

实时互动-教育版 自定义界面功能





【版权声明】

©2013-2025 腾讯云版权所有

本文档(含所有文字、数据、图片等内容)完整的著作权归腾讯云计算(北京)有限责任公司单独所有,未经腾讯云事先明确书面许 可,任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯,腾讯云将 依法采取措施追究法律责任。

【商标声明】

🔗 腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算(北京)有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标,依法由权利 人所有。未经腾讯云及有关权利人书面许可,任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为,否则将 构成对腾讯云及有关权利人商标权的侵犯,腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况,部分产品、服务的内容可能不时有所调整。 您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定,除非双方另有约定,否则,腾讯云对本文档内 容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务,及相应的技术售后服务,任何问题请联系 4009100100或95716。



文档目录

自定义界面功能

快速开始

基本概念介绍

自定义样式

关键概念文案替换

自定义页面布局

修改下课按钮行为

修改虚拟背景

自定义功能

屏蔽无关业务功能

麦克风及摄像头开关控制(学生侧)

配置业务功能

分角色设置下课倒计时提醒

配置白板功能权限

配置签到功能

事件监听

自定义界面功能 快速开始

最近更新时间: 2025-01-14 11:54:42

() 说明:

阅读本章节前,请确保您已经看完 Web 和 H5 的集成指引。

通过自定义能实现什么?

课堂的 iOS/Android/Electron 用户界面都基于网页实现,自定义能根据您的需要**修改所有端界面并补充业务逻辑。**业务中主要有以 下使用场景:

| | 业务场景 | 业务举例 |
|-------|----------|---|
| 自定义样式 | 关键概念文案替换 | 页面展示的信息更新为行业内专有名称,例如将"课堂"更新为"会议"等。可参考 <mark>操</mark> 作示例 。 |
| | 自定义页面布局 | ・竖屏布局下修改样式、调整界面 icon 样式(如摄像头、麦克风)等。 修改弹窗按钮的行为。可参考 操作示例。 自定义虚拟背景。可参考 操作示例。 |
| 自定义功能 | 屏蔽无关业务功能 | 隐藏头像区域,隐藏在线人数等。可参考 操作示例 。 |
| | 配置业务功能 | 分角色设置下课倒计时提醒。可参考操作示例。 配置白板功能权限。可参考操作示例。 |
| | 查询事件信息 | 展示和查询课堂信息,可参考基本概念介绍。 在特定事件发生时,进行与业务相关的处理。如查询摄像头/麦克风状态上报后台;进入房间后展示弹窗等。事件监听列表罗列了当前支持的所有事件类型。 |

方式1: 自定义 js 调试

为了简化您开发自定义界面的流程,我们准备了常用的自定义 js 示例,您可以到 github 地址 获取。您可以通过以下命令快速启动一 个 demo:



进入课堂后,在上课地址后面添加 &debugjs=http://localhost:3000/custom.js ,如果有 css,再拼接 &debugcss=http://localhost:3000/custom.css,即

&debugjs=http://localhost:3000/custom.js&debugcss=http://localhost:3000/custom.css,课堂页面将会加载本地自 定义的 js 和 css,从而可以进行自定义调试。

发布到生产环境



在本地调试完成后,通过 npm run build 打包出 js 和 css 文件,通过 自定义场景配置 上传打包的 js 和 css 文件,并在创建课堂 时传入对应的场景名称,即可在课堂中生效。

方式2: 页面调试 (需准备开发环境)

在进行调整前,需要您先准备好开发环境,请按照下文指引操作。

一、准备开发环境

在本文假定您已经具备以下技能:

- 1. 搭建本地静态服务器。
- 2. 了解浏览器端的 js/css 开发和运作原理。

参见 Web 和 H5 集成指引,假设您的本地服务器运行在8080端口,根目录下已创建文件 test.js,test.css。

二、准备测试课堂

1. 单击课堂演示的 登录页面,选择创建课堂,然后单击**立即进入**,将会跳转到课程页面。



2. 进入课程页面,您将看到如下图效果。请复制课程页面的 URL。





3. 将 JS/CSS 字符串附加到 URL 后面,附加字符串如下所示。

debugjs=http://localhost:8080/test.js&debugcss=http://localhost:8080/test.css

4. 将修改后的 URL 粘贴回浏览器,打开浏览器的控制台,检查网络请求。如果出现 localhost 文件,那么就表示操作成功,如下图 所示:



| yzeiztsrab09 | 1050033110050100110-332313 | odenv-prodaing-znaroie-acamera- | Iaiiiic-Taspeaker-TaderaultDeviceOrien | | | | | |
|--------------|--|--------------------------------------|--|--|--|--|--|--|
| | Elements Console | Sources Network Performance | Memory Application Security Lighth | | | | | |
| 頁 00:13:51) | | log Visable cache No throttling V | | | | | | |
| | localhost Invert Hide data URLs Hide extension URLs All Fetch/XHR JS CSS Img M | | | | | | | |
| ļ | Blocked response cookies | locked requests 🗌 3rd-party requests | | | | | | |
| | Big request rows | | Group by frame | | | | | |
| | ✓ Overview | | C Screenshots | | | | | |
| | 5000 ms 10000 ms 15 | 000 ms 20000 ms 25000 ms 30000 | 0 ms 35000 ms 40000 ms 45000 ms | | | | | |
| | | | | | | | | |
| | Name | X Headers Preview Response In | nitiator Timing | | | | | |
| | E class.html?classid=30825915 | ▼ General | | | | | | |
| | pv?id=oEOmJCiuPqFxeUfMV | Request URL: | http://localhost:9009/test/test.css | | | | | |
| | <pre>\$\$ whitelist?id=oEOmJCiuPqFxe</pre> | Request Method: | GET | | | | | |
| | S test.css | atus Code: | • 200 OK | | | | | |
| | 🗴 main.js | Remote Address: | 127.0.0.1:9009 | | | | | |
| | ✓ test.css | Referrer Policy: | strict-origin-when-cross-origin | | | | | |
| | 🖸 main.js | | | | | | | |
| | speed?id=oEOmJCiuPqFxeUf | ▼ Response Headers ∐ Raw | | | | | | |
| | speed?id=oEOmJCiuPqFxeUf | Accept-Ranges: | bytes | | | | | |

完成上述步骤准备好环境后,则可以开始进行功能开发。

▲ 注意:

用 vue 或者 react 开发自定义 js 的时候,不要把自定义应用挂载到#app的 DOM 节点上面,需要自己创建一个 div,插到 DOM 中再进行挂载。

基本概念介绍

最近更新时间: 2024-12-31 09:54:42

🕛 说明:

阅读本章节前,请确保您已经了解 快速开始 的内容。通过前面章节的示例,可以了解到部分课堂的生命周期状态和事件,我 们可以通过这些与课堂业务互动。

一、课堂生命周期介绍

1、流程状态

课堂加载基本流程如下:





() 说明:

- 历史原因状态值有两类,一类挂载在 TCIC.TMainState 对象上,一类在 TCICUI.Constant 对象,具体参考 TCIC-SDK API 文档。
- 除了 关键概念文案替换 等少量自定义需求需要在 Class_Info_Ready 前执行外,自定义代码的逻辑都应该在 Class_Info_Ready 状态之后再执行。

进入课堂时,打开控制台,过滤 setState 字段可以看到状态设置时序。多刷新几次会发现状态流程是固定的,有些逻辑是在状态修改后触发,例如加载自定义 js/css 文件是在学校信息获取才开始。下图是部分流程状态值说明。



| | ſo | Element | s | Console | Sources | Network | Performance | Memory | Application | Security |
|-----------------------------|-----------|-------------|---|-------------|---------------|--------------|-----------------|-------------|---------------|----------|
| I | \oslash | top 🔻 | 0 | setState | | | | | | |
| | := : | 505 mess | | Hide netwo | rk | | | | | |
| • | <u> </u> | 192 user | | Preserve lo | a | | | | | |
| r | | | | | 9 | | | | | |
| | | | U | Selected co | ontext only | | | | | |
| • | | 19 warnings | | Group simil | ar message | s in console | | | | |
| ►. | 19 wa | rnings | | Show COR | S errors in c | onsole | | | | |
| | ĕ : | 34 verbose | | TState::s | etState->j | E在加载 = ti | rue : 23 | | | |
| | | | | TState::s | etState-> | n载学校信息完 | 武 = true t: | 357 | | |
| | | | | TState::s | etState-> | 加载课堂信息完 | 成 = true t: | 676 | | |
| | | | | ► TState: | :setState | ->tcic@tmai | n@join-quit-ti | ps not reg | ister=>(0,) | |
| | | | | ► TState: | :setState- | ->tcic@tmai | n@hand-up-tips | not regis | ter=>(0,) | |
| | | | | TState::s | etState->/ | 底部工具条是否 | 显示 = true : | 0 | | |
| | | | | TState::s | etState->≨ | 且件加载完成 | = true : 1206 | | | |
| | | | | TState::s | etState−>ī | 己加入trtc房 | 间 = "entering" | t: 1082 | | |
| | | | | TState::s | etState-> | 白板已创建 = | true t: 12 | | | |
| | | | | TState::s | etState->ī | 己加入trtc房i | 司 = "waiting" | t: 50 | | |
| | | | | TState::s | etState-> | 白板已初始化 | = true t: 85 | | | |
| | | | | TState::s | etState->j | E在加载 = fa | alse : 1188 | | | |
| | | | | TState::s | etState->> | 本地音频设备状 | 춦 = 2 TTrtcWe | b: 1237 | | |
| | | | | TState::s | etState->> | 本地视频设备划 | 态 = 2 TTrtcWe | b: 1244 | | |
| | | | | TState::s | etState->> | 本地音频设备划 | 态 = 1 TTrtcWe | b: 50 | | |
| | | | | TState::s | etState->> | 本地音频采集 | = true TTrtcWe | b: 1287 | | |
| | | | | TState::s | etState->1 | [M已登录 = t | rue t: 333 | | | |
| | | | | TState::s | etState->1 | IM信令通道可用 | 引 = true t: 3 | 97 | | |
| | | | | TState::s | etState->1 | IM聊天通道可用 | 引 = true t: 3 | 98 | | |
| | | | | TState::s | etState-> | 白板已可用 = | true t: 392 | | | |
| | | | | TState::s | etState-> | 白板画刷颜色: | = "#FF0000" : | 1503 | | |
| | | | | TState::s | etState->i | 果堂状态 = 1 | t: 1632 | | | |
| | | | | TState::s | etState->i | 设备方向 = 0 | : 1630 | | | |
| TState::setState->已加入课堂 = t | | | | | | | true t: 1633 | | | |
| | | | | TState::s | etState->[| 司意连麦用户: | = "2V6Vm5UAcgi | BrdJQEqH9gr | nl1FVJ" t: 1 | 632 |
| | | | | TState::s | etState->_ | 上台状态 = tı | rue t: init 163 | 33 | | |
| | | | | TState::s | etState-> | 白板操作权限 | = true t: init | 1634 | | |

下表是部分属性的说明:

| 含义 | 状态名 | 类型 | 备注 |
|---------|---------------------------------------|---------|--|
| 学校信息已获取 | TCIC.TMainState.Sch ool_Info_Ready | Boolean | 自定义 js/css 地址是从获取学校信息接口中获 取,js/css 加载完成后, School_Info_Ready状态会变成 true。 |
| 课堂信息已获取 | TCIC.TMainState.Clas s_Info_Ready | Boolean | 获取到课堂信息后,会处理布局和文案相关内 容,所以有部分操作调用需要在 |



| | | | Class_Info_Ready 之前,例如 关键概念文 案替换 。 |
|----------|--|---------|---|
| 加入TRTC房间 | TCIC.TMainState.Join ed_TRTC | Boolean | 在改状态改变后,用户才能使用 TRTC 房间相 关功能,例如开启麦克风或者打开视频。 |
| 是否进行设备检测 | TCICUI.Constant.TSta teDeviceDetect | Boolean | 需要连麦视频的用户完成设备检测再进入房间 可以减少连麦异常的情况。 |

示例:

//屏蔽设备检测功能

```
2、状态相关方法
// 设置状态值
TCIC.SDK.instance.setState(name ,val)
/**
 * promiseState可以确保当前状态满足条件的时候立即执行一次。
 */
TCIC.SDK.instance.promiseState(name,val)
/**
 * 获取状态值
 */
TCIC.SDK.instance.getState(name)
/**
 * 示例:
 * 当TCIC.TMainState.School_Info_Ready为true时,立即执行一次
 */
TCIC.SDK.instance
 .promiseState(TCIC.TMainState.School_Info_Ready, true)
 .then(() => {
    /**
    * 表示此时可以获取到学校信息
    */
    console.log("DEMO:::schoolInfo:", TCIC.SDK.instance.getSchoolInfo());
 });
```

3、课堂事件

除开状态之外,我们还提供很多课堂事件以供使用。打开课堂控制台,过滤 Tevent:: 可以观察到很多事件行为打印。



| | ſ | Element | S | Console | Sources | Network | Performance | Memory | Application | Security | Lighthouse | Recorder 🛽 | NiM |
|-----|-----------|-------------|---|-----------------------------------|---|--|---|-------------------------------------|--------------------------------|--|---------------------------------|----------------------------------|--|
| ٩ | \oslash | top 🔻 | 0 | Tevent:: | | | | | | | | | |
| • | Ξ | 520 mess | | Hide netwo | ork | | | | | | | | Log XMLHttpRequests |
| ▶ (| 2 | 507 user | | Preserve lo | og | | | | | | | | Eager evaluation |
| (| \otimes | No errors | | Selected c | ontext only | | | | | | | | Autocomplete from hist |
| • | \wedge | 16 warnings | | Group sim | ilar message | s in console | | | | | | | Treat code evaluation a |
| ▶ (| (j) | 470 info | | Show COE | | | | | | | | | |
| • | Ă | 34 verbose | | Show COr | | | com /1 ivo /1400 | 212220 251 | 456126 mix fl | ull lloffetor | MicControllu | 0 UfachidEillT | |
| | • | | | [{"seq": []","creater layout": | 1,"taskId" ateTime":1 "top","mes | <u>יי</u> trophy","s 698739070,"נ sageRecordUr | status":1,"con pdateTime":16 "l":"","enable | tent":" 98739070," DirectCont | expireTime":0 rol":0,"inter | . <u>v</u> , offstag),"bindingUs cactionMode" | ser":"","isSel :0,"gradingAf | fUpdate":false terClass":0,"r | <pre>immit i, classiblype i }],"updateAt":169873907 roomType":0,"audienceMix</pre> |
| | | | | TEvent:: [{"userI Id":""," | tcic@tmain d":"2V6Vm5 silence":fa | @av-add-> UAcgiBrdJQEc alse,"handUp | qH9gml1FVJ","u ":false,"hand | serName":" UpTimes":0 | finlaywu","st ,"currentStat | age":true," us":1,"join | 'camera":true, Type":3,"hand | "mic":true,"sc UpTimeStamp":0 | reen":1,"board":true,"m),"device":2,"lastEnterT |
| | | | | TEvent:: [[{"user rId":"", | tcic@tmain Id":"2V6Vm "silence": | @permission- 5UAcgiBrdJQE false,"handl | -update–> EqH9gml1FVJ"," Jp":false,"han | userName": dUpTimes": | "finlaywu","s 0,"currentSta | tage":true, tus":1,"joi | "camera":true nType":3,"han | ,"mic":true,"s dUpTimeStamp": | creen":1,"board":true," 0,"device":2,"lastEnter |
| | | | | TEvent:: | tcic@tmain | @stream-upda | ate->[[{"userI | d":"mix"," | url":" <u>https:/</u> | /liveplay11 | 927.qcloudcla | ss.com/live/14 | 00313729 351456136 mix. |
| | | | | TEvent:: []","cre | tcic@tmain ateTime":1 | @task–update 698739070,"u | ed->{"seq":1," updateTime":16 | taskId":"t 98739070," | rophy","statu expireTime":0 | ıs":1,"conte),"bindingUs | ent":" ser":"","isSel | fUpdate":false | :} |
| | | | 6 | TEvent:: | :TEventUpda | ateLayout->u | ndefined | | | | | | |
| | | | | TEvent:: true"} | TStateLoad | ingCameraSta | ate->{"userId" | :"2V6V | ya ceyn | ","ty | <pre>/pe":"loading0</pre> | wnCamera","val | ue":{"loading":false,"o |
| | | | | TEvent:: false"} | TStateLoad | ingCameraSta | ate->{"userId" | :"2V6V | its canad | ","ty | /pe":"loading0 | wnCamera","val | ue":{"loading":false,"o |
| | | | | TEvent:: | TStateLoad | ingMicState- | ->{"userId":"2 | V6\ | COLUMN 1 | ","type" | ':"loadingOwnM | ic","value":{" | 'loading":false,"open":t |
| | | | | TEvent:: | TStateLoad | ingMicState- | ->{"userId":"2 | V6' | t c té artur 1 | ","type" | ':"loadingOwnM | ic","value":{" | loading":false,"open":f |
| | | | (| TEvent:: | :TEventUpda | ateLayout->u | ndefined | | | | | | |
| | | | | TEvent:: | tcic@tmain | @watermark-u | update->{"vide | o":null,"b | oard":null,"c | over":null, | "text":null," | videoRect":nul | l,"boardRect":null} |
| | | | | TEvent:: | tcic@trtc@ | audio-change | ed->{"userId": | "tic_push_ | user_3 | "," | 'available":tr | ue} | |
| | | | | TEvent:: | tcic@trtc@ | video-change | ed->{"userId": | "tic_push_ | user_3 | "," | 'available":tr | ue,"state":"up | dateStream of reason: u |
| | | | | TEvent:: [[{"user BrdJQEqH | tcic@tmain Id":"2V6 9gml1FVJ", | @permission- "lastEnterTi | -update-> "," ime":169873966 | userName": 6}],"membe | "fi ","s r_report"] | stage":true, | "camera":true | ,"mic":true,"s | creen":1,"board":true," |
| | | | | TEvent:: | TEventUpda | teLayout->ur | ndefined | | | | | | |
| | | | | TEvent:: [[{"user BrdJQEqH | tcic@tmain Id":"2V6V 9gml1FVJ", | @permission- "lastEnterTi | -update-> "'," ime":169873966 | userName": 6}],"membe | "fi","s r_report"] | tage":true, | "camera":true | ,"mic":true,"s | creen":1,"board":true," |

提供交互的事件方法如下:



下表是部分事件说明,事件可以在 TCIC.TMainEvent/TCIC.TIMEvent 里找到:



| 属性名 | 说明 |
|------------------------------|--|
| TCIC.TMainEvent.Show_Msg_Box | 当弹窗组件出现时调用,使用示例可以参考 <mark>修改下课按钮行</mark> <mark>为</mark> 。 |
| TCIC.TMainEvent.Member_Join | 课堂有人加入时会触发此事件。 |
| TCIC.TIMEvent.Recv_Msg | 收到消息时触发事件,可以获取消息内容。 |
| TCIC.TIMEvent.Send_Msg | 发送消息时触发事件,可以获取消息内容。 |

二、功能开关介绍

() 说明:

在不同的业务场景中,业务功能可能会有所不同。通过自定义配置(功能开关)的方式,可以灵活满足各类业务需求。

目前支持如下功能开关:

| 开关名称 | 默认值 | 说明 |
|-------------------------------|-------|----------------------------|
| WhiteBoardList | true | 获得白板权限后,是否支持用户新增/切换白板。 |
| WhiteBoardPPT | true | 获得白板权限后,是否支持用户课件翻页。 |
| WhiteBoardPPT.WheelPa ging | false | 获得白板权限后,是否支持用户使用鼠标滚轮对课件翻页。 |

设置功能开关应在课堂信息加载完成之后进行,以下是示例代码:



其它相关文档

更多 SDK 方法请参考 TCIC-SDK API 文档。

自定义样式 关键概念文案替换

最近更新时间: 2024-12-24 17:16:12

() 说明:

阅读本章节前,请确保您已经了解 快速开始 的内容。

功能场景

更新课堂/直播见的专有名词,以更好适配客户自己的业务场景。

页面位置



该产品主要适用于教育环境,因此大多数文本内容与课程有关。但我们也提供文本替换功能以满足更多需求。

操作方法

在 快速开始 创建的 test.js 文件中,粘贴以下代码。刷新页面即可查看效果。 例如,更换会议场景中用到的专有名词。

```
/**
 * 注入JS会在全局环境下找到TCIC对象
 * TCIC.SDK.instance 为SDK实例
 * TCIC.TMainState 为SDK状态枚举
 * School_Info_Ready 为学校信息加载完成
 * promiseState 表示当状态为School_Info_Ready时,必定会执行一次
 *
 * 注意一定要在School_Info_Ready状态下才能调用initNameConfig
 */
TCIC.SDK.instance
 .promiseState(TCIC.TMainState.School_Info_Ready, true)
 .then(() => {
    console.log("[customJS] initNameConfig");
}
```



| * 配置关键文案 |
|---|
| |
| <pre>TCIC.SDK.instance.initNameConfig({</pre> |
| zh: { |
| teacher: " 主持人 ", |
| assistant: " 联席主持人", |
| supervisor: "巡会", |
| student: "参会人", |
| visitor: " 访客", |
| defaultAppName: " 低代码互动会议", |
| room: "会议", |
| publicRoom: " 公开会议", |
| roomID: " 会议号 ", |
| roomName: " 会议名 ", |
| startRoom: " 开会 ", |
| endRoom: " 结束会议", |
| enterRoom: "进入会议", |
| leaveRoom: " 离开会议", |
| courseware: " 方案 ", |
| memberList: " 成员列表", |
| |
| |
| |

其它文档

更多 SDK 方法请参考 TCIC-SDK API 文档。

自定义页面布局 修改下课按钮行为

最近更新时间: 2024-07-29 16:36:31

| 说明: 阅读本章节前,请确保您已经了解 快速开始 的内容。 | 当我们希望老师不 | [、] 能提前下课,那2 | 么就可以修改下课按 | 钮的行为。 |
|--|----------|-----------------------|-----------|-------|
| 确认下课/离开课堂? | × | | | |
| 如果您想结束正在进行的课堂,请点击"下课" 如果您想晚点再次加入课堂,请点击"离开课堂" | | | | |
| 下课 离开课堂 取消 | | | | |

操作方法

粘贴以下代码到 快速开始 里准备好的 test.js 中,即可实现修改下课按钮。



```
let footer = dialog.querySelector(".dialog-footer");
let myclickBtn = document.createElement("BUTTON");
myclickBtn.className = "el-button el-button--primary";
myclickBtn.innerText = "确定";
myclickBtn.addEventListener("click", () => {
    /**
    * 这里填入你的地址
    */
    window.location.href =
        "https://class.qcloudclass.com/latest/login.html";
    });
    footer.appendChild(myclickBtn);
    }, 20);
    if (data.title === "确认下课/离开课堂? ") {
        data.title = "暂时离开课堂? ";
        data.title = "暂时离开课堂? ";
        data.title = [];
    }
    return data;
});
```

其它文档

更多 SDK 方法请参考 TCIC-SDK API 文档。



修改虚拟背景

最近更新时间: 2025-05-09 14:30:43

() 说明:

阅读本章节前,请确保您已经了解 快速开始 的内容。如果需要修改虚拟背景,参见如下说明:

通过设置面板修改

单击右上角设置,随后在出现的面板直接选择预设的虚拟背景即可生效。



使用自定义的虚拟背景

粘贴以下代码到 快速开始 里准备好的 test.js 中,通过 SDK 实例的方法进行设置,设置后会直接将设置的图片作为虚拟背景。



添加其他自定义虚拟背景(支持用户自己按需选择)



粘贴以下代码到 快速开始 里准备好的 test.js 中,通过下方 SDK 实例方法进行设置,即可实现在预设的虚拟背景基础上,添加自 有的虚拟背景,供用户自己选择。

```
TCIC.SDK.instance.promiseState(TCIC.TMainState.Video_Publish, true).then(() => {
    setTimeout(() => {
        const dom = document.querySelector('.raw-virtual-background-select-
component');
    const vueInstance = dom.__vue__;
    // / 下面可以添加多个虚拟背景
    vueInstance.imgArr.push({
        name: '背景',
        url: 'https://static.sxqgrow.cn/upload/beijing1.jpg',
        sceneKey: '背景',
        });
    // / 这里是默认选中哪一个虚拟背景
    vueInstance.imgArr.length = 1);
    }, 300);
    });
```

其它

更多 SDK 方法请参见 TCIC-SDK API 文档。

自定义功能 屏蔽无关业务功能

最近更新时间: 2024-07-29 16:36:31

() 说明:

阅读本章节前,请确保您已经了解 快速开始 的内容。为了帮助用户更专注,我们会精简课程页面的内容,以实现更优质的产 品体验。隐藏部分功能模块是一种有效的策略,本章节将展示两个隐藏功能的示例。

示例一: CSS 文件屏蔽

1. 首先打开浏览器控制台,鼠标右键查看元素,查找我们希望屏蔽的元素。以图中元素为例。



2. 观察到有 js 在元素上修改样式,所以我们用如下代码写入 快速开始 里准备好的 test.css。



修改效果如下:



| 实时互动–教育版 ❶ | all | ● 未开始开会(已过开会时间) | 」 屏幕共享 | 公 成员列表 | ○ 消息 | □□ □□ 更多 | 开会 |
|------------|-----|-----------------|------------------|---------------|---------|----------------|--------|
| | | | | | | | |
| | | | | | | | |
| | | 会议已到 0 位参会人 | | | | | |
| | | 开会离开会议 | | | | | |
| | | | | | | | |
| | | | | | | | K |
| | | | | | | | \sim |

同样的方式可以用于屏蔽更多内容。

⚠ 注意:

使用 CSS 强制覆盖前,建议仔细观察样式名称,避免屏蔽预期外的元素。

示例二: 结合 js 临时屏蔽模块

在示例一中,可以看到我们有些业务组件有特殊名称。如图所示,这些特殊名称可以帮助我们快速定位业务组件位置。



假设我们希望在用户操作时自动隐藏某些组件,而在不操作时则显示这些组件。这时仅仅通过 CSS 无法实现,您可以在 快速开始 创 建的 test.js 中粘贴以下代码来实现。





```
TCIC.SDK.instance.promiseState(TCIC.TMainState.Joined_Class, true).then(() => {
    console.log("%c [customJS] willHide", "font-size:16px;color:red");
    /**
    * 这些是上图所示的组件名称
    */
    let hideList = ["header-component", "videowrap-component"];
    let timer = null;
    /**
    * 监听鼠标移动事件, 用户移动鼠标就隐藏组件
    */
    document.body.addEventListener("mousemove", () => {
        hideList.forEach((item) => {
            TCIC.SDK.instance.getComponent(item).getVueInstance().hide();
        });
        clearTimeout(timer);
        /**
    * 用户停止移动鼠标1秒后,显示组件
        */
    timer = setTimeout(() => {
            hideList.forEach((item) => {
            TCIC.SDK.instance.getComponent(item).getVueInstance().show();
        });
        timer = null;
        }, 1000);
    });
    //**
    // TCIC.SDK.instance.getComponent(item).getVueInstance().show();
        });
        timer = null;
        }, 1000);
    });
    //**
    // */
}/**
```

其它文档

更多 SDK 方法请参考 TCIC-SDK API 文档。

麦克风及摄像头开关控制(学生侧)

最近更新时间: 2025-01-14 11:54:42

() 说明:

阅读本章节前,请确保您已经了解 快速开始 的内容。为了帮助用户更专注,我们会精简课程页面的内容,以提供更优质的产 品体验。隐藏部分功能模块是一种有效的策略,本章节将展示两个隐藏功能的示例。

为优化业务侧在不同教学场景的交互使用体验,目前支持配合后台 API 接口 创建房间 的参数和前端自定义方法,实现课中对学生麦克 风、摄像头权限的精细化控制。

前端自定义 JS 方法

通过以下代码即可控制上台学生的麦克风/摄像头的开/关状态:

具体场景使用说明示例

EnableDirectControl 含义: 是否允许老师/助教直接控制学生的摄像头/麦克风(及是否需要学生授权同意)。 该设置适用于以下场景:

- 当老师邀请学生上台时;
- 当老师播放音视频课件时。

自定义 js 内容含义:无需授权的情况下,麦克风和摄像头的开启/关闭状态。

| 接口配置 | 同时: 前端自定义 js 内容 | 学生侧一交互表现 |
|-----------------------|------------------------|---------------|
| 设置创建房间接口的 | micAutoOpen: false, | 麦克风:展示弹窗以进行授权 |
| EnableDirectControl=0 | cameraAutoOpen: false, | 摄像头:展示弹窗以进行授权 |
| 设置创建房间接口的 | micAutoOpen: false, | 麦克风:展示弹窗以进行授权 |
| EnableDirectControl=0 | cameraAutoOpen: true, | 摄像头:默认打开 |
| 设置创建房间接口的 | micAutoOpen: true, | 麦克风:默认打开 |
| EnableDirectControl=0 | cameraAutoOpen: false, | 摄像头:展示弹窗以进行授权 |
| 仅需设置创建房间接口的 | micAutoOpen: false, | 麦克风:默认关闭 |
| EnableDirectControl=1 | cameraAutoOpen: false, | 摄像头:默认关闭 |
| 设置创建房间接口的 | micAutoOpen: true, | 麦克风:默认打开 |
| EnableDirectControl=1 | cameraAutoOpen: true, | 摄像头:默认打开 |
| 设置创建房间接口的 | micAutoOpen: true, | 麦克风:默认打开 |
| EnableDirectControl=1 | cameraAutoOpen: false, | 摄像头:默认关闭 |





| 设置创建房间接口的 | micAutoOpen: false, | 麦克风:默认关闭 |
|-----------------------|-----------------------|----------|
| EnableDirectControl=1 | cameraAutoOpen: true, | 摄像头:默认打开 |

配置业务功能 分角色设置下课倒计时提醒

最近更新时间: 2024-12-24 17:16:12

() 说明:

腾讯云

阅读本章节前,请确保您已经了解 快速开始 的内容。

功能描述

区分角色,针对老师、巡课、助教、学生在距离下课还有「x分钟」时,进行弹窗提醒。

页面展示

弹窗文案仅供参考。

| 温馨提示 | × |
|-------------------------|---|
| 本堂课将于 5 分钟后结束,请老师合理安排时间 | |
| 确定 | |

操作方法

粘贴以下代码到 快速开始 里准备好的 test.js 中,即可实现下课倒计时提醒。 以下案例为:距离下课5分钟时,老师侧进行弹窗提醒。

```
TCIC.SDK.instance
.promiseState(TCIC.TMainState.Class_Status, TCIC.TClassStatus.Already_Start)
.then(() => {
    const classInfo = TCIC.SDK.instance.getClassInfo();
    const { endTime, startTime } = classInfo;
    const now = Date.now();
    // 提前 5 分钟弹窗
    /**
    * note: 您也可以通过订阅 TCIC.TMainState.Class_Duration 来获取课程进行时间,做些定制化处理
    * TCIC.SDK.instance.subscribeState(TCIC.TMainState.Class_Duration, console.warn)
    */
    const timeToAlert = (endTime = 5 * 60) * 1000;
    if (TCIC.SDK.instance.isTeacher() && now <= timeToAlert) {
        setTimeout(() => {
            TCIC.SDK.instance.showMessageBox('温馨提示', '本堂课将于 5 分钟后结束,请老师合理安排
        timeToAlert = now);
        }, timeToAlert = now);
    } else {
    }
}
```







配置白板功能权限

最近更新时间: 2025-06-06 15:29:42

说明:
 阅读本章节前,请确保您已经了解 快速开始 的内容。

功能描述

场景

课中老师邀请学生上台后,需要给学生开启白板权限。不同行业场景里,学生需要的白板功能不同,故支持客户根据应用 id 维度自定 义配置功能(对应前端隐藏操作入口)。

默认配置

白板权限开启后,新增/删除白板、涂鸦权限、课件操作(翻页、放大/缩小)功能都开启。

页面位置



操作方法



粘贴以下代码到 快速开始 里准备好的 test.js 中,即可实现配置白板功能权限。 例如,仅开启学生在白板上的涂鸦权限,如下代码案例:

```
TCIC.SDK.instance.promiseState(TCIC.TMainState.Board_Ready, true).then(()
TCIC.SDK.instance.subscribeState(TCIC.TMainState.Board_Permission, async (value)
     await TCIC.SDK.instance.promiseState(TCIC.TMainState.Board_Ready, true);
```

配置签到功能

最近更新时间: 2025-04-10 17:03:12

签到是课堂教学中的一个通用功能,老师可在课堂中单击对应的按钮,学生端可在弹出的签到弹窗上单击按钮完成签到。 下文提供了使用 Vue 实现签到功能的方法,完整代码请点击 这里 查看。

开发老师端

我们通过 useTask 创建一个 Task,当老师调用 updateTask 时,学生端会收到老师端发来的签到请求。

```
<template>
<div class="custom-checkin-btn" @click="checkIn">
<span class="header__btn-text">签到</span>
</div>
</template>
<script setup>
// 创建一个签到任务 custom-check-in-tool
const { updateTask } = useTask('custom-check-in-tool');
// 点击签到按钮时,发起签到
const checkIn = () => {
 updateTask({
 type: 'ask-check-in',
 });
};
</script>
<style lang="less">
....
</style>
```

开发学生端

在学生端,我们监听 custom-check-in-tool 这个任务,当收到签到请求时,进行弹窗,让学生完成签到。

```
<template>
<div class="checkin-modal" v-if="showModal">
<div class="checkin-modal__content">
<div class="checkin-modal__header">
</div>
</div>
</div class="checkin-modal__body">
</div>
</div class="checkin-modal__body">
</div class="checkin-modal__body">
</div class="checkin-modal__body">
</div class="checkin-modal__body">
</div class="checkin-modal__footer">
</div class="checkin-modal_footer">
</div class="checkin-modal_footer">
</div class="checkin-modal_footer">
</div class="checkin-modal_footer">
<
```





```
// 监听到老师发起签到任务的回调函数
  console.log('收到签到请求');
  // 检查是否已签到,没有签到就展示签到弹窗
       // 设备检测完成后展示弹窗
   console.log('已经签到');
 // 发送请求到服务器
```

🕛 说明:

完成开发后,我们可以通过 场景配置 将打包后的 js 上传,创建该场景的课堂后,课堂将自动加载自定义 js。



事件监听

最近更新时间: 2025-04-10 17:03:12

使用场景

在实际业务需求中,您可能需要在特定事件发生时进行与业务相关的处理,例如:

• 当上课正式开始时,向业务后台进行一些上报处理。

• 当成员加入房间时,向成员展示弹窗。

您可以通过 TCIC.SDK.instance.on(eventName, handler) 监听成员进入退出等课堂事件,或者通过 TCIC.SDK.instance.subscribeState(stateName, handler) 监听课堂状态变更。

事件列表(TCIC.TMainEvent)

| Event | 事件 |
|------------------------|--------------|
| After_Enter | 已加入房间 |
| Modify_Class | 房间信息已更改 |
| Leave_Class | 离开房间 |
| Kick_Out_By_Teacher | 被踢出房间 |
| Kick_Out_By_Another | 多端登录被踢出房间 |
| Kick_Out_By_Expire | 签名过期被踢出房间 |
| Member_Join | 成员加入房间 |
| Member_Exit | 成员退出房间 |
| Member_Info_Update | 成员信息更新 |
| Member_Hand_Up | 成员举手 |
| Member_Hand_Up_Cancel | 成员取消举手 |
| Question_Valid | 存在可用答题 |
| Question_Begin | 答题开始 |
| Question_End | 答题结束 |
| Question_Abandon | 终止答题 |
| Question_Close | 关闭答题 |
| Question_Been_Answered | 有学生作答 |
| App_Resized | 应用大小变化 |
| Error | 发生错误(影响主线流程) |
| Recv_IM_Msgs | 收到 IM 消息 |



Recv_Custom_IM_Msg

收到自定义 IM 消息

使用示例:

```
function afterEnter() => {
  console.debug('You have joined this room');
}
// 监听
TCIC.SDK.instance.on(TCIC.TMainEvent.After_Enter, afterEnter);
// 取消监听
TCIC.som (TCIC.TMainEvent.After_Enter, afterEnter);
```

状态列表(TCIC.TMainState)

| Event | 事件 | 说明 |
|--------------------------------------|----------------|---|
| Class_Info_Ready | 课堂信息已加载 | - |
| Joined_Class | 已加入课堂 | _ |
| Sub_Camera | 辅助摄像头状态 | ● 0:开始 ● 2: 结束 |
| Screen_Share | 屏幕分享状态 | ● 0:分享中 ● 1:暂停中 ● 2:未开始/已结束 |
| Video_Publish | 本地视频推流是否开 启 | _ |
| Audio_Capture | 本地音频采集是否开 启 | _ |
| Class_Duration | 课堂持续时间 | 单位秒。 • < 0:未到上课时间,距离上课开始的时间。 • = 0:到上课时间未开始上课、课堂已结束、课堂已过期。 • > 0:上课中,已开始上课的时间。 |
| Member_Count | 课堂当前在线成员数 量 | 仅学生在线数,不包含老师、助教和巡课。 |
| Member_List_Total_Mem ber_Count | 成员总数 | 包括老师,助教在内,在线和离线的成员总数。 |
| Member_List_Offline_Me mber_Count | 离线成员总数 | 包括老师,助教在内,离线的成员总数。 |
| Board_Permission | 白板操作权限 | _ |



| Chat_Permission | 文字聊天权限 | _ |
|-----------------------------|--------|--|
| Screen_Share_Permissio n | 屏幕分享权限 | _ |
| Hand_Up | 举手状态 | _ |
| Mute_All | 全员静音状态 | _ |
| Mute_Video_All | 全员视频状态 | _ |
| Silence_All | 全员禁言状态 | _ |
| Message_Unread_Count | 未读消息 | _ |
| HandUp_Count | 举手人数 | _ |
| Class_Status | 课堂状态 | 0:未开始 1:已开始 2:已结束 3:已过期 |

// promiseState 可以确保当前状态满足条件的时候立即执行一次

TCIC.SDK.instance.promiseState(TCIC.TMainState.Joined_Class, true).then(() => {
 console.debug('You have joined this room');

});

```
function listener() ·
```

console.debug('You have joined this room');

}

// 监听

TCIC.SDK.instance.subscribeState(TCIC.TMainState.Joined_Class, listener);

// 取消监听

TCIC.SDK.instance.unsubscribeState(TCIC.TMainState.Joined_Class, listener);