

Passport-v4

- 概述
 - 支持功能
 - 引入
 - 历史版本
- API
 - 约定
 - ru参数的使用
 - 创建实例
 - new SohuPassoport(options)
 - 实例方法
 - 帐号密码登录
 - 调用逻辑
 - 调用接口
 - 手机验证登录
 - 调用逻辑
 - 调用接口
 - 退出登录
 - 手机注册
 - 调用逻辑
 - 调用接口
 - 三方登录
 - 调用逻辑
 - 调用接口
 - 微信公众号登录
 - 调用逻辑
 - 调用接口
 - 其他接口服务

概述

Passport安全升级，现提供V4版本JS-SDK。

详细产品逻辑请见：[新系统接入-产品文档](#)

支持功能

- 帐号+密码登录
- 手机号+短信/语音验证码登录
- 三方登录
- 扫码登录（待开发）
- 注册
- 退出登录

引入

- 支持直接加载和标准AMD/CMD模式加载。
- 开发(或者测试)环境地址：<https://passportv4-web.bjcnc.scs.sohucs.com/sdk/passport-test.js>
- 生产环境cdn地址(最新版本v4.0.3)：<https://39d0825d09f05.cdn.sohucs.com/sdk/passport-4.0.3.js>
- demo（passport主站）
pc: <https://v4.passport.sohu.com>
mobile: <https://m.passport.sohu.com>
- 提供了未压缩的debug版本的js地址：
开发测试环境：<https://passportv4-web.bjcnc.scs.sohucs.com/sdk/passport-test-debug.js>
生产环境：<https://39d0825d09f05.cdn.sohucs.com/sdk/passport-4.0.3-debug.js>
（v4.0.1开始）
- 其他

历史版本

- v4.0.2: <https://39d0825d09f05.cdn.sohucs.com/sdk/passport-4.0.2.js>
- v4.0.1: <https://39d0825d09f05.cdn.sohucs.com/sdk/passport-4.0.1.js>
- v4.0.0: <https://39d0825d09f05.cdn.sohucs.com/sdk/passport-4.0.0.js>

API

约定

- 安全考虑，接口调用都启用了https，因为协议跨域限制，业务线调用以下接口的页面要求必须是https。
- 只提供通信接口调用，不再提供UI层面相关API
- error回调函数的返回值，数据结构为：Object {status: , body: "", msg: ""}
- 对于非.sohu.com域的业务线，调用登录，注册和退出接口时，需通过回调地址(即ru参数)的方式来获取接口返回值。
- 海外手机号格式约束：“区号-手机号码”，如：“86-18611019389”（分隔符为英文半角减号）

ru参数的使用

当调用接口时，传递了ru参数，则返回结果会以url参数的形式追加到ru的回跳地址后并302跳转过去。

返回参数：

参数名	说明
status	状态码
message	状态码说明
crotoken	校验码，用于获取用户的passport，ppinf和pprdig的信息，获取接口见 校验跨域token 并获取 cookie

创建实例

new SohuPassport(options)

创建passport实例对象（目前版本在创建过程中会发送异步种实例方法依赖的cookie，所以建议在new之后稍滞后调用实例方法）

```
const passport = new SohuPassport({
  appid:888, //必传。为passport为各个业务线分配的专属appid
  gidinf: xxx, //非必传。用于扫码登录时，被扫设备的gid
  serialno: xxx //非必传。用于扫码登录时，二维码序列号
})
```

实例方法

以下示例假定创建的实例对象为passport。

帐号密码登录

调用逻辑

1. 用户输入帐号密码后，点击登录，调用 帐号+密码登录 接口。校验逻辑具体如下：
 - 如果当前帐号未设置密码，会返回 未设置密码。错误码：423。业务可引导用户通过找回密码设置密码或使用其他方式登录（如手机验证码登录）
 - 当前如果用户连续2次输入密码错误，第三次校验时，会要求输入图形验证码（必须）。错误码：465
 - 帐号或密码校验错误次数超过5次，ip会被封锁，封锁时间为1小时，1小时后自动解锁，用户可重试。错误码：453
 - 如果当前帐号没有绑定手机（邮箱+密码登录），会返回需要绑定手机的错误码（461）
2. 如果接口返回需要图形验证码，则调用 getLoginPicture() 进行获取和校验，需要说明如下：
 - 单个图形验证码在5分钟内有效，可以反复校验
 - 如果用户图形验证码校验失败，需要业务线重新调取验证码来更新验证码
3. 如果返回需要绑定手机，则调用：getBindPhoneCapcha(options) 获取验证码，调用 bindPhone(opeions) 校验
 - 校验通过后，即登录完成，会返回相应的登录数据，具体返回值，见： bindPhone(opeions)返回值
 - 如果用户未完成绑定，则为登录失败

注:在getBindPhoneCapcha(options)返回值中，会明确说明当前手机是绑定成了登录手机和安全校验手机，还是仅绑定为安全校验手机

调用接口

- (1) 帐号+密码登录

```
passport.login({
  success: Function(data, status)    // 当status为2**时回调(可选)
  error: Function(err)              // 当status不是2**时回调(可选)
  params: {
    userid: String // 帐号(必传)
    password: String // 密码(必传)
    captcha: String // 图形验证码(可选)，当登录失败2次后，需要传此值
    persistentCookie: String | Number //
    是否保留cookie, 0: cookie为session级别(默认), 1: 保留cookie14天(可选)
    ru: String //
    回跳地址(可选)，解决接口跨域问题。接口返回值将以参数形式提交到该地址，由业务线自行处理登录结果。若登
    录成功，返回值中会包含ppinf和pprdig参数
  }
})
```

状态码说明：

status	message	说明
200		登录成功
400	bad request	缺少所需的cookie (gidinf, jv)
404	resource dose not exist	用户名或密码错误
420	picture captcha error	图片验证码不正确
423	password not set	账户未设置密码 (手机号+验证码形式登录过的手机号，在帐号+密码形式登录时会提示此错误)
450	parameter error	用户名或密码为空
453	account was locked	帐号登录失败超过5次，ip被封锁(不含图片验证码错误)
461	mobile error	需要绑定手机
465	need captcha	用户需输入图片验证码

(2) getLoginPicture()

获取用户名+密码方式登录的图片验证码

```
let pictureUrl = passport.getLoginPicture()
```

注：每次调用，都会返回一个新的url

(3) getBindPhoneCaptcha(options) - 获取绑定手机验证码

```
passport.getBindPhoneCaptcha({
  success: Function(data, status)    // 当status为2**时回调(可选)
  error: Function(err)              // 当status不是2**时回调(可选)
  params: {
    mobile: String // 手机号(必传)
    way: String || Number // 短信发送方式, 0: 短信, 1: 语音(必传)
  }
})
```

状态码说明：

status	message	说明
--------	---------	----

200		验证码已发送
400	bad request	非法请求
421	operation is limited	验证码发送次数太多
450	parameter error	参数错误: mobile为空, biz错误或way错误
461	mobile error	手机号格式不正确
474	only support voice smcode	只能发送语音验证码
500	server error	服务器繁忙

(4) bindPhone(options) -登录绑定手机号

```
passport.bindPhone({
  success: Function(data, status)    // 当status为2**时回调(可选)
  error: Function(err)              // 当status不是2**时回调(可选)
  params:{
    mobile: String  // 手机号(必传)
    smcode: String  // 短信验证码
    ru: String      //
  }
})
```

回跳地址(可选), 解决接口跨域问题。接口返回值将以参数形式提交到该地址, 由业务线自行处理登录结果。若登录成功, 返回值中会包含ppinf和pprdig参数

状态码说明:

status	message	说明
200	success	绑定登录手机成功
201	success	只绑定成安全手机
422	captcha error	验证码错误
42201	captcha not set	验证码不存在或已过期
427	security mobile limited	安全手机绑定上限
450	parameter error	手机号或验证码为空
462	account has been binding mobile	当前帐号已绑定过登录手机
482	operation timeout	操作超时
500	server error	服务器繁忙

手机验证登录

调用逻辑

- 在获取手机验证码前, 需要用户校验图形验证码, 调用getPhonePicture() 获取
 - 用户输入手机号后, 调用 getSignInCaptcha(options) 获取手机验证码。
 - 在下发手机验证码前会校验图形验证码。如果校验失败, 会返回图形验证码不正确。错误码: 420
 - 如果1小时内一个手机已经发送5次短信验证码, 会返回 验证码发送次数太多的错误码。错误码: 421。可提示用户1小时后再试
 - 因为图形验证码有效期为5分钟, 如果在获取图形验证码5分钟后, 调用获取手机验证码时, 会返回操作过时的错误。错误码: 482。可重新调用获取图形验证码来更新图形验证码
 - 针对当前市场上存在的众多黑产手机, 我们在监测到当前用户ip与手机号码归属地不一致时, 会要求用户必须使用语音验证码进行校验。错误码: 474
 - 下发成功会返回 验证码已发送 (200)
- 注: 当需要用户接听语音验证码时, 建议提前弹层(或其他方式)提示用户: 您将收到一个950开头的语音验证码, 请注意接听。避免用户拒接
- 用户输入短信验证码, 点击登录时, 调用: mobileLogin(options)

- 获取到的单个验证码可被校验5次，第6次校验时，会返回 验证码已过期的错误，错误码：42201
- 如果用户连续错误次数超过5次，ip会被封禁1小时，错误码：453
- 如果一个手机已经被5个帐号绑定，则在被第6个帐号绑定时，会返回错误码：427

调用接口

(1) mobileLogin(options)

手机号+验证码方式登录，若手机号不存在会自动注册。

```
passport.mobileLogin({
  success: Function(data, status)    // 当status为2**时回调(可选)
  error: Function(err)              // 当status不是2**时回调(可选)
  params:{
    mobile: String  // 手机号(必传)
    smcode: String  // 短信验证码(必传)
    persistentCookie: String | Number  //
    是否保留cookie, 0: cookie为session级别(默认), 1: 保留cookie14天(可选)
    ru: String      //
    回跳地址(可选)，解决接口跨域问题。接口返回值将以参数形式提交到该地址，由业务线自行处理登录结果。若登录成功，返回值中会包含ppinf和pprdig参数
  }
})
```

状态码说明：

status	message	说明
200		登录成功
400	bad request	缺少所需的cookie (gidinf, jv)
422	captcha error	手机验证码不正确
42201	captcha not set	手机验证码不存在或已过期
427	security mobile limited	手机绑定上限，无法注册
450	parameter error	手机号或验证码为空
453	account was locked	失败超过5次，ip被封锁

(2) getPhonePicture()

获取手机号+验证码方式登录的图片验证码

```
let pictureUrl = passport.getPhonePicture()
```

注：每次调用，都会返回一个新的url

(3) getSignInCaptcha(options)

获取登录手机验证码

```
passport.getSignInCaptcha({
  success: Function(data, status)    // 当status为2**时回调(可选)
  error: Function(err)              // 当status不是2**时回调(可选)
  params: {
    mobile: String  // 手机号(必传)
    way: String || Number // 短信发送方式, 0: 短信, 1: 语音(必传)
    captcha: String // 图形验证码(必传)
  }
})
```

状态码说明：

status	message	说明
200		验证码已发送
400	bad request	非法请求
420	picture captcha error	图片验证码不正确
421	operation is limited	验证码发送次数太多
450	parameter error	参数错误：mobile为空，biz错误或way错误
461	mobile error	手机号格式不正确
474	only support voice smcode	只能发送语音验证码
482	operation timeout	操作过时
500	server error	服务器繁忙

退出登录

logout(options)

```
passport.logout({
  success: Function(data, status)    // 当status为2**时回调(可选)
  error: Function(err)              // 当status不是2**时回调(可选)
  params: {
    ru: String                      //
    回跳地址(可选)，解决接口跨域问题。接口返回值将以参数形式提交到该地址，由业务线自行处理登录结果。
  }
})
```

状态码说明：

status	message	说明
200		退出成功
400	bad request	缺少所需的cookie (gidinf, jv)

手机注册

调用逻辑

- 在获取手机验证码前，需要用户校验图形验证码，调用 getRegisterPicture() 获取
- 用户输入手机号后，调用 getRegisterCaptcha(options) 获取手机验证码。
 - 在下发手机验证码前会校验图形验证码。如果校验失败，会返回图形验证码不正确。错误码：420

- 如果1小时内一个手机已经发送5次短信验证码，会返回 验证码发送次数太多的错误码。错误码：421。可提示用户1小时后再试
- 因为图形验证码有效期为5分钟，如果在获取图形验证码5分钟后，调用获取手机验证码时，会返回操作过时的错误。错误码：482。可重新调用获取图形验证码来更新图形验证码
- 针对当前市场上存在的众多黑产手机，我们在监测到当前用户ip与手机号码归属地不一致时，会要求用户必须使用语音验证码进行校验。错误码：474
- 如果手机已经注册，会返回手机已经注册的错误码（463）
- 下发成功会返回 验证码已发送（200）

注：当需要用户接听语音验证码时，建议提前弹层（或其他方式）提示用户：您将收到一个950开头的语音验证码，请注意接听。避免用户拒接

3. 用户输入短信验证码、密码，点击注册时，调用：register(options)

- 第一步 会校验手机是否已注册，如果已注册，返回错误码（463）
- 第二步 密码校验，若为弱密码（我们的弱密码库），则需要用户更换密码。错误码：438
- 第三步 验证码校验
 - 获取到的单个验证码可被校验5次，第6次校验时，会返回 验证码已过期的错误，错误码：42201
 - 如果用户连续错误次数超过5次，ip会被封禁1小时，错误码：453
- 第四步 手机是否可被绑定为安全校验手机，如果一个手机已经被5个帐号绑定为安全校验手机，则在被第6个帐号绑定时，会返回错误码：427

调用接口

(1) getRegisterPicture() - 获取注册图片验证码

```
let pictureUrl = getRegisterPicture()
```

注：每次调用，都会返回一个新的url

(2) getRegisterCaptcha(options) - 获取注册手机验证码

```
passport.getRegisterCaptcha({
  success: Function(data, status) // 当status为2**时回调(可选)
  error: Function(err) // 当status不是2**时回调(可选)
  params: {
    mobile: String // 手机号(必传)
    way: String || Number // 短信发送方式, 0: 短信, 1: 语音(必传)
    captcha: String // 图形验证码(必传)
  }
})
```

状态码说明：

status	message	说明
200	success	验证码已发送
400	bad request	非法请求
421	operation is limited	验证码发送次数太多
420	picture captcha error	图片验证码错误
450	parameter error	参数错误：mobile为空，biz错误或way错误
461	mobile error	手机号格式不正确
463	mobile have been binding	该手机号码已被绑定过
474	only support voice smcode	只能发送语音验证码
500	server error	服务器繁忙

(3) register(options) - 注册

```
passport.register({
  success: Function(data, status)    // 当status为2**时回调(可选)
  error: Function(err)              // 当status不是2**时回调(可选)
  params: {
    mobile: String  // 手机号(必传)
    password: String // 密码(必传)
    smcode: String  // 短信验证码(必传)
    ru: String      //
    // 回跳地址(可选), 解决接口跨域问题。接口返回值将以参数形式提交到该地址, 由业务线自行处理登录结果。若登录成功, 返回值中会包含ppinf和pprdig参数
  }
})
```

状态码说明:

status	message	说明
200		注册成功
422	captcha error	手机验证码错误
427	security mobile limited	安全手机绑定上限
438	password unsafe	弱密码
450	parameter error	手机号、验证码或密码为空
463	mobile already has bind	手机号已存在
500	server error	服务器繁忙

三方登录

调用逻辑

- 业务线调用 `getThirdLoginUrl(options)` 接口, 通行证会根据调用的业务线拉起三方授权。
注: 需要在对接研发阶段拉起三方时需要的信息给到通行证研发
- 如果在返回信息中有ppok的cookie且没有ppinf的cookie, 则表明当前三方未绑定手机, 需调用 `getBindPhoneCaptcha(options)` 接口进行绑定 (详见 帐号+密码登录下的 绑定手机)

调用接口

`getThirdLoginUrl(options)` - 三方登录

```
const url = passport.getThirdLoginUrl({
  provider: String  // qq, weChat, sina, baidu, taobao, alipay N选一(必传)
  ru: String        // 三方登录成功后回调地址(需要encodeURIComponent编码, 否则带#等特殊符号的会异常)
})
```

特殊说明: 有ppok的cookie且没有ppinf的cookie为需要绑定手机号的状态

注: 测试环境要绑host: 10.10.1.222 passport.sohu.com plus.sohu.com

微信公众号登录

指在微信公众号内使用微信登录

调用逻辑

- 业务线调用 `getThirdLoginUrl(options)` 接口, 通行证会根据调用的业务线拉起三方授权。
注: 需要在对接研发阶段拉起三方时需要的信息给到通行证研发
- 如果在返回信息中有ppok的cookie且没有ppinf的cookie, 则表明当前三方未绑定手机, 需调用 `getBindPhoneCaptcha(options)`

接口进行绑定（详见 帐号+密码登录下的 绑定手机）

调用接口

getWechatMPLoInUrl(options)

```
const url = passport.getWechatMPLoInUrl({
  openkey: String // 微信公众号开发者帐号的appid
  ru: String // 三方登录成功后回调地址(需要encodeURIComponent编码，否则带#等特殊符号的会异常)
})
```

特殊说明：有ppok的cookie且没有ppinf的cookie为需要绑定手机号的状态

接入说明：

方式一，将自己公众号的授权回调域配置成 plus.sohu.com。（注：此域名后期将下线，推荐使用方法二）

方式二（推荐），不用配置授权回调域，业务线自己接收微信回调结果，并302到passport的三方授权接口进行登录。具体参见0Auth2.0 登录

其他接口服务

1. checkPhone(options)

检查登录或安全校验手机号

```
passport.checkPhone({
  success: Function(data, status) // 当status为2**时回调(可选)
  error: Function(err) // 当status不是2**时回调(可选)
  params:{
    mobile: String // 手机号(必传)
    log: Number|String // 是否只校验登录手机。0：校验安全手机（默认），1：校验登录手机
  }
});
```

状态码说明：

status	message	说明
200	success	手机号可以使用
421	operation is limited	操作频繁，1小时只能校验5次
450	parameter error	手机号为空
463	mobile have been binding	手机号已被绑定，包括安全手机绑定超限

2. checkUserPhoneBind(options)

检查当前已登录的用户是否绑定了手机

```
passport.checkUserPhoneBind({
  success: Function(data, status) // 当status为2**时回调(可选)
  error: Function(err) // 当status不是2**时回调(可选)
  params:{
    log: Number|String // 是否只校验登录手机。0：校验安全手机（默认），1：校验登录手机
  }
});
```

状态码说明：

status	message	说明
200	success	用户已绑定手机

405	user no login	用户未登录
421	operation is limited	操作频繁，1小时只能校验5次
450	parameter error	appid错误
455	account has not bind mobile	用户未绑定手机

3. passportLogin(options)

供56千帆使用，passport+密码登录

```
passport.passportLogin({
  success: Function(data, status)    // 当status为2**时回调(可选)
  error: Function(err)               // 当status不是2**时回调(可选)
  params: {
    passport: String                // 登录的用户名(必传)
    password: String                // 密码(必传)
    captcha: String                 // 图形验证码(非必传)
    pagetoken: String               // 图片验证码识别号(非必传)
    persistentCookie: String | Number //
    是否保留cookie, 0: cookie为session级别(默认), 1: 保留cookie14天(可选)
    ru: String                      //
    回跳地址(可选), 解决接口跨域问题。接口返回值将以参数形式提交到该地址, 由业务线自行处理登录结果。若登
    录成功, 返回值中会包含ppinf和pprdig参数
  }
})
```

状态码说明:

status	message	说明
200	success	登录成功
400	bad request	缺少所需的cookie (gidinf, jv)
404	resource dose not exist	用户名或密码错误
405	operation is restricted	不支持的业务线
450	parameter error	用户名或密码为空
453	account was locked	失败超过5次, ip被封锁
461	mobile error	需要绑定手机
465	need captcha	用户需输入图片验证码

4. scanCodeLogin(options)

扫码登录接口, 用于扫码后进行用户名密码登录

```
passport.scanCodeLogin({
  success: Function(data, status)    // 当status为2**时回调(可选)
  error: Function(err)              // 当status不是2**时回调(可选)
  params: {
    userid: String                  // 登录的用户名(必传)
    password: String                // 密码(必传)
    captcha: String                 // 图形验证码(非必传)
    pagetoken: String               // 图片验证码识别号(非必传)
    ru: String                      //
  }
  // 回跳地址(可选)，解决接口跨域问题。接口返回值将以参数形式提交到该地址，由业务线自行处理登录结果。若登录成功，返回值中会包含ppinf和pprdig参数
})
```

状态码说明：

status	message	说明
200	success	登录成功
400	bad request	缺少所需的cookie (gidinf, jv, srn)
404	resource dose not exist	用户名或密码错误
423	password not set	账户未设置密码
450	parameter error	参数错误
453	account was locked	失败超过5次，ip被封锁
461	mobile error	需要绑定手机
465	need captcha	用户需输入图片验证码