原生与web端方法文档

1. 单个播放

//单个播放

function oneplayer(id) {

if (/(iPhone|iPad|iPod|iOS)/i.test(navigator.userAgent)) {

//判断iPhone|iPad|iPod|iOS

} else if (/(Android)/i.test(navigator.userAgent)) {

//判断Android

Android.playMusic(JSON.stringify({

mUrl: $("#song" + id).attr("data-murl"),

mCoverImageUrl: $("#song" + id).attr("data-mcoverimageurl"),

mTitle: $("#song" + id).attr("data-mtitle"),

mAlbum: $("#song" + id).attr("data-malbum"),

mArtist: $("#song" + id).attr("data-martist")

}));

}

}

安卓层面实现

方法：

Android.playMusic

参数：

mUrl：曲目地址

mCoverImageUrl：曲目封面

mTitle：曲目标题

mAlbum：曲目所属专辑

mArtist：曲目作者

参数形式：

通过json的字符串传递

1. 全部播放 待定

//判断客户端进入页面

function checkAIEnter(back, conText, shareUrl, layerDepth) {

if (/(iPhone|iPad|iPod|iOS)/i.test(navigator.userAgent)) {

//判断iPhone|iPad|iPod|iOS

alert(conText);

// iPhone.enterPlayView(back, conText, shareUrl, layerDepth);

} else if (/(Android)/i.test(navigator.userAgent)) { //判断Android

alert(conText);

// Android.enterPlayView(back, conText, shareUrl, layerDepth);

} else { //pc

alert(conText)

};

}

安卓层面实现

方法：

Android.enterPlayView

参数：

back：是否有返回按钮

有传递“back”

没有传递“”或者“scan”

conText：页面标题

shareUrl：分享地址

传递“option”出现分享按钮

layerDepth：层级

参数形式：

通过参数一一对应形式 不是为json字符串

4、

//返回

window.loadHistoryPage = function() { $("#mainbox").load(window.localStorage.getItem("historyhtml"));

}

安卓层面实现

方法：

Android.loadHistoryPage

历史页面由web端来记录，只需要调用源码的返回功能即可

5、一期扫二维码

window.openUrl = function(melodyUrl) {

$.ajax({

type: 'POST',

url: url + 'melody/query.do',

data: {

id: melodyUrl.melodyUrl

},

dataType: 'json',

success: function(data) {

},

error: function(xhr, error, ex) { if

}

});

}

安卓层面实现

方法：

window.openUrl

参数：

melodyUrl扫描二维码返回的id 以对象形式返回id:melodyUrl.melodyUrl

1. 充值

后台接口：

app/accountRechargeRecord/add.do

参数：

accountId：用户id

rechargeMoney：充值钱

rechargePlatform：充值方式 例如微信，支付宝

还在和康哥商定如何和源码方面交互

1. 下载 待定