

IRS应用发布

操作手册



法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您继续阅读本文档正文内容，表示认可本声明的全部内容。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本文档内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的文档。您应当实时关注文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或惩罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任(即使阿里云已被告知该等损失的可能性)。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。未经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

| 格式 | 说明 | 样例 |
|---|------------------------------------|--|
|  危险 | 该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  危险 重置操作将丢失用户配置数据。 |
|  警告 | 该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  警告 重启操作将导致业务中断，恢复业务时间约十分钟。 |
|  注意 | 用于警示信息、补充说明等，是用户必须了解的内容。 |  注意 权重设置为0，该服务器不会再接受新请求。 |
|  说明 | 用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。 |  说明 您也可以通过按Ctrl+A选中全部文件。 |
| > | 多级菜单递进。 | 单击设置>网络>设置网络类型。 |
| 粗体 | 表示按键、菜单、页面名称等UI元素。 | 在结果确认页面，单击确定。 |
| Courier字体 | 命令或代码。 | 执行 cd /d C:/window 命令，进入 Windows系统文件夹。 |
| 斜体 | 表示参数、变量。 | <code>bae log list --instanceid</code> <code>Instance_ID</code> |
| [] 或者 [a b] | 表示可选项，至多选择一个。 | <code>ipconfig [-all -t]</code> |
| {} 或者 {a b} | 表示必选项，至多选择一个。 | <code>switch {active stand}</code> |

目录

| | |
|-----------------------|----|
| 1.系统简介 | 09 |
| 2.基本概念 | 10 |
| 3.开发商 | 11 |
| 3.1. 登录开发商工作台 | 11 |
| 3.2. 创建系统 | 11 |
| 3.3. API管理 | 13 |
| 3.3.1. API注册流程 | 14 |
| 3.3.2. 新增API | 14 |
| 3.3.3. RPC网关API参数配置 | 18 |
| 3.3.3.1. HTTP和HTTPS协议 | 18 |
| 3.3.3.2. HSF协议 | 19 |
| 3.3.4. 调试API | 20 |
| 3.3.5. 上线API | 20 |
| 3.3.6. API权限和流量控制 | 21 |
| 3.3.7. 升级API | 22 |
| 3.4. 服务接入 | 24 |
| 3.4.1. 申请添加服务 | 24 |
| 3.4.2. API申请 | 25 |
| 3.5. 设置跨域白名单 | 25 |
| 3.6. H5应用接入 | 27 |
| 3.7. 应用调用API | 36 |
| 3.7.1. 应用网关接入 | 36 |
| 3.7.2. JSAPI接入 | 39 |
| 3.8. 应用调试 | 39 |
| 3.8.1. Debug调试工具 | 39 |
| 3.8.1.1. 简介 | 39 |

| | |
|-------------------------|----|
| 3.8.1.2. 准备工作 | 42 |
| 3.8.1.3. 查看日志 | 43 |
| 3.8.1.4. 前端页面调试 | 46 |
| 3.8.1.5. 调试JSBridge API | 49 |
| 3.8.2. H5页面调试 | 51 |
| 3.9. 应用部署发布 | 54 |
| 3.9.1. 构建发布系统简介 | 54 |
| 3.9.2. 部署发布应用 | 55 |
| 3.9.3. 源码检测（服务侧应用） | 56 |
| 3.9.4. 应用提交验收 | 58 |
| 3.9.5. 配置完成（治理侧应用） | 59 |
| 3.10. 代码发布管理 | 59 |
| 3.10.1. 线上版本代码回滚 | 59 |
| 3.11. 我的任务 | 60 |
| 3.12. JSBridge API | 61 |
| 3.12.1. API概览 | 62 |
| 3.12.2. 调试JSBridge API | 64 |
| 3.12.3. ZWJSBridge接入方式 | 67 |
| 3.12.4. 缓存 | 68 |
| 3.12.4.1. 存储数据缓存 | 69 |
| 3.12.4.2. 读取数据缓存 | 69 |
| 3.12.4.3. 删除缓存 | 70 |
| 3.12.5. Navigation窗口类 | 70 |
| 3.12.5.1. 导航栏标题 | 70 |
| 3.12.5.2. 菜单 | 71 |
| 3.12.5.3. 新开窗口 | 72 |
| 3.12.5.4. 关闭当前页面 | 73 |
| 3.12.6. 用户 | 73 |

| | |
|----------------------|----|
| 3.12.6.1. 获取用户类型 | 73 |
| 3.12.6.2. 获取用户当前UI风格 | 74 |
| 3.12.6.3. 支付宝扫脸认证 | 75 |
| 3.12.6.4. 公安一所可信扫脸认证 | 77 |
| 3.12.7. Device设备类 | 78 |
| 3.12.7.1. 打电话 | 79 |
| 3.12.7.2. 发短信 | 79 |
| 3.12.7.3. 获取经纬度信息 | 80 |
| 3.12.7.4. 获取设备唯一标识 | 81 |
| 3.12.7.5. 获取网络类型 | 81 |
| 3.12.7.6. 剪切板 | 82 |
| 3.12.8. 业务类 | 82 |
| 3.12.8.1. 埋点 | 83 |
| 3.12.8.2. 获取当前地区 | 84 |
| 3.12.8.3. 选择图片 | 84 |
| 3.12.8.4. 图片保存到本地 | 86 |
| 3.12.8.5. 扫一扫 | 86 |
| 3.12.8.6. 启用支付功能 | 87 |
| 3.12.8.6.1. 支付能力 | 89 |
| 3.12.8.6.2. 启用支付功能 | 91 |
| 3.12.9. UI界面类 | 93 |
| 3.12.9.1. 确认框 | 93 |
| 3.12.9.2. 提示框 | 94 |
| 3.12.9.3. 弱提示 | 94 |
| 3.12.9.4. 文本输入框 | 95 |
| 3.12.9.5. 单选列表 | 97 |
| 3.12.9.6. 等待蒙版显示 | 98 |
| 3.12.9.7. 等待蒙版隐藏 | 99 |

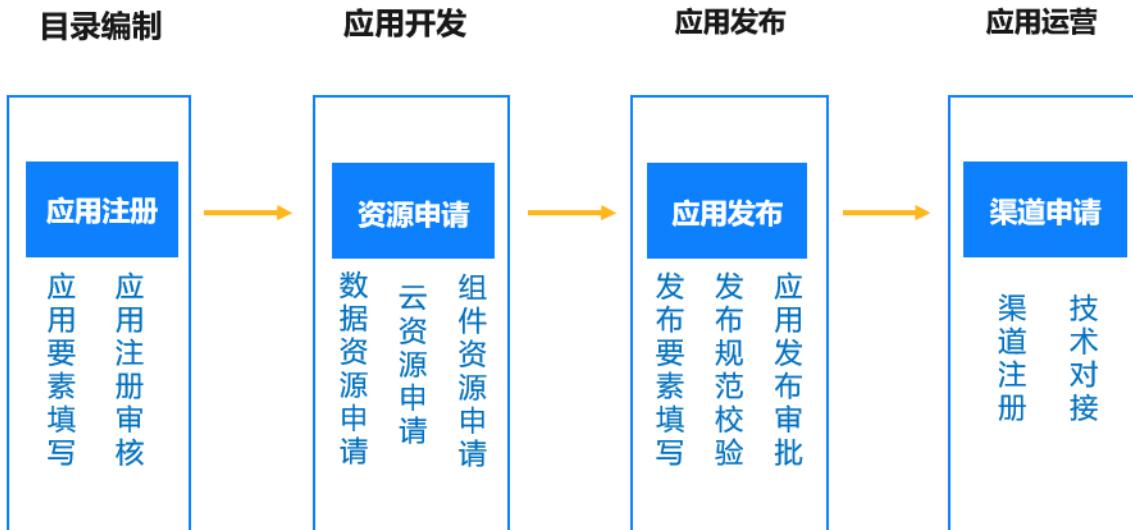
| | |
|-----------------|-----|
| 3.12.9.8. 选择城市 | 100 |
| 3.12.9.9. 分享 | 100 |
| 3.12.9.10. 直接分享 | 101 |
| 3.12.10. 请求类 | 103 |
| 3.12.10.1. 无线网关 | 103 |

1.系统简介

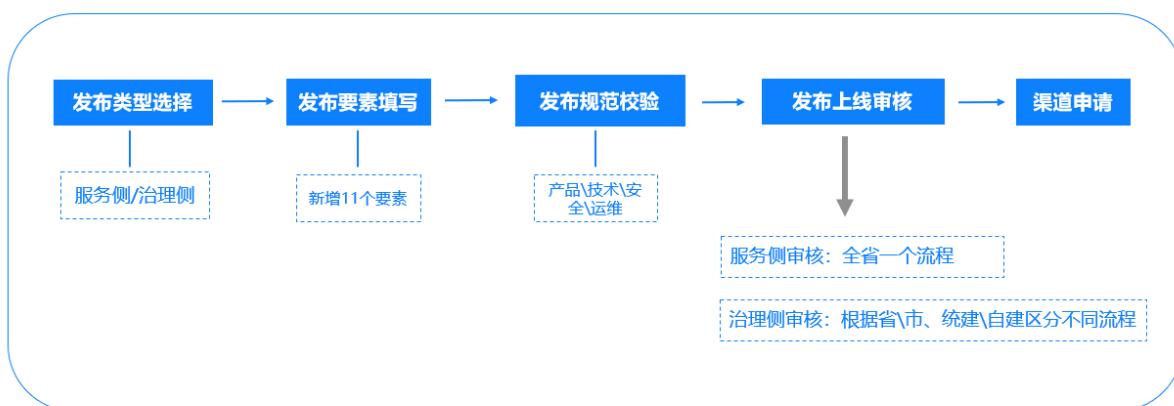
IRS应用发布系统定位

应用发布是IRS（一体化资源系统，Integrated Resources System，简称IRS）应用闭环管理的重要环节，该环节定义应用发布相关的目录要素、规范标准、审核流程。

根据发布端不同，IRS应用发布系统主要分为面向群众和企业掌上办事的服务侧发布，以及面向公务人员掌上办公的治理侧发布。应用发布完成后，支持渠道申请，实现一次开发，多渠道共享。



IRS应用发布流程



2. 基本概念

服务侧

服务侧负责将应用发布至浙里办APP和政务服务网，应用发布类型不同，应用发布流程也不同：

- 服务侧发布类型为浙里办APP时，代码需要托管在开发商工作台，服务侧名称列表中应用初始化状态为部署准备中。服务侧创建成功后，由开发商完成应用部署发布以及提交验收等流程，请参见[部署发布应用](#)。
- 服务侧发布类型为政务服务网时，代码无需托管在开发商工作台，服务侧名称列表中应用初始化应用状态为待发布审批。服务侧创建成功后，您可以直接提交应用发布[审批](#)。

治理侧

治理侧负责将应用发布至浙政钉APP，应用发布类型不同，应用发布流程也不同：

- 应用发布类型为H5时，系统默认代码托管，创建治理侧发布后，由开发商在开发商工作台完成应用部署、提交验收等操作，业主单位政府工作人员在IRS主工作台完成应用验收、提交审核以及发布审核操作，审核通过后将应用发布上线。
- 应用发布类型为小程序时，系统默认代码不托管，创建治理侧发布后，由业主单位政府工作人员在IRS主工作台提交应用发布审核，相关人员审核通过后将应用发布上线。

主工作台

浙江省一体化数字资源系统（<http://irs.zj.gov.cn/home>）工作台的操作对象为政府工作人员，用于完成创建服务侧/治理侧发布、发布管理、发布审核等操作，本手册中将该工作台简称为主工作台。

开发商工作台

浙江省一体化数字资源系统（<https://op-irs.zj.gov.cn/mobile/index>）工作台的操作对象为厅局或者地市开发商，用于完成代码上传、代码检测、提交验收等操作，本手册中将该工作台简称为开发商工作台。

原浙里办应用开发管理平台

- 原浙里办应用开发管理平台入口将逐步关闭。
- 对于已在原浙里办应用开发管理平台上建设的存量浙里办应用，需要您前往IRS工作台进行[存量应用](#)关联，后续在IRS应用发布系统进行应用变更和发布等操作。
- 如需新增应用，请参照IRS应用发布规范在IRS应用发布系统中完成应用发布。

3.开发商

3.1. 登录开发商工作台

开发商通过浙江省一体化数字资源系统开发商工作台完成服务侧和治理侧的应用部署发布操作，包含注册RPC接口、服务接入、应用代码部署、代码检测、提交验收等操作，本手册中将该工作台简称为开发商工作台。本节以Chrome浏览器为例，介绍开发商如何登录开发商工作台。

前提条件

- 开发商已获取登录账号。
浙江省一体化资源系统开发商联系业主单位获取账号，推荐使用Chrome浏览器。
- 业主单位政府工作人员已[创建服务侧发布](#)、已完成[存量应用关联](#)或者已[创建治理侧发布](#)。

操作步骤

1. 在浏览器地址栏中，输入登录地址：<https://op-irs.zj.gov.cn/mobile/index>，按回车键。
2. 通过手机浙政钉扫码登录开发商工作台。开发商登录成功后，系统通过手机号码自动将服务同步至开发商工作台的服务侧名称列表或者治理侧名称列表中。



The screenshot shows the IRS developer workbench interface. At the top, there's a header with the IRS logo and navigation links. Below the header, a sidebar on the left contains sections for '我的任务' (My Tasks) and '发布流程' (Release Process). The main content area has two tabs: '服务侧名称列表' (Service Side Name List) and '治理侧名称列表' (Governance Side Name List). Both tabs show tables with columns like '服务侧名称' (Service Side Name), '创建时间' (Creation Time), '更新时间' (Last Update), '类型' (Type), '当前状态' (Current Status), '服务接入数' (Number of Services Connected), 'PC地址' (PC Address), and '移动端地址' (Mobile Address). Each row in the table includes a QR code and buttons for '详情' (Details), '监控' (Monitoring), and '更多' (More).

3.2. 创建系统

系统是唯一识别服务提供方或服务调用方身份的标识，同时API（若应用开发过程使用API时，需要在开发商工作台注册API）以系统为载体。系统创建之后，您就可以在系统中注册API，将一个或多个API封装成一个服务，您可以自行使用该服务，其他开发商可以向您申请使用该服务。

操作步骤

1. [登录开发商工作台](#)。
2. 单击顶部菜单栏的RPC接入，然后单击左侧导航栏系统管理。



3. 在系统管理页面，单击新建系统。
4. 在系统信息页面，配置系统信息、部门信息 以及系统开发商信息。
 - 系统信息

| 参数 | 说明 |
|--------|--------------------|
| 系统名称 | 设置系统的中文名称。 |
| 系统英文名称 | 设置系统的英文名称。 |
| 系统所在区域 | 从下拉列表框中选择系统所在区域。 |
| 系统简介 | 为方便后续管理，为系统设置一段描述。 |

- 部门信息

说明

自系统首次提交审核后，不支持修改系统创建部门和统一社会信用代码，请谨慎填写系统信息。

| 参数 | 说明 |
|-----------|--------------------------------|
| 系统创建部门 | 从下拉列表框中选择系统所属部门。 |
| 统一社会信用代码 | 设置统一社会信用代码。 |
| 业务负责人联系方式 | 设置业务负责人的电话号码。 系统校验手机号码的有效性。 |

| 参数 | 说明 |
|---------|--------------------------------|
| 业务负责人邮箱 | 设置业务负责人的邮箱地址。 系统校验邮箱地址的有效性。 |

◦ 系统开发商信息

| 参数 | 说明 |
|----------|-------------------------------|
| 系统开发商名称 | 设置系统开发商的名称。 |
| 开发人员姓名 | 设置开发人员的姓名。 |
| 开发人员联系方式 | 设置开发人员的联系方式。 系统校验手机号码的有效性。 |
| 开发人员邮箱 | 设置开发人员的邮箱地址。 系统校验邮箱地址的有效性。 |
| 运维人员姓名 | 设置运维人员的姓名。 |
| 运维人员联系方式 | 设置运维人员的手机号码。 系统校验手机号码的有效性。 |
| 运维人员邮箱 | 设置运维人员的邮箱地址。 系统校验邮箱地址的有效性。 |

5. 完成上述参数配置后，单击提交，系统进行参数校验，验证无误后进入审核中状态。

| + 新建系统 | | | | | | |
|--------|-----------|-------|---------------------|-----|------|---------------------------------------|
| 系统名称 | 系统建设部门 | 开发商名称 | 创建时间 | 创建人 | 审批状态 | 操作 |
| 【测试】 | 市、县 (市、区) | 【测试】 | 2020-09-21 13:12:46 | | 审核中 | 编辑 删除 |

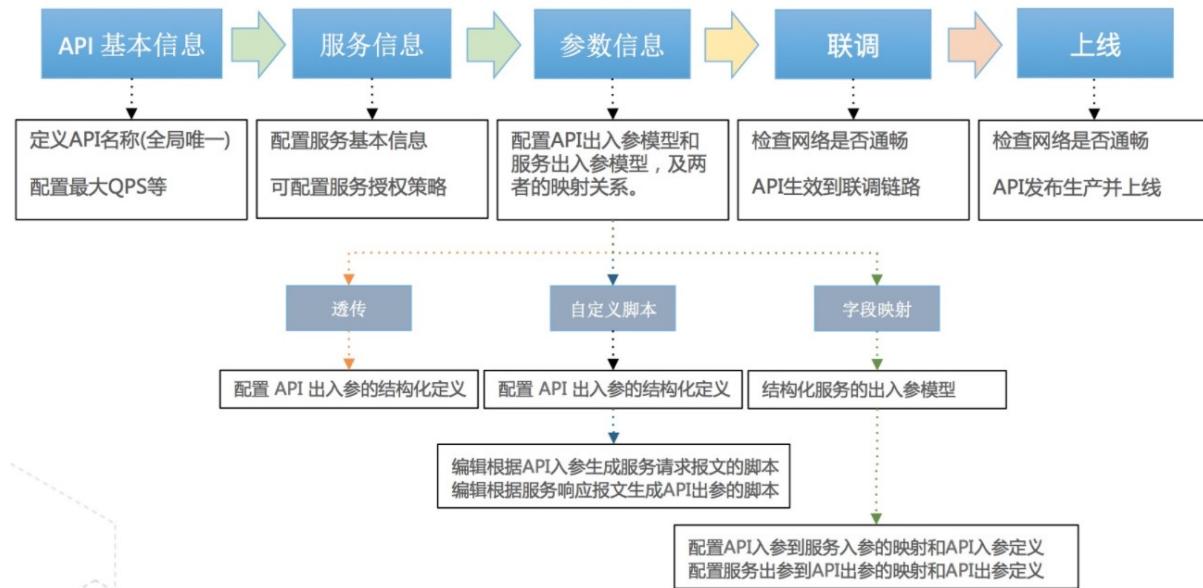
对于审核中的系统：

- 单击编辑，支持修改除系统开发商信息、部门信息中系统创建部门以外的参数，修改完成后单击提交。
- 单击删除，删除该系统。
- 由开发商工作台运维人员在[我的任务](#)中完成系统创建审批。

3.3. API管理

3.3.1. API注册流程

API的注册流程如下图所示：



通过以下三步完成API注册：

1. 注册API
2. 调试API
3. 上线API

3.3.2. 新增API

部署在开发商工作台上的应用使用统一分配的域名，所有网络请求通过统一的无线网关进行转发。因此，应用中使用的API需要在开发商工作台注册，跨域API将无法被访问。

前提条件

开发商已完成系统创建，并通过审核。

操作步骤

1. [登录开发商工作台](#)。
2. 单击顶部菜单栏的RPC接入，然后单击左侧导航栏的API管理。



3. 在API管理页面，单击新建API。
4. 在设置API基本信息页签下，配置API基本信息。

| 参数 | 说明 |
|-----------|---|
| API名称 | 设置API的名称，由英文字母、点号（.）和阿拉伯数字组成，格式为 mgop.[公司名称].[系统名称].[操作名称]，例如 mgop.alibaba.huarong.doTest。 ② 说明 自API名称首次设置完成后，不支持修改，请谨慎操作。 |
| 中文名称 | 设置API中文名称。 |
| API功能描述 | 为便于后续管理，为API添加一段功能描述。 |
| API从属系统 | 从下拉列表框中选择API从属系统。 |
| 允许访问入口 | 允许访问入口选择APP或者H5，本示例选择H5。 |
| API是否需要登录 | 根据业务需要设置调用API时是否需要登录校验。 ② 说明 创建API时，选择需要登录，后续通过APP打开应用时，APP将自带guc-platform等参数。通过其他应用打开应用时，网关检测到没有提供guc-platform等参数，系统将提示参数无guc-platform的错误。 |

5. 完成基本信息配置后，单击下一步。
6. 在填写服务信息页签下，按照页面提示进行参数配置。

The screenshot shows a configuration interface for service access. At the top, there is a radio button group for "Service Access Protocol": HTTP, HTTPS, and HSF. Below this are several input fields:

- "Production Environment Target Address": http://portal.[REDACTED]
- "Intelligent Environment Target Address": http://pre-portal.[REDACTED]
- "Request Encoding": UTF-8
- "Request Method": POST
- "Timeout Time": 1000 ms
- "Flow Control": 1000 TPS

| 参数 | 说明 |
|----------|--|
| 服务接入协议 | 设置API的服务的接入协议类型，包括HTTP、HTTPS和HSF，本示例选择HTTP。 |
| 生产环境目标地址 | 当服务接入协议为HTTP或HTTPS时，设置服务生产环境的地址。 |
| 联调环境目标地址 | 当服务接入协议为HTTP或HTTPS时，设置API联调环境地址。 |
| 请求编码 | 当服务接入协议为HTTP或HTTPS时，设置服务请求编码，默认为UTF-8。 |
| 请求方法 | 当服务接入协议为HTTP或HTTPS时，设置服务的请求方法，包括：POST、GET、HEAD、DELETE、PUT、OPTIONS，本示例选择POST。 |
| 超时时间 | 设置服务请求的超时时间，单位为毫秒。 |
| 限流 | 设置服务限流值，单位为TPS。 例如1000 TPS表示每秒调用API的次数不能超过1000次 |
| HSF接口名称 | 当服务接入协议为HSF时，设置HSF的接口名称。 |
| HSF方法名称 | 当服务接入协议为HSF时，设置HSF的方法名称。 |
| 预发版本号 | 当服务接入协议为HSF时，设置HSF的预发版本号。 |

| 参数 | 说明 |
|---------|---------------------------|
| 预发HSF分组 | 当服务接入协议为HSF时，设置HSF的预发分组。 |
| 生产版本号 | 当服务接入协议为HSF时，设置HSF的生产版本号。 |
| 生产HSF分组 | 当服务接入协议为HSF时，设置HSF的生产分组。 |

7. 完成服务信息配置后，单击下一步。
8. 在填写参数映射关系页签下，配置的API的入参和出参。

Body 入参 application/x-www-form-urlencoded

| 参数名称 | 参数类型 | 具体类型 | 必选 | 长度 | 示例 | 描述 | 文档顺序 | 操作 |
|----------------------|------|------|-------------------------------------|----------------------|----------------------|----------------------|----------------------|---|
| <input type="text"/> | 请选择 | 请选择 | <input checked="" type="checkbox"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | 添加属性 删除 |

[+ 添加参数](#)

Body 出参

| 参数名称 | 参数类型 | 具体类型 | 长度 | 示例 | 描述 | 文档顺序 | 操作 |
|------|------|------|----|----|----|------|----|
| 没有数据 | | | | | | | |

[+ 添加参数](#)

| 参数 | 说明 |
|-----------|---|
| 参数名 | 设置API的参数名。 |
| 参数类型和具体类型 | 设置参数大类和具体类型： ◦ 基本类型包括String、Int、Long等 ◦ 复杂类型包括Object、Map类型。 |
| 必选 | 若参数为必填参数，单击打开必选开关；若参数为可选填参数，需要关闭必填开关，默认为关闭状态。 |
| 长度 | 参数类型为Number或者String类型时，设置参数最大取值长度。 |

| 参数 | 说明 |
|------|--|
| 示例 | <p>参数示例应满足真实使用需求，不得杜撰，不得违反内容安全规定。若涉及数据安全或信息泄露需要脱敏，后四位用*替换。</p> <div style="background-color: #e1f5fe; padding: 10px; border-radius: 5px;"><p>② 说明</p><ul style="list-style-type: none">◦ 银行卡：6004*****1234◦ 邮箱：alice@example.com◦ IP地址：192.168.XX.XX</div> |
| 描述 | <p>设置参数说明：</p> <ul style="list-style-type: none">◦ 对于不易理解的参数，须写清楚参数来源和注意事项。◦ 对于Boolean类型的参数必须说明True和False的取值意义，并提供默认值。◦ 对于枚举类型的参数必须列举全部枚举值并说明每个枚举值的意义。◦ 如果该入参是需要通过其他接口获取，必须提供接口使用说明链接。 |
| 文档顺序 | 设置参数在对应文档中的显示顺序。 |
| 添加属性 | 参数类型为Map、Object时，需要添加子属性，在子属性中设置Map、Object中的具体参数信息。 |
| 删除 | 单击删除删除参数。 |

服务接入协议为HTTP或者HTTPS，API参数配置请参见[API参数配置](#)。

9. 完成参数配置后，单击下一步创建API。

API创建完成后，需要在联调环境对API进行调试，确认API的调用结果是否符合预期，请参见[调试API接口](#)。

3.3.3. RPC网关API参数配置

3.3.3.1. HTTP和HTTPS协议

GET

服务接入协议为HTTP或者HTTPS，请求方法为GET时，支持以下三种传参方式：

- query透传

网关将请求的query参数原样透传至下游服务。

例如用户请求网关为 `http://网关地址?a=1&amp;b=2` 时，网关通过 `http://下游服务地址?a=1&amp;b=2` 方式将参数透传至下游服务。

- 包装data透传

网关将请求的query参数中key为data的值取出，转化为kv形式的query参数传递给下游服务。

例如用户请求网关为 `http://网关地址?data={"a":1,"b":2}` 时，网关通过 `http://下游服务地址?a=1&amp;b=2` 将参数传递至下游服务。

- 自定义脚本

网关将请求参数解析到脚本apply方法中，根据apply结果请求下游服务。

```
package com.alipay.gateway.upstream.groovy;
import java.util.Map;
import java.util.HashMap;
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONObject;
public class TestGroovyTemplate extends GroovyScriptTemplate {
    @Override
    protected Map<String, Object> apply(Object requestMessage, Map<String, Object> context) {
        HashMap<String, Object> innerRequest = (HashMap<String, Object>) requestMessage;
        JSONObject request = new JSONObject();
        try {
            request.put("query", JSONObject.parseObject(innerRequest.get("data")));
        } catch (Exception e) {
        }
        return request.toJavaObject(Map.class);
    }
}
```

② 说明

- apply方法中的requestMessage变量为KV格式的请求网关query参数。
- 需要在request map中填充query、body、header值，这三个值分别对应网关请求下游服务Http协议中的query参数、body参数、header参数。

POST

服务接入协议为HTTP或者HTTPS，请求方法为POST时，支持以下三种传参方式：

- query透传或者包装data透传

请求方法为POST时，参数配置在body中，参数传递方式为query透传或者包装data透传。

- 自定义脚本

同GET请求中自定义脚本传参方式，但requestMessage变量来源于网关请求中的body参数。

3.3.3.2. HSF协议

服务接入协议为HSF时，支持以下两种传参方式：

- query透传

- 将对网关请求的query参数、body参数转化为HSF协议的参数传递至下游服务。
- 同HTTP协议中的自定义脚本传参方式，但request Message变量中的key变为method和context，分别对应HSF协议的参数和上下文。

3.3.4. 调试API

API创建成功后，需要在联调环境对API功能进行测试，当调试结果符合预期时，将API发布上线，否则需要重新编辑并调试API直至符合预期。

操作步骤

- 登录开发商工作台。
- 单击顶部菜单栏的RPC接入，然后单击API管理。



- 在API管理页面的搜索区域输入API名称、选择所属系统、API状态设置为调试中，单击查询快速查找目标API。
- 单击目标API右侧的调试。
- 在API调试页面，配置测试数据参数，单击调试，测试结果展示区域实时显示测试结果。
测试参数包括调用来源、测试环境、请求参数。

3.3.5. 上线API

在联调环境调试通过的API，需要手动发布上线。

操作步骤

- 登录开发商工作台。
- 单击顶部菜单栏的RPC接入，然后单击左侧导航栏API管理。



3. 在API管理页面的搜索区域输入API名称、选择所属系统、AP状态设置为调试中，单击查询快速查找目标API。
4. 单击目标API右侧的上线上线API。

3.3.6. API权限和流量控制

API创建成功后，可配置API允许访问的权限和限流限制条件。

1. 登录开发商工作台。
2. 单击顶部菜单栏的RPC接入，然后单击左侧导航栏API管理。



3. 在API管理页面的搜索区域输入API名称、选择所属系统、设置API状态，单击查询快速查找目标API。
4. 单击目标API右侧的API控制。
5. 在API控制页面，设置允许访问入口、API是否需要登录和限流等参数，单击保存。

The screenshot shows the "API控制" (API Control) configuration dialog box. It contains three sections:

- "* 允许访问入口": A dropdown menu showing "APP" (unchecked) and "H5" (checked).
- "* API是否需要登录": A radio button group where "否" (No) is selected.
- "* 限流": A text input field containing "1000" and a dropdown menu showing "TPS".

- 允许访问入口：限制API允许接入的应用类型，本示例中API只允许H5应用接入。

- API是否需要登录：限制调用API时是否需要输入账号和密码登录，功能即将上线。
- 限流：限制API的调用流量，例如1000 TPS表示每秒调用API的次数不能超过1000次。

3.3.7. 升级API

API上线后，可通过升级API操作修改API配置。

操作步骤

1. 登录**开发商工作台**。
2. 单击顶部菜单栏的**RPC接入**，然后单击左侧导航栏**API管理**。



3. 在**API管理**页面的搜索区域输入API名称、选择所属系统、API状态设置为**调试中**，单击**查询**快速查找目标API。
4. 单击目标API右侧的**升级**生成一个新版本。
5. 单击待升级API右侧的**编辑**。
6. 在**设置API基本信息**页签下，配置API基本信息。

| 参数 | 说明 |
|-----------|--|
| API名称 | 设置API的名称，由英文字母、点号(.) 和阿拉伯数字组成，格式为 <code>mgo p.[公司名称].[系统名称].[操作名称]</code> ，例如 <code>mgop.alibaba.huarong.do Test</code> 。 <div style="background-color: #e0f2ff; padding: 5px; border-radius: 5px; margin-top: 10px;">② 说明 自API名称首次设置完成后，不支持修改，请谨慎操作。</div> |
| 中文名称 | 设置API中文名称。 |
| API功能描述 | 为便于后续管理，为API添加一段功能描述。 |
| API从属系统 | 从下拉列表框中选择API从属系统。 |
| 允许访问入口 | 允许访问入口选择APP或者H5，本示例选择H5。 |
| API是否需要登录 | 即将上线。 |

7. 完成基本信息配置后，单击**下一步**。

8. 在设置API基本信息页签下，配置API基本信息。

| 参数 | 说明 |
|-----------|---|
| API名称 | 设置API的名称，由英文字母、点号(.)和阿拉伯数字组成，格式为 <code>mgo p.[公司名称].[系统名称].[操作名称]</code> ，例如 <code>mgop.alibaba.huarong.do Test</code> 。 <div style="background-color: #e1f5fe; padding: 5px; margin-top: 10px;"> 说明 自API名称首次设置完成后，不支持修改，请谨慎操作。 </div> |
| 中文名称 | 设置API中文名称。 |
| API功能描述 | 为便于后续管理，为API添加一段功能描述。 |
| API从属系统 | 从下拉列表框中选择API从属系统。 |
| 允许访问入口 | 允许访问入口选择APP或者H5，本示例选择H5。 |
| API是否需要登录 | 即将上线。 |

9. 完成服务信息配置后，单击下一步。

10. 在填写参数映射关系页签下，配置的API的入参和出参。

Query 入参

| 参数名称 | 参数类型 | 具体类型 | 必选 | 长度 | 示例 | 描述 | 文档顺序 | 操作 |
|----------------------|------|------|-------------------------------------|----------------------|----------------------|----------------------|----------------------|---|
| <input type="text"/> | 请选择 | 请选择 | <input checked="" type="checkbox"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | 添加属性 删除 |

[+ 添加参数](#)

Body 入参 application/x-www-form-urlencoded

| 参数名称 | 参数类型 | 具体类型 | 必选 | 长度 | 示例 | 描述 | 文档顺序 | 操作 |
|----------------------|------|------|-------------------------------------|----------------------|----------------------|----------------------|----------------------|---|
| <input type="text"/> | 请选择 | 请选择 | <input checked="" type="checkbox"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | 添加属性 删除 |

[+ 添加参数](#)

Body 出参

| 参数名称 | 参数类型 | 具体类型 | 长度 | 示例 | 描述 | 文档顺序 | 操作 |
|------|------|------|----|----|----|------|----|
| 没有数据 | | | | | | | |

[+ 添加参数](#)

| 参数 | 说明 |
|-----------|---|
| 参数名 | 设置API的参数名。 |
| 参数类型和具体类型 | 设置参数大类和具体类型： ◦ 基本类型包括String、Int、Long等 ◦ 复杂类型包括Object、Map类型。 |
| 必选 | 若参数为必填参数，单击打开必选开关；若参数为可选填参数，需要关闭必填开关，默认为关闭状态。 |
| 长度 | 参数类型为Number或者String类型时，设置参数最大取值长度。 |

| 参数 | 说明 |
|------|--|
| 示例 | <p>参数示例应满足真实使用需求，不得杜撰，不得违反内容安全规定。</p> <p>若涉及数据安全或信息泄露需要脱敏，后四位用*替换。</p> <div style="background-color: #e1f5fe; padding: 10px; border-radius: 5px;"><p>② 说明</p><ul style="list-style-type: none">◦ 银行卡：6004*****1234◦ 邮箱：alice@example.com◦ IP地址：192.168.XX.XX</div> |
| 描述 | <p>设置参数说明：</p> <ul style="list-style-type: none">◦ 对于不易理解的参数，必须写清楚参数来源和注意事项。◦ 对于Boolean类型的参数必须说明true和false的取值意义，并提供默认值。◦ 对于枚举类型的参数必须列举全部枚举值并说明每个枚举值的意义。◦ 如果该入参是需要通过其他接口获取，必须提供接口使用说明链接。 |
| 文档顺序 | 设置参数在对应文档中的显示顺序。 |
| 添加属性 | 参数类型为Map、Object时，需要添加子属性，在子属性中设置Map、Object中的具体参数信息。 |
| 删除 | 单击删除删除参数。 |

服务接入协议为HTTP或者HTTPS，API参数配置请参见[API参数配置](#)。

11. 完成参数映射配置后，单击下一步更新API。

API更新完成后，需要在联调环境对API进行调试，确认API的调用结果是否符合预期，请参见[调试API接口](#)。

3.4. 服务接入

3.4.1. 申请添加服务

API发布上线后，可以将API添加到您的应用中，在应用中使用该API。

1. 登录**开发商工作台**。
2. 单击左侧导航栏**发布管理**。



3. 根据业务需要，单击左侧导航栏服务侧名称列表或者治理侧名称列表。
4. 应用状态设置为部署准备中，选择应用类型，输入应用关键字，单击查询。
5. 单击目标应用右侧的去接入。
6. 单击申请添加服务，在申请添加服务页面选择服务，然后单击提交申请，由开发商工作台运维人员在我的任务中完成服务接入申请审核。

3.4.2. API申请

若其他应用申请使用了您发布上线的API，可通过API申请查看申请详情，包含已申请通过的API、正在审核中的API以及审核未通过的API。

操作步骤

1. 登录开发商工作台。
2. 单击顶部菜单栏RPC接入，然后单击左侧导航栏的API申请。



3. 在我的API页签下，输入API名称关键字，单击查询，查询API申请详情。
4. 在API申请列表页签，选择系统或者输入API名称关键字，单击查询，查询API申请详情。

3.5. 设置跨域白名单

应用在开发阶段部署时需要与本地服务器进行开发联调时，需要将待访问的本地域名或IP地址添加到API网关白名单中，然后才能调用mgop接口进行跨域访问。本文介绍如何为应用设置跨域访问白名单。

操作步骤

1. 登录**开发商工作台**。
2. 单击左侧导航栏**发布管理**。



3. 根据业务需要，单击左侧导航栏**服务侧名称列表**或者**治理侧名称列表**。
4. 在搜索区域通过关键字快速搜索目标服务，然后单击**查询**。
5. 单击目标服务右侧的**更多>配置**，在**服务侧名称配置**页面进行以下操作：
 - **获取应用APPKEY**：单击打开配置开关，则调用mgop接口时，需要填写APPKEY；关闭开关则调用mgop接口时，无需填写APPKEY。
 - **API网关关联调用白名单配置**：服务在开发阶段部署与本地服务器进行开发联调时，需要将要访问的本地域名或IP地址（包含端口号）添加到API网关白名单中，然后才能调用mgop接口进行跨域访问，多个域名或者IP地址之间以换行符分隔。
 - **API网关关联Header白名单配置**：调用mgop接口进行跨域访问时，需要填写服务自定义的HTTP header参数，多个参数之间以换行符分隔。
 - **WEB网关跨域白名单配置**：如需跨域访问应用，需要将访问域名加入WEB网关跨域白名单中。



6. 完成上述参数配置后单击修改。

3.6. H5应用接入

开发商工作台支持基于类React语法的Rax应用和自定义两种方式完成应用代码开发。本文以Windows 10操作系统为例介绍基于React语法进行应用开发。

环境准备

- 安装node.js

- i. 登录nodejs官网[下载Node.js](#)，本文以Windows为例演示安装（.msi）64位的nodejs，单击Windows安装包64位。

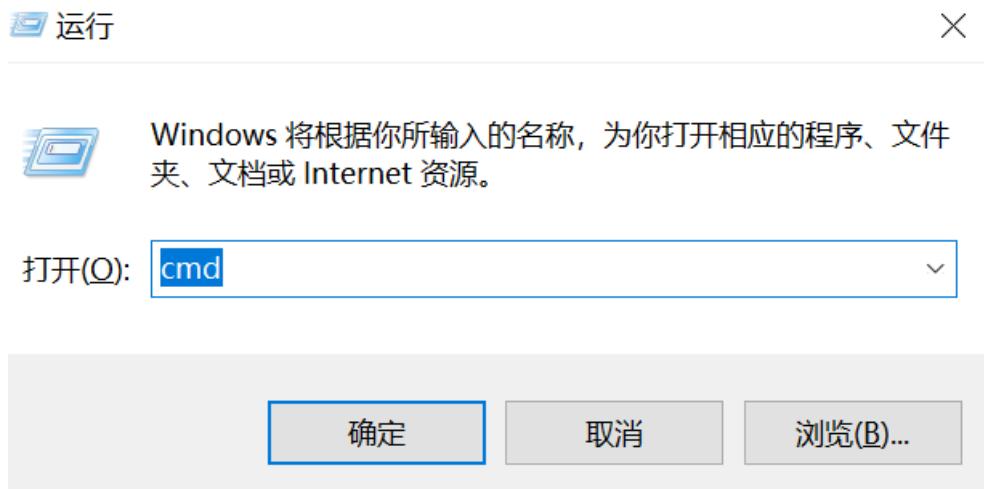
The screenshot shows the Node.js download page. At the top, there's a navigation bar with links: 首页 | 下载 | 入门教程 | API文档 | 云服务器. Below the navigation bar, it says "当前版本: 14.15.0". There are three main download options: "Windows 安装包" (Windows Installer), "macOS 安装包" (macOS Installer), and "源代码" (Source Code). Under the "Windows 安装包" section, there are four options: "Windows 安装包 (.msi)" (32位, 64位), "Windows 二进制文件 (.zip)" (32位, 64位), "macOS 安装包 (.pkg)" (64位), and "macOS 二进制文件 (.tar.gz)" (64位). A red arrow points to the "64位" link under the ".msi" option.

- ii. 下载完成后，双击安装包。



- iii. 选择安装路径，本示例为C:\Program Files\nodejs，安装完成后单击Finish完成安装。

- iv. 打开命令提示符窗口，输入 cmd。



- v. 执行 npm -v 命令，查看安装后的Node.js版本，如果输出版本号，则表示安装成功，如下图所示。



● 配置环境变量

以下以Windows 10操作系统为例介绍如何更改环境配置。

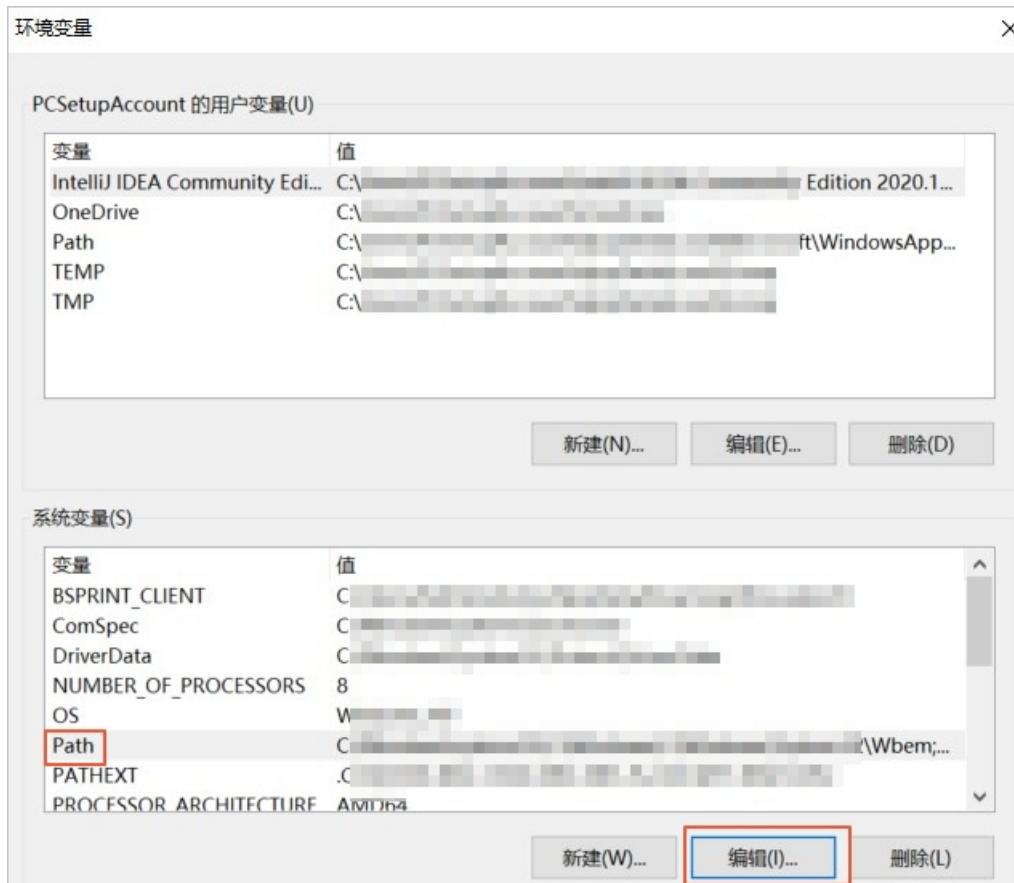
- i. 单击本地电脑控制面板>系统和安全>系统>高级系统设置。



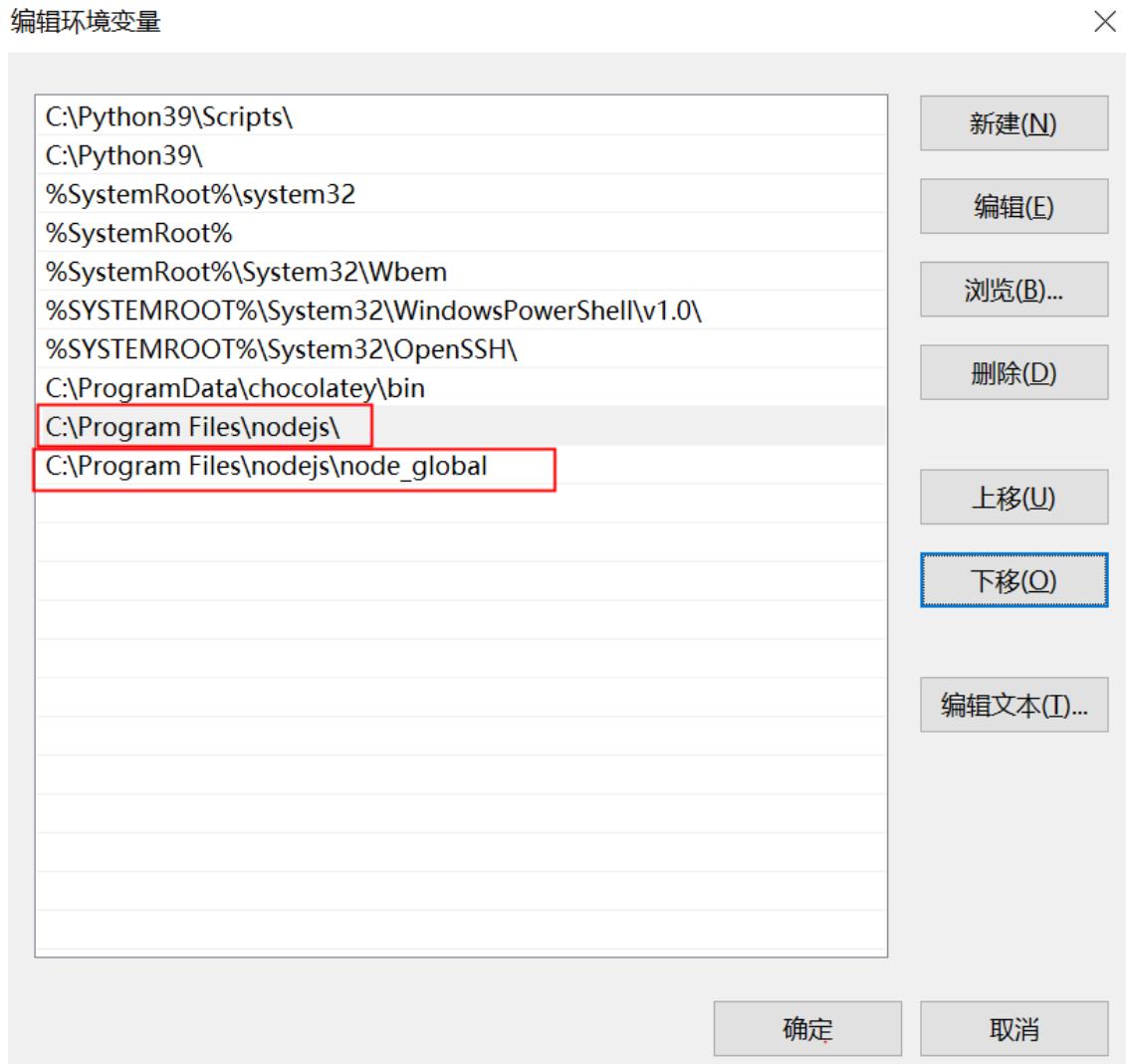
- ii. 在高级系统设置页面单击高级>环境变量 (N) ...。



- iii. 在环境变量页面选中系统变量 (S) 框中的Path，单击编辑。



- iv. 在编辑环境变量页面单击新建 (N) , 添加安装目录C:\Program Files\nodejs和C:\Program Files\nodejs\node_global, 添加完成后单击确定。



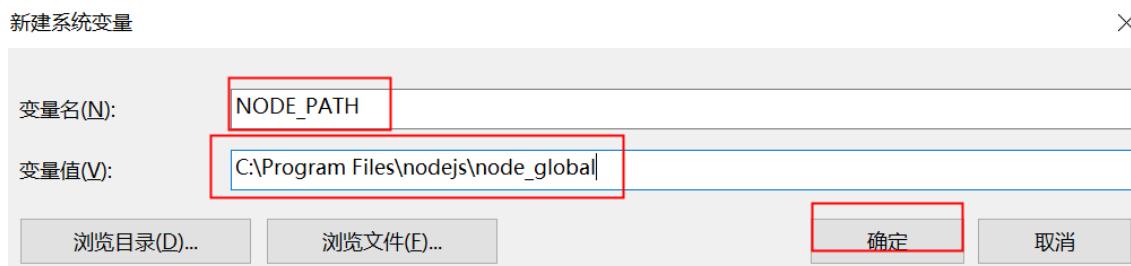
- v. 在命令窗口输入以下命令，完成npm全局模块的存放路径、cache的路径。

```
npm config set prefix "C:\Program Files\nodejs\node_global"
```

```
npm config set cache "C:\Program Files\nodejs\node_cache"
```

- vi. 在环境变量页面单击新建，在新建系统变量页面按照页面提示完成参数配置，参数配置完成后单击确定，参数说明如下：

- 变量名 (N) : NODE_PATH
- 变量值 (V) : C:\Program Files\nodejs\node_global



vii. 在命令提示符窗口或者管理员命令提示符窗口输入 `npm install express -g` 安装express。

 注意

-g 表示安装到global目录下，就是上面设置的node_global下。

```
C:\Users\innfotech\Desktop>npm install express -g
+ express@4.16.4
added 48 packages from 36 contributors in 10.575s
```

 管理员: Windows PowerShell

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。
尝试新的跨平台 PowerShell https://aka.ms/pscore6
PS C:\Windows\system32> npm install express -g
+ express@4.17.1
added 50 packages from 37 contributors in 6.91s
PS C:\Windows\system32>
```

viii. 在命令提示符窗口或者管理员命令提示符窗口输入 `node`，回车后，再输入 `require('express')`，出现如下报错页面：

```
> require('express')
Uncaught Error: Cannot find module 'express'
Require stack:
- <repl>
  at Function.Module._resolveFilename (internal/modules/cjs/loader.js:880:15)
  at Function.Module._load (internal/modules/cjs/loader.js:725:27)
  at Module.require (internal/modules/cjs/loader.js:952:19)
  at require (internal/modules/cjs/helpers.js:88:18) {
    code: 'MODULE_NOT_FOUND',
    requireStack: [ '<repl>' ]
}
```

ix. 在node命令行输入如下命令。

```
require('C:/Program Files/nodejs/node_global/node_modules/express')
```

● 安装cnpm

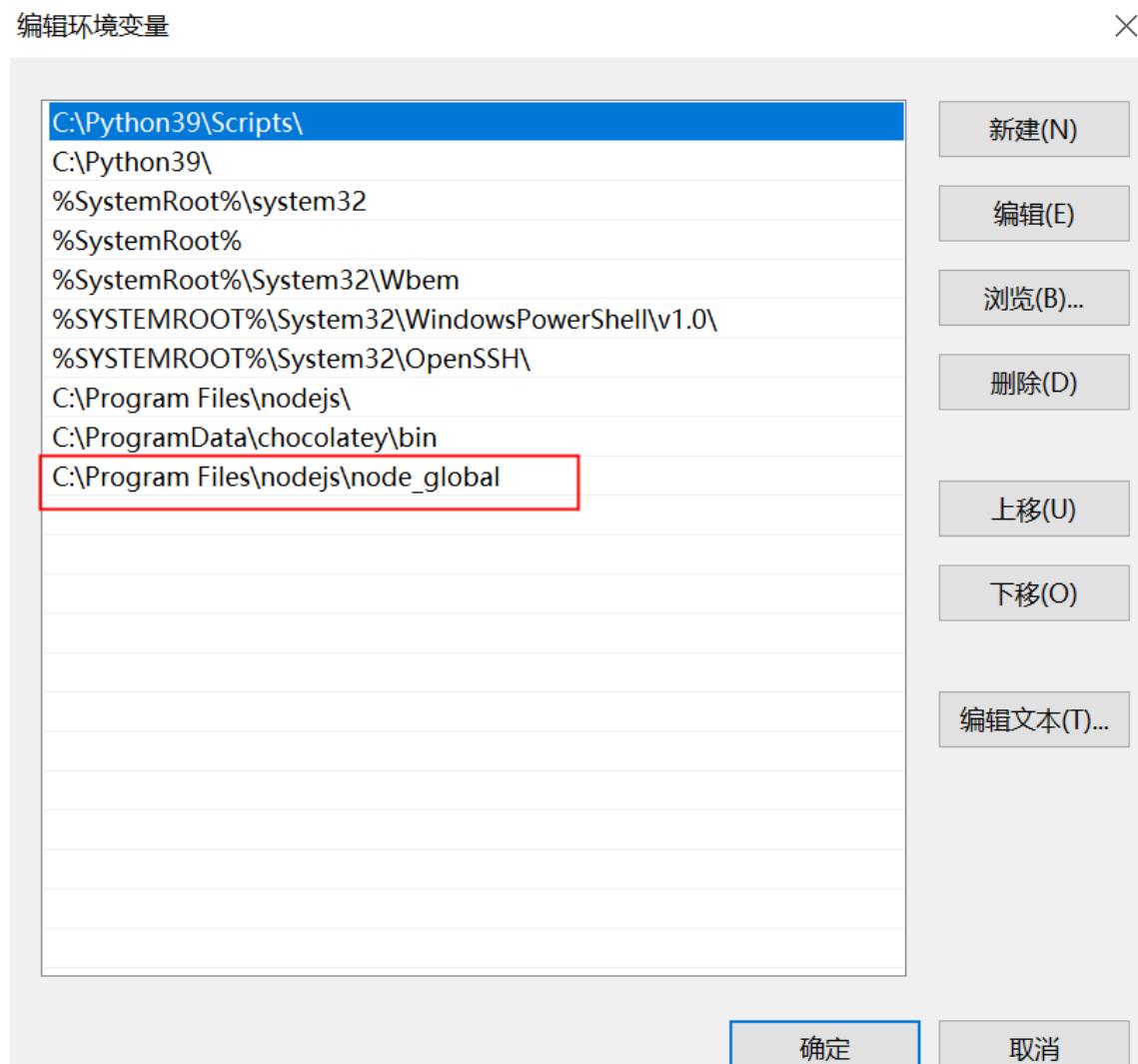
因为npm安装插件是从国外服务器下载，受网络的影响比较大，可能会出现异常，所以需要安装cnpm，cnpm是一个完整npmjs.org镜像，您可以用此代替官方版本(只读)，同步频率目前为10分钟一次以保证尽量与官方服务同步。

i. 在命令提示符窗口输入如下命令。

```
npm install -g cnpm --registry=https://registry.npm.taobao.org
```

```
命令提示符  
C:\Users\PCSetupAccount>cnpm -v  
'cnpm' 不是内部或外部命令，也不是可运行的程序  
或批处理文件。  
C:\Users\PCSetupAccount>npm install -g cnpm --registry=https://registry.npm.taobao.org  
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142  
npm WARN deprecated har-validator@5.1.5: this library is no longer supported  
C:\Users\PCSetupAccount\Program Files\nodejs\node_global\cnpm -> C:\Users\PCSetupAccount\Program Files\nodejs\node_global\node_modules\cnpm\bin\cnpm  
+ cnpm@6.1.1  
added 689 packages from 974 contributors in 31.785s
```

- ii. 在命令提示符窗口输入 `cnpm -v`，检查是否安装成功，如果出现如下图报错，参照上页环境配置在系统变量配置Path中添加 `cnpm` 安装目录。



- iii. 在命令提示符窗口输入 `cnpm -v`，出现如下图所示，表示 `cnpm` 安装成功。

基于Rax开发应用

1. 在命令提示符窗口或者管理员命令提示符窗口输入：`npm i file -g`。

```
PS C:\Windows\system32> npm i fie -g
npm WARN request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN har-validator@5.1.5: this library is no longer supported
C:\Program Files\nodejs\node_global\fie -> C:\Program Files\nodejs\node_global\node_modules\fie\bin\fie
> spawn-sync@1.0.15 postinstall C:\Program Files\nodejs\node_global\node_modules\fie\node_modules\spawn-sync
> node postinstall
+ fie@3.1.6
added 418 packages from 259 contributors in 51.69s
PS C:\Windows\system32>
```

2. 在命令提示符窗口输入：`fie init taiyi`, 选择Alibaba Extranet（阿里巴巴合作方用户）,完成初始化项目。

```
C:\Users\Leo>fie init taiyi
[core-core] fie development environment doesn't initialized!
? Please choose the development environment: (Use arrow keys)
> Alibaba Intranet      -    Alibaba employees/VPN users
    Alibaba Extranet      -    ISV/No access to intranet users
```

- ### 3. 选择Alibaba Extranet（阿里巴巴合作方用户）。

```
[core-commands] Init environment succeed!  
[core-core] For switching enviroment please use $ fie switch!
```

注意

若误选为Alibaba Intranet，可导致工具不可用，需要执行 `fie clear` 恢复出厂模式，然后重新选择 Alibaba Extranet。

4. 在命令提示符窗口输入： file init taiyi ， 提示目录已存在， 输入： y 。

```
C:\Users\Leo>fie init taiyi
'tnpm' 似乎是 npm 的别名
[core-commands] 当前目录下已存在文件, 继续执行初始化会覆盖已存在的同名文件
? 确认需要继续执行初始化, 请输入(y) v
```

5. 命令提示符窗口，提示安装成功。

```
[core-module] 本地尚未安装 fie-toolkit-taiyi , 正在执行自动安装...
✓ Installed 1 packages
✓ Linked 370 latest versions
✓ Run 0 scripts
deprecate fie-toolkit-taiyi@1.1.7 > fie-api@3.1.6 > fie-config@3.1.7 > fie-report@3.1.4 > request@^2.79.0 request has been deprecated, see https://github.com/request/request/issues/3142
deprecate fie-toolkit-taiyi@1.1.7 > fie-api@3.1.6 > fie-config@3.1.7 > fie-report@3.1.4 > request@2.88.2 > har-validator@^5.1.3 this library is no longer supported
Recently updated (since 2020-10-30): 4 packages (detail see file C:\Users\Leo\.fie\node_modules\.recently_updates.txt)
✓ All packages installed (424 packages installed from npm registry, used 7s(network 7s), speed 1.02MB/s, json 371(723.0 kB), tarball 6.45MB)
Leo@LEO-OptiPlex-5090 安装成功
```

- ## 6. 项目类型选择H5。

7. 输入项目模板，H5应用接入应选择 @aligov/h5-rax-scaffold 项目模板。

8. 为项目取一个名称，项目名称由小写字母、数字、中划线以及下划线组成，例如my_app。

? 项目名称(小写字母/数字/中划线/下划线) : leo_nodeapp

9. 在命令提示符窗口输入以下命令进行应用开发。

npm start

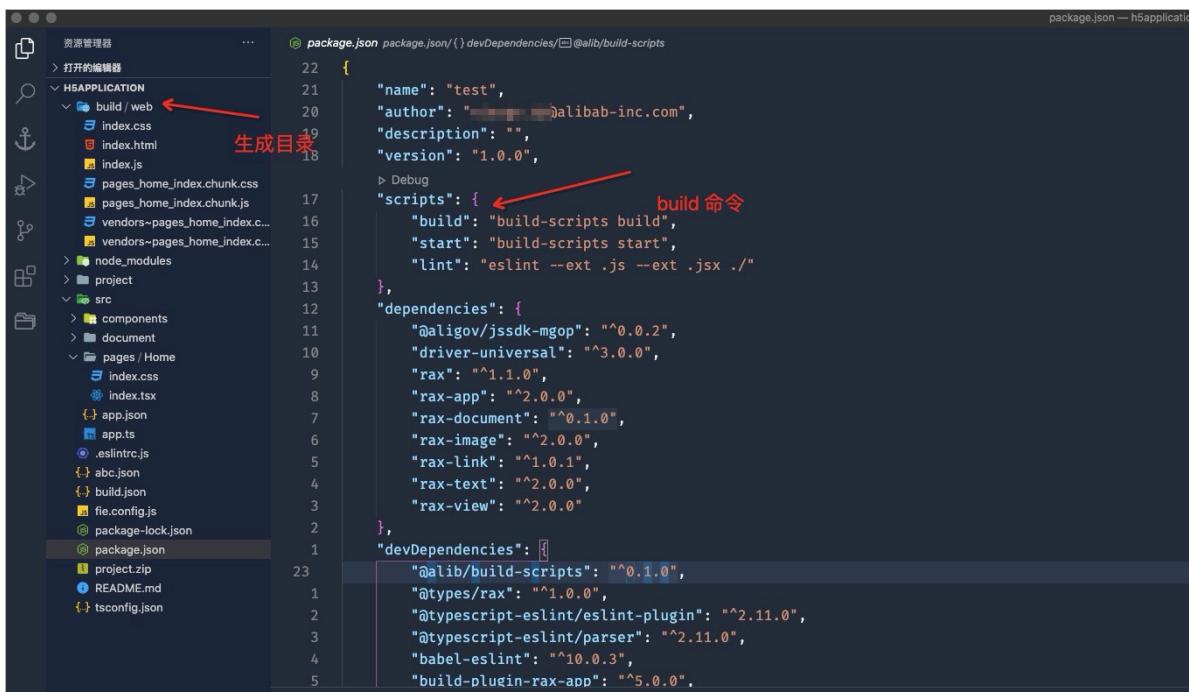
10. 执行 `fie pack` 命令，在目录下得到一个 `src.zip` 压缩包，在应用发布页面上传该代码包。

Rax官方文档请参见[Rax](#)，组件库请参见[Alibaba Fusion Design](#)。

自定义应用开发说明

开发者可以使用自定义框架构建前端应用，需遵循以下约定。

- 在 `package.json` 文件中的 `script` 标签下添加 `build` 命令，代码上传到应用后，开发商工作台将通过 `build` 命令构建应用。应用构建完成后，需要将生成的文件保存在同级目录的 `build` 目录下。



- 将代码上传至开发商工作台之前，需要手动将除node_modules之外的整个工程压缩为xxx.zip的压缩文件，然后上传至应用，否则应用端无法识别代码，上传成功后应用端执行build命令构建应用。

3.7. 应用调用API

3.7.1. 应用网关接入

开发者通过jssdk-mgop调用API。

操作步骤

1. 执行以下命令下载和安装mgop依赖。

npm i --save @aligov/jssdk-mgop

2. 加载mgop依赖，同时调用mgop。

```
import { mgop } from '@aligov/jssdk-mgop';
mgop({
  api: 'mgop.h5.http.getnews',//必填
  host: 'https://gw-irs.zj.gov.cn',
  dataType: 'JSON',
  type: 'GET',
  appKey: '6xxfslcv+200600801+tlkciqg',//必填
  onSuccess: data =&gt; {
    console.log('data', data)
  },
  onFail: err =&gt; {
    console.log(err, 'err')
  }
});
```

请求参数

| 参数 | 类型 | 是否必填 | 描述 |
|-----------|----------|------|----------------------------|
| api | String | 是 | 请求API。 |
| host | String | 是 | API网关地址。 |
| data | Object | 是 | 请求数据。 |
| appKey | String | 是 | 请求应用标识。 |
| timeout | Number | 否 | API调用超时时间，单位毫秒，默认值为2000毫秒。 |
| onSuccess | Function | 否 | API调用成功时的成功回调函数。 |
| onFail | Function | 否 | API调用失败时的失败回调函数。 |

返回数据

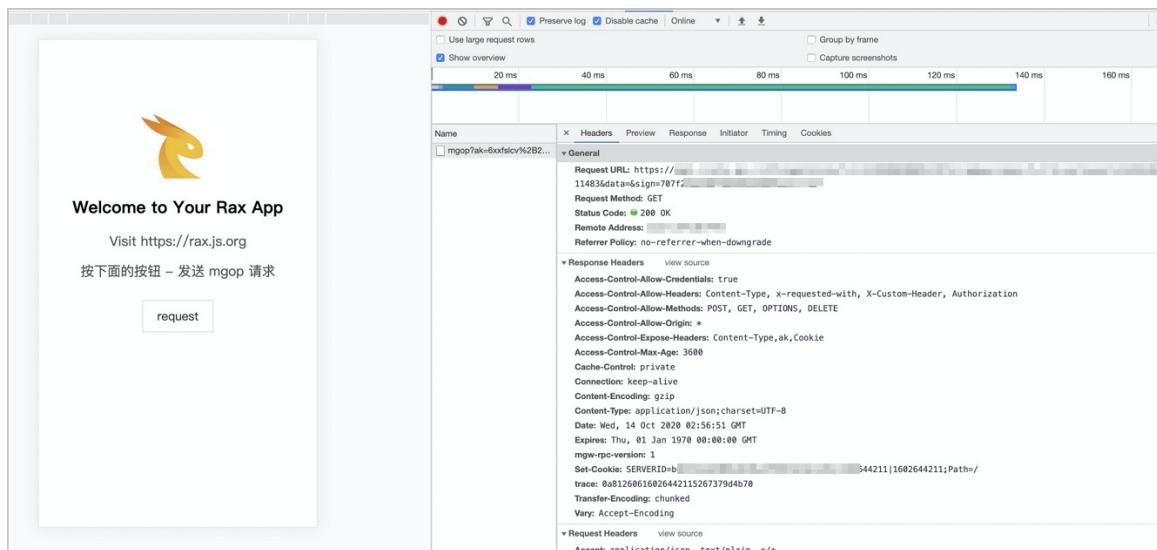
| 参数 | 来源 | 描述 |
|-----|------|--------|
| api | 服务返回 | 请求API。 |

| 参数 | 来源 | 描述 |
|------|------|--|
| ret | 服务返回 | <p>请求结果，例如API调用成功，返回"1000::调用成功"。</p> <p>通用格式为 <code>code::message</code>。</p> <ul style="list-style-type: none"> ◦ code: 响应码。 1000表示成功，其他数值表示失败。 ◦ message: 描述。 |
| data | 服务返回 | 请求返回的业务数据。 |

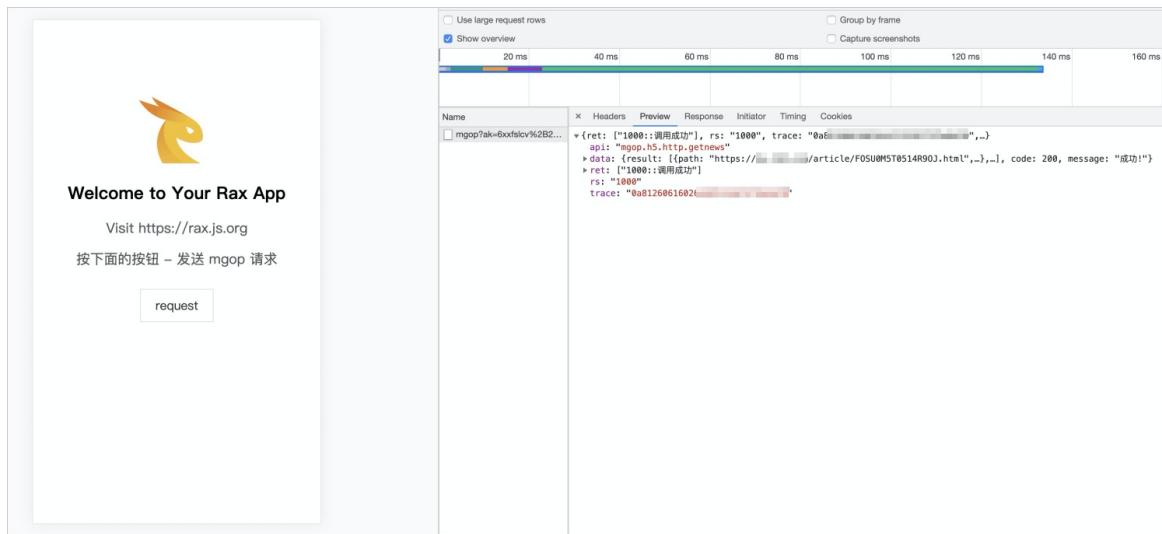
3. 在开发过程中，支持将本地开发环境的域名加入API接口的请求白名单中，进行请求调试，请参见[设置跨域白名单](#)。

例如应用在本地运行时，域名为`http://localhost:3334/`，调试前需要将`localhost:3334`加入API接口的请求白名单中。

发起调试：



调试结果：



3.7.2. JSAPI接入

调用JSAPI需要引入ZWJSBridge，请参见[ZWJSBridge接入方式](#)。

3.8. 应用调试

3.8.1. Debug调试工具

3.8.1.1. 简介

功能简介

Debug调试工具支持日志查询和前端页面调试：

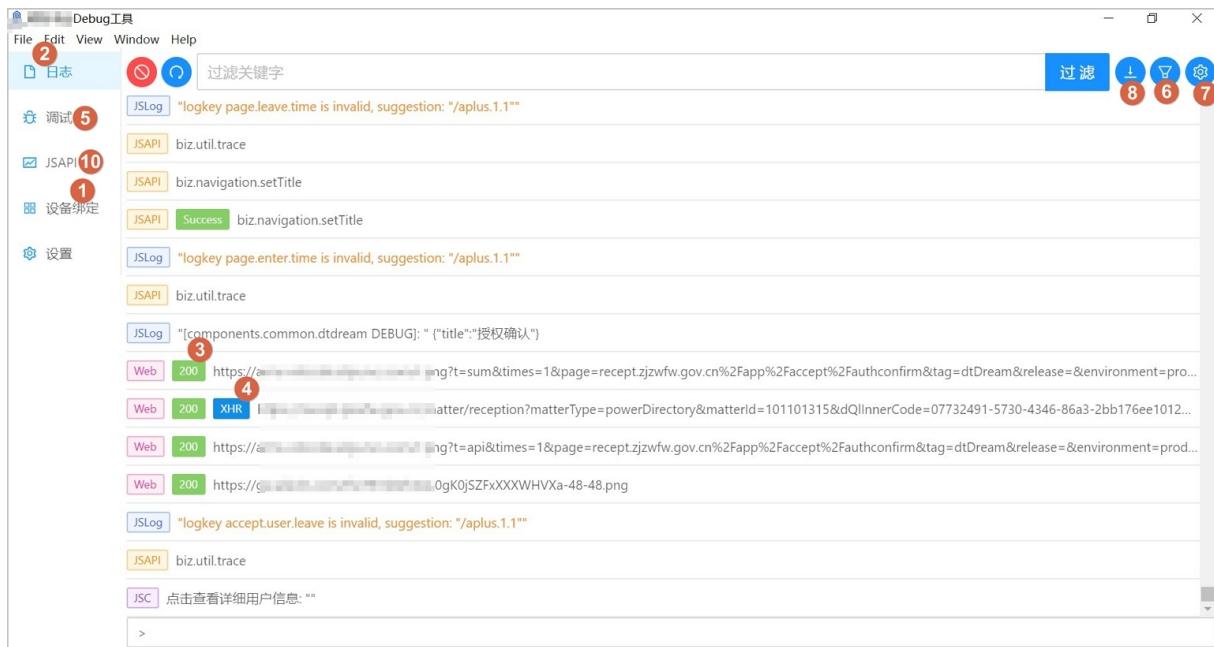
- 日志查询：支持查看前端日志、JSBridge API执行日志、网络日志、Cookie、localStorage等日志，同时可以在调试工具中动态执行JS命令，覆盖一般前端日志工具，例如覆盖Eruda的全部功能，请参见[查看日志](#)。
- 前端页面调试：提供在浙里办APP环境内调试前端页面的能力。

之前的模式中，由于Chrome中不含浙里办APP运行环境，开发者需要在Chrome中调试不含JSBridge API的页面功能，调试完成后加上JSBridge API再在浙里办APP中运行查看效果。

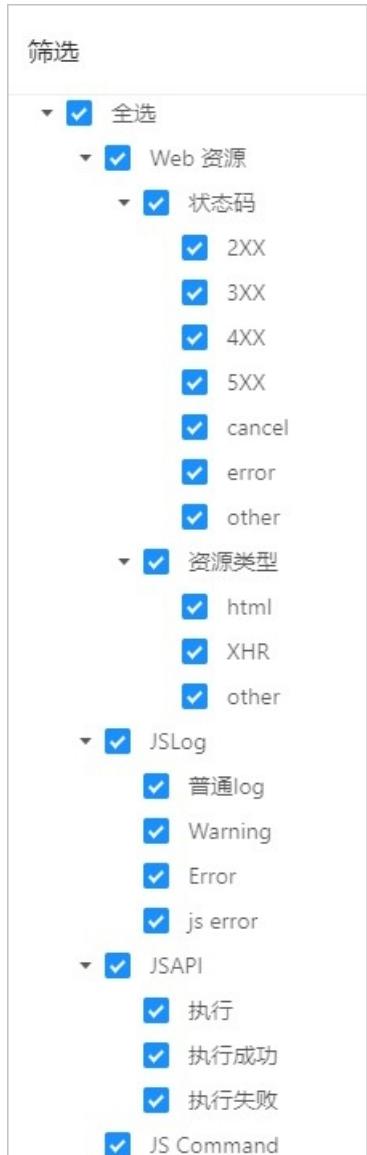
通过Debug调试工具可以直接在浙里办APP环境中进行页面开发，模拟端侧环境运行前端页面，请参见[前端页面调试](#)。

支持在Mac和Windows系统中安装Debug调试工具。

概览



- 1：连接设备，单击打开二维码，通过浙里办APP扫码连接Debug工具。
- 2：红色Tag表示H5请求。
- 3：请求状态码：
 - 200~299为绿色表示请求成功。
 - 300~399为黄色表示将进一步请求。
 - 400及以上为红色表示请求失败。
- 4：蓝色Tag表示请求类型：
 - HTML为链接请求
 - XHR为页面内的网络请求，包含请求的所有信息，排查问题时重点关注XHR。
 - 没有Tag表示除HTML、XHR以外的所有请求，包括CSS、JS、图片、字体等下载请求。
- 5：H5页面调试，请参见[H5页面调试](#)。
- 6：日志筛选按钮，筛选要查看的日志类型。



- 7: 功能按钮

设置

[导出日志](#)[导入日志](#)[清空缓存](#)[显示Cookie](#)[显示localStorage](#)[JSAPI调试](#)

| 功能 | 说明 |
|----------------|---|
| 导出日志 | 将全量日志导出到本地。 |
| 导入日志 | 将本地前端日志导入Debug工具进行问题排查。 |
| 清空缓存 | 清除前端资源（页面JS、CSS、图片等）、Cookie（用户登录信息）。 |
| 显示Cookie | 显示前端页面Cookie。 |
| 显示localStorage | 显示前端页面的localStorage存储信息。 |
| JSAPI调试 | 显示浙里办APP当前页面支持的JSBridge API，支持编辑和调试API。 |

- 8：冻结当前日志页面按钮。
- 9：命令执行框，执行前端页面JS命令。
- 10：JSBridge API调试按钮，无需连接Debug调试工具即可实时调试JSBridge API。

3.8.1.2. 准备工作

使用Debug工具之前，您需要完成以下准备工作：

1. 手机安装浙里办APP，需要通过浙里办APP扫码连接Debug调试工具。
2. 下载并安装Debug工具，下载地址：

- Windows: ZWEurope 3.1.0
- Mac: ZWEurope 3.1.0

3.8.1.3. 查看日志

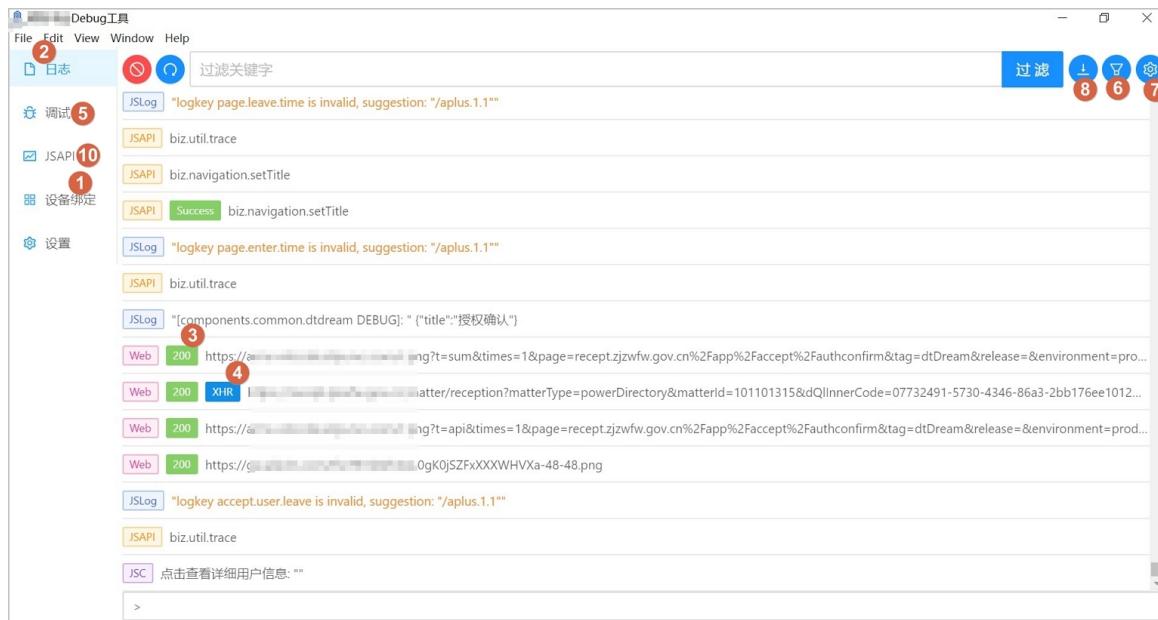
支持查看前端日志、JSBridge API执行日志、网络日志、Cookie、localStorage等日志，同时可以在调试工具中动态执行JS命令，覆盖一般前端日志工具，例如覆盖Eruda的全部功能。

操作步骤

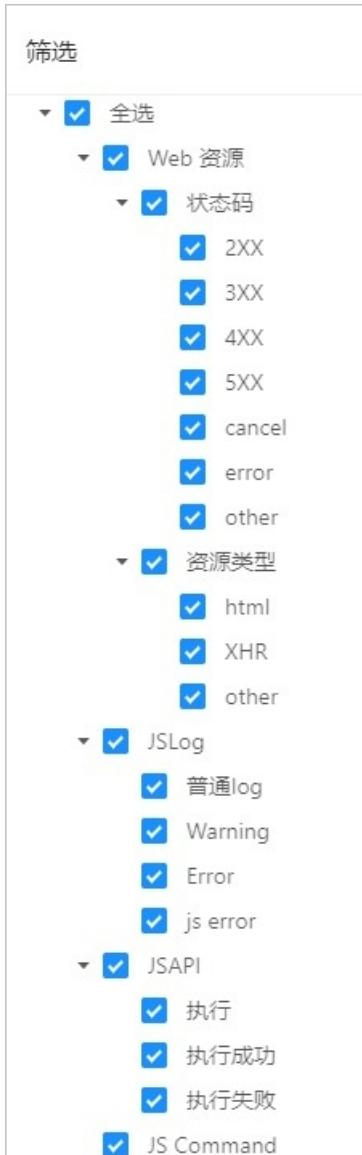
1. 打开Debug调试工具。
2. 单击左侧导航栏设备绑定，打开连接码。
3. 手机登录浙里办APP，打开扫码功能，扫码连接Debug工具。

连接成功后进入查看日志页面，Debug工具自动拉取浙里办APP支持的JSBridge API。





- 1：连接设备，单击打开二维码，通过浙里办APP扫码连接Debug工具。
- 2：红色Tag是h5请求。
- 3：状态码：
 - 200~299为绿色表示请求成功
 - 300~399为黄色表示将进一步请求
 - 400及以上为红色表示请求失败
- 4：蓝色Tag表示请求类型：
 - HTML为链接请求
 - XHR为页面内的网络请求，包含请求的所有信息，排查问题时重点关注XHR。
 - 没有Tag表示除HTML、XHR以外的所有请求，包括CSS、JS、图片、字体等下载请求。
- 5：H5页面调试，请参见[H5页面调试](#)。
- 6：日志筛选按钮，筛选要查看的日志类型。



- o 7: 设置按钮



| 功能 | 说明 |
|----------------|---|
| 导出日志 | 将全量日志导出到本地。 |
| 导入日志 | 将本地前端日志导入Debug工具进行问题排查。 |
| 清空缓存 | 清除前端资源（页面JS、CSS、图片等）、Cookie（用户登录信息）。 |
| 显示Cookie | 显示前端页面Cookie。 |
| 显示localStorage | 显示前端页面的localStorage存储信息。 |
| JSAPI调试 | 显示浙里办APP当前页面支持的JSBridge API，支持编辑和调试API。 |

- 8：冻结日志页面按钮。
 - 9：打开浙里办APP主页时，输入前端页面URL地址（例如http://www.baidu.com），回车跳转到相应页面；进入页面后，可直接执行JS命令（例如location.href = 'http://www.baidu.com'）跳转到相应页面。
 - 10：JSBridge API调试按钮，无需连接Debug调试工具即可实时调试JSBridge API。
4. 在JSBridge API列表中，单击日志列表中的日志查看日志详情。

3.8.1.4. 前端页面调试

提供在浙里办APP环境内调试前端页面的能力。之前的模式中，由于Chrome中不含浙里办APP运行环境，开发者需要在Chrome中调试不含JSBridge API的页面功能，调试完成后加上JSBridge API再在浙里办APP中运行查看效果。通过Debug调试工具可以直接在浙里办APP环境中进行页面开发，模拟端侧环境运行前端页面。

注意事项

浙里办APP 安卓6.6.0、IOS 6.5.0及以上版本支持调试功能。

调试步骤

1. 打开Debug调试工具。
2. 单击左侧导航栏设备绑定，打开连接码。
3. 手机登录浙里办APP，打开扫码功能，扫码连接Debug调试工具。

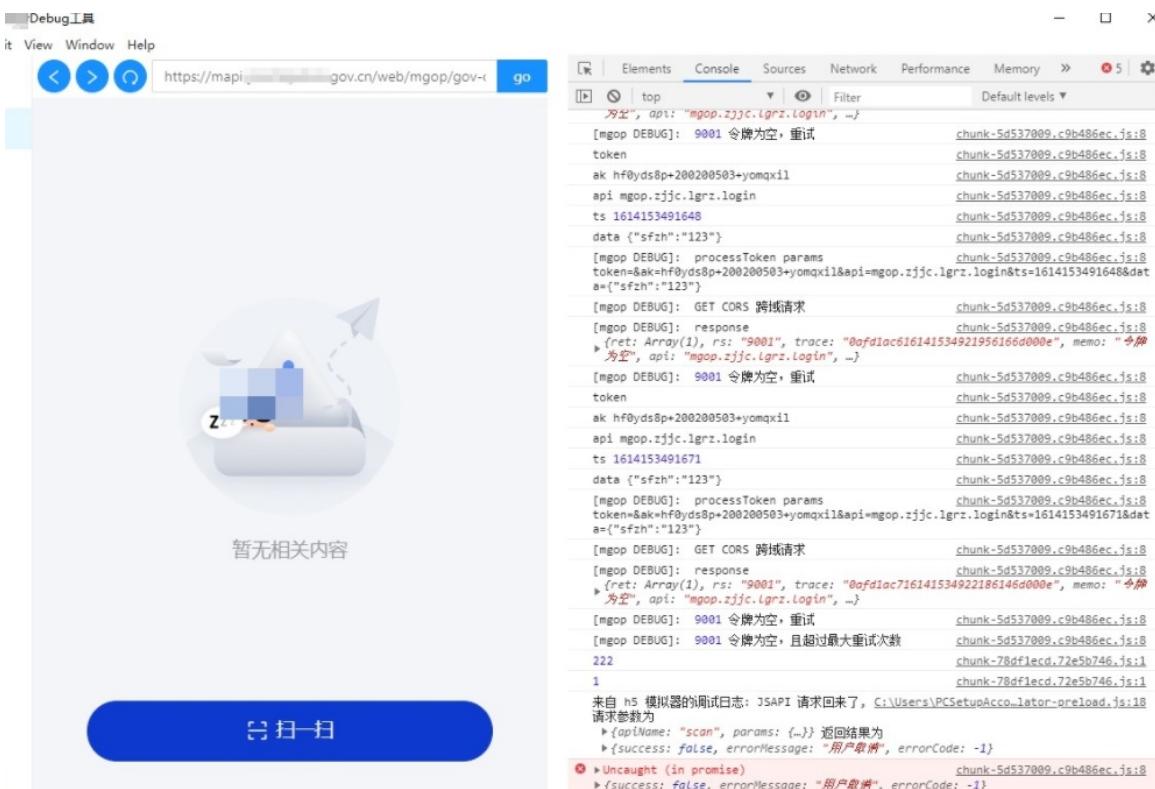


连接成功后进入查看日志页面，Debug调试工具自动拉取浙里办APP支持的JSBridge API。

② 说明

要求手机和Debug调试工具处于同一局域网中，否则无法扫码连接Debug调试工具。

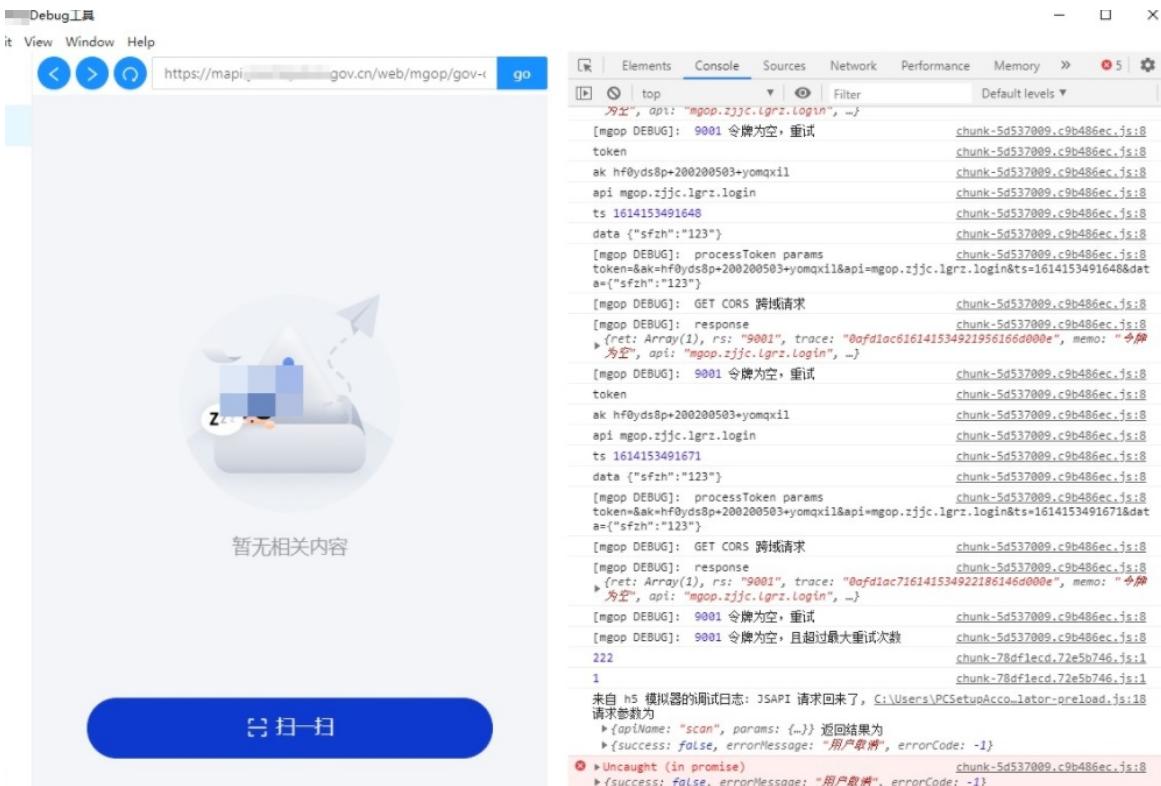
- 单击左侧导航栏调试，在地址框中输入本地应用地址或者应用开发管理平台生成的应用地址，单击go进入调试页面。



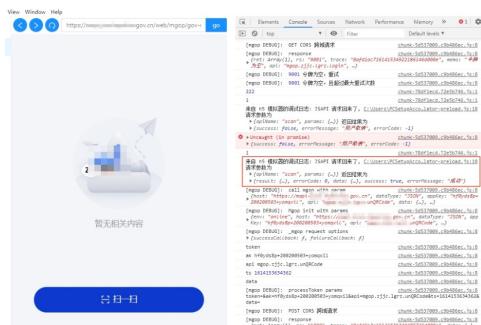
调试页面分为实时页面效果页和代码调试页：

- 1：页面前进按钮。
- 2：页面后退按钮。
- 3：刷新当前页面按钮。
- 4：代码调试页。

本示例以调试scan扫码接口为例，介绍如何使用Debug工具调试JSBridge API。在地址输入框输入待调试应用地址，单击go。



5. 单击扫一扫启用手机扫码功能，扫任一可用的二维码，Debug调试工具显示扫码结果。



3.8.1.5. 调试JSBridge API

下载安装Debug调试工具后，可以在调试工具中查看和调试全量JSBridge API。

前提条件

已下载和安装Debug调试工具，下载地址：

- Windows: [ZWEurope 3.1.0](#)
- Mac: [ZWEurope 3.1.0](#)

操作步骤

1. 打开Debug调试工具。
2. 单击左侧导航栏设备绑定，打开连接码，通过手机浙里办APP扫码连接Debug调试工具。
3. 单击JSAPI按钮，打开调试页面。

4. 在JSAPI调试列表单击目标JSBridge API（例如sms API），右侧调试区域同步显示对应JSBridge API的调用示例。
5. 单击扫码查看打开二维码，手机登录浙里办APP并扫码即可对JSBridge API功能进行调试。

② 说明

支持在调试页面更改JSBridge API入参，然后通过手机扫码实时验证调试结果。

JSScript调试

| | |
|------------------|----------|
| hidePreloader | 隐藏蒙版 |
| openLink | 新开一个页面 |
| chooseImage | 选择图片 |
| zmAuthentication | 实人认证 |
| share | 分享 |
| directShare | 分享 |
| getUUID | 获取设备唯一标识 |
| sms | 发送短信 |
| close | 关闭窗口 |
| toast | 吐司信息 |
| getNetworkType | 获取网络状态 |
| getLocation | 获取地理位置 |

sms 发送短信

```
1 ZWJSBridge.sms({  
2   "text": "你吃饭了么",  
3   "phoneNumber": "16602194414"  
4 }, res => {  
5   console.log(res)  
6 }, err => {  
7   console.log(err)  
8 })
```

扫码查看

中国移动 上午 11:15 78%

Rax App

JSBridge调试界面

API名称为:

sms

调试入参为:

```
{"text": "你吃饭了么", "phoneNumber": "16602194414"}
```

响应结果为:

点击调试



单击点击调试按钮，系统自动跳转到发送短信页面。



3.8.2. H5页面调试

开发者可以通过Debug工具查看H5页面的请求、JS Log、Cookie等，进行H5页面调试。

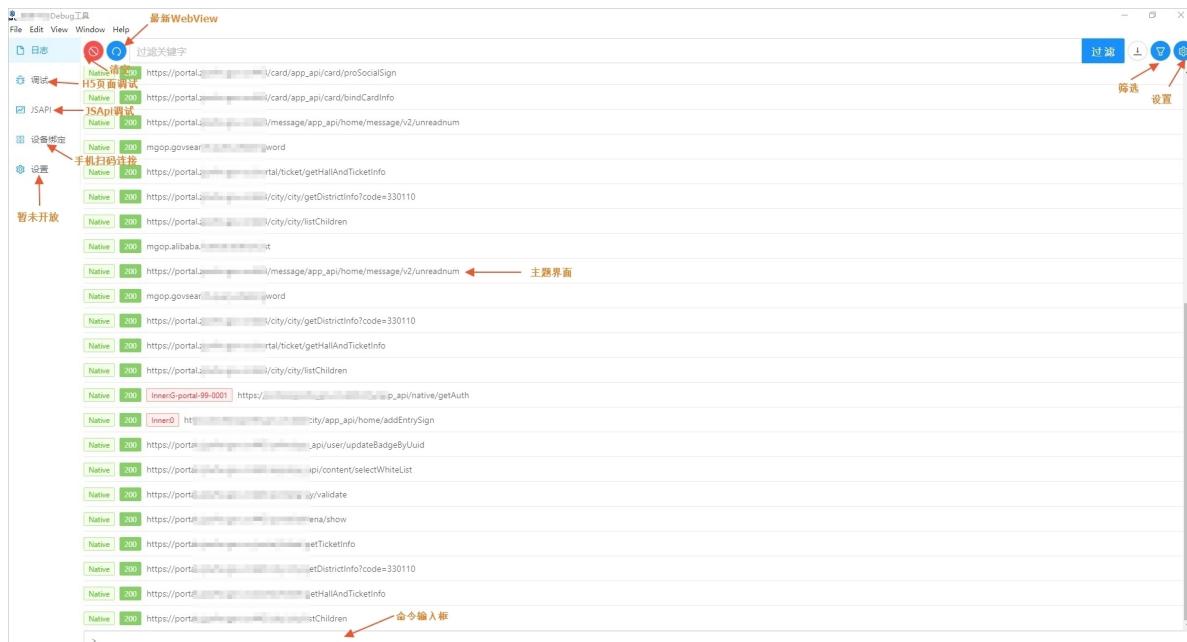
前提条件

已下载和安装Debug调试，下载地址：

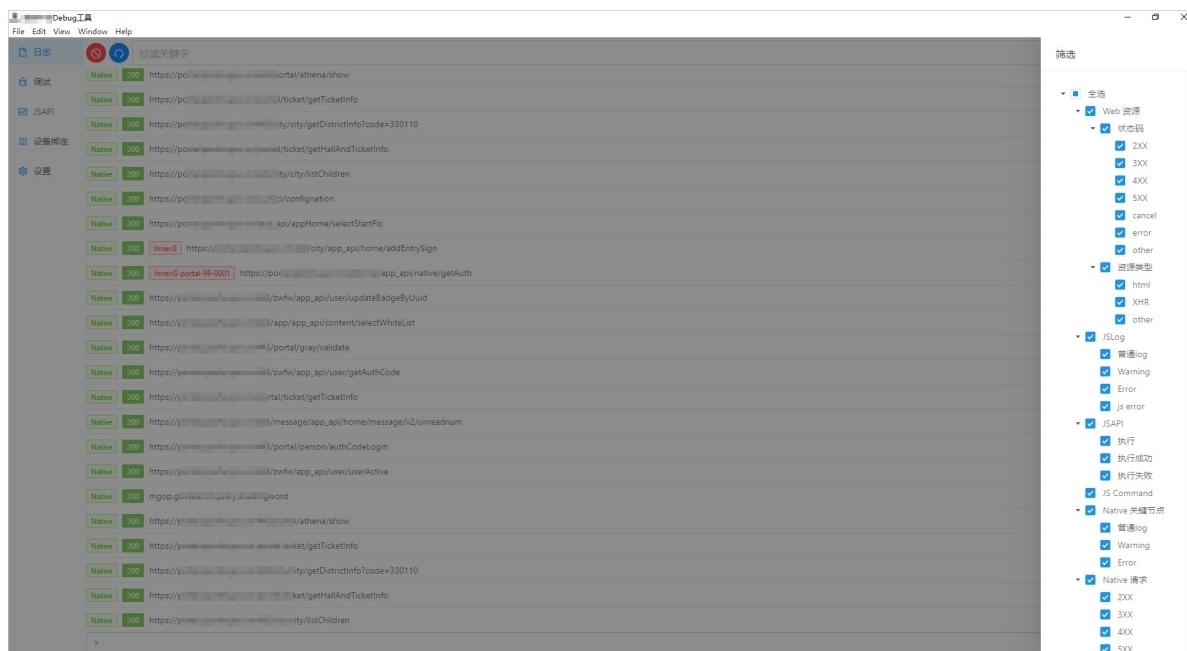
- Windows: [ZWEurope 3.1.0](#)
- Mac: [ZWEurope 3.1.0](#)

Debug工具界面概览

- 基本界面



● 筛选界面



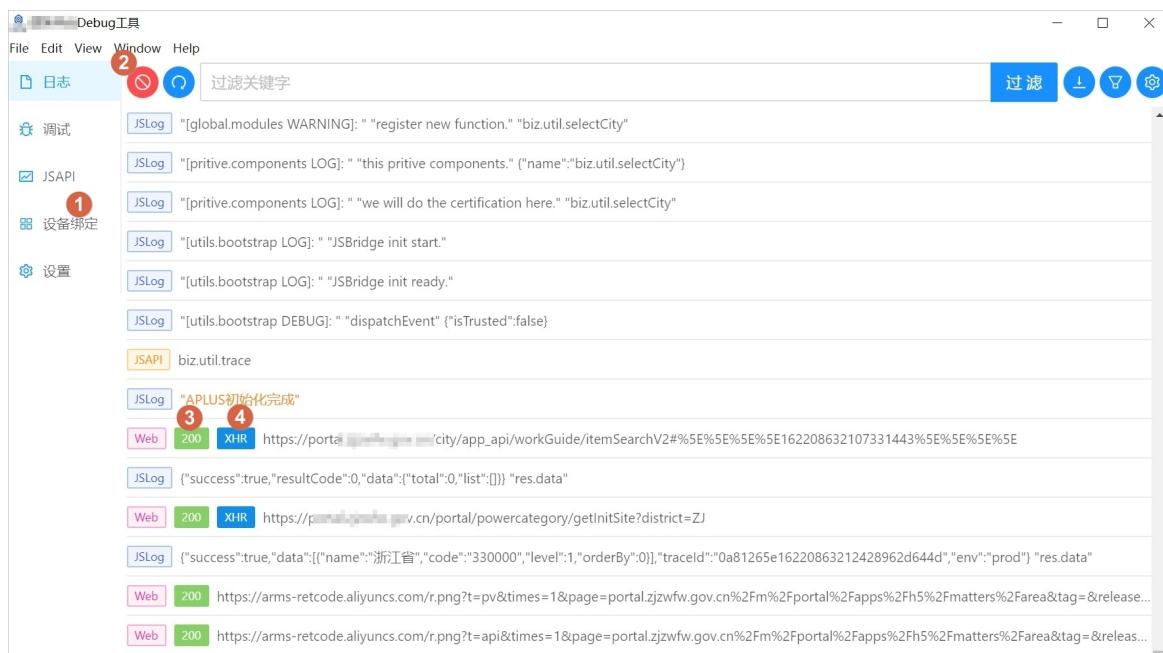
连接Debug工具

1. 打开Debug工具。
2. 单击左侧导航栏中的设备绑定，打开连接码。
3. 通过手机中的浙里办APP扫码功能，扫码连接Debug工具。

连接成功，Debug工具自动拉取浙里办对应功能的JSAPI。

② 说明

要求手机和Debug工具处于同一局域网中，否则无法扫码连接Debug工具。



- 1, 单击按钮打开二维码，通过浙里办APP扫码连接Debug工具。
- 2, 红色Tag是h5请求。
- 3, 状态码：
 - 200~299为绿色表示请求成功
 - 300~399为黄色表示将进一步请求
 - 400及以上为红色表示请求失败
- 4, 蓝色Tag表示请求类型：
 - HTML为链接请求
 - XHR为页面内的网络请求，包含请求的所有信息，排查问题时重点关注XHR。
 - 没有Tag表示除HTML、XHR以外的所有请求，包括CSS、JS、图片、字体等下载请求。

4. 单击左侧导航栏的调试，输入待调试页面的URL地址，单击go进行调试。



3.9. 应用部署发布

3.9.1. 构建发布系统简介

开发商工作台基于Docker体系和NodeJS构建发布系统，Docker镜像版本为NodeJS Long Term Support (LTS) 版（node:lts），根据NodeJS升级计划自动更新默认版本。构建流程如下所示：



● 构建任务的运行环境

- Linux Debain 9
- Node 14
- CNPM镜像源

● 项目源码规范

接入发布系统的源码需满足以下条件：

- 支持 `npm run build` 命令
- 支持默认将 `npm run build` 命令的执行结果存储在项目根路径的build目录。

● 项目配置

配置文件gbc.json需存储在项目根路径下，如果使用默认配置，则无需放置该文件。

`outputPath`

- 作用：系统默认要求 `npm run build` 命令的执行结果存储在build目录下，若build目录已被其他文件占用，可以修改配置文件gbc.json中的`outputPath`参数指定 `npm run build` 命令的执行结果。
- type: string
- 默认值 '`'build'`'

3.9.2. 部署发布应用

前提条件

服务侧应用发布端包含浙里办APP或者治理侧应用发布类型为H5，系统默认进行代码托管，需要开发商在开发商工作台完成应用部署发布。

操作步骤

1. 登录开发商工作台。
2. 单击左侧导航栏发布管理。



3. 根据业务需要，在左侧导航栏选择服务侧名称列表或者治理侧名称列表，本示例选择服务侧名称列表，然后在搜索区域输入应用名称，选择应用类型和状态（部署准备中），单击查询。
4. 单击目标应用右侧的更多>>部署发布，

| 服务侧名称 | 创建时间 | 更新时间 | 类型 | 当前状态 | 服务接入数 | PC地址 | 移动端地址 | 操作 |
|--------------|---------------------|---------------------|---------|-------|-------|-----------------------|---------------------|--|
| 测试真实表单100401 | 2021-10-04 14:20:29 | 2021-10-04 14:28:45 | H5 | 部署准备中 | 0去接入 | - | - | 详情 监控 更多 |
| 测试PC端服务向应用 | 2021-10-03 20:40:01 | 2021-10-03 20:40:02 | 浙江政务服务网 | 待发布审批 | 0去接入 | http://www.tmall.com | - | 部署发布 配置 |
| 测试升级1003 | 2021-10-03 19:41:01 | 2021-10-04 15:25:03 | H5 | 升级部署中 | 0去接入 | - | 二维码 | 详情 监控 更多 |
| 测试全流程1003 | 2021-10-03 19:19:13 | 2021-10-03 19:50:42 | H5 | 待升级审批 | 0去接入 | - | 二维码 | 详情 监控 更多 |
| 测试信息变更01 | 2021-10-03 14:24:13 | 2021-10-03 20:01:57 | H5 | 待内审审批 | 0去接入 | - | 二维码 | 详情 监控 更多 |
| 测试100302 | 2021-10-03 10:49:23 | 2021-10-03 10:49:23 | H5 | 部署准备中 | 0去接入 | https://www.tmall.com | - | 详情 监控 更多 |
| 测试100301 | 2021-10-03 10:47:41 | 2021-10-03 21:02:14 | H5 | 待升级验收 | 0去接入 | - | 二维码 | 详情 监控 更多 |
| 测试1003 | 2021-10-03 10:44:03 | 2021-10-03 20:53:15 | H5 | 待升级审批 | 0去接入 | - | 二维码 | 详情 监控 更多 |

5. 在测试环境模块单击创建版本。

The screenshot shows a table with columns: 服务侧名称 (Service Side Name), 创建时间 (Creation Time), 更新时间 (Last Update), 类型 (Type), 当前状态 (Current Status), 服务接入数 (Number of Services), PC地址 (PC Address), 移动端地址 (Mobile Address), and 操作 (Operations). The table contains three rows:

| 服务侧名称 | 创建时间 | 更新时间 | 类型 | 当前状态 | 服务接入数 | PC地址 | 移动端地址 | 操作 |
|-------|---------------------|---------------------|----|------|-------|------|-------|--|
| 服务01 | 2021-10-02 16:27:47 | 2021-10-02 16:35:18 | H5 | 0连接 | - | - | - | 详情 监控 更多 |
| 外链测试2 | 2021-10-02 00:42:21 | 2021-10-02 15:09:18 | H5 | 0连接 | - | - | - | 部署使用 配置 |
| 外链测试 | 2021-10-02 00:10:31 | 2021-10-02 00:10:31 | H5 | 0连接 | - | - | - | 部署使用 配置 |

6. 在上传代码页面，设置版本名称并上传代码，单击确认。

The dialog box has the following fields and sections:

- * 当前版本: v1.0
- 上传代码: (File upload area with placeholder text: 点击或者拖动文件到虚线框内上传)
- 仅支持 zip 类型的文件，文件命名请使用字母、数字、横杠和点
- 预览: demo.zip (207.57K)
- 操作按钮: 确认 (Confirm) | 取消 (Cancel)

7. 代码编译通过时，可以将**应用提交验收**。单击查看详情查看代码构建状态和时长。

8. 代码检查未通过时，单击**查看详情**查看源码检测结果，请参见**源码检测**。

注意 目前系统仅对服务侧应用进行源码检测。

3.9.3. 源码检测（服务侧应用）

应用为服务侧应用时，若提交的源代码已经完成源代码构建，但未通过源代码合规检测（应用状态为编译失败）时，禁止直接上线发布，需根据源代码合规检测结果进行修复，修复完成后重新部署代码。若存在某些客观原因导致源代码未通过合规检测，可以提交审批，经相关人员审批后可将代码发布上线。

前提条件

应用为服务侧应用。

注意事项

如存在以下客观情况之一导致源代码未通过合规检测，可以提交审批发起审批流，经相关人员审批后可上线发布：

- 需要紧急上线，修复源代码不合规项次的工作在会在之后的版本迭代中完成。
- 源代码没有问题，合规检测出现误报。

- 其他合理的要求，需要忽略源代码合规检测结果。

提交审批

- 登录开发商工作台。
- 单击左侧导航栏发布管理。



- 在左侧导航栏选择服务侧名称列表或者治理侧名称列表，本示例选择服务侧名称列表，然后在搜索区域输入应用名称，选择应用类型和状态（部署准备中），单击查询。
- 单击目标应用右侧的更多>>部署发布，在测试环境模块单击详情。

| 测试环境 | 板本号 | 更新时间 | 预览地址 | 查看日志 | 应用状态 | 应用出入参数 | 操作 |
|------|-----|---------------------|------|--------------------|--------|--------|---|
| | 1 | 2021-10-02 19:48:41 | | 详情 | 代码编译通过 | 配置 | 上传代码 联调 提交验收 删除 |

- 代码构建页面显示代码构建时长等信息，单击源码检测。
- 在源码页面提示您源代码合规检测未通过，并给出未通过具体原因。
- 在外链检测区域，显示检测状态、文件总数量以及需要扫描的文件数量（例如25个），单击未通过检测文件，显示未通过原因，需要您根据未通过原因修复代码。

注意 您需要单击查看每一个文件中检测未通过的原因，并完成代码修复后，再重新部署发布应用。

◎ 源代码合规检测未通过

所提交的源代码已经完成源代码构建，未通过源代码合规检测，禁止直接上线发布，请根据源代码合规检测结果进行修复。

如果存在以下常见情况之一：

- 需要紧急上线，修复源代码不合规问题的工作会在之后的版本迭代中完成。
- 源代码没有问题，台帐检测出现误报。
- 其他合理的要求，需要忽略源代码合规检测结果。

可以发起审批流，经过相关人员统一审批后才可以上线发布。

[提交审批](#)

数字政府_浙江开放_外跳检测

用于检测是否存在对外跳转

| 状态 | 未通过 | 文件数量 | 41 | 扫描数量 | 25 | 提示 | 未通过 |
|---|-----|------|----|------|----|----|-----|
| 未通过 index.html | | | | | | | |
| 发现html中存在 a 标签 href转跳: http://www.baidu.com | | | | | | | |
| 未通过 index.html | | | | | | | |
| 发现html script 标签内存在 location.href 调用 | | | | | | | |
| 未通过 index.jsx | | | | | | | |
| 未通过 index.vue | | | | | | | |
| 未通过 raxh5/index.html | | | | | | | |
| 通过 index.js | | | | | | | |

- 如存在客观情况导致源代码未通过合规检测，可以单击提交审批，经相关人员审批后可上线发布。

3.9.4. 应用提交验收

应用代码编译通过后，可以将代码提交给业主单位政府工作人员进行审核。

操作步骤

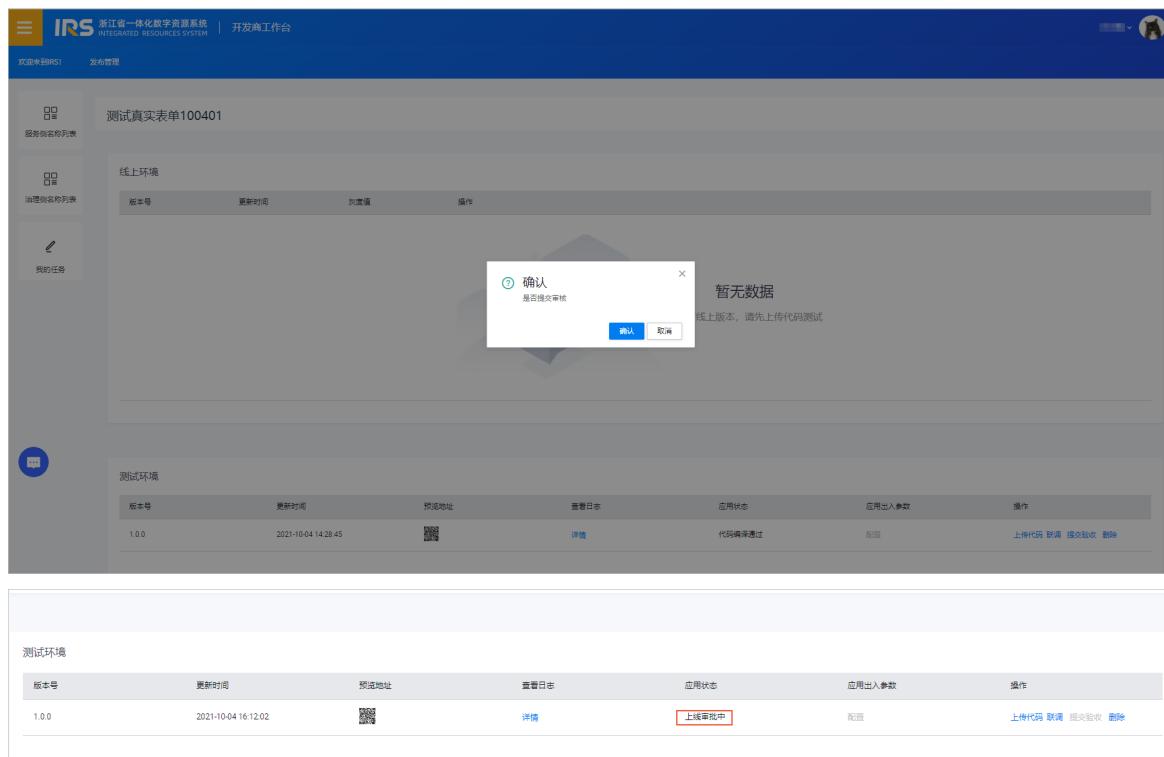
- 登录开发商工作台。
- 单击左侧导航栏发布管理。



- 在左侧导航栏选择服务侧名称列表或者治理侧名称列表，本示例选择服务侧名称列表，然后在搜索区域输入应用名称，选择应用类型和状态（部署准备中），单击查询。
- 单击目标应用右侧的更多>部署发布。
- 在测试环境模块单击提交验收，单击确认，由业主单位政府工作人员在主工作台中完成应用验收。

注意 提交验收后开发商工作台中应用状态为上线审批中，IRS主工作台中应用状态为待验收，政府工作人员可进行验收或者重新部署操作。

| 测试环境 | | | | | | |
|------|---------------------|------|------|--------|--------|-----------------|
| 版本号 | 更新时间 | 预览地址 | 查看日志 | 应用状态 | 应用出入参数 | 操作 |
| 1 | 2021-10-02 19:48:41 | | 详情 | 代码编译通过 | 配置 | 上传代码 联网 提交验收 删除 |



应用发布后自动发布至线上环境。

3.9.5. 配置完成（治理侧应用）

前提条件

- 应用为治理侧应用。
- 应用状态为配置中。

操作步骤

1. 登录开发商工作台。
2. 根据业务需要，单击左侧导航栏治理侧名称列表。
3. 在搜索区域通过关键字快速搜索目标服务，应用状态为配置中，然后单击查询。
4. 单击目标应用右侧的配置完成，将应用状态更新为已发布。

3.10. 代码发布管理

3.10.1. 线上版本代码回滚

可以将线上环境的代码回滚至指定版本。

操作步骤

1. 登录开发商工作台。
2. 单击左侧导航栏发布管理。



3. 根据业务需要，在左侧导航栏选择服务侧名称列表或者治理侧名称列表，在搜索区域输入应用名称，选择类型和状态（已发布），单击查询。
4. 单击目标服务右侧的部署发布，在线上环境模块单击回滚到上一个版本或者选择指定版本进行回滚。

3.11. 我的任务

开发商工作台运维人员负责系统创建审核、服务接入申请、外链审核。

操作步骤

1. 登录**开发商工作台**。
2. 单击左侧导航栏**发布管理**。



3. 单击左侧导航栏的**我的任务**。
4. 在待办任务页签下，输入申请对象（例如系统名称等关键字），或者从下拉框中选择任务类型，本示例选择系统创建，单击查询。
5. 单击目标任务右侧的详情进入任务详情页面。

申请详情

申请对象信息

系统英文名称: doc-test

系统名称: 文档测试

描述: 文档测试

申请人信息

申请人姓名: huarong

申请人联系方式: 12345678765

申请人邮箱: huarong@163.com

申请人单位: 文档测试

操作按钮

通过 **不通过** **取消**

6. 阅读并确认申请详情，完成审批：

- **通过**: 单击**通过**，审核通过。
- **不通过**: 单击**不通过**，需要填写不通过原因，审批被驳回。提交者可根据驳回原因修改任务，重新提交审核。

7. 在已办任务页签下，查看已处理过的申请任务。

3.12. JSBridge API

3.12.1. API概览

缓存

| API | 浙里办APP | 浙里办支付宝小程序 | 描述 |
|--------|--------|-----------|-------------|
| 存储数据缓存 | √ | √ | 通过键值对存储数据缓存 |
| 读取数据缓存 | √ | √ | 读取数据缓存 |
| 删除缓存 | √ | √ | 删除缓存 |

Navigation窗口类

| API | 浙里办APP | 浙里办支付宝小程序 | 描述 |
|--------|--------|-----------|----------|
| 导航栏标题 | √ | √ | 设置导航栏标题 |
| 菜单 | √ | X | 设置菜单 |
| 新开窗口 | √ | √ | 从新窗口加载页面 |
| 关闭当前页面 | √ | √ | 关闭当前页面 |

用户

| API | 浙里办APP | 浙里办支付宝小程序 | 描述 |
|------------|--------|-----------|--------------|
| 获取用户类型 | √ | √ | 获取用户类型 |
| 获取用户UI风格 | √ | X | 获取用户当前UI风格 |
| 支付宝扫脸认证 | √ | √ | 启用支付宝扫脸认证 |
| 公安一所可信扫脸认证 | √ | X | 启用公安一所可信认证功能 |

Device设备类

| API | 浙里办APP | 浙里办支付宝小程序 | 描述 |
|----------|--------|-----------|-----------|
| 打电话 | √ | √ | 唤起拨打电话 |
| 发短信 | √ | ✗ | 唤起发送短信 |
| 获取经纬度信息 | √ | √ | 获取经纬度位置信息 |
| 获取设备唯一标识 | √ | ✗ | 获取设备的唯一标识 |
| 获取网络类型 | √ | √ | 获取网络类型 |
| 剪切板 | √ | ✗ | 向剪贴板中复制数据 |

业务类

| API | 浙里办APP | 浙里办支付宝小程序 | 描述 |
|---------|--------|-----------|------------|
| 埋点 | √ | ✗ | 埋点 |
| 获取当前地区 | √ | √ | 获取当前用户所属地区 |
| 图片保存到本地 | √ | √ | 将图片保存到本地 |
| 选择图片 | √ | √ | 选择图片 |
| 扫一扫 | √ | √ | 启用扫一扫功能 |
| 支付 | √ | ✗ | 启用支付功能 |

UI界面类

| API | 浙里办APP | 浙里办支付宝小程序 | 描述 |
|-----|--------|-----------|-------|
| 确认框 | √ | √ | 添加确认框 |

| API | 浙里办APP | 浙里办支付宝小程序 | 描述 |
|--------|--------|-----------|----------------------|
| 提示框 | √ | √ | 添加提示框 |
| 弱提示 | √ | √ | 添加弱提示 |
| 文本输入框 | √ | √ | 添加文本输入框 |
| 单选列表 | √ | √ | 添加单选列表 |
| 等待蒙版显示 | √ | √ | 添加等待蒙版提示 |
| 等待蒙版隐藏 | √ | √ | 添加等待蒙版隐藏 |
| 选择城市 | √ | √ | 选择城市 |
| 分享 | √ | X | 内容分享，通过系统弹出窗口选择分享通道。 |
| 直接分享 | √ | X | 内容分享，通过参数指定分享通道。 |

3.12.2. 调试JSBridge API

下载安装Debug调试工具后，可以在调试工具中查看和调试全量JSBridge API。

前提条件

已下载和安装Debug调试，下载地址：

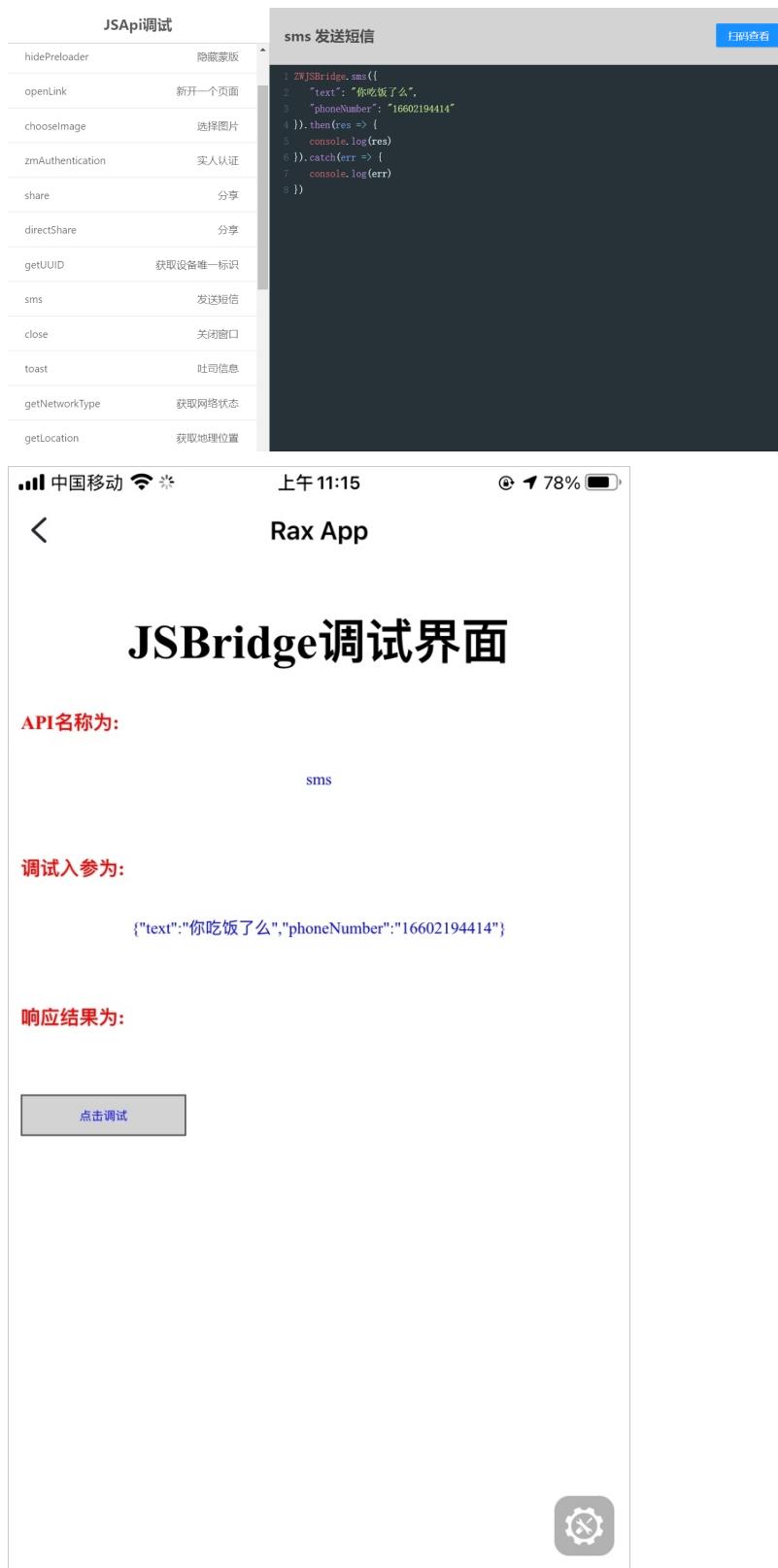
- Windows: [ZWEurope 3.1.0](#)
- Mac: [ZWEurope 3.1.0](#)

操作步骤

1. 打开Debug调试工具。
2. 单击左侧导航栏设备绑定，打开连接码，通过手机浙里办APP扫码连接Debug调试工具。
3. 单击JSAP按钮，打开调试页面。
4. 在JSAP调试列表单击目标JSBridge API（例如sms API），右侧调试区域同步显示对应JSBridge API的调用示例。
5. 单击扫码查看打开二维码，手机登录浙里办APP并扫码即可对JSBridge API功能进行调试。

② 说明

支持在调试页面更改JSBridge API入参，然后通过手机扫码实时验证调试结果。



单击点击调试按钮，系统自动跳转到发送短信页面。



3.12.3. ZWJSBridge接入方式

接入bridge

通过以下方式接入bridge：

```
<script type="text/javascript" src="//assets.zjzwfw.gov.cn/assets/ZWJSBridge/1.0.1/zwsbridge.js">
</script>
```

src替换为：//assets.zjzwfw.gov.cn/assets/ZWJSBridge/1.0.1/zwsbridge.js

引用jsbridge会自动引入 `portal.zjzwfw.gov.cn/assets/jsbridge/1.0.1/jsbridge.js`。

初始化

通过 `ZWJSBridge.onReady(callback)` 初始化jsapi，初始化完成即onReady之后再调用jsapi。

```
ZWJSBridge.onReady(() =&gt; {
  console.log('初始化完成后，执行bridge方法')
})
```

调用方式

```
ZWJSBridge.方法({
  参数1: '',
  参数2: '',
  onSuccess: function(e) {},
  onFailure: function(e) {}
})
```

支持promise回调：

```
ZWJSBridge.方法([
  参数1: '',
  参数2: '',
]).then(e => {
  // 成功
}).catch(e => {
  // 失败
})
```

以openLink接口为例：

```
ZWJSBridge.openLink({
  url: 'https://www.baidu.com'
}).then(res => {
  console.log('跳转成功')
}).catch(res => {
  console.log('跳转失败')
})
```

3.12.4. 缓存

3.12.4.1. 存储数据缓存

调用setLocalStorage接口通过键值对存储数据缓存。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|-------|--------|------|-----|----|
| key | String | 否 | aa | 健 |
| value | String | 是 | 123 | 值 |

示例

请求示例

```
ZWJSBridge.setLocalStorage({
  key:'aa',
  value:'123'
}).then((result) =&gt; {
  console.log(result);
}).catch((error) =&gt; {
  console.log(error);
});
```

3.12.4.2. 读取数据缓存

调用getLocalStorage读取数据缓存。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|-----|--------|------|-----|----|
| key | String | 否 | 123 | 健 |

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|----------|----|-----|----|
| data.key | | | 值 |

示例

请求示例

```
ZWJSBridge.getLocalStorage({
  key:'aa'
})
  .then((data) =&gt; {
  // data -&gt; {key: xxxx}
}).catch((error) =&gt; {
  console.log(error);
});
```

3.12.4.3. 删除缓存

调用removeItem接口删除缓存。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|-----|--------|------|-----|----|
| key | String | 否 | aa | 键 |

示例

请求示例

```
ZWJSBridge.removeItem({
  key:'aa'
})
  .then((result) =&gt; {
  console.log(result);
}).catch((error) =&gt; {
  console.log(error);
});
```

3.12.5. Navigation窗口类

3.12.5.1. 导航栏标题

调用setTitle接口设置导航栏。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|-------|--------|------|------|----------------------|
| title | String | 否 | 邮箱正文 | 设置导航栏标题文本，默认文本为空字符串。 |

示例

请求示例

```
ZWJSBridge.setTitle({  
    title: '邮箱正文'  
}).then((result) => {  
    console.log(result);  
}).catch((error) => {  
    console.log(error);  
});
```

3.12.5.2. 菜单

调用setMenu接口设置菜单。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|--------------|-------------|------|----------------------------|----------|
| items | array[item] | 否 | | 按钮。 |
| item.id | String | 否 | 1 | 按钮ID。 |
| item.iconUrl | String | 否 | http://t.cn/im age1.png | 图标地址。 |
| item.text | String | 否 | 帮助 | 按钮的说明文字。 |

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|---------|--------|-----|-------|
| data.id | String | | 按钮的ID |

示例

请求示例

```
ZWJSBridge.setMenu({
  items: [
    {
      "id":"1", // 按钮ID，点击后客户端返回这个ID标识
      "iconUrl":"http://t.cn/image1.png",// icon显示的样式
      "text":"分享"
    },
    {
      "id":"2",
      "iconUrl":"http://t.cn/image2.png",
      "text":"订阅"
    }
  ]).then((result) =>{
  console.log(result);
}).catch((error) =>{
  console.log(error);
});
```

3.12.5.3. 新开窗口

调用openLink接口从新窗口加载页面。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|-----|--------|------|----------------------|--|
| url | String | 是 | http://www.dream.com | <p>打开窗口的url地址。</p> <p>注意</p> <p>若要唤起外部导航，url格式为</p> <p><code>zwfw://openThirdMap? longitude=xx&latitude=xx&locationName=xx</code></p> <p>，参数取值：</p> <ul style="list-style-type: none">• longitude: 经度• latitude: 纬度• locationName: 地点名称 <p>例如<code>zwfw://openThirdMap? longitude=119.908949&latitude=31.014303&locationName=长兴体育中心。</code></p> |

示例

请求示例

```
ZWJSBridge.openLink({  
    url:'http://xxxxx'  
}).then((result) => {  
    console.log(result);  
}).catch((error) => {  
    console.log(error);  
});
```

3.12.5.4. 关闭当前页面

调用close()接口关闭通过openLink打开的页面。

示例

请求示例

```
ZWJSBridge.close().then((result) => {  
    console.log(result);  
}).catch((error) => {  
    console.log(error);  
});
```

3.12.6. 用户

3.12.6.1. 获取用户类型

调用getUserType接口获取用户类型。

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|----|----|-----|----|
| | | | |

| 参数 | 类型 | 示例值 | 描述 |
|---------------|--------|-----|--|
| data.userType | Number | 0 | <p>用户类型，取值：</p> <ul style="list-style-type: none">• 0：公务员• 1：除公务员以外的个人• 2：法人 <div style="background-color: #e1f5fe; padding: 10px; border-radius: 5px;"><p>? 说明</p><p>公务员属于特殊类型的个人用户，当应用服务需要同时支持个人和法人时，可通过getUserType接口获取APP当前登录用户类型，然后发起个人或法人的登录认证，0或1都属于个人用户，2为法人用户。</p></div> |

示例

请求示例

```
ZWJSBridge.getUserType().then((result) =&gt;{  
    console.log(result);  
}).catch((error) =&gt;{  
    console.log(error);  
});
```

3.12.6.2. 获取用户当前UI风格

调用getUiStyle接口获取用户当前UI风格。

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|----|----|-----|----|
|----|----|-----|----|

| 参数 | 类型 | 示例值 | 描述 |
|--------------|--------|--------|---|
| data.uiStyle | String | normal | <p>当前UI风格，取值：</p> <ul style="list-style-type: none"> • normal：常规版 • elder：老年版 |

示例

请求示例

```
ZWJSBridge.getUiStyle().then((result) =>
  console.log(result);
).catch((error) =>
  console.log(error);
});
```

3.12.6.3. 支付宝扫脸认证

调用zmAuthentication接口进行实名认证。

② 说明

该接口涉及业务签约，一旦服务到期后，将改变相应接口的调用方式，请及时按照本文档进行适配。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|----------|--------|------|------------------------|--|
| certNo | String | 否 | 411111111111 11**** | <p>身份证号，默认值为当前登录账号所属身份证号码。</p> <div style="background-color: #e1f5fe; padding: 5px;"> ② 说明 适用于v4.3.1以上版本。 </div> |
| certName | String | 否 | 张某某 | <p>姓名，默认值为当前登录账号所属名字</p> <div style="background-color: #e1f5fe; padding: 5px;"> ② 说明 适用于v4.3.1以上版本。 </div> |

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|-------------------|---------|---------|---|
| status | String | success | <ul style="list-style-type: none">● success: 成功。● fail: 失败。● complete: 完成。 |
| msg | String | | 返回信息。 |
| pass | Boolean | true | 成功回调时为true, 否则为false。 ② 说明 Android v4.3.1及以上版本支持该参数。 |
| passId | String | | 从后台返回的用户票据信息。 |
| data.pass | Boolean | true | 成功回调时为true, 否则为false。 ② 说明 <ul style="list-style-type: none">● Android v4.3.1及以上版本支持该参数。● 仅小程序支持该参数。 |
| data.passId | String | | 从后台返回的用户票据信息。 ② 说明 仅小程序支持该参数。 |
| data.verifyResult | Object | | 认证结果 ② 说明 仅小程序支持该参数。 |

示例

请求示例

入参为空，使用当前登录账号唤起扫脸。

```
ZWJSBridge.zmAuthentication({
  }
  .then((data) =&gt; {
    // data=&gt; {"pass" : true,
    // "passId" : "xxxxx"}
  }).catch((error) =&gt; {
    console.log(error);
  });
});
```

直接传递所需认证人信息。

```
ZWJSBridge.zmAuthentication({
  certNo:'身份证',
  certName:'姓名'
})
  .then((data) =&gt; {
    // data=&gt; {"pass" : true,
    // "passId" : "xxxxx"}
  }).catch((error) =&gt; {
    console.log(error);
  });
});
```

错误码

| 错误代码 | 描述 |
|------|-----------|
| 1001 | 支付宝认证失败 |
| 1003 | 姓名或身份证号错误 |

3.12.6.4. 公安一所可信扫脸认证

调用kexinAuthentication接口启用公安一所可信认证功能。

请求参数

| 参数 | 类型 | 是否必填 | 示例值 | 描述 |
|--------|--------|------|------------------|--------------------------------|
| certNo | String | 否 | 412*****3 426 | 待认证的身份证号码，type为0时，certNo为必传参数。 |

| 参数 | 类型 | 是否必填 | 示例值 | 描述 |
|----------|---------|------|-----|---|
| certName | String | 否 | 张某 | 待认证的名字，type为0时，certName为必传参数。 |
| ticketId | String | 否 | | 查询标识号，type为1时，ticketId为必传参数。 |
| type | Integer | 否 | 0 | 设置认证方式，取值： • 0：上传身份证件和姓名，默认值为0。 • 1：上传ticketId。 |

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|---------------|--------|-----|-----------------|
| data.ticketId | String | | 查询标识号，用于查询认证结果。 |

示例

请求示例

```
ZWJSBridge.kexinAuthentication({
  type: 0,
  certNo: '',
  certName: '')
  .then((data) => {
    // data -> {ticketId: xxxx}
  })
  .catch((error) => {
    console.log(error);
  });
});
```

```
ZWJSBridge.kexinAuthentication({
  type:1,
  ticketId:'')
  .then((data) => {
    // data -> {ticketId: xxxx}
  })
  .catch((error) => {
    console.log(error);
  });
});
```

3.12.7. Device设备类

3.12.7.1. 打电话

调用phoneCall接口唤起拨打电话。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|--------|--------|------|-------------|-------|
| corpid | String | 是 | 136****2531 | 电话号码。 |

示例

请求示例

```
ZWJSBridge.phoneCall({  
    corpid:'136****2531'  
}).then((result) => {  
    console.log(result);  
}).catch((error) => {  
    console.log(error);  
});
```

3.12.7.2. 发短信

调用sms接口唤起发送短信。

② 说明

发送短信前，已选择接收人并输入短信内容。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|-------------|--------|------|-------------|------|
| phoneNumber | String | 否 | 136****2531 | 电话号码 |
| text | String | 否 | 消息 | 短信内容 |

示例

请求示例

```
ZWJSBridge.sms({
  phoneNumber:'136****2531',
  text:'内容'
}).then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

错误码

| 错误代码 | 描述 |
|------|-----------|
| 1001 | 设备不支持短信功能 |
| 1002 | 定位失败 |

3.12.7.3. 获取经纬度信息

调用getLocation接口获取经纬度位置信息。

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|--------------------|----|-----|------|
| data.longitude | | | 经度 |
| data.latitude | | | 纬度 |
| data.cityName | | | 城市名称 |
| data.region | | | 区域名称 |
| data.townCode | | | 地区编码 |
| data.detailAddress | | | 详细地址 |

示例

请求示例

```
ZWJSBridge.getLocation()  
  .then((result) => {  
    console.log(result);  
  }).catch((error) => {  
    console.log(error);  
});
```

错误码

| 错误代码 | 描述 |
|------|---------|
| 1001 | 未开启定位权限 |
| 1002 | 定位失败 |

3.12.7.4. 获取设备唯一标识

调用getUUID接口获取设备唯一标识。

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|-----------|----|--------------|--------|
| data.uuid | | 112233...566 | 设备唯一标识 |

示例

请求示例

```
ZWJSBridge.getUUID().then((data) => {  
  // data -> {uuid: xxx}  
}).catch((error) => {  
  console.log(error);  
});
```

3.12.7.5. 获取网络类型

调用getNetworkType接口获取网络类型。

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|-------------|--------|------|--|
| data.result | String | wifi | 网络类型，取值： <ul style="list-style-type: none">• wifi• 2g• 3g• 4g• unknown• none，表示离线 |

示例

请求示例

```
ZWJSBridge.getNetworkType().then((data) =&gt; {
  // data -&gt; {result: 'wifi'}
}).catch((error) =&gt; {
  console.log(error);
});
```

3.12.7.6. 剪切板

向剪贴板中复制数据。

输入参数

| 参数 | 类型 | 示例值 | 描述 |
|------|--------|-----|------------|
| text | String | | 复制到剪切板中的数据 |

请求示例

```
ZWJSBridge.setClipboardData({
  text: "xx"
}).then((result =&gt; {
  ...
})).catch(error =&gt; {
  ...
})
```

3.12.8. 业务类

3.12.8.1. 埋点

调用monitorTrace接口来调用埋点。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|-------------|--------|------|----------|---|
| monitorType | String | 是 | success | 埋点类型： • success: 成功 • fail: 失败 • count: 计数 • timeCost: 时延 • pageIn: 进入页面 • pageOut: 离开页面 • exposure: 曝光 • click: 单击 |
| module | String | 否 | Login | 模块。 |
| modulePoint | String | 否 | password | 模块点。 |
| bizInfo | Map | 否 | | 业务参数。 |
| errorCode | String | 否 | | 错误码。 |
| errorMsg | String | 否 | | 错误信息。 |
| count | Int | 否 | | 计数。 |
| timeCost | Long | 否 | | 时延。 |
| pageName | String | 否 | | 页面。 |
| actionName | String | 否 | | 行动点。 |

示例

请求示例

```
ZWJSBridge.monitorTrace({
  monitorType: 'success',
  module: 'Login',
  modulePoint: 'password',
  bizInfo: {

  }
}).then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

3.12.8.2. 获取当前地区

调用getCurrentLocationCity接口获取当前城市。

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|---------------|--------|--------|--------|
| data.cityId | String | 339900 | 行政区划编码 |
| data.cityName | String | 浙江省本级 | 行政区划名 |

示例

请求示例

```
ZWJSBridge.getCurrentLocationCity()
  .then((data) => {
    // data={"cityId":"339900","cityName":"浙江省本级"}
  }).catch((error) => {
    console.log(error);
});
```

3.12.8.3. 选择图片

调用chooseImage接口来选择图片，支持拍照和从相册中选取两种方式。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|--------|---------|------|-------|---|
| upload | Boolean | 是 | false | <p>是否将选取的图片上传到服务器。</p> <ul style="list-style-type: none"> 默认值为false，图片不上传到服务器。 值为true时，图片上传到服务器，上传成功后返回公网可访问的URL。 <div style="background-color: #e1f5fe; padding: 5px; margin-top: 10px;"> ? 说明 敏感图片不可以设置为true。 </div> |

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|---------|-------|--|--|
| picSrc | Array | | Base64编码格式的图片数据数组。 |
| picPath | Array | {http://zjzwfw.gov.cn/picture/0/xxx.png} | upload取值为true时，picPath为图片的网络地址数组，支持下载。 |

示例

请求示例

```
ZWJSBridge.chooseImage({
  upload:true
}).then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

错误码

| 错误代码 | 描述 |
|------|--------------|
| 1001 | 没有摄像头或摄像头不可用 |
| 1002 | 没有拍照权限 |

| 错误代码 | 描述 |
|------|---------|
| 1003 | 没有图片库权限 |
| 1004 | 图片上传失败 |

3.12.8.4. 图片保存到本地

调用saveImage接口将图片保存到本地，其中Android系统是将图片保存在外部存储上，iOS系统将图片保存在相册中。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|-----|--------|------|---|------|
| url | String | 否 | http://zjzwfw.gov.cn/picture/0/xxx.xx.png | 图片地址 |

示例

请求示例

```
ZWJSBridge.saveImage({
  url: "http://zjzwfw.gov.cn/picture/0/xxx.png"
}).then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

错误码

| 错误代码 | 描述 |
|------|------|
| 1001 | 保存失败 |

3.12.8.5. 扫一扫

调用scan接口启用扫一扫功能。

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|------|--------|--------|---------|
| text | String | 扫描到的内容 | 扫描到的内容。 |

示例

请求示例

```
ZWJSBridge.scan({
  type: "qrCode"
}).then((data) => {
  // data = {"text": "扫描到的内容"}
}).catch((error) => {
  console.log(error);
});
```

错误码

| 错误代码 | 描述 |
|------|--------------|
| 1001 | 没有摄像头或摄像头不可用 |
| 1002 | 没有拍照权限 |
| 1003 | 没有图片库权限 |
| 5 | 未初始化 |

3.12.8.6. 启用支付功能

调用pay接口启用支付功能。

② 说明

实际支付结果以后端查询接口为准，成功回调仅表示支付宝支付调用成功，相关数据可能不会同步至后端。若回调失败，可通过返回数字获取失败信息。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|----------------|---------|------|----------|---|
| platform | String | 否 | 1 | 支付平台，取值： • 1: 支付宝 • 2: 微信，微信支付暂不支持 • 3: 银联云闪付 |
| arg.credential | String | 否 | 待支付的订单信息 | 等待支付的订单信息 |
| arg.inSandBox | Boolean | 否 | False | 是否为测试环境，缺省为False。 ② 说明 支付宝只支持Android端。 |

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|-------|----|-----|--|
| error | | | 调用支付宝支付发生错误时的错误码，请参见 error错误码 。 |

示例

请求示例

```
ZWJSBridge.pay({
  platform:'1',
  arg:{
    "credential":"待支付的订单信息",
    "inSandBox": false
  }
}).then((result) => {
  console.log(result);
}).catch((error) => {
  // error = "8000"
});
```

error错误码

| 错误码 | 描述 |
|-------|-----------------------------------|
| 8000 | 支付正在处理中，支付结果未知，请查询商户订单列表中订单的支付状态。 |
| 40000 | 订单支付失败。 |
| 5000 | 重复请求。 |
| 6001 | 用户中途取消支付。 |
| 60002 | 网络连接出错。 |
| 60004 | 支付结果未知，请查询商户订单列表中订单的支付状态。 |

3.12.8.6.1. 支付能力

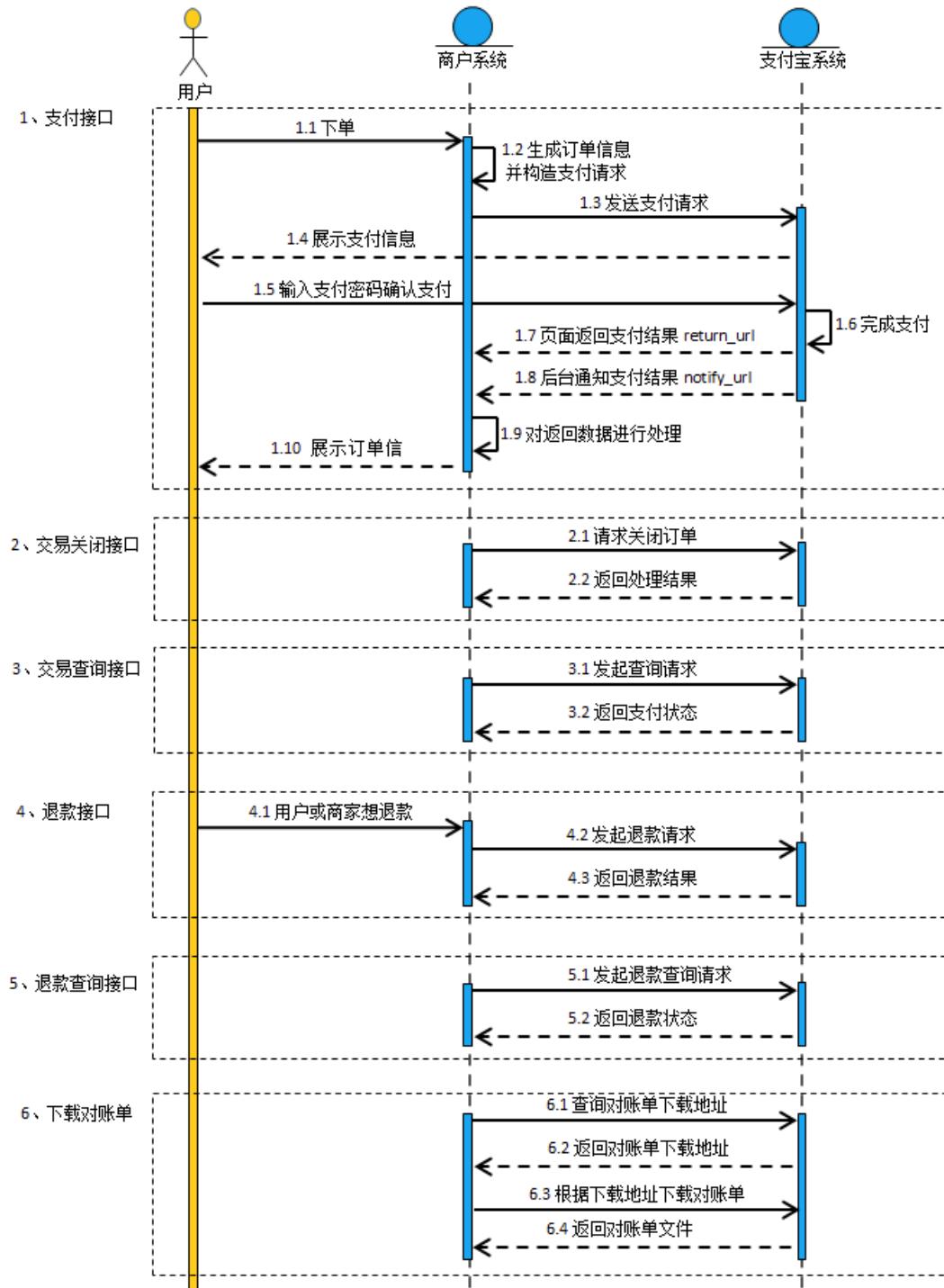
应用开放平台支持H5和App两种支付能力，这两种支付能力依赖于支付宝，本文介绍如何在H5和App能力中集成支付宝支付功能。

⌚ 注意

根据当前window.navigator.userAgent取值判断运行环境为App或者H5，其中window.navigator.userAgent取值包含Jupiter时为App，其他值则为H5。

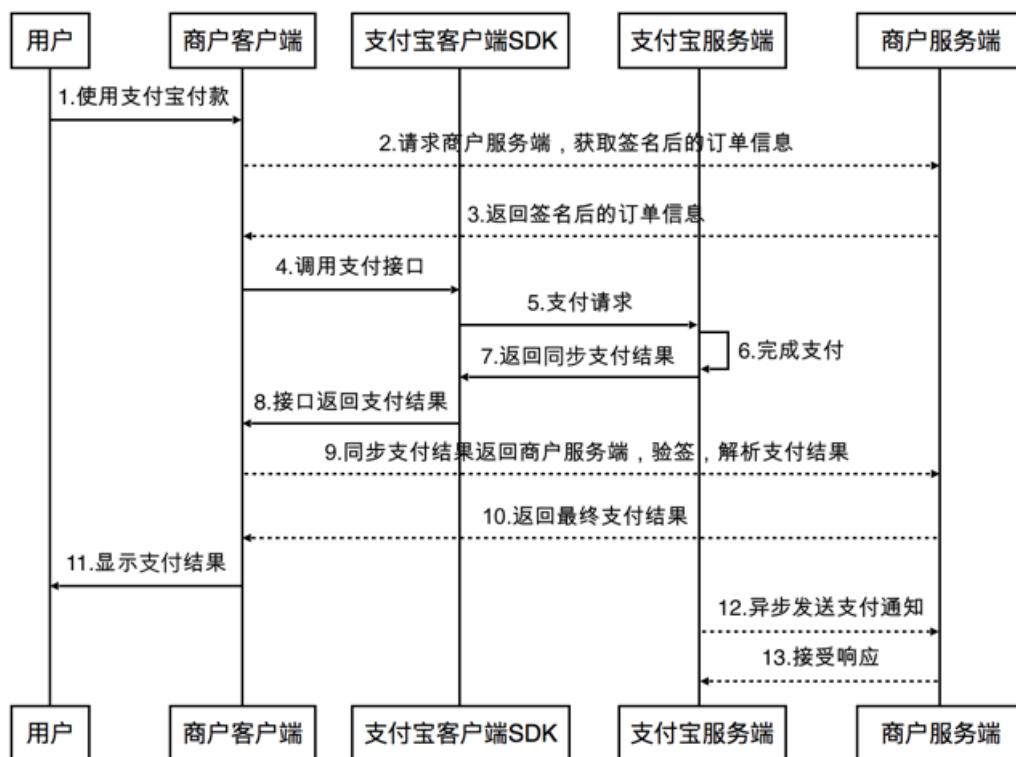
H5集成支付宝支付功能

H5集成支付宝支付功能的流程如下图所示，详情请参见[服务端接入流程](#)。



App集成支付宝支付功能

App集成支付宝支付功能流程图如下所示，详情请参见[服务端接入流程](#)。其中步骤4调用支付接口、步骤5支付请求、步骤6完成支付、步骤7返回同步支付结果、步骤8接口返回支付结果均由[pay](#)接口完成，其他流程可参照[服务端接入流程](#)中的内容来完成。



3.12.8.6.2. 启用支付功能

调用pay接口启用支付功能。

② 说明

实际支付结果以后端查询接口为准，成功回调仅表示支付宝支付调用成功，相关数据可能不会同步至后端。若回调失败，可通过返回数字获取失败信息。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|----------------|--------|------|----------|---|
| platform | String | 否 | 1 | 支付平台，取值： • 1：支付宝 • 2：微信，微信支付暂不支持 • 3：银联云闪付 |
| arg.credential | String | 否 | 待支付的订单信息 | 等待支付的订单信息 |

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|---------------|---------|------|-------|--|
| arg.inSandBox | Boolean | 否 | False | <p>是否为测试环境，缺省为False。</p> <div style="background-color: #e1f5fe; padding: 5px;"> ② 说明 支付宝只支持Android端。 </div> |

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|-------|----|-----|--|
| error | | | 调用支付宝支付发生错误时的错误码，请参见 error错误码 。 |

示例

请求示例

```
ZWJSBridge.pay({
  platform:'1',
  arg:{
    "credential":"待支付的订单信息",
    "inSandBox": false
  }
}).then((result) => {
  console.log(result);
}).catch((error) => {
  // error = "8000"
});
```

error错误码

| 错误码 | 描述 |
|-------|-----------------------------------|
| 8000 | 支付正在处理中，支付结果未知，请查询商户订单列表中订单的支付状态。 |
| 40000 | 订单支付失败。 |
| 5000 | 重复请求。 |

| 错误码 | 描述 |
|-------|---------------------------|
| 6001 | 用户中途取消支付。 |
| 60002 | 网络连接出错。 |
| 60004 | 支付结果未知，请查询商户订单列表中订单的支付状态。 |

3.12.9. UI界面类

3.12.9.1. 确认框

调用confirm接口添加确认框，单击相应按钮关闭确认框。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|--------------|---------------|------|--------------------|--------------|
| title | String | 否 | 删除 | 确认框的标题。 |
| message | String | 否 | 确认删除图片吗？数据删除后不可恢复。 | 确认框中的实际消息内容。 |
| buttonLabels | Array[String] | 否 | ["确定","取消"] | 确认框中的可单击按钮。 |

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|------------------|--------|-----|------------------------------|
| data.buttonIndex | Number | 0 | 确认框中可单击按钮的索引值，Number类型，从0开始。 |

示例

您可以在[codesandbox](#)中直接运行调试该接口。

请求示例

```
ZWJSBridge.confirm({
  title: "删除",
  message: "确认删除图片吗？数据删除后不可恢复。",
  buttonLabels: ["确定", "取消"])
.then((data) => {
  // data = {buttonIndex: 0}
}).catch((error) => {
  console.log(error);
});
```

3.12.9.2. 提示框

调用alert接口添加提示框，单击相应的按钮可关闭提示框。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|------------|--------|------|--------------|-------|
| title | String | 否 | 确认删除 | 提示框标题 |
| message | String | 否 | 确认删除图片 吗？ | 消息内容 |
| buttonName | String | 否 | 确定 | 按钮名称 |

示例

请求示例

```
ZWJSBridge.alert({
  title: "提示",
  message: "确认删除图片？",
  buttonName: "确定"})
.then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

3.12.9.3. 弱提示

调用toast接口添加弱提示。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|----------|--------|------|-------|---|
| message | String | 否 | 网络不给力 | 提示信息。 |
| duration | Number | 否 | 2000 | <p>消息显示持续时间，单位毫秒，默认值为2000s。</p> <div style="background-color: #e0f2ff; padding: 5px;"> ? 说明 &nbsp;Android仅支持duration<=2s或者duration>2s。 </div> |
| type | String | 否 | none | <p>根据toast类型展示相应图标，取值：</p> <ul style="list-style-type: none"> • none, 默认值 • success • fail • exception, 值为exception时，必须上传文字信息。 <div style="background-color: #e0f2ff; padding: 5px;"> ? 说明 目前仅小程序支持type参数。 </div> |

示例

请求示例

```
ZWJSBridge.toast({
  message: '提示信息',
  duration: 2000
}).then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

3.12.9.4. 文本输入框

调用prompt接口添加文本输入框。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|---------------|--------|------|--------|---|
| message | String | 否 | 再说一遍？ | 文本框中的实际消息内容。 |
| title | String | 否 | 输入金额 | 文本框的标题。 |
| placeholder | String | 否 | | 占位符，缺省为空。 |
| inputType | String | 否 | number | 取值： • text，缺省值 • number • password |
| cancelButton | String | 否 | 取消 | 按钮名称，默认为取消。 |
| confirmButton | String | 否 | 继续 | 按钮名称，默认为确定。 |

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|------------------|--------|-----|--------------------------|
| data.buttonIndex | Number | 0 | 可单击按钮的索引值，number类型，从0开始。 |
| data.value | | | 输入的值。 |

示例

请求示例

```
ZWJSBridge.prompt({
  title: "输入金额",
  placeholder: "不小于10元",
  message: "再说一遍？",
  inputType: "number",
  cancelButtonTitle: "取消",
  confirmButton: "继续"
}).then((data) =&gt; {
  // data =&gt; {buttonIndex : 0, value : "输入的内容"}
}).catch((error) =&gt; {
  console.log(error);
});
```

错误码

| 错误代码 | 描述 |
|------|------|
| 1 | 未知错误 |

3.12.9.5. 单选列表

调用actionSheet接口添加单选列表。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|--------------|---------------|------|--------------------------|----------|
| title | String | | 谁是最棒哒？ | 单选列表的标题。 |
| otherButtons | Array[String] | | ["孙悟空","猪八戒","唐僧","沙和尚"] | 其他按钮列表。 |
| cancelButton | String | | 取消 | 取消按钮文本 |

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|------------------|--------|-----|--------------------------|
| data.buttonIndex | Number | 0 | 可单击按钮的索引值，number类型，从0开始。 |

| 参数 | 类型 | 示例值 | 描述 |
|------------|----|-----|-------|
| data.value | | | 输入的值。 |

示例

请求示例

```
ZWJSBridge.actionSheet({
  title: "谁是最棒哒？",
  cancelButtonTitle: '取消',
  otherButtons: ["孙悟空", "猪八戒", "唐僧", "沙和尚"]
}).then((data) => {
  // data => {buttonIndex: 0}
}).catch((error) => {
  console.log(error);
});
```

错误码

| 错误代码 | 描述 |
|------|------|
| 1 | 未知错误 |

3.12.9.6. 等待蒙版显示

调用showPreloader接口添加等待蒙版显示。

② 说明

showPreloader与hidePreloader搭配使用，可显示浮层。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|------|--------|------|-----|------------------------|
| text | String | 否 | | Loading显示的字符，空表示不显示文字。 |

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|----------|---------|------|------|--|
| showIcon | Boolean | 否 | true | <p>小程序不支持) 设置是否显示Icon, 默认值为true, 显示Icon。</p> <p>② 说明 小程序中暂不支持该参数。</p> |
| delay | Number | 否 | 0 | <p>设置延迟显示的时长, 单位为毫秒, 默认值为0。</p> <p>如果在延迟显示时长之前调用hidePreloader 则不会显示文字。</p> <p>② 说明 仅小程序中支持该参数。</p> |

示例

请求示例

```
ZWJSBridge.showPreloader({
  text: "使劲加载中..",
  showIcon: true
}).then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

3.12.9.7. 等待蒙版隐藏

调用hidePreloader接口添加等待蒙版隐藏。

示例

请求示例

```
ZWJSBridge.hidePreloader().then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

3.12.9.8. 选择城市

调用selectCity接口选择城市。

② 说明

仅iOS系统支持该接口。

返回数据

| 参数 | 类型 | 示例值 | 描述 |
|----------|--------|--------|------|
| cityName | String | 宁波,余姚市 | 城市名称 |

示例

请求示例

```
ZWJSBridge.selectCity()  
.then((data) =&gt; {  
  // data =&gt; {  
    // cityName: "宁波,余姚市"  
}  
}).catch((error) =&gt; {  
  console.log(error);  
});
```

3.12.9.9. 分享

调用share接口分享内容，分享时系统弹出分享通道选择窗口。

② 说明

分享通道包括微信、钉钉、微博等。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|---------|--------|------|---------|-------|
| title | String | 否 | Dtdream | 分享标题。 |
| content | String | 否 | 阿里巴巴 | 分享内容。 |

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|-------|--------|------|--|------------|
| image | String | 否 | http://www.dtdream.com/r/cms/www/default/img/default/sm-logo-82-54.png | 分享图片的路径。 |
| url | String | 是 | http://www.dtdream.com | 分享链接，不能为空。 |

示例

请求示例

```
ZWJSBridge.share({
  'title':'Dtdream',
  'content':'阿里巴巴',
  'image':"http://www.dtdream.com/r/cms/www/default/img/default/sm-logo-82-54.png",
  'url':"http://www.dtdream.com"
}).then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

错误码

| 错误代码 | 描述 |
|------|-----------|
| 1001 | 分享至不支持的平台 |
| 1002 | 分享失败 |
| -1 | 用户取消 |
| 5 | 未初始化 |

3.12.9.10. 直接分享

调用directShare接口分享内容，支持通过参数指定分享通道。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|---------|--------|------|--|---|
| channel | String | 否 | wechat | 内容分享通道，不区分大小写。取值： <ul style="list-style-type: none">wechat，微信好友。wechat_moments，微信朋友圈。weibo，微博。dingtalk，钉钉。 |
| title | String | 否 | Dtdream | 分享标题。 |
| content | String | 否 | 阿里巴巴 | 分享内容。 |
| image | String | | http://www.dtdream.com/r/cms/www/default/img/default/sm-logo-82-54.png | 分享图片的路径。 |
| url | String | 是 | http://www.dtdream.com | 分享链接，不能为空。 |

示例

请求示例

```
ZWJSBridge.directShare({
  'channel': "wechat",
  'title': "Dtdream",
  'content': "阿里巴巴",
  'image': "http://www.dtdream.com/r/cms/www/default/img/default/sm-logo-82-54.png",
  'url': "http://www.dtdream.com"
}).then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

错误码

| 错误代码 | 描述 |
|------|-----------|
| 1001 | 分享至不支持的平台 |
| 1002 | 分享失败 |
| -1 | 用户取消 |
| 5 | 未初始化 |

3.12.10. 请求类

3.12.10.1. 无线网关

调用无线网关egop接口向应用开放平台发送网络请求。

请求参数

| 参数 | 类型 | 是否必选 | 示例值 | 描述 |
|--------|------------|------|-----------------|----------------------|
| api | String | 是 | egop.zjzw.demo1 | 业务方mgop api名字。 |
| method | String | 是 | method | 网络请求类型，取值为POST或者GET。 |
| header | JsonObject | 否 | | 网络请求需要附加的header。 |
| param | JsonObject | 否 | | 网络请求附加的参数。 |

返回数据

返回数据为对应egop后端业务方接口返回的Response Body部分，返回数据类型为JSONObject。

示例

请求示例

```
ZWJSBridge.egop({
  api: 'egop.zjzw.demo1',
  method: 'GET' // POST, GET,
  header: {
    "header1": "value1"
  },
  param: {
    "keys": "value1"
  }
}).then((result) =>{
  console.log(result);
}).
catch((error) =>{
  console.log(error);
});
});
```