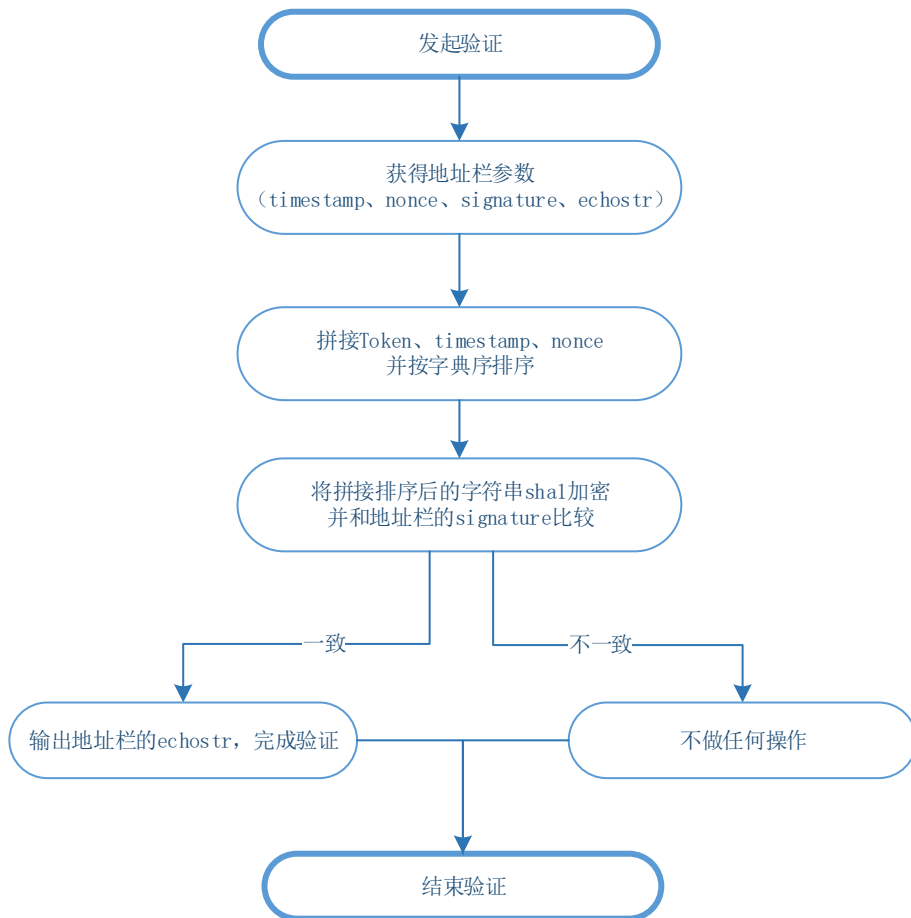


验证服务器配置流程



// 验证服务器配置（公众平台修改服务器配置时验证用）。返回true/false
weixin.Valid_Server(req);

开发者中心

配置项

接口报警

服务器配置(已启用)

修改配置

停用

停用服务器配置后，用户消息和开发者需要的事件推送，将不会被转发到该URL中

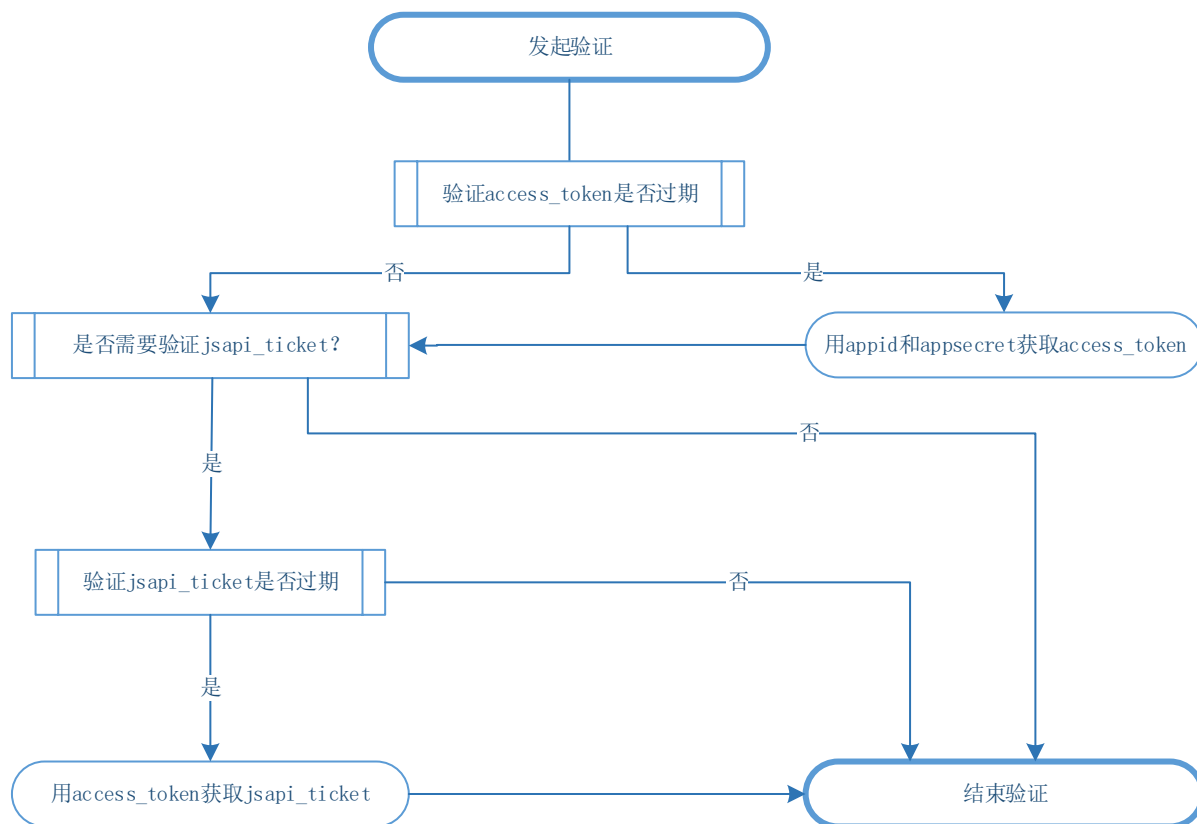
URL(服务器地址) t.aspx

Token(令牌) 4

EncodingAESKey(消息加解密密钥 ) t

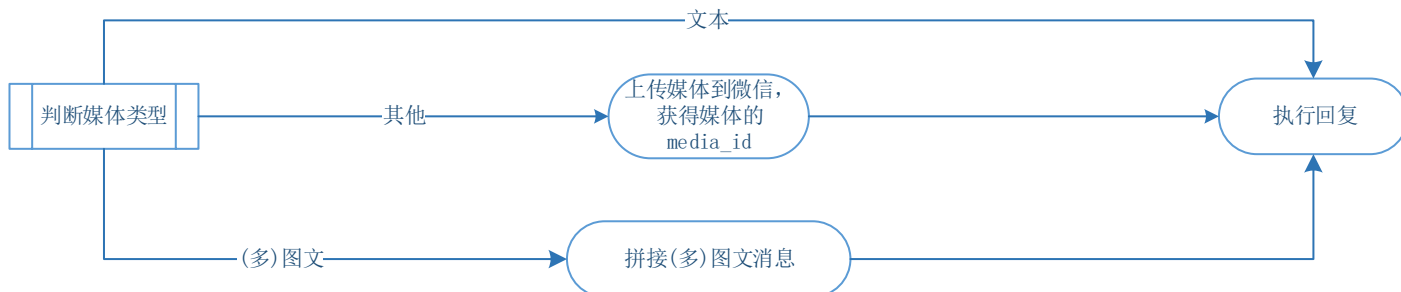
消息加解密方式 安全模式

access_token及jsapi_ticket验证流程



```
// 【异步】验证并获得 access_token和jsapi_ticket
/*
    need_jsapi_ticket: true/false 需要验证并更新jsapi_ticket
    Callback_success(access_token, jsapi_ticket): 成功回调
*/
weixin.get_accessToken_jsapiTicket(req, need_jsapi_ticket, Callback_success);
```

被动回复用户消息



```
// 【异步】上传媒体到微信
/*
    返回media_id, 3天有效
    opt = {
        filename: 文件路径
        type: 媒体类型 image(图片, 1M, 支持JPG格式) voice(语音, 2M, 播放长度不超过
60s, 支持AMR\MP3格式) video(视频, 10MB, 支持MP4格式) thumb(缩略图, 64KB, 支持JPG格式)
        again: 视频是否需要二次上传(高级群发需要, 被动回复不需要) true-需要 else-不需
要
        Callback_success(media_id): 成功回调
    }
*/
weixin.upload_file_temp(req, opt);

// 【同步】拼接被动回复(多)图文消息
/*
    高京
    2016-04-27
    返回拼接好的消息字符串, 直接输出即可
    opt = {
        Title: 标题
        Description: 描述
        Picurl: 图片地址(绝对路径)
        Url: 跳转地址
    }
*/
weixin.appendReply_Article(opt);

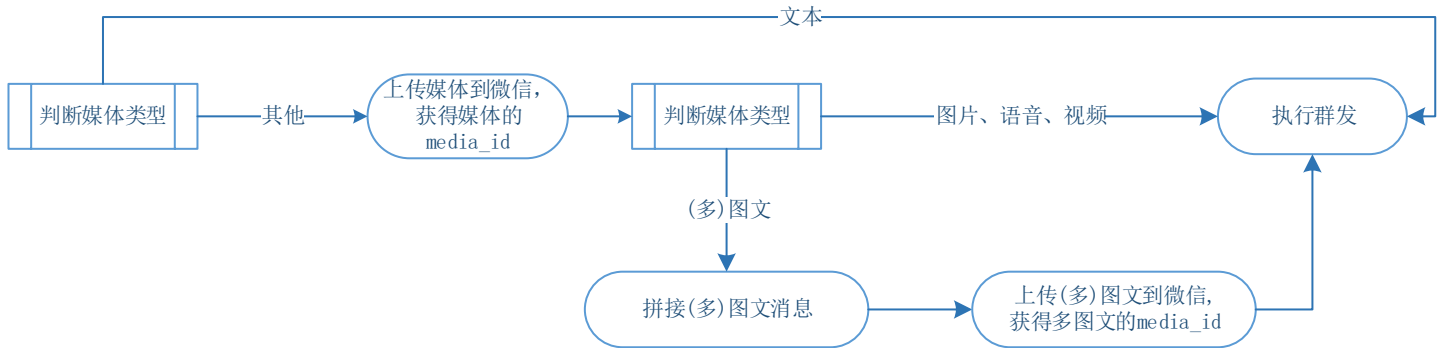
// 【同步】被动回复消息
/*
    返回字符串, 输出即可
    opt = {
        xml_json: 微信发来的xml转json对象
        Reply: 回复内容或media_id或(多)图文拼接消息
        Kind: 回复种类 1-文字 2-图片 3-语音 4-视频 5-图文
        video_title: 视频标题。默认值: ""
        video_description: 视频描述。默认值: ""
        Article_count: (多)图文消息数量。默认值: 0
    }
*/
weixin.reply_msg(opt);
```

素材

```
// 【异步】新增临时素材
/*
    返回media_id, 3天有效
    opt = {
        filename: 文件路径
        type: 媒体类型 image(图片, 1M, 支持JPG格式) voice(语音, 2M, 播放长度不超过60s, 支持AMR\MP3格式) video(视
频, 10MB, 支持MP4格式) thumb(缩略图, 64KB, 支持JPG格式)
        again: 视频是否需要二次上传(高级群发需要, 被动回复不需要) true-需要 else-不需要
        Callback_success(media_id): 成功回调
    }
*/
weixin.upload_file_temp(req, opt);

// 【异步】下载临时素材
/*
    opt = {
        media_id: 临时素材的media_id,
        save_dir: 保存文件夹(如文件夹不存在则自动创建, 但不支持多级文件夹均不存在的情况), 不包含文件名。默认
"/"。
        ext: 文件后缀, 不含".", 默认jpg
        Callback_success(file): 成功回调
    }
    * file = {
        dir: "/UploadFile/", // 最终保存的文件夹, "/" 开头
        filename: "201608101530503724.jpg", // 文件名, 不含路径, 格式: yyyyMMddHHmmssRRRR.ext
    }
*/
weixin.download_file_temp(req, opt);
```

高级群发和预览



需要文件: /Bin/wx_class.dll

```
/// <summary>
/// 上传媒体到微信, 获得media_id, 3天有效
/// </summary>
/// <param name="access_token">有效的access_token</param>
/// <param name="filename">Server.MapPath(文件绝对路径)</param>
/// <param name="type">媒体类型: image(图片, 1M, 支持JPG格式)
voice(语音, 2M, 播放长度不超过60s, 支持AMR\MP3格式) video(视频, 10MB, 支持MP4格式)
thumb(缩略图, 64KB, 支持JPG格式)</param>
/// <param name="again">视频是否需要二次上传(高级群发需要, 被动回复不需要)</param>
string wx_class.mp_event.upload_file(string access_token, string filename, string type,
bool again)

/// <summary>
/// 拼接高级群发(多)图文消息
/// </summary>
/// <param name="media_id">图文消息缩略图的media_id, 可以在基础支持-
上传多媒体文件接口中获得</param>
/// <param name="author">图文消息的作者</param>
/// <param name="title">图文消息的标题</param>
/// <param name="content_source_url">在图文消息页面点击“阅读原文”后的页面地址</param>
/// <param name="content">图文消息页面的内容, 支持HTML标签</param>
/// <param name="digest">单图文消息的描述</param>
/// <param name="show_cover_pic">是否显示封面, 1为显示, 0为不显示</param>
string wx_class.mp_event.appendArticle(string media_id, string author, string title,
string content_source_url, string content, string digest, string show_cover_pic)

/// <summary>
/// 上传(多)图文到微信, 获得media_id, 3天有效
/// </summary>
/// <param name="access_token">有效的access_token</param>
/// <param name="para">拼接好的图文消息素材参数</param>
string wx_class.mp_event.upload_article(string access_token, string para)

/// <summary>
/// 高级群发和预览。返回MsgID
/// </summary>
/// <param name="appId">公众平台的appId</param>
/// <param name="appSecret">公众平台的appSecret</param>
/// <param name="access_Token">有效的access_Token</param>
/// <param name="Preview">预览</param>
/// <param name="Kind">群发种类 1-文字 2-图片 3-语音 4-视频 5-图文</param>
/// <param
name="Content">群发内容(可以是文字; 可以是图片、语音或视频的media_id; 可以是图文素材对应的Aid</
param>
/// <param
name="is_to_all">是否发送给所有关注用户, 预览时无作用。为true时, OpenID和groupId无作用</param>
/// <param
name="OpenID">接收者OpenID。预览时, 只能且必须发给一个OpenID; OpenID.Length>0时, 发送给指定OpenI
D, groupId无作用</param>
/// <param name="groupId">接收组编号</param>
/// <param name="vidio_thumb">视频缩略图的media_id</param>
string wx_class.mp_event.batch(string appId, string appSecret, string access_Token,
bool Preview, int Kind, string Content, bool is_to_all, string[] OpenID, long groupId)
```

用户管理

// 【异步】获取用户列表，每次最多10000条

/*

```
    返回json: {
        total: 2,
        count: 2,
        data: {
            openid: [ 'o_QMyuBnU1MEx_5ZmYGeR40vcjE',
                      'o_QMyuBvy9_x58-9zAJ002q2MVWk' ],
        },
        next_openid: 'o_QMyuBvy9_x58-9zAJ002q2MVWk'
    }
```

* next_openid: 如未拉取完，则为下一个有效OpenID，下次拉取可用此值作为OpenID_start;
如已拉取完，则为最后一个有效OpenID

OpenID_start: 开始拉取的OpenID。可为null

Callback_success: 成功回调

*/

```
weixin.getOpenID(req, Callback_success, OpenID_start);
```

// 【异步】根据OpenID获取用户信息

/*

```
    返回Json: {
        "subscribe": 1,
        "openid": "o6_bmjrrPTlm6_2sgVt7hMZOPfL2M",
        "nickname": "Band",
        "sex": 1,
        "language": "zh_CN",
        "city": "广州",
        "province": "广东",
        "country": "中国",
        "headimgurl":
        "http://wx.qlogo.cn/mmopen/g3MonUZtNHkdmzicIlibx6iaFqAc56vxLSUfpb6n5WKSYVY0ChQKkiaJSgQ1dZuTogvLLrhJbERQQ4eMsv84eavHiaiceqxibJxCfHe/0",
        "subscribe_time": 1382694957,
        "unionid": "o6_bmasdasdsad6_2sgVt7hMZOPfL"
        "remark": "",
        "groupid": 0
    }
```

* subscribe == 0 时，不会有其他信息被获取

OpenID: OpenID

Callback_success: 成功回调

*/

```
weixin.getUserInfo(req, OpenID, Callback_success);
```

/// <summary>

/// 创建用户分组。返回分组id，创建失败返回错误信息

/// </summary>

/// <param name="access_token">有效的access_token</param>

/// <param name="group_name">分组名称</param>

string wx_class.users.group_create(string access_token, string group_name)

/// <summary>

/// 修改用户分组。返回处理结果

/// <summary>

/// 查询用户所在分组。返回groupid

/// </summary>

/// <param name="access_token">有效的access_token</param>

/// <param name="OpenID">用户的OpenID</param>

/// <param name="group_name">分组名称</param>

/// <summary>

/// 移动用户分组。返回处理结果

/// </summary>

/// <param name="access_token">有效的access_token</param>

/// <param name="OpenID">用户的OpenID。</param>

/// <param name="group_id">新分组id</param>

string wx_class.users.user_moveGroup(string access_token, string[] OpenID, string

group id)

发布3×5菜单

```
// 【同步】 插入主菜单
/*
    返回菜单字符串
    opt:{
        type: 类型[click/view等, 具体见官方文档],
        name: 菜单标题, 含有子菜单时无意义
        key: 菜单值/链接地址, 含有子菜单时无意义
        have_childs: true-含有子菜单, else-不含
    }
*/
weixin.insert_Menu_parent(opt);

// 【同步】 插入子菜单
/*
    返回菜单字符串
    opt:{
        type: 类型[click/view等, 具体见官方文档],
        name: 菜单标题,
        key: 菜单值/链接地址
    }
*/
weixin.insert_Menu_child(opt);

// 【同步】 关合含有子菜单的主菜单
/*
    返回菜单字符串
*/
weixin.insert_Menu_parent_close();

// 【异步】 发布菜单
/*
    menu_str: 菜单字符串
    Callback_success(result): 成功回调
*/
weixin.publish_menu(req, menu_str, Callback_success);

// 【e.x】
var menu_str = "";
menu_str += weixin.insert_Menu_parent({ name: "我要", have_childs: true });
menu_str += weixin.insert_Menu_child({ type: "view", name: "新手指南", key: "http://" +
req.hostname + "/t/Task_detailed.aspx?Novice=1&Tid=10060" });
menu_str += weixin.insert_Menu_parent_close();
menu_str += weixin.insert_Menu_parent({ type: "view", name: "官网", key: "http://
www.topu.net" });

var Callback_success = function(json) {
    res.send(JSON.stringify(json));
};

weixin.publish_menu(req, menu_str, Callback_success);
```

微信红包

需要提供：

1. 微信支付API证书文件：登录pay.weixin.qq.com后下载
2. 微信支付API证书密码：默认为客户的微信支付商户号
3. 微信支付API证书密钥：登录pay.weixin.qq.com后设置

```
// 【异步】 发送红包
/*
    * opt: {
        mch_id: 微信商户号。默认根据二级域名获得,
        send_name: 红包发送者名称,
        re_openid: 接受红包的用户OpenID,
        total_amount: 付款金额, 单位分, 最小100,
        total_num: 红包发放总人数,
        wishing: 红包祝福语,
        act_name: 活动名称,
        remark: 备注
    }
    * Callback_success(json): 成功回调
*/
weixin.send_cash_red(req, opt, Callback_success)


// 【异步】 查看红包状态
/*
    * opt = {
        mch_id: 微信商户号。默认根据二级域名获得,
        mch_billno: 待查红包的商家订单号
    }
    * Callback_success(json): 成功回调
*/
weixin.getStatus_cash_red(req, opt, Callback_success)


// 【异步】 企业付款
/*
    * opt: {
        mchid: 微信商户号。默认根据二级域名获得,
        openid: 收款用户OpenID,
        check_name: 校验用户姓名选项, 默认NO_CHECK,
        re_user_name: 收款用户姓名, check_name!=NO_CHECK时起作用,
        amount: 付款金额, 单位分, 最小100,
        desc: 描述
    }
    * Callback_success(json): 成功回调
*/
weixin.pay_transfers (req, opt, Callback_success)
```


用户网页授权

```
// 【异步】 通过授权code换取用户授权access_token和OpenID等
/*
    code: 授权成功后获得的code
    Callback_success(json): 成功回调
*/
weixin.get_accessToken_byCode(req, code, Callback_success);
```

模板消息

```
// 【异步】 发送模板消息
/*
    opt: {
        OpenID: 接收者OpenID,
        template_id: 模板ID
        link: 点击跳转链接地址,
        data_json: 模板消息, json格式, 根据模板设定
    }
    * data_json: {
        "keynote1": {
            "value": "巧克力",
            "color": "#173177"
        },
        "keynote2": {
            "value": "39.8元",
            "color": "#173177"
        }
    }

    Callback_success(json): 成功回调
*/
weixin.send_template_message(req, opt, Callback_success);
```

【e. x. 】

```
var Callback_success = function(json) {
    res.send(JSON.stringify(json));
};

var opt = {
    OpenID: OpenID,
    template_id: "iOMUm0dBjdzMvhyXzooVsLTk8wA78SRkk3rJd9-MX28",
    link: "http://www.topu.net",
    data_json: {
        title: {
            value: "这里是标题",
            color: "#173177"
        },
        content: {
            value: "这里是内容",
            color: "#ff0000"
        }
    }
};

weixin.send_template_message(req, opt, Callback_success);
```

生成二维码图片

// 【异步】 生成二维码图片。

/*

 * opt = {

 expire_seconds: 该二维码有效时间，以秒为单位。 最大不超过2592000（即30天），此字段如果不填，则默认有效期为30秒。,

 action_name: 二维码类型，QR_SCENE为临时, QR_LIMIT_SCENE为永久（scene_id参数传值）, QR_LIMIT_STR_SCENE为永久（scene_str参数传值），

 scene_v: 场景值，QR_SCENE为32位非0整型，QR_LIMIT_SCENE最大值为100000（目前参数只支持1--100000），QR_LIMIT_STR_SCENE为字符串，长度限制为1到64,

 }

 * callback(json): 成功回调

*/

`weixin.create_qrcode(req, opt, Callback_success)`

// 【异步】 用二维码ticket换取二维码图片并保存为服务器图片。

/*

 * opt = {

 ticket: ticket值,

 filepath: 文件完整路径，需确保文件目录存在

 }

 * callback(): 成功回调

*/

`weixin.get_qrcode_by_ticket(opt, Callback_success)`

```
// 【异步】 更新jsapi的wx_config相关数据
/*
    * opt: {
        jsApiList: 使用接口，字符串。默认: "[\"onMenuShareTimeline\", \"onMenuShareAppMessage\", \"chooseImage\", \"uploadImage\", \"hideAllNonBaseMenuItem\"]"
    }
    * Callback_success({
        appId: appId,
        timestamp: 时间戳 ,
        nonceStr: 随机数,
        signature: 签名,
        jsApiList: 接口列表,
    }): 成功回调
*/
weixin.get_jsapi_config(req, opt, Callback_success)
```