CEC
2.5.0.0.0

用户接入——网页版轻量级客户端集成（JS）

文档版本	01
发布日期	2024-03-01

华为技术有限公司

版权所有 © 华为技术有限公司 2024。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

HUAWEI和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： https://www.huawei.com

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
https://www.huawei.com/cn/psirt/vul-response-process
如企业客户须获取漏洞信息，请参见如下网址：
https://securitybulletin.huawei.com/enterprise/cn/security-advisory

目 录

1 集成轻量级 WEB 聊天控件（引入 Token 认证方式）.....1
1.1 概述.....1
1.2 集成原理.....2
1.3 集成步骤.....3
1.4 开发前准备.....4
1.4.1 资源准备.....4
1.4.2 在云客服上开通在线客服功能.....6
1.5 集成开发.....16
1.5.1 开发 token 生成机制与 token 认证接口.....16
1.5.2 开发集成页面.....18
1.5.2.1 核心代码简析.....18
1.5.2.2 开发样例：JS 页面集成样例.....20
1.6 测试与验证.....22
1.7 常见问题.....28
1.7.1 xmlHttpRequest 请求云客服的 URL，报跨域问题.....28
2 集成轻量级 WEB 聊天控件（引入 Authorization 认证方式）.....31
2.1 概述.....31
2.2 集成原理.....32
2.3 集成步骤.....32
2.4 开发前准备.....33
2.4.1 资源准备.....33
2.4.2 在云客服上开通在线客服功能.....35
2.5 集成开发.....46
2.5.1 开发 Authorization 签名生成机制.....46
2.5.2 开发集成页面.....50
2.5.2.1 核心代码简析.....50
2.5.2.2 开发样例：JS 页面集成样例.....52
2.6 测试与验证.....54
2.7 常见问题.....59
2.7.1 xmlHttpRequest 请求云客服的 URL，报跨域问题.....60

1 集成轻量级 WEB 聊天控件（引入 Token 认证方式）

1.1 概述
1.2 集成原理
1.3 集成步骤
1.4 开发前准备
1.5 集成开发
1.6 测试与验证
1.7 常见问题

1.1 概述

轻量级web聊天控件是一款能快速集成到网站的在线客服用户端产品，**当前仅支持PC端WEB集成**。选择我们的产品，您可以在云客服 SaaS（Software as a Service）系统场景下，快速构建智能在线客服系统，实现与用户在网页进行在线聊天，您可以通过**图1-1**和**图1-2**了解我们的产品界面。

轻量级WEB聊天控件提供的网页与客户在线聊天的功能，具有如下特点：

- 轻量级，容易集成到您的网页，不占用门户和工作台的主要页面。
- 易于操作，您的用户能轻松从页面找到客服窗口，并在界面直接与在线座席进行交谈。

说明

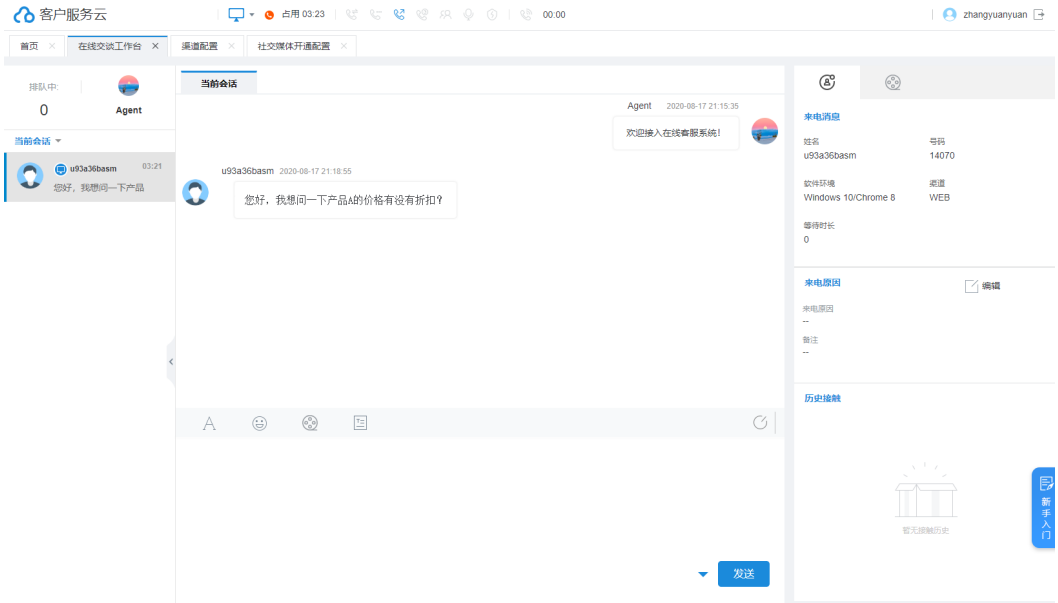
使用手册前，请您先了解我们提供的两种用户认证方式：

1. Token认证方式：对用户有认证要求，适用于认证接入用户的场景。
2. authorization认证方式：游客模式，不认证接入用户的场景。

请依据您的需求选择对应的开发指南，如您需要使用authorization认证方式，请进入**2 集成轻量级WEB聊天控件（引入Authorization认证方式）**后继续阅读。

图 1-1 在线客服客户端窗口

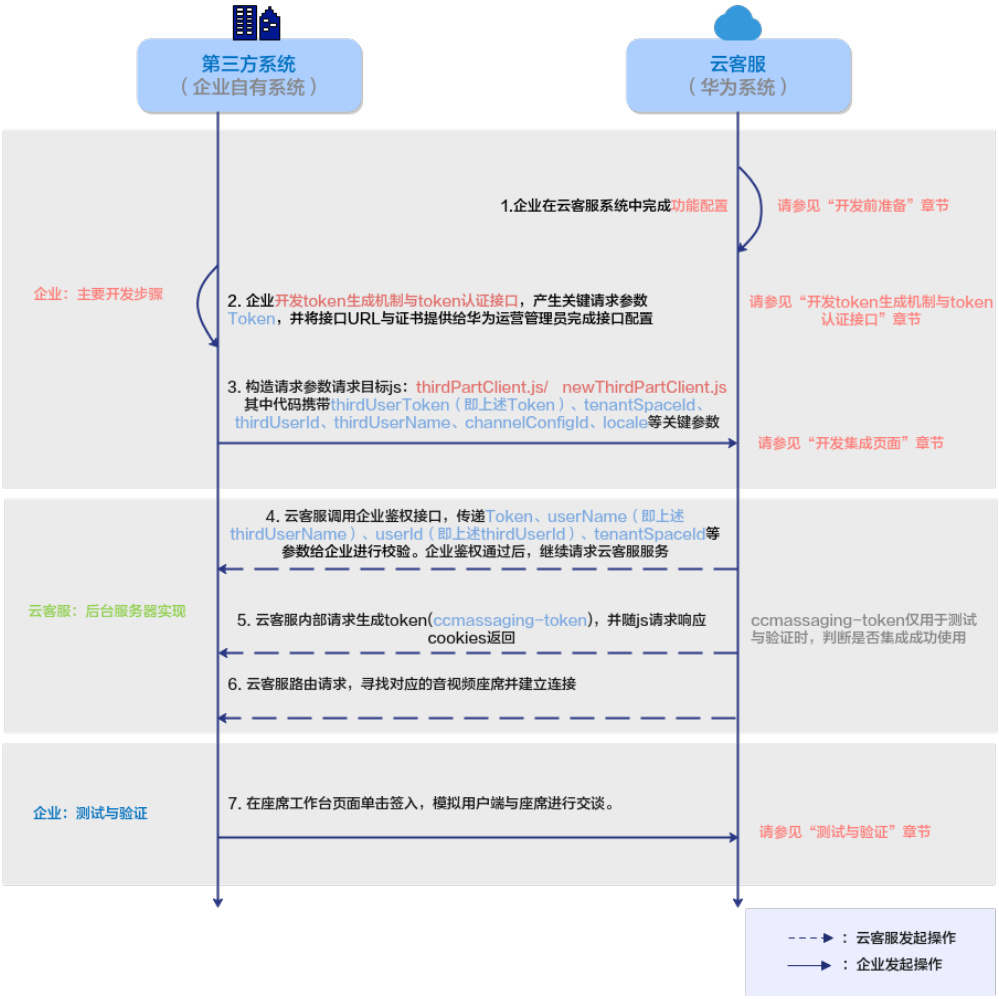
图 1-2 在线客服座席端界面



1.2 集成原理

如果您期望了解我们的集成原理，请参见**图1-3**。

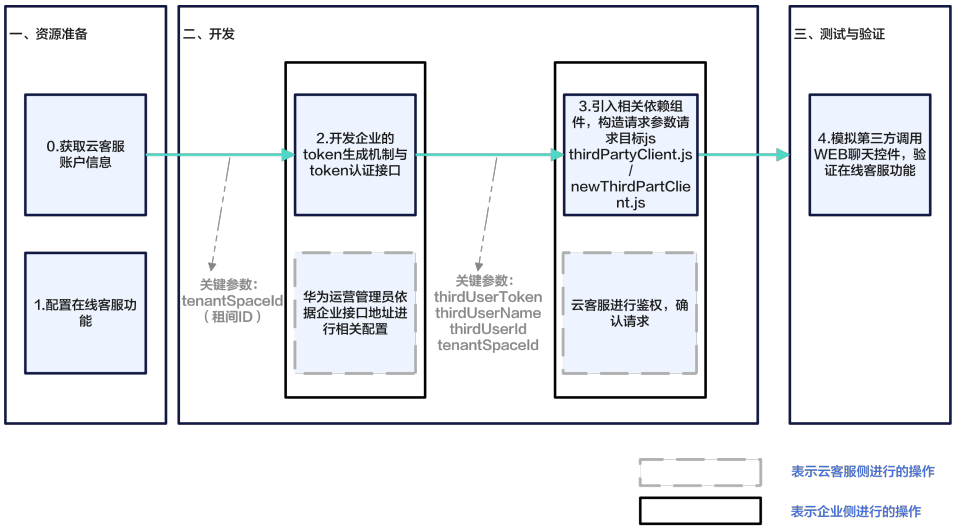
图 1-3 Web 聊天控件 token 认证方式集成原理



1.3 集成步骤

轻量级WEB聊天控件可以快速高效的集成到您的页面，请您根据**图1-4**进行集成开发。

图 1-4 集成开发步骤



步骤1 完成开发前资源准备，获取登录信息，配置在线客服功能，请参见**1.4 开发前准备**。

步骤2 开发token生成机制与token认证接口，提供给云客服。云客服在鉴权通过后，会将Token发回给第三方系统，由第三方系统确认是否为本系统发的请求，请参见**1.5.1 开发token生成机制与token认证接口**。

须知

华为方使用HTTPS传输协议确保信息传输通道安全，请第三方确保开发的鉴权功能具备“口令复杂度校验”、“防暴力破解”、“防DOS攻击”等安全防护能力。

步骤3 引入相关依赖框架，构造请求参数请求thirdPartyClient.js和newThirdPartyClient.js，将Token信息、租户ID、企业侧的用户ID、用户名称等信息返回云客服，请求鉴权，鉴权通过后可实现云客服网页聊天能力的轻量级接入，请参见**1.5.2 开发集成页面**。

步骤4 测试与验证WEB聊天控件功能，通过客户端窗口发起在线聊天检查与座席聊天功能，请参见**1.6 测试与验证**。

----结束

1.4 开发前准备

此章节主要描述在集成开发前，需要在云客服中完成的准备工作。

1.4.1 资源准备

在集成前，请您完成以下资源准备：

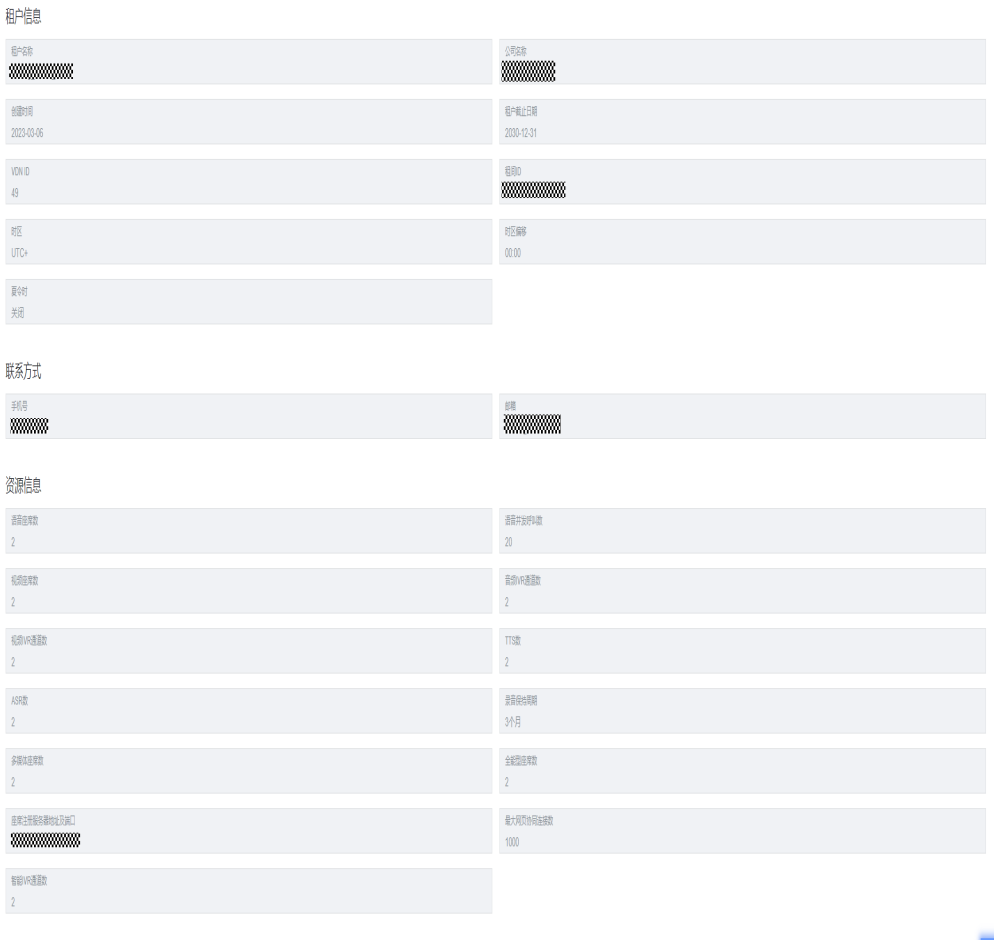
1. 您已经向云客服申请了租户信息，系统运维管理员已经为您添加了租户信息，并将如下信息提供给您：

表 1-1 参数信息

参数名	说明
租间ID	创建租间（即您的云客服）成功后，系统生成的租间ID。 您也可以登录您的租间后，选择“配置中心 > 系统管理 > 租户基本信息”，查看您的租间ID。
租户管理员账号	登录云客服的账号。
租户登录密码	登录云客服的密码。

2. 用租户管理员账号密码登录您的云客服，在登录界面完成初始密码修改。
3. 确认您的租间已经具有多媒体座席特性，检查方法如下：
   登录租间，选择“配置中心 > 系统管理 > 租户基本信息”。
   检查多媒体座席数，如果数量为0，表示没有多媒体座席特性，请联系华为运营管理员开通。

图 1-5 多媒体座席特性检查



4. （可选）如果需要开通点击通话功能（在与座席进行文字在线交谈时，希望能与座席通过语音电话或视频沟通），请联系华为运营管理员开通点击通话功能。

5. （可选）如果需要开通网页协同功能（在与坐席进行文字在线交谈时，希望能与坐席网页画面的共享、标记等操作），请联系华为运营管理员开通网页协同功能。
6. 获取云客服提供的JS文件地址，如下所示。请根据实际的云客服域名替换域名部分。

   https://***servicestage.besclouds.com***/service-cloud/webclient/chat_client/js/thirdPartyClient.js?&t=***1595993533588***

1.4.2 在云客服上开通在线客服功能

步骤1 以租户管理员角色登录云客服。

步骤2 添加多媒体类型的技能队列。

登录后，选择“配置中心 > 员工中心 > 技能队列”，多媒体技能队列配置页面请参见图1-6。如需配置点击通话功能，请同时完成点击通话技能队列的配置，参见图1-7。

步骤3 单击“创建”，完成参数配置。具体参数描述请参见表1-2。

图 1-6 配置多媒体技能队列界面

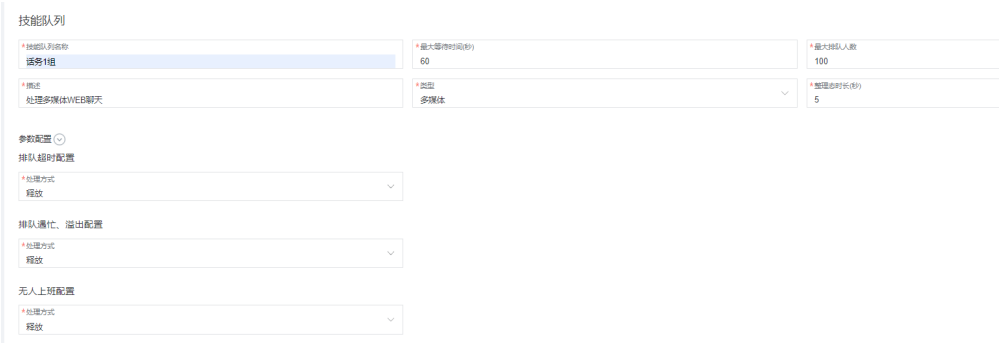


图 1-7 配置点击通话技能队列界面（可选）

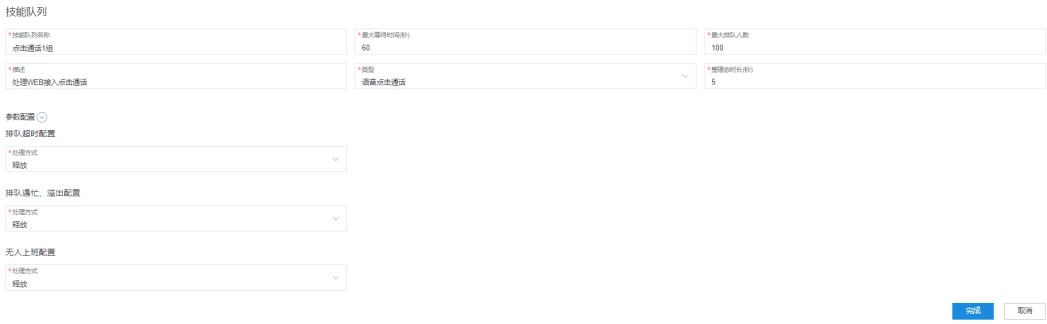


表 1-2 技能队列配置的参数属性

参数名	是否必填	说明
技能队列名称	True	字符串长度<=20，不能使用空格。

参数名	是否必填	说明
最大等待时间（秒）	True	默认：60 单位：秒 输入范围：1-60000
最大排队人数	True	默认：100 输入范围：1-10000
描述	True	字符串长度<=50。
类型	True	取值范围： ● 语音：专门处理语音类业务 ● 多媒体：专门处理多媒体业务 ● 视频：专门处理视频业务 ● 语音点击通话：协同多媒体业务使用，提供客户与座席进行文字交谈时，直接与座席进行音频通话的能力。 ● 视频点击通话：协同多媒体业务使用，提供客户与座席进行文字交谈时，直接与座席进行视频通话的能力。 **说明** 点击通话类型的技能队列目前仅用于WEB渠道。 默认：语音
整理态时长（秒）	True	当通话结束后，座席进入整理态的时长，默认为5秒。 超过该值，座席进入空闲状态，可以接听客户来电。 输入范围：0-3600

参数名	是否必填	说明
参数配置	False	个性化配置，用于设置当客户主动呼入该技能队列时遇到无法正常接通的情况下的处理策略。取值范围： ● 排队超时配置 – 处理方式：客户呼入后无空闲座席接听导致排队超时后的处理策略。 ▪ 释放（默认） ▪ 转接 – 设备类型：处理策略为转接时，需要配置转接的技能队列或者IVR流程。 ▪ 技能队列 ▪ IVR ● 排队遇忙、溢出配置 – 处理方式：客户呼入后无空闲座席接听导致需要排队等待，或者排队人数超出限制后的处理策略。 ▪ 释放（默认） ▪ 转接 – 设备类型：处理策略为转接时，需要配置转接的技能队列或者IVR流程。 ▪ 技能队列 ▪ IVR ● 无人上班配置 – 处理方式：客户在无座席值班时间内呼入导致无人接听的处理策略。 ▪ 释放（默认） ▪ 转接 – 设备类型：处理策略为转接时，需要配置转接的技能队列或者IVR流程。 ▪ 技能队列 ▪ IVR ● 排队等待配置：客户呼入后需要排队等待时，可以播放语音，优化客户等待过程。 – 排队等待方式 ▪ 默认等待音

参数名	是否必填	说明
		▪ 自定义等待音 ▪ IVR ● 保持等待配置：在交谈过程中需要保持通话，让客户等待时，可以播放语音，优化客户等待过程。 – 保持等待方式 ▪ 默认等待音 ▪ 自定义等待音 ● 业务代表应答方式配置：座席接听客户来电后，支持配置播报工号通知客户所属座席。 – 业务代表应答方式 ▪ 报工号 ▪ 不报语音 **说明** 语音、视频、点击通话类型技能队列支持转接IVR和技能队列，多媒体类型的技能队列仅支持转接技能队列。 转接的技能队列类型要与创建的技能队列类型一致。 等待音配置目前仅支持语音、视频类型的技能队列，点击通话类型的技能队列该配置不展示。

1. 单击“完成”完成配置。

步骤4 添加多媒体类型的被叫配置。

1. 选择“配置中心 > 接入配置 > 被叫配置”，进入配置页面。
2. 单击“新增”，为虚拟呼叫中心添加图1-8参数信息，具体参数说明请参考表1-3，并单击“完成”完成配置。如需开启点击通话功能，请在此处同时配置点击通话类型的被叫配置。

图 1-8 新增界面

表 1-3 被叫配置的参数属性

参数名	是否必填	说明
接入码	True	面向客户提供的客服热线，客户通过拨打接入码接通到座席侧。 单击"⊕"选择接入码时，弹出框中展示接入码的类型，选择“多媒体”类型的接入码。
扩展码	False	如果希望一个接入码能设置多个目的设备，可通过配置扩展码的方式来实现。 例如，接入码为12345，我们可以添加扩展码1，路由到技能队列A，扩展码2，路由到技能队列B。这样客户可直接拨打123451实现直接接入技能队列A。
设备类型	True	请选择“技能队列”，完成技能队列的被叫配置。
技能队列	True	关联**步骤3**中已创建的技能队列。 技能队列：单击"⊕"选择技能队列时，弹出框中供选择的技能队列的类型与“接入码”的类型一致，例如“接入码”为“多媒体”类型，那么可供选择的技能队列全部为“多媒体”类型。

步骤5 配置业务账号以及技能队列。

1. 选择“配置中心 > 员工中心 > 座席管理”，进入管理座席页签。
2. 选择一条座席工号信息，操作栏单击“配置”，进入座席信息配置页面。
3. 关联对应座席的业务账号以及技能队列。如需开启点击通话功能，请在此处同时关联点击通话的技能队列。

图 1-9 座席信息配置

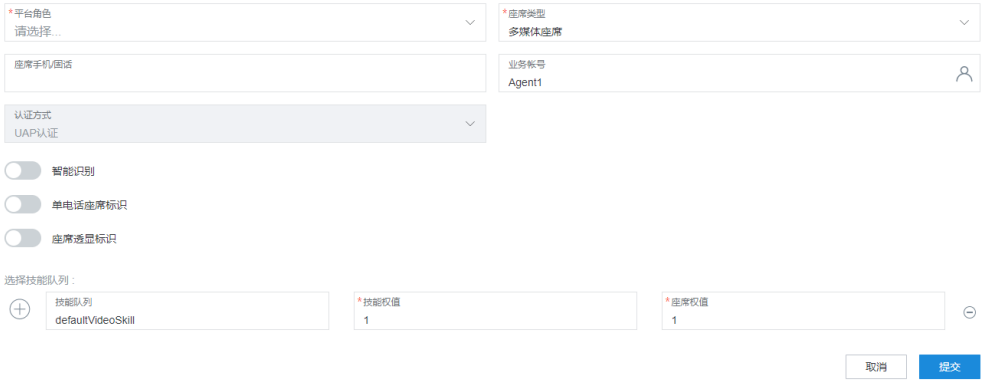


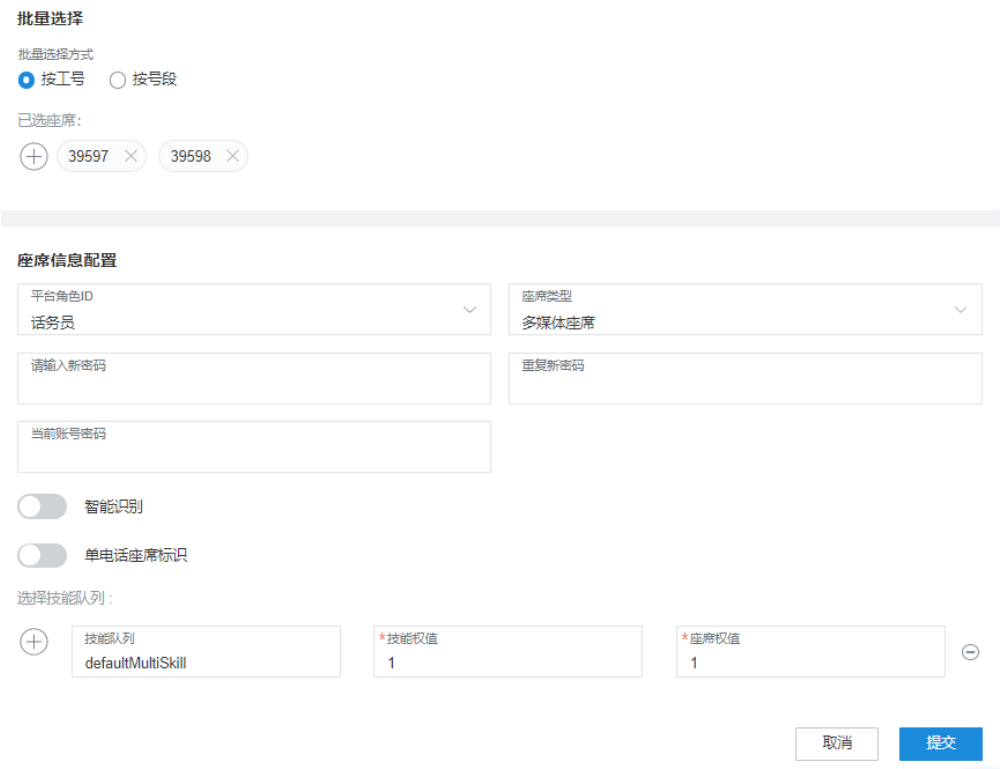
表 1-4 座席信息配置的参数属性

参数名	是否必填	说明
平台角色	True	必填，选择该座席所属角色。 – 话务员：对客户的来话进行接听和转接。 – 质检员：对普通话务员与客户的通话进行干预，例如插入、拦截、强制示忙等，用于对话务员处理来话的表现做辅导和监督。 – 外呼座席：对客户的来话进行接听、转接，不接来话。
座席类型	True	必填，选择该座席类型，可以处理的业务类型。 – 音频座席 – 视频座席 – 多媒体座席 – 全能型座席
座席手机/固话	False	配置座席使用的手机或者固定电话。
业务帐号	True	配置已创建的人员账号。 具体可参考管理人员章节。
智能识别	False	默认关闭，配置座席是否为智能座席。智能座席除了基本的语音控制功能外，支持实时的语音转文本，以及相关的智能推荐功能。当您开启智能识别开关时，需要保证开启智能座席的数量不能超过租户开通时分配的智能座席数。
单电话座席标识	False	开启这个特性后，座席拨指定接入码就可以进一个IVR流程， 根据IVR提示按某个键就能输入工号、密码签入，然后就可以用手机接电话。当您开启开关时，需要系统运维人员给租户基于平台定制单电话座席流程，并由租户提供号码资源，供单电话座席流程接入，才可正常使用该功能。
座席透显标识	False	用于第三方标记座席是否具有透显特性，不是功能特性开关，仅作为一个标签。透显特性是指座席可以自定义用户侧主叫显示号码（用户看到的主叫号码）与座席侧主叫显示号码（客户经理看到的主叫号码）。

参数名	是否必填	说明
选择技能队列	True	配置座席的技能队列。如果需要添加多个技能队列，那么需要保证添加的所有技能队列的媒体类型一致，全能型座席除外，例如都为“音频”类型或者都为“多媒体”类型。 **说明** – 如果座席类型选择“视频座席”，您在申请租户资源时，需要申请对应的视频座席数。 – 如果座席类型选择“多媒体座席”，您在申请租户资源时，需要申请对应的多媒体座席数。 – 如果座席类型选择“全能型座席”，您在申请租户资源时，需要申请对应的全能型座席数。 – 如果业务账号不够用，请到“配置中心 > 员工中心 > 员工”下添加，此处不赘述。

4. 单击提交，为座席工号分配业务账号和技能队列完成。
5. （可选）单击“批量配置”，进入座席信息配置页面，进行批量配置。

图 1-10 批量配置



– 批量选择：可选择按工号或者按号段的方式选择需要配置的座席。
– 座席信息配置：可参考**步骤5**进行配置。

步骤6 配置WEB渠道。

1. 选择“配置中心 > 接入配置 > 渠道配置”。

2. 单击“新增”，输入渠道接入编码，选择“WEB”渠道，然后单击“下一步”，进入WEB渠道的配置页面。

说明

渠道接入编码必须是唯一的，输入值只能由字母，数字或下划线组成（首字符只能用字母和下划线）。

3. 根据您的需求填写WEB渠道信息，请参见图1-11，具体参数说明请参见表1-5。如需开启点击通话功能，请在该页面配置“点击通话被叫配置”与“点击通话技能队列”。

图 1-11 WEB 渠道配置页面

表 1-5 WEB 渠道配置参数

参数	说明
公共配置	
信息配置	需要配置以下参数： – 技能队列：配置框会查询当前租间的所有多媒体被叫配置，用户酌情选择。 – 转人工关键字：用户在H5客户端输入后该关键字后会由机器人服务转为人工座席服务。 **说明** 若不开启智能机器人，转人工关键字可不设置。 – 点击通话被叫配置：配置框会查询当前租间的所有音视频座席和IVR。 – 击通话技能队列：“点击通话被叫配置”中选择音视频座席后，需要配置相关技能队列。 – 点击获取IVR接入码：“点击通话被叫配置”中选择IVR后，需要配置IVR接入码。

参数	说明
会话结束方式	默认关闭，若不开启，默认20分钟超时，打开后需要配置以下参数： – 客户未回复提醒时间间隔：用户在客户端无会话达到该时间后会断开连接。 – 会话结束语：会话断开后结束提醒。
第三方认证密钥	默认关闭。开关开启，需要输入secretKey。当采用Authorization认证方式时，需要配置此参数。
座席工作时间	– 工作日：可配置0-24小时间最多4个工作时间段。默认展示一个时间段，单击新增，可增加时间段配置。 – 非工作日：可配置0-24小时间最多4个工作时间段。默认展示一个时间段，单击新增，可增加时间段配置。 – 非工作时间提醒信息：不在非工作日配置的时间段内，客户呼入时，提醒信息。
排队提醒	需要配置以下参数： – 可配置排队提醒间隔（秒）：必填，默认为10秒。 – 排队提醒内容：自定义客户呼入需要排队时的提醒内容。
机器人设置	默认关闭。开关开启，则默认先接入机器人进行对话。需配置以下参数： – 更换头像：可更换机器人头像。 – 名字：为机器人添加名字。 – 性别：为机器人添加性别。 – 机器人接入码：从智能IVR中选择配置好的机器人接入码。 – 默认回复语：当机器人不能识别客户语意，自定义回复语。 – 超时回复语：当客户会话超时，自定义超时回复语。 – 转人工提示语：自定义转人工成功提示语 **说明** 如果您想开启机器人设置，实现机器人对话功能，请参考**配置智能IVR**章节，结合您所需要的机器人类型进行配置。
机器人助手配置	默认关闭。开关开启，则在座席侧开启机器人助手。 – 机器人助手接入码：从智能IVR中选择配置好的机器人接入码。 – 静默座席技能队列：配置框会查询当前租间的所有多媒体被叫配置，用户酌情选择。

4. 单击保存并进行下一步，进入到集成说明页面。

步骤7 （可选）在“集成说明”页签，单击“试一试”，可以在弹出页面中设置客户信息，模拟客户端的会话窗口。

您可以测试客户通过当前渠道与座席或者机器人交谈。

1. 单击“试一试”，在弹出页面的右下角单击“ ”，弹出“在线客服”弹出框。在线客服有两种交谈模式：
   a. 当“接入智能机器人”开关开启后，默认接入机器人，客户与机器人交谈；当客户输入的交谈内容存在机器人可识别的关键词，机器人会识别关键词并进行答复。
   b. 当“接入智能机器人”开关未开启时，对话框自动接入座席，用户可以与在线座席进行交谈。此时需要在云客服页面点击“签入”，签入多媒体座席，然后选择“在线交谈工作台”，弹出当前会话的工作台，完成接入后可进行在线交谈。

   说明

   当您与机器人交谈时，您可以通过单击“转人工”按钮由机器人转到客服座席，也可以通过输入包含“转人工关键词”的内容由机器人转到客服座席（“转人工关键词”可在**步骤6**中的页面配置）。但是，当您与客服座席交谈时，不能转人工或者转机器人。
2. 在弹出的“在线客服”对话框中，客户输入与机器人或客服座席的交谈内容，单击“发送”，测试机器人或客服座席的答复。
3. （可选）在对话框附近单击“ ”，然后单击“评价”，你可以对当前客服的服务进行评价，如**图1-12**所示，包含星级满意度分数评价和内容评价，单击“确认”。

图 1-12 评价界面

说明

用户可以对进行过会话互动的客服座席进行满意度评价。客户在会话过程中和会话结束后，均可以对客服座席的服务主动发起满意度评价，随时可以评价，以最后一次提交结果为准。

----结束

渠道配置完成后，请记录该渠道的配置ID，在开发认证验证接口以及页面集成时会用到该值。

配置ID	渠道接入编码	渠道类型	绑定技能队列	操作
869525871671777281	WECHAT001	WECHAT	defaultMediaSkill (10001+)	修改 删除
853180664420335631	wechat001	WECHAT	defaultMediaSkill (10001+)	修改 删除
827635713993441281	hyy	WECHAT	defaultMediaSkill (10001+)	修改 删除
817854979183314945	WEBCHAT	WEB	defaultMediaSkill (10001+)	修改 删除
743032024951966724	A20191231	WECHAT	defaultMediaSkill (10001+)	修改 删除
740930049837911041	a123456	WECHAT	defaultMediaSkill (10001+)	修改 删除

1.5 集成开发

此章节主要描述如何在您的系统的完成我们WEB聊天控件的集成。

1.5.1 开发 token 生成机制与 token 认证接口

您需要开发一个token生成机制和token认证接口，提供给云客服用于鉴权等功能。云客服收到您的js请求时会调用该接口，将企业token、租间ID、userAccount发给您的系统进行确认，确认后再进行内部校验和请求。

说明

我们提供了token生成机制与token认证接口的demo供您参考，可以在我们的论坛（https://bbs.huaweicloud.com/forum/thread-192048-1-1.html）进行下载阅读。Demo主要提供算法和校验逻辑的思路，实际使用的时候请使用安全算法。该Demo不能直接用于生产，仅限DEMO演示，请依据您系统的安全要求进行合理设计。

操作步骤

步骤1 开发token生成机制。请您依据您系统的特性与对安全的要求进行自行开发。

步骤2 请根据云客服的对认证验证接口的规格，开发认证验证接口。您可以通过**表1-6**和**表1-7**了解各请求参数与返回参数属性要求，具体如下：

POST http(s)://ip:port/rest/cc-messaging/v1/thirdparty/chatThirdPartyValidate(具体rest路径可以由企业进行定制)。

说明

HTTP协议是不安全协议，可能会对系统带来风险，不推荐使用。建议使用安全的HTTPS协议。若使用HTTPS协议，还需要自备cer格式的证书，用于验证HTTPS网站的合法性。

表 1-6 Requestbody 说明

参数名	参数类型	参数位置	是否必填	说明
Token	string	body	True	企业/伙伴侧生成的校验信息。
userName	string	body	True	企业/伙伴侧提供的用户名称。 须在企业的鉴权系统中已经存在。
userId	string	body	True	企业/伙伴侧提供的用户ID。 须在企业的鉴权系统中已经存在。
tenantSpaceId	string	body	True	云客服侧提供的租间ID。请在“配置中心 > 系统管理 > 租户基本信息”中查看。

- Example

```
{
   "Token":"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
   "userName":"Tom",
   "userId":"10010001"
   "tenantSpaceId":"202007159031"
 }
```

说明

上述用户名、用户ID为样例，请确保在企业客户真实存在。

表 1-7 Response 说明

参数名	参数类型	参数位置	是否必填	说明
retCode	int	body	True	接口结果标识符。 0：成功 其它：失败
message	string	body	True	结果信息

- Example

```
{
   "retCode":0,
   "message":"validate success"
}
```

步骤3 该鉴权接口的URL和证书文件（使用HTTPS协议需要）请您交给云客服的运营管理员，由运营管理员在云客服中进行配置。

----结束

1.5.2 开发集成页面

1.5.2.1 核心代码简析

前提条件

华为云运营管理员已经在云客服完成鉴权接口相关配置。

操作步骤

步骤1 请先完成JS依赖框架的引入，在以下的样例中，需要引入jquery和axios组件，版本请参考Jquery 1.8.0.js和Axios.min.js，如果需要使用网页协同功能，请引入cobrowse.js和cobrowseCommon.js组件或发布插件给用户使用（插件开发参考3.1：cobrowse插件开发、使用指导），cobrowse.js和cobrowseCommon.js采用cdn方式引入，其中ip:port需替换成实际云客服的ip和port，或者直接使用域名，如：https://www.test.com/service-cloud/resource.root/cobrowse/sdk/cobrowse.js，引用代码示例如图1所示（注：1、网页协同功能不支持打开新的页签后在新页签上进行协同操作。2、引入网页协同的cobrowse.js和cobrowseCommon.js组件时，第三方页面中包含多层iframe嵌套时，仅引入了js组件的iframe内可以进行网页协同，若该iframe存在父iframe框架，则无法在父iframe中进行网页协同操作）。

图 1-13 引用代码示例

```
<script type="text/javascript" src="https://ip:port/service-cloud/resource.root/cobrowse/sdk/cobrowse.js"></script>
<script type="text/javascript" src="https://ip:port//service-cloud/webclient/chat_client/js/cobrowseCommon.js"></script>
<script type="text/javascript" src="js/jquery-1.8.0.js"></script>
<script type="text/javascript" src="js/axios.min.js"></script>
```

cobrowse.js和cobrowseCommon.js采用cdn方式引入，其中ip:port需替换成实际云客服的ip和port，或者直接使用域名，如：https://www.test.com/service-cloud/resource.root/cobrowse/sdk/cobrowse.js

步骤2 获取云客服提供的JS文件地址，如下所示。请根据实际的云客服域名替换域名部分。

https://***servicestage.besclouds.com***/service-cloud/webclient/chat_client/js/thirdPartyClient.js?&t=***1595993533588***

步骤3 在企业侧使用ajax请求集成云客服侧的JS脚本，下面为JS代码片段，片段中的变量含义见下表。

```
  //定义变量$aicc_ContextPath，必须定义名称为$aicc_ContextPath，内容为https://ip:port或域名/service-cloud/
  const $ContextPath = "https://ip:port/service-cloud"
  // 构造请求参数
  let timestamp = new Date().getTime();
  let serviceUrl = $ContextPath+ "/webclient/chat_client/js/thirdPartyClient.js?"+"&t=" + timestamp;
  // 以下变量在使用前请先定义,本demo在后续代码axios.js引用时进行的定义。详细参数说明请参见表1-8
  let thirdUserData = {};
  thirdUserData['thirdUserToken'] ="XXXXXXXX";
  thirdUserData['thirdUserName'] = "XXXXXXXX"
  thirdUserData['thirdUserId'] = "XXXXXXXX";
  thirdUserData['tenantSpaceId'] = "XXXXXXXX";
  thirdUserData['channelConfigId'] = "XXXXXXXX";
  thirdUserData['locale'] = "xx";
  thirdUserData['mapService'] = 'google';

  var importScript = (function (oHead) {
     function loadError(oError) {
        throw new URIError("The script " + oError.target.src + " is not accessible.");
     }
```

```
        return function (sSrc, fOnload) {
            var oScript = document.createElement("script");
            oScript.type = "text\/javascript";
            oScript.onerror = loadError;
            if (fOnload) { oScript.onload = fOnload; }
            oHead.appendChild(oScript);
            oScript.innerHTML = sSrc;
        }

    })(document.head || document.getElementsByTagName("body")[0]);

    function thirdValidate(thirdUserData) {
        let timestamp = new Date().getTime();
        let serviceUrl = $ContextPath + "/webclient/chat_client/js/newThirdPartyClient.js?"+"&t=" + timestamp;
    // 请求newThirdPartyClient.js
        var request = $.ajax({
            url: serviceUrl,
            type: "POST",
            data: JSON.stringify(thirdUserData),
            crossDomain: true,
            dataType:"text",
            xhrFields: {withCredentials: true},
            error: function (XMLHttpRequest, textStatus, errorThrown) { console.log(XMLHttpRequest.status);
console.log(XMLHttpRequest.readyState); console.log(textStatus); },
            success: function (data) {
                importScript(data)
            }
        });
    }

    applyChatWindow();
    function applyChatWindow() {

        thirdUserData['timestamp'] = timestamp;
        // 请求service-cloud获取JS文件
        axios({
            method: 'post',
            url: serviceUrl,
            data: JSON.stringify(thirdUserData),
            withCredentials: true
          })
          // 定义JS文件变量，并写入到当前网页中
          .then(resp => {
          if (resp && resp.status === 200) {
              let str = 'var configId = "' + thirdUserData.channelConfigId + '";var userId = "' +
thirdUserData.thirdUserId + '";var userName = "' + thirdUserData.thirdUserName
                    + '";var tenantSpaceId = "' + thirdUserData.tenantSpaceId + '";var locale = "' +
thirdUserData.locale
                    + '";var token="'+ thirdUserData.thirdUserToken+'";';
              importScript(str+resp.data);
          } else {
            this.$alert("链接失败！");
            console.log(resp.status);
            console.log(resp);
          }
        });
    }
```

----结束

表 1-8 集成页面参数说明

参数名	是否必填	说明
$ContextPath	True	**步骤2**中请求地址的可变部分，通常为域名，或IP地址端口号，请将ip:port换成AICC的公网域名。 建议先将域名或IP地址端口号配置到配置文件或配置表中，$ContextPath通过获取配置项的方式赋值。
serviceUrl	True	即您在**步骤2**中获取的完整路径。 通常serviceUrl不会直接写全量地址，而是通过可变部分*$ContextPath*+固定部分构成。无论采用哪种方式，serviceUrl 的值请确保与云客服提供的请求的地址保持一致。
thirdUserToken	True	企业原生的校验信息，与企业认证验证接口中的**Token**一致。
tenantSpaceId	True	租间ID，请在“配置中心 > 系统管理 > 租户基本信息”中查看。
thirdUserId	True	企业用户ID。与**表1-6**中userId保持一致。
thirdUserName	True	企业用户名。与**表1-6**中userName保持一致。
channelConfigId	True	渠道配置ID。完成**1.4.2 在云客服上开通在线客服功能-步骤7**后，请在“配置中心 > 接入配置 > 渠道配置”中查看。
locale	True	云客服提供的租间语言信息。 ● zh：中文 ● en：英文
mapService	False	用户地图服务。 ● tencent ● google 缺省值：tencent **说明** 腾讯地图暂不支持国外位置。

1.5.2.2 开发样例：JS 页面集成样例

环境要求	-
引用库	Jquery 1.8.0.js/Axios.min.js
集成样例	**index.html**

须知

- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

index.html

```
<!DOCTYPE html>
<!--
  Tutorial.
  Check available devices.
-->
<html>

<head>
   <meta http-equiv="x-ua-compatible" content="IE=edge">
   <meta charset="UTF-8">
   <title>在线客服</title>
   <!-- 如果需要使用网页协同功能，需引入cobrowse.js，cobrowseCommon.js这两个js，其中ip:port需替换成实际云客服的ip和port，或者直接使用域名 -->
   <script type="text/javascript" src="https://ip:port/service-cloud/resource.root/cobrowse/sdk/cobrowse.js"></script>
   <script type="text/javascript" src="https://ip:port//service-cloud/webclient/chat_client/js/cobrowseCommon.js"></script>
   <script type="text/javascript" src="js/jquery-1.8.0.js"></script>
   <script type="text/javascript" src="js/axios.min.js"></script>
</head>

<body>
   <!-- Check that browser is not IE -->
   <script>
      var ua = window.navigator.userAgent;
      if (ua.indexOf('MSIE ') > 0 || ua.indexOf('Trident/') > 0) {
         alert("Internet Explorer is not supported. Please use Chrome or Firefox");
      }
   </script>
   <script>
   //定义变量$aicc_ContextPath，必须定义名称为$aicc_ContextPath，内容为https://ip:port或域名/service-cloud/
   const $ContextPath = "https://ip:port/service-cloud"
   // 构造请求参数
   let timestamp = new Date().getTime();
   let serviceUrl = $ContextPath+ "/webclient/chat_client/js/thirdPartyClient.js?"+"&t=" + timestamp;
   // 以下变量在使用前请先定义,本demo在后续代码axios.js引用时进行的定义。详细参数说明请参见表1-8
   let thirdUserData = {};
   thirdUserData['thirdUserToken'] ="XXXXXXXX";
   thirdUserData['thirdUserName'] = "XXXXXXXX"
   thirdUserData['thirdUserId'] = "XXXXXXXX";
   thirdUserData['tenantSpaceId'] = "XXXXXXXX";
   thirdUserData['channelConfigId'] = "XXXXXXXX";
   thirdUserData['locale'] = "xx";

   var importScript = (function (oHead) {
      function loadError(oError) {
         throw new URIError("The script " + oError.target.src + " is not accessible.");
      }
      return function (sSrc, fOnload) {
         var oScript = document.createElement("script");
         oScript.type = "text\/javascript";
         oScript.onerror = loadError;
         if (fOnload) { oScript.onload = fOnload; }
         oHead.appendChild(oScript);
         oScript.innerHTML = sSrc;
```

```
        }

    })(document.head || document.getElementsByTagName("body")[0]);

    function thirdValidate(thirdUserData) {
        let timestamp = new Date().getTime();
        let serviceUrl = $ContextPath + "/webclient/chat_client/js/newThirdPartyClient.js?"+"&t=" + timestamp;
    // 请求newThirdPartyClient.js
        var request = $.ajax({
            url: serviceUrl,
            type: "POST",
            data: JSON.stringify(thirdUserData),
            crossDomain: true,
            dataType:"text",
            xhrFields: {withCredentials: true},
            error: function (XMLHttpRequest, textStatus, errorThrown) { console.log(XMLHttpRequest.status);
console.log(XMLHttpRequest.readyState); console.log(textStatus); },
            success: function (data) {
                importScript(data)
            }
        });
    }

    applyChatWindow();
    function applyChatWindow() {

        thirdUserData['timestamp'] = timestamp;
        // 请求service-cloud获取JS文件
        axios({
            method: 'post',
            url: serviceUrl,
            data: JSON.stringify(thirdUserData),
            withCredentials: true
          })
          // 定义JS文件变量，并写入到当前网页中
          .then(resp => {
          if (resp && resp.status === 200) {
              let str = 'var configId = "' + thirdUserData.channelConfigId + '";var userId = "' +
thirdUserData.thirdUserId + '";var userName = "' + thirdUserData.thirdUserName
                    + '";var tenantSpaceId = "' + thirdUserData.tenantSpaceId + '";var locale = "' +
thirdUserData.locale
                    + '";var token='"+ thirdUserData.thirdUserToken+'";';
            importScript(str+resp.data);
          } else {
            this.$alert("链接失败！");
            console.log(resp.status);
            console.log(resp);
          }
        });
    }

</script>

    <!--  HTML components of simple GUI -->
    <div id="status_line">
    </div>
</body>

</html>
```

1.6 测试与验证

完成页面集成之后，您需要测试与验证WEB聊天控件是否能正常使用。以下以Chrome浏览器为例。在验证过程中，您可以使用Nginx服务器来模拟第三方调用WEB聊天控件，也可以使用其他您熟悉的访问方式进行。

步骤1 本地开启Nginx服务器（Nginx版本可参考**nginx/Windows-1.22.0**），在nginx.conf文件中配置服务地址、证书等信息。

步骤2 输入服务器地址并发送请求，模拟第三方调用WEB聊天控件。

步骤3 使用多媒体技能队列的座席登录云客服，点击“签入”，进入在线交谈工作台，并设置为空闲态，如**图1-14**所示。

图 1-14 签入按钮

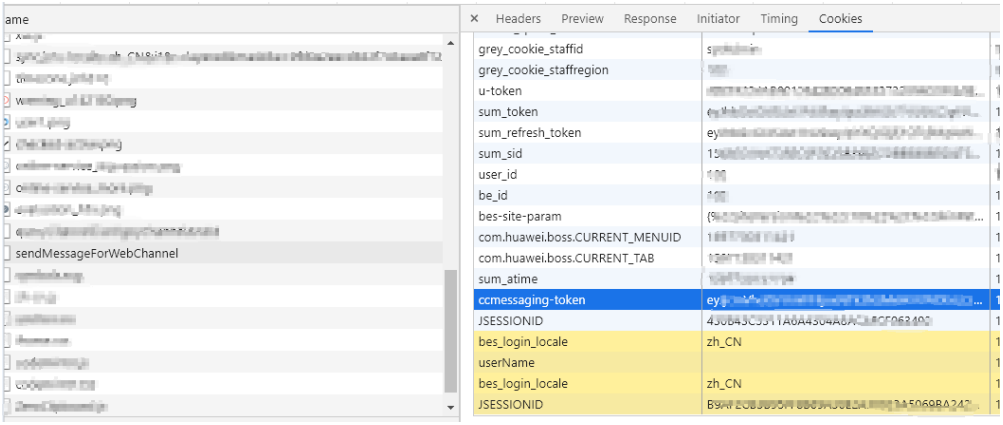
步骤4 在企业客户页面按F12，打开控制台，选择“Network”，然后刷新页面。

单击控制台中出现的thirdPartyClient.js的请求，在右侧单击Response，如果有内容返回，右下角出现蓝色小圆圈图标，点击后出现接入座席提示，如**图1-15**所示，说明调用成功。

图 1-15 客户端对话框

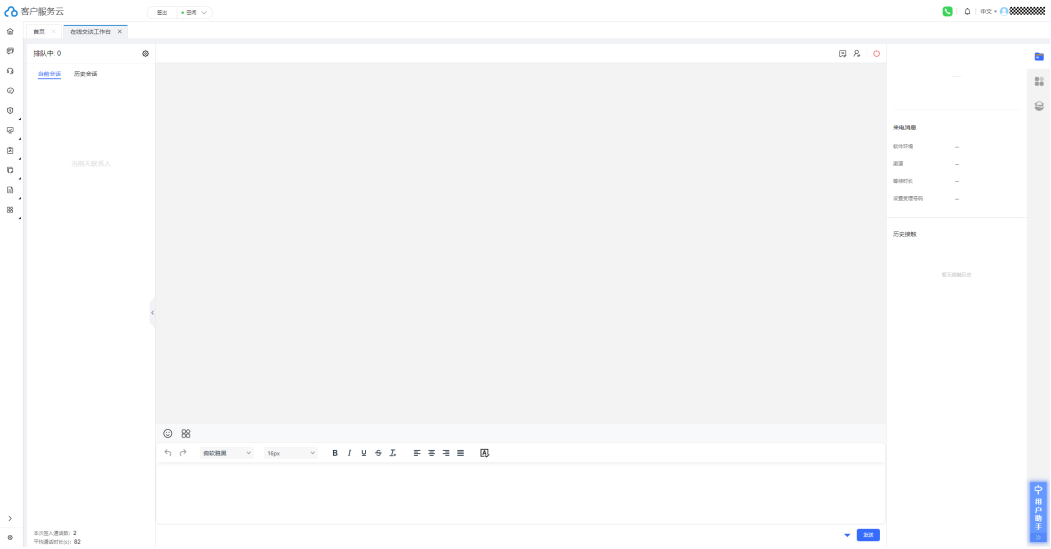
步骤5 在浏览器控制台，选择Application，展开左侧Storage > Cookies > 您的域名，查看cookies中是否写入ccmessaging-token，若出现图1-16所示字段则表示集成成功。

图 1-16 HTML 页面 cookies 中的 ccmessaging-token



步骤6 在图1-17所示工作台查看座席是否处于占用态，是否收到了客户端发送的请求，若可以正常交谈，则表示集成成功。

图 1-17 座席工作台



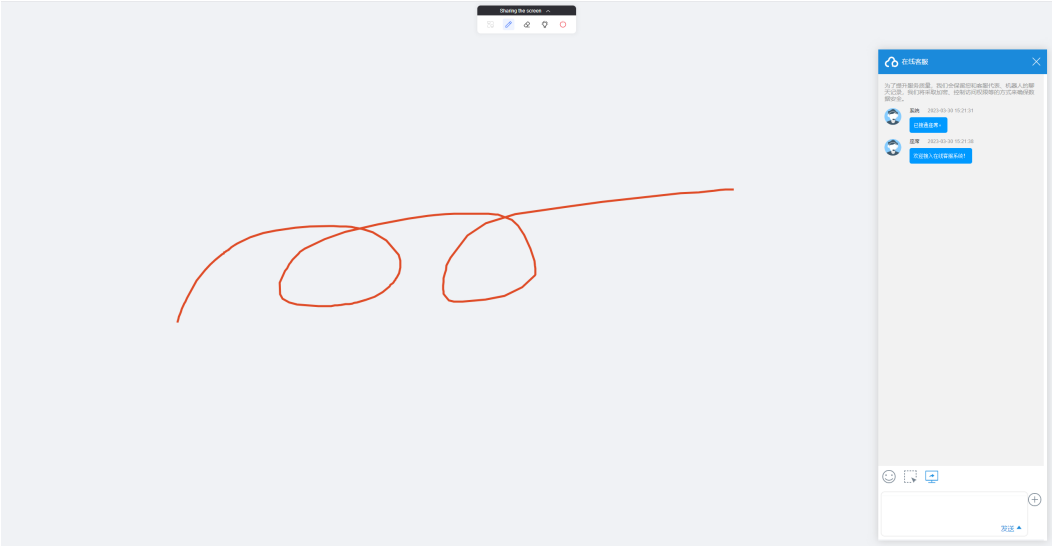
步骤7 若租户开启了网页协同特性，可在客户端对话框发起网页协同，点击网页协同按钮，会弹出确认提示框，用户可确认发起网页协同。

图 1-18 客户对话框

在线客服
为了提升服务质量，我们会保留您和客服代表、机器人的聊天记录，我们将采取加密、控制访问权限等的方式来确保数据安全。
系统 2023-03-30 15:21:31
已接通座席。
座席 2023-03-30 15:21:38
欢迎接入在线客服系统！
提示
当前操作会共享您的浏览器页面，可能会包含您的个人数据，是否继续？
否
是
发送

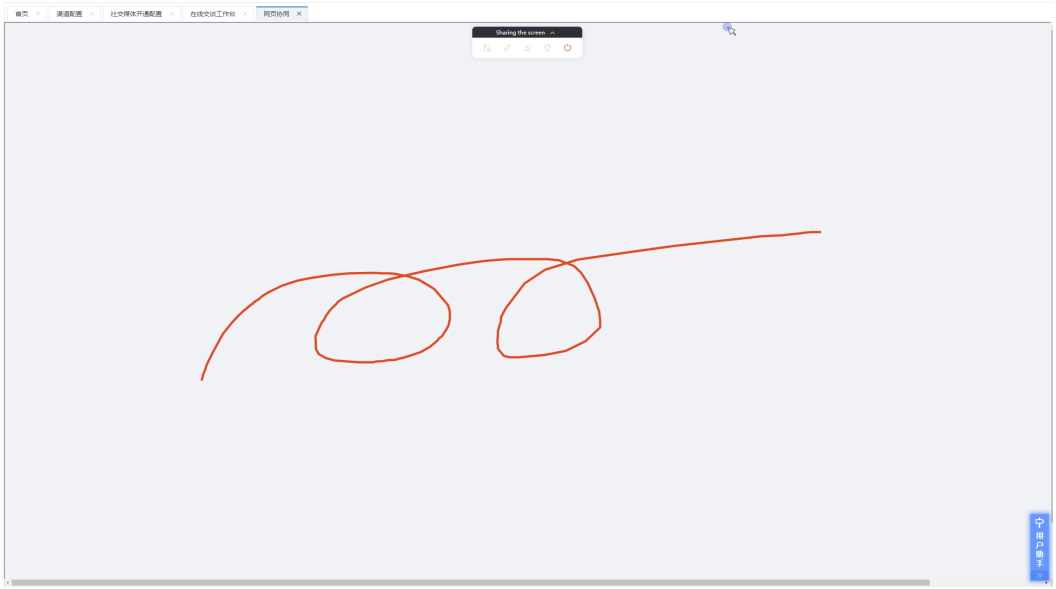
步骤8 座席工作台接受网页协同后，客户端可分享当前页并且可进行画笔、高亮、申请座席远程控制，当看到页面右下角出现如**图6 客户网页协同页面**的网页协同图标时，则表示成功发起网页协同。

图 1-19 客户网页协同页面



步骤9 座席工作台可观看客户端当前的页面和画笔、高亮操作，也可请求远程控制客户端页面。

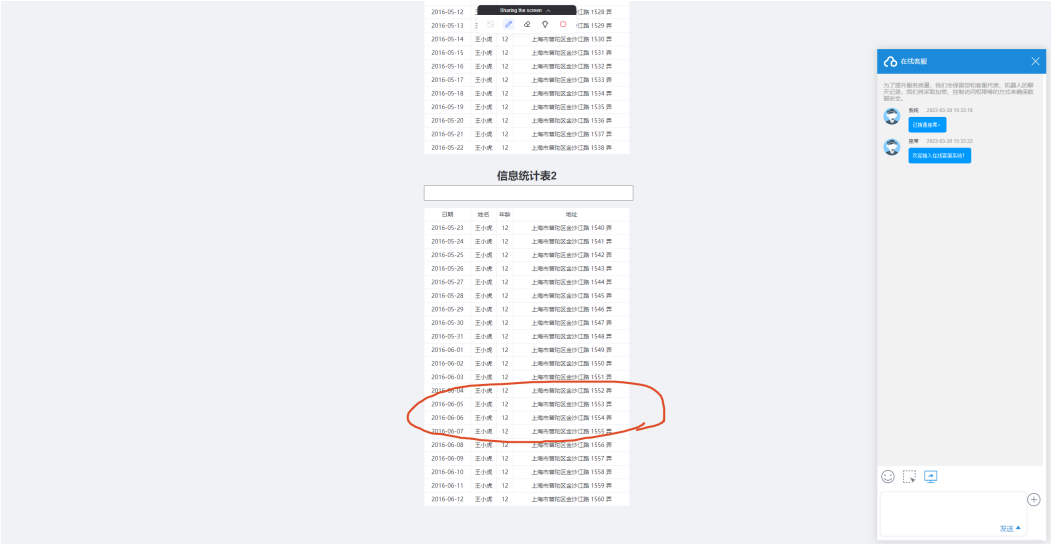
图 1-20 座席工作台网页协同页面



步骤10 滚动网页操作，画笔所画的标记不支持随网页滚轮滑动，如**图8**、**图9**所示，若业务场景需要滚动网页，请先清除当前画笔标记，然后在滑动页面后重新使用画笔功能。

图 1-21 使用画笔功能圈出部分网页内容

图 1-22 将网页随意滑动，画笔所画的圈不会随着网页的滑动而移动



说明

请确保您的租间对应多媒体技能队列下只有一个座席签入，否则根据路由规则，系统有可能会将会话路由到其他座席侧，您可能不会每次都能收到客户请求。

----结束

1.7 常见问题

1.7.1 xmlHttpRequest 请求云客服的 URL，报跨域问题

问题现象

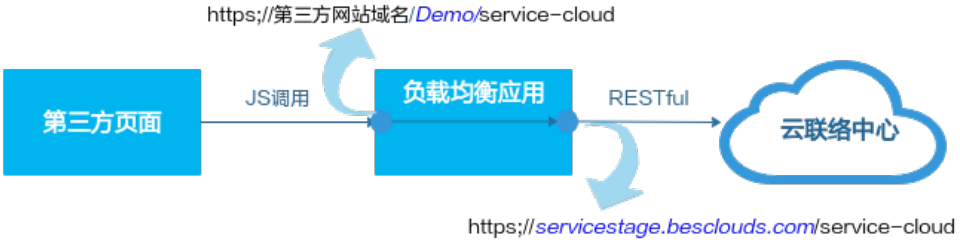
报错提示如下：

```
Access to XMLHttpRequest at "requested js" from origin xx has been blocked by CROS policy：No 'Access-Control-Allow—Origin' header is present on the requested response;
```

解决方法

跨域问题，是因为集成方的网站做了安全限制，不允许请求非本域提供的资源造成的。可通过使用负载均衡应用（如Nginx）的反向代理来解决。

图 1-23 地址在负载均衡应用上的映射原理



即第三方页面使用JS调用时，调用本地域名下的一个服务，该服务放通到云客服的域名地址手上。

由于云客服只有识别到service-cloud，才会识别该请求，因此第三方页面的请求地址必须包含service-cloud。

以Nginx为例，整体配置要点如下：

步骤1 在nginx.conf文件中添加首层服务地址，如下所示。

```
location /demo/ {
    proxy_set_header Host $host;
    set $Real $proxy_add_x_forwarded_for;
    if ($Real ~ (\d+)\.(\d+)\.(\d+)\.(\d+)\,(.*)){
        set $Real $1.$2.$3.$4;
    }
    proxy_set_header X-Real-IP $Real;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_pass https://servicestage.besclouds.com/;
}
```

说明

- *demo*为样例，集成方可根据需要自行定义。
- *servicestage.besclouds.com*请以云客服提供的地址为准。

```
location /service-cloud/ {
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_request_buffering off;
    proxy_pass https://servicestage.besclouds.com/service-cloud;
}
```

步骤2 在1.5.2 开发集成页面中采用如下地址进行JS请求，其中location.protocol请使用**集成方的域名+上一步配置的服务名**（*例如demo*）。

```
const $ContextPath = location.protocol + "/service-cloud"
let serviceUrl = $ContextPath+ "/webclient/chat_client/js/thirdPartyClient.js?"+"&t=" + timestamp;
```

```
let thirdUserData = {};
......
```

----结束

2 集成轻量级 WEB 聊天控件（引入 Authorization 认证方式）

2.1 概述
2.2 集成原理
2.3 集成步骤
2.4 开发前准备
2.5 集成开发
2.6 测试与验证
2.7 常见问题

2.1 概述

轻量级WEB聊天控件是一款能快速集成到网站的在线客服用户端产品，**当前仅支持PC端WEB集成**。选择我们的产品，您可以在云客服 SaaS（Software as a Service）系统场景下，快速构建智能在线客服系统，实现与用户在网页进行在线聊天。

轻量级WEB聊天控件提供的网页与客户在线聊天的功能，具有如下特点：

- 轻量级，容易集成到您的网页，不占用门户和工作台的主要页面。
- 易于操作，您的用户能轻松从页面找到与客服交流的窗口，客服人员发送文字、表情、图片、快捷语等操作均可通过座席端的界面完成。

说明

使用手册前，请您先了解我们提供的两种用户认证方式：

1、Token认证方式：对用户有认证要求，适用于认证接入用户的场景。

2、authorization认证方式：游客模式，不认证接入用户的场景。

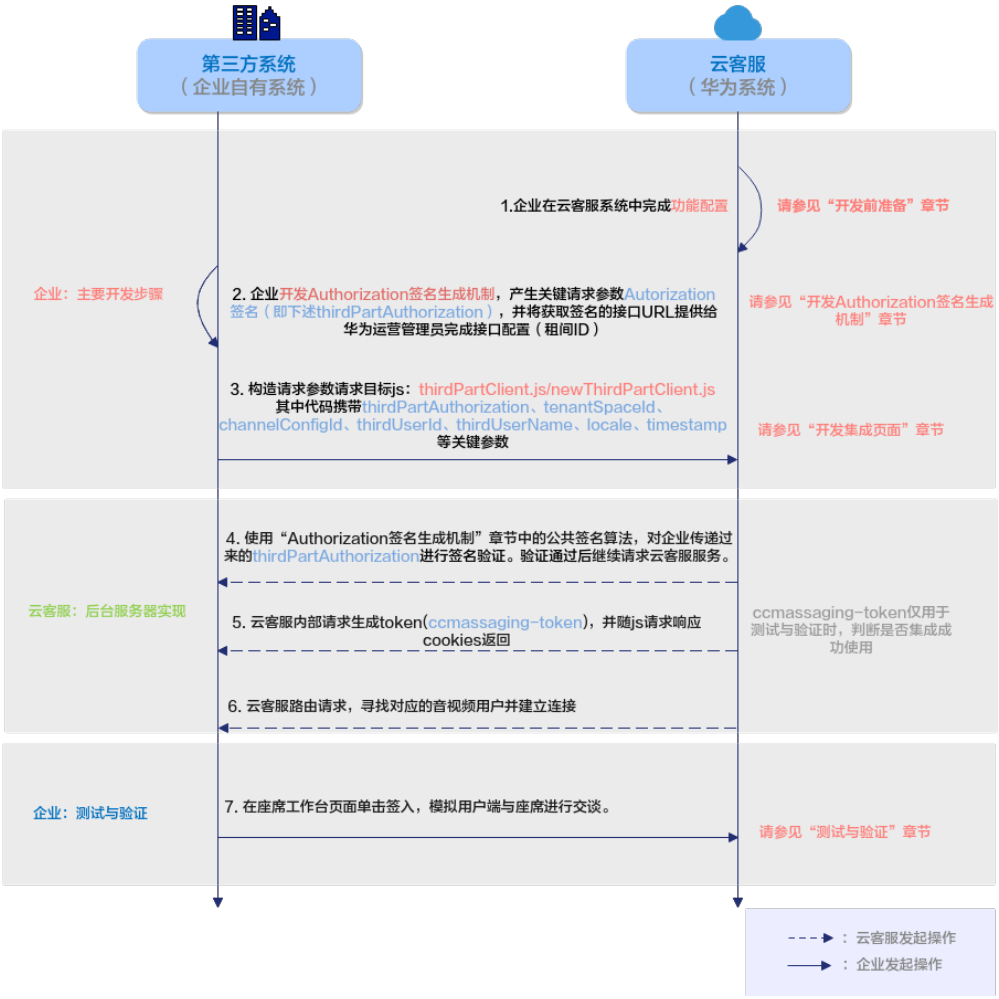
请依据您的需求选择对应的开发指南，如您需要使用Token认证方式，请进入**1 集成轻量级WEB聊天控件（引入Token认证方式）**后继续阅读。

版权所有 © 华为技术有限公司

2.2 集成原理

如果您期望了解我们的实现原理，请参见图2-1。

图 2-1 Web 聊天控件 Authorization 认证方式集成原理

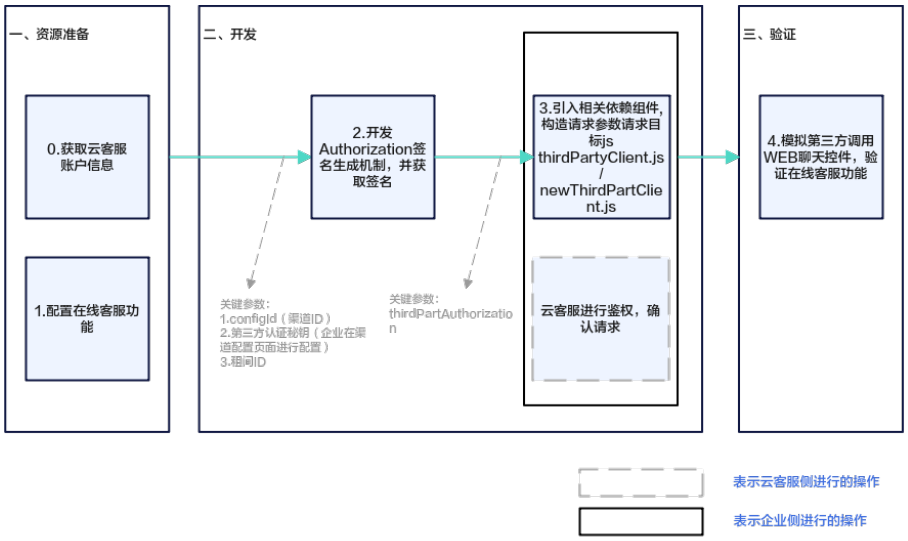


2.3 集成步骤

轻量级WEB聊天控件可以快速高效的集成到您的页面，请您根据图2-2进行集成开发。

版权所有 © 华为技术有限公司

图 2-2 集成开发步骤



步骤1 完成开发前资源准备，获取登录信息，配置在线客服功能，请参见**2.4 开发前准备**。

步骤2 开发Authorization签名生成机制，在您的服务端进行Authorization签名认证，请参见**2.5.1 开发Authorization签名生成机制**。

步骤3 引入相关的依赖框架，构造请求参数请求thirdPartyClient.js和newThirdPartyClient.js ，将Authorization签名、租间ID、企业侧的用户ID、用户名称等信息返回云客服，请求鉴权，鉴权通过后可实现云客服网页聊天能力的轻量级接入，请参见**2.5.2 开发集成页面**。

步骤4 测试与验证WEB聊天控件功能，通过客户端窗口发起在线聊天检查与座席聊天功能，请参见**2.6 测试与验证**。

----结束

2.4 开发前准备

此章节主要描述在集成开发前，需要在云客服中完成的准备工作。

2.4.1 资源准备

在集成前，请您完成以下资源准备：

1. 您已经向云客服申请了租户信息，系统运维管理员已经为您添加了租户信息，并将如下信息提供给您：

表 2-1 华为运营管理员提供参数信息

参数名	说明
租间ID	创建租间（即您的云客服）成功后，系统生成的租间ID。 您也可以登录您的租间后，选择“配置中心 > 系统管理 > 租户基本信息”，查看您的租间ID。

版权所有 © 华为技术有限公司

参数名	说明
租户管理员账号	登录云客服的账号。
租户登录密码	登录云客服的密码。

同时，您需要获取与配置您的configId、accessKey和secretKey等参数：

表 2-2 需在云客服中获取或配置的参数信息

参数名	说明
configId	渠道ID。完成**2.4.2 在云客服上开通在线客服功能**步骤后，在“配置中心 > 接入配置 > 渠道配置”路径下查看。
accessKey	开发者ID，推荐您使用configId（渠道ID）作为唯一标识。
secretKey	认证签名的密钥，用户可以在“配置中心 > 接入配置 > 渠道配置 > 新增（或修改） > 第三方认证密钥 ”进行配置密钥（规则：长度为43，只能包含A-Z\a-z\0-9）。

说明

- 出于安全性考虑，企业的密钥应该在服务端获取，不能通过前台页面携带该参数传递。
- secretKey为认证签名的唯一密钥，请务必妥善保存。

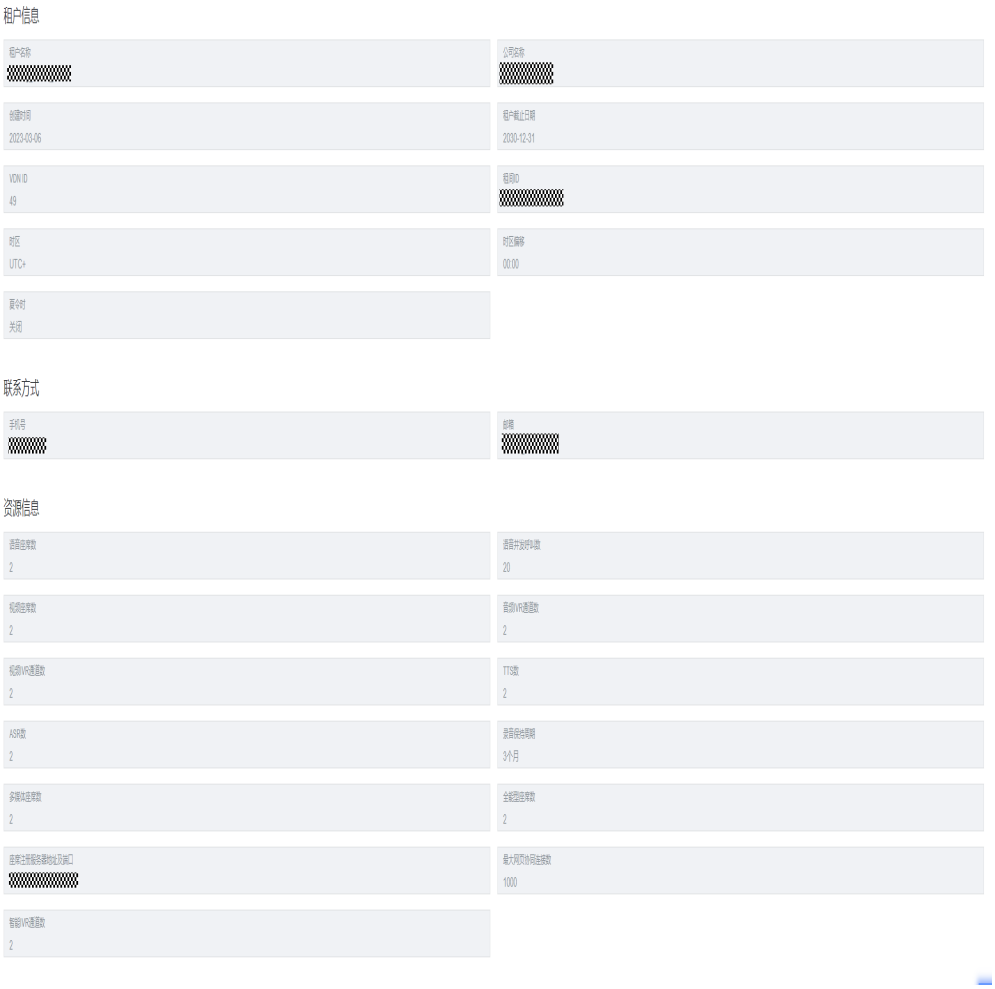
2. 确认您的租间已经具有多媒体座席特性，检查方法如下：

   登录租间，选择“配置中心 > 系统管理 > 租户基本信息”。

   检查多媒体座席数，如果数量为0，表示没有多媒体座席特性，请联系华为运营管理员开通。

版权所有 © 华为技术有限公司

图 2-3 多媒体座席特性检查



3. （可选）如果您希望配置网页协同功能（在与坐席进行文字在线交谈时，希望能与坐席网页画面的共享、标记等操作），请联系华为运营管理员为您开通网页协同功能。

2.4.2 在云客服上开通在线客服功能

步骤1 以租户管理员角色登录云客服。

步骤2 添加多媒体类型的技能队列。

登录后，选择“配置中心 > 员工中心 > 技能队列”，多媒体技能队列配置页面请参见图2-4。如需配置点击通话功能，请同时完成点击通话技能队列的配置，参见图2-5。

步骤3 单击“创建”，完成参数配置。具体参数描述请参见表2-3。

版权所有 © 华为技术有限公司

图 2-4 配置多媒体技能队列界面

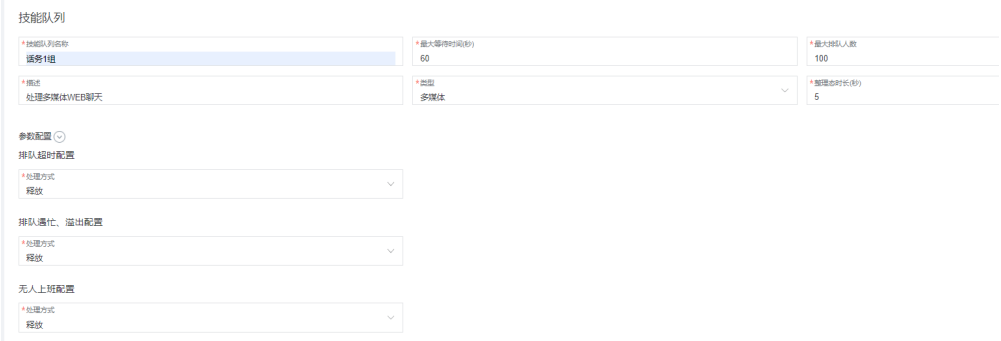


图 2-5 配置点击通话技能队列界面（可选）

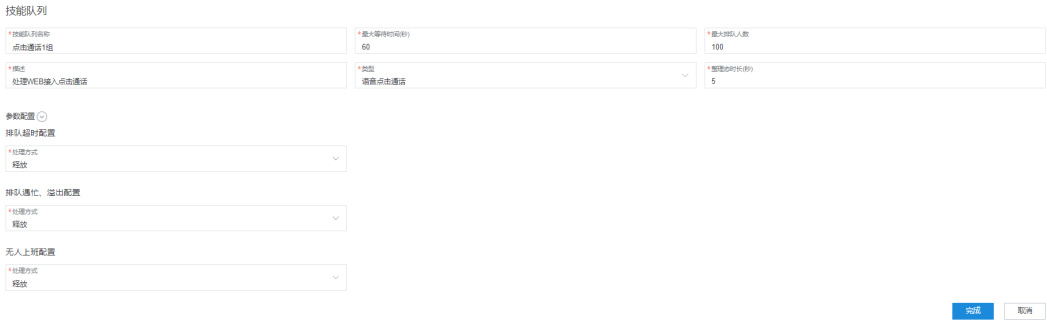


表 2-3 技能队列配置的参数属性

参数名	是否必填	说明
技能队列名称	True	字符串长度<=20，不能使用空格。
最大等待时间（秒）	True	默认：60 单位：秒 输入范围：1-60000
最大排队人数	True	默认：100 输入范围：1-10000
描述	True	字符串长度<=50。

版权所有 © 华为技术有限公司

参数名	是否必填	说明
类型	True	取值范围： ● 语音：专门处理语音类业务 ● 多媒体：专门处理多媒体业务 ● 视频：专门处理视频业务 ● 语音点击通话：协同多媒体业务使用，提供客户与座席进行文字交谈时，直接与座席进行音频通话的能力。 ● 视频点击通话：协同多媒体业务使用，提供客户与座席进行文字交谈时，直接与座席进行视频通话的能力。 **说明** 点击通话类型的技能队列目前仅用于WEB渠道。 默认：语音
整理态时长（秒）	True	当通话结束后，座席进入整理态的时长，默认为5秒。 超过该值，座席进入空闲状态，可以接听客户来电。 输入范围：0-3600

版权所有 © 华为技术有限公司

参数名	是否必填	说明
参数配置	False	个性化配置，用于设置当客户主动呼入该技能队列时遇到无法正常接通的情况下的处理策略。取值范围： ● 排队超时配置 – 处理方式：客户呼入后无空闲座席接听导致排队超时后的处理策略。 ▪ 释放（默认） ▪ 转接 – 设备类型：处理策略为转接时，需要配置转接的技能队列或者IVR流程。 ▪ 技能队列 ▪ IVR ● 排队遇忙、溢出配置 – 处理方式：客户呼入后无空闲座席接听导致需要排队等待，或者排队人数超出限制后的处理策略。 ▪ 释放（默认） ▪ 转接 – 设备类型：处理策略为转接时，需要配置转接的技能队列或者IVR流程。 ▪ 技能队列 ▪ IVR ● 无人上班配置 – 处理方式：客户在无座席值班时间内呼入导致无人接听的处理策略。 ▪ 释放（默认） ▪ 转接 – 设备类型：处理策略为转接时，需要配置转接的技能队列或者IVR流程。 ▪ 技能队列 ▪ IVR ● 排队等待配置：客户呼入后需要排队等待时，可以播放语音，优化客户等待过程。 – 排队等待方式 ▪ 默认等待音

版权所有 © 华为技术有限公司

参数名	是否必填	说明
		▪ 自定义等待音 ▪ IVR ● 保持等待配置：在交谈过程中需要保持通话，让客户等待时，可以播放语音，优化客户等待过程。 – 保持等待方式 ▪ 默认等待音 ▪ 自定义等待音 ● 业务代表应答方式配置：座席接听客户来电后，支持配置播报工号通知客户所属座席。 – 业务代表应答方式 ▪ 报工号 ▪ 不报语音 **说明** 语音、视频、点击通话类型技能队列支持转接IVR和技能队列，多媒体类型的技能队列仅支持转接技能队列。 转接的技能队列类型要与创建的技能队列类型一致。 等待音配置目前仅支持语音、视频类型的技能队列，点击通话类型的技能队列该配置不展示。

1. 单击“完成”完成配置。

步骤4 添加多媒体类型的被叫配置。

1. 选择“配置中心 > 接入配置 > 被叫配置”，进入配置页面。
2. 单击“新增”，为虚拟呼叫中心添加图2-6参数信息，具体参数说明请参考表2-4，并单击“完成”完成配置。如需开启点击通话功能，请在此处同时配置点击通话类型的被叫配置。

图 2-6 新增界面

版权所有 © 华为技术有限公司

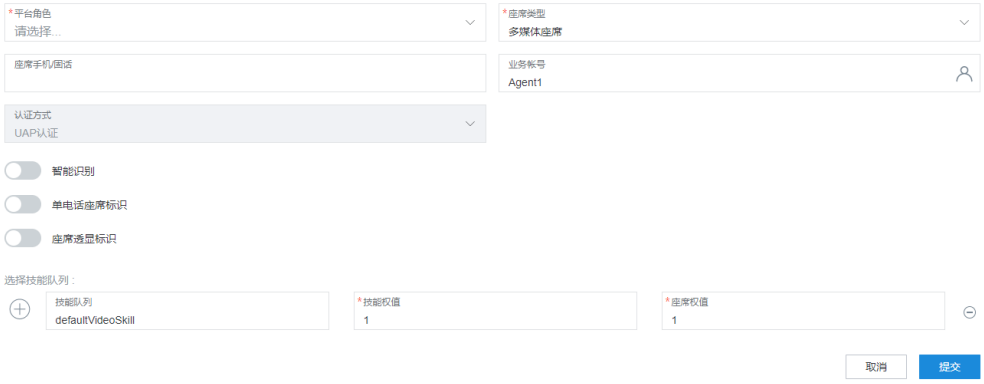
表 2-4 被叫配置的参数属性

参数名	是否必填	说明
接入码	True	面向客户提供的客服热线，客户通过拨打接入码接通到座席侧。 单击"⊕"选择接入码时，弹出框中展示接入码的类型，选择“多媒体”类型的接入码。
扩展码	False	如果希望一个接入码能设置多个目的设备，可通过配置扩展码的方式来实现。 例如，接入码为12345，我们可以添加扩展码1，路由到技能队列A，扩展码2，路由到技能队列B。这样客户可直接拨打123451实现直接接入技能队列A。
设备类型	True	请选择“技能队列”，完成技能队列的被叫配置。
技能队列	True	关联**步骤3**中已创建的技能队列。 技能队列：单击"⊕"选择技能队列时，弹出框中供选择的技能队列的类型与“接入码”的类型一致，例如“接入码”为“多媒体”类型，那么可供选择的技能队列全部为“多媒体”类型。

步骤5 配置业务账号以及技能队列。

1. 选择“配置中心 > 员工中心 > 座席管理”，进入管理座席页签。
2. 选择一条座席工号信息，操作栏单击“配置”，进入座席信息配置页面。
3. 关联对应座席的业务账号以及技能队列。如需开启点击通话功能，请在此处同时关联点击通话的技能队列。

图 2-7 座席信息配置



版权所有 © 华为技术有限公司

表 2-5 座席信息配置的参数属性

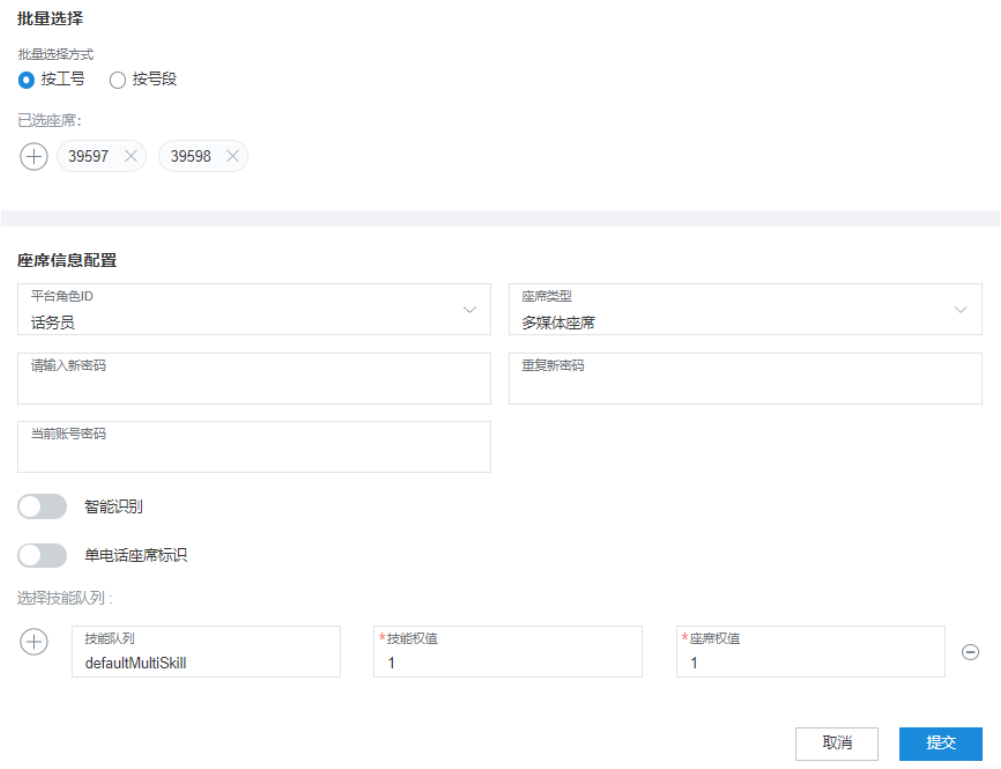
参数名	是否必填	说明
平台角色	True	必填，选择该座席所属角色。 – 话务员：对客户的来话进行接听和转接。 – 质检员：对普通话务员与客户的通话进行干预，例如插入、拦截、强制示忙等，用于对话务员处理来话的表现做辅导和监督。 – 外呼座席：对客户的来话进行接听、转接，不接来话。
座席类型	True	必填，选择该座席类型，可以处理的业务类型。 – 音频座席 – 视频座席 – 多媒体座席 – 全能型座席
座席手机/固话	False	配置座席使用的手机或者固定电话。
业务帐号	True	配置已创建的人员账号。 具体可参考管理人员章节。
智能识别	False	默认关闭，配置座席是否为智能座席。智能座席除了基本的语音控制功能外，支持实时的语音转文本，以及相关的智能推荐功能。当您开启智能识别开关时，需要保证开启智能座席的数量不能超过租户开通时分配的智能座席数。
单电话座席标识	False	开启这个特性后，座席拨指定接入码就可以进一个IVR流程， 根据IVR提示按某个键就能输入工号、密码签入，然后就可以用手机接电话。当您开启开关时，需要系统运维人员给租户基于平台定制单电话座席流程，并由租户提供号码资源，供单电话座席流程接入，才可正常使用该功能。
座席透显标识	False	用于第三方标记座席是否具有透显特性，不是功能特性开关，仅作为一个标签。透显特性是指座席可以自定义用户侧主叫显示号码（用户看到的主叫号码）与座席侧主叫显示号码（客户经理看到的主叫号码）。

版权所有 © 华为技术有限公司

参数名	是否必填	说明
选择技能队列	True	配置座席的技能队列。如果需要添加多个技能队列，那么需要保证添加的所有技能队列的媒体类型一致，全能型座席除外，例如都为“音频”类型或者都为“多媒体”类型。 **说明** – 如果座席类型选择“视频座席”，您在申请租户资源时，需要申请对应的视频座席数。 – 如果座席类型选择“多媒体座席”，您在申请租户资源时，需要申请对应的多媒体座席数。 – 如果座席类型选择“全能型座席”，您在申请租户资源时，需要申请对应的全能型座席数。 – 如果业务账号不够用，请到“配置中心 > 员工中心 > 员工”下添加，此处不赘述。

4. 单击提交，为座席工号分配业务账号和技能队列完成。
5. （可选）单击“批量配置”，进入座席信息配置页面，进行批量配置。

图 2-8 批量配置



– 批量选择：可选择按工号或者按号段的方式选择需要配置的座席。
– 座席信息配置：可参考**步骤5**进行配置。

步骤6 配置WEB渠道。

1. 选择“配置中心 > 接入配置 > 渠道配置”。

2. 单击“新增”，输入渠道接入编码，选择“WEB”渠道，然后单击“下一步”，进入WEB渠道的配置页面。

说明

渠道接入编码必须是唯一的，输入值只能由字母，数字或下划线组成（首字符只能用字母和下划线）。

3. 根据您的需求填写WEB渠道信息，请参见图2-9，具体参数说明请参见表2-6。如需开启点击通话功能，请在该页面配置“点击通话被叫配置”与“点击通话技能队列”。

图 2-9 WEB 渠道配置页面

表 2-6 WEB 渠道配置参数

参数	说明
公共配置	
信息配置	需要配置以下参数： – 技能队列：配置框会查询当前租间的所有多媒体被叫配置，用户酌情选择。 – 转人工关键字：用户在H5客户端输入后该关键字后会由机器人服务转为人工座席服务。 **说明** 若不开启智能机器人，转人工关键字可不设置。 – 点击通话被叫配置：配置框会查询当前租间的所有音视频座席和IVR。 – 击通话技能队列：“点击通话被叫配置”中选择音视频座席后，需要配置相关技能队列。 – 点击获取IVR接入码：“点击通话被叫配置”中选择IVR后，需要配置IVR接入码。

版权所有 © 华为技术有限公司

参数	说明
会话结束方式	默认关闭，若不开启，默认20分钟超时，打开后需要配置以下参数： – 客户未回复提醒时间间隔：用户在客户端无会话达到该时间后会断开连接。 – 会话结束语：会话断开后结束提醒。
第三方认证密钥	默认关闭。开关开启，需要输入secretKey。当采用Authorization认证方式时，需要配置此参数。
座席工作时间	– 工作日：可配置0-24小时间最多4个工作时间段。默认展示一个时间段，单击新增，可增加时间段配置。 – 非工作日：可配置0-24小时间最多4个工作时间段。默认展示一个时间段，单击新增，可增加时间段配置。 – 非工作时间提醒信息：不在非工作日配置的时间段内，客户呼入时，提醒信息。
排队提醒	需要配置以下参数： – 可配置排队提醒间隔（秒）：必填，默认为10秒。 – 排队提醒内容：自定义客户呼入需要排队时的提醒内容。
机器人设置	默认关闭。开关开启，则默认先接入机器人进行对话。需配置以下参数： – 更换头像：可更换机器人头像。 – 名字：为机器人添加名字。 – 性别：为机器人添加性别。 – 机器人接入码：从智能IVR中选择配置好的机器人接入码。 – 默认回复语：当机器人不能识别客户语意，自定义回复语。 – 超时回复语：当客户会话超时，自定义超时回复语。 – 转人工提示语：自定义转人工成功提示语 **说明** 如果您想开启机器人设置，实现机器人对话功能，请参考**配置智能IVR**章节，结合您所需要的机器人类型进行配置。
机器人助手配置	默认关闭。开关开启，则在座席侧开启机器人助手。 – 机器人助手接入码：从智能IVR中选择配置好的机器人接入码。 – 静默座席技能队列：配置框会查询当前租间的所有多媒体被叫配置，用户酌情选择。

4. 单击保存并进行下一步，进入到集成说明页面。

步骤7 （可选）在“集成说明”页签，单击“试一试”，可以在弹出页面中设置客户信息，模拟客户端的会话窗口。

您可以测试客户通过当前渠道与座席或者机器人交谈。

版权所有 © 华为技术有限公司

1. 单击“试一试”，在弹出页面的右下角单击“ ”，弹出“在线客服”弹出框。在线客服有两种交谈模式：
   a. 当“接入智能机器人”开关开启后，默认接入机器人，客户与机器人交谈；当客户输入的交谈内容存在机器人可识别的关键词，机器人会识别关键词并进行答复。
   b. 当“接入智能机器人”开关未开启时，对话框自动接入座席，用户可以与在线座席进行交谈。此时需要在云客服页面点击“签入”，签入多媒体座席，然后选择“在线交谈工作台”，弹出当前会话的工作台，完成接入后可进行在线交谈。

   说明

   当您与机器人交谈时，您可以通过单击“转人工”按钮由机器人转到客服座席，也可以通过输入包含“转人工关键词”的内容由机器人转到客服座席（“转人工关键词”可在**步骤6**中的页面配置）。但是，当您与客服座席交谈时，不能转人工或者转机器人。
2. 在弹出的“在线客服”对话框中，客户输入与机器人或客服座席的交谈内容，单击“发送”，测试机器人或客服座席的答复。
3. （可选）在对话框附近单击“ ”，然后单击“评价”，你可以对当前客服的服务进行评价，如**图2-10**所示，包含星级满意度分数评价和内容评价，单击“确认”。

图 2-10 评价界面

版权所有 © 华为技术有限公司

说明

用户可以对进行过会话互动的客服座席进行满意度评价。客户在会话过程中和会话结束后，均可以对客服座席的服务主动发起满意度评价，随时可以评价，以最后一次提交结果为准。

----结束

渠道配置完成后，请记录该渠道的配置ID，在开发认证验证接口以及页面集成时会用到该值。

配置ID	渠道接入编码	渠道类型	绑定技能队列	操作
869525871671777281	WECHAT001	WECHAT	defaultMediaSkill (10001+)	修改 删除
853180664420335631	wechat001	WECHAT	defaultMediaSkill (10001+)	修改 删除
827635713993441281	hyy	WECHAT	defaultMediaSkill (10001+)	修改 删除
817854979183314945	WEBCHAT	WEB	defaultMediaSkill (10001+)	修改 删除
743032024951966724	A20191231	WECHAT	defaultMediaSkill (10001+)	修改 删除
740930049837911041	a123456	WECHAT	defaultMediaSkill (10001+)	修改 删除

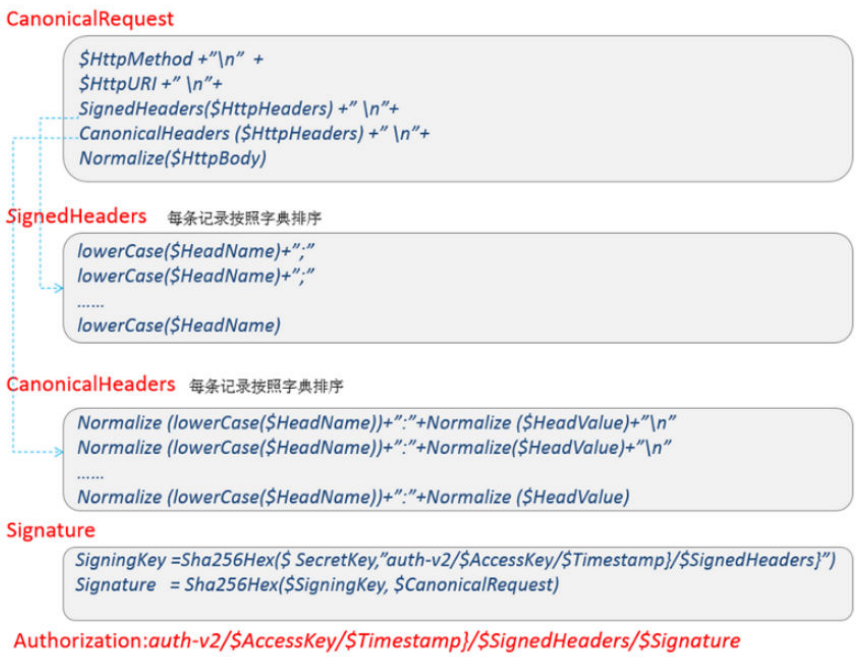
2.5 集成开发

2.5.1 开发 Authorization 签名生成机制

您需要开发一个生成Authorization签名的认证机制，云客服需要用您生成的authorization签名来进行鉴权。

在开发前，请您先了解生成Authorization签名的内部规则，如图2-11所示。

图 2-11 Authorization 签名生成机制的主要规则



版权所有 © 华为技术有限公司

了解完以上原理与规则后，请您依据以下步骤进行Authorization签名认证机制的开发：

步骤1 生成SignedHeaders：遍历参与编码的HttpHead里面的Header name。

1. 将Header name都改为小写，即调用lowerCase()函数，SignerUtils工具类请参考1，参考如下：

```
private Map<String, String> lowerCaseSignedHeaders(Map<String, String> signedHeaders) {
    if ((null == signedHeaders) || signedHeaders.isEmpty()) {
        throw new IllegalArgumentException("signedHeaders cann't be null.");
    }
    Map<String, String> headers = new HashMap<>(SignerUtils.HASH_MAP_INITIALIZATION_SIZE);
    for (Map.Entry<String, String> e : signedHeaders.entrySet()) {
        String name = e.getKey();
        String value = e.getValue();
        headers.put(name.toLowerCase(Locale.ENGLISH), value.trim());
    }
    return headers;
}
```

2. 将步骤1-1中Header name转换后的字符后面追加分割符";"，生成一条记录,注意最后一个字段不追加";"。
3. 将步骤1-1中所有记录按照字典排序，然后按照顺序连接成一个大字符串，参考如下。

```
private String appendSignedHeaders(StringBuilder buffer) {
    int start = buffer.length();
     Set<String> headerNames = new TreeSet<>(this.signedHeaders.keySet());
    for (String name : headerNames) {
        buffer.append(name).append(';');
    }
    buffer.deleteCharAt(buffer.length() - 1);
     int end = buffer.length();
    String signedHeadersStr = buffer.substring(start, end);
    return signedHeadersStr;
}
```

说明

请对以下Header进行编码：

Content-Length="***"

Content-Type="application/json;charset=UTF-8"

步骤2 生成authStringPrefix。

将以下字段用“/”进行拼接authVersion、accessKey、timestamp、SignedHeaders。格式如下：

```
authStringPrefix="auth-v2/{accessKey}/{timestamp}/{SignedHeaders}";
```

说明

- auth-v2：鉴权版本号，当期版本为固定字符串“auth-v2”。
- accessKey：第三方系统使用configId（渠道ID）作为唯一标识。
- timestamp：取第三方发起服务时的时间，timestamp为String类型。时间字符串格式化为"yyyy-MM-dd'T'HH:mm:ss.SSS'Z"。
- SignedHeaders：参与编码的HttpHead里面的Header name，由步骤1生成。

步骤3 生成signingKey。

将步骤2生成的authStringPrefix按照sha256Hex进行加密，其中密钥SecretKey为第三方系统在渠道配置页面配置的密钥。sha256Hex算法参考如下，SignerUtils工具类请参考1：

版权所有 © 华为技术有限公司

```
public static String sha256Hex(String key, String toSigned) throws
NoSuchAlgorithmException,InvalidKeyException, UnsupportedEncodingException {
    Mac mac = Mac.getInstance("HmacSHA256");
    mac.init(new SecretKeySpec(key.getBytes(SignerUtils.CHARSET), "HmacSHA256"));
    String digit = new String(SignerUtils.encodeHex(mac.doFinal(toSigned.getBytes(SignerUtils.CHARSET))));
    return digit;
}
```

步骤4 生成CanonicalHeaders。

说明

要求计算的编码规则与SignedHeaders一致，但增加了head value的编码。

1. 遍历参与编码的HttpHead里面的head name，将Header name都改为小写，即调用lowerCase()函数（请参考**步骤1**）。
2. 调用normalize函数，对刚才转换后的小写字符串进行格式化，PathUtils工具类请参考**2**。

```
/**
 * normalize
 * @param value payload信息
 * @return builder
 */
public static String normalize(String value) {
    try {
        StringBuilder builder = new StringBuilder(PathUtils.DEFAULT_CAPACIT);
        for (byte b : value.getBytes(PathUtils.CHARSET)) {
            if (PathUtils.URI_UNRESERVED_CHARACTERS.get(b & 0xFF)) {
                builder.append((char) b);
            } else {
                builder.append(PathUtils.PERCENT_ENCODED_STRINGS[b & 0xFF]);
            }
        }
        return builder.toString();
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException(e);
    }
}
```

3. 将**步骤4-1**中的记录按照字典排序，进行排序。
4. 遍历排序后的记录，中间追加字符串"\n"连接成一个大的字符串，最后一条记录不追加"\n"。

步骤5 生成canonicalRequest。

将字段“HttpMethod”、“HttpURI”、“SignedHeaders”、“CanonicalHeaders”、“NormalizePath”用“\n”拼接，最后一条记录不追加"\n"，PathUtils工具类请参考**2**。。

```
private String canonicalRequest() {
    StringBuilder buffer = new StringBuilder(PathUtils.DEFAULT_CAPACITY);
    buffer.append(this.httpMethod).append(System.lineSeparator());
    buffer.append(this.uri).append(System.lineSeparator());
      this.appendSignedHeaders(buffer);
    buffer.append(System.lineSeparator());
      this.appendCanonicalHeaders(buffer);
    buffer.append(System.lineSeparator());
     if (this.isNotEmpty(this.payload))
     {
       buffer.append(PathUtils.normalize(this.payload));
     }
     return buffer.toString();
 }
```

格式参考如下：

版权所有 © 华为技术有限公司

```
CanonicalRequest = $HttpMethod + "\n" + $HttpURI+ "\n" + SignedHeaders($HttpHeaders) + "\n" +
CanonicalHeaders ($HttpHeaders) + "\n" + NormalizePath($HttpBody)
```

说明

- CanonicalRequest参数说明：

  $HttpMethod：指HTTPS协议中定义的GET、PUT、POST等请求，必须使用全大写的形式。

  $HttpURI：指接口请求的URI，必须以“/”开头，不以“/”开头的需要补充上，空路径为“/”，样例：/service-cloud/webclient/chat_client/js/newThirdPartyClient.js。

  SignedHeaders：**步骤1**生成的SignedHeaders。

  CanonicalHeaders：**步骤4**生成的CanonicalHeaders。

  NormalizePath：格式化处理后的Body体。

- 仅对NormalizePath中的如下参数进行编码：

  thirdUserName：企业的用户名。

  thirdUserId：企业用户ID。

  tenantSpaceId：企业系统提供的租间ID。

  channelConfigId：企业接入的渠道ID。

步骤6 生成signature。将**步骤5**生成的canonicalRequest再按照sha256Hex进行加密，此处加密的密钥key为**步骤3**生成的signingKey。

步骤7 生成签名authorization：将**步骤2**生成的authStringPrefix和**步骤6**生成的signature用“/”进行拼接。格式参考如下：

```
Authorization:$authStringPrefix/$Signature
```

----结束

相关参考

生成Authorization签名的认证机制过程中，涉及到SignerUtils与PathUtils工具类，格式参考如下：

1. SignerUtils

```
import java.nio.charset.StandardCharsets;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;

public class SignerUtils {
    private static final int HASH_MAP_INITIALIZATION_SIZE = 5;
    private static final int ONE_CHAR_BITS_NUM = 4;
    private static final String CHARSET = "UTF-8";
    private static final char[] DIGITS_LOWERS = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a',
        'b', 'c', 'd', 'e', 'f'};
    private SignerUtils() {
    }

    private static char[] encodeHex(final byte[] data) {
        final int le = data.length;
        final char[] outs = new char[le << 1];
        for (int i = 0, j = 0; i < le; i++) {
            outs[j++] = SignerUtils.DIGITS_LOWERS[(0xF0 & data[i]) >>> ONE_CHAR_BITS_NUM];
            outs[j++] = SignerUtils.DIGITS_LOWERS[0x0F & data[i]];
        }
        return outs;
    }
}
```

2. PathUtils

```
import java.io.UnsupportedEncodingException;
import java.util.BitSet;
```


版权所有 © 华为技术有限公司

```
import java.util.Locale;
import java.util.concurrent.CompletionException;

public class PathUtils {
    private static final String CHARSET = "UTF-8";
    private static final int NUM_256 = 256;
    private static final int DEFAULT_CAPACITY = 16;
    private static final BitSet URI_UNRESERVED_CHARACTERS = new BitSet();
    private static final String[] PERCENT_ENCODED_STRINGS = new String[NUM_256];
    static {
        for (int i = 97; i <= 122; i++) {
            PathUtils.URI_UNRESERVED_CHARACTERS.set(i);
        }
        for (int i = 65; i <= 90; i++) {
            PathUtils.URI_UNRESERVED_CHARACTERS.set(i);
        }
        for (int i = 48; i <= 57; i++) {
            PathUtils.URI_UNRESERVED_CHARACTERS.set(i);
        }
        PathUtils.URI_UNRESERVED_CHARACTERS.set(45);
        PathUtils.URI_UNRESERVED_CHARACTERS.set(46);
        PathUtils.URI_UNRESERVED_CHARACTERS.set(95);
        PathUtils.URI_UNRESERVED_CHARACTERS.set(126);

        for (int i = 0; i < PathUtils.PERCENT_ENCODED_STRINGS.length; i++) {
            PathUtils.PERCENT_ENCODED_STRINGS[i] = String.format(Locale.ROOT, "%%%02X", new
Object[]{Integer.valueOf(i)});
        }
    }
    private PathUtils() {}
}
```

2.5.2 开发集成页面

2.5.2.1 核心代码简析

步骤1 引入所需的JS框架，在以下提供的样例中需要引入jQuery，参考版本为jQuery-v3.4.1，如果需要使用网页协同功能，请引入cobrowse.js和cobrowseCommon.js组件或发布插件给用户使用（插件开发参考3.1：cobrowse插件开发、使用指导），cobrowse.js和cobrowseCommon.js采用cdn方式引入，其中ip:port需替换成实际云客服的ip和port，或者直接使用域名，如：https://www.test.com/service-cloud/resource.root/cobrowse/sdk/cobrowse.js，引用代码示例如**图1**所示。

图 2-12 JS 框架引用示例

```
<script type="text/javascript" src="https://ip:port/service-cloud/resource.root/cobrowse/sdk/cobrowse.js"></script>
<script type="text/javascript" src="https://ip:port//service-cloud/webclient/chat_client/js/cobrowseCommon.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
```

说明

1、网页协同功能不支持打开新的页签后在新页签上进行协同操作。

2、引入网页协同的cobrowse.js和cobrowseCommon.js组件时，第三方页面中包含多层iframe嵌套时，仅引入了js组件的iframe内可以进行网页协同，若该iframe存在父iframe框架，则无法在父iframe中进行网页协同操作）。

步骤2 请按照章节**2.5.1 开发Authorization签名生成机制**在服务端生成Authorization签名。

步骤3 在企业自有系统前台通过ajax请求，获取服务端生成的Authorization签名。

```
function testAjax() {
    let timestamp = new Date().getTime();
    let serviceUrl = "authorizationService";
    let thirdUserData = {};
```


版权所有 © 华为技术有限公司

```
    thirdUserData['thirdUserName'] = userName;
    thirdUserData['thirdUserId'] = userId;
    thirdUserData['tenantSpaceId'] = tenantSpaceId;
    thirdUserData['channelConfigId'] = configId;
    thirdUserData['locale'] = locale;
    thirdUserData['mapService'] = tencent;
    thirdUserData['timestamp'] = timestamp;
    var request = $.ajax({
        url: serviceUrl,
        method: "POST",
        contentType: "application/json",
        data: JSON.stringify(thirdUserData),
    });
    request.done(function(message){
        if(message){
            thirdUserData['thirdPartAuthorization'] = message;
            thirdValidate(thirdUserData);
        }
        return  message;
    });
```

步骤4 获得签名后，用ajax请求需要集成的js脚本。

```
const $ContextPath = "https://ip:port/service-cloud";
function thirdValidate(thirdUserData) {
  let timestamp = new Date().getTime();
  let serviceUrl = $ContextPath + "/webclient/chat_client/js/newThirdPartyClient.js?"+"&t=" + timestamp;
      var request = $.ajax({
        url: serviceUrl,
        method: "POST",
        data: JSON.stringify( ),
        crossDomain: true,
        dataType:"text",
        xhrFields: {withCredentials: true},
        error: function (XMLHttpRequest, textStatus, errorThrown)
            { console.log(XMLHttpRequest.status); console.log(XMLHttpRequest.readyState);
console.log(textStatus); },
        success: function (data) {
            importScript(data)
        }
});
```

表 2-7 开发集成页面参数说明

参数名	是否必填	说明
$ContextPath	True	实际域名。请将上述https:// ip:port/换成云客服的公网域名。
thirdPartAuthorization	True	企业生成的Authorization签名，与**2.5.1 - 步骤7**中签名保持一致。
tenantSpaceId	True	租间ID，请在“配置中心 > 系统管理 > 租户基本信息”中查看。
thirdUserId	True	企业用户ID。**2.5.1 - 步骤5**中的同名参数保持一致。
thirdUserName	True	企业用户名。**2.5.1 - 步骤5**中的同名参数保持一致。
channelConfigId	True	渠道配置ID。完成**配置WEB渠道**后，请在“配置中心 > 接入配置 > 渠道配置”中查看。

版权所有 © 华为技术有限公司

参数名	是否必填	说明
locale	True	第三方租间语言信息。 ● zh：中文 ● en：英文
timestamp	True	时间戳。时间字符串格式化为"yyyy-MM-dd'T'HH:mm:ss.SSS'Z""
mapService	False	用户地图服务。 ● tencent ● google 缺省值：tencent **说明** 腾讯地图暂不支持国外位置。

----结束

2.5.2.2 开发样例：JS 页面集成样例

环境要求	-
引用库	jquery.js
下载链接	**index.html**

须知

- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

index.html

```
<!DOCTYPE html>
<html>

<head>
    <!-- 如果需要使用网页协同功能，需引入cobrowse.js，cobrowseCommon.js这两个js，其中ip:port需替换成实际云客服的ip和port，或者直接使用域名 -->
    <script type="text/javascript" src="https://ip:port/service-cloud/resource.root/cobrowse/sdk/cobrowse.js"></script>
    <script type="text/javascript" src="https://ip:port//service-cloud/webclient/chat_client/js/cobrowseCommon.js"></script>
    <!-- 0.引入jquery组件 -->
    <script type="text/javascript" src="jquery.js"></script>
</head>

<body>
```


版权所有 © 华为技术有限公司

```
<script>
// 1.定义变量,后续聊天框页面需要使用，详细参数说明请参见表2-7
    const userId = "XXXXXXXX";                  // 用户id（即为表中thirdUserId）
    const userName = "XXXXXXXXXXXX";                // 用户昵称（即为表中thirdUserName）
    const tenantSpaceId ="XXXXXXXXXX";                // 租间id(tenant id)
    const configId = "XXXXXXXXXXXXXXXXXXX";            // 渠道id(wWEBchannelConfig id)
    const locale = "zh";                        // 语言，locale:[zh,en]
    const $ContextPath = "https://ip:port/service-cloud";        //请求地址

// 2.构造请求参数
    let serviceUrl = $ContextPath + "/webclient/chat_client/js/newThirdPartyClient.js?&t="+new
Date().getTime();
    let thirdUserData = {};
    thirdUserData['thirdUserName'] = userName;
    thirdUserData['thirdUserId'] = userId;
    thirdUserData['tenantSpaceId'] = tenantSpaceId;
    thirdUserData['channelConfigId'] = configId;
    thirdUserData['locale'] = locale;
thirdUserData['timestamp'] = new Date().getTime();                            // 时间戳
thirdUserData['secretKey'] = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX';          // 认证秘钥
    getAuthorization(thirdUserData);                                        // Authorization 签名

    // 3.请求newThirdPartyClient.js
    var request = $.ajax({
        url: serviceUrl,
        method: "POST",
        data: JSON.stringify(thirdUserData),
        xhrFields: { withCredentials: true },
        success: function (data) {
            // 4.创建script标签并运行newThirdPartyClient.js脚本
            importScript(data);
        },
        error: function (XMLHttpRequest, textStatus) {
            alert("unauthorized,validate failed")
        }
    });

// 5.在集成页面加载运行JS
    var importScript = (function (oHead) {
        function loadError(oError) {
            throw new URIError("The script " + oError.target.src + " is not accessible.");
        }
        return function (sSrc, fOnload) {
            var oScript = document.createElement("script");
            oScript.type = "text/javascript";
            oScript.onerror = loadError;
            if (fOnload) { oScript.onload = fOnload; }
            oHead.appendChild(oScript);
            oScript.innerHTML = sSrc;
        }

    })(document.head || document.getElementsByTagName("body")[0]);
// 根据参数获取authration签名
function getAuthorization(thirdUserData){
    $.ajax({
        url:"/webchat/authorizationService",
        method:"post",
        data:JSON.stringify(thirdUserData),
        async:false,
        success: function (data) {
            thirdUserData['thirdPartAuthorization'] = data;
            }
        });
    }
</script>
</body>

</html>
```


版权所有 © 华为技术有限公司

2.6 测试与验证

完成页面集成之后，您需要测试与验证WEB聊天控件是否能正常使用。以下以Chrome浏览器为例。在验证过程中，您可以使用Nginx服务器来模拟第三方调用WEB聊天控件，您也可以使用其他您熟悉的访问方式进行。

步骤1 本地开启Nginx服务器（Nginx版本可参考**nginx/Windows-1.22.0**），在nginx.conf文件中配置服务地址、证书等信息。

步骤2 输入服务器地址并发送请求，模拟第三方调用WEB聊天控件。

步骤3 使用多媒体技能队列的座席登录云客服，签入，并设置为空闲态。

步骤4 在企业客户页面按F12，打开控制台，选择“Network”，然后刷新页面。

单击控制台中出现的thirdPartyClient.js的请求，在右侧单击Response，如果有内容返回，右下角出现蓝色小圆圈图标，点击后出现接入座席提示，说明调用成功。

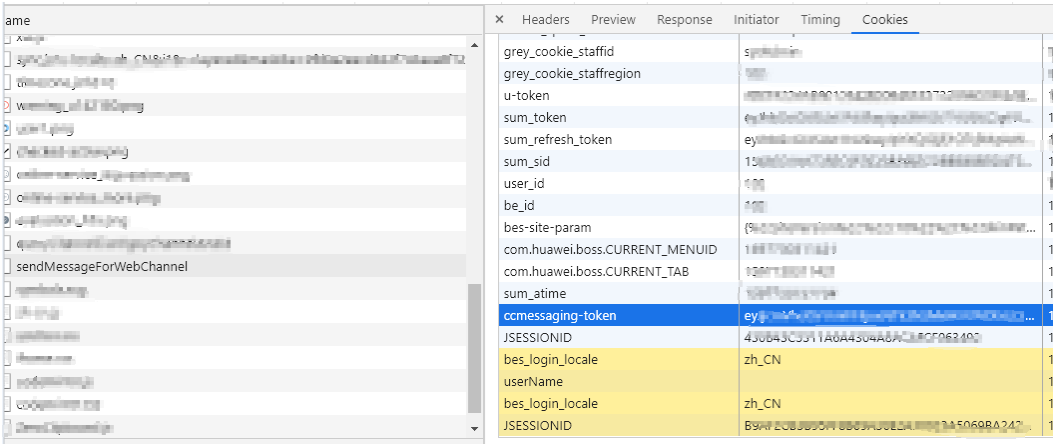
版权所有 © 华为技术有限公司

图 2-13 在线客服

步骤5 在浏览器控制台，选择Application，展开左侧Storage > Cookies > 您的域名，查看cookies中是否写入ccmessaging-token。

版权所有 © 华为技术有限公司

图 2-14 查看 ccmessaging-token



步骤6 云客服的座席页面，查看座席是否处于占用态，是否收到了客户端发送的请求。

步骤7 若租户开启了网页协同特性，可在客户端对话框发起网页协同，点击网页协同按钮，会弹出确认提示框，用户可确认发起网页协同。

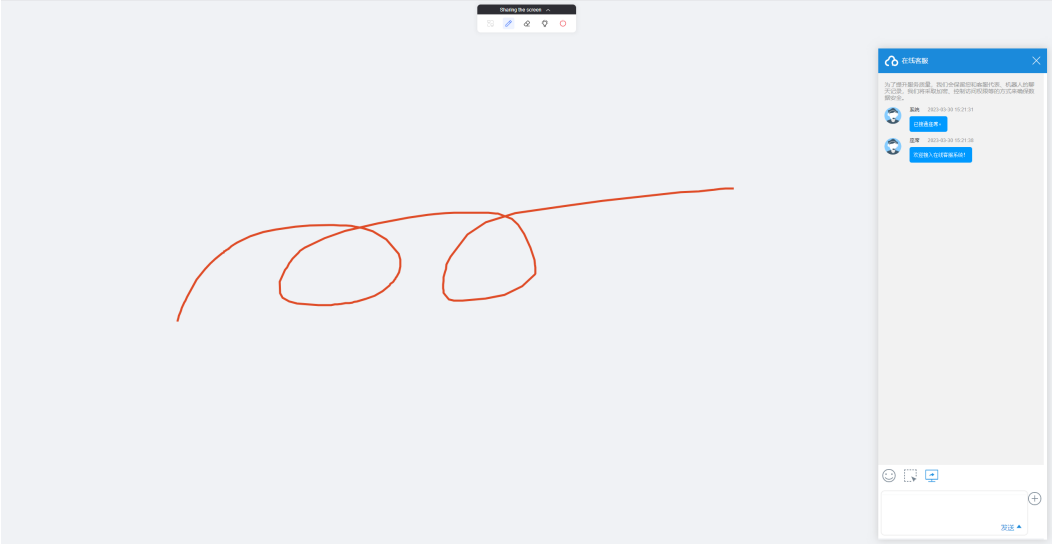
版权所有 © 华为技术有限公司

图 2-15 客户端对话框

版权所有 © 华为技术有限公司

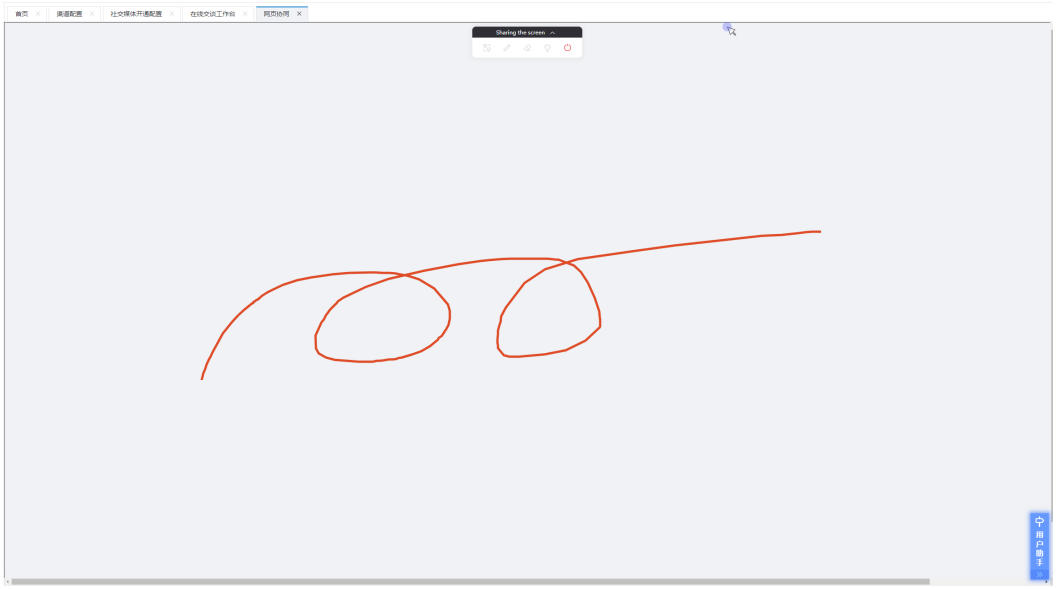
步骤8 座席工作台接受网页协同后，客户端可分享当前页并且可进行画笔、高亮、申请座席远程控制，当看到页面右下角出现如**图4 客户网页协同页面**的网页协同图标时，则表示成功发起网页协同。

图 2-16 客户网页协同页面



步骤9 座席工作台可观看客户端当前的页面和画笔、高亮操作，也可请求远程控制客户端页面。

图 2-17 座席工作台网页协同页面



步骤10 滚动网页操作，画笔所画的标记不支持随网页滚轮滑动，如**图6**、**图7**所示，若业务场景需要滚动网页，请先清除当前画笔标记，然后在滑动页面后重新使用画笔功能。

版权所有 © 华为技术有限公司

图 2-18 使用画笔功能圈出部分网页内容

图 2-19 将网页随意滑动，画笔所画的圈不会随着网页的滑动而移动

步骤11 第三方网页包含动画效果时，网页协同时座席工作台无法展示含动画效果的部分。

说明

请确保您的租间对应多媒体技能队列下只有一个座席签入，否则根据路由规则，系统有可能会将会话路由到其他座席侧，您可能不会每次都能收到客户请求。

----结束

2.7 常见问题

版权所有 © 华为技术有限公司

2.7.1 xmlHttpRequest 请求云客服的 URL，报跨域问题

问题现象

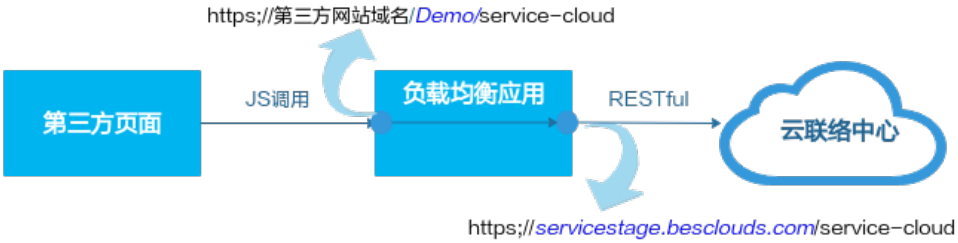
报错提示如下：

```
Access to XMLHttpRequest at "requested js" from origin xx has been blocked by CROS policy：No 'Access-Control-Allow—Origin' header is present on the requested response;
```

解决方法

跨域问题，是因为集成方的网站做了安全限制，不允许请求非本域提供的资源造成的。可通过使用负载均衡应用（如Nginx）的反向代理来解决。

图 2-20 地址在负载均衡应用上的映射原理



即第三方页面使用JS调用时，调用本地域名下的一个服务，该服务放通到云客服的域名地址手上。

由于云客服只有识别到service-cloud，才会识别该请求，因此第三方页面的请求地址必须包含service-cloud。

以Nginx为例，整体配置要点如下：

步骤1 在nginx.conf文件中添加首层服务地址，如下所示。

```
location /demo/ {
    proxy_set_header Host $host;
    set $Real $proxy_add_x_forwarded_for;
    if ($Real ~ (\d+)\.(\d+)\.(\d+)\.(\d+)\,(.*)){
        set $Real $1.$2.$3.$4;
    }
    proxy_set_header X-Real-IP $Real;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_pass https://servicestage.besclouds.com/;
}
```

说明

- *demo*为样例，集成方可根据需要自行定义。
- *servicestage.besclouds.com*请以云客服提供的地址为准。

```
location /service-cloud/ {
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_request_buffering off;
    proxy_pass https://servicestage.besclouds.com/service-cloud;
}
```


版权所有 © 华为技术有限公司

步骤2 在**1.5.2 开发集成页面**中采用如下地址进行JS请求，其中location.protocol请使用**集成方的域名+上一步配置的服务名**（*例如demo*）。

```
const $ContextPath = location.protocol + "/service-cloud"
let serviceUrl = $ContextPath+ "/webclient/chat_client/js/thirdPartyClient.js?"+"&t=" + timestamp;
let thirdUserData = {};
......
```

----结束

版权所有 © 华为技术有限公司