魅族集成推送服务端JAVA_SDK

JavaPushSdk发布说明(请使用最新版本)

- 中央仓库获取 MVN Repository或者 Central Repository
- 直接下载获取 Java Server SDK

更新日志

[2017-12-06]V1.0.0.20171206

- 1.0.0标准版

[2018-04-18]V1.0.0.20180418

- 增加应用全网推送

[2018-06-12]V1.0.0.20180612

- 增加取消应用全网推送

类型定义

推送服务(IFlymeUpsPush)

> 调用该类实例的方法来推送消息， 构造函数说明如下：

参数名称	类型	必填	默认	描述
appSecret	String	是	null	注册应用appSecret
useSSL	Boolean	否	false	是否使用https接口调用：true 使用https连接，false使用http连接

通知栏消息体(Message)

> 推送消息实体（抽象类）

子类	说明
VarnishedMessage	通知栏消息体
UnVarnishedMessage	透传消息体

通知栏消息(VarnishedMessage)

参数名称	类型	必填	默认	描述
appId	Long	是	null	注册应用appId
title	String	是	null	推送标题，【字数限制1~32】
content	String	是	null	推送内容，【字数限制1~100】
clickType	int	否	0	点击动作 (0,"打开应用"),(1,"打开应用页面"),(2,"打开URI页面")【非必填，默认值为0】
url	String	否	null	URI页面地址，【clickType为打开URI页面时，必填，长度限制1000byte】
parameters	JSONObject	否	null	透传参数 【JSON格式，非必填】
activity	String	否	null	应用页面地址，【clickType为打开应用页面时，格式pkg.activity eg: com.meizu.upspushdemo.TestActivity 必填】
isOffLine	Boolean	否	true	是否进离线消息, (false 否 true 是) 【非必填，默认值为true】
validTime	int	否	24	有效时长 (1~72小时内的正整数)，【isOffLine值为true时，必填，值的范围1~72】
pushTimeType	int	否	0	定时推送 (0, "即时"),(1, "定时")，【只对全部用户推送生效】
startTime	Date	否	null	任务定时开始时间,【非必填，pushTimeType为True必填】只对全部用户推送生效
isFixSpeed	Boolean	否	false	是否定速推送，【非必填，默认值为False】
fixSpeedRate	Long	否	0	定速速率,【isFixSpeed为true时，必填】
isSuspend	Boolean	否	true	是否通知栏悬浮窗显示 (true显示，false不显示) 【非必填，默认True】
isClearNoticeBar	Boolean	否	true	是否可清除通知栏 (true可以，false不可以) ，【非必填，默认true】
vibrate	Boolean	否	true	震动 (false关闭 true 开启) ，【非必填，默认true】
lights	Boolean	否	true	闪光 (false关闭 true 开启) ，【非必填，默认true】
sound	Boolean	否	true	声音 (false关闭 true 开启) ，【非必填，默认true】

透传消息(UnVarnishedMessage)

参数名称	类型	必填	默认	描述
appId	Long	是	null	注册应用appId

参数名称	类型	必填	默认	描述
title	String	否	null	推送标题,任务推送建议填写，方便数据查询,【字数限制1~32】
content	String	是	null	推送内容,【必填，字数限制3800byte以内】
isOffLine	Boolean	否	true	是否进离线消息,【非必填，默认为true】
validTime	int	否	24	有效时长 (1~72小时内的正整数),【isOffLine值为true时，必填，值的范围1--72】
pushTimeType	int	否	0	定时推送 (0, "即时"),(1, "定时"), 【只对全部用户推送生效】
startTime	Date	否	null	任务定时开始时间，【pushTimeType为1必填】只对全部用户推送生效
isFixSpeed	Boolean	否	false	是否定速推送,【非必填，默认值为false】
fixSpeedRate	Long	否	0	定速速率 【isFixSpeed为true时，必填】

接口返回值(ResultPack)

方法名称	类型	描述
code()	String	接口响应码
Comment()	String	接口响应描述
value()	T	接口响应内容
errorCode()	Enum	接口响应异常枚举 详见ErrorCode

消息推送结果(PushResult)

方法名称	类型	描述
getMsgId()	String	推送消息ID，用于推送流程明细排查
getRespTarget()	Map	推送目标结果状态(KEY：推送响应码 VALUE：响应码对应的目标用户)
logs	Map	app渠道推送日志，用于问题排查(KEY：渠道 value：log)

接口响应码定义(ErrorCode)

名称	Code	Commen
UNKNOWN_ERROR	-1	未知错误
SUCCESS	200	成功
SYSTEM_ERROR	1001	系统错误

名称	Code	Commen
SYSTEM_BUSY	1003	服务器忙
PARAMETER_ERROR	1005	参数错误，请参考API文档
INVALID_SIGN	1006	签名认证失败
INVALID_APPLICATION_ID	110000	appId不合法
INVALID_APPLICATION_KEY	110001	appKey不合法
UNSUBSCRIBE_PUSHID	110002	pushId未注册
INVALID_PUSHID	110003	pushId非法
PARAM_BLANK	110004	参数不能为空
APP_REQUEST_EXCEED_LIMIT	110010	应用请求频率过快
APP_PUSH_TIME_EXCEED_LIMIT	110051	超过该应用的次数限制
APP_REQUEST_PUSH_LIMIT	110019	超过该应用每天推送次数限制
INVALID_APPLICATION_PACKAGENAME	110031	packageName不合法
INVALID_TASK_ID	110032	非法的taskId
INVALID_APPLICATION_SECRET	110033	非法的appSecret
DIRECT_OUT_LIMIT	110053	透传超过限制

推送响应码定义（PushResponseCode）

名称	Code	Commen
RSP_NO_AUT	201	没有权限，服务器主动拒绝
RSP_DB_ERROR	501	推送消息失败（db_error）
RSP_INTERNAL_ERROR	513	推送消息失败
RSP_SPEED_LIMIT	518	推送超过配置的速率
RSP_OVERFLOW	519	推送消息失败服务过载
RSP_UNSUBSCRIBE_PUSHID	110002	pushId未订阅(包括推送开关关闭的设备)
RSP_INVALID_PUSHID	110003	pushId非法
RSP_UNSUBSCRIBE_ALIAS	110005	别名未订阅(包括推送开关关闭的设备)

推送类型（PushType）

枚举	类型	描述
STATUSBAR	Enum	通知栏消息类型

枚举	类型	描述
DIRECT	Enum	透传消息类型

渠道类型（CP）

枚举	类型	描述
MEIZU	Enum	魅族
XIAOMI	Enum	小米
HUAWEI	Enum	华为

任务推送统计（天）（DailyPushStatics）

名称	类型	描述
date	Date	日期
pushedNo	Long	推送数
acceptNo	Long	接收数
clickNo	Long	点击数

接口说明

非任务推送

描述

> 向指定的pushId推送消息

> 注：推送平台使用pushId来标识每个独立的用户，每一台终端上每一个app拥有一个独立的pushId

应用场景

> - 场景1：查找手机业务需要远程定位位置，可发送消息指令到对应的设备
> - 场景2：社区用户回帖消息提醒，用户对发表的帖子有最新回复时，消息提醒发帖者

pushId通知栏消息推送（pushMessage）

- 接口说明

接口	说明
`ResultPack<PushResult> pushMessage(VarnishedMessage message, List<String> pushIds)`	推送通知栏消息

接口	说明
`ResultPack<PushResult> pushMessage(VarnishedMessage message, List<String> pushIds, int retries)`	推送通知栏消息

- 参数说明

参数名称	类型	必需	默认	描述
message	VarnishedMessage	是	null	推送消息
pushIds	List	是	null	推送目标，一批最多不能超过1000个
retries	int	否	0	超时or异常重试次数

- 返回值

```
PushResult

msgId;  推送消息ID，用于推送流程明细排查
respTarget;  推送目标结果状态(key：推送响应码  value：响应码对应的目标用户 )

注：只返回不合法、超速以及推送失败的目标用户，业务一般对超速的pushId处理。
```

- 示例

```
    /**
     * 通知栏消息推送（pushMessage）
     * @throws Exception
     */
    @Test
    public void testVarnishedMessagePush() throws Exception {
        //推送对象
        IFlymeUpsPush push = new IFlymeUpsPush(APP_SECRET_KEY);

        //组装消息
        VarnishedMessage message = new
VarnishedMessage.Builder().appId(appId)
                .title("Java SDK 推送标题").content("消息内容")
                .build();

        //目标用户
        List<String> pushIds = new ArrayList<String>();
        pushIds.add("pushId_1");
        pushIds.add("pushId_2");
        // 1 调用推送服务
        ResultPack<PushResult> result = push.pushMessage(message, pushIds);
        if (result.isSucceed()) {
            // 2 调用推送服务成功 （其中map为设备的具体推送结果，一般业务针对超速的
code类型做处理）
```

```
            PushResult pushResult = result.value();
            String msgId = pushResult.getMsgId();//推送消息ID，用于推送流程明细
排查
            Map<String, List<String>> targetResultMap =
pushResult.getRespTarget();//推送结果，全部推送成功，则map为empty
            System.out.println("push result:" + pushResult);
            if (targetResultMap != null && !targetResultMap.isEmpty()) {
                System.err.println("push fail token:" + targetResultMap);
            }
        } else {
            // 调用推送接口服务异常 eg: appId、appKey非法、推送消息非法.....
            // result.code(); //服务异常码
            // result.comment();//服务异常描述
            System.out.println(String.format("pushMessage error code:%s
comment:%s", result.code(), result.comment()));
        }
    }
```

pushId透传消息推送（pushMessage）

- 接口说明

接口	说明
ResultPack<PushResult> pushMessage(UnVarnishedMessage message, List<String> pushIds)	推送透传消息
ResultPack<PushResult> pushMessage(UnVarnishedMessage message, List<String> pushIds, int retries)	推送透传消息

- 参数说明

参数名称	类型	必需	默认	描述
message	UnVarnishedMessage	是	null	推送消息
pushIds	List	是	null	推送目标，一批最多不能超过1000个
retries	int	否	0	超时or异常重试次数

- 返回值

```
PushResult

msgId;  推送消息ID，用于推送流程明细排查
respTarget;  推送目标结果状态(key：推送响应码  value：响应码对应的目标用户 )
注：只返回不合法、超速以及推送失败的目标用户，业务一般对超速的pushId处理。
```

- 示例

```
    /**
     * 透传消息推送（pushMessage）
     * @throws Exception
     */
    @Test
    public void testUnVarnishedMessagePush() throws Exception {
        //推送对象
        IFlymeUpsPush push = new IFlymeUpsPush(APP_SECRET_KEY);
        //组装透传消息
        UnVarnishedMessage message = new UnVarnishedMessage.Builder()
                .appId(appId)
                .title("Java SDK 透传推送标题")
                .content("Java Sdk透传推送内容")
                .build();

        //目标用户
        List<String> pushIds = new ArrayList<String>();
        pushIds.add("pushId_1");
        pushIds.add("pushId_2");

        // 1 调用推送服务
         ResultPack<PushResult> result = push.pushMessage(message,
pushIds);
        if (result.isSucceed()) {
            // 2 调用推送服务成功 （其中map为设备的具体推送结果，一般业务针对超速的
code类型做处理）
            PushResult pushResult = result.value();
            String msgId = pushResult.getMsgId();//推送消息ID，用于推送流程明细
排查
            Map<String, List<String>> targetResultMap =
pushResult.getRespTarget();//推送结果，全部推送成功，则map为empty
            System.out.println("push result:" + pushResult);
            if (targetResultMap != null && !targetResultMap.isEmpty()) {
                System.err.println("push fail token:" + targetResultMap);
            }
        } else {
            // 调用推送接口服务异常 eg: appId、appKey非法、推送消息非法.....
            // result.code(); //服务异常码
            // result.comment();//服务异常描述
            System.out.println(String.format("pushMessage error code:%s
comment:%s", result.code(), result.comment()));
        }
    }
```

别名通知栏消息推送（pushMessageByAlias）

- 接口说明

接口	说明

接口	说明
`ResultPack<PushResult> pushMessageByAlias(VarnishedMessage message, List<String> alias)`	推送通知栏消息
`ResultPack<PushResult> pushMessageByAlias(VarnishedMessage message, List<String> alias, int retries)`	推送通知栏消息

- 参数说明

参数名称	类型	必需	默认	描述
message	VarnishedMessage	是	null	推送消息
alias	List	是	null	推送目标，一批最多不能超过1000个
retries	int	否	0	超时or异常重试次数

- 返回值

```
PushResult

msgId;  推送消息ID，用于推送流程明细排查
respTarget;  推送目标结果状态(key：推送响应码  value：响应码对应的目标用户 )
注：只返回不合法、超速以及推送失败的目标用户
```

- 示例

```
/**
 * 别名通知栏消息推送（pushMessage）
 *
 * @throws Exception
 */
    @Test
    public void testVarnishedMessagePushByAlias() throws Exception {
        //推送对象
        IFlymeUpsPush push = new IFlymeUpsPush(APP_SECRET_KEY);

        //组装消息
        VarnishedMessage message = new
VarnishedMessage.Builder().appId(appId)
                .title("Java SDK 推送标题").content("Java SDK 推送内容")
                .build();

        //目标用户
        List<String> alias = new ArrayList<String>();
        alias.add("alias1");
        alias.add("alias2");

        // 1 调用推送服务
        ResultPack<PushResult> result = push.pushMessageByAlias(message,
```

```
alias);
        // 2 处理推送结果
        if (result.isSucceed()) {
            // 2 调用推送服务成功 （其中map为设备的具体推送结果，一般业务针对超速的
code类型做处理）
            PushResult pushResult = result.value();
            String msgId = pushResult.getMsgId();//推送消息ID，用于推送流程明细
排查
            Map<String, List<String>> targetResultMap =
pushResult.getRespTarget();//推送结果，全部推送成功，则map为empty
            System.out.println("push result:" + pushResult);
            if (targetResultMap != null && !targetResultMap.isEmpty()) {
                System.err.println("push fail token:" + targetResultMap);
            }
        } else {
            // 调用推送接口服务异常 eg: appId、appKey非法、推送消息非法.....
            // result.code(); //服务异常码
            // result.comment();//服务异常描述
            System.out.println(String.format("pushMessage error code:%s
comment:%s", result.code(), result.comment()));
        }
    }
```

别名透传消息推送（pushMessageByAlias）

- 接口说明

接口	说明
ResultPack<PushResult> pushMessageByAlias(UnVarnishedMessage message, List<String> alias)	推送透传消息
ResultPack<PushResult> pushMessageByAlias(UnVarnishedMessage message, List<String> alias, int retries)	推送透传消息

- 参数说明

参数名称	类型	必需	默认	描述
message	UnVarnishedMessage	是	null	推送消息
alias	List	是	null	推送目标，一批最多不能超过1000个
retries	int	否	0	超时or异常重试次数

- 返回值

```
PushResult

msgId;  推送消息ID，用于推送流程明细排查
```

respTarget; 推送目标结果状态(key：推送响应码 value：响应码对应的目标用户)
注：只返回不合法、超速以及推送失败的目标用户

- 示例

```
/**
 * 别名透传推送
 *
 * @throws Exception
 */
@Test
public void testUnVarnishedMessagePushByALias() throws Exception {
    //推送对象
    IFlymeUpsPush push = new IFlymeUpsPush(APP_SECRET_KEY);
    //组装透传消息
    UnVarnishedMessage message = new UnVarnishedMessage.Builder()
            .appId(appId)
            .title("Java SDK 透传推送标题")
            .content("Java Sdk透传推送内容")
            .build();

    //目标用户
    List<String> alias = new ArrayList<String>();
    alias.add("alias");
    alias.add("alias2");

    // 1 调用推送服务
    ResultPack<PushResult> result = push.pushMessageByAlias(message,
alias);
    if (result.isSucceed()) {
        // 2 调用推送服务成功 （其中map为设备的具体推送结果，一般业务针对超速的code类
型做处理）
        PushResult pushResult = result.value();
        String msgId = pushResult.getMsgId();//推送消息ID，用于推送流程明细排查
        Map<String, List<String>> targetResultMap =
pushResult.getRespTarget();//推送结果，全部推送成功，则map为empty
        System.out.println("push result:" + pushResult);
        if (targetResultMap != null && !targetResultMap.isEmpty()) {
            System.err.println("push fail token:" + targetResultMap);
        }
    } else {
        // 调用推送接口服务异常 eg: appId、appKey非法、推送消息非法.....
        // result.code(); //服务异常码
        // result.comment();//服务异常描述
        System.out.println(String.format("pushMessage error code:%s
comment:%s", result.code(), result.comment()));
    }
}
```

应用全网推送（pushToApp）

- 接口说明

接口	说明
ResultPack<Long> pushToApp(PushType pushType, Message message)	应用全网推送

- 参数说明

参数名称	类型	必需	默认	描述
message	UnVarnishedMessage	是	null	推送消息

- 返回值

```
Long  任务ID
```

- 示例

```
    /**
     * 应用全网推送
     * @throws IOException
     */
    @Test
    public void testPushToApp() throws IOException {
        //推送对象
        IFlymeUpsPush push = new IFlymeUpsPush(APP_SECRET_KEY);
        JSONObject param = new JSONObject();
        param.put("key", "v1");
        param.put("k2", "v2");
        param.put("k3", "v3");

        //组装消息
        VarnishedMessage message = new
VarnishedMessage.Builder().appId(appId)
                .title("Java SDK 推送标题").content("消息内容")
                .build();

        // 1 调用推送服务
        ResultPack<Long> result = push.pushToApp(PushType.STATUSBAR,
message);
        System.out.println(result);
    }
```

取消推送任务（cancelTaskPush）

- 接口说明

接口	说明
ResultPack<Boolean> cancelTaskPush(PushType pushType, long appId, long taskId)	只针对全部用户推送且推送状态为待推送或者推送中的任务取消

- 参数说明

参数名称	类型	必需	默认	描述
pushType	PushType	是	null	消息类型
appId	Long	是	null	应用ID
taskId	Long	是	null	任务ID

- 返回值

```
Boolean  true:成功  false：失败
```

- 示例

```
    /**
     * 取消推送任务(cancelTaskPush) 只针对全网推送生效
     *
     * @throws IOException
     */
    @Test
    public void testCancelTaskPush() throws IOException {
        //推送对象
        IFlymeUpsPush push = new IFlymeUpsPush(APP_SECRET_KEY);
        long taskId = 60;
        ResultPack<Boolean> resultPack =
push.cancelTaskPush(PushType.STATUSBAR, appId, taskId);
        System.out.println(resultPack);
    }
```

获取应用推送统计(dailyPushStatics)

- 接口说明

接口	说明
public ResultPack<List<DailyPushStatics>> dailyPushStatics(long appId, Date startTime, Date endTime)	获取应用推送统计

- 参数说明

参数名称	类型	必需	默认	描述

参数名称	类型	必需	默认	描述
appId	Long	是	null	应用ID
startTime	Date	是	null	开始日期
endTime	Date	是	null	结束日期

- 返回值

```
List<DailyPushStatics>
```

- 示例

```
    /**
     * 获取任务统计结果
     *
     * @throws IOException
     */
     @Test
     public void testDailyPushStatics() throws IOException {
         //推送对象
         IFlymeUpsPush push = new IFlymeUpsPush(APP_SECRET_KEY);
         Date startTime = DateUtils.str2Date("2017-06-03", "yyyy-MM-dd");
         Date endTime = DateUtils.str2Date("2017-06-10", "yyyy-MM-dd");
         ResultPack<List<DailyPushStatics>> resultPack =
push.dailyPushStatics(appId, startTime, endTime);
         System.out.println(resultPack);
     }
```