[SCA] Driver App离线数据监控

1、离线操作订单占比

- 离线Pickup订单占比 = 离线picked up的订单 / 所有picked up的订单
- 离线delivered订单占比 = 离线delivered的订单 / 所有delivered的订单
- 离线delivery onhold订单占比 = 离线delivery onhold的订单 / 所有delivery onhold的订单
- 离线returned订单占比 = 离线returned的订单 / 所有returned的订单
- 离线return onhold订单占比 = 离线return onhold的订单 / 所有return onhold的订单

离线场景	单包裹key	多包裹key
delivery离线delivered订单	lm_delivery_single_offline	lm_delivery_multi_offline
delivery在线delivered订单	lm_delivery_single_online	lm_delivery_multi_online
delivery离线onhold订单	lm_onhold_single_offline	lm_onhold_multi_offline
delivery在线onhold订单	lm_onhold_single_online	lm_onhold_multi_online
return离线returned订单	rts_return_single_offline_with_photo rts_return_single_offline_without_photo	rts_return_multi_offline_with_photo rts_return_multi_offline_without_photo
return在线returned订单	rts_return_single_online_with_photo rts_return_single_online_without_photo	rts_return_multi_online_with_photo rts_return_multi_online_without_photo
return离线return onhold订单	rts_onhold_single_offline_with_photo rts_onhold_single_offline_without_photo	rts_onhold_multi_offline_with_photo rts_onhold_multi_offline_without_photo
return在线return onhold订单	rts_onhold_single_online_with_photo rts_onhold_single_online_without_photo	rts_onhold_multi_online_with_photo rts_onhold_multi_online_without_photo

pickup

离线场景	key
pickup离线picked up订单	fm_order_picked_up_offline
pickup在线picked up订单	fm_order_picked_up_online
pickup离线onhold操作次数	fm_order_onhold_offline
pickup在线onhold操作次数	fm_order_onhold_online
pickup离线onhold task	fm_task_onhold_offline
pickup离线sign task	fm_task_sign_offline
pickup离线disable task	fm_task_disable_offline
pickup在线disable task	fm_task_disable_online
pickup离线arrived task	fm_task_arrived_offline
pickup在线arrived task	fm_task_arrived_online
pickup离线handed over task	fm_task_handover_offline
pickup在线handed over task	fm_task_handover_online

2、使用离线骑手的占比

某段时间内使用过离线成功操作了一个订单的骑手/所有在线骑手数量

在线指也成功操作了一个订单，操作订单为改变订单状态；

在线骑手key：hotfix_try_fetch_patch

pickup：

使用离线成功场景	key
order scan	fm_order_picked_up_with_offline
order onhold	fm_order_onhold_with_offline
task arrived	fm_task_arrived_with_offline
task disabled	fm_task_disable_with_offline
task handed over	fm_task_handed_over_with_offline

task sign和task onhold离线和在线流程一致，无法区分，这里不做统计

delivery：

使用离线成功场景	key
order delivery	`lm_delivery_with_offline`
order onhold	lm_onhold_with_offline

return：

使用离线成功场景	key
order returned	rts_returned_with_offline
order onhold	rts_onhold_with_offline

由于一个市场站点过多，根据站点维度统计MDAP不支持

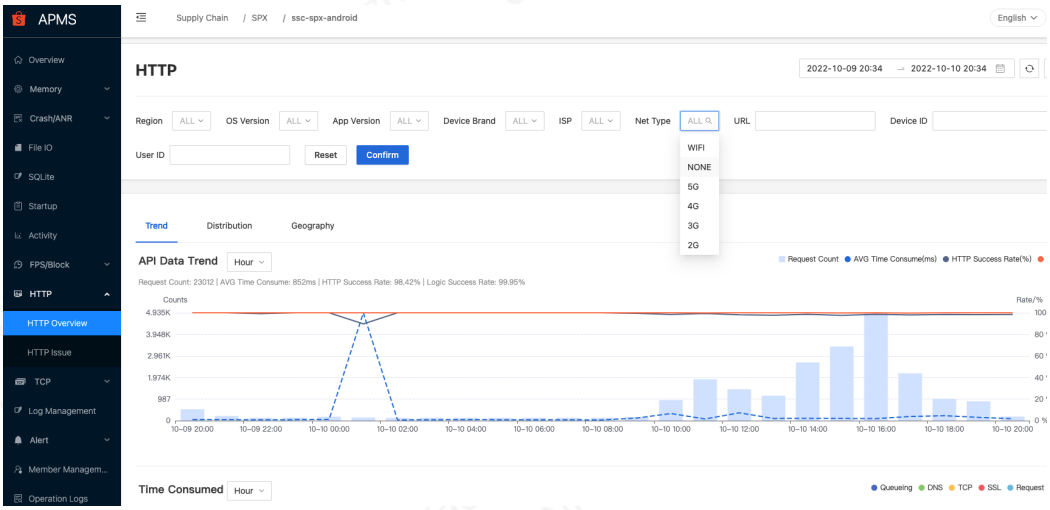
ID：170*24个站点，PH：69*24个站点

3、使用离线的时长占比

离线的时长/在线操作的时长

https://apms.exp.shopee.io/apps/ssc-spx-android/android-http/overview?from=1665318887&tab=trend&to=1665405287

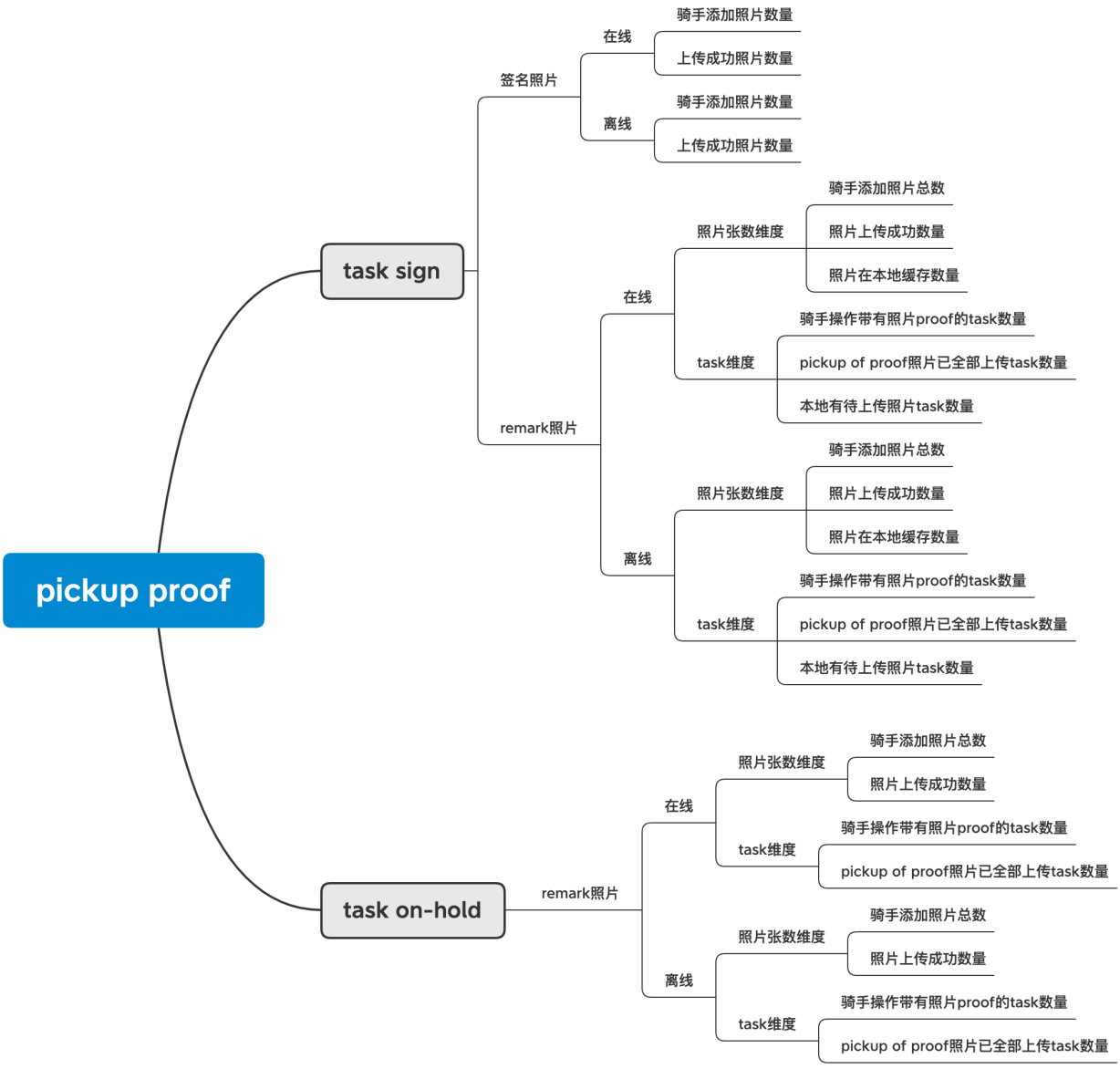
APMS平台在HTTP Overview可根据用户UserId、DeviceId查看骑手网络情况，统计某个时间范围骑手网络情况



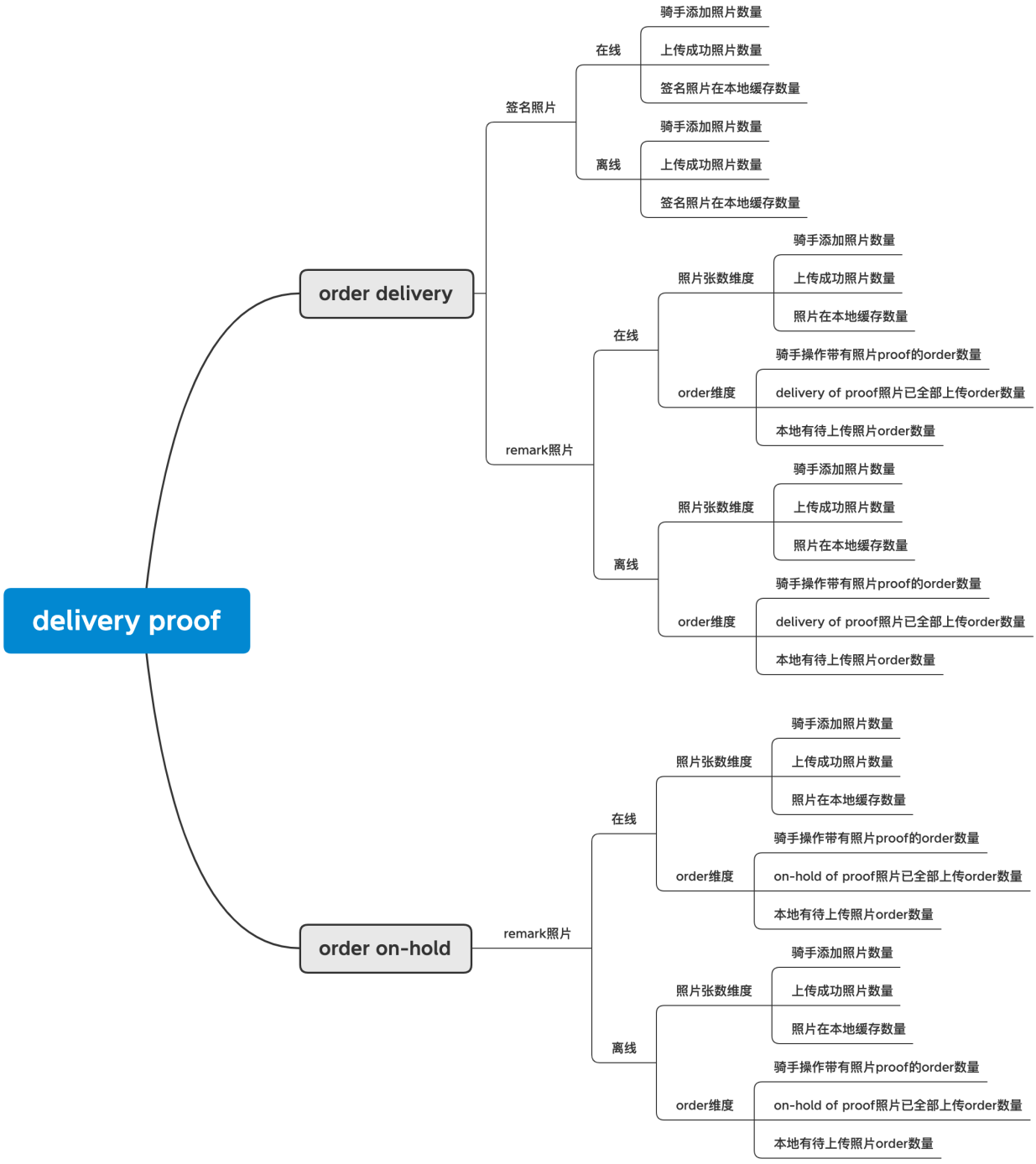
4、离线的稳定性

4.1 离线模式下Proof照片丢失率 ---pickup/delivery/return

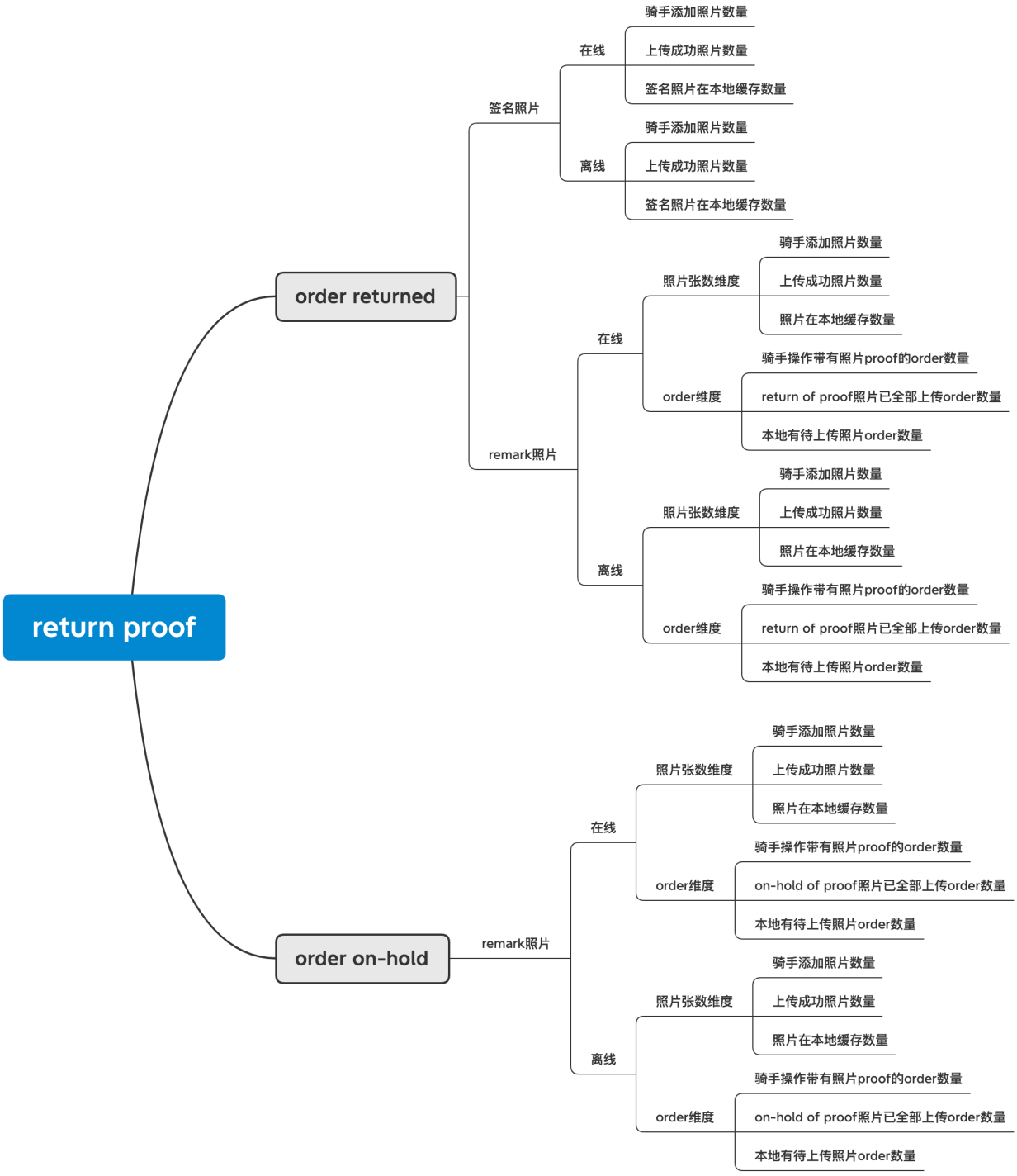
pickup:


pickup proof
task sign
签名照片
在线
骑手添加照片数量
上传成功照片数量
离线
骑手添加照片数量
上传成功照片数量
remark照片
在线
照片张数维度
骑手添加照片总数
照片上传成功数量
照片在本地缓存数量
task维度
骑手操作带有照片proof的task数量
pickup of proof照片已全部上传task数量
本地有待上传照片task数量
离线
照片张数维度
骑手添加照片总数
照片上传成功数量
照片在本地缓存数量
task维度
骑手操作带有照片proof的task数量
pickup of proof照片已全部上传task数量
本地有待上传照片task数量
task on-hold
remark照片
在线
照片张数维度
骑手添加照片总数
照片上传成功数量
task维度
骑手操作带有照片proof的task数量
pickup of proof照片已全部上传task数量
离线
照片张数维度
骑手添加照片总数
照片上传成功数量
task维度
骑手操作带有照片proof的task数量
pickup of proof照片已全部上传task数量

delivery:


delivery proof
order delivery
签名照片
在线
骑手添加照片数量
上传成功照片数量
签名照片在本地缓存数量
离线
骑手添加照片数量
上传成功照片数量
签名照片在本地缓存数量
remark照片
在线
照片张数维度
骑手添加照片数量
上传成功照片数量
照片在本地缓存数量
order维度
骑手操作带有照片proof的order数量
delivery of proof照片已全部上传order数量
本地有待上传照片order数量
离线
照片张数维度
骑手添加照片数量
上传成功照片数量
照片在本地缓存数量
order维度
骑手操作带有照片proof的order数量
delivery of proof照片已全部上传order数量
本地有待上传照片order数量
order on-hold
remark照片
在线
照片张数维度
骑手添加照片数量
上传成功照片数量
照片在本地缓存数量
order维度
骑手操作带有照片proof的order数量
on-hold of proof照片已全部上传order数量
本地有待上传照片order数量
离线
照片张数维度
骑手添加照片数量
上传成功照片数量
照片在本地缓存数量
order维度
骑手操作带有照片proof的order数量
on-hold of proof照片已全部上传order数量
本地有待上传照片order数量

return:



~~4.2 APP本地存在离线数据上报~~

~~pickup:~~

- ~~OfflineUpdateTaskStatusBean ----task各个状态扭转离线数据~~
- ~~UpdateScanPickupOrderParams ----order scan离线数据~~
- ~~OnHoldPickupOrderParams ----order on-hold离线数据~~
- ~~OfflineUpdateTaskPhotoBean~~

  ~~(1)task sign时seller、driver photo和remark第一张图片离线数据~~

~~(2)task on-hold时remark三张图片离线数据~~
- ~~UploadPhotoLocalBean ----remark图片~~

~~delivery:~~

- ~~FleetOrderOfflineSyncBean ----delivery order离线数据~~
- ~~UploadPhotoLocalBean ----remark图片~~
- ~~UploadVideoLocalBean ----video数据~~

~~return:~~

- ~~FleetOrderOfflineSyncBean ----return order离线数据~~
- ~~UploadPhotoLocalBean ----remark图片~~

离线数据丢失率一般和问题直接挂钩，这里每个统计意义不大，暂统计proof丢失率

5、离线数据上传耗时

耗时 = 离线上传同步成功时间 - 离线数据存储时间(骑手离线操作时时间戳)

pickup：

- task on-hold
- task disable
- task arrived
- task sign
- task handover
- order scan
- order on-hold

```
public static final String FM_TASK_ONHOLD_OFFLINE_SYNC_DATA_TIME = "fm_task_onhold_offline_sync_data_time";
public static final String FM_TASK_DISABLE_OFFLINE_SYNC_DATA_TIME = "fm_task_disable_offline_sync_data_time";
public static final String FM_TASK_ARRIVED_OFFLINE_SYNC_DATA_TIME = "fm_task_arrived_offline_sync_data_time";
public static final String FM_TASK_SIGN_OFFLINE_SYNC_DATA_TIME = "fm_task_sign_offline_sync_data_time";
public static final String FM_TASK_HANDOVER_OFFLINE_SYNC_DATA_TIME = "fm_task_handover_offline_sync_data_time";
public static final String FM_ORDER_SCAN_OFFLINE_SYNC_DATA_TIME = "fm_order_scan_offline_sync_data_time";
public static final String FM_ORDER_ONHOLD_OFFLINE_SYNC_DATA_TIME = "fm_order_onhold_offline_sync_data_time";
```

delivery：

- order delivery
- order on-hold

```
public static final String LM_DELIVERY_OFFLINE_SYNC_DATA_TIME = "lm_delivery_offline_sync_data_time";
public static final String LM_ONHOLD_OFFLINE_SYNC_DATA_TIME = "lm_onhold_offline_sync_data_time";
```

return

- order return
- order on-hold

```
public static final String RTS_RETURN_OFFLINE_SYNC_DATA_TIME = "rts_return_offline_sync_data_time";
public static final String RTS_ONHOLD_OFFLINE_SYNC_DATA_TIME = "rts_onhold_offline_sync_data_time";
```

6、骑手操作作业记录

- 通过各个步骤操作时间推测: 在线操作/离线操作 ----可以考虑将各个步骤操作时间上报到MDAP，以userId和task维度
- 通过日志查看 ----依赖日志埋点

7、check app本地数据与后台一致性

pickup：

- 以task维度检查APP本地task状态是否与后台task状态一致，如果不一致，那么检查是否有对应的离线数据，如果离线数量匹配不上，那么这条task的状态存在异常
- 以task维度检查APP本地task pending、picked、on-hold数量是否一致，如果不一致，检查是否有对应的order scan、order on-hold离线数据，如果离线数量匹配不上，那么这条task的order数量存在异常

当前对于task的拉取，如果本地存在task对应离线数据，则不刷新，否则更新存储，能解决这个数据一致性问题

- order维度检查状态，可能数据量比较大.....

当前对于order的拉取，在首页刷新触发，如果task存在order scan或者on-hold离线数据，不会更新order列表数据，否则直接清除再存储，能解决这个数据一致性问题

与后台交流，这个check意义不大，如果存在离线操作，本身就存在较多不一致情况，通过检测两端数据一致性，不好判断问题

可在前期灰度时观察一段时间后再看

delivery/return：

- check order状态是否一致，如果不一致，如何表现？这个check动作通过直接拉取order列表即可做到

如果两端状态不一致，又没有离线数据，这个在首页拉取时就存在这个逻辑，直接在APP覆盖本地数据，以后端当前order数据为准；如果存在离线数据，则不刷新

8、日志上报

放开日志回捞限制

```
//apm日志回捞设置
public static LogInitParams buildLogParams() {
    final int maxLogAliveDayTime = 10;

    return new LogInitParams() {
        @Nullable
        @Override
        public LogLevel logLevel() {
            //如无自定义填写，默认debug版本使用全打印，release版本打印info级别及以上；
            return LogLevel.LEVEL_FATAL;
        }

        @Override
        public long maxFileSize() {
```

return null

排查补充全部离线场景打印日志，形成离线作业日志查询

9、梳理APP离线可能出现的问题

9.1 根据Delivery派送前期离线出现问题

待补充

9.2 根据开发过程梳理可能出现离线问题

见