1. 初始化

`LiveSDK.WX_PAY_APP_ID` 必须设置，否则微信支付充值功能无法正常使用。

```
/**
 * 现金打赏微信支付 APP_ID，跟包名绑定需要客户传入
 */
public static String WX_PAY_APP_ID;

/**
 * 直播带货是否展示技术支持字样
 */
public static boolean
SHOW_LIVE_SHOW_TECH_SUPPORT = true;

/**
 * 直播带货礼物打赏冷却时间，单位毫秒
 */
public static long
LIVE_SHOW_GIFT_SEND_TIME_OUT = 10000;
```

直播带货相关 API 封装在 `LiveShowVM` 中，通过 `liveRoom.getLiveShowVM()` 获取。

2. 商品管理

2.1 拉取商品列表

初始化 `LiveShowVM` 内部会请求第一页数据，外部监听商品变化即可。

```
/**
 * 获取商品列表变更监听
 */
Observable<List<LPLiveProductModel>> getObservableOfSellProducts();
```

还提供了主动获取商品信息的 API，需要保证在 `getObservableOfSellProducts` 之后调用才能正确获取。

```
/**
 * 获取当前所有商品
 *
 * @return 学生仅会返回所有的上架商品
 */
List<LPLiveProductModel> getSellProductsAll();
```

`LPliveProductModel` 为商品 model 对象，字段如下

```
public class LPLiveProductModel extends LPDataModel {
    /**
     * 商品 id
     */
    @SerializedName("id")
    String id;
    /**
     * 直播间 id
     */
    @SerializedName("room_id")
    int roomId;
```

```
    /**
     * 商品名称
     */
    @SerializedName("good_name")
    String name;
    /**
     * 商品图片地址
     */
    @SerializedName("good_img")
    String imgUrl;
    /**
     * 商品价格
     */
    @SerializedName("price")
    float price;
    /**
     * 商品打折价格
     */
    @SerializedName("discount_price")
    float discountPrice;

    /**
     * 购买链接，跳转购买的 url
     */
    @SerializedName("url")
    String buyUrl;
    /**
     * 商品描述
     */
    @SerializedName("desc")
    String desc;

    /**
     * 是否为上架
     */
    transient boolean isOnShelf;
```

```
48. }
```

商品列表支持分页加载和刷新

```
1.  /**
2.     * 请求下一页商品内容
3.     */
4.    void requestNextPage();
5.
6.    /**
7.     * 刷新商品列表，请求第一页的商品信息
8.     */
9.    void refreshProductList();
```

2.2. 上架/下架

```
1.    /**
2.     * 请求变更商品上下架状态，仅老师和助教有权限
3.     *
4.     * @param productId 商品 id
5.     * @param reqShelfState true 上架；false 下架
6.     */
7.    void requestChangeSaleState(String productId,
   boolean reqShelfState);
```

上架或下架后商品状态回调仍然是 `getObservableOfSellProducts`，SDK 内部会更改 `LPLiveProductModel.isOnShelf` 字段。

在 `getObservableOfSellProducts` 监听中可以使用以下 API 快速获取上架和下架商品列表。

```
1.    /**
2.     * 获取上架商品
```

```
    */
    List<LPLiveProductModel>
getProductsOnSale();

    /**
     * 获取已下架商品
     */
    List<LPLiveProductModel>
getProductsNotSale();
```

2.3. 讲解商品

老师/助教可以选中商品进行讲解，学生端 UI 上可以展示被讲解的商品信息。

```
/**
     * 讲解商品
     *
     * @param productId 商品 id
     */
    void requestExplainProduct(String productId);

    /**
     * 获取正在讲解的商品
     *
     * @return
     */
    LPLiveProductModel getExplainProduct();

    /**
     * 讲解商品回调
     *
     * @return
     */
```

```
    Observable<LPLiveProductModel> getObservableOfProductExplain();
```

2.4 其它

```
    /**
     * 获取商品总数
     */
    int getProductCount();

    /**
     * 获取商品总数变更监听
     */
    Observable<Integer> getObservableOfProductCount();

    /**
     * 搜索商品
     *
     * @param query 按商品名称/序号搜索
     * @return 返回服务端匹配的商品信息 LPResSellProductModel.productModels 类型为 List<LPLiveProductModel>
     */
    Observable<LPResSellProductModel> getObservableOfSearchResults(String query);
```

3. 点赞

不同于其它班型的点赞，这里的点赞效果和抖音小红心类似。

```
    /**
```

```
    * 获取当前点赞总数，由 {@link #getObservableOfLiveLikeCountChange} 返回
    */
    void requestLiveLikeCount();

    /**
    * 直播点赞数变更监听
    */
    Observable<Integer> getObservableOfLiveLikeCountChange();

    /**
    * 发送点赞请求
    *
    * @param likeCount 新增的点赞数
    */
    void requestSendLiveLike(int likeCount);

    /**
    * 获取当前点赞数
    */
    int getLiveLikeCount();
```

4. 礼物打赏

4.1. 获取打赏配置

```
/**
    * 后台打赏配置更新监听
    *
    * @return
    */
    Observable<LPRewardConfigResponse> getObservableOfRewardConfigUpdate();
```

LPLiveRewardConfigModel说明：

```
/** 特效图片url */
String imgUrl;
/** 礼物名称 */
String name;
/** 礼物代表价格，价格0则免费；积分则代表积分个数(整值) */
float price;
/** 是否显示浮窗效果 */
boolean isFloat;
/** 是否在直播间显示 */
boolean isShow;
/** 是否显示特效 */
boolean specialEffects;
/** 礼物对应的key值 */
int rewardKey = -1;
/** 打赏人名称 */
String sendUserName;
```

4.2. 绑定手机号

传入当前手机号参数，请求获取短信验证码

```
/**
     * 获取手机验证码
     *
     * @param phoneNumber 手机号
     * @return 是否发送成功
     */
    Observable<Boolean> getVerificationCode(String phoneNumber);
```

校验并绑定手机号，code为上一步短信发送的验证码，绑定成功会返回**user_token**

```
/**
    * 校验验证码登录
    *
    * @param phoneNumber 手机号
    * @param code        验证码
    * @return token
    */
   Observable<String> checkPhoneCode(String phoneNumber, String code);
```

4.3. 请求账户余额

```
/**
    * 请求账户余额
    *
    * @param token 校验成功返回的账户唯一的token
    * @return int->余额，单位分
    */
   Observable<Integer> getAccountBalance(String token);
```

4.4. 发起打赏

若发送免费礼物，则直接使用如下api；若发送付费礼物，需要先检查余额，若余额不足则需要进入充值流程。

```
/**
    * 发起打赏
    *
    * @param lpRewardModel
```

```
     * @return
     */
    Observable<LPRewardResultModel>
startReward(LPRewardModel lpRewardModel);
```

发起打赏

```
liveRoom.liveShowVM.startReward(LPRewardModel)
        .filter {
            // 若是该礼物配置更新，则视为打赏失败，则通
知更新配置
            return@filter if (it.isConfigChange) {
                // 后台配置已更改，触发请求后台配置

liveRoom.liveShowVM.triggerUpdateRewardConfig()
                false
            } else {
                true
            }
        }
        .subscribe({
            // 打赏成功
        }, {
            // 打赏失败
        })
```

LPRewardModel说明：

```
/** 打赏名称："现金"、礼物名称*/
String rewardName;

/** 金额/分 */
int money;

/** 类型，1现金红包，2礼物，3积分 */
```

```
int type;

/** 打赏图片效果，和配置返回值一致 */
String customImg;

/** 打赏特效，默认0，1为有特效，和配置返回值一致 */
boolean specialEffects;

/** 浮窗效果，默认0，1为有浮窗效果，和配置返回值一致 */
boolean isFloat;

/** money大于0时必传 */
String token;
```

打赏通知显示特效回调，回调中可以根据 `LPLiveRewardConfigModel` 的信息显示礼物特效。

```
/**
    * 打赏回调显示特效
    *
    * @return url
    */
   Observable<LPLiveRewardConfigModel> getObservableOfSpecialEffectsDisplay();
```

打赏通知发送聊天消息回调

```
liveRoom.liveShowVM.observableOfSendRewardMes

.observeOn(AndroidSchedulers.mainThread())
        .subscribe {
            // 通过返回LPRewardModel构造LPMessageDataModel
            // ...
```

```
            // 调用sendMessage发送打赏消息，其中LPMessageDataModel中的isFloat字段表示是否显示浮窗效果
routerViewModel.liveRoom.chatVM.sendMessage(m
 lpMessageDataModel)
    }
```

5. 充值

请求预支付相关参数，其中appId为集成端app的微信相关appId，money的单位是分，token为手机号绑定时的user_token，返回的参数为LPRechargeParamsModel

```
liveRoom.liveShowVM.getObservableOfStartRechar
 money, token)
        .subscribe({
            // 获取成功，进入调起微信支付流程
        }, {
            // 获取失败
        })
```

LPRechargeParamsModel说明：

```
/** app唯一标识，由本地提供*/
String appId;
/** 商户号，目前只能由后端返回 */
String partnerId;
/** 预支付ID */
String prepayId;
/** 签名 */
String sign;
/** 订单编号，用于查询订单状态 */
String code;
```

```
// ...
```

调起微信支付：直播带货模版中定义有标准的微信支付调起流程实现接口WePayAPI，可以实现该接口或继承微信支付流程基类BaseWePayImpl实现自己的微信支付流程类

```
public interface WePayAPI {

    /**
     * 获取app唯一的ID，由接入端自己提供
     */
    String getAppId();

    /**
     * 注册app
     *
     * @param context activity
     * @param appId app的唯一标识
     */
    void registerApp(Context context, String appId);

    /**
     * 注册支付回调
     * @param callback
     */
    void registerPayCallback(ResultCallback callback);

    /**
     * 通过IWXApi拉起微信进行支付
     *
     * @param signModel 带签名的参数
     * @return false：调起微信失败
     */
```

```
    boolean requestWePay(SignModel signModel);

    /**
     * 支付回调返回当前支付错误码
     */
    void notifyPaymentCallback(int errCode, String errString);

    /**
     * 是否安装微信app
     */
    boolean isWxAppInstalled();

    /**
     * 是否支持微信支付api
     */
    boolean isWxApiSupported();

    /**
     * unregister from weixin.
     * If you want to use some methods, you may need to be in the method cycle between {@link #registerApp}
     * and {@link #unRegisterApp}
     */
    void unRegisterApp();
}
```

查询订单状态，其中code为预支付接口返回的code参数，token为user_token

```
liveRoom.liveShowVM.checkLastOrderStatus(code, token)
    .subscribe ({
```

```
        showToastMessage(if (it) "支付成功" else "支付失败")
    })
```

6. 回调

6.1. 商品回调

SDK 默认逻辑是打开外部浏览器显示商品页，如果客户自行设置回调后将仅回调事件。

```
CallbackManager.getInstance().setShopProductCalll
```

`LPLiveShowRoomListener.LPShopProductCallback` 定义如下

```
interface LPShopProductCallback {
    /**
     * 点击购买按钮回调
     * @param context
     * @param productModel  商品信息
     */
    void onOpenBrowserToBuy(Context context, LPLiveProductModel productModel);

    /**
     * 点击商品详情回调
     * @param context
     * @param productModel  商品信息
     */
    void onOpenProductDetail(Context context, LPLiveProductModel productModel);

```

```
    /**
     * 点击商品讲解回调
     * @param context
     * @param productModel  商品信息
     */
    void onExplainProduct(Context context, LPLiveProductModel productModel);

    /**
     * 点击商品上架/下架回调
     * @param context
     * @param productModel  商品信息
     * @param onShelf  是否上架
     */
    void onProductShelfStateChanged(Context context, LPLiveProductModel productModel, boolean onShelf);
  }
```

6.2. 礼物回调

点击礼物回调

```
CallbackManager.getInstance().setGiftClickListener(
        LPLiveShowRoomListener.LPRoomGiftClickListener() {
            @Override
            public void onClick(LPLiveRewardConfigModel lpLiveRewardConfigModel) {

            }
        });
```

6.3. 现金打赏回调

现金打赏回调

```
CallbackManager.getInstance().setCashClickListene

LPLiveShowRoomListener.LPRoomCashClickListener
 {
        @Override
        public void onClick(String price) {

        }
    });
```

下载为pdf格式