Navi SDK 集成指南

# 简介

该SDK主要用于向极氪内部应用或者三方接入应用提供导航服务接口的能力。

# 平台支持

|  |  |
| --- | --- |
| **Item** | **SDK兼容说明** |
| Java | JDK 1.7 以上 |
| Android SDK | Android 5.0 (apiLevel >= 16) 以上 |
| 芯片架构 | armv7或arm64 |

# 名词解释

|  |  |
| --- | --- |
| **名称** | **描述** |
| SAPA | 服务区、停车区 |
| ETA | 预计到达时间 |
| TBT | 逐向导航 |
| POI | 兴趣点 |
| GPS | 全球定位系统 |
| GCJ02 | 火星坐标 |

# 前置条件

1. 需要在Zeekr应用服务平台（后续补充）注册应用，并为应用申请Navi模块的使用权限，权限审核通过之后下载zeekr-api-license.txt，并配置到项目工程assets目录中。
2. 安装极氪自研地图apk。
3. 将ZeekrCoreService的APK推到system/priv-app下，重启车机。（或者刷已经集成好的版本）

# 混淆配置

# 入门集成

1. 下载zeekr-sdk-base-1.0.0.jar、zeekr-sdk-navi-1.0.0.jar。
2. 将下载的jar文件放在libs目录下，并在build.gradle中添加引用。

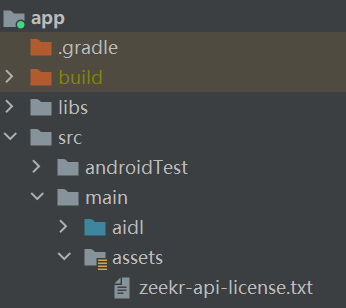
dependencies {

implementation files("libs/zeekr-sdk-base-1.0.0.jar")

implementation files("libs/zeekr-sdk-navi-1.0.0.jar")

}

1. 将获取的license文件放在assets目录下，如下：



# SDK初始化

Navi SDK提供的所有方法，需要在初始化成功以后才能使用，初始化的样例代码如下：

NaviAPI.get().init(context, new ApiReadyCallback({

@Override

public void onAPIReady(boolean b, String s) {

//b：是否初始化成功。true:初始化成功；false:初始化失败

//s：初始化失败时的原因说明。

}

});

# SDK接口说明

1. 主图接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 接口含义 | 入参 | 出参 |
| showMyLocation | 在地图上显示车所在位置/获取当前车辆所在位置信息 | MapOperaShowMyLocation | RspShowMyLocation |
| switchMapTraffic | 切换地图路况开关 | MapOperaSwitchTraffic | NaviBaseModel |
| switchMapViewMode | 切换地图的视图模式 2D/3D | MapOperaViewMode | NaviBaseModel |
| mapZoomInOut | 缩放地图 | MapOperaZoomInOut | NaviBaseModel |
| launchMap | 回地图(默认地图) |  | NaviBaseModel |
| getLastLocation | 获取最后一次定位信息 |  | RspLocation |
| getNaviStatus | 获取导航相关状态 |  | RspNaviStatus |
| getDayStatus | 获取地图当前昼夜模式 |  | int(RspDayNightStatus) |

1. 搜索接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 接口含义 | 入参 | 出参 |
| searchByKeyword | 关键字搜索Poi | SearchByKeyword | RspPoiResult |
| searchAround | 周边搜索Poi | SearchAround | RspPoiResult |
| searchAlongWay | 沿途搜索Poi | SearchAlongWay | RspPoiResult |
| requestAntiGeo | 通过经纬度获取详细位置信息。需要使用GCJ02坐标系。 | MapOperaAntiGeo | RspAntiGeoInfo |
| getEtaInfoFromPoi | 获取POI的ETA信息 | PoiInfo | RspEtaInfo |

1. 导航接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 接口含义 | 入参 | 出参 |
| getFrontTraffic | 获取前方路况 | NaviGetFrontTrafficRadio | RspTrafficSummaryInfo |
| getHighwayExitInfo | 导航时获取高速出口信息 | NaviGetHighwayExit | RspHighwayExitInfo |
| routePlanOrNavi | 路线规划或者发起导航 | NaviRoutePlan | RspRoutePlanResult |
| routeSelect | 路线规划后选择路线 | NaviRouteSelect | NaviBaseModel |
| specialPoiNavi | 特殊Poi导航:回家/去公司 | NaviSpecialPoi | RspRoutePlanResult |
| naviViaEdit | 设置途经点,并更新路线 | NaviViaModify | NaviBaseModel |
| cancelNavi | 取消导航 |  | NaviBaseModel |
| getGuideInfo | 获取引导信息 |  | RspGuideInfo |
| getRouteInfo | 获取路线信息 |  | RspCurrentRoute |
| getLanesInfo | 获取车道信息 |  | RspLanesInfo |
| switchFullView | 切换全览视图状态 |  | NaviBaseModel |
| getSpeedLimitInfo | 获取当前道路限速信息 |  | RspSpeedLimitInfo |
| reRouting | 导航状态下，重新规划路线 |  | NaviBaseModel |
| getArrivedFrontPoiInfo | 导航状态下，到达前方某个POI点的时间/距离 | NaviFrontPoiInfo | RspFrontPoiInfo |
| getCurrentElectricity | 获取当前剩余电量 | CurrentElectricityReq | RspCurrentElectricity |
|  |  |  |  |

1. 设置接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 接口含义 | 入参 | 出参 |
| muteNaviAudio | 静音导航音频 |  |  |
| unmuteNaviAudio | 解除导航音频静音 |  |  |
| setRoutePrefer | 路线偏好设置 | NaviRoutePrefer | NaviBaseModel |
| setCurrentBroadcastMode | 设置导航播报模式 | NaviBroadCastMode | NaviBaseModel |

1. 用户接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 接口含义 | 入参 | 出参 |
| getHistoryPois | 查询历史目的地 | RequestHistoryPois | RspHistoryPois |
| editFavoritePois | 编辑Poi收藏夹 | UserFavoritePoiEdit | NaviBaseModel |
| getFavoritePois | 获取收藏夹的poi | UserGetFavoritePois | RspFavoritePois |
| actionFavoritePage | 通过此接口打开地图收藏夹 | UserFavoritePage | NaviBaseModel |
| requestAccountInfo | 请求高德账号信息 | AccountInfoReq | RspAccountInfo |
| requestAccountLogout | 请求高德账号退出 | AccountLogoutReq | NaviBaseModel |
| checkHasBind | 根据车企账号检查是否绑定 | AccountBindCheckReq | NaviBaseModel |
| checkTokenForServer | 检查车企账号是否为高德账号 | AccountCheckTokenReq | RspCheckTokenInfo |
| autoLoginForServer | 快速登录（服务对服务） | AccountAutoLoginServerReq | NaviBaseModel |
| sendReqCarLtdQuickRegister | 车企账号快速注册 | AccountRegisterQuickReq | NaviBaseModel |

1. 监听接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 接口含义 | 入参 | 出参 |

|  |  |  |  |
| --- | --- | --- | --- |
| addNaviEventListener | 监听导航服务非高频事件，如地图状态、定位信息、路线规划信息等 | INaviEventListener  filters过滤参数：  ONNAVISTATUSCHANGED：导航状态监听  ONMAPSTATUSCHANGED：地图状态监听  ONFAVORITECHANGED：收藏点操作监听  ONROUTEINFOCHANGED：路线信息监听  ONGUIDEINFOCHANGED：引导信息监听  ONHIGHWAYEXITINFOUPDATE：高速路出口信息监听  ONARRIVEDDESTCHANGED：到达目的地通知  ONADAREACHANGED：行政区变化  ONTRAFFICCONDITIONSCHANGED：实时路况更新  ONLANESINFOCHANGED：车道线变更  ONMAPSPEEDLIMITINFOCHANGED：道路限速变更  ONTBTETADATACHANGED：TBT ETA数据变更  ONSAPADATACHANGED：SAPA数据变更  ONDAYNIGHTCHANGED：昼夜模式变更 | RspNaviStatus  RspFavoriteChange  RspGuideInfo  RspLanesInfo  RspLocation  NaviTripReporter  RoadTrafficInfo  MapSpeedLimitInfo  ArrivedDestModel  RspLastMileNaviModel  RspDayNightStatus |
| removeNaviEventListener | 删除监听器 | INaviEventListener |  |

1. 参考示例：

|  |  |
| --- | --- |
| 接口名 | 示例代码 |
| switchMapViewMode | //设置2D车头朝上视图模式  MapOperaViewMode viewMode = new MapOperaViewMode(MapOperaViewMode.MODE\_2D\_FRONT\_UP);  naviAPI.switchMapViewMode(viewMode, new INaviAPICallback() {  @Override  public void onSuccess(NaviBaseModel naviBaseModel) {  //成功回调  }  @Override  public void onError(NaviErrorModel naviErrorModel) {  //错误回调  }  }); |
| mapZoomInOut | //放大地图  MapOperaZoomInOut zoomInOut = new MapOperaZoomInOut(MapOperaZoomInOut.ACTION\_ZOOM\_IN);  naviAPI.mapZoomInOut(zoomInOut, new INaviAPICallback() {  @Override  public void onSuccess(NaviBaseModel model) {  //请求成功  }  @Override  public void onError(NaviErrorModel model) {  //错误码  int errorCode = model.getErrorCode();  //错误消息  String errorMsg = model.getMessage();  if (APIResultCode.RESULT\_SCALE\_IS\_LARGEST == errorCode) {  //当前地图已放大到最大  } else if (APIResultCode.RESULT\_SCALE\_IS\_SMALLEST == errorCode) {  //当前地图已缩小到最小  }  }  }); |
| searchByKeyword | SearchByKeyword reqModel = new SearchByKeyword();  //设置搜索类型为关键字搜索  reqModel.setSearchType(SearchByKeyword.SEARCH\_TYPE\_KEYWORD);  //设置搜索关键字  reqModel.setKeywords("凤凰机场");  naviAPI.searchByKeyword(reqModel, new INaviAPICallback() {  @Override  public void onSuccess(NaviBaseModel model) {  RspPoiResult poiResult = (RspPoiResult) model;  //POI列表  List<PoiInfo> poiList = poiResult.getPoiInfoList();  }  @Override  public void onError(NaviErrorModel model) {  }  }); |
| routePlanOrNavi | PoiInfo poiInfo = new PoiInfo();  poiInfo.setLatLng(new LatLng(31.245369, 121.50626));  poiInfo.setName("东方明珠");  //设置目的地  NaviRoutePlan reqModel = new NaviRoutePlan(poiInfo);  PoiInfo viaPoi = new PoiInfo();  viaPoi.setLatLng(new LatLng(30.697702, 120.806668));  viaPoi.setName("嘉兴南站");  List<PoiInfo> viaList = new ArrayList<>();  viaList.add(viaPoi);  //设置途经点  reqModel.setViaPoiInfos(viaList);  //算路偏好-使用地图默认的算路规则  reqModel.setStrategy(RoutePlanStrategy.DEFAULT);  //发起路线规划  reqModel.setAction(NaviRoutePlan.ACTION\_ROUTE\_PLAN);  naviAPI.routePlanOrNavi(reqModel, new INaviAPICallback() {  @Override  public void onSuccess(NaviBaseModel model) {  RspRoutePlanResult routePlanResult = (RspRoutePlanResult) model;  //路线信息  List<RouteInfo> routes = routePlanResult.getRouteInfoList();  }  @Override  public void onError(NaviErrorModel model) {  }  }); |
| getFavoritePois | //查询家和公司的POI  UserGetFavoritePois getFavoritePois = new UserGetFavoritePois(UserGetFavoritePois.FAVORITE\_TYPE\_HOME\_AND\_COMPANY);  naviAPI.getFavoritePois(getFavoritePois, new INaviAPICallback() {  @Override  public void onSuccess(NaviBaseModel model) {  RspFavoritePois results = (RspFavoritePois) model;  List<FavoritePoiInfo> favoritePoiInfoList = results.getFavoritePoiInfoList();  //收藏夹POI信息  }  @Override  public void onError(NaviErrorModel model) {  }  }); |
| addNaviEventListener | ArrayList<String> filters = new ArrayList<>();  // filters.add(RouterConstant.NotifyModule.ONLOCATIONCHANGED);  filters.add(RouterConstant.NotifyModule.ONMAPSTATUSCHANGED);  // filters.add(RouterConstant.NotifyModule.ONGUIDEINFOCHANGED);  // filters.add(RouterConstant.NotifyModule.ONLANESINFOCHANGED);  // filters.add(RouterConstant.NotifyModule.ONTRIPREPORTERUPDATE);  // filters.add(RouterConstant.NotifyModule.ONFAVORITECHANGED);  // filters.add(RouterConstant.NotifyModule.ONDAYNIGHTCHANGED);  INaviEventListener iNaviEventListener = naviBaseModel -> {  Log.d(TAG, "setNaviEventListener: " + naviBaseModel.getMethodCode());  if (RouterConstant.NotifyModule.ONLOCATIONCHANGED.equalsIgnoreCase(naviBaseModel.getMethodCode())) {  RspLocation location = (RspLocation) naviBaseModel;  LocationInfo locationInfo = location.getLocationInfo();  //定位信息  UserApiControl.this.wirte(true, "NAVI\_NTF\_LOCATION\_CHANGED", locationInfo.toString());  }  if (RouterConstant.NotifyModule.ONMAPSTATUSCHANGED.equalsIgnoreCase(naviBaseModel.getMethodCode())) {  RspNaviStatus rspNaviStatus = (RspNaviStatus) naviBaseModel;  Log.d(TAG, "rspNaviStatus: " + rspNaviStatus.getNaviStatus().toString());  //地图状态  UserApiControl.this.wirte(true, "NAVI\_NTF\_MAP\_STATUS", rspNaviStatus.toString());  }  if (RouterConstant.NotifyModule.ONGUIDEINFOCHANGED.equalsIgnoreCase(naviBaseModel.getMethodCode())) {  RspGuideInfo rspGuideInfo = (RspGuideInfo) naviBaseModel;  //引导信息  UserApiControl.this.wirte(true, "ONGUIDEINFOCHANGED", rspGuideInfo.toString());  }  if (RouterConstant.NotifyModule.ONLANESINFOCHANGED.equalsIgnoreCase(naviBaseModel.getMethodCode())) {  RspLanesInfo rspLanesInfo = (RspLanesInfo) naviBaseModel;  //车道信息  UserApiControl.this.wirte(true, "NAVI\_OP\_GET\_LANES\_INFO", rspLanesInfo.toString());  }  if (RouterConstant.NotifyModule.ONTRIPREPORTERUPDATE.equalsIgnoreCase(naviBaseModel.getMethodCode())) {  NaviTripReporter naviTripReport = (NaviTripReporter) naviBaseModel;  //行程信息  UserApiControl.this.wirte(true, "NAVI\_NTF\_TRIP\_REPORT", naviTripReport.toString());  }  if (RouterConstant.NotifyModule.ONFAVORITECHANGED.equalsIgnoreCase(naviBaseModel.getMethodCode())) {  //收藏点变化信息  RspFavoriteChange rspFavoriteChange = (RspFavoriteChange) naviBaseModel;  UserApiControl.this.wirte(true, "ONFAVORITECHANGED", rspFavoriteChange.toString());  }  };  if  (RouterConstant.NotifyModule.ONDAYNIGHTCHANGED.equalsIgnoreCase(naviBaseModel.getMethodCode())) {  //昼夜模式变化信息  RspDayNightStatus rspDayNightStatus= (RspDayNightStatus ) naviBaseModel;  UserApiControl.this.wirte(true, "RspDayNightStatus", "rspDayNightStatus.dayNightStatus=" + rspDayNightStatus.dayNightStatus);  };  naviAPI.addNaviEventListener(iNaviEventListener, filters); |
|  |  |

具体字段参考导航java-doc文档



# 注意事项

1. XXXXX
2. XXXXX

# 更新日志

## V1.0.1

发布时间：2022-08-22

新增/修复功能点：

1. 添加callbackID来映射回调，解决多次调用同一接口只有一次回调问题。

# V1.0.2

发布时间：2022-10-8

新增/修复功能点：

1. 添加获取地图当前昼夜模式及状态监听

# 常见问题

Q1：XXXXXXXXX

A1：XXXXXXXXX

Q2: XXXXXXX

A2: XXXXXXX