|  |
| --- |
| **FATOS Software Development Kit**  **V3.0.0** |

표준 제/개정이력

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **버전** | **표준 제/개정일자** | **표준 제/개정내용** | **작성자** | **승인자** |
| 3.0.0 | 2019. 12. 02 | V3.0.0 초안 배포 | 우경일 | 박현재 |

목차

1. FATOS SDK코드 받기 9

2. 개발 프로젝트 설정 9

3. API key 받기 및 적용 12

4. FATOS SDK for Android 기능 구성 13

5. 코드 이해 13

5.1 지도 그리기 13

5.1.1 Map Fragment 생성 13

5.1.2 사용자 Activity에 지도 그리기 14

5.1.3 지도 이동하기 16

5.1.4 지도 모드 변경하기 17

5.1.5 현 위치로 이동하기 18

5.1.6 Map Object Interface 18

void onSetMapStatus(int nStatus); // 0 : 주행 , 1: 경로요약 19

5.2 검색 19

5.2.1 POI(Point Of Interest) 검색 19

5.2.2 검색 결과 20

5.3 경로 탐색 23

5.3.1 현 위치 기준 경로탐색 23

5.3.2 출발지 설정 경로 탐색 25

5.4 경로 정보 26

5.4.1 경로 요약 정보 26

5.4.2 경로 상세 정보 27

5.5 경로 안내 31

5.5.1 경로안내 시작 31

5.5.2 경로 정보 32

6. API 목록 33

6.1 초기화 과정 33

6.1.1 public int initFatosNaviEngine() 33

6.1.2 public int FM\_Init(Context context, String strApiKey) 34

6.1.3 public int FM\_Destory() 34

6.2 지도 34

6.2.1 MapStatusListener 34

6.2.2 public void setMainMapOption(boolean bDrawStartImg,boolean bDrawGoalImg,boolean bDrawPinImg,boolean bDrawEVWare) 35

6.2.3 public void onMapReady() 35

6.2.4 public void FM\_MoveCurPosition(FMBaseActivity.OnFatosMapListener onFatosMapListener) 36

6.2.5 public void FM\_SetMapAirMode(int nMapType) 36

6.2.6 public void FM\_EnableSmartZoomInOut(Boolean bEnable) 36

6.2.7 public void FM\_EnableWalkingMode(Boolean bEnable) 36

6.2.8 public int FM\_GetAddress(Handler \_resultHandler, double xlon, double ylat) 37

6.2.9 public int FM\_GetPosFromScreen(int x, int y, double[] xlonlat) 37

6.2.10 public ArrayList<GPSPOS> FM\_GetLastGPSTrace() 37

6.2.11 public void FM\_DrawGrid(double[] x, double[] y) 38

6.2.12 public void FM\_SetMapCenter(float hCenter, float vCenter) 38

6.2.13 public void FM\_AutoSetMapCenter(boolean bMapMove) 38

6.2.14 public void FM\_SetMapPosition(FMBaseActivity.OnFatosMapListener onFatosMapListener, double xlon, double ylat) 38

6.2.15 public void FM\_GetTbtImageRes(int nTbtCode) 39

6.2.16 public void FM\_SetNaviVolume(int volume) 39

6.2.17 public void FM\_IsPlaying() 39

6.2.18 public void FM\_GetMapVersion(NaviVersion versionInfo) 39

6.2.19 public class NaviVersion 40

6.2.20 public class DBVersion 40

6.2.21 public void FM\_RequestUpdate(int nDownloadGubun, long freeSpace, final MapUpdateListenerCallback mapUpdateListener) 41

6.2.22 public interface MapUpdateListenerCallback 41

6.2.23 public void FM\_MapUpdateCancel() 41

6.2.24 41

6.3 경로 탐색 42

6.3.1 public void FM\_StartRGService(FMBaseActivity.OnFatosMapListener onFatosMapListener) 42

6.3.2 public ArrayList<RouteSummaryData> FM\_RouteSummary(FMBaseActivity.OnFatosMapListener onFatosMapListener) 42

6.3.3 public void FM\_SelectRoute(int nType) 42

6.3.4 public class RouteSummaryData 42

6.3.5 public ArrayList<RouteSummaryDataDetail> FM\_RouteSummaryDetail(int nType, int nFromIdx) 43

6.3.6 public class RouteSummaryDataDetail 44

6.3.7 public void FM\_CancelRoute() 44

6.3.8 public void FM\_SetSimulationSpeed(int nSpeed) 45

6.3.9 public List<RoutePosition>FM\_GetViaPOIList() 45

6.3.10 public void FM\_RouteVol2\_Via(Handler \_resultHandler, List<RoutePosition> m\_PoiList) 45

6.3.11 public class RoutePosition 46

6.3.12 public Boolean FM\_GoNextGoal() 47

6.3.13 public void FM\_StartSimulation () 47

6.3.14 public void FM\_StopSimulation () 47

6.3.15 public void FM\_GetMapCenterPos(double[] lonlat) 47

6.3.16 public void FM\_GetSubMapCenterPos(double[] lonlat) 47

6.3.17 48

6.4 POI 검색 48

6.4.1 public int FM\_SearchPOI(Handler \_resultHandler,String strSearchName, boolean bSaveRecent) 48

6.4.2 public int FM\_RecommendWord(Handler \_resultHandler,String strSearchName) 48

6.4.3 public ArrayList<POIItem> FM\_GetSearchResult(int nSortOption) 48

6.4.4 public ArrayList<String> FM\_GetRecommentWordResult() 49

6.4.5 49

6.5 49

# FATOS SDK코드 받기

GitHub에서 FATOS SDK for Android를 복제하거나 다운로드 하십시오.

# 개발 프로젝트 설정

Android Studio를 다운로드 하여 [설치하십시오](https://developer.android.com/studio/install.html?pkg=studio&hl=ko).  
(<https://developer.android.com/studio/index.html?hl=ko)>

FATOS SDK는 Android 5.0(API Level 20이상)을 지원합니다.

FATOS SDK for Androids AAR(Android Archive Package) File 형태로 배포 된다. 따라서 프로젝트 상에서 AAR 파일을 링크해 줘야 한다.

app/build.gradle에 libs 폴더의 aar file link 한다.   
(GitHub의 [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) 기준)

|  |
| --- |
| Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest)  repositories {  flatDir {  dirs 'libs'  }  }  dependencies {  implementation fileTree(dir: 'libs', include: ['\*.jar'])  implementation(name:'fire\_sdk\_vX.X.X', ext:'aar') |

FATOS SDK for Android는 다음 3가지의 필수 permission을 설정 해야 합니다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (TNaviSplashActivity.java)  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.M) {  int permissionResultWRITE\_EXTERNAL\_STORAGE = checkSelfPermission(Manifest.permission.WRITE\_EXTERNAL\_STORAGE);  int permissionResultACCESS\_FINE\_LOCATION = checkSelfPermission(Manifest.permission.ACCESS\_FINE\_LOCATION);  int permissionResultACCESS\_READ\_PHONE= checkSelfPermission(Manifest.permission.READ\_PHONE\_STATE);  if (permissionResultWRITE\_EXTERNAL\_STORAGE == PackageManager.PERMISSION\_DENIED  || permissionResultACCESS\_FINE\_LOCATION == PackageManager.PERMISSION\_DENIED  || permissionResultACCESS\_READ\_PHONE == PackageManager.PERMISSION\_DENIED) {  requestPermissions(new String[]{Manifest.permission.WRITE\_EXTERNAL\_STORAGE, Manifest.permission.ACCESS\_FINE\_LOCATION, Manifest.permission.READ\_PHONE\_STATE}, 1000);  }  else {  m\_Handler.postDelayed(startMainActivity, 10);  getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,  WindowManager.LayoutParams.FLAG\_FULLSCREEN);  }  }  else {  m\_Handler.postDelayed(startMainActivity, 10);  getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,  WindowManager.LayoutParams.FLAG\_FULLSCREEN);  }  } |

FATOS SDK for Android는 사용하려면 Project의 Application class와 MainActivity는 각각 FATOS SDK의 ANaviApplication Class와 FMBaseActivity Class를 상속 받아야 한다. 이를 통해 원활한 길안내를 할 수 있다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (TNaviApplication.java, TNaviMainActivity.java)  @Override  public class TNaviApplication extends ANaviApplication {  private Context m\_Context;  @Override  public void onCreate() {    public class TNaviMainActivity extends FMBaseActivity { |

FATOS SDK for Android는 Navi Engine 초기화를 반드시 해야 합니다. FMBaseActivity를 상속 받은 MainActivity의 onCreate() 시점에 하는 것을 추천합니다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (TNaviMainActivity.java)  //==============================================================================================  private void InitProcess() throws IOException {  m\_Context = this;  m\_gApp = (ANaviApplication) m\_Context.getApplicationContext();  m\_route = m\_gApp.getRouteApiInstance();  FMInterface.CreateInstance(m\_Context);  m\_FMInterface = FMInterface.GetInstance();  m\_iEngineInit = initFatosNaviEngine();  if(m\_iEngineInit != 1)  {  Intent intent = new Intent();  intent.setAction(TNaviActionCode.READY\_MAIN\_MAP); // Action name  sendBroadcast(intent);  return;  } |

# API key 받기 및 적용

API key 요청

[dev@fatoscorp.com](mailto:dev@fatoscorp.com) email로 API key 발급 요청하여 발급 받아야 합니다.

API key 적용

Application Class에서 발급 받은 API key를 onCreate() 시점에 FMInterface.initKey(..) 함수를 통해 반드시 설정해야 합니다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (TNaviApplication.java) public class TNaviApplication extends ANaviApplication {  private Context m\_Context;    @Override  public void onCreate() {  m\_Context = this;  FMInterface.initKey(m\_Context, "**FATOS SDK API key**");  super.onCreate();  ...  ... |

# FATOS SDK for Android 기능 구성

**지도**

FatosMap Fragment 제공

Map Event, 상태 정보 등에 대한 Callback 제공

유동적으로 FatosMap을 사용 할 수 있게 함.

Map Option 기능 제공.

지도 모드 변경, 좌표 이동, Zoom In/Out, Map Picker 등등에 대한 Map Control API 제공.

**검색**

FMS v1.01은 T map 검색 결과 제공

**경로 탐색**

FMS v1.01은 T map 경로 탐색 결과 제공

경로 탐색 데이터 관리 기능 제공

**경로 안내**

길안내 가로/세로 Fragment 제공 및 Custom Fragment 연동 기능 제공.

**Interface & Base Class제공.**

Map , Route Guidance 관련 기능 및 제어를 Base Class를 통해 기능 제공해 중복 작업을 줄이고, Navigation 기능을 손 쉽게 추가 하도록 한다.

# 코드 이해

# 지도 그리기

# Map Fragment 생성

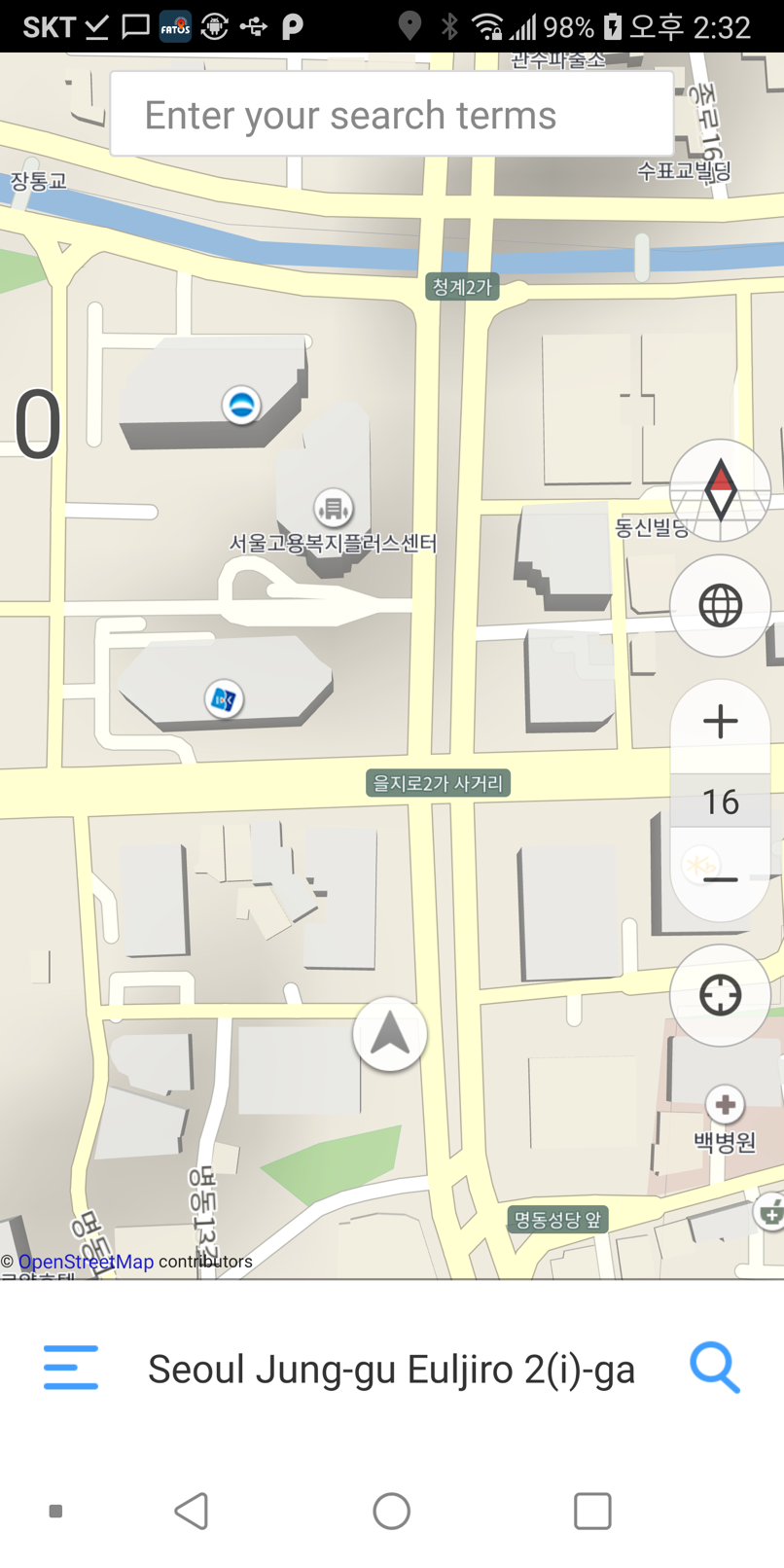
App에 FATOS map을 표출하기 위해서는   
AMapBaseFragment로 부터 상속 받은 Fragment를 생성해야 한다.   
해당 Fragment에서 Map Control, Map Setting 등의 기능이 동작 된다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (FatosMapFragment.java)  public class FatosMapFragment extends AMapBaseFragment  implements TNaviMainActivity.OnFatosMapListener,  FatosMainMapView.OnFatosMapListener {  …  …  } |

# 사용자 Activity에 지도 그리기

생성한 Map Fragment를 그리고 싶은 Activity에 추가하면 FATOS map이 그려진다.

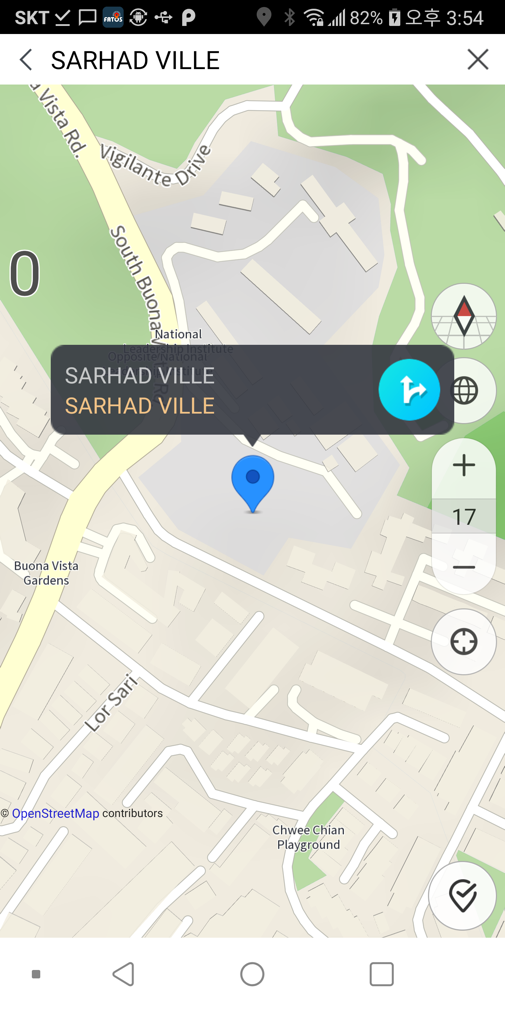
|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (activity\_tnavi\_main.xml)  <?xml version="1.0" encoding="utf-8"?>  <biz.fatossdk.exlib.fatoslayout.AMapSCRelativeLayout  xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:id="@+id/rl">  <biz.fatossdk.exlib.fatoslayout.AMapSCLinearLayout  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  >  <biz.fatossdk.exlib.fatoslayout.AMapSCRelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <fragment  android:id="@+id/map\_fragment"  **android:name="kr.fatos.tnavi.tnavifragment.FatosMapFragment"**  android:layout\_width="match\_parent"  android:layout\_height="match\_parent" /> |



# 지도 이동하기

WGS84 좌표를 전달해 해당 위치로 지도를 이동 할 수 있다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (TNaviMainActivity.java)  //지도보기에서 좌표이동  public void MovePoint(double dX, double dY, String i\_strAddr)  {  m\_FMInterface.FM\_SetMapPosition(0, dX, dY, 16.f);  fragmentManager = getSupportFragmentManager();  searchShowMapFragment = (SearchShowMapFragment)fragmentManager.findFragmentById(R.id.container);  } |



# 지도 모드 변경하기

2D, 3D, North up 등의 지도방향에 대한 모드와 항공/위성, 건물, 도로 등의 Map Object draw 변경/설정 할 수 있다.

m\_FMInterface.FM\_SetMapAirMode(FMInterface.MAPMODE\_AIR\_OFF\_BUILDING\_ON); // 항공모드 off 건물 on

public static final int MAPMODE\_AIR\_ON\_BUILDING\_ON = 0; // 항공모드 on, 건물 on

public static final int MAPMODE\_AIR\_ON\_BUILDING\_OFF = 1; // 항공모드 on, 건물 off

public static final int MAPMODE\_AIR\_OFF\_BUILDING\_ON = 2; // 항공모드 off, 건물 on



# 현 위치로 이동하기

지도 이동 상태에서 gps (또는 마지막 수신 위치) 좌표로 이동하는 기능을 제공한다.

public void FM\_MoveCurPosition(FMBaseActivity.OnFatosMapListener onFatosMapListener)

# Map Object Interface

Map Flicking, Two Touch Zoom In/out 등 지도 기능이 동작하기 위해서는 callback 함수에 대한 기능 정의가 되어야 한다.

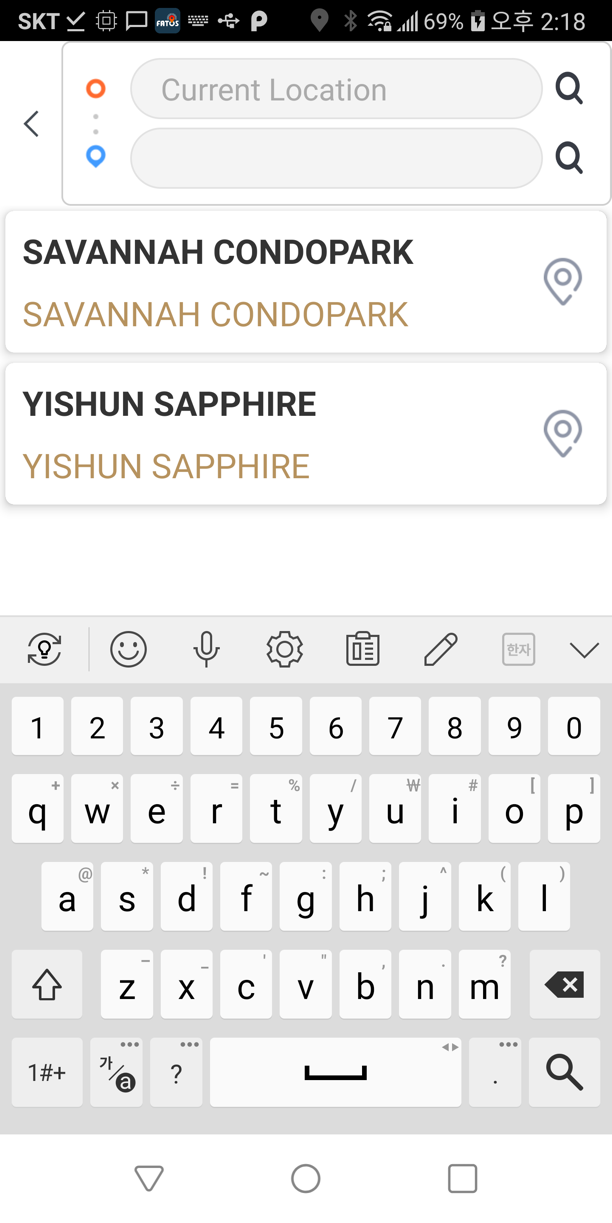
|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (FatosMapFragment.java)  public interface OnFatosMapListener {  void onCustomZoomInOutEnd(boolean bEnd);  void moveMapCurrentPos();  void mapMoveAniReset();  void mapMoveDirectCurPos();  void mapCenterUpdate();  void setMapMode();  void onMapAnimation(MapAnimation aniInfo);  void onCustomZoomInOut();  void onMapLevelInOut(float fLevel);  void onMapLevelInOut(float fLevel, float fTilt);  void onMapDrawPinImg(double x, double y, int nPinType); void onSetMapStatus(int nStatus); // 0 : 주행 , 1: 경로요약 } |

# 검색

# POI(Point Of Interest) 검색

발급 받은 API key 속성에 따라 모든 카테고리 기준으로 검색한다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (TNaviMainActivity.java)  public void getSearchPoiItem(String strMsg)  {  ArrayList<String> arInfo = new ArrayList<>();  arInfo.add(strMsg);  int index = prefs.getInt(SettingsCode.getKeyIndex(), 0);  String[] countrycodes;  countrycodes = getResources().getStringArray(R.array.wecountry\_codes);  arInfo.add(countrycodes[0]);  arInfo.add(String.valueOf(AMapPositionManager.getCurrentLonX()));  arInfo.add(String.valueOf(AMapPositionManager.getCurrentLatY()));  m\_FMInterface.FM\_SearchPOIForTNavi(new HttpResultHandler(TNaviMainActivity.this), arInfo, false);  } |



# 검색 결과

FM\_SearchPOIForTNavi 통해 검색 되는 POI의 결과는 전달 Handler를 통해 전달 됩니다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (TNaviMainActivity.java)  private void handleMessage(Message msg) {  String result = msg.getData().getString(AMapGoogleSearchUtil.RESULT);  ArrayList<String> searchList = new ArrayList<String>();  if (result.equals(FMError.FME\_SUCCESS\_SEARCH\_SUCCESS)) {  searchResultPOI = m\_FMInterface.FM\_GetSearchResult(FMSortOption.FM\_SORT\_BY\_DIST);  if(searchResultPOI != null)  {  SearchFragment sf = (SearchFragment) getSupportFragmentManager().findFragmentById(R.id.container);  sf.MergePOIItem(searchResultPOI, result);  } |

검색 되는 POI의 정보입니다. POI에 따라 존재하는 항목과 존재하지 않는 항목이 있을 수 있습니다.

하지만 좌표와 명칭은 필수로 존재합니다.

|  |
| --- |
| public class POIItem {  // poi data items  public String id = ""; // poi id  public String name = ""; // poi name  public String telNo; // poi telephone  public String frontLat = null; // 입구점 좌표  public String frontLon = null; // 입구점 좌표  public String noorLat; // 중심 좌표  public String noorLon; // 중심 좌표  …  …  public int nFromCurPosDist; // 현위치에서 해당 위치까지의 거리 (meter 단위)    public String secondBuildNo;  public String radius;  public String bizName;  public String upperBizName;  public String middleBizName;  public String lowerBizName;  public String detailBizName;  public String rpFlag = "";  public String parkFlag;  public String detailInfoFlag;  public String desc; |

# 경로 탐색

# 현 위치 기준 경로탐색

FM\_RouteVol2\_Via api를 통해 현 위치 기준 경로 탐색을 제공합니다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (TNaviMainActivity.java)  //목적지로 경로 탐색  public void routeTovia(double x, double y, String flag, @Nullable NPoiItem i\_PoiItem){  show\_ProgressDialog(R.string.string\_higetting\_search\_direction , false);  if(i\_PoiItem != null) {  m\_RecentPOIItem = i\_PoiItem;  if (i\_PoiItem.getAddressFull() == null){  i\_PoiItem.setAdminLevel1LocalName("");  }  if(i\_PoiItem.getEnglishName() == null){  i\_PoiItem.setEnglishName("");  }  if(i\_PoiItem.getAddressFull().equals("") || i\_PoiItem.getEnglishName().equals("")){  String address\_wgs = String.format("%.5f",i\_PoiItem.getLocationPointX()) + ", " +  String.format("%.5f",i\_PoiItem.getLocationPointY());  m\_RecentPOIItem.setAdminLevel1EnglishName("No Address Name");  m\_RecentPOIItem.setEnglishName(address\_wgs);  }  }  m\_FMInterface.FM\_GetViaPOIList().clear();  List<RoutePosition> positionList = m\_FMInterface.FM\_GetViaPOIList();  RoutePosition positionList0 = new RoutePosition();  RoutePosition positionList1 = new RoutePosition();  switch(flag){  case TNaviActionCode.JUST\_GOAL :  positionList.clear();  positionList0.x = this.m\_dStartCoord[0];  positionList0.y = this.m\_dStartCoord[1];  positionList0.name = this.strAddr[0];  positionList.add(positionList0);  positionList1.x = x;  positionList1.y = y;  positionList1.name = this.strAddr[1];  if(this.strAddr[1].equals("")){  positionList1.name = "Goal";  }  positionList1.bPassingPoint = false;  positionList.add(positionList1);  saved\_data.clearData();  saved\_data.viaX = positionList0.x;  saved\_data.viaY = positionList0.y;  saved\_data.goalX = x;  saved\_data.goalY = y;  break;  ...  ...  }  m\_FMInterface.FM\_RouteVol2\_Via(new HttpResultHandler(TNaviMainActivity.this), positionList);  } |

# 출발지 설정 경로 탐색

GPS 현 위치가 아닌 출발지 좌표 설정으로 해당 출발지 부터 목적지 까지의 경로 탐색을 요청 할 수 있습니다.

|  |
| --- |
| public void setStartCoord(double i\_dX, double i\_dY)  {  if(i\_dX <= 0 || i\_dY <= 0)  {  this.m\_dStartCoord[0] = m\_FMInterface.FM\_GetDriveInfo().getM\_fCurLonX();  this.m\_dStartCoord[1] = m\_FMInterface.FM\_GetDriveInfo().getM\_fCurLatY();  }  else {  this.m\_dStartCoord[0] = i\_dX;  this.m\_dStartCoord[1] = i\_dY;  }  }  …  …  positionList.clear();  positionList0.x = this.m\_dStartCoord[0];  positionList0.y = this.m\_dStartCoord[1];  positionList0.name = this.strAddr[0];  positionList.add(positionList0);  m\_FMInterface.FM\_RouteVol2\_Via(new HttpResultHandler(TNaviMainActivity.this), positionList); |

# 경로 정보

# 경로 요약 정보

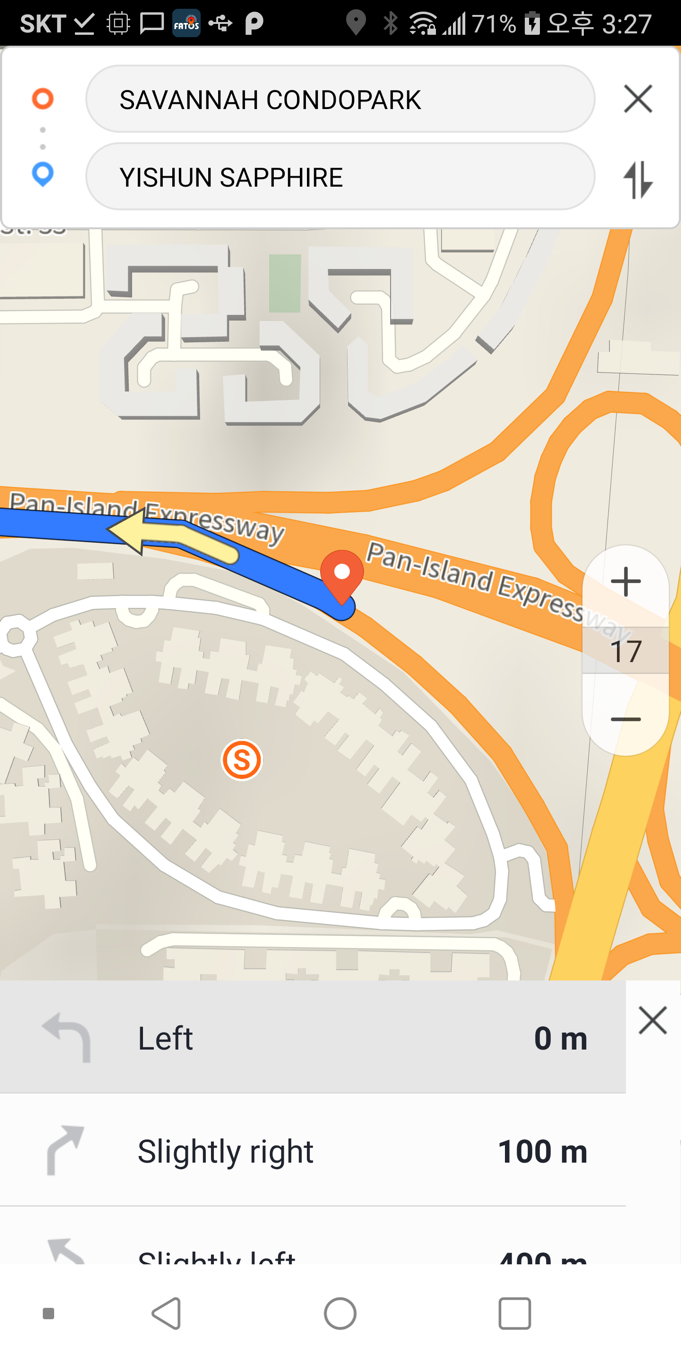
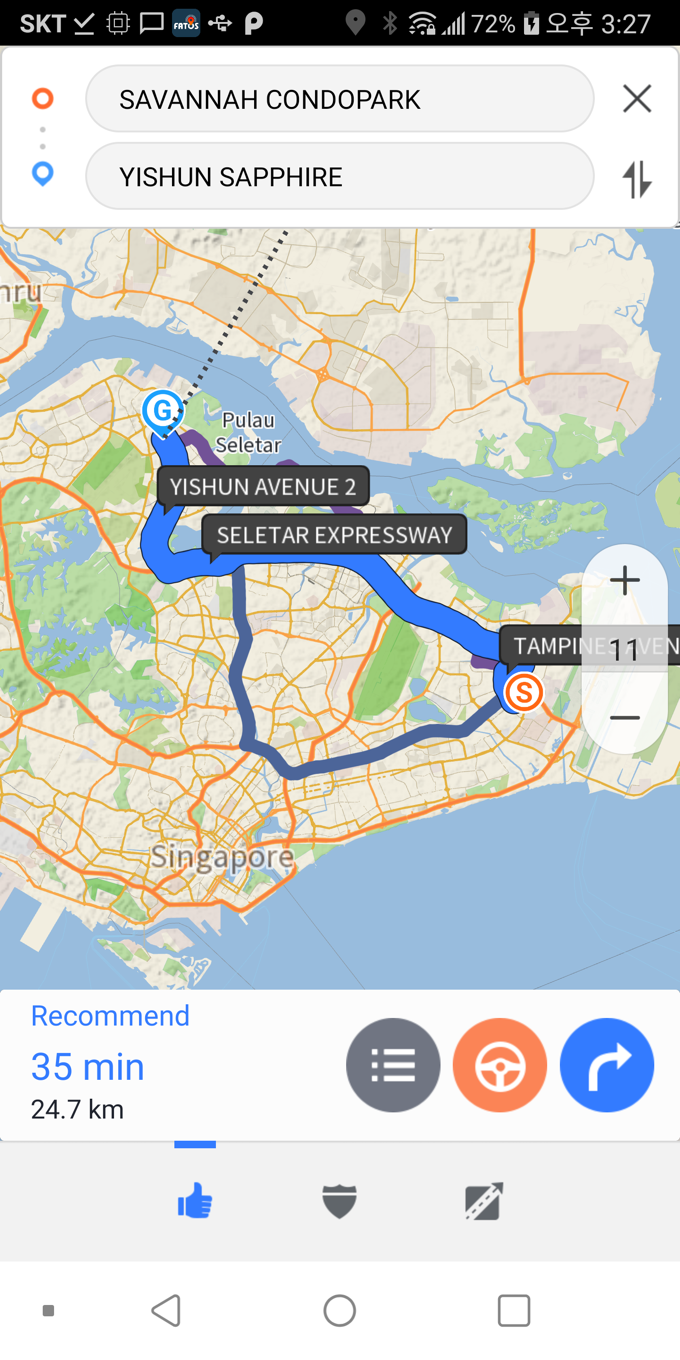
FM\_RouteVol2\_Via 에 전달 되는 Handler를 통해 경로 탐색 결과를 알 수 있으며, 성공 했을 때 경로 요약 정보를 제공합니다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (TNaviMainActivity.java) //경탐 완료시 실행,  private void routeSuccess(){  ArrayList<RouteCardData> summaryCardData = new ArrayList<RouteCardData>();  ArrayList<RouteSummaryData> summaryData = m\_FMInterface.FM\_RouteSummary(FMBaseActivity.onFatosMapListener);  for(int i=0; i<summaryData.size(); i++ ){  RouteSummaryData summarydata = summaryData.get(i);  //String strTypeName, int nType, int nLength, int nTime, int nFee, int nAvgSpeed, int nTurnCongestion, int nOptionColor  RouteCardData data = new RouteCardData(summarydata.strTypeName,summarydata.nType,summarydata.nLength,summarydata.nTime,summarydata.nFee,  summarydata.nAvgSpeed,  summarydata.nTurnCongestion,m\_Context);  summaryCardData.add(data);  }  saved\_summaryCardData = new ArrayList<RouteCardData>();  saved\_summaryCardData = summaryCardData;  if(!bLastRouteFlag) {  setAPP\_MODE(TNaviActionCode.APP\_MODE\_SHOWING\_SUMMARY);  fragmentManager = getSupportFragmentManager();  Bundle args = new Bundle();  args.putParcelableArrayList(TNaviActionCode.LONGTOUCH\_ROUTESUMMARY, summaryCardData);  SummaryFragment summaryFragment = new SummaryFragment();  summaryFragment.setArguments(args);  GoLib.getInstance().goFragment(getSupportFragmentManager(), R.id.container,summaryFragment,tag\_summary\_fragment,args);  }  } |

# 경로 상세 정보

FATOS SDK Sample에서는 AMapSDKCommonRGInfoFragment.java를 통해 요약 관련 화면을 제공하고 있습니다.

|  |
| --- |
| #Sample in [FATOS SDK Guide App for Android](https://github.com/devfatoscorp/FatosSDKTest) (activity\_tnavi\_main.xml)  <biz.fatossdk.exlib.fatoslayout.AMapSCLinearLayout  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  >  <biz.fatossdk.exlib.fatoslayout.AMapSCRelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <fragment  android:id="@+id/map\_fragment"  android:name="kr.fatos.tnavi.tnavifragment.FatosMapFragment"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent" />  <fragment  android:id="@+id/drive\_ui\_fragment"  android:name="**biz.fatossdk.fminterface.drivefragment.AMapSDKCommonRGInfoFragment**"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent" />  <ImageView  android:layout\_width="28dp"  android:layout\_height="28dp"  android:id="@+id/imageView\_MapCenterPoint"  android:src="@drawable/img\_mapcenter"  android:layout\_centerInParent="true"  android:visibility="gone"/>  </biz.fatossdk.exlib.fatoslayout.AMapSCRelativeLayout>  </biz.fatossdk.exlib.fatoslayout.AMapSCLinearLayout>  ….  ….  private int m\_nUIMode = ANaviApplication.UI\_DRIVE\_MODE;  public void setUIMode(int nUIMode) {  m\_nUIMode = nUIMode;  switch (m\_nUIMode)  {  case ANaviApplication.UI\_DRIVE\_MODE:  {  m\_CurSpeed.setVisibility(View.VISIBLE);  m\_AerialLayout.setVisibility(View.VISIBLE);  m\_CommpassLayout.setVisibility(View.VISIBLE);  m\_layoutCurPos.setVisibility(View.VISIBLE);  }break;  case ANaviApplication.UI\_SUMMARY\_MODE:  {  m\_CurSpeed.setVisibility(View.INVISIBLE);  m\_AerialLayout.setVisibility(View.GONE);  m\_CommpassLayout.setVisibility(View.GONE);  m\_layoutCurPos.setVisibility(View.GONE);  }break;  case ANaviApplication.UI\_SIMUL\_MODE:  {  m\_CurSpeed.setVisibility(View.VISIBLE);  m\_AerialLayout.setVisibility(View.VISIBLE);  m\_CommpassLayout.setVisibility(View.VISIBLE);  m\_layoutCurPos.setVisibility(View.VISIBLE);  }break;  case ANaviApplication.UI\_NORMAL\_DRIVE\_MODE:  {  m\_CurSpeed.setVisibility(View.VISIBLE);  m\_AerialLayout.setVisibility(View.VISIBLE);  m\_CommpassLayout.setVisibility(View.VISIBLE);  m\_layoutCurPos.setVisibility(View.VISIBLE);  }break;  }  } |

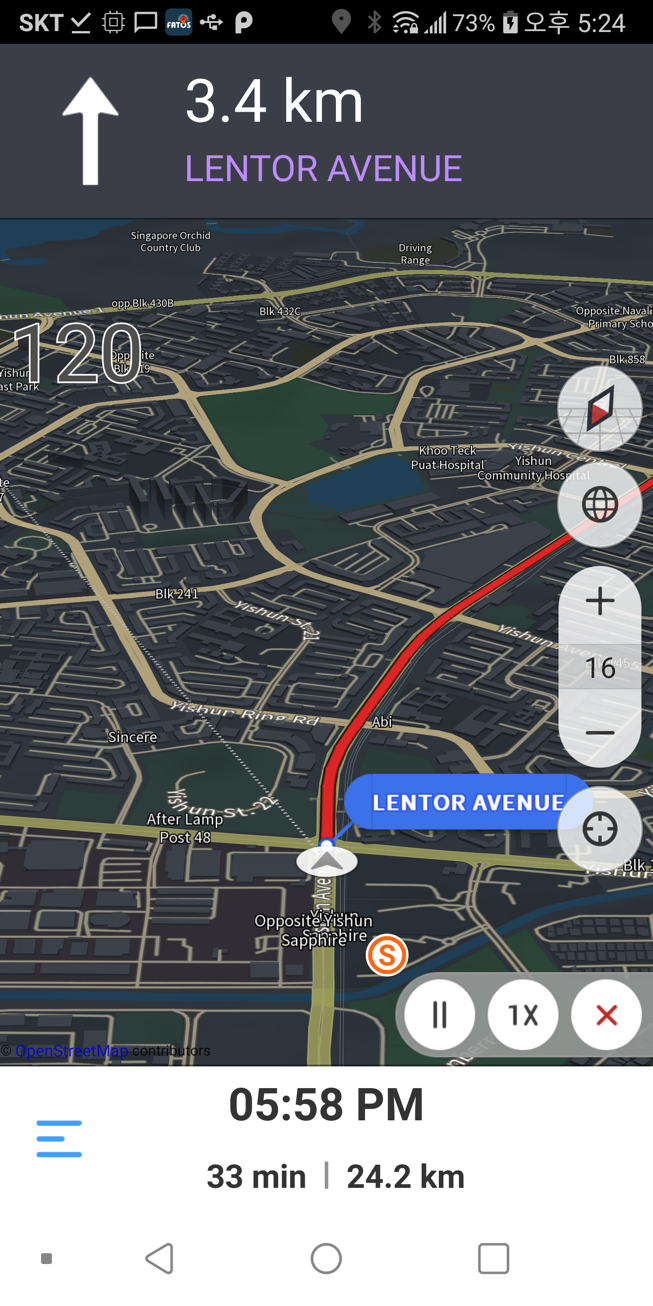


# 경로 안내

# 경로안내 시작

경로 탐색이 완료 된 상태에서 FM\_StartRGService를 통해 경로 안내를 받을 수 있다.

|  |
| --- |
| m\_FMInterface.FM\_StartRGService(FMBaseActivity.onFatosMapListener) |



# 경로 정보

주행 중 FM\_GetDriveInfo()를 통해 주행 관련 정보를 얻을 수 있다.

|  |
| --- |
| **public class FMDriveInfo**  private int m\_iGpsSpeed;   * gps speed   private boolean m\_bIsRoute;   * true: 경로 주행 중 false : 일반 주행 중   private String m\_strCurPosName;   * 주행 중인 행정동명   private double m\_fCurLonX = -180;   * 현위치 좌표(경도, x)   private double m\_fCurLatY = -180;   * 현위치 좌표(위도, y)   private int m\_nCurAngle = 0;   * 현위치 좌표(각도)   private int m\_nGpsStatus = 0;   * gps 상태 값   0 : bad  1 : simulation  2 : good  3: test  good 일 때의 좌표만 유효한 값임.  private boolean m\_bCannotEnterRoad = false;   * 진입 불가 도로   private int m\_nCannotEnterRoadCode = 0;   * 진입 불가 도로   private int m\_nGateNearbyDist = 0;   * 게이트 주변 거리   private int m\_nTotalRemainderDist= 0;   * 목적지까지 남은 거리   private int m\_nServiceLinkRemainderTime= 0;   * 목적지까지 남은 시간   public int nViaTotalCount = 0;   * 경유 목적지 전체 수   public int[] listViaRemainderDist;   * 경유 목적지까지 남은 거리(미터 단위)   public int[] listViaRemainderTime;   * 경유 목적지까지 남은 시간(초 단위)   private int m\_nTripTime = 0;   * 총 주행 시간   private int m\_nTripDist= 0;  총 주행 거리 |

# API 목록

# 초기화 과정

# public int initFatosNaviEngine()

|  |  |
| --- | --- |
| 설명 | FATOS SDK 를 사용하는 App에서 처음 실행 시 FMBaseActivity의 initFatosNaviEngine을 호출해야 한다. |
| Parameter & Return | 1 : 성공  INIT\_ERROR\_JNI = -1; // -1: JNI 초기화 실패  INIT\_ERROR\_SDL = -2; // -2 : SDL 초기화 실패  INIT\_ERROR\_RP = -3; // -3 : 네트워크 초기화 실패  INIT\_ERROR\_FORCEUPDATE = -4; // -4 : 강제 업데이트  INIT\_ERROR\_DAL = -5; // -5 : DAL 기타 에러  INIT\_ERROR\_CONFIG = -6; // -6 :컨피그 로딩 실패  INIT\_ERROR\_LOCAL\_SEARCH\_DB = -7;//-7 : 로컬(검색) DB 없음  INIT\_ERROR\_LOCAL\_RP\_DB = -8; //-8 : 도로(검색) DB 없음 |

# 

# public int FM\_Init(Context context, String strApiKey)

|  |  |
| --- | --- |
| 설명 | FMS를 사용하는 App에서 초기화 시에 반드시 호출 해야 한다. |
| Parameter & Return | Context : Application Context를 전달.  Return : FMError.Success -> 성공  FMError.Error -> 실패 |

# public int FM\_Destory()

|  |  |
| --- | --- |
| 설명 | App 종료 시에 호출 되야 하는 함수 |
| Parameter & Return | Return : FMError.Success -> 성공  FMError.Error -> 실패 |

# 지도

# MapStatusListener

|  |  |
| --- | --- |
| 설명 | 경로 안내 종료 / Map 상태 정보에 interface |
| Parameter & Return | public void onRouteFinish()  public void onMapMove(boolean bMapMove) |

# public void setMainMapOption(boolean bDrawStartImg,boolean bDrawGoalImg,boolean bDrawPinImg,boolean bDrawEVWare)

|  |  |
| --- | --- |
| 설명 | Map Option 상태를 설정하는 |
| Parameter & Return | 함수 NativeNavi.MAP\_OBJ\_FONTTBT,   * Font TBT Draw 여부 설정   NativeNavi.MAP\_OBJ\_FLAG\_START,   * 출발지 깃발 Draw 여부 설정   NativeNavi.MAP\_OBJ\_FLAG\_GOAL,   * 목적지 깃발 Draw 여부 설정   NativeNavi.MAP\_OBJ\_FLAG\_PIN   * poi Pin image draw 여부 설정   NativeNavi.MAP\_OBJ\_EVWHERE,   * Map에 전기차 정보 object draw 여부 설정   NativeNavi.MAP\_OBJ\_ADVENCED\_TBT,   * Advance tbt draw 여부 설정   NativeNavi.MAP\_OBJ\_SCALEBAR  Scale bar draw 여부 설정 |

# public void onMapReady()

|  |  |
| --- | --- |
| 설명 | FMS를 통해 Map 생성 후 Map Drawing할 준비가 되어 있을 때 오는 Callback 함수로 해당 Callback을 받은 수 Map Control Api를 호출해야 한다. |
| 예재 | @Override  public void onMapReady() {  ((FMBaseActivity)getActivity()).onMapReady();  } |

# public void FM\_MoveCurPosition(FMBaseActivity.OnFatosMapListener onFatosMapListener)

|  |  |
| --- | --- |
| 설명 | 현 위치로 이동 |
| 예재 | m\_FMInterface.FM\_MoveCurPosition(FMBaseActivity.OnFatosMapListener) |

# public void FM\_SetMapAirMode(int nMapType)

|  |  |
| --- | --- |
| 설명 | 항공모드 설정 |
| Parameter & Return | nMapType :  FMInterface.MAPMODE\_AIR\_ON\_BUILDING\_ON; // 항공모드 on 건물 on  FMInterface.MAPMODE\_AIR\_ON\_BUILDING\_OFF; // 항공모드 on 건물 off  FMInterface.MAPMODE\_AIR\_OFF\_BUILDING\_ON; // 항공모드 off 건물 on |

# public void FM\_EnableSmartZoomInOut(Boolean bEnable)

|  |  |
| --- | --- |
| 설명 | 주행 중 자동 축척 모드 사용 할지에 대한 여부 설정 |
| Parameter & Return | bEnable  true : 자동 주행 모드 on  false : 자동 주행 모드 off |

# public void FM\_EnableWalkingMode(Boolean bEnable)

|  |  |
| --- | --- |
| 설명 | 도보 모드 사용 할지에 대한 여부 설정 |
| Parameter & Return | bEnable  true : 도보 모드 on  false : 도보 모드 off |

# public int FM\_GetAddress(Handler \_resultHandler, double xlon, double ylat)

|  |  |
| --- | --- |
| 설명 | 기준 좌표 기준 행정동명을 얻을 때. |
| Parameter & Return | #Param \_resultHandler : 검색 결과 또는 상태를 받을 handler  xlon, ylat : 주소를 얻으려는 좌표 |

# public int FM\_GetPosFromScreen(int x, int y, double[] xlonlat)

|  |  |
| --- | --- |
| 설명 | 화면 터치 point의 wgs84 좌표를 반환 한다. |
| Parameter & Return | #Param x,y : 지도 터치한 화면 position  xlonlat : 터치 위치에 대한 wgs84 좌표 |

# public ArrayList<GPSPOS> FM\_GetLastGPSTrace()

|  |  |
| --- | --- |
| 설명 | 경로안내 시작 부터의 GPS 궤적 정보를 가져온다 |
| Parameter & Return | .#Return  ArrayList<GPSPOS> : GPS 궤적 정보 List |

# public void FM\_DrawGrid(double[] x, double[] y)

|  |  |
| --- | --- |
| 설명 | 지도에 그리드를 그린다. |
| Parameter & Return | #Param  x,y : 그리드 좌표 리스트 |

# public void FM\_SetMapCenter(float hCenter, float vCenter)

|  |  |
| --- | --- |
| 설명 | 왼쪽 하단 0.f, 0.f 오른쪽 상단 : 1.f, 1.f 기준 중심을 설정한다.  ex) 0.5f, 0.5f 는 지도 화면 중심 위치 |
| Parameter & Return | #Param  hCenter : 지도 영역의 가로 위치  vCenter : 지도 영역의 세로 위치 |

# public void FM\_AutoSetMapCenter(boolean bMapMove)

|  |  |
| --- | --- |
| 설명 | bMapMove 값에 따라 지도 중심을 주행모드와 지도 이동 모드로 구분해 이동 시킨다. |
| Parameter & Return | #Param  bMapMove : 현재 맵 이동 상태인지에 대한 여부 |

# public void FM\_SetMapPosition(FMBaseActivity.OnFatosMapListener onFatosMapListener, double xlon, double ylat)

|  |  |
| --- | --- |
| 설명 | bMapMove 값에 따라 지도 중심을 주행모드와 지도 이동 모드로 구분해 이동 시킨다 |
| Parameter & Return | .#Param  onFatosMapListener : Mapview와 연결된 listener  xlon, ylat : 이동 하려는 위치의 WGS84 좌표 |

# public void FM\_GetTbtImageRes(int nTbtCode)

|  |  |
| --- | --- |
| 설명 | TBTCode를 받아서 해당 이미지 리소스 ID를 return |
| Parameter & Return | #Param  nTbtCode : summary 정보에 있는 Tbtcode |

# public void FM\_SetNaviVolume(int volume)

|  |  |
| --- | --- |
| 설명 | 기본 음성의 볼륨을 조절 한다. |
| Parameter & Return | #Param  volume : 볼륨 크기(0~100) – 0은 mute로 본다. |

# public void FM\_IsPlaying()

|  |  |
| --- | --- |
| 설명 | 현재 기본 음성이 발생 중인지 체크 한다 |
| Parameter & Return | . #return  true : 기본 음성이 발성 중 일 때  false : 기본 음성이 발성 중이 아닌 상태 |

# public void FM\_GetMapVersion(NaviVersion versionInfo)

|  |  |
| --- | --- |
| 설명 | 현재 배경, 검색, 경로탐색, 기타(SDI,EV..)의 Map 이름/Version, 업데이트 가능 여부, 업데이트 상태 등의 정보를 얻어 온다. |
| Parameter & Return | #Param  NaviVersion 구조체 정보 참고 |

# public class NaviVersion

|  |
| --- |
| Map Data의 정보를 담고 있는 구조체 |
| int nMapDate;   * 배경 DB 빌드날짜   DBVersion[] listMap;   * 배경 DB 정보 리스트   int nSearchDate;   * 검색 DB 빌드날짜   DBVersion[] listSearch;   * 검색 DB 정보 리스트   int nNetworkDate;   * 경로탐색 DB 빌드날짜   DBVersion[] listNetwork;   * 경로탐색 DB 정보 리스트   DBVersion[] listEtc;   * 기타 DB 정보 리스트 (SDI, EV...) |

# public class DBVersion

|  |
| --- |
| 배경, 검색, 경로탐색, 기타(SDI,EV..) 각각의 DB 정보를 담고 있는 구조체 |
| String strName;   * DB 이름   int nCurDate;   * 현재 DB 빌드날짜   int nUpdateDate;   * 업데이트 가능한 DB 빌드날짜   double fVersion;   * 버전   boolean bUpdate;   * 업데이트 가능 여부   int nUpdateStatus;   * 0:다운로드안됨, 1:다운로드중, 2:다운로드완료 |

# public void FM\_RequestUpdate(int nDownloadGubun, long freeSpace, final MapUpdateListenerCallback mapUpdateListener)

|  |  |
| --- | --- |
| 설명 | **DBVersion** 기준 정보의 업데이트 가능 여부 (bUpdate) 값 기준으로 각 구분 별 Map Download를 요청한다. |
| Parameter & Return | #Param  nDownloadGubun : 배경, 검색, 경로탐색, 기타(SDI,EV..)의 구분  # Gubun값 정의  public static final int MAP\_DOWNLOAD\_BASEMAP = 1;  public static final int MAP\_DOWNLOAD\_SEARCH = 2;  public static final int MAP\_DOWNLOAD\_RP = 3;  public static final int MAP\_DOWNLOAD\_ANDO = 4;  freeSpace : 현재 Map DB가 있는 storage의 남은 용량  mapUpdateListener : update 상태 정보를 받기 위한 Listener |

# public interface MapUpdateListenerCallback

|  |
| --- |
| 배경, 검색, 경로탐색, 기타(SDI,EV..) 각각의 DB 정보를 담고 있는 구조체 |
| void onData(mapDownloadResult resultMapDownload);  void onError(int nErrorCode); |

# public void FM\_MapUpdateCancel()

|  |  |
| --- | --- |
| 설명 | 맵 업데이트 중 취소 할 때 호출한다. |

# 

# 경로 탐색

# public void FM\_StartRGService(FMBaseActivity.OnFatosMapListener onFatosMapListener)

|  |  |
| --- | --- |
| 설명 | 경로요약에서 경로 선택 후 주행 화면으로 갈 때 호출해야 한다. |
| Parameter & Return | onFatosMapListener : Mapview와 연결된 listener |

# public ArrayList<RouteSummaryData> FM\_RouteSummary(FMBaseActivity.OnFatosMapListener onFatosMapListener)

|  |  |
| --- | --- |
| 설명 | 경로 탐색 성공 후에 각 경로 옵션에 대한 정보를 얻기 위해 호출한다. |
| Parameter & Return | Parm: onFatosMapListener : Mapview와 연결된 listener  Return : 각 경로 옵션에 대한 정보 list |

# public void FM\_SelectRoute(int nType)

|  |  |
| --- | --- |
| Parameter & Return | # Parm  nType : RouteSummaryData의 type 정보 |
| 설명 | 경로요약 화면에서 경로 옵션에 대해 선택 시 호출해 준다. |

# public class RouteSummaryData

|  |  |
| --- | --- |
| 설명 | POI 검색 결과 구조체  public String strTypeName;   * 경로탐색 옵션 타이틀   public int nType;   * 경로탐색 옵션 (최소, 최단, 추천 등)   public int nLength;   * 경로 총 길이 (단위 미터)   public int nTime;   * 경로 총 소요시간 (단위 초)   public int nFee;   * 요금 정보 (원)   public int nAvgSpeed;   * 구간 평균 속도   public int nTurnCongestion;   * 구간 혼잡도   public int nOptionColor;   * 경로 옵션 Color |
| Parameter & Return | 경로 요약 정보 |

# public ArrayList<RouteSummaryDataDetail> FM\_RouteSummaryDetail(int nType, int nFromIdx)

|  |  |
| --- | --- |
| 설명 | 각 Type별 세부 경로 정보를 전달한다.. |
| Parameter & Return | # Parm  nType : RouteSummaryData의 type 정보  nFromIdx : 현재 RG Index  ( nFromIdx는 샘플과 같이 public void updateServiceLinkIdx(int nIdx, boolean bReset)에서 얻어야 한다. ) |

# public class RouteSummaryDataDetail

|  |  |
| --- | --- |
| 설명 | public int nTbtCode;   * tbt code (1~53) * SDK에 포함 된. R.drawable.list\_01\_straight ~ R.drawable.list\_53\_via3 사용   public int nLength;   * 해당 Turn 길이 (단위 미터)   public int nTime;   * 해당 Turn 통과시간 (단위 초)   public double dLonX;  public double dLatY;   * Turn 지점 좌표 정보   public String strDongName;   * Turn 지점 행정동명   public int nTurnCongestion;   * 구간 혼잡도   public int nAvgSpeed;   * 구간 평균 속도   public String strTurnInfoTextName;   * Turn Text Info |
| Parameter & Return | 경로 요약 상세 정보 |

# public void FM\_CancelRoute()

|  |  |
| --- | --- |
| 설명 | 경로가 있을 경우 해당 경로를 취소한다.  (경로가 취소 되면서 경로선/출,목 깃발 등을 그리지 않는다)  취소 후에는 onRouteFinish Callback 호출 된다. |

# public void FM\_SetSimulationSpeed(int nSpeed)

|  |  |
| --- | --- |
| 설명 | 모의 주행 중 속도를 설정한다. |
| Parameter & Return | nSpeed : 모의주행 속도 |

# public List<RoutePosition>FM\_GetViaPOIList()

|  |  |
| --- | --- |
| 설명 | 경유지 탐색을 위한 Poi 정보 list로 경로탐색 요청시 최소 2개 이상 설정 되어 있어야 한다. |
| Parameter & Return | Return : 좌표 정보를 담을 list return |

# public void FM\_RouteVol2\_Via(Handler \_resultHandler, List<RoutePosition> m\_PoiList)

|  |  |
| --- | --- |
| 설명 | M\_PoiList에 설정된 좌표 기준으로 경유지 탐색 한다.  ( 0 idx : 출발지 , 마지막 idx : 목적지, 그 외 경유지)  \_resultHandler의 에러 코드 중 FMError.FME\_ERROR\_VIACOUNT\_ROUTE는 좌표 리스트가 2개 미만 일 때 발생함.  Ex)  RoutePosition positionList0 = new RoutePosition();  positionList0.x = 0;  positionList0.y = 0;  positionList0.name = "출발지";  positionList.add(positionList0);  RoutePosition positionList1 = new RoutePosition();  positionList1.x = 127.46534606;  positionList1.y = 37.28005176;  positionList1.name = "이천시법원";  positionList1.addr = "경기 이천시 안흥동";  positionList.add(positionList1);  RoutePosition positionList2 = new RoutePosition();  positionList2.x = 127.43412687;  positionList2.y = 37.27288535;  positionList2.name = "이천시청";  positionList2.addr = "경기 이천시 중리동";  positionList.add(positionList2);  RoutePosition positionList3 = new RoutePosition();  positionList3.x = 127.46534606;  positionList3.y = 37.28005176;  positionList3.name = "이천종합버스터미널";  positionList3.addr = "경기 이천시 안흥동";  m\_FMInterface.FM\_RouteVol2\_Via(new  HttpResultHandler(FatosMapSDKSampleLandFragment.this),positionList |
| Parameter & Return | resultHandler : 검색 결과 또는 상태를 받을 handler  m\_PoiList : 출,경,목 poi 정보 List |

# public class RoutePosition

|  |  |
| --- | --- |
| 설명 | 경유지 poi의 좌표를 담고 있는 구조체  아래의 정보 이외의 정보는 SDK 상에서 쓰지 않는다.  public int nPosKind; // 0 출발지 , 1목적지, 2 입력 대기  public double x = 0;  public double y = 0;   * Poi wgs84 좌표   public String name; // 명칭 ex) “강남역(2호선)”  public int nPoiID; // POI ID (없을 경우 0)  public String poiFlag; // POI RPFlag ( 없을 경우 0)  public int nFloor = 1; // 층수(1=1층, -2=지하2층, ...)  public boolean bDir = true; // 방향성체크  public String addr; // 주소 정보(없을 경우 null)  public Boolean bPassingPoint = false; // 경로만 잡는 경유지 인지 설정 |

# public Boolean FM\_GoNextGoal()

|  |  |
| --- | --- |
| 설명 | 경유지 주행에서 바로 앞 경유지를 패스하고 다음 경유지 부터 경로 잡고 주행 할 때 사용한다. |
| Parameter & Return | # Parm  return : 정상적으로 다음경유지로 설정된 경우 : true  다음경유지로 설정하지 못하는 경우 : false  (목적지 밖에 안 남았을 경우) |

# public void FM\_StartSimulation ()

|  |  |
| --- | --- |
| 설명 | 경로 탐색 후 호출 시 모의 주행 한다. |

# public void FM\_StopSimulation ()

|  |  |
| --- | --- |
| 설명 | 모의 주행을 종료하고 주행 모드로 돌아 갈 때 호출 한다 |

# public void FM\_GetMapCenterPos(double[] lonlat)

|  |  |
| --- | --- |
| 설명 | 지도 중심의 WGS84 좌표를 얻는다. |
| Parameter & Return | 지도 중심 좌표를 전달 받을 double array |

# public void FM\_GetSubMapCenterPos(double[] lonlat)

|  |  |
| --- | --- |
| 설명 | 서브 맵 WGS84 좌표를 얻는다. |
| Parameter & Return | 지도 중심 좌표를 전달 받을 double array |

# 

# POI 검색

# public int FM\_SearchPOI(Handler \_resultHandler,String strSearchName, boolean bSaveRecent)

|  |  |
| --- | --- |
| 설명 | 특정 명칭으로 검색 결과를 얻고자 할 때의 API |
| Parameter & Return | \_resultHandler : 검색 결과 또는 상태를 받을 handler  strSearchName : POI 검색 명칭  bSaveRecent : 최근 검색 결과 목록에 저장 할지 여부 |

# public int FM\_RecommendWord(Handler \_resultHandler,String strSearchName)

|  |  |
| --- | --- |
| 설명 | 초성 포함 검색 문자에 따른 추천어 리스트를 준다. |
| Parameter & Return | strSearchName : 추천어 검색 문자(초성포함) |

# public ArrayList<POIItem> FM\_GetSearchResult(int nSortOption)

|  |  |
| --- | --- |
| 설명 | 통합 검색 결과를 가져 올 때 쓴다. nSortOption 값에 따라 정렬한다. |
| Parameter & Return | nSortOption  0 : 정확도 검색  1 : 거리순 검색 |

# public ArrayList<String> FM\_GetRecommentWordResult()

|  |  |
| --- | --- |
| 설명 | 추천어 검색 결과를 가져 온다. |

# 

# 