

司机端接口说明（iOS）

MADriverRouteManager

状态枚举

```
///行程状态
typedef NS_ENUM(NSInteger, MASCTXDRouteStatus)
{
    MASCTXDRouteStatusUnspecified = 0, ///<未指定状态
    MASCTXDRouteStatusPickupPassenger = 1, ///<去接乘客
    MASCTXDRouteStatusWaitPassenger = 2, ///<等待乘客上车
    MASCTXDRouteStatusPassengerOnBoard = 3, ///<乘客已上车
    MASCTXDRouteStatusOrderComplete = 4, ///<订单已结束
};
```

司机端控制器

```
@interface MADriverRouteManager : NSObject

///司机信息自动上传时间间隔，单位秒，默认为10s。
@property (nonatomic, assign) NSInteger driverInfoUploadTimeInterval;

///地图对象
@property (nonatomic, weak) MAMapView *mapView;

///代理对象
@property (nonatomic, weak) id<MADriverRouteManagerDelegate> delegate;

///驾车导航manager，司乘同显司机端生命周期期间需要导航的请使用此属性。
@property (nonatomic, weak, readonly) AMapNaviDriveManager *driveManager;

///导航策略，默认AMapNaviDrivingStrategySingleDefault
@property (nonatomic, assign) AMapNaviDrivingStrategy drivingStrategy;

///起点标注
@property (nonatomic, strong, readonly) MAPointAnnotation *beginAnnotation;

///终点标注
@property (nonatomic, strong, readonly) MAPointAnnotation *endAnnotation;

///车辆标注
@property (nonatomic, strong, readonly) MAAnimatedAnnotation *carAnnotation;

///是否自动调整地图显示区域，默认为YES。默认行为：在司机位置回调后，调整地图可视范围以显示carAnnotation和终点annotation。注意，当前版本内部自动调整不在选中annotation。如果要自定义此行为，请设置此属性为NO，且在routeStatusChangeForManager回调中对地图进行操作，获取添加timer进行自定义操作。
@property (nonatomic, assign) BOOL automaticAdjustMapRegion;

///调整地图显示区域时间间隔，默认10s
@property (nonatomic, assign) NSTimeInterval autoAdjustMapTimeInterval;

///地图显示区域改变后，自动调整地图图时间间隔，默认10s
@property (nonatomic, assign) NSTimeInterval autoAdjustMapAfterRegionChangedTimeInterval;

///默认调整地图时需要设置的padding，默认为(30, 30, 80, 30)
@property (nonatomic, assign) UIEdgeInsets automaticAdjustPadding;
```

```

///小车不进行动画的移动距离，单位米。默认为800。
@property (nonatomic, assign) double ignoreCarAnimationDistance;

///是否开启log输出，默认为NO。
@property (nonatomic, assign) BOOL enableDebugLog;

///是否开启模拟导航，默认为NO。
@property (nonatomic, assign) BOOL startWithEmulatorNavi;

///小车图片
@property (nonatomic, strong) UIImage *carImage;

///起点图片
@property (nonatomic, strong) UIImage *startImage;

///终点图片
@property (nonatomic, strong) UIImage *endImage;

///路况纹理 - 拥堵
@property (nonatomic, strong) UIImage *blockedImage;

///路况纹理 - 非常缓慢
@property (nonatomic, strong) UIImage *verySlowImage;

///路况纹理 - 缓慢
@property (nonatomic, strong) UIImage *slowImage;

///路况纹理 - 畅通
@property (nonatomic, strong) UIImage *goodImage;

///路况纹理 - 未知
@property (nonatomic, strong) UIImage *unknownImage;

///路线默认纹理
@property (nonatomic, strong) UIImage *defaultImage;

///步行路线纹理
@property (nonatomic, strong) UIImage *walkImage;

///线宽，默认 18
@property (nonatomic, assign) CGFloat lineWidth;

///行程状态，设置行程状态会改变manager的行为
@property (nonatomic, assign) MASCTXDRouteStatus status;

/**
 @brief 设置订单信息，如果没有经纬度，请传 CLLocationCoordinate2DInvalid。注意：如果endPosition传invalid，则送乘客阶段不显示路线。
 @param orderId 订单id
 @param begin 起点坐标
 @param end 终点坐标
 */
- (void)setOrderInfoWith:(NSString *)orderId
    beginPosition:(CLLocationCoordinate2D)begin
    endPosition:(CLLocationCoordinate2D)end;

/**
 @brief 设置小车弹出框，如果执行了此接口，则不显示默认callout。
 @param calloutView 气泡显示内容
 */
- (void)updateDriverInfoWith:(UIView *)calloutView;

/**
 @brief 设置起点annotation弹出框

```

```

    @brief 设置起点annotation弹出框
    @param calloutView 气泡显示内容
    */
- (void)updateStartAnnotationInfoWith:(UIView *)calloutView;

/**
    @brief 设置终点annotation弹出框
    @param calloutView 气泡显示内容
    */
- (void)updateEndAnnotationInfoWith:(UIView *)calloutView;

/**
    @brief 设置订单信息，只在默认calloutView时有效果，如果设置了自定义calloutView，则无效。
    @param distance 剩余距离，单位米
    @param time 预估时间，单位秒
    @param price 预估价格，默认只在送驾时有效，单位元
    */
- (void)updateDriverInfoWithDistance:(double)distance time:(NSTimeInterval)time price:(double)price;

///判断是否是内部overlay
- (BOOL)isInternalOverlay:(id<MAOverlay>)overlay;

///重新算路，只有在接乘客和送乘客阶段才会成功
- (void)calculateNaviRoute;

#pragma mark - MapView代理相关方法

/**
    * @brief 在mapView对应的delegate方法内调用此方法
    * @param mapView 地图View
    * @param animated 是否动画
    */
- (void)mapView:(MAMapView *)mapView regionWillChangeAnimated:(BOOL)animated;

/**
    * @brief 在mapView对应的delegate方法内调用此方法
    * @param mapView 地图View
    * @param animated 是否动画
    */
- (void)mapView:(MAMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

/**
    * @brief 在mapView对应的delegate方法内调用此方法
    * @param mapView 地图view
    * @param wasUserAction 标识是否是用户动作
    */
- (void)mapView:(MAMapView *)mapView mapWillMoveByUser:(BOOL)wasUserAction;

/**
    * @brief 在mapView对应的delegate方法内调用此方法
    * @param mapView 地图view
    * @param wasUserAction 标识是否是用户动作
    */
- (void)mapView:(MAMapView *)mapView mapDidMoveByUser:(BOOL)wasUserAction;

/**
    * @brief 在mapView对应的delegate方法内调用此方法
    * @param mapView 地图view
    * @param wasUserAction 标识是否是用户动作
    */
- (void)mapView:(MAMapView *)mapView mapWillZoomByUser:(BOOL)wasUserAction;

/**
    * @brief 在mapView对应的delegate方法内调用此方法
    * @param mapView 地图view

```

```

    @param mapView 地图view
    * @param wasUserAction 标识是否是用户动作
    */
- (void)mapView:(MAMapView *)mapView mapDidZoomByUser:(BOOL)wasUserAction;

/**
    @brief 在mapView对应的delegate方法内调用此方法
    @param mapView 地图对象
    @param overlay overlay
    @return 对于属于当前controller的overlay，返回相应的MAOverlayRenderer对象，否则返回nil
    */
- (MAOverlayRenderer *)mapView:(MAMapView *)mapView rendererForOverlay:(id <MAOverlay>)overlay;

/**
    * @brief 在对应的地图delegate方法中调用此方法
    * @param mapView 地图
    * @param annotation 标注
    * @return 对于属于当前controller的annotation，返回相应的annotationview，否则返回nil
    */
- (MAAnnotationView *)mapView:(MAMapView *)mapView viewForAnnotation:(id <MAAnnotation>)annotation;

@end

```

MADriverRouteManagerDelegate

司机端控制器代理

```

@protocol MADriverRouteManagerDelegate <NSObject>
@optional

///路线上传成功回调
- (void)driverRouteManger:(MADriverRouteManager *)manager didFinishUploadRoute:(NSString *)routeString;

///发生请求错误时会回调，导航相关错误回调
- (void)driverRouteManger:(MADriverRouteManager *)manager requestFailedWithError:(NSError *)error;

///remainingDistance:剩余距离 estimatedTime:预计时间
- (void)routeStatusChangeForManager:(MADriverRouteManager *)manager
    remainingDistance:(double)remainingDistance
    estimatedTime:(NSTimeInterval)estimatedTime;

@end

```