

Called after application has executed the call.

Arguments will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *flow* | true if traffic flow display is enabled. |
| *events* | true if traffic events display are enabled. |

## 0.6.47 SpeedCamera Class Reference

### 0.6.47.1 Description

Speed camera data.

**Static Public Member Functions**

- static SpeedCamera fromJSON (JSONObject o)

  *Convert form JSON.*

**Public Attributes**

- final int LimitKmh

  *Speed limit ink km/h.*
- final int DistanceMeters

  *Distance to camera in meters.*

### 0.6.47.2 Member Function Documentation

#### 0.6.47.2.1 fromJSON()

```
static SpeedCamera fromJSON (
        JSONObject o )  [static]
```

Convert form JSON.

**Parameters**

| | |
|---|---|
| *o* | JSONObject |

**Returns**

SpeedCamera instance

### 0.6.47.3 Member Data Documentation

#### 0.6.47.3.1 DistanceMeters `final int DistanceMeters`

Distance to camera in meters.

#### 0.6.47.3.2 LimitKmh `final int LimitKmh`

Speed limit ink km/h.

## 0.6.48 GeoAddress.SphereUtils Class Reference

### 0.6.48.1 Description

Utilities for conversion between internal coordinate representation and WGS84 coordinates.

## 0.6.49 EasyAPI.StringArrayResult Interface Reference

### 0.6.49.1 Description

Callback interface for EasyAPI calls that return string array.

**Public Member Functions**

- void onResult (int status, String[ ] array)

  *Called after application has executed the call.*

### 0.6.49.2 Member Function Documentation

#### 0.6.49.2.1 onResult()

```
void onResult (
        int status,
        String[ ] array )
```

Called after application has executed the call.

Array of strings will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *array* | result of the call. |

## 0.6.50 EasyAPI.StringResult Interface Reference

### 0.6.50.1 Description

Callback interface for EasyAPI calls that return string values.

**Public Member Functions**

- void onResult (int status, String value)

  *Called after application has executed the call.*

### 0.6.50.2 Member Function Documentation

#### 0.6.50.2.1 onResult()

```
void onResult (
    int status,
    String value )
```

Called after application has executed the call.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *value* | Result value. It is only valid of status is API::RESULT_OK. |

## 0.6.51 Route.Subtrip Class Reference

### 0.6.51.1 Description

Route subtrip ("leg") data.

**Public Attributes**

- int distanceMeters

  *Subtrip length in meters.*
- int durationSeconds

  *Subtrip duration in seconds.*
- ArrayList< Advice > advices

  *Optional list of instructions.*

### 0.6.51.2 Member Data Documentation

#### 0.6.51.2.1 advices `ArrayList<Advice> advices`

Optional list of instructions.

Collectiong instructions is an expensive operation, not all API calls fill this list.

#### 0.6.51.2.2 distanceMeters `int distanceMeters`

Subtrip length in meters.

#### 0.6.51.2.3 durationSeconds `int durationSeconds`

Subtrip duration in seconds.

## 0.6.52 TrafficStatus Class Reference

### 0.6.52.1 Description

Class that represents current live-traffic status.

**Public Attributes**

- final boolean installed

  *true if live-traffic is installed*
- final boolean enabled

  *true if live-traffic is enabled in settings*
- final boolean internet_available

  *true if internet connection is available*
- final int traffic_state

  *live traffic fetch status - one of the UNKNOWN, PENDING, TRAFFIC_OK, UNAVAILABLE*
- final int delay_seconds

  *current live traffic delay in seconds or -1 if road is blocked*
- final boolean road_blocked

  *true road ahead is blocked*

### 0.6.52.2 Member Data Documentation

#### 0.6.52.2.1 delay_seconds `final int delay_seconds`

current live traffic delay in seconds or -1 if road is blocked

#### 0.6.52.2.2 enabled `final boolean enabled`

true if live-traffic is enabled in settings

#### 0.6.52.2.3 installed `final boolean installed`

true if live-traffic is installed

#### 0.6.52.2.4 internet_available `final boolean internet_available`

true if internet connection is available

#### 0.6.52.2.5 road_blocked `final boolean road_blocked`

true road ahead is blocked

#### 0.6.52.2.6 traffic_state `final int traffic_state`

live traffic fetch status - one of the UNKNOWN, PENDING, TRAFFIC_OK, UNAVAILABLE

## 0.6.53 EasyAPI.TrafficStatusResult Interface Reference

### 0.6.53.1 Description

Callback interface for trafficStatus.

**See also**

EasyAPI::trafficStatus(TrafficStatusResult)

**Public Member Functions**

- void onTrafficStatus (int status, TrafficStatus trafficStatus)
  *Called after application has executed the call.*

### 0.6.53.2 Member Function Documentation

#### 0.6.53.2.1 onTrafficStatus()

```
void onTrafficStatus (
        int status,
        TrafficStatus trafficStatus )
```

Called after application has executed the call.

Traffic status value will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *trafficStatus* | traffic status. May be null if error occurred. |

## 0.6.54 Enums.TypeFamily Enum Reference

### 0.6.54.1 Description

POI types.

**Deprecated**

## 0.6.55 VersionInfo Class Reference

### 0.6.55.1 Description

Version info.

**See also**

EasyAPI::getVersionEx(EasyAPI.VersionResult)

**Public Member Functions**

- String formatAppVersion (boolean hexBuildNumber)

  *Returns formatted appVersion together with build number.*

**Static Public Member Functions**

- static VersionInfo fromJSON (JSONObject o)

  *Make VersionInfo object from JSON.*

**Public Attributes**

- final String appVersion

  *Application version.*
- final String mapVersion

  *Maps version.*
- final String buildNumber

  *Application's build number.*
- final String serial

  *Device's serial number.*

### 0.6.55.2 Member Function Documentation

#### 0.6.55.2.1 formatAppVersion()

```
String formatAppVersion (
        boolean hexBuildNumber )
```

Returns formatted appVersion together with build number.

**Parameters**

| | |
|---|---|
| *hexBuildNumber* | include build number as decimal/hexadecimal value |

**Returns**

formatted appVersion

#### 0.6.55.2.2 fromJSON() `static VersionInfo fromJSON ( JSONObject o ) [static]`

Make VersionInfo object from JSON.

**Parameters**

| | |
|---|---|
| *o* | JSON object |

**Returns**

version info

### 0.6.55.3 Member Data Documentation

#### 0.6.55.3.1 appVersion `final String appVersion`

Application version.

#### 0.6.55.3.2 buildNumber `final String buildNumber`

Application's build number.

#### 0.6.55.3.3 mapVersion `final String mapVersion`

Maps version.

#### 0.6.55.3.4 serial `final String serial`

Device's serial number.

## 0.6.56 EasyAPI.VersionResult Interface Reference

### 0.6.56.1 Description

Callback interface for getVersionEx.

**See also**

EasyAPI::getVersionEx(VersionResult)

VersionInfo

#### Public Member Functions

- void onVersion (int status, VersionInfo versionInfo)

  *Called after application has executed the call.*

### 0.6.56.2 Member Function Documentation

#### 0.6.56.2.1 onVersion()

```
void onVersion (
    int status,
    VersionInfo versionInfo )
```

Called after application has executed the call.

Version info will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *versionInfo* | Version data. |

## 0.6.57 Voice Class Reference

### 0.6.57.1 Description

Guidance voice information.

#### Public Attributes

- final String id

  *Voice ID.*
- final String name

  *Voice name.*
- final String language

  *Voice language.*
- final boolean selected

  *Is this voice currently selected.*

#### 0.6.57.2 Member Data Documentation

##### 0.6.57.2.1 id `final String id`

Voice ID.

##### 0.6.57.2.2 language `final String language`

Voice language.

##### 0.6.57.2.3 name `final String name`

Voice name.

##### 0.6.57.2.4 selected `final boolean selected`

Is this voice currently selected.

### 0.6.58 EasyAPI.VoiceListResult Interface Reference

#### 0.6.58.1 Description

Callback interface for getInstalledVoices EasyAPI call.

**Public Member Functions**

- void onVoiceList (int status, List< Voice > voices)

    *Called after application has executed the call.*

#### 0.6.58.2 Member Function Documentation

##### 0.6.58.2.1 onVoiceList()

```
void onVoiceList (
    int status,
    List< Voice > voices )
```

Called after application has executed the call.

List of voices will be valid only if status is API::RESULT_OK.

**Parameters**

| | |
|---|---|
| *status* | Status of the API call. |
| *voices* | List of installed voices. |

## 0.7 Example Documentation

### 0.7.1 GeoFencesFragment.java

Using Geofences API

```
package hr.mireo.arthur.apidemo;
import android.annotation.SuppressLint;
import android.content.Context;
import android.location.Location;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import java.util.ArrayList;
import java.util.Random;
import hr.mireo.arthur.api.API;
import hr.mireo.arthur.api.EasyAPI;
import static hr.mireo.arthur.api.API.RESULT_OK;
public class GeoFencesFragment extends Fragment {
    private Adapter mAdapter;
    private TextView mEmptyView;
    static class Adapter extends ArrayAdapter<String> {
        private class Holder  {
            String text;
            TextView line1;
            ImageButton remove;
        }
        public Adapter(@NonNull Context context, int resource) {
            super(context, resource);
        }
        @NonNull
        @Override
        public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
            String name = getItem(position);
            Holder holder;
            if (convertView == null) {
                convertView = LayoutInflater.from(getContext()).inflate(R.layout.geofence_item, parent,
       false);
                holder = new Holder();
                holder.text = name;
                holder.line1 = convertView.findViewById(R.id.gfName);
                holder.remove = convertView.findViewById(R.id.gfRemove);
                holder.remove.setOnClickListener(v -> MainActivity.mNavigation.removeGeoFence(holder.text,
       (s) -> {
                    if (s == API.RESULT_OK) {
                        Toast.makeText(getContext(), "Geofence " + holder.text + " removed",
       Toast.LENGTH_SHORT).show();
                        remove(holder.text);
                        notifyDataSetChanged();
                    }
                    else
                        Toast.makeText(getContext(), "Error removing geofence: " + EasyAPI.statusMessage(s),
       Toast.LENGTH_SHORT).show();
                }));
                convertView.setTag(holder);
                convertView.setOnClickListener(v -> MainActivity.mNavigation.showGeoFence(name, true, status
       -> {}));
            }
            else {
                holder = (Holder) convertView.getTag();
            }
            holder.text = name;
            holder.line1.setText(name);
            return convertView;
        }
    }
    @SuppressLint("ShowToast")
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.geofences, container, false);
        mAdapter = new Adapter(this.getContext(), R.layout.geofence_item);
        ListView listView = rootView.findViewById(R.id.gflList);
        listView.setAdapter(mAdapter);
        listView.setEmptyView(mEmptyView = rootView.findViewById(R.id.gfEmpty));
        MainActivity.mNavigation.listGeoFences((status, list) -> {
            if (status == RESULT_OK)
                mAdapter.addAll(list);
            else {
                mEmptyView.setText(R.string.api_not_supported);
                mAdapter.notifyDataSetChanged();
            }
        });
        Button btn = rootView.findViewById(R.id.gflAddGeofence);
        btn.setOnClickListener(v -> addNewGeofence());
        return rootView;
    }
    static Random rnd = new Random();
    private void addNewGeofence() {
       MainActivity.mNavigation.getCurrentAddress((status, address) -> {
           if (status != RESULT_OK) {
               Toast.makeText(getContext(), "Cannot get current GPS position to make geo fence on",
       Toast.LENGTH_LONG).show();
               return;
           }
           // make random ellipses near current location
           double a = (rnd.nextDouble() + 0.5) * 10.0 / 6700.0;
           double b = (rnd.nextDouble() + 0.5) * 10.0 / 6700.0;
           double x = (rnd.nextDouble() - 0.5) * 100.0 / 6700.0;
           double y = (rnd.nextDouble() - 0.5) * 100.0 / 6700.0;
           ArrayList<Location> locations = new ArrayList<>();
           for (int i = 0; i < 360; ++i) {
                Location l = new Location("API");
                l.setLongitude(address.longitude + x + a * Math.cos(i/(2 * Math.PI)));
                l.setLatitude(address.latitude + y + b * Math.sin(i/(2 * Math.PI)));
                locations.add(l);
           }
           int color = EasyAPI.rgba(rnd.nextInt(255), rnd.nextInt(255), rnd.nextInt(255), 128);
           String gfName = "GF - " + System.currentTimeMillis();
           MainActivity.mNavigation.addGeoFence(gfName, locations, color, status1 -> {
                if (status1 == RESULT_OK) {
                    mAdapter.add(gfName);
                    mAdapter.notifyDataSetChanged();
                }
                else
                    Toast.makeText(getContext(), "geo fence error", Toast.LENGTH_SHORT).show();
          });
       });
    }
}
```

## 0.7.2 MainActivity.java

Example activity with couple of fragments to showcase some of the API functionality.EasyAPI instance is initialized here and it is shared with fragments.

```
package hr.mireo.arthur.apidemo;
import android.app.ActionBar;
import android.app.ActionBar.Tab;
import android.content.Context;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentActivity;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentStatePagerAdapter;
import androidx.viewpager.widget.ViewPager;
import android.view.inputmethod.InputMethodManager;
import android.widget.Toast;
import hr.mireo.arthur.api.EasyAPI;
public class MainActivity extends FragmentActivity {
    static EasyAPI mNavigation;
    MyPagerAdapter mPagerAdapter;
```

```
ViewPager mViewPager;
ApiBroadcastReceiver mBroadcastReceiver;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mNavigation = new EasyAPI(this.getText(R.string.url_scheme).toString(), this);
    mNavigation.setOnLinkDied((link)-> {
        Toast.makeText(MainActivity.this, "API link has died - reconnecting",
   Toast.LENGTH_SHORT).show();
        link.open();
    });
    registerApiReceiver();
    setContentView(hr.mireo.arthur.apidemo.R.layout.main_layout);
    mPagerAdapter = new MyPagerAdapter(getSupportFragmentManager());
    mViewPager = findViewById(R.id.pager);
    mViewPager.setAdapter(mPagerAdapter);
    final ActionBar actionBar = getActionBar();
    actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
    // Specify that tabs should be displayed in the action bar.
    actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
    mViewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
        @Override
        public void onPageSelected(int position) {
            final InputMethodManager imm = (InputMethodManager) getSystemService(
                    Context.INPUT_METHOD_SERVICE);
            imm.hideSoftInputFromWindow(mViewPager.getWindowToken(), 0);
            actionBar.setSelectedNavigationItem(position);
        }
        @Override
        public void onPageScrolled(int position, float offset, int offsetPixels) {
        }
        @Override
        public void onPageScrollStateChanged(int state) {
        }
    });
    // Create a tab listener that is called when the user changes tabs.
    ActionBar.TabListener tabListener = new ActionBar.TabListener() {
        @Override
        public void onTabReselected(Tab tab,
                                    android.app.FragmentTransaction ft) {
            //mViewPager.setCurrentItem(tab.getPosition());
        }
        @Override
        public void onTabSelected(Tab tab,
                                  android.app.FragmentTransaction ft) {
            mViewPager.setCurrentItem(tab.getPosition());
        }
        @Override
        public void onTabUnselected(Tab tab,
                                    android.app.FragmentTransaction ft) {
            // TODO Auto-generated method stub
        }
    };
    actionBar.addTab(actionBar.newTab()
            .setText("Map")
            .setTabListener(tabListener));
    actionBar.addTab(actionBar.newTab()
            .setText("Search")
            .setTabListener(tabListener));
    actionBar.addTab(actionBar.newTab()
            .setText("Route Planning")
            .setTabListener(tabListener));
    actionBar.addTab(actionBar.newTab()
            .setText("Favorites")
            .setTabListener(tabListener));
    actionBar.addTab(actionBar.newTab()
            .setText("Saved Routes")
            .setTabListener(tabListener));
    actionBar.addTab(actionBar.newTab()
            .setText("Settings")
            .setTabListener(tabListener));
    actionBar.addTab(actionBar.newTab()
            .setText("Notifications")
            .setTabListener(tabListener));
    actionBar.addTab(actionBar.newTab()
            .setText("Geo fences")
            .setTabListener(tabListener));
}
private void registerApiReceiver() {
    mBroadcastReceiver = new ApiBroadcastReceiver();
    registerReceiver(mBroadcastReceiver, mBroadcastReceiver.getFilter());
}
// Since this is an object collection, use a FragmentStatePagerAdapter,
// and NOT a FragmentPagerAdapter.
public static class MyPagerAdapter extends FragmentStatePagerAdapter {
    MyPagerAdapter(FragmentManager fm) {
        super(fm, BEHAVIOR_RESUME_ONLY_CURRENT_FRAGMENT);
        }
        @NonNull
        @Override
        public Fragment getItem(int i) {
            switch (i) {
                case 1:
                    return new SearchFragment();
                case 2:
                    return new RouteFragment();
                case 3:
                    return new SavedPlacesFragment();
                case 4:
                    return new SavedRoutesFragment();
                case 5:
                    return new SettingsFragment();
                case 6:
                    return new NotificationsFragment();
                case 7:
                    return new GeoFencesFragment();
            }
            return new MapFragment();
        }
        @Override
        public int getCount() {
            return 8;
        }
        @Override
        public CharSequence getPageTitle(int position) {
            switch (position) {
                case 1:
                    return "Search";
                case 2:
                    return "Route Planning";
                case 3:
                    return "Saved Places";
                case 4:
                    return "Saved Routes";
                case 5:
                    return "Settings";
                case 6:
                    return "Notifications";
            }
            return "Show";
        }
    }
    @Override
    public void onDestroy() {
        unregisterReceiver(mBroadcastReceiver);
        mNavigation.destroy();
        super.onDestroy();
    }
}
```

### 0.7.3 MapFragment.java

This fragment show some basic API usage: retrieve application version, show main application screen, zoom in/out.

```
package hr.mireo.arthur.apidemo;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
public class MapFragment extends Fragment {
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        final View rootView = inflater.inflate(hr.mireo.arthur.apidemo.R.layout.map_layout, container,
       false);
        Button show_map = rootView.findViewById(R.id.showMap);
        show_map.setOnClickListener(v -> MainActivity.mNavigation.showMap(null));
        Button zoom = rootView.findViewById(R.id.zoom);
        zoom.setOnClickListener(v -> {
            float zoom_factor;
            String s = ((EditText) rootView.findViewById(R.id.zoomFactor)).getText().toString();
            try {
                zoom_factor = Float.parseFloat(s);
            } catch (NumberFormatException e) {
                return;
            }
            if (zoom_factor != 0) {
                MainActivity.mNavigation.mapSmartZoom(zoom_factor, true, null);
            }
        });
        MainActivity.mNavigation.getGpsPosition((i, b, b1, longitude, latitude) -> ((TextView)
       rootView.findViewById(R.id.gpsPosition))
                .setText(String.format(getString(R.string.gps_label), latitude,longitude)));
        MainActivity.mNavigation.getVersionEx((i, version) -> {
            if (version != null) {
                ((TextView)rootView.findViewById(R.id.version)).setText(version.appVersion);
                ((TextView)rootView.findViewById(R.id.map_version)).setText(version.mapVersion);
            }
            else {
                ((TextView)rootView.findViewById(R.id.version)).setText(R.string.unknown);
                ((TextView)rootView.findViewById(R.id.map_version)).setText(R.string.unknown);
            }
        });
        return rootView;
    }
}
```

### 0.7.4 NotificationsFragment.java

Using Notifications API

```
package hr.mireo.arthur.apidemo;
import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import android.text.Html;
import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import java.util.Arrays;
import java.util.List;
import java.util.Locale;
import hr.mireo.arthur.api.API;
import hr.mireo.arthur.api.Advice;
import hr.mireo.arthur.api.EasyAPI;
import hr.mireo.arthur.api.INotificationListener3;
import hr.mireo.arthur.api.PlaceChange;
import hr.mireo.arthur.api.PositionData;
import hr.mireo.arthur.api.Route;
import hr.mireo.arthur.api.SavedRouteChange;
public class NotificationsFragment extends Fragment {
    private TextView mMaskView;
    private Adapter mAdapter;
    private static class NameValue  {
        final String name;
        String value;
        private NameValue(String name, String value) {
            this.name = name;
            this.value = value;
        }
    }
    static class Adapter extends ArrayAdapter<NameValue> {
        private static class Holder  {
            TextView nameText;
            TextView valueText;
        }
        public Adapter(@NonNull Context context, int resource, List<NameValue> data) {
            super(context, resource, data);
        }
        @NonNull
        @Override
        public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
            NameValue nv = getItem(position);
            Holder holder;
            if (convertView == null) {
                convertView = LayoutInflater.from(getContext()).inflate(R.layout.nv_item, parent, false);
                holder = new Holder();
                holder.nameText = convertView.findViewById(R.id.nvName);
                holder.valueText = convertView.findViewById(R.id.nvValue);
                convertView.setTag(holder);
            }
            else {
                holder = (Holder) convertView.getTag();
            }
            holder.nameText.setText(nv.name);
            if (nv.value.contains("</") && Build.VERSION.SDK_INT >= Build.VERSION_CODES.N) {
                holder.valueText.setText(Html.fromHtml(nv.value, Html.FROM_HTML_MODE_LEGACY));
            }
            else
                holder.valueText.setText(nv.value);
            return convertView;
        }
    }
    static final List<NameValue> data = Arrays.asList(
            new NameValue("Routing", ""),
            new NameValue("Advices", ""),
            new NameValue("Speed limits", ""),
            new NameValue("Speed cameras", ""),
            new NameValue("Position data", ""),
            new NameValue("Speed violations", ""),
            new NameValue("Geo Fences", ""),
            new NameValue("Favorites", ""),
            new NameValue("Routes", ""));
    static class NotificationListener implements INotificationListener3 {
        private Adapter mAdapter;
        private Advice lastAdvice;
        private void sendNotification() {
            if (mAdapter != null)
                mAdapter.notifyDataSetInvalidated();
        }
        NotificationListener() {}
        public void setAdapter(Adapter adapter) {
            this.mAdapter = adapter;
        }
        @Override
        public void OnGeoFenceEvent(@NonNull String geoFenceName, int evt, PositionData pd) {
            data.get(6).value = geoFenceEvent(evt) + " " + geoFenceName;
            sendNotification();
        }
        @Override
        public void OnSavedRouteChanged(@NonNull List<SavedRouteChange> changes) {
            StringBuilder builder = new StringBuilder();
            builder.append("# of changes: ").append(changes.size()).append("\n");
            for (SavedRouteChange pc: changes) {
                builder.append(pc.itemId).append(" ").append(routeChangeType(pc.changeType)).append("\n");
            }
            data.get(8).value = builder.toString();
            sendNotification();
        }
        @Override
        public void OnPlaceChanged(@NonNull List<PlaceChange> changes) {
            StringBuilder builder = new StringBuilder();
            builder.append("# of changes: ").append(changes.size()).append("\n");
            for (PlaceChange pc: changes) {
                builder.append(pc.place.formatted[0]).append("
       ").append(changeType(pc.changeType)).append("\n");
            }
            data.get(7).value = builder.toString();
            sendNotification();
        }
        @Override
        public void OnPositionData(PositionData positionData) {
            if (positionData.longitude > 0)
                if (positionData.latitude > 0)
                    data.get(4).value = String.format(Locale.US, "%fE %fN", positionData.longitude,
       positionData.latitude);
                else
                    data.get(4).value = String.format(Locale.US, "%fE %fS", positionData.longitude,
       -positionData.latitude);
            else
            if (positionData.latitude > 0)
                data.get(4).value = String.format(Locale.US, "%fW %fN", -positionData.longitude,
       positionData.latitude);
            else
                data.get(4).value = String.format(Locale.US, "%fW %fS", -positionData.longitude,
       -positionData.latitude);
            sendNotification();
        }
        @Override
        public void OnSpeedViolations(int limit, int speed) {
            data.get(5).value = "limit: " + limit + " km/h speed: " + speed + " km/h";
            sendNotification();
        }
        @Override
        public void OnRoute(Route rt) {
            data.get(0).value = "Total route length:" + rt.totalMeters / 1000.0 + " km";
            sendNotification();
        }
        @Override
        public void OnAdvice(Advice adv) {
            data.get(1).value = adv.adviceText;
            sendNotification();
        }
        @Override
        public void OnSpeedLimit(int limit_kmh) {
            data.get(2).value = "limit: " + limit_kmh + " km/h";
            sendNotification();
        }
        @Override
        public void OnSpeedCamera(int limit_kmh, int distance_to_camera_in_meters) {
            data.get(3).value = "distance  to camera " + distance_to_camera_in_meters + "m\nlimit: " +
       limit_kmh + " km/h";
            sendNotification();
        }
    };
    private final static NotificationListener notificationListener = new NotificationListener();
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        final View rootView = inflater.inflate(hr.mireo.arthur.apidemo.R.layout.notifications_layout,
       container, false);
        ListView listView = rootView.findViewById(R.id.notifications);
        mAdapter = new Adapter(this.getContext(), R.layout.nv_item, data);
        listView.setAdapter(mAdapter);
        mMaskView = rootView.findViewById(R.id.etMask);
        rootView.findViewById(R.id.maskHelp).setOnClickListener(v -> Toast.makeText(getActivity(),
            "Routing: 0x1\n" +
            "Advices: 0x2\n" +
            "Speed Limits: 0x4\n" +
            "Speed Cameras: 0x8\n" +
            "Position Data: 0x10\n" +
            "Speed Violations: 0x20\n" +
            "Geo Fences: 0x40\n" +
            "Saved places/routes: 0x80",
        Toast.LENGTH_LONG).show());
        notificationListener.setAdapter(mAdapter);
        Button requestNotifications = rootView.findViewById(R.id.requestNotifications);
        requestNotifications.setOnClickListener(v -> MainActivity.mNavigation.requestNotifications(
                "apiDemo",
                getNotificationMask(),
                notificationListener, (status, value ) -> {
                    if (status == API.RESULT_OK)
                        Toast.makeText(getContext(), "Waiting for notifications...",
       Toast.LENGTH_LONG).show();
                    else
                        Toast.makeText(getContext(), "Error: " + EasyAPI.statusMessage(status),
       Toast.LENGTH_LONG).show();
                }));
        Button removeNotification = rootView.findViewById(R.id.removeNotifications);
        removeNotification.setOnClickListener(v -> MainActivity.mNavigation.removeNotifications(
                "apiDemo",
                getNotificationMask(),
                notificationListener, (s) -> {
                    for (NameValue d: data)
                        d.value = "";
                    mAdapter.notifyDataSetChanged();
                }
        ));
        return rootView;
    }
    @Override
    public void onStop() {
        notificationListener.setAdapter(null);
        super.onStop();
    }
    private static String geoFenceEvent(int evt) {
        return evt == API.GEOFENCE_ENTER ? "Enter": "Exit";
    }
    private static String changeType(int changeType) {
        switch (changeType) {
            case PlaceChange.SET_AS_FAVORITE: return  "place added to favorites";
            case PlaceChange.REMOVE_FAVORITE: return  "place removed from favorites";
            case PlaceChange.SET_AS_RECENT: return  "place added to recents";
            case PlaceChange.REMOVE_RECENT: return  "place removed from recents";
            case PlaceChange.SET_HOME: return  "home place set";
            case PlaceChange.REMOVE_HOME: return  "home place cleared";
            case PlaceChange.SET_WORK: return  "work place set";
            case PlaceChange.REMOVE_WORK: return  "work place cleared";
        }
        return "";
    }
    private static String routeChangeType(int changeType) {
        switch (changeType) {
            case SavedRouteChange.SAVE_ROUTE: return  "route saved to favorites";
            case SavedRouteChange.REMOVE_ROUTE: return  "route removed from favorites";
        }
        return "";
    }
    int getNotificationMask() {
        if (TextUtils.isEmpty(mMaskView.getText()))
            return API.NOTIFY_ALL;
        return Integer.parseInt(String.valueOf(mMaskView.getText()));
    }
}
```

### 0.7.5 RouteFragment.java

Using Route Planning API

```
package hr.mireo.arthur.apidemo;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import hr.mireo.arthur.api.API;
import hr.mireo.arthur.api.EasyAPI;
import hr.mireo.arthur.api.GeoAddress;
import hr.mireo.arthur.api.Route;
public class RouteFragment extends Fragment implements OnClickListener {
    private GeoAddress ga;
    private EditText mLongitude;
    private EditText mLatitude;
    private TextView mRouteStatus;
    TextWatcher mTextWatcher = new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {
        }
        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
        }
        @Override
        public void afterTextChanged(Editable s) {
            updateLocation();
        }
    };
    private double getLongitude() {
        String longitude = mLongitude.getText().toString();
        try {
            return Double.parseDouble(longitude);
        } catch (NumberFormatException ex) {
            return 0.0;
        }
    }
    private double getLatitude() {
        String lat = mLatitude.getText().toString();
        try {
            return Double.parseDouble(lat);
        } catch (NumberFormatException ex) {
            return 0.0;
        }
    }
    private void updateLocation() {
        double lat = getLatitude();
        double lon = getLongitude();
        if (lat != 0 && lon != 0)
            ga = GeoAddress.fromLonLat(lon, lat);
        else
            ga = null;
    }
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(hr.mireo.arthur.apidemo.R.layout.route_layout, container, false);
        mLongitude = rootView.findViewById(R.id.longitude);
        mLatitude = rootView.findViewById(R.id.latitude);
        mLongitude.addTextChangedListener(mTextWatcher);
        mLatitude.addTextChangedListener(mTextWatcher);
        mRouteStatus = rootView.findViewById(R.id.routeStatusOutput);
        rootView.findViewById(R.id.startRoute).setOnClickListener(this);
        rootView.findViewById(R.id.addDestination).setOnClickListener(this);
        rootView.findViewById(R.id.clearRoute).setOnClickListener(this);
        rootView.findViewById(R.id.startNavigation).setOnClickListener(this);
        rootView.findViewById(R.id.repeatGuidance).setOnClickListener(this);
        rootView.findViewById(R.id.endNavigation).setOnClickListener(this);
        rootView.findViewById(R.id.routeStatus).setOnClickListener(this);
        rootView.findViewById(R.id.saveCurrentRoute).setOnClickListener(this);
        return rootView;
    }
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case hr.mireo.arthur.apidemo.R.id.startRoute:
                if (ga == null) break;
                MainActivity.mNavigation.startRoute(ga, null);
                break;
            case hr.mireo.arthur.apidemo.R.id.addDestination:
                if (ga == null) break;
                MainActivity.mNavigation.addDestination(ga, null);
                break;
            case hr.mireo.arthur.apidemo.R.id.clearRoute:
                MainActivity.mNavigation.clearRoute(null);
                break;
            case hr.mireo.arthur.apidemo.R.id.startNavigation:
                MainActivity.mNavigation.startNavigation(null);
                break;
            case hr.mireo.arthur.apidemo.R.id.repeatGuidance:
                MainActivity.mNavigation.repeatGuidance(null);
                break;
            case hr.mireo.arthur.apidemo.R.id.endNavigation:
                MainActivity.mNavigation.endNavigation(null);
                break;
            case R.id.saveCurrentRoute:
                MainActivity.mNavigation.saveCurrentRoute("kind=business", (status, route) -> {
                    if (status == API.RESULT_OK)
                        Toast.makeText(getContext(), "Route saved", Toast.LENGTH_LONG).show();
                    else
                        Toast.makeText(getContext(), "Error saving route: " + EasyAPI.statusMessage(status),
       Toast.LENGTH_LONG).show();
                });
                break;
            case hr.mireo.arthur.apidemo.R.id.routeStatus:
                mRouteStatus.setText("");
                MainActivity.mNavigation.getNextDestination(new
       RouteStatusListener("get_next_destination"));
                MainActivity.mNavigation.getFinalDestination(new
       RouteStatusListener("get_final_destination"));
                MainActivity.mNavigation.getDtg(new IntStatusListener("DTG"));
                MainActivity.mNavigation.getTtg(new IntStatusListener("TTG"));
                MainActivity.mNavigation.getWholeDtg(new IntStatusListener("Whole DTG"));
                MainActivity.mNavigation.getWholeTtg(new IntStatusListener("Whole TTG"));
                break;
        }
    }
    class RouteStatusListener implements EasyAPI.AddressResult {
        private final String action;
        RouteStatusListener(String action) {
            this.action = action;
        }
        @Override
        public void onAddress(int i, GeoAddress geoAddress) {
            mRouteStatus.append(action + "\n");
            mRouteStatus.append(EasyAPI.statusMessage(i) +  "\n\n");
        }
    }
    class IntStatusListener implements  EasyAPI.IntResult  {
        private final String action;
        IntStatusListener(String action) {
            this.action = action;
        }
        @Override
        public void onResult(int i, int i1) {
            mRouteStatus.append(action + "\n");
            mRouteStatus.append(EasyAPI.statusMessage(i) +  "\n\n");
        }
    }
}
```

## 0.7.6 SavedPlacesFragment.java

Using Favorite Places API

```java
package hr.mireo.arthur.apidemo;
import android.annotation.SuppressLint;
import android.content.Context;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import hr.mireo.arthur.api.API;
import hr.mireo.arthur.api.Enums;
import hr.mireo.arthur.api.GeoAddress;
public class SavedPlacesFragment extends Fragment {
    private AddressListAdapter mAddressListAdapter;
    private TextView mEmptyView;
    static class AddressListAdapter extends ArrayAdapter<GeoAddress> {
        private class Holder implements OnClickListener {
            GeoAddress ga;
            TextView line1;
            TextView line2;
            ImageButton show;
            ImageButton navigate;
            ImageButton removeFromFavorites;
            @Override
            public void onClick(View v) {
                if (v == show)
                    MainActivity.mNavigation.showOnMap(ga, (status)-> {});
                else if (v == navigate)
                     MainActivity.mNavigation.navigateTo(ga, true, (status, address) -> {
                        if (status == API.RESULT_OK) {
                            Toast.makeText(getContext(), "Navigation succeeeded", Toast.LENGTH_LONG).show();
                        }
                     });
                else if (v == removeFromFavorites) {
                    MainActivity.mNavigation.setAsFavorite(ga, false, (status) -> {
                        if (status == API.RESULT_OK) {
                            Toast.makeText(getContext(), "Place is removed from favorite",
       Toast.LENGTH_LONG).show();
                            remove(ga);
                        }
                        else {
                            Toast.makeText(getContext(), "ERROR: Place is NOT set as favorite",
       Toast.LENGTH_LONG).show();
                        }
                    });
                }
            }
        }
        public AddressListAdapter(@NonNull Context context, int resource) {
            super(context, resource);
        }
        @NonNull
        @Override
        public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
            GeoAddress ga = getItem(position);
            Holder holder;
            if (convertView == null) {
                convertView = LayoutInflater.from(getContext()).inflate(R.layout.address_item, parent,
       false);
                holder = new Holder();
                holder.line1 = convertView.findViewById(R.id.searchResult);
                holder.line2 = convertView.findViewById(R.id.searchDetails);
                holder.show = convertView.findViewById(R.id.showPlace);
                holder.navigate = convertView.findViewById(R.id.navigateTo);
                holder.show.setOnClickListener(holder);
                holder.navigate.setOnClickListener(holder);
                holder.removeFromFavorites = convertView.findViewById(R.id.addToFavorites);
                holder.removeFromFavorites.setImageResource(android.R.drawable.ic_menu_delete);
                holder.removeFromFavorites.setOnClickListener(holder);
                convertView.setTag(holder);
            }
            else {
                holder = (Holder) convertView.getTag();
            }
            holder.ga = ga;
            String[] lines = SearchFragment.formatAddress(ga);
            holder.line1.setText(lines[0]);
            holder.line2.setText(lines[1]);
            return convertView;
        }
    }
    @SuppressLint("ShowToast")
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.list_layout, container, false);
        mAddressListAdapter = new AddressListAdapter(this.getContext(), R.layout.address_item);
        ListView listView = rootView.findViewById(R.id.listLayoutList);
        listView.setAdapter(mAddressListAdapter);
        listView.setEmptyView(mEmptyView = rootView.findViewById(R.id.empty));
        MainActivity.mNavigation.getPlaces(Enums.SavedPlaceType.favorites, (status, list) -> {
            if (status == API.RESULT_OK)
                mAddressListAdapter.addAll(list);
            else {
                mEmptyView.setText(R.string.api_not_supported);
                mAddressListAdapter.notifyDataSetChanged();
            }
        });
        return rootView;
    }
}
```

## 0.7.7 SavedRoutesFragment.java

Using Favorite Routes API

```
package hr.mireo.arthur.apidemo;
import android.annotation.SuppressLint;
import android.content.Context;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import hr.mireo.arthur.api.API;
import hr.mireo.arthur.api.Route;
import static hr.mireo.arthur.api.API.RESULT_OK;
public class SavedRoutesFragment extends Fragment {
    private RouteListAdapter mAdapter;
    private TextView mEmptyView;
    class RouteListAdapter extends ArrayAdapter<Route> {
        private class Holder  {
            Route route;
            TextView line1;
            TextView line2;
            ImageButton navigate;
            ImageButton remove;
        }
        public RouteListAdapter(@NonNull Context context, int resource) {
            super(context, resource);
        }
        @NonNull
        @Override
        public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
            Route route = getItem(position);
            String[] lines = formatRoute(route);
            Holder holder;
            if (convertView == null) {
                convertView = LayoutInflater.from(getContext()).inflate(R.layout.route_item, parent, false);
                holder = new Holder();
                holder.line1 = convertView.findViewById(R.id.routeName);
                holder.line2 = convertView.findViewById(R.id.routeDetails);
                holder.navigate = convertView.findViewById(R.id.setAsActive);
                holder.navigate.setOnClickListener( v ->
                        MainActivity.mNavigation.loadRoute(route.routeId, (status, r) -> {
                            if (RESULT_OK == status) {
                                String destination = r != null && r.routePoints.size() > 0 ?
                                        r.routePoints.get(r.routePoints.size() - 1).formatted[0]: "";
                                Toast.makeText(getContext(), "Route set to " + destination,
       Toast.LENGTH_LONG).show();
                            }
                        })
                );
                holder.remove = convertView.findViewById(R.id.removeRoute);
                holder.remove.setOnClickListener(v ->
                        MainActivity.mNavigation.removeRoute(route.routeId, (s) -> {
                            if (s == API.RESULT_OK) {
                                Toast.makeText(getContext(), "Route removed", Toast.LENGTH_LONG).show();
                                refresh();
                            }
                            else
                                Toast.makeText(getContext(), "Error removing route",
       Toast.LENGTH_LONG).show();
                        })
                );
                convertView.setTag(holder);
            }
            else {
                holder = (Holder) convertView.getTag();
            }
            holder.route = route;
            holder.line1.setText(lines[0]);
            holder.line2.setText(lines[1]);
            return convertView;
        }
    }
    @SuppressLint("ShowToast")
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.list_layout, container, false);
        mAdapter = new RouteListAdapter(this.getContext(), R.layout.route_item);
        ListView listView = rootView.findViewById(R.id.listLayoutList);
        listView.setAdapter(mAdapter);
        listView.setEmptyView(mEmptyView = rootView.findViewById(R.id.empty));
        refresh();
        return rootView;
    }
    private static String[] formatRoute(Route r) {
        return new String[] {
             r.name,
             "Distance: " + Math.round(r.totalMeters * 10.0f / 1000) / 10.f + "km; " + (r.metadata == null ?
       "": r.metadata)
        };
    }
    private void refresh() {
        MainActivity.mNavigation.getSavedRoutes((status, list) -> {
            if (status == RESULT_OK) {
                mAdapter.clear();
                mAdapter.addAll(list);
                mAdapter.notifyDataSetChanged();
            }
            else {
                mEmptyView.setText(R.string.api_not_supported);
            }
        });
    }
}
```

## 0.7.8 SearchFragment.java

Using Search API

```
package hr.mireo.arthur.apidemo;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.inputmethod.InputMethodManager;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.graphics.drawable.DrawableCompat;
import androidx.core.graphics.drawable.IconCompat;
import androidx.fragment.app.Fragment;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.List;
import hr.mireo.arthur.api.API;
import hr.mireo.arthur.api.EasyAPI;
import hr.mireo.arthur.api.GeoAddress;
public class SearchFragment extends Fragment  {
    private EditText mQuery;
    private EditText mCountry;
    private ListView mSearchResults;
    private Toast mToast;
    private CheckBox mTokenSearch;
    private AddressListAdapter mSearchListAdapter;
    static class AddressListAdapter extends ArrayAdapter<GeoAddress> {
        private class Holder implements View.OnClickListener {
            GeoAddress ga;
            TextView line1;
            TextView line2;
            ImageButton addToFavorites;
            ImageButton show;
            ImageButton navigate;
            @Override
            public void onClick(View v) {
                if (v == show)
                    MainActivity.mNavigation.showOnMap(ga, (status)-> {});
                else if (v == navigate) {
                     MainActivity.mNavigation.navigateTo(ga, true, (status, address) -> {
                        if (status == API.RESULT_OK) {
                            Toast.makeText(getContext(), "Navigation succeeeded", Toast.LENGTH_LONG).show();
                        }
                     });
                }
                else if (v == addToFavorites) {
                    final boolean setAsFav = !ga.isFavorite();
                    MainActivity.mNavigation.setAsFavorite(ga, setAsFav, (status) -> {
                        if (status == API.RESULT_OK) {
                            addToFavorites.setImageResource(setAsFav ?
                                    android.R.drawable.ic_menu_delete:
                                    android.R.drawable.ic_menu_add
                            );
                            ga.setFavorite(setAsFav);
                            if (setAsFav)
                                Toast.makeText(getContext(), "Place is set as favorite",
       Toast.LENGTH_LONG).show();
                            else
                                Toast.makeText(getContext(), "Place is removed from favorite",
       Toast.LENGTH_LONG).show();
                        }
                        else {
                            Toast.makeText(getContext(), "ERROR: Place is NOT set as favorite",
       Toast.LENGTH_LONG).show();
                        }
                    });
                }
            }
        }
        public AddressListAdapter(@NonNull Context context, int resource) {
            super(context, resource);
        }
        @NonNull
        @Override
        public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
            GeoAddress ga = getItem(position);
            String[] lines = formatAddress(ga);
            Holder holder;
            if (convertView == null) {
                convertView = LayoutInflater.from(getContext()).inflate(R.layout.address_item, parent,
       false);
                holder = new Holder();
                holder.line1 = convertView.findViewById(R.id.searchResult);
                holder.line2 = convertView.findViewById(R.id.searchDetails);
                holder.show = convertView.findViewById(R.id.showPlace);
                holder.navigate = convertView.findViewById(R.id.navigateTo);
                holder.show.setOnClickListener(holder);
                holder.navigate.setOnClickListener(holder);
                holder.addToFavorites = convertView.findViewById(R.id.addToFavorites);
                holder.addToFavorites.setImageResource(ga.isFavorite() ?
                        android.R.drawable.ic_menu_delete:
                        android.R.drawable.ic_menu_add
                );
                holder.addToFavorites.setOnClickListener(holder);
                convertView.setTag(holder);
            }
            else {
                holder = (Holder) convertView.getTag();
            }
            holder.ga = ga;
            holder.line1.setText(lines[0]);
            holder.line2.setText(lines[1]);
            return convertView;
        }
    }
    @SuppressLint("ShowToast")
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(hr.mireo.arthur.apidemo.R.layout.search_layout, container, false);
        mQuery = rootView.findViewById(R.id.query);
        mQuery.setOnFocusChangeListener((v, hasFocus) -> {
            if (!hasFocus) hideKeyboard(v);
        });
        mCountry = rootView.findViewById(R.id.country);
        mCountry.setOnFocusChangeListener((v, hasFocus) -> {
            if (!hasFocus) hideKeyboard(v);
        });
        mSearchListAdapter = new AddressListAdapter(this.getContext(), R.layout.address_item);
        mSearchResults = rootView.findViewById(R.id.searchResults);
        mSearchResults.setAdapter(mSearchListAdapter);
        mToast = Toast.makeText(SearchFragment.this.getActivity(), "", Toast.LENGTH_LONG);
        mToast.setGravity(Gravity.CENTER, 0, 0);
        Button mSearchButton = rootView.findViewById(R.id.search);
        mSearchButton.setOnClickListener(v -> {
            if (mQuery.hasFocus())
                hideKeyboard(mQuery);
            if (mCountry.hasFocus())
                hideKeyboard(mCountry);
            String query = mQuery.getText().toString();
            String country = mCountry.getText().toString();
            mToast.setText("Searching...");
            mToast.show();
            mSearchListAdapter.clear();
            if (mTokenSearch.isChecked())
                doTokenSearch(query);
            else
                doRegularSearch(query, country);
        });
        Button mShowResultsOnMapButton = rootView.findViewById(R.id.show_on_map);
        mShowResultsOnMapButton.setOnClickListener(v -> {
            if (mQuery.hasFocus())
                hideKeyboard(mQuery);
            if (mCountry.hasFocus())
                hideKeyboard(mCountry);
            String query = mQuery.getText().toString();
            String country = mCountry.getText().toString();
            mToast.setText("Searching...");
            mToast.show();
            mSearchListAdapter.clear();
            MainActivity.mNavigation.search(query, country, 30, true, (i, addresses) -> {
                mToast.cancel();
                if (addresses != null && addresses.size() > 0) {
                    mSearchListAdapter.addAll(addresses);
                    mSearchResults.forceLayout();
                }
            });
        });
        mTokenSearch = rootView.findViewById(R.id.tokenSearch);
        mTokenSearch.setOnClickListener(v -> {
            if (mTokenSearch.isChecked()) {
                mQuery.setHint("tokens (JSON)");
                mCountry.setVisibility(View.GONE);
            }
            else {
                mQuery.setHint("address or query");
                mCountry.setVisibility(View.VISIBLE);
            }
        });
        return rootView;
    }
    private void hideKeyboard(View view) {
        InputMethodManager inputMethodManager = (InputMethodManager)
                view.getContext().getSystemService(Activity.INPUT_METHOD_SERVICE);
        inputMethodManager.hideSoftInputFromWindow(view.getWindowToken(), 0);
    }
    static String[] formatAddress(GeoAddress ga) {
        if (ga.formatted != null && ga.formatted.length > 1) return ga.formatted;
        String[] result = new String[2];
        result[0] = !ga.POI.isEmpty() ? ga.POI : !ga.street.isEmpty() ? ga.street + " " + ga.houseNumber :
       ga.city;
        result[1] = (ga.POI.isEmpty() ? ga.street + " " + ga.houseNumber + ", " : "") + ga.postal + " " +
       ga.city + ", " + ga.country;
        return result;
    }
    private void doRegularSearch(String query, String country) {
        MainActivity.mNavigation.search(query, country, 30, false, mSearchCallback);
    }
    private void doTokenSearch(String tokens) {
        try {
            new JSONObject(tokens);
            MainActivity.mNavigation.tokenSearch(tokens, 30, false, mSearchCallback);
        } catch (JSONException ex) {
            mToast.setText("tokens is not valid JSON!");
        }
    }
    final EasyAPI.AddressListResult mSearchCallback = new EasyAPI.AddressListResult() {
        @Override
        public void onAddressList(int i, List<GeoAddress> addresses) {
            mToast.cancel();
            if (addresses != null && addresses.size() > 0) {
                mSearchListAdapter.addAll(addresses);
            } else {
                mSearchListAdapter.clear();
            }
            mSearchResults.forceLayout();
        }
    };
}
```

### 0.7.9 SettingsFragment.java

Settings API usage

```
package hr.mireo.arthur.apidemo;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.Spinner;
import android.widget.Switch;
import java.util.List;
import java.util.Objects;
import hr.mireo.arthur.api.Voice;
public class SettingsFragment extends Fragment implements Switch.OnCheckedChangeListener,
       OnItemSelectedListener {
    private int lastSelectedVoice = 0;
    private int lastSelectedLanguage = 0;
    private int lastSelectedUnits = 0;
    private int lastSelectedMapMode = 0;
    private List<Voice> installedVoices;
    private static SettingsFragment mThis;
    private Spinner createSpinner(View root, int res_id, String[] items) {
        Spinner s = root.findViewById(res_id);
        if (items == null) items = new String[]{};
        ArrayAdapter<CharSequence> adapter = new ArrayAdapter<>(Objects.requireNonNull(this.getActivity()),
       android.R.layout.simple_spinner_item, items);
        // Specify the layout to use when the list of choices appears
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        // Apply the adapter to the spinner
        s.setAdapter(adapter);
        return s;
    }
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        mThis = this;
        final View rootView = inflater.inflate(hr.mireo.arthur.apidemo.R.layout.settings_layout, container,
       false);
        ((Switch)
       rootView.findViewById(hr.mireo.arthur.apidemo.R.id.mute)).setOnCheckedChangeListener(this);
        ((Switch)
       rootView.findViewById(hr.mireo.arthur.apidemo.R.id.speedAlerts)).setOnCheckedChangeListener(this);
        ((Switch)
       rootView.findViewById(hr.mireo.arthur.apidemo.R.id.safetyCameraAlerts)).setOnCheckedChangeListener(this);
        ((Switch)
       rootView.findViewById(hr.mireo.arthur.apidemo.R.id.poiAlongTheRoute)).setOnCheckedChangeListener(this);
        ((Switch)
       rootView.findViewById(hr.mireo.arthur.apidemo.R.id.liveTraffic)).setOnCheckedChangeListener(this);
        ((Switch)
       rootView.findViewById(hr.mireo.arthur.apidemo.R.id.timeFormat)).setOnCheckedChangeListener(this);
        MainActivity.mNavigation.getInstalledVoices((status, list) -> {
            installedVoices = list;
            String[] labels = new String[installedVoices.size()];
            for (int i = 0; i < labels.length; i++)
                labels[i] = installedVoices.get(i).name + " (" + installedVoices.get(i).language + ")";
            createSpinner(rootView, R.id.guidanceVoice, labels).setOnItemSelectedListener(mThis);
        });
        MainActivity.mNavigation.getInstalledLanguages((i, strings) -> createSpinner(rootView,
       R.id.uiLanguage, strings).setOnItemSelectedListener(mThis));
        createSpinner(rootView, hr.mireo.arthur.apidemo.R.id.distanceUnits, new String[]{"Kilometers",
       "Miles"}).setOnItemSelectedListener(mThis);
        createSpinner(rootView, hr.mireo.arthur.apidemo.R.id.displayMode, new String[]{"Automatic", "Day
       mode", "Night mode"}).setOnItemSelectedListener(mThis);
        Button resetDefaults = rootView.findViewById(R.id.resetDefaults);
        resetDefaults.setOnClickListener(v -> MainActivity.mNavigation.resetToDefaults(null));
        Button restorePurchases = rootView.findViewById(R.id.restorePurchases);
        restorePurchases.setOnClickListener(v -> MainActivity.mNavigation.restorePurchases(null));
        return rootView;
    }
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        switch (buttonView.getId()) {
            case hr.mireo.arthur.apidemo.R.id.mute:
                MainActivity.mNavigation.setMute(isChecked, null);
                break;
            case hr.mireo.arthur.apidemo.R.id.speedAlerts:
                MainActivity.mNavigation.enableSpeedAlerts(isChecked, null);
                break;
            case hr.mireo.arthur.apidemo.R.id.safetyCameraAlerts:
                MainActivity.mNavigation.enableSafetyCameraAlert(isChecked, null);
                break;
            case hr.mireo.arthur.apidemo.R.id.poiAlongTheRoute:
                MainActivity.mNavigation.enablePoiAlongRoute(isChecked, null);
                break;
            case hr.mireo.arthur.apidemo.R.id.liveTraffic:
                MainActivity.mNavigation.enableTraffic(isChecked, null);
                break;
            case hr.mireo.arthur.apidemo.R.id.timeFormat:
                MainActivity.mNavigation.setTimeAMPM(!isChecked, null);
                break;
        }
    }
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        switch (parent.getId()) {
            case hr.mireo.arthur.apidemo.R.id.guidanceVoice:
                if (position != lastSelectedVoice)
                    MainActivity.mNavigation.setVoice(installedVoices.get(position).id, null);
                lastSelectedVoice = position;
                break;
            case hr.mireo.arthur.apidemo.R.id.uiLanguage:
                if (position != lastSelectedLanguage)
                    MainActivity.mNavigation.setLanguage(parent.getItemAtPosition(position).toString(),
       null);
                lastSelectedLanguage = position;
                break;
            case hr.mireo.arthur.apidemo.R.id.distanceUnits:
                if (position != lastSelectedUnits) {
                    MainActivity.mNavigation.setUnits(position == 0, null);
                }
                lastSelectedUnits = position;
                break;
            case hr.mireo.arthur.apidemo.R.id.displayMode:
                if (position != lastSelectedMapMode)
                    MainActivity.mNavigation.setDisplayMode(position, null);
                lastSelectedMapMode = position;
                break;
        }
    }
    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
}
```

# Index