**Revision history:**

**Mireo Android API 3.0.0**

**Android API specification**

Document date: April 10, 2020.

|  |  |  |
| --- | --- | --- |
| **Date** | **Person** | **Description** |
| October 13, 2014. | Boris Bralo, Mireo | Document created |
| February 23, 2016 | Boris Bralo, Mireo | Updated for EasyAPI |
| November 11, 2018 | Boris Bralo, Mireo | Remove requirement that application needs to be started at least once. New APIs added: setExternalDisplayMode, mapSmartZoom. New notifications APIs added,old ones are deprecated |
| April 1, 2019 | Boris Bralo, Mireo | New APIs: getCurrentAddress, getAddressAtLonLat |
| April 26, 2019 | Boris Bralo, Mireo | New APIs: getBooleanSettings, getAudioAlertSettings, showTrafficSettings, getDisplayMode |
| September 10, 2019 | Boris Bralo, Mireo | Add confidence to search result |
| October 24, 2019 | Boris Bralo, Mireo | Added trafficStatus method |
| February 18, 2020 | Sanjin Ružić, Mireo | Added showSearchResultsPage method and ESuperFamily enumeration |
| April 18, 2020 | Boris Bralo, Mireo | Broadcast is replaced with bound service as preferred method of communication. Addition of custom API bring\_to\_foreground parameters |

**Table of contents**

[Mireo Android API 3.0.0 4](#_Toc38354015)

[Android project setup 5](#_Toc38354016)

[EasyAPI class 6](#_Toc38354017)

[Callback interfaces 6](#_Toc38354018)

[Constructors 7](#_Toc38354019)

[Methods 7](#_Toc38354020)

[Interface INotificationListener 25](#_Toc38354021)

[Interface InotificationListener2 27](#_Toc38354022)

[Class Route 28](#_Toc38354023)

[Class Advice 30](#_Toc38354024)

[Class GeoAddress 32](#_Toc38354025)

[Methods 33](#_Toc38354026)

[Class PositionData 34](#_Toc38354027)

[Class TrafficStatus 35](#_Toc38354028)

[Class VersionInfo 36](#_Toc38354029)

[APPENDIX A – Mireo extended ISO codes 37](#_Toc38354030)

[APPENDIX B – Mireo Type Codes 41](#_Toc38354031)

# Mireo Android API 3.0.0

This document describes Mireo 3.0.0 Android API. Communication with Mireo GENIUS MAPS is performed using this APIs. Besides GENIUS MAPS, the same API can be used to control customized GENIUS MAPS derivatives. The target application is selected using custom URL scheme provided by Mireo.

## Android project setup

Mireo Genius Maps API is wrapped in java library **api.jar** which has to be included in your android project. To include Mireo API to your Android project, just copy api.jar into your Android project's libs directory.

API consist of following java classes:

* EasyAPI – main entry for Genius Maps API. Each method returns API request.
* APIAsyncRequest – encapsulated API request
* Various other classes to wrap complex data (Route, SpeedCamera, Voice, Advice..)

The main object that handles API functionality is instance of EasyAPI class, and API functionality is exposed in a set of public methods, whose names clearly indicate the functionality they perform. Each of these calls will internally use Android system facilities to pass the request to GENIUS MAPS.

### Example

Along with this documentation, both the api.jar and the API test application with full source code, have been provided, demonstrating in full the use of the API.

**import** hr.mireo.arthur.api.EasyAPI;

// to use callbacks without EasyAPI prefix

**import static** hr.mireo.arthur.api.EasyAPI.\*;

**public void** mireoApiTest() {

// Create API and open link

// The prefix „gm“ is for Mireo GeniusMaps.

// Each custom application has its own URL prefix

mAPI = **new** EasyAPI(„gm“, **this**);

**// Call API asynchonously with callback**

mAPI.showMap(status -> {

if (status != API.RESULT\_OK) // ERROR

return;

});

**// Call API synchonously, wait for result**

*mApi*.search(„Mireo“, „Croatia“, 30, **false**,

(**status**, addresses) -> {

if (status == API.RESULT\_OK) **process\_results(addresses);**   
}).**waitForResult**(-1);

mApi.destroy();  
}

# EasyAPI class

This is the wrapper class that contains all currently supported API functionality as a list of supported API call names.

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

## Callback interfaces

Callback intefaces are static members of **EasyAPI** class.

|  |
| --- |
| **AddressListResult**  Callback interface for EasyAPI calls that return list of addresses. |
| **AddressResult**  Callback interface for EasyAPI calls that return single address. |
| **AdviceListResult**  Callback interface for getAdviceList EasyAPI call. |
| **GpsResult**  Callback interface for EasyAPI getGpsPosition call |
| **IntResult**  Callback interface for EasyAPI calls that return integer values. |
| **Result**  Callback interface for EasyAPI calls without value to return. Only status of the call is reported. |
| **RouteSettingsResult**  Callback interface for getRouteSettings. |
| **StringArrayResult**  Callback interface for EasyAPI calls that return string array. |
| **StringResult**  Callback interface for EasyAPI calls that return string values. |
| **VoiceListResult**  Callback interface for getInstalledVoices EasyAPI call. |
| **VersionResult**  Callback interface for getVersionEx EasyAPI call. |
| **AudioAlertResult**  Callback interface for getAudioAlertSettings EasyAPI call. |
| **ShowTrafficResult**  Callback interface for showTrafficSettings EasyAPI call. |
| **TrafficStatusResult**  Callback interface for trafficStatus EasyAPI call. |

## Constructors

**public  EasyAPI(String urlScheme, Context context)**

Create EasyAPI client.

**Parameters:**

urlScheme – Mireo application URL scheme (gm:// for GENIUS MAPS)

context - Parent context . It is usualy main Application context.

## Methods

**public APIAsyncRequest addDestination(**GeoAddress**address,** AddressResult**cb)**

Add destination to the current route. The route can be current navigation route or the route created with startRoute call.

**Parameters:**

address - destination address

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**addWaypoint(**[GeoAddress](file:///C:\\Temp\\apidoc\\hr\\mireo\\arthur\\api\\GeoAddress.html" \o "class in hr.mireo.arthur.api)**address,boolean addBeforeEnd,** [AddressResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressResult.html)**cb)**

Add waypoint to the current route. Waypoint is added after start if addBeforeEnd is false and before last destination otherwise.

**Parameters:**

address - waypoint address

addBeforeEnd - add waypoint before end

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**clearRoute(**[Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Clear active route

**Parameters:**

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**clearSearchResults(**[Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Clear search results from map.

**Parameters:**

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**closeWindow(**[Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

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:**

cb - Result callback (may be null)

**public void destroy()**

Destroy must be called when we are done with the API

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**enablePoiAlongRoute(**boolean**value,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Enable/disable POI along the route.

**Parameters:**

value  - true or false

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**enableSafetyCameraAlert(**boolean**value,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Enable/disable safety camera alerts .

**Parameters:**

value - true or false

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**enableSpeedAlerts(**boolean**value,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Enable/disable speed alerts.

**Parameters:**

value - Enable/Dissable speed alerts

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**enableSignposts(**boolean**value,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Enable/disable sigposts.

**Parameters:**

value - Enable/Dissable signpost display

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**enableJunctionViews(**boolean**value,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Enable/disable junction views. Junction view is a simplified 3D image of the junction.

**Parameters:**

value - Enable/Dissable junction views.

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**enableUiSounds(**boolean**value,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Enable/disable UI sounds: clicks and/or TTS alerts. UI sounds are all non-guidance sounds.

**Parameters:**

value - Enable/Dissable UI sounds

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**enableTraffic(**boolean**value,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Enable/disable Live Traffic

**Parameters:**

value  - Enable/Dissable traffic

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**endNavigation(**[Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

End navigation.

**Parameters:**

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**exactSearch(String city, String street, String houseNumber, String country,   
 boolean showOnMap,** [AddressListResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressListResult.html)**cb)**

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

**Parameters:**

city – city  
street – street  
houseNumber - house number  
country – country – empty string or null means all countries.   
showOnMap - show results on map  
cb - Result callback (MUST NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**findClosestPoi(**[GeoAddress](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html)**origin, int[] poiCategories,**

**int[] extraPoiTypes, int maxResults,boolean showOnMap,**

[AddressListResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressListResult.html)**cb)**

Find POIs closest to the **origin**.

**Parameters:**

origin - origin of search

poiCategories - list of poi categories (Enums.TypeFamily)

extraPoiTypes - extra POI types, outside listed categories

maxResults - maximum number of results to return

showOnMap - show results on map

cb - Result callback (MUST NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getActiveRoute(**[AddressListResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressListResult.html)**rr)**

Get active route. Resulting Route is reported to callback as a list of waypoints.

**Parameters:**

rr - Result callback (MUST NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getAdviceList(int maxItems,** [AdviceListResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AdviceListResult.html)**alr)**

Get the list of advices on the active route.

**Parameters:**

maxItems - maximum number of advices to return

alr - Result callback (MUST NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getDtg(**[IntResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.IntResult.html)**ir)**

Get distance to go to the next destination in meters.

**Parameters:**

ir - Result callback (MUST NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getFinalDestination(**[AddressResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressResult.html)**cb)**

Get the last destination on active route. Resulting GeoAddress is reported in callback.

**Parameters:**

cb - Result callback (may NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getGpsPosition(**[GpsResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.GpsResult.html)**cb)**

Get current GPS position. Result of this method will be delivered in callback.

**Parameters:**

cb - callback where result will be delivered (may NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getInstalledLanguages(**[StringArrayResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.StringArrayResult.html)**r)**

Get the list of installed languages.

**Parameters:**

r - Result callback (MUST NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getInstalledVoices(**[VoiceListResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.VoiceListResult.html)**vlr)**

Get the list of installed guidance voices.

**Parameters:**

vlr – Result callback where result will be delivered (may NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getNextDestination(**[AddressResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressResult.html)**cb)**

Get next destination on active route ahead of current position. Resulting GeoAddress is reported in callback.

**Parameters:**

cb - Result callback (may NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getPlaces(**[Enums.SavedPlaceType](file:///C:\Temp\apidoc\hr\mireo\arthur\api\Enums.SavedPlaceType.html)**kind,** [AddressListResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressListResult.html)**alr)**

Get saved places

**Parameters:**

kind - 1 - favourites, 2 - recent destinations, 3- all

alr  - Result callback (may NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getRouteSettings(**[RouteSettingsResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.RouteSettingsResult.html)**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)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getTtg(**[IntResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.IntResult.html)**ir)**

Get time to go to the next destination in seconds

**Parameters:**

ir - Result callback (may NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getVersion(**[IntResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.IntResult.html)**cb)**

Get navigation version. Result will be returned in callback as 4 digit number MmBP where M is major version, m is minor version, B is build number and P is patch level.

**Parameters:** cb - Result callback (may NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getVersionEx(**[VersionResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.IntResult.html)**cb)**

Get extended version information. Result will be returned in callback as vesrionInfo structure.

**Parameters:** cb - Result callback (may NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getWholeDtg(**[IntResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.IntResult.html)**ir)**

Get distance to go to the final destination in meters.

**Parameters:**

ir - Result callback (may NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getWholeTtg(**[IntResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.IntResult.html)**ir)**

Get time to go to the final destination in seconds.

**Parameters:**

ir - Result callback (may NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**mapZoom(**long**display\_num,** double**factor,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

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

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**mapSmartZoom(**double**factor,** bool **enable\_smart\_zoom**, [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

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 on route: vehicle speed, distance to next maneuver, distance to next POI or safety camera etc..

**Parameters:**

factor - zoom factor

enable\_smart\_zoom – enable smart zoom

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**navigateTo(**[GeoAddress](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html)**address,** boolean**noUI,** [AddressResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressResult.html)**cb)**

Navigate to destination

**Parameters:**

address - destination address

noUI - do not show UI ie. start navigation in background

cb  - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**registerActivationCode(**String**activationCode,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**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  
cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**repeatGuidance(**[Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**rr)**

Repeat current voice guidance instruction.

**Parameters:**cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**replaceDestination(**[GeoAddress](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html)**address,** [AddressResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressResult.html)**cb)**

Replace last destination. Part of route planning API.

**Parameters:**

address - new destination address

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**requestNotifications(**String **client\_id,** int**mask,** [INotificationListener](file:///C:\Temp\apidoc\hr\mireo\arthur\api\INotificationListener.html)**notificationListener,** Bool[Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Request notifications from the application.

**Parameters:**

client\_id – Unique client id.

mask - Notification events mask (combination of API.NOTIFY\_\* constants). Setting mask to 0 has the same effect as calling removeNotifications

notificationListener - Notification callback

cb - Result callback (may be null). If set, the resultin boolean value will return if route is active.

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**removeNotifications(**String **client\_id,** [INotificationListener](file:///C:\Temp\apidoc\hr\mireo\arthur\api\INotificationListener.html)**notificationListener,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Unsubcribe from all notifications.

**Parameters:**

client\_id – Unique client id.

notificationListener - Notification callback

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**removeNotifications(**String **client\_id,** int **mask**,[INotificationListener](file:///C:\Temp\apidoc\hr\mireo\arthur\api\INotificationListener.html)**notificationListener,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Unsubcribe from specific notifications.

**Parameters:**

client\_id – Unique client id.

mask - Notification events mask (combination of API.NOTIFY\_\* constants)

notificationListener - Notification callback

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**resetToDefaults(**[Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Reset all settings to defaults.

**Parameters:**

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**restorePurchases(**[Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Restore in-app purchases.

**Parameters:**

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**search(**String**query,** String**country,** int**maxResults,** boolean**showOnMap,** [AddressListResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressListResult.html)**cb)**

Free-form search

**Parameters:**

query - search query

country - searchj country, empty string or null means all countries

maxResults - maximum number of results to return

showOnMap - show results on map

cb  - Result callback (MAY NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**setDisplayMode(**int**mode,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Set map display mode. Auotmatic mode calculates appropriate display mode according to the GPS postion and time.

**Parameters:**

mode - 0 - automatic, 1 - day, 2 - night

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**setExternalDisplayMode(**int**mode,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Set external map display mode. As oppesed to **setDisplayMode** API, this method will only set display mode if user has selected „automatic“ display mode in UI. Also, this API overrides default calculation, as if this value is set from sensor.

**Parameters:**

mode - 0 - automatic, 1 - day, 2 - night

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**setLanguage(**String**language\_id,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Set application language

**Parameters:**

language\_id - language id from the list of installed languages

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**setMute(**boolean**mute,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Mute/unmute navigation sounds

**Parameters:**

mute - true means mute, false unmute

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**setRouteSettings(**int**routeType,**int**avoidMask,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Set default route setings

**Parameters:**

routeType - default route type: 1 - quickest, 2- shortest

avoidMask - or-ed values of EAvoidMask enum

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**setTimeAMPM(**boolean**value,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Set time format

**Parameters:**

value – **true -** use 12 hour (AM/PM) time format , **false** - use 24 hour format otherwise

cb - Result callback (may be null)

**public**[**APIAsyncRequest**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**setUnits(**boolean**metric,** [**Result**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Set measurements units

**Parameters:**

metric - if true, use metric units, use imperial units otherwise

cb - Result callback (may be null)

**public**[**APIAsyncRequest**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**setVoice(**String**voiceName,** [**Result**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Set the voice for voice guidance.

**Parameters:**

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

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**showMap(**[Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

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

**Parameters:**

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**showOnMap(**[GeoAddress](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html)**address,** [Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Show address on map. Address must be georeferenced (ie. it must have valid coordinetes)

**Parameters:**

address - place to show

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**startNavigation(**[Result](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.Result.html)**cb)**

Start navigating planned route.

**Parameters:**

cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**startRoute(**[GeoAddress](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html)**address,** [AddressResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressResult.html)**cb)**

Start planning new route. Current route is cleared, if it exists. Client can build route further using addDestination and addWaypoint APIs. When route planning is finished, call startNavigation() to initiate driving the route.

**Parameters:**address - start address  
cb - Result callback (may be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getCurrentAddress(**[AddressResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressResult.html)**cb)**

Get address on current GPS position

**Parameters:**cb - Result callback (must NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getAddressAtLonlat(**float **longitude**, float **latitude,** [AddressResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressResult.html)**cb)**

Get address at the point with given longitude and latitude.

**Parameters:**longitude – geographic longitude in degrees  
latitude – geographic latitude in degrees  
cb - Result callback (must NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getBooleanSettings(**[String](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressResult.html)**name,** [BoolResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressResult.html)**cb)**

Get boolean setting value

**Parameters:**name – setting name

cb - Result callback (must NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getAudioAlertSettings(**AudioAlertResult **cb)**

Get audio alert settings: do we play audio alerts for speeding, safety cameras and next petrol station.

**Parameters:**cb - Result callback (must NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**showTrafficSettings (**ShowTrafficResult **cb)**

Get traffic display settings: do we draw traffic flow, traffic events or both

**Parameters:**cb - Result callback (must NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**getDisplayMode (**boolean **external,** IntResult **cb)**

Get display mode settings. Display mode can be external, i.e. set by device by some means; light sensors, for example, or internal, set by user in application settings. Note that internal display mode setting takes precedence - only if it is set to AUTOMATIC (0) external display mode is taken into account.

**Parameters:**

external – should callback return external display mode, as set by device, or internal display mode, set by user.

cb - Result callback (must NOT be null)

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**trafficStatus (**TrafficStatusResult **cb)**

Get traffic status. **TrafficStatus** instance with all data is returned in callback.

**Parameters:**cb - Result callback (must NOT be null)

**public static String statusMessage(**int**status)**

Get human readable status message for returned status code. Status messages are in English.

**Parameters:**

status - Status code

**Returns:**

Status message

**public**[**APIAsyncRequest**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**tokenSearch(**String**tokens,**

int**maxResults,**

boolean**showOnMap,**

[EasyAPI.AddressListResult](file:///D:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressListResult.html)**alr)**

Search with tokens.   
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

Status message

public APIAsyncRequest tokenSearch(String tokens,

int maxResults,

boolean showOnMap,

EasyAPI.AddressListResult alr) Status message

**public**[**APIAsyncRequest**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**showSearchResultsPage(**String**query,** 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

**public**[APIAsyncRequest](file:///C:\Temp\apidoc\hr\mireo\arthur\api\APIAsyncRequest.html)**customApi(**String**apiName,** boolean**bringToForeground,** [StringResult](file:///C:\Temp\apidoc\hr\mireo\arthur\api\EasyAPI.AddressListResult.html)**cb,** Obj... **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**

apiName – the name of the custom API  
bringToForeground  – bring GENIUS MAPS to forground  
cb – callback, may be null if not needed.  
args  – custom API arguments

# Interface INotificationListener

Notification callbacks interface. When client registers itself to certain kinds of events using  EasyAPI.requestNotifications API, the events are sent to this callback.

### Methods

**void OnAdvice(**[Advice](file:///C:\Temp\apidoc\hr\mireo\arthur\api\Advice.html)**advice)**

Called when advice changes. Client can expect update every second , in sync with GPS signal.

**Parameters:**

advice - Current active advice.

**void OnRoute(**[Route](file:///C:\Temp\apidoc\hr\mireo\arthur\api\Route.html)**route)**

Called when active route changes. This event is triggered when route is started, canceled or when destination is reached. intervention

**Parameters:**

route - active route

**void OnSpeedCamera(**int**limit\_kmh,** int**distance\_to\_camera\_in\_meters)**

Called to notify about speed camera information changes. When there is active speed camera, this callback is called every second (in sync with GPS signal) and distance to camera is updated.

**Parameters:**

limit\_kmh - Speed limit in km/h.

distance\_to\_camera\_in\_meters - Distance to speed camera in meters.

**void OnSpeedLimit(**int**limit\_kmh)**

Called when speed limit is changed.

**Parameters:**

limit\_kmh - Current speed limit in km/h

# Interface InotificationListener2

Notification callbacks interface. When client registers itself to certain kinds of events using  EasyAPI.requestNotifications API, the events are sent to this callback. This interface adds additional method to INotificationListener inteface.

### Methods

**void OnPositionData(**[PositionData](file:///C:\Temp\apidoc\hr\mireo\arthur\api\Advice.html)**data)**

Called when position changes. Client can expect update every second , in sync with GPS signal.

**Parameters:**

data - Current posizion data.

**void OnSpeedViolations(**int **limit\_kmh**, int **current\_speed\_kmh)**

Called on speed violation.

**Parameters:**

limit\_kmh - Current speed limit in km/h.

current\_speed\_kmh - Current speed in km/h.

# Class Route

Class that represents current route.

|  |  |
| --- | --- |
| Fields | |
| int | **completedMeters**  Distance in meters completed so far |
| int | **dtgToNext**  Remaining distance to the next waypoint |
| int | **elapsedSeconds**  Duration in seconds spent on route so far |
| boolean | **isActive**  True if route is active |
| boolean | **isCompleted**  True if route is completed |
| int | **remainingMeters**  Remaining distance to final destination in meters (DTG) |
| int | **remainingSeconds**  Remaining time to final destination (TTG) |
| int | **requestedFeatures**  Requested avoidance features. |
| int | **totalMeters**  Route lenght in meters |
| int | **totalSeconds**  Calculated route duration in seconds |
| int | **ttgToNext**  Remaining time to the next waypoint |
| int | **usedFeatures**  Used avoidance features. |

# Class Advice

Class that represents navigation instructions. Active navigation advice is sent to application using API notifications mechanism.

|  |  |
| --- | --- |
| Fields | |
| String | **adviceText**  Localized and formatted advice text. Formatting is very basic, only <b>,</b> for bold and <i></i> for italic is supported. |
| int | **adviceType**  Type of advice. One of enumetared constants in Enums.AdviceType |
| int | **blowupRadius**  Radius of advice validity. |
| String | **currentStreet**  Formatted street name of the current street |
| int | **entryCourseDeg**  Entry road course in degrees |
| boolean | **isDestination**  If true, advice is destination advice |
| boolean | **isViaPoint**  If true, advice is waypoint advice |
| int | **metersToAdvice**  Distance to advice in meters |
| String | **nextStreet**  Formatted street name of the street where advice is leading. |
| **Route** | **route**  Active route object |
| **int** | **roundaboudArcAngle**  **Roundabout arc angle in degrees counted clockwise** |
| **int** | **roundaboutExit**  **Roundabout exit number counted from 1** |
| **I18** | **i18N**  Advice data localized to currently selected language and time format.  **String distanceToAdvice –** localized distance  **String timeToAdvice –** localized time to advice  **String distanceToDestination –** self explanatory  **String timeToDestination -** self explanatory  **String routeETA –** localized ETA. Uses curent  **int units –** unit used 0 – metric, 1 - imperial |

# Class GeoAddress

|  |  |
| --- | --- |
| Fields | |
| String | [**area**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#area)  **Area name. This field is country specific. It may be city part** |
| String | [**city**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#city)  **Name of the city** |
| String | [**country**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#country)  **Country name** |
| Int | [**distance**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#distance)  **This field is only valid when GeoAddress object is result of the findClosestPoi call.** |
| String | [**houseNumber**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#houseNumber) |
| String | [**name**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#name) |
| String | [**phone**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#phone) |
| String | [**POI**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#POI)  **POI (Point of Interest) name** |
| String | [**postal**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#postal)  **Postal code** |
| String | [**street**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#street) |
| String | [**type**](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html#type) |
| Int | **iso – numeric ISO 3166 code, except for big countries where we have extended code given in Appendix A** |
| Double | confidence  **Confidence level (0-1) for search result. This field is only valid when GeoAddress object is result of the search and exactSearch calls** |

## Methods

**public static**[GeoAddress](file:///C:\Temp\apidoc\hr\mireo\arthur\api\GeoAddress.html)**fromLonLat(double longitude, double latitude)**

Create GeoAddress from WGS84 coordinates (longitude and latitude). GeoAddress created in this way can be used in routing API. GENIUS MAPS will do reverse geocoding if necessary.

**public void setLonLat(double lon, double lat)**

Set GeoAddress coordinate fields. GeoAddress use internal compact coordinate representation (fields x and y). This method set those fields from WGS84 coordinates (longitude and latitude).

# Class PositionData

Class that represents navigation position

|  |  |
| --- | --- |
| Fields | |
| Double | **longitude** WGS84 longitude in degrees. |
| Double | **latitude** WGS84 longitude in degrees |
| Int | **speed** Speed in km/h. |
| Int | **course** Course (bearing) in degrees from north |
| Int | **utc** UTC time |
| Int | **quality** HDOP |
| Int | **altitude** altitude in meters |
| Int | **speed\_limit** Current road speed limit, if available, in km/h |
| Int | **camera\_limit** Current speed camera limit, if available, in km/h. |
| **Int** | **camera\_distance** Distance to speed camera, in meters. |

# Class TrafficStatus

Class that represents current live-traffic status

|  |  |
| --- | --- |
| Fields | |
| int | [**delay\_seconds**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\TrafficStatus.html#delay_seconds)Current live traffic delay in seconds or -1 if road is blocked |
| boolean | [**enabled**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\TrafficStatus.html#enabled)True if live-traffic is enabled in settings |
| boolean | [**installed**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\TrafficStatus.html#installed)True if live-traffic is installed |
| boolean | [**internet\_available**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\TrafficStatus.html#internet_available)True if internet connection is available |
| boolean | [**road\_blocked**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\TrafficStatus.html#road_blocked)True if road ahead is blocked |
| int | [**traffic\_state**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\TrafficStatus.html#traffic_state)  Live traffic fetch status. One of the codes listed bellow. |
| static int | [**UNKNOWN**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\TrafficStatus.html#UNKNOWN)**,** [**PENDING**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\TrafficStatus.html#PENDING)**, TRAFFIC\_OK, UNAVAILABLE**  Live traffic fetch status codes. |

# Class VersionInfo

Version info.

|  |  |
| --- | --- |
| Fields | |
| String | [**appVersion**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\VersionInfo.html#appVersion)  Application version |
| String | [**buildNumber**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\VersionInfo.html#buildNumber)  Application's build number |
| String | [**mapVersion**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\VersionInfo.html#mapVersion)  Maps version |
| String | [**serial**](file:///D:\Temp\apidoc\hr\mireo\arthur\api\VersionInfo.html#serial)  Device's serial number |

# APPENDIX A – Mireo 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

# APPENDIX B – Mireo 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

440 Exhibition or Conference

442 Unspecified Events

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

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

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