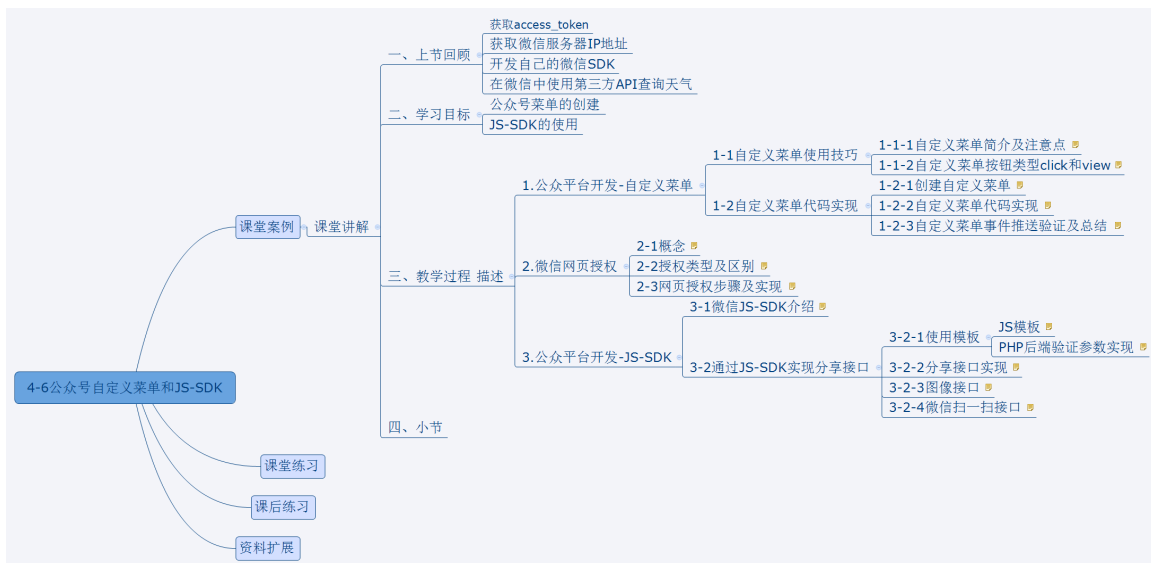


4-6 公众号自定义菜单和JS-SDK

4-6 公众号自定义菜单和JS-SDK	1
1. 课堂案例	2
课堂讲解	2
一、上节回顾	2
获取access_token.....	2
获取微信服务器IP地址	2
开发自己的微信SDK.....	2
在微信中使用第三方API查询天气	2
二、学习目标	2
公众号菜单的创建	2
JS-SDK的使用.....	2
三、教学过程 描述	2
1. 公众平台开发-自定义菜单	2
2. 微信网页授权	9
3. 公众平台开发-JS-SDK	13
四、小节	20
2. 课堂练习	20
3. 课后练习	20
4. 资料扩展	20



1. 课堂案例

课堂讲解

一、上节回顾

获取access_token

获取微信服务器IP地址

开发自己的微信SDK

在微信中使用第三方API查询天气

二、学习目标

公众号菜单的创建

JS-SDK的使用

三、教学过程描述

1. 公众平台开发-自定义菜单

1-1 自定义菜单使用技巧

1-1-1 自定义菜单简介及注意点

注意事项:

- 1、自定义菜单最多包括3个一级菜单, 每个一级菜单最多包含5个二级菜单。
- 2、一级菜单最多4个汉字, 二级菜单最多7个汉字, 多出来的部分将会以“...”代替。
- 3、创建自定义菜单后, 菜单的刷新策略是, 在用户进入公众号会话页或公众号profile页时, 如果发现上一次拉取菜单的请求在5分钟以前, 就会拉取一下菜单, 如果菜单有更新, 就会刷新客户端的菜单。测试时可以尝试取消关注公众账号后再次关注, 则可以看到创建后的效果。

1-1-2 自定义菜单按钮类型click和view

自定义菜单接口可实现多种类型按钮

- 1、click: 点击推事件用户点击click类型按钮后, 微信服务器会通过消息接口推送消息类型为event的结构给开发者(参考消息接口指南), 并且带上按钮中开发者填写的key值, 开发者可以通过自定义的key值与用户进行交互;
- 2、view: 跳转URL用户点击view类型按钮后, 微信客户端将会打开开发者在按钮中填写的网页URL, 可与网页授权获取用户基本信息接口结合, 获得用户基本信息。
- 3、scancode_push: 扫码推事件用户点击按钮后, 微信客户端将调起扫一扫工具, 完成扫码操作后显示扫描结果(如果是URL, 将进入URL), 且会将扫码的结果传给开发者, 开发者可以下发消息。

1-2 自定义菜单代码实现

1-2-1 创建自定义菜单

原理: **POST** **一个json数据包给到指定的接口: https://api.weixin.qq.com/cgi-bin/menu/create?access_token=ACCESS_TOKEN**

JSON 数据包格式:

案例一:

{

```

"button":[
{
  "type":"click",
  "name":"今日歌曲",
  "key":"V1001_TODAY_MUSIC"
},
{
  "name":"菜单",
  "sub_button":[
    {
      "type":"view",
      "name":"搜索",
      "url":"http://www.soso.com/"
    },
    {
      "type":"miniprogram",
      "name":"wxa",
      "url":"http://mp.weixin.qq.com"
      "appid":"wx286b93c14bbf93aa",
      "pagepath":"pages/lunar/index.html"
    },
    {
      "type":"click",
      "name":"赞一下我们",
      "key":"V1001_GOOD"
    }
  ]
}
]

```

案例二：

```

{
  "button": [
    {
      "name": "扫码",
      "sub_button": [
        {
          "type": "scancode_waitmsg",
          "name": "扫码带提示",
          "key": "rselfmenu_0_0",
          "sub_button": [ ]
        }
      ]
    }
  ]
}

```

```

    },
    {
        "type": "scancode_push",
        "name": "扫码推事件",
        "key": "rselfmenu_0_1",
        "sub_button": [ ]
    }
]
},
{
    "name": "发图",
    "sub_button": [
        {
            "type": "pic_sysphoto",
            "name": "系统拍照发图",
            "key": "rselfmenu_1_0",
            "sub_button": [ ]
        },
        {
            "type": "pic_photo_or_album",
            "name": "拍照或者相册发图",
            "key": "rselfmenu_1_1",
            "sub_button": [ ]
        },
        {
            "type": "pic_weixin",
            "name": "微信相册发图",
            "key": "rselfmenu_1_2",
            "sub_button": [ ]
        }
    ]
},
{
    "name": "发送位置",
    "type": "location_select",
    "key": "rselfmenu_2_0"
},
{
    "type": "media_id",
    "name": "图片",
    "media_id": "MEDIA_ID1"
},
{

```

```

        "type": "view_limited",
        "name": "图文消息",
        "media_id": "MEDIA_ID2"
    }
]
}

```

参数说明:

参数	是否必须	说明
button	是	一级菜单数组，个数应为1~3个
sub_button	否	二级菜单数组，个数应为1~5个
type	是	菜单的响应动作类型，view表示网页类型，click表示点击类型，miniprogram表示小程序类型
name	是	菜单标题，不超过16个字节，子菜单不超过60个字节
key	click等点击类型必须	菜单KEY值，用于消息接口推送，不超过128字节
url	view、miniprogram类型必须	网页链接，用户点击菜单可打开链接，不超过1024字节。type为miniprogram时，不支持小程序的老版本客户端将打开本url。
media_id	media_id类型和view_limited类型必须	调用新增永久素材接口返回的合法media_id
appid	miniprogram类型必须	小程序的appid（仅认证公众号可配置）
pagepath	miniprogram类型必须	小程序的页面路径

1-2-2自定义菜单代码实现

关键代码:

```
/**
 * 创建微信公众号菜单
 * @param Array $arr_menuConfig 需要创建菜单的数组配置
 * @return bool 成功返回true,失败返回false
 */
public function createMenu($arr_menuConfig){
    /*步骤
    * 1. 取得access_token
    * 2. 创建菜单json数据
    * 3. 请求接口 : https://api.weixin.qq.com/cgi-bin/menu/create?access_token=ACCESS_TOKEN
    */

    //1. 取得access_token
    $access_token = $this->getToken();
    if(!$access_token){
        exit("access_token 不存在");
    }

    //2.创建菜单json数据, 注意中文字符处理, JSON_UNESCAPED_UNICODE 不将中文转成unicode php版本大于 5.4 才有此参数
    $post = json_encode($arr_menuConfig,JSON_UNESCAPED_UNICODE);

    //3 请求接口
    $api = "https://api.weixin.qq.com/cgi-bin/menu/create?access_token=".$access_token; //API接口地址
    $return = $this->postRequest($api,$post); //请求接口

    if($return['errcode']==0){
        exit('菜单创建成功');
    }else{
        exit($this->getErrorMsg($return['errcode'])); //提示错误信息
    }
}
```

1-2-3自定义菜单事件推送验证及总结

推送XML数据包示例:点击菜单拉取消息时的事件推送

```
<xml>
  <ToUserName><![CDATA[toUser]]></ToUserName>
  <FromUserName><![CDATA[FromUser]]></FromUserName>
  <CreateTime>123456789</CreateTime>
  <MsgType><![CDATA[event]]></MsgType>
  <Event><![CDATA[CLICK]]></Event>
  <EventKey><![CDATA[EVENTKEY]]></EventKey>
</xml>
```

参数说明:

参数	描述
ToUserName	开发者微信号
FromUserName	发送方帐号(一个OpenID)
CreateTime	消息创建时间(整型)

MsgType	消息类型, event
Event	事件类型, CLICK
EventKey	事件KEY值, 与自定义菜单接口中KEY值对应

点击菜单跳转链接时的事件推送：推送XML数据包示例：

```
<xml>
  <ToUserName><![CDATA[toUser]]></ToUserName>
  <FromUserName><![CDATA[FromUser]]></FromUserName>
  <CreateTime>123456789</CreateTime>
  <MsgType><![CDATA[event]]></MsgType>
  <Event><![CDATA[VIEW]]></Event>
  <EventKey><![CDATA[www.qq.com]]></EventKey>
  <MenuId>MENUID</MenuId>
</xml>
```

参数说明：

参数	描述
ToUserName	开发者微信号
FromUserName	发送方帐号(一个OpenID)
CreateTime	消息创建时间(整型)
MsgType	消息类型, event
Event	事件类型, VIEW
EventKey	事件KEY值, 设置的跳转URL
MenuID	

指菜单ID, 如果是个性化菜单, 则可以通过这个字段, 知道是哪个规则的菜单被点击了。

关键代码请看DEMO


```

//菜单点击事件
private function _click(){
    //根据菜单创建时定义时的key值，进行事件处理，此处是回复当前用户的openid
    //取得xml数据数组
    $arr_xml = $this->arr_xml;
    //判断数据数组
    if(!is_array($arr_xml) || count($arr_xml) <= 0){
        return;
    }

    //获取数据
    $user_uid = $arr_xml['FromUserName']; //微信用ID
    $key = $arr_xml['EventKey']; //菜单定义的key值

    //通过key值来回复特定内容
    if($key == 'get_openid'){
        A('Message','Weixin')->textMsg("你的openid为: ".$user_uid);
    }
}

//菜单页面跳转事件
private function _view(){
    //根据菜单创建时定义时的key值，进行事件处理，一般用于统计功能
    //取得xml数据数组
    $arr_xml = $this->arr_xml;
    //判断数据数组
    if(!is_array($arr_xml) || count($arr_xml) <= 0){
        return;
    }

    //获取数据
    $key = $arr_xml['EventKey']; //菜单定义的key值

    //通过跳转的页面URL
    $this->writeLog('0',$key);
}
}

```

2.微信网页授权

2-1概念

什么是授权：

就是我们的网页 在 微信客户端中打开，我们的网页可以申请获取，当前打开此页面的用户的微信用户资料，而要获取这些用户数据是需要授权才能获得，申请这个过程叫做授权

配置回调域名

2、以 `snsapi_userinfo` 为scope发起的网页授权，是用来获取用户的基本信息的。但这种授权需要用户手动同意，并且由于用户同意过，所以无须关注，就可在授权后获取该用户的基本信息。

注意：

用户管理类接口中的“获取用户基本信息接口”，是在用户和公众号产生消息交互或关注后事件推送后，才能根据用户OpenID来获取用户基本信息。这个接口，包括其他微信接口，都是需要该用户（即openid）关注了公众号后，才能调用成功的。

关于 网页授权access_token 和 普通access_token 的区别

1、微信网页授权是通过OAuth2.0机制实现的，在用户授权给公众号后，公众号可以获取到一个网页授权特有的接口调用凭证（网页授权access_token），通过网页授权access_token可以进行授权后接口调用，如获取用户基本信息；

2、其他微信接口，需要通过基础支持中的“获取access_token”接口来获取到的普通access_token调用。

关于特殊场景下的静默授权

1、上面已经提到，对于以snsapi_base为scope的网页授权，就静默授权的，用户无感知；

2、对于已关注公众号的用户，如果用户从公众号的 会话 或者 自定义菜单 进入本公众号的网页授权页，即使是scope为snsapi_userinfo，也是静默授权，用户无感知。

2-3网页授权步骤及实现

具体而言，网页授权流程分为四步：

- 1、引导用户进入授权页面同意授权，获取code
- 2、通过code换取网页授权access_token（与基础支持中的access_token不同）
- 3、如果需要，开发者可以刷新网页授权access_token，避免过期
- 4、通过网页授权access_token和openid获取用户基本信息（支持UnionID机制）

关键代码：

```
//获取用户信息
public function getUserInfo(){
    $oauth = A('Oauth','Weixin'); //实例化网页授权
    $curr_url = "http://".$_SERVER['HTTP_HOST'].$_SERVER['REQUEST_URI']; //当前页面URL

    /**
     * 实现功能，我们每小时授权一次，即每小时与微信服务器同步一次用户数据，以便更新用户数据
     * 步骤：
     * 1. 判断上次更新数据是否是1小时以前
     * 2. 如果超过1小时，执行如下授权操作，否则跳到第4步
     *     1》 引导用户授权
     *     2》 通过 code 获取 access_token 和 openid
     *     3》 通过access_token 和 openid 获取用户信息
     * 3. 执行完2后将用户信息缓存，过期时间设置为一个小时，并将用户信息存储到数据库
     */

    //1. 判断上次更新数据是否是1小时以前
    $userinfo = session('openid'); //检查缓存是否存在用户信息

    //2. 缓存不存在
    if(!$userinfo){
        //1》 引导用户授权
        if(!I('get.code')){
            $oauth->oauthEntry($curr_url); //引导授权入口
        }
        $code = I('get.code'); //获取code

        //2》 通过 code 获取 access_token 和 openid
        $return = $oauth->getAccessToken($code);
        session('access_token',$return); //缓存access_token 以备（此步骤可以省略，但建议保留，可以且于排错）
        $access_token = $return['access_token']; //access_token
        $openid = $return['openid']; //openid

        //3》 通过access_token 和 openid 获取用户信息
        $userinfo = $oauth->getUserInfo($access_token,$openid);

        //3. 缓存用户信息
        session('openid',$userinfo);
        //将用户信息存储到数据库
        $this->writeUserInfo($userinfo);
    }
    //打印用户信息
    dump($userinfo);
}
```

```

<?php
namespace Home\Weixin;
use Think\Controller;
use Home\Weixin;
class OauthWeixin extends CommonWeixin {
    //授权入口
    public function oauthEntry($url){
        if($url==""){
            exit('重定向地址不正确');
        }
        //授权入口地址
        $api = "https://open.weixin.qq.com/connect/oauth2/authorize?appid=".C('WEIXIN_APPID')."&redirect_uri=".urlencode($url)."&response_type=code&scope=snsapi_userinfo&state=123456#wechat_redirect";
        //重定向到入口地址
        redirect($api);
    }

    //取得access_token 和 openid
    public function getAccessToken($code){
        $api = "https://api.weixin.qq.com/sns/oauth2/access_token?appid=".C('WEIXIN_APPID')."&secret=".C('WEIXIN_APPSECRET')."&code=".$code."&grant_type=authorization_code";
        $result = file_get_contents($api); //服务端返回结果 { "access_token":"ACCESS_TOKEN", "expires_in":7200, "refresh_token":"REFRESH_TOKEN", "openid":"OPENID", "scope":"SCOPE" }
        $sarr = json_decode($result,true); //转为数组
        if($sarr['errcode']!=0){
            exit("获取access_token出错: 错误码: ".$sarr['errcode']." 错误消息: ". $this->getErrMsg($sarr['errcode'])); //输出错误信息
        }
        return $sarr; //返回access_token信息数组
    }

    /**取得用户信息
     * |
     */
    public function getUserInfo($access_token,$openid){
        $api = "https://api.weixin.qq.com/sns/userinfo?access_token=".$access_token."&openid=".$openid."&lang=zh_CN"; //接口地址
        $result = file_get_contents($api); //取得用户信息
        $sarr = json_decode($result,true); //转为数组
        if($sarr['errcode']!=0){
            exit("获取用户信息出错: 错误码: ".$sarr['errcode']." 错误消息: ". $this->getErrMsg($sarr['errcode'])); //输出错误信息
        }
        return $sarr; //返回用户信息
    }
}

//将用户信息写入用户信息表
public function writeUserInfo($userinfo){
    //创建用户信息表
    /*
    CREATE TABLE `demo`.`<table_name>` (
      `id` int(10) NOT NULL AUTO_INCREMENT,
      `openid` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '用户openid',
      `nickname` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '用户昵称',
      `sex` tinyint(1) DEFAULT 0 COMMENT '用户的性别, 值为1时是男性, 值为2时是女性, 值为0时是未知',
      `city` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '城市',
      `province` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '省份',
      `country` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '国家',
      `headimgurl` varchar(250) CHARACTER SET utf8 DEFAULT '' COMMENT '用户头像, 最后一个数值代表正方形头像大小 (有0、46、64、96、132数值可选, 0代表640*640正方形头像) , 用户没有头像时该项为空。若用户更换头像, 原有头像URL将失效。',
      `create_time` int(11) DEFAULT 0 COMMENT '更新时间',
      PRIMARY KEY (`id`)
    ) COMMENT='用户信息表';
    */

    $m = M('WeixinUserInfo'); //实例化

    $rs = $m->where(array('openid'=>$userinfo['openid']))->find();
    $userinfo['create_time'] = time();
    if(!$rs){
        //写入用户信息
        $m->add($userinfo);
    }else{
        $m->where(array('id'=>$rs['id']))->save($userinfo);
    }
    return true;
}

```

3. 公众平台开发-JS-SDK

3-1 微信JS-SDK介绍

微信JS-SDK是微信公众平台面向网页开发者提供的基于微信内的网页开发工具包。

通过使用微信JS-SDK, 网页开发者可借助微信高效地使用

1. 拍照、选图、语音、位置等手机系统的能力,
2. 微信分享
3. 扫一扫
4. 卡券

5. 支付

使用步骤：

步骤一：绑定域名

先登录微信公众平台进入“公众号设置”的“功能设置”里填写“JS接口安全域名”。

备注：登录后可在“开发者中心”查看对应的接口权限。

步骤二：引入JS文件

在需要调用JS接口的页面引入如下JS文件，（支持https）：<http://res.wx.qq.com/open/js/jweixin-1.2.0.js>

备注：支持使用 AMD/CMD 标准模块加载方法加载

步骤三：通过config接口注入权限验证配置

所有需要使用JS-SDK的页面必须先注入配置信息，否则将无法调用

```
wx.config({
  debug:                                true,                                //
  // 开启调试模式,调用的所有api的返回值会在客户端alert出来,若要查看传入的参数,可以在pc端打开,
  // 参数信息会通过log打出,仅在pc端时才会打印。
  appId: "", // 必填, 公众号的唯一标识
  timestamp: , // 必填, 生成签名的时间戳
  nonceStr: "", // 必填, 生成签名的随机串
  signature: "", // 必填, 签名, 见附录1
  jsApiList: [] // 必填, 需要使用的JS接口列表, 所有JS接口列表见附录2
});
```

步骤四：通过ready接口处理成功验证

```
wx.ready(function(){
  //
  // config信息验证后会执行ready方法,所有接口调用都必须在config接口获得结果之后,config是一个客户端的异步操作,所以如果需要在页面加载时就调用相关接口,则须把相关接口放在ready函数中调用来确保正确执行。对于用户触发时才调用的接口,则可以直接调用,不需要放在ready函数中。
});
```

步骤五：通过error接口处理失败验证

```
wx.error(function(res){
```

```
//
```

config信息验证失败会执行error函数，如签名过期导致验证失败，具体错误信息可以打开config的debug模式查看，也可以在返回的res参数中查看，对于SPA可以在这里更新签名。

```
});
```

3-2通过JS-SDK实现分享接口

3-2-1使用模板

JS模板

```
<script type="text/javascript">
wx.config({
  debug: true, // 开启调试模式,调用的所有api的返回值会在客户端alert出来,若要查看传入的参数,可以在pc端打开,参数信息会通过log打出,仅在pc端时才会打印。
  appId: '{C("WEIXIN_APPID")}', // 必填,公众号的唯一标识
  timestamp: {$timestamp}, // 必填,生成签名的时间戳
  nonceStr: '{$noncestr}', // 必填,生成签名的随机串
  signature: '{$signature}', // 必填,签名,见附录1
  jsApiList: ['onMenuShareTimeline','onMenuShareAppMessage','onMenuShareQQ','onMenuShareWeibo','onMenuShareQZone','startRecord','stopRecord','onVoiceRecordEnd','playVoice','pauseVoice','stopVoice','onVoicePlayEnd','uploadVoice','downloadVoice','chooseImage','previewImage','uploadImage','downloadImage','translateVoice','getNetworkType','openLocation','getLocation','hideOptionsMenu','showOptionsMenu','hideMenuItems','showMenuItems','hideAllNonBaseMenuItem','showAllNonBaseMenuItem','closeWindow','scanQRCode','chooseWXPay','openProductSpecificView','addCard','chooseCard','openCard'] // 必填,需要使用的JS接口列表,所有JS接口列表见附录2
});

//验证成功回调
wx.ready(function(){
  //验证接口
  $('#checkJsApi').on('click',function(){
    wx.checkJsApi({
      jsApiList: ['onMenuShareTimeline'], // 需要检测的JS接口列表,所有JS接口列表见附录2,
      success: function(res) {
        // 以键值对的形式返回,可用的api值true,不可用为false
        // 如: {"checkResult":{"chooseImage":true},"errMsg":"checkJsApi:ok"}
      }
    });
  });
  //分享到朋友圈
  wx.onMenuShareTimeline({});
  //分享给朋友
  wx.onMenuShareAppMessage({});
  //分享到QQ
  wx.onMenuShareQQ({});
  //其它接口
});

//难失败回调
wx.error(function(res){
  // config信息验证失败会执行error函数,如签名过期导致验证失败,具体错误信息可以打开config的debug模式查看,也可以在返回的res参数中查看,对于SPA可以在这里更新签名。
});
</script>
```

PHP后端验证参数实现

JS模板中,我们需要通过PHP后台提供的参数如下:

appId 可以直接到公众平台获取
timestamp 直接time()函数获取
nonceStr 通过自定义函数 getNoncestr() 生成
signature 通过算法生成

```

/**
 * 取得随机字符串
 * @return string 返回16位随机字符串
 */
public function getNoncestr(){
    $str1 = range('a','z'); //a-z数组
    $str2 = range('A','Z'); //A-Z数组
    $str3 = range(0,9);      //0-9数组
    $arr  = array_merge($str1,$str2,$str3); //合并数组
    $return = "";           //申明16位随机字符串
    //循环一个16位的随机字符串，包括字母大小，数字组合
    for($i=0;$i<16;$i++){
        $return .= $arr[array_rand($arr)];
    }
    return $return;
}

```

取得 signature 签名：

算法：

1. 将以下参数
noncestr(随机字符串),jsapi_ticket(jsapi票据),timestamp(时间戳), url(当前网页的URL, 不含#及后面部份) 参数名ASCII排序
2. 使用URL键值对拼接如：(即key1=value1&key2=value2...)
3. 使用 sha1加密

关键代码及函数


```

/**
 * 取得网页JsSDK 签名
 * @param string $noncestr 16位随机字符串
 * @param number $timestamp 时间戳
 * @return string 签名
 */
public function getJsdkSignature($noncestr,$timestamp){
    /**
     * 算法: 1. 将以下参数 noncestr(随机字符串),jsapi_ticket(jsapi票据),timestamp(时间戳), url(当前网页的URL, 不含#及后面部份)
     * 参数名ASCII排序
     * 2. 使用URL键值对拼接如: (即key1=value1&key2=value2...)
     * 3. 使用 sha1加密
     */
    //1. 排序
    $arr_tmp = array();
    $arr_tmp['noncestr'] = $noncestr; //取得随机字符串
    $arr_tmp['jsapi_ticket'] = $this->getJsapiTicket(); //取得jsapi_ticket 票据
    $arr_tmp['timestamp'] = $timestamp; //时间戳
    $arr_tmp['url'] = $this->currentUrl(); //当前网页URL
    ksort($arr_tmp); //排序

    //2.使用URL键值对拼接
    $str = "";
    foreach ($arr_tmp as $key => $value) {
        $str .= "&.$key.=".$value;
    }

    $str = trim($str, '&');

    //3. sha1 加密
    $signature = sha1($str);
    return $signature;
}

```

```

/**
 * 取得当前页面的URL地址, 不包括#及#后面部份
 * @return [type] [description]
 */
public function currentUrl(){
    $server = I('server.');
    //dump($server);
    $scheme = $server['REQUEST_SCHEME']; //协议
    $host = $server['HTTP_HOST']; //主机
    $request_uri = $server['REQUEST_URI']; //请求uri

    $url = $scheme."://".$host.$request_uri;
    return $url;
}

```

```

/**
 * 取得 jsapi_ticket
 * @return string 返回jsapi_sticket 值
 */
public function getJsapiTicket(){
    //缓存初始化
    S(array('type'=>'file','prefix'=>'dodiapp_', 'expire'=>3600));

    $jsapi_ticket = S('jsapi_ticket'); //查看缓存是否存在

    if(!$jsapi_ticket){
        //取得access_token
        $access_token = $this->getToken();

        $api = "https://api.weixin.qq.com/cgi-bin/ticket/getticket?access_token=".$access_token."&type=jsapi"; //
        jsapi_ticket接口地址

        //取得ticket 数据
        $sarr = $this->getRequest($api);
        //如果获取出错, 将错误信息写到日志中
        if($sarr['errcode']!=0){
            $this->writeLog($sarr['errcode'], "获取jsapi_ticket出错:错误详情(".$this->getErrMsg($sarr['errcode']).")");
        }
        //jsapi_ticket 值
        $jsapi_ticket = $sarr['ticket'];
        //缓存 jsapi_ticket
        S('jsapi_ticket', $jsapi_ticket);
    }
    return $jsapi_ticket; //返回jsapi_ticket
}

```

3-2-2分享接口实现

注意事项:

请注意不要有诱导分享等违规行为, 对于诱导分享行为将永久回收公众号接口权限

<http://kf.qq.com/faq/131117ne2MV7141117JzI32q.html>

分享我们讲解, 分享到微信朋友圈, 和分享给朋友两个分享接口:

1. 获取“分享到朋友圈”按钮点击状态及自定义分享内容接口

```

wx.onMenuShareTimeline({
    title: "", // 分享标题
    link: "", // 分享链接, 该链接域名或路径必须与当前页面对应的公众号JS安全域名一致
    imgUrl: "", // 分享图标
    success: function () {
        // 用户确认分享后执行的回调函数
    },
    cancel: function () {
        // 用户取消分享后执行的回调函数
    }
});

```

2. 获取“分享给朋友”按钮点击状态及自定义分享内容接口

```
wx.onMenuShareAppMessage({
  title: ", // 分享标题
  desc: ", // 分享描述
  link: ", // 分享链接, 该链接域名或路径必须与当前页面对应的公众号JS安全域名一致
  imgUrl: ", // 分享图标
  type: ", // 分享类型,music、video或link, 不填默认为link
  dataUrl: ", // 如果type是music或video, 则要提供数据链接, 默认为空
  success: function () {
    // 用户确认分享后执行的回调函数
  },
  cancel: function () {
    // 用户取消分享后执行的回调函数
  }
});
```

3-2-3 图像接口

步骤

1. 拍照或从手机相册选图

```
wx.chooseImage({
  count: 1, // 默认9
  sizeType: ['original', 'compressed'], // 可以指定是原图还是压缩图, 默认二者都有
  sourceType: ['album', 'camera'], // 可以指定来源是相册还是相机, 默认二者都有
  success: function (res) {
    var localIds = res.localIds; //
    // 返回选定照片的本地ID列表, localId可以作为img标签的src属性显示图片
  }
});
```

2. 预览图片

```
wx.previewImage({
  current: ", // 当前显示图片的http链接
  urls: [] // 需要预览的图片http链接列表
});
```

3. 上传图片(不常用)

```
wx.uploadImage({
  localId: ", // 需要上传的圖片的本地ID, 由chooseImage接口获得
```

```

        isShowProgressTips: 1, // 默认为1, 显示进度提示
        success: function (res) {
            var serverId = res.serverId; // 返回图片的服务器端ID
        }
    });

```

4. 下载图片（不常用）

```

wx.downloadImage({
    serverId: "", // 需要下载的图片的服务器端ID, 由uploadImage接口获得
    isShowProgressTips: 1, // 默认为1, 显示进度提示
    success: function (res) {
        var localId = res.localId; // 返回图片下载后的本地ID
    }
});

```

3-2-4 微信扫一扫接口

调起微信扫一扫接口

```

wx.scanQRCode({
    needResult: 0, // 默认为0, 扫描结果由微信处理, 1则直接返回扫描结果,
    scanType: ["qrCode", "barCode"], // 可以指定扫二维码还是一维码, 默认二者都有
    success: function (res) {
        var result = res.resultStr; // 当needResult 为 1 时, 扫码返回的结果
    }
});

```

四、小节

2. 课堂练习

3. 课后练习

4. 资料扩展