```
//添加数据到地图点
-(void)addAnnountionInMap
{
    if(_mapView.annotations)
    {
        [_mapView removeAnnotations:_mapView.annotations];
    }
    [self didGroupWithModelArray:self.dataArray divisibleWithCount:200
sucessBlock:^(NSMutableArray *totalArray) {
            [self.announArray addObjectsFromArray:totalArray];

                dispatch_async(dispatch_get_main_queue(), ^{

            [_mapView addAnnotations:self.announArray];
            [self.announArray removeAllObjects];
            [FFcbw.hud hide:YES];
            [_activityVc stopAnimating];
            [_mapView reloadInputViews];
                });


    }];



}
/**
 *  组线程处理点位信息 使点位较多时能够处理更快
 *
 *  @param modelArray 网络获取的数据点位数组
 *  @param count      每个组要处理的数目 建议不要超过300
 *  @param block      返回已经处理完毕的点位数组
 */
-(void)didGroupWithModelArray:(NSMutableArray *)modelArray divisibleWithCount:
(NSInteger)count sucessBlock:(isSucessBlock)block
{
    dispatch_queue_t queue =
dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
    dispatch_group_t group = dispatch_group_create();
    dispatch_semaphore_t semaphore = dispatch_semaphore_create(1);
    NSMutableArray * totalArray = [[NSMutableArray alloc]init];
    if(modelArray.count <= count)
    {
        [totalArray addObjectsFromArray:[self filterWithArray:modelArray]];
        block(totalArray);
    }
    else
    {
        for(int i =0;i<(modelArray.count/count + 1);i ++)
        {
            NSMutableArray * mutableArray = [[NSMutableArray alloc]init];
            NSArray * array;
            if(i<modelArray.count/count)
            {
                array =[modelArray subarrayWithRange:NSMakeRange(i * count, count)];
            }
            else
            {
                array = [modelArray subarrayWithRange:NSMakeRange(i * count, 
modelArray.count - i * count)];
            }
            dispatch_semaphore_wait(semaphore, DISPATCH_TIME_FOREVER);
            dispatch_group_async(group, queue, ^{
                [mutableArray addObjectsFromArray:[self filterWithArray:(NSMutableArray 
*)array]];
                [totalArray addObjectsFromArray:mutableArray];
                dispatch_semaphore_signal(semaphore);

            });
        }
        dispatch_group_notify(group, queue, ^{
            block(totalArray);
        });
    }

}
/**
 *  过滤数组
 *
 *  @param array 传过去要过滤的数组
 *
 *  @return 返回已经过滤的数组
 */
-(NSMutableArray *)filterWithArray:(NSMutableArray * )array
{
    NSMutableArray * complitArray = [[NSMutableArray alloc]init];
    for(ComeBaseModel * model in array)
    {
        CLLocationCoordinate2D coordinate = 
CLLocationCoordinate2DMake(model.latitude , model.longitude);
        CGPoint annoPoint = [_mapView convertCoordinate:coordinate 
toPointToView:self.view];
        //过滤 是否是合法的经纬度
        BOOL isSure = CLLocationCoordinate2DIsValid(coordinate);
        if(isSure)
        {
            BMKPointAnnotation* annotation = [[BMKPointAnnotation alloc]init];
            annotation.coordinate = coordinate;
            annotation.subtitle = model.contacts==nil?@"":model.contacts;
            annotation.title = model.department;
            BOOL isIn = BMKMapRectContainsPoint(BMKMapRectMake(0, 64, 
viewWidth, viewHeight - 64), BMKMapPointMake(annoPoint.x, annoPoint.y));
            if(isIn)
            {

                [complitArray addObject:annotation];

            }

        }

    }
    return complitArray;
}
```