H5 一键登录（标准版）jssdk 集成文档

V1.02

版本历史

版本	更新内容	更新时间
V1.02	H5 一键登录（标准版）的 JSSDK 集成文档，包含 jssdk 和服务端接口说明，支持移动、联通和电信三网。 **注：H5 一键登录（标准版）即为互联网能力开放平台的一键登录标准版（H5），能力编码 59。**	2022-06-08

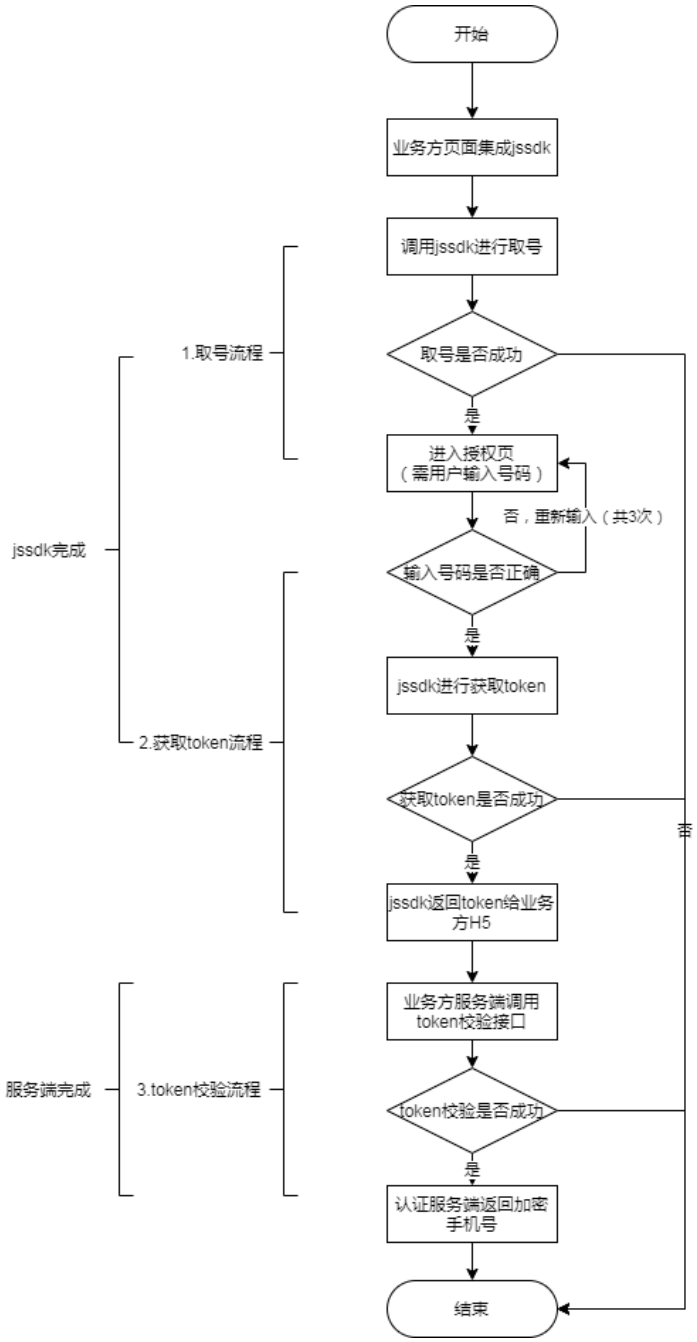
版权说明

目录

1. 业务及交互流程

1.1. 业务流程

1.1.1. 流程图



1.1.2. 流程说明

1）业务方页面集成 一键登录 H5 取号 jssdk；

2）业务方 H5 调用 jssdk 提供的方法获取 token，jssdk 完成取号流程和获取 token 流程

- jssdk 取号流程：

  jssdk 进行取号，若取号成功，则进入授权页，用户在授权页补充正确的号码授权后，jssdk 进行获取 token 流程；若取号失败则返回报错

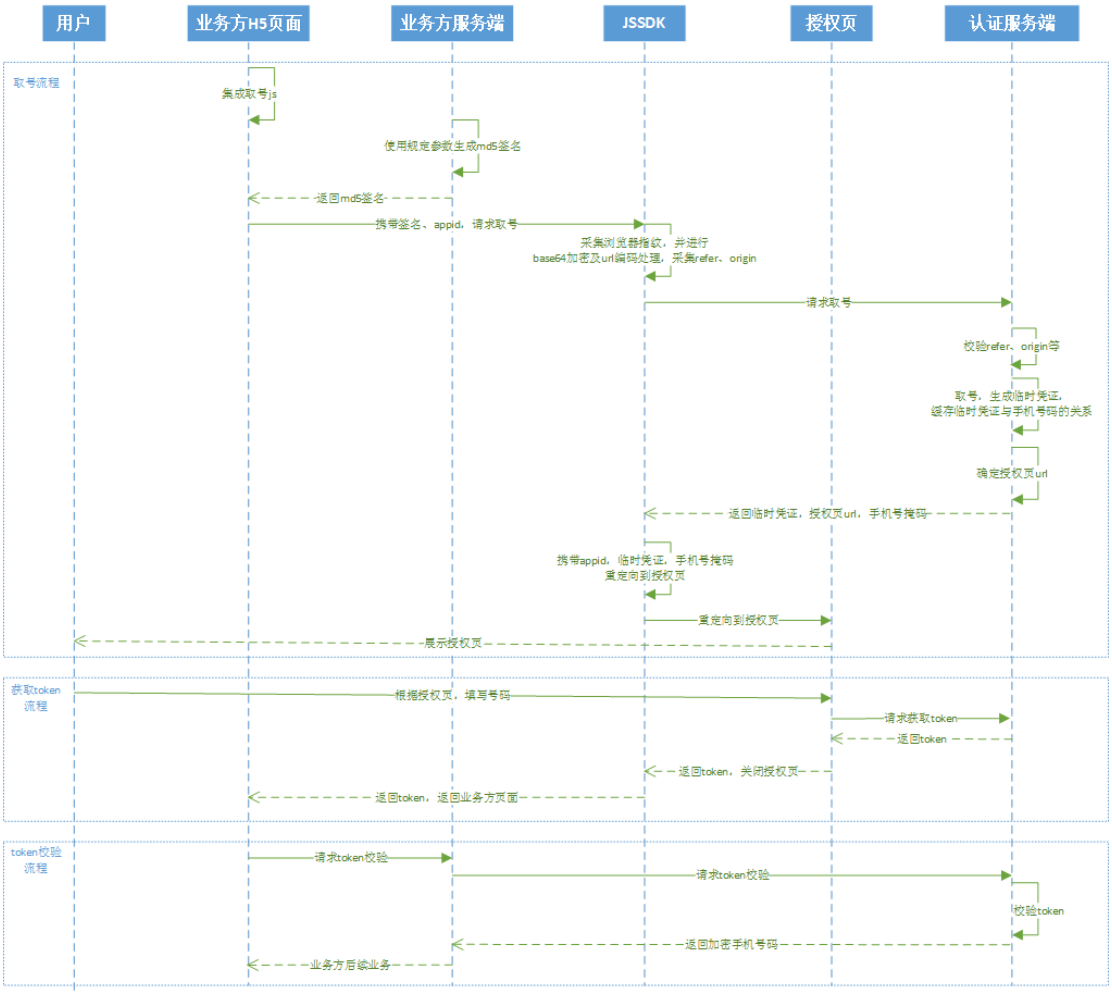
- jssdk 获取 token 流程：

  jssdk 进行获取 token，若获取 token 成功过，则返回 token 至业务方 H5；若 token 获取失败则返回报错

3）token 校验流程：业务方 H5 获取 token 后可传至业务方服务端，由业务方服务端调用 token 校验接口，若校验通过，则返回加密手机号；若失败则返回报错

1.2. 交互流程

1.2.1. 时序图



1.2.2. 流程说明

1）业务方 H5 集成取号 js，获取 md5 签名

2）业务方 H5 携带签名、appid，调用 jssdk 请求取号

3）jssdk 向认证服务端发起取号请求，取号成功后认证服务端返回临时凭证、授权页 url、手机号掩码，jssdk 重定向展示授权页面

4）用户授权并补齐正确的手机号码，授权页面发起请求获取 token，认证服务端返回 token 并返回业务方页面

5）业务方服务端发起 token 校验请求，认证服务端校验通过后返回手机号

2. 重要参数说明

2.1.集成页面地址（referer）

指集成了取号的页面访问地址，若多个页面集成用英文逗号分隔上报。必须报备可访问的完整页面地址，不允许使用通配符！

注：

1）假如业务方集成取号的页面访问地址为<http://www.abc.com/a/mobile.html/#/abc>，则报备地址为<http://www.abc.com/a/mobile.html/>

2） 假如业务方集成取号的页面访问地址为<http://www.abc.com/a/mobile.html/?abc>，则报备地址为<http://www.abc.com/a/mobile.html/>

3） 生产上有 https 访问的，https 请求跨域，会导致上报的 referer 为空，请在 head 中添加代码：<meta content="always" name="referrer">解决此问题。

2.2.请求来源（origin）

指发起请求的业务来源。一般为 protocol+host，不包含路径等信息。几种示例如下：

1）假如业务方集成取号的页面访问地址为<http://www.abc.com/a/mobile.html>，则 origin 为<http://www.abc.com>

2）假如业务方集成取号的页面访问地址为<https://www.abc.com/a/mobile.html>，则 origin 为<https://www.abc.com>

3）假如业务方集成取号的页面访问地址为<http://127.0.0.1:8080/a/mobile.html>，则 origin 为<http://127.0.0.1:8080>

4）假如业务方集成取号的页面访问地址为<https://127.0.0.1:8080/a/mobile.html>，则 origin 为 https://127.0.0.1:8080

2.3.authPageType

业务授权样式。

1）若值为“1”时展示弹窗，即标准弹窗样式，如需修改配置项需联系移动认证侧进行申请；

2）若值为“2”时展示自定义弹窗版，请参考“3.5 弹窗版自定义配置项”设置配置项；

3）若值为“3”时展示自定义页面版，请参考“3.6 页面版自定义配置项”设置配置项；

4）若为其它值则展示页面，即标准页面样式，如需修改配置项需联系移动认证侧进行申请。

3. jssdk 集成说明

3.1. 引用

3.1.1. 引用 crypto-js.js（支持本地引入）

注：crypto-js.js 请在官网下载。

3.1.2. 引用 jssdk.min.js

```
<script type="text/javascript" src=" https://www.cmpassport.com/h5/js/jssdk_auth/jssdk-1.0.0.min.js "></script>
```

3.1.3. 如果使用弹窗版本（authPageType 值为 1/2 的情况）需要引入 css 文件

```
<link rel="stylesheet" type="text/css"
href="https://www.cmpassport.com/h5/js/jssdk_auth/css/ydrz-layer.css">
```

3.2. 使用方法

```
YDRZAuthLogin.getTokenInfo({
  data: { //请求的参数
    version: '2.0', //接口版本号 （必填）
    appId: appId, //应用 Id （必填）
    sign: sign, //加密后的 sign（必填）。sign 生成规则： MD5(appId +
    businessType + msgId + timestamp + traceId + version+appkey)
    注：businessType 为 8；msgID 要和 traceID 保持一致，此处 sign 与 token 校验接口 sign 的生成规则不同
    traceId: traceId, //业务方生成唯一标识
```

```
    timestamp:timestamp, //请求消息发送的系统时间，精确到毫秒，共 17 位，格式：
20121227180001165
    openType: '1', //取号类型
    expandParams: '', //扩展参数 格式：参数名=值 多个时使用 \| 分割 （选填）
    authPageType: '0' , //若值为“1”时展示弹窗，若值为“2”时展示自定义弹窗版，若值为“3”
时展示自定义页面版，若为其它值则展示页面。更多说明见“2.重要参数说明”中“2.3.authPageType”
  },
  success: function(res) { //成功回调},
  error: function(res) { //错误回调},
  layerCallback:function(res) {//authPageType 等于 2 时可以通过该回调方法监听，用户输入中间四
位号码并勾选协议后触发 }
})
```

3.2.1. 成功回调

1）authPageType 不为 2

```
{code："103000"，message："token 获取成功",token:"",userInformation:'浏览器指纹'}
```

2）authPageType=2（无蒙层）时，YDRZAuthLogin.getTokenInfo 方法在弹窗渲染完成会回调弹窗加载成功

```
{"code":"103000","msgId":"HNNN2E3TH3S77LFUFUFTTP5C9J77GKER","message":"弹窗加载成功"}
```

3.2.2. 失败回调

1）取号失败

```
{
  msgId:'',
  YDData:{code:' 错误码', message:'移动取号失败描述，详见表格'},
  CTData:{ code:' 错误码', message:'电信取号失败描述，详见表格'},
  CUData:{ code:' 错误码', message:'联通取号失败描述，详见表格'}
}
```

2）获取 token 失败

```
{msgId:' ' , code:' 错误码' ,message:' 错误描述，详见表格' }
```

3.2.3. layerCallback 回调

```
{"code":"103000","msgId":"HNNN2E3TH3S77LFUFUFTTP5C9J77GKER","message":"用户已输入中间四位号码并勾选协议"}
```

3.3. 获取网络类型（可选）

注：此方法为可选，可以获取也可不获取。iOS 系统下由于兼容问题，netType 基本返回结果为 unknown 未知。

定义 connection 变量通过 YDRZAuthLogin.getConnection 方法获取当前手机网络状态

示例：

```
var connection = YDRZAuthLogin.getConnection(appid);
```

返回对象：

```
{
  appid:"appid",
  msgid:"唯一标识",
  netType:（cellular 数据流量、unknown 未知、wifi）3 种状态
}
```

可根据 netType 状态自行选择后续流程。

3.4. 结束获取 token

注：此方法当业务侧希望停止继续获取 token 时使用。调用此方法后，getTokenInfo 返回 503：获取 token 结束。

```
YDRZAuthLogin.endGetToken();
```

3.5. 弹窗版自定义配置项（authPageType 为 2）

3.5.1. 无蒙层

1）在您 HTML 代码需要引入弹窗的地方引入 div；

```
<html>
  <head><!-- 您的代码 --></head>
  <body>
```

```
    <!-- 您的代码 -->
    <div id="ydrzCustomControls"></div>
    <!-- 您的代码 -->
  </body>
</html>


<div id="ydrzCustomControls"></div>
```

2）在调用 YDRZAuthLogin.getTokenInfo 方法之前先初始化配置项；

```
var Options = {
  layerStyle: {
    width:"",
    height:"240px",
    bgColor:"#236",
    borderRadius:"23px"
  },
  phoneStyle:{
    fontSize:"",
    fontColor:"#F829FF",
    high:"",
    left:"20px",
  },
  agreeStyle: {
    fontSize:"",
    textalign:"",
    fontColor:"",
    hrefColor:"",
    high:"",
    left:"",
    agreeArr:[
      {name:"《中国移动服务协议》",url:"协议链接"}
    ],
  },
  closeBtnStyle:{
    ifShowBtn:true,
```

```
    btnImage:"",
    top:"",
    right:"",
    width:"",
    height:""
  },
  customControlStyle:{
    ifShow:true,
    width:"",
    height:"24px",
    high:"center",
    left:"center",
    bgColor:"#fff",
    border:"0",
    borderRadius:"",
    url:"控件链接",
    name:"其他登录方式",
    fontSize:"16px",
    fontColor:"#392211",
    textAlign:"center",
    textDecoration:''
  },
}
YDRZAuthLogin.CustomControlsInit("ydrzCustomControls",options);
```

2） 调用 YDRZAuthLogin.getTokenInfo 方法取号， 详见 3.2

3） 在用户完成输入之后调用 YDRZAuthLogin.authGetTokenByLayer 方法获取 token

```
YDRZAuthLogin.authGetTokenByLayer(
  success: function(res) { //成功回调},
  error: function(res) { //错误回调}
)
```

3.5.2. 有蒙层

1）在调用 YDRZAuthLogin.getTokenInfo 方法之前先初始化配置项；

```
var Options = {
  layerStyle: {
    width:"",
    height:"240px",
    bgColor:"#236",
    borderRadius:"23px"
  },
  maskStyle: {
    ifShowMask:true,
    bgColor:"",
    opacity:""
  },
  phoneStyle:{
    fontSize:"",
    fontColor:"#F829FF",
    high:"",
    left:"20px",
  },
  agreeStyle: {
    fontSize:"",
    textalign:"",
    fontColor:"",
    hrefColor:"",
    high:"",
    left:"",
    agreeArr:[
      {name:"《中国移动服务协议》",url:"协议链接"}
    ],
  },
  closeBtnStyle:{
    ifShowBtn:true,
    btnImage:"",
    top:"",
    right:"",
    width:"",
```

```
    height:""
  },
  customControlStyle:{
    ifShow:true,
    width:"",
    height:"24px",
    high:"center",
    left:"center",
    bgColor:"#fff",
    border:"0",
    borderRadius:"",
    url:"控件链接",
    name:"其他登录方式",
    fontSize:"16px",
    fontColor:"#392211",
    textAlign:"center",
    textDecoration:''
  },
}
YDRZAuthLogin.CustomControlsInit("ydrzCustomControls",options);
```

2）调用 YDRZAuthLogin.getTokenInfo 方法取号，详见 3.2

3.5.3. 配置项详细说明

配置项	字段	字段含义	值	说明
layerStyle	width	弹窗宽度	支持百分比或者数值,如“200px”	可选
	height	弹窗高度	支持百分比或者数值,如“200px”	可选
	bgColor	弹窗背景颜色	十六进制颜色码，如“#FFFFFF”	可选
	borderRadius	弹窗圆角	支持百分比或者数值,如“20px”	可选

titleStyle	ifShow	是否显示标题栏	Boolean 值，默认值为 FALSE	如果要展示标题栏，则必选
	name	标题内容	字符串	可选
	fontColor	字体颜色	十六进制颜色码，如“#FFFFFF”	可选
	fontSize	字体大小	支持数值，比如“20px”	可选
	left	距离弹窗左边框边距	支持百分比或者数值,如“200px”	可选
	high	距离弹窗上边框高度	支持百分比或者数值,如“20px”	可选
maskStyle	ifShowMask	是否显示蒙层	Boolean 值，默认值为 FALSE	如果要展示蒙层，则必选
	bgColor	蒙层颜色	十六进制颜色码，如“#FFFFFF”	可选
	opacity	透明度	从 0.0 （完全透明）到 1.0（完全不透明）	可选
phoneStyle	fontSize	字体大小	支持数值，比如“20px”	可选
	fontColor	字体颜色	十六进制颜色码，如“#FFFFFF”	可选
	high	距离弹窗上边框高度	支持百分比或者数值,如“20px”	可选
	left	距离弹窗左边框边距	支持百分比或者数值,如“200px”	可选

agreeStyle	fontSize	字体大小	支持数值，比如“20px”	可选
	textalign	文本对齐选项	“center/left/right”	可选
	fontColor	字体颜色	十六进制颜色码，如“#FFFFFF”	可选
	hrefColor	协议链接颜色	十六进制颜色码，如“#FFFFFF”	可选
	high	协议文案距离弹窗上边框高度	支持百分比或者数值,如“200px”	可选
	left	协议文案距离弹窗左边框边距	支持百分比或者数值,如“200px”	可选
	agreeArr	自定义协议数组	如：[{name:"协议名称",url:"协议链接"}]	可选
closeBtnStyle	ifShowBtn	是否展示关闭按钮	Boolean 值，默认值为 TRUE	可选
	btnImage	关闭按钮图片	Url 链接	可选
	top	关闭按钮距离弹窗上边框高度	支持百分比或者数值,如“12px”	可选
	right	关闭按钮距离	支持百分比或者数值,如“12px”	可选

		弹窗左边框边距		
	width	关闭按钮宽度	支持数值，比如“20px”	可选
	height	关闭按钮高度	支持数值，比如“20px”	可选
customControlStyle	ifShow	是否展示自定义控件	Boolean 值，默认值为 FALSE	如要展示自定义控件，则为必选
	width	自定义控件宽度	支持百分比或者数值,如“200px	必选
	height	自定义控件高度	支持百分比或者数值,如“200px	必选
	high	自定义控件距离弹窗上边框高度	支持百分比或者数值,如“200px	必选
	left	自定义控件距离弹窗左边框边距	支持百分比或者数值,如“20px	必选
	bgColor	自定义控件背景颜色	十六进制颜色码，如“#FFFFFF”	必选
	border	自定义控件边框	可以设置边框粗细，样式,颜色,如”1px solid #FFFFF”	必选

	borderRadius	自定义控件圆角	支持百分比或者数值,如“20px”	必选
	url	自定义控件跳转URL	Url链接	必选
	name	自定义控件显示文案	字符串	必选
	fontSize	自定义控件字体大小	支持数值，比如“20px”	必选
	fontColor	自定义控件字体颜色	十六进制颜色码，如“#FFFFFF”	必选
	textAlign	自定义控件文本对齐选项	“center/left/right”	必选
	textDecoration	自定义控件下划线	“none/underline”	必选
submitBtnStyle	ifShow	是否展示登录按钮	Boolean值，默认值为FALSE,	如要展示登录按钮，则为必选
	name	确认按钮显示文案	字符串	可选
	fontColor	确认按钮字体颜色	十六进制颜色码，如“#FFFFFF”	可选
	fontSize	确认按	支持数值，比如“20px”	可选

		钮字体大小		
	textAlign	确认按钮文本对齐选项	“center/left/right”	可选
	bgColor	确认按钮背景颜色	十六进制颜色码，如“#FFFFFF”	可选
	width	确认按钮宽度	支持百分比或者数值,如“200px”	可选
	height	确认按钮高度	支持者数值，如“200px”	可选
	borderRadius	确认按钮圆角	支持数值，比如“20px”	可选
	high	确认按钮距离弹窗上边框高度	支持百分比或者数值,如“200px”	可选
	left	确认按钮距离弹窗左边框高度	支持百分比或者数值,如“200px”	可选
errTipStyle	high	距离页面边框高度	支持“center“，百分比或者数值，如“20px”,	可选
	left	距离页面左边框边距	支持“center“，百分比或者数值，如“200px”	可选

3.6.页面版自定义配置项（authPageType 为 3）

3.6.1. 使用方式

1）在调用 YDRZAuthLogin.getTokenInfo 方法之前先通过 YDRZAuthLogin.authPageInit 方法初始化配置项；

```
var Options = {
"bgColor":"#FFFFFF",
"titleStyle":{"name":"本机号码登录","fontFamily":"PingFangSC-Medium, PingFang
SC","fontSize":"1.33rem","fontColor":"#444444","width":"70%","height":"1.83rem","le
ft":"center","high":"1rem","textAlign":"center"},
"logoStyle":{"url":"https://www.cmpassport.com/h5/js/jssdk_auth/image/logo.png","wi
dth":"6.96rem","height":"7.32rem","high":"7.9rem","left":"center"},
"authTextStyle":{"fontFamily":"PingFangSC-Medium, PingFang
SC","fontSize":"1.08rem","fontColor":"#444444","appNameColor":"#444444","width":"10
0%","textAlign":"center","high":"22.75rem","left":"center","fontWeight":"500"},
"phoneNumStyle":{"fontFamily":"PingFangSC-Semibold, PingFang
SC","fontSize":"2.08rem","fontColor":"#444444","bgColor":"#FFFFFF","fontWeight":"60
0","width":"15.42rem","left":"center","high":"19.58rem","inputStyle":{"width":"1.83
rem","height":"2.17rem"}},
"agreeStyle":{"fontFamily":"PingFangSC-Regular, PingFang
SC","fontSize":"1rem","fontColor":"#999999","high":"30.58rem","left":"center","chec
kedButton":{"width":"1.33rem","height":"1.33rem","uncheckColor":"#cccccc","checkedC
olor":"#1E82EB","uncheckUrl":"","checkedUrl":""},"hrefStyle":{"fontColor":"#1E82EB"
,"agreeArr":[]}},"tipStyle":{"fontFamily":"PingFangSC-Regular, PingFang
SC","fontSize":"0.92rem","fontColor":"#999999","high":"27rem","left":"center"},
"returnBtnStyle":{"width":"0.65rem","height":"1.1rem","left":"1rem","high":"1rem","
url":"https://www.cmpassport.com/h5/js/jssdk_auth/image/returnIcon.png"},
"customControlStyle":{"ifShow":"ture","width":"120px","height":"24px","high":"450px
","left":"center","bgColor":"#fff","border":"0","borderRadius":"","url":"https://ww
w.baidu.com","name":"其他登录方式
","fontSize":"16px","fontColor":"#392211","textAlign":"center","textDecoration":""}
}
YDRZAuthLogin.authPageInit(options);
```

数值支持传入 rem，根元素<html>的 fontsize 值为 12px；

2）调用 YDRZAuthLogin.getTokenInfo 方法取号，authPageType 值为“3”，详见 3.2

3.6.2. 配置项详细说明

	配置项	字段	字段含义	值	说明
背景	bgColor		背景颜色	十六进制颜色码，如“#FFFFFF”	可选
页面标题	titleStyle	name	页面标题的文案，默认“本机号码登录”	string	可选，标题支持为空，为空传“”
		fontFamily	文案的字体	string	可选
		fontSize	文案的字体大小	支持数值，如“200px”	可选
		fontColor	文案的字体颜色	十六进制颜色码，如“#FFFFFF”	可选
		high	文案距离页面上边	支持数值，如“20px”	可选

			框高度		
		left	文案距离页面左边框距离	支持数值； 居中可传入“center”	可选
		width	标题宽度	支持数值；	可选
		height	标题高度	支持数值；	可选
		textAlign	标题文案位置	left/right/center	可选
Logo	logoStyle	url	logo链接		可选
		high	logo距离页面上边框高度	支持数值，如“20px”	可选
		left	logo距离页面左边框距离	支持数值，如“20px”	可选
授权栏	authTextStyle	fontFamily	文案的字体	string	可选

		fontSize	文案的字体大小	支持数值，如“200px”	可选
		fontColor	文案的字体颜色	十六进制颜色码，如“#FFFFFF”	可选
		appNameColor	APP名称文案的字体颜色	十六进制颜色码，如“#FFFFFF”	可选
		high	文案距离页面上边框高度	支持数值，如“20px”	可选
		left	文案距离页面左边框距离	支持数值，如“20px”； 居中可传入“center”	可选
号码栏	phoneNumStyle	fontFamily	文案的字体	string	可选
		fontSize	文案的字体大小	支持数值，如“200px”	可选

		fontColor	文案的字体颜色	十六进制颜色码，如“#FFFFFF”	可选
		bgColor	配置号码栏的底色	十六进制颜色码，如“#FFFFFF”	可选
		high	文案距离页面上边框高度	支持数值，如“20px”	可选
		left	文案距离页面左边框距离	支持数值，如“20px”；居中可传入“center”	可选
		width	配置号码栏的宽度	支持数值，如“20px”	可选
		inputStyle	号码栏输入框配置	Object，详细见下表	可选
提示栏	tipStyle	fontFamily	文案的字体	string	可选
		fontSize	文案的字	支持数值，如“200px”	可选

			体大小		
		fontColor	文案的字体颜色	十六进制颜色码，如“#FFFFFF”	可选
		high	文案距离页面上边框高度	支持数值，如“20px”	可选
		left	文案距离页面左边框距离	支持数值，如“20px”；居中可传入“center”	可选
协议栏	agreeStyle	fontFamily	文案的字体	string	可选
		fontSize	文案的字体大小	支持数值，如“200px”	可选
		fontColor	文案的字体颜色	十六进制颜色码，如“#FFFFFF”	可选
		checkedButton	协议勾选按钮	Object，详细见下表	

			配置项		
		hrefStyle	协议名称配置项	Object，详细见下表	
		left	文案距离页面左边框距离	支持数值，如“20px”；居中可传入“center”	可选
		high	文案距离页面上边框高度	支持数值，如“20px”	可选
自定义按钮控件	customControlStyle	ifShow	是否展示自定义控件	Boolean 值，默认值为 FALSE	如要展示自定义控件，则为必选
		width	自定义控件宽度	支持百分比或者数值,如“200px	可选
		height	自定义控件高度	支持百分比或者数值,如“200px	可选

		high	自定义控件距离弹窗上边框高度	支持百分比或者数值,如“200px	必选
		left	自定义控件距离弹窗左边框边距	支持百分比或者数值,如“20px”；居中可传入“center”	必选
		bgColor	自定义控件背景颜色	十六进制颜色码，如“#FFFFFF”	可选
		border	自定义控件边框	可以设置边框粗细，样式,颜色,如”1px solid #FFFFF”	可选
		borderRadius	自定义控件圆角	支持百分比或者数值,如“20px”	可选
		url	自定义控件跳转URL	Url 链接	必选

		name	自定义控件显示文案	字符串	必选
		fontSize	自定义控件字体大小	支持数值,比如“20px”	可选
		fontColor	自定义控件字体颜色	十六进制颜色码，如“#FFFFFF”	可选
		textAlign	自定义控件文本对齐选项	“center/left/right”	可选
		textDecoration	自定义控件下划线	“none/underline”	可选
		fontFamily	文案的字体	string	可选
返回键	returnBtnStyle	width	返回键图片宽度	支持百分比或者数值，如“200px	可选

		height	返回键图片高度	支持百分比或者数值，如“200px	可选
		url	配置返回键图片地址	url	可选
		high	控件距离弹窗上边框高度	数值，如“200px	可选
		left	控件距离弹窗左边框边距	数值，如“20px”；居中可传入“center”	可选
登录按钮	submitBtnStyle	ifShow	是否展示登录按钮	Boolean 值，默认值为 FALSE，	如要展示登录按钮，则为必选
		name	按钮文案	String	可选
		fontSize	按钮文案	支持数值,比如“20px”	可选

			字体大小		
		fontColor	按钮文案字体颜色	十六进制颜色码，如“#FFFFFF”	可选
		fontFamily	按钮文案的字体	string	可选
		bgColor	按钮的底色	十六进制颜色码，如“#FFFFFF”	可选
		high	控件距离弹窗上边框高度	数值，如“200px	可选
		left	控件距离弹窗左边框边距	数值，如“20px”；居中可传入“center”	可选
		textAlign	按钮文案位置	left/right/center	可选
		borderRadius	按钮圆角	数值，如“20px	可选
		width	按钮宽度	数值，如“20px	可选

		height	按钮高度	数值，如“20px	可选

号码栏输入框配置项如下：

	配置项	字段	字段含义	值	说明
号码栏输入框	inputStyle	width	宽度	支持百分比或者数值，如“200px	可选
		height	高度	支持百分比或者数值，如“200px	可选

协议勾选按钮配置项如下：

	配置项	字段	字段含义	值	说明
协议勾选按钮	checkedButton	width	宽度	支持百分比或者数值，如“200px	可选
		height	高度	支持百分比或者数值，如“200px	可选
		uncheckColor	未勾选状态颜色	十六进制颜色码，如“#FFFFFF”	可选
		checkedColor	勾选状态颜色	十六进制颜色码，如“#FFFFFF”	可选
		checkedUrl	勾选状态图片	URL 链接	可选，必须同时配置 checkedUrl 和 uncheckUrl 才生效
		uncheckUrl	未勾选状态图片	URL 链接	可选，必须同时配置 checkedUrl 和 uncheckUrl 才生效

协议名称配置项如下：

	配置项	字段	字段含义	值	说明

协议名称	hrefStyle	fontFamily	文案的字体	string	可选
		fontSize	文案的字体大小	支持数值，如“200px”	可选
		fontColor	文案的字体颜色	十六进制颜色码，如“#FFFFFF”	可选
		agreeArr	自定义协议数组	如:[{name:"协议名称",url:"协议链接"}]	可选

4. token 校验接口（服务端）

4.1. 接口定义

接口名称	tokenValidate
接口描述	h5 前端通过请求电信回调接口,获取到凭证
承载协议	HTTP
承载网络	公网
请求方式	POST
请求限制	
数据格式	Json
接口 URL	https://www.cmpassport.com/h5/onekeylogin/tokenValidate
使用说明	

4.2. Request

注：请使用 post json 格式

4.2.1. Header(http 请求头)

字段名称	是否必填	描述
interfaceVersion	是	接口版本号，填 1.0
appId	是	应用 ID
traceId	是	时间跟踪 ID
timestamp	是	请求消息发送的系统时间，精确到毫秒，共 17 位，格式：20121227180001165
businessType	是	业务类型，填 8

4.2.2. body

字段标识	字段类型	是否必填	字段含义	说明
token	String	是	token 身份凭证	
sign	String	是	签名，业务方 RSA 私钥生成签名，(appId+traceId+timestamp+token+version)（注：“+”号为合并意思，不包含在被加密的字符串中，生成工具详见附录 1）	
userInformation	String	是	浏览器加密指纹（根据 jssdk 返回）	
expandParams	String	否	扩展参数，不同业务取号特定参数放在此参数中，格式：param1=value1|param2=value2 方式传递，参数以竖线 | 间隔方式传递，此参数需 urlencode 编码。	

请求示例：

注：设置 header 头：-H 'Content-Type: application/json'

```
curl -L -X POST 'https://www.cmpassport.com/h5/onekeylogin/tokenValidate' -H
'interfaceVersion: 2.0' -H 'appId: 300012030047' -H 'traceId: 68ccd9f2-8d8e-4e6d-b646-
6aec73278822' -H 'timestamp: 20210318183422402' -H 'businessType: 8' -H 'Content-Type:
application/json' --data
'{"token":"H5HTTPS6c325f3467873abbcbf9d90b0bf09b67a67a5527ed7f48c19f6377f3b401523c","sign":"
72AD4B4AFEF5B22110E722B2B27B7846B432836EDAA5CF1191E2EEA449A5401C79830F9D544168649AEA3AE14584
DA963D601A43C68DEA6278F63020803BA0C8D5C00C602F862F9D117883D05BE1A8183FCA6291E49F5D36C18431E9
0E0BA64B93751B7F0FFB6753330E0F291276590A9FC1E499D75962479C6B978EBBF7FC92","userInformation":
"aVBob25IQEBNb3ppbGxhLzUuMCAoaVBob25IOyBDUFUgaVBob251IE9TIDEyXzNfMSBsaWtlIE1hYyBPUyBYKSBBcHB
sZVdlYktpdC82MDUuMS4xNSAoSOhUTUwslGxpa2UgR2Vja28pIFZIQEAOZmNmYmZhY2Y1YzU3Mjg4NzlkY2Y2ODYOYTc
5Mzg2OQ==”}'
```

4.3. Response

4.3.1. Response Header

参数名称	约束	参数类型	说明
traceId	必选	string	对应请求头的 traceId
appId	必选	string	应用 ID

timestamp	必选	string	响应消息发送的系统时间，精确到毫秒，共 17 位，格式：20121227180001165

4.3.2. Response Body

参数名称	父对象	约束	参数类型	说明
resultCode		必选	string	状态码
desc		必选	string	状态码描述
bcode		可选	string	附加业务状态码
serviceTime		必选	string	时间戳，如：202010201200030
data		必选	object	数据子对象，resultCode 成功时候返回
msisdn	data	必选	string	AES 加密手机号码，加密 key 使用 appkey （AES 解密方法见附录 1）
expandParams	data	可选	string	扩展参数格式：param1=value1|param2=value2 方式传递，参数以竖线 | 间隔方式传递，此参数需 urlencode 编码。
securityPhone	data	必选	string	手机号掩码，如：138****5491

响应示例：

```
{
  "resultCode": "103000",
  "desc": "success",
  "serviceTime": "20210310150254479",
  "data": {
    "msisdn": "m+FDykYrD1qsjpOcBATZug==",
    "expandParams": null,
    "securityPhone": "188****1079"
  }
}
```

5. 返回码及描述

5.1. jssdk

5.1.1. 取号

1）移动取号：

返回码	描述语
103000	成功
500	网络异常，请检查网络设置
503	获取 token 结束
504	js 判断为 wifi 状态
130010	参数为空
105002	移动网关取号失败
105112	时间戳非法
105113	APPID 非法或为空
103101	错误的请求签名
103134	WAP 取号为空
110024	businessType 配置错误
110023	应用没有权益
110025	权益已失效
110026	Origin 校验失败
103111	WAP 网关 IP 错误
111002	黑名单号码用户
103609	物联网 IP 不允许取号
105002	移动网关取号失败
170001	referer 校验失败
103211	其他错误

2）电信取号：

返回码	描述语
103000	成功
301	参数错误
500	网络异常，请检查网络设置
502	电信/联通取号能力关闭
503	参数缺失
103101	错误的请求签名
105113	APPID 非法或为空

301	参数错误
130030	参数为空
105117	TRACEID 为空
108001	businessType is null（电信预取号接口）
104001	businessType config error
110025	权益已失效
105113	APPID 非法或为空
108001	businessType is null（电信回调接口）
105003	电信网关取号失败
110023	应用没有权益
170001	referer 校验失败
110026	Origin 校验失败
130035	其他错误
0	处理结果正常
-10000	取号异常
-10001	取号失败
-10002	参数错误
-10003	解密失败
-10004	无效的 IP
-10005	异网授权回调参数异常
-10006	授权失败，且属于电信网络
-10007	重定向到异网取号
-10008	超过预设取号阀值
-10009	时间超期
-10010	号码识别异常
-10011	运营商不匹配
-10012	区域不匹配
-10013	业务类型不支持该运营商
-10014	AES 解密失败
-10015	Ipv6 取号失败
-10016	安全校验失败
-10017	redirect 方式需要 https 的 callback 地址
-20005	签名非法
-20006	应用不存在
-20007	公钥数据不存在
-20100	内部解析错误
-20102	加密参数解析失败

-30001	时间戳非法
-30003	topClass 失效
-99999	服务内部错误
51002	参数为空
51114	无法获取手机号数据
51207	获取 accessCode 使用的 appid 与本次操作的 appid 不一致
51208	无效的 accessCode,该 accessCode 无法在该业务中使用

3）联通取号：

返回码	描述语
103000	成功
500	网络异常，请检查网络设置
502	电信/联通取号能力关闭
503	参数缺失
103101	错误的请求签名
105113	APPID 非法或为空
301	参数错误
130030	参数为空
105117	TRACEID 为空
108001	businessType is null（电信预取号接口）
104001	businessType config error
110025	权益已失效
105001	联通网关取号失败
105113	APPID 非法或为空
108001	businessType is null（电信回调接口）
105003	联通网关取号失败
110023	应用没有权益
110025	权益已失效
170001	referer 校验失败
110026	Origin 校验失败
130035	其他错误

5.1.2. 获取 token

返回码	描述语
501	用户取消授权
507	用户未补齐 4 位号码
508	用户未勾选协议

103002	没有填写必传参数
104000	app 不存在
104001	businessType 校验失败
104003	应用没有权益
104004	权益已失效
104005	referer 校验失败
104006	origin 校验失败
104007	accessToken 不存在
104008	accessToken 校验失败
104009	超过失败次数限制获取 Token
104010	配置错误
104011	手机号不能为空
104012	本机号码校验失败
104014	businese TYpe 校验失败
104015	运管配置错误
104016	开放配置错误
104018	正在处理，请稍后
104020	电信取号异常
104021	联通取号异常

5.2. 服务端 token 校验

返回码	返回码描述
103000	成功
103001	fail
103002	没有填写必传参数
103003	forbidden
103004	resource not found
103005	inner error
104000	app 不存在
104001	businessType 校验失败
104003	应用没有权益
104004	权益已失效

104007	token 不存在(token 只有 2 分钟有效期)
104008	token 校验失败
104010	配置错误
104013	验证签名失败
104014	businese Type 校验失败
104015	运管配置错误
104016	开放平台配置错误
104017	IP 校验失败（接口设置了 ip 白名单）
104019	token 校验失败[MSISDN_OWNERS]

附录 1：token 校验接口工具

1.RSA 公私钥

方法 1：使用下面的 demo(请在官网下载)

版本	下载附件
java 版本 （maven 项目）	H5一键登录java版工具类.zip
Php 版本	H5一键登录php版工具类.php
Go 版本	H5一键登录Go版工具类.go

（1）Java 版本说明

入口方法：test.Test#main

```
//生成公私钥
generateKeyPair();
// 校验 token 接口 参数 需要签名
testGenSign();
//校验 token 接口 返回手机号 被 aes 加密了 需要解密
testAesDecode();
```

详细描述：

生成 RSA 公私钥 方法：

```
KeyPair keyPair = SecureUtil.generateKeyPair(AsymmetricAlgorithm.RSA.getValue(), 1024);
//注意转成 base64
System.out.println("私钥（base64）：" +
Base64.getEncoder().encodeToString(keyPair.getPrivate().getEncoded()));
System.out.println("公钥（base64）：" +
Base64.getEncoder().encodeToString(keyPair.getPublic().getEncoded()));
```

（2） php 版本说明

入口方法：

```
// 校验 token 接口 参数 需要签名
testGenSign();
//校验 token 接口 返回手机号 被 aes 加密了 需要解密
testAesDecode();
//生成公私钥
generateRsaKeyPair();
```

详细描述：

生成 RSA 公私钥 方法：（记得更改 config 配置指向自己本地目录）

```
function generateRsaKeyPair()
{
    $config = array(
        'config' => 'C:\app\php\php-8.0.6-Win32-vs16-x64\extras\ssl\openssl.cnf',        //
安装 PHP7 会自带这个配置文件
        'digest_alg' => 'sha256',//可以用 openssl_get_md_methods() 查看支持的加密方法
        'private_key_bits' => 1024,
        'private_key_type' => OPENSSL_KEYTYPE_RSA,
```

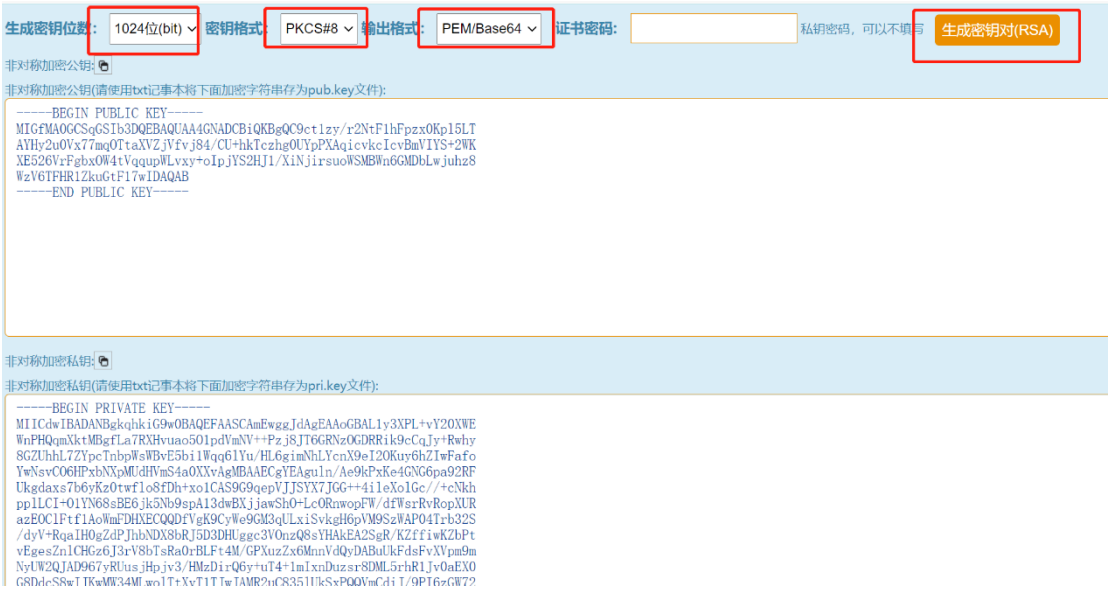
```
    );
    $res = openssl_pkey_new($config);
    openssl_pkey_export($res, $private_key_pem, null, $config);

    $details = openssl_pkey_get_details($res);
    $public_key_pem = $details['key'];

    $keypair = array('private_key' => $private_key_pem, 'public_key' => $public_key_pem);
//    var_dump($keypair);

}
```

方法 2：在线生成，url：http://web.chacuo.net/netrsakeypair
设置项参考下图：



2.SHA256withRSA 签名方法（使用上面的 demo）

（1）java 版本

```
LySign lySign = LySign.of(SignAlgorithm.SHA256withRSA, privateKey, null, true);
System.out.println("签名数据：" + lySign.sign(data));
```

```
public static void testGenSign() throws NoSuchAlgorithmException {
    String appId = "1";
    String traceId = "2";
    String timestamp = LocalDateTimeUtil.format(LocalDateTime.now(), format: "yyyyMMddHHmmssSSS");
    String token = "CM123";
    String version = "1.0";
    String data = appId + traceId + timestamp + token + version;
    LySign lySign = LySign.of(SignAlgorithm.SHA256withRSA, privateKey, publicKey: null, hexStr: true);
    System.out.println("签名数据：" + lySign.sign(data));
}
```

（2）php 版本

```
$lySign = new  LySign(privateKey, publicKey);
$sign = $lySign->sign($data);System.out.println("签名数据：" + lySign.sign(data));
```

```
function testGenSign()
{
    echo "生成签名demo：=================\n";
    echo "当前公钥：" . publicKey . "\n";
    echo "当前私钥：" . privateKey . "\n";

    $appId = "300012034469";
    $traceId = "78be31760e824862816b2ae9a3a220e2";
    $timestamp = "20210426094753203";
    $token = "H5HTTPS855239703d5157accf98686a8595d360d70869a6e29b4b13af986a90d6dbe078";
    $version = "1.0";

    $data = $appId . $traceId . $timestamp . $token . $version;

    echo "待签名数据：" . $data;
    $lySign = new  LySign( privateKey: privateKey,  publicKey: publicKey);

    $sign = $lySign->sign($data);

    echo "签名数据：" . $sign . "\n";
    echo "验证结果:" . $lySign->verify($data, $sign) . "\n";
}
```

3.AES 解密手机号（使用上面的 demo）

（1）java 版本

```
String phone = "18664805491";
String key = "B50BBEF6C4CD4FA8";
AES aes = new AES(getAesKey(key));
String encoded = aes.encryptBase64(phone.getBytes(StandardCharsets.UTF_8));
System.out.println("手机号加密后:" + encoded);
System.out.println("手机号解密:" + new String(aes.decryptBase64(encoded)));
```

```
public static void testAesDecode() throws NoSuchAlgorithmException {
//      String encoded = "C2120D63EFADC9420610F2A5E9C9D7B7";
    String phone = "18664805491";
    String key = "B50BBEF6C4CD4FA8";
    AES aes = new AES(getAesKey(key));
    String encoded = aes.encryptBase64(phone.getBytes(StandardCharsets.UTF_8));
    System.out.println("手机号加密后:" + encoded);
    System.out.println("手机号解密:" + new String(aes.decryptBase64(encoded)));
}
```

（2）php 版本

```
$phone = "18664805491";
$key = "B50BBEF6C4CD4FA8";
$aes = new LyAes($key);
$encoded = $aes->encrypt($phone);
echo "手机号加密后:" . $encoded . "\n";
echo "手机号解密:" . $aes->decrypt($encoded) . "\n";
```

```
function testAesDecode()
{
    echo "解密手机号demo: ==================\n";

    $phone = "18664805491";
    $key = "B50BBEF6C4CD4FA8";
    $aes = new LyAes($key);
    $encoded = $aes->encrypt($phone);
    echo "手机号加密后:" . $encoded . "\n";
    echo "手机号解密:" . $aes->decrypt($encoded) . "\n";
}
```

附录 2：常见问题

1、是否支持 wifi 环境下使用？

答：仅支持数据流量下使用，不支持 wifi、热点环境下使用。请提醒用户请勿在热点环境下使用。

2、授权页输入错误号码是否有限制次数？

答：同一号码连续输入 3 次错误，将会被锁定。为避免此问题，请开发者注意在联调过程中请勿连续输错 3 次。

3、 jssdk 和 token 校验接口中的 sign 是否相同？

答：不相同。两者的生成工具也不同，生成 jssdk 中的 sign 请使用 MD5 签名工具类；生成 token 校验接口的 sign 请使用附录 1 中的“2.SHA256withRSA 签名方法”。

附录 3：常见返回码排查

返回码	排查方法
170001	referer 校验失败。报备的 referer 与实际使用的 referer 不一致。请在平台能力配置页面的“集成网页地址”配置项上补充报备完整的 referer，多个 referer 用英文逗号隔开。
110026	Origin 校验失败。报备的 origin 与实际使用的 origin 不一致。请在平台能力配置页面的“请求来源”配置项上补充报备完整的 origin，多个 origin 用英文逗号隔开。
105112	时间戳非法。时间戳为非 17 位或生成格式不正确。正确格式为 yyyyMMddHHmmssSSS。
105113	Appid 非法或为空。请联系认证侧进行排查。
103101	错误的请求签名。常见原因是签名拼接顺序或拼接字段不正确。
103111	WAP 网关 IP 错误。原因是使用了 wifi 或电信和联通数据网络。
110023	应用没有权益。请检查是否创建正确的能力（能力编码为 59）、businesstype 参数设置正确（businesstype 为 8）、体验量和合同配置量是否有效。
110025	权益已失效。请检查能力配置页面中的“上下线时间”配置项是否过期。
105025	已超限。输错号码 3 次之后再次取号受限。测试时请避免连续 3 次输错。
104009	限制获取 token。在第 3 次输入错误号码时返回。测试时请避免连续 3 次输错。
104012	本机号码校验失败。用户输错中间的四位号码。
104007	① 获取 token 环节：accessToken 不存在。accesstoken 超过 2 分钟有效期或重复校验。 ② 校验 token 环节中，token 超过有效期或重复校验。
104013	验证签名失败。常见原因是客户公私钥的生成未使用指定工具类方法、报备的客户公钥与客户私钥不配对、生签未使用指定工具类方法、签名拼接顺序或字段出错。，请查看文档附录 1 工具类说明。
104017	Ip 校验失败。原因是实际使用的服务器出口 IP 地址未报备。请在平台能力配置页面的 “服务器 IP 白名单配置”配置，多个 ip 地址用英文逗号隔开。
103002	没有填写必传参数。请确认层级是否正确，header 是 http 请求头，不是放 body 中。