

```
@Override
public boolean addTakeUpSeatOrder(long roleId, long seatId,boolean isStaff) {
    //将座位标记成占用对应两张情况： 1、预订的人到店吃饭： 由预订状态变为占用状态
    //2、堂食： 直接由可用状态变为占用状态
    Timestamp start = new Timestamp(System.currentTimeMillis());
    LocationCardLog takeUpCardLog=new LocationCardLog();
    takeUpCardLog.setRoleId(roleId);
    takeUpCardLog.setLocationCardId(seatId);
    SeatLog seatLog=new SeatLog();
    seatLog.setType(SeatOrderType.TAKE_UP.ordinal());
    takeUpCardLog.setState(SeatOrderState.ON_GOING.ordinal());
    takeUpCardLog.setBegin(start);
    takeUpCardLog.setType(LocationCardLogType.TAKE);
    //结束时间默认为座位的开放时间的结束点
    LocationCard seat = seatService.getSeat(seatId);
    long warehouseId = seat.getLocationManagerId();
    takeUpCardLog.setLocationManagerId(warehouseId);
    WarehouseOpenConfigForm openConfig = virtualWarehouseService.getVirtualWarehouseOpenConfig(warehouseId);
    //预订的结束时间默认为座位的开放时间的结束点
    Timestamp end = getEndTime(start, openConfig.getPeriods());
    takeUpCardLog.setLocationCardId(warehouseId);
    takeUpCardLog.setEnd(end);
    SeatCard seatCard=JSON.parseObject(seat.getExtraInfo(),SeatCard.class);
    if(isStaff) {seat.setState(SeatState.TAKE_UP.getIndex());
        seat.update();}
    return locationCardLogDao.save(takeUpCardLog);
}
```