

2017-12-01Dcloud打包常见问题

Dcloud 打包 APP 常见问题

1 . Android 手机的 back 键直接退出应用的问题

```
<script>

    document.addEventListener('plusready', function() {

        var webview = plus.webview.currentWebview();

        plus.key.addEventListener('backbutton', function() {

            webview.canBack(function(e) {

                if(e.canBack) {

                    webview.back();

                } else {

                    //webview.close(); //hide,quit

                    //plus.runtime.quit();

                    mui.plusReady(function() {

                        //首页返回键处理

                        //处理逻辑：1秒内，连续两次按返回键，则退出应用；

                        var first = null;

                        plus.key.addEventListener('backbutton', function() {

                            //首次按键，提示‘再按一次退出应用’

                            if (!first) {

                                first = new Date().getTime();

                                mui.toast('再按一次退出应用');

                                setTimeout(function() {

                                    first = null;

                                }, 1000);

                            } else {

                                if (new Date().getTime() - first < 1500) {

                                    plus.runtime.quit();

                                }

                            }

                        }, false);

                    });

                }

            })

        });

    });

</script>
```

2 . 个推iOS 和 Android 端的消息处理 （Dcloud 推送开发指南 <http://ask.dcloud.net.cn/article/34>）

最近做的app中使用到了推送，iOS 和 Android 的处理方式不一样，Android 只需要把平台生成的 appid, appkey, appsecret 添加到 sdk 配置里面就可以直接推送消息，而 iOS 只支持传透消息，iOS 还区分应用在线和不在线，两个系统的处理方式完全不同；

1 . Android ：应用在前台会直接受到传透消息，然后自己去处理；

如果后台不杀死push进程收到透传我去创建本地推送在通知栏显示plus.push.createMessage(str, jsonData, options); 杀死push进程什么也收不到了。当再次打开应用会收到之前的推送。（问了官方说android杀死应用就是收不到郁闷）；

2 . iOS ：应用在前台会直接收到透传然后自己去处理；

应用关闭和在后台就需要将消息推送个APNS然后APNS在推送给iOS手机。（如果后台和关闭状态收不到推送就有可能是 iOS的正式配置有问题 你需要一个iOS的哥们帮你了）。

3 . android的push监听可以写很多都没事而iOS你多写了它会多次调用push监听方法。

```
var Push = function() {

}

/**
```

```

*
* 推送的监听
*/

Push.prototype.pushListener = function () {
    var pointer = this;
    plus.push.addEventListener("click", function (msg) {
        switch (msg.payload) {
            case "LocalMSG":
                ApiConfig.staticToast("点击本地创建消息启动: ");
                break;
            default:
                ApiConfig.staticToast("点击离线推送消息启动: ");
                break;
        }
        if (msg.payload) {
            pointer.handle(msg);
        }
    }, false);
    plus.push.addEventListener("receive", function (msg) {
        if (msg.aps) { // Apple APNS message
            ApiConfig.staticToast("接收到在线APNS消息: ");
        } else {
            ApiConfig.staticToast("接收到在线透传消息: ");
        }
        if (plus.os.name == 'iOS') {
            if (msg.payload) {
                pointer.notificationMessage(msg);
            }
        } else {
            pointer.notificationMessage(msg);
        }
    }, false);
}

/**
* 根据推送消息在通知栏中显现对应的提示
* @param {Object} msg
*/
Push.prototype.notificationMessage = function(msg) {
    ApiConfig.staticToast(msg, 1);
    ApiConfig.staticIsDebug("notificationMessage", msg, 1);
    var content = ''; // 你要展示的提示
    var pointer = this;
    var jsonData = '';
    switch(plus.os.name) {
        case "Android":
            jsonData = eval("(" + msg.payload + ")");
            break;
        case "iOS":
            jsonData = msg.payload;
            break;
    }
    pointer.createLocalPushMsg(msg, content);
}

/**
* 处理透传消息

```

```

* @param {Object} msg
* @param {Object} content
*/
Push.prototype.createLocalPushMsg = function(msg, content) {
    //创建一个符合你自己要显示推送通知
    pointer.createMessage(str, jsonData, options);
}

/**
* 创建本地推送消息
* @param {Object} str
* @param {Object} jsonData
* @param {Object} options
*/
Push.prototype.createMessage = function(str, jsonData, options) {
    switch(plus.os.name) {
        case "Android":
            jsonData = jsonData;
            break;
        case "iOS":
            jsonData = jsonData.eid;
            break;
    }
    plus.push.createMessage(str, jsonData, options);
}

/**
* 处理通知方法
* @param {Object} msg
*/
Push.prototype.handle = function(msg) {
    //清除ios推送小红点
    this.cancelPushClear();
}

Push.prototype.cancelPushClear = function () {
    plus.push.clear();
}

```