

乘客端接口说明（Android）

PassengerRouteManager

1、构造方法

```
/**
 * 构造方法
 * @param context 上下文
 * @param aMap AMap对象
 * @param options 配置项，用来自定义轨迹线颜色、起终点图标、InfoWindow样式等属性
 */
public PassengerRouteManager(Context context, AMap aMap, RouteOverlayOptions options)
```

2、设置订单信息

```
/**
 * 设置订单信息
 * @param orderId 订单ID
 * @param startPosition 起点坐标
 * @param pickUpPosition 上车点坐标
 * @param endPosition 终点坐标
 * @throws AMapException 无效参数异常
 */
public void setOrderProperty(String orderId, LatLng startPosition, LatLng pickUpPosition, LatLng endPosition) throws AMapException
```

3、更新订单状态

```
/**
 * 更新订单状态
 * @param orderState 订单状态 详情参见{@link SCTXConfig}
 */
public void setOrderState(int orderState)
```

4、设置获取司机端位置的回调

```
/**
 * 设置获取司机端位置的回调 （乘客端获取不到司机位置的降级策略）
 * @param driverPositionCallback
 */
public void setDriverPositionCallback(DriverPositionCallback driverPositionCallback)
```

5、设置当前乘客位置到上车点的虚线

```
/**
 * 设置是否显示当前乘客位置到上车点的虚线
 * @param isWalkRouteLineEnable 是否显示当前乘客位置到上车点的虚线
 */
public void setWalkRouteLineEnable(boolean isWalkRouteLineEnable)
```

6、更新InfoWindow信息内容

```
/**
 * 更新InfoWindow信息内容
 * @param remainingDistance 剩余距离
 * @param estimatedTime 预计时间
 * @param cost 费用
 */
public void updateInfoWindow(float remainingDistance,long estimatedTime,float cost)
```

7、设置刷新司机位置的时间间隔

```
/**
 * 设置 刷新司机位置时间间隔
 * @param interval 时间间隔 单位: s
 */
public static void setRefreshTrackInterval(int interval)
```

8、设置刷新刷新路况的时间间隔

```
/**
 * 设置刷新路况时间间隔
 * @param interval 时间间隔 单位: s
 */
public static void setRefreshTrafficInterval(int interval)
```

9、设置状态监听

```
/**
 * 设置状态监听
 * @param callback
 */
public void setPassengerOverlayRouteCallback(PassengerRouteCallback callback)
```

10、获取起点的Marker对象

```
/**
 * 获取起点的Marker对象
 * @return
 */
public Marker getStartPointMarker()
```

11、获取终点的Marker对象

```
/**
 * 获取终点的Marker对象
 * @return
 */
public Marker getEndPointMarker()
```

12、获取小车的Marker对象

```
/**
 * 获取小车的Marker对象
 * @return
 */
public Marker getCarMarker()
```

13、销毁资源

```
/**
 * 销毁资源及停止Timer
 */
public void destroy()
```

14、设置路线边界值

```

/**
 * 设置路线边界值
 * @param left      左边界
 * @param right     右边界
 * @param top       上边界
 * @param bottom    下边界
 */
public void setNavigationLineMargin(int left, int right, int top, int bottom)

```

15、设置操作地图后刷新地图视野间隔时间

```

/**
 * 设置操作地图后刷新地图视野间隔时间
 * @param operateTimeOut 单位：秒
 */
public void setOperateMapTimeOut(int operateTimeOut)

```

RouteOverlayOptions

```

/**
 * 设置起点图标
 * @param bitmapDescriptor
 * @return
 */
public RouteOverlayOptions startPointIcon(BitmapDescriptor bitmapDescriptor)

/**
 * 设置终点图标
 * @param bitmapDescriptor
 * @return
 */
public RouteOverlayOptions endPointIcon(BitmapDescriptor bitmapDescriptor)

/**
 * 设置小车图标
 * @param bitmapDescriptor
 * @return
 */
public RouteOverlayOptions carIcon(BitmapDescriptor bitmapDescriptor)

/**
 * 设置默认轨迹线填充纹理
 * @param bitmapDescriptor
 * @return
 */
public RouteOverlayOptions defaultRouteRes(BitmapDescriptor bitmapDescriptor)

/**
 * 设置步行轨迹线填充纹理
 * @param bitmapDescriptor
 * @return
 */
public RouteOverlayOptions walkRouteRes(BitmapDescriptor bitmapDescriptor)

/**

```

```
* 设置路况轨迹线填充纹理--畅通
* @param bitmapDescriptor
* @return
*/
public RouteOverlayOptions smoothTrafficRes(BitmapDescriptor bitmapDescriptor)

/**
 * 设置路况轨迹线填充纹理--缓慢
 * @param bitmapDescriptor
 * @return
 */
public RouteOverlayOptions slowTrafficRes(BitmapDescriptor bitmapDescriptor)

/**
 * 设置路况轨迹线填充纹理--拥堵
 * @param bitmapDescriptor
 * @return
 */
public RouteOverlayOptions jamTrafficRes(BitmapDescriptor bitmapDescriptor)

/**
 * 设置路况轨迹线填充纹理--严重拥堵
 * @param bitmapDescriptor
 * @return
 */
public RouteOverlayOptions veryJamTrafficRes(BitmapDescriptor bitmapDescriptor)

/**
 * 设置路况轨迹线填充纹理--未知路况
 * @param bitmapDescriptor
 * @return
 */
public RouteOverlayOptions unknownTrafficRes(BitmapDescriptor bitmapDescriptor)

/**
 * 设置InfoWindow适配器
 * @param infoWindowAdapter
 * @return
 */
public RouteOverlayOptions infoWindowAdapter(AMap.InfoWindowAdapter infoWindowAdapter)

/**
 * 设置路线宽度
 * @param width
 * @return
 */
public RouteOverlayOptions routeLineWidth(float width)

/**
 * 设置司机位置刷新时间间隔
 * @param interval
 * @return
 */
public RouteOverlayOptions intervalForRefreshDriverPosition(long interval)

/**
 * 设置路况刷新时间间隔
 * @param interval
 * @return
 */
public RouteOverlayOptions intervalForRefreshTrafficStatus(long interval)

/**
 * 设置地图边距
```

```

    * @param marginLeft    左边距
    * @param marginRight    右边距
    * @param marginTop      上边距
    * @param marginBottom   下边距
    * @return
    */
    public RouteOverlayOptions margin(int marginLeft, int marginRight, int marginTop, int marginBottom)
```

PassengerRouteCallback

客端状态监听

```

/**
 * @param distance 已行驶距离
 * @param time 已行驶时间
 * @param remainingDistance 剩余距离
 * @param estimatedTime 预计时间
 */
void onRouteStatusChange(int status, float distance, long time, float remainingDistance, long estimatedTime);

/**
 * 司机位置回调
 * @param position 司机位置
 */
void onDriverPositionChange(LatLng position);

/**
 * 请求回调错误接口
 * @param errorCode 错误码
 * @param message 错误信息
 */
void onError(int errorCode, String message);
```

DriverPositionCallback

司机位置监听

```

/**
 * 获取司机位置 （注： 主要用于乘客端无法获取到司机位置的降级策略， 从APP中获得）
 * @return 司机位置
 */
LatLng getDriverPosition();
```