
Gensee web 接口

开发常见问题

FAQ

修订记录

修订日期	描述	修订人
2016-4-21	创建编辑	Wenkai.li

目录

修订记录.....	2
目 录.....	3
1 介绍.....	5
1.1 目的和范围.....	5
1.2 接口文档类型和功能介绍.....	5
2 URL API 接口开发调试常见问题.....	6
2.1 为何 URL 传入昵称后依然无法跳过登陆界面?	6
2.2 为何传入中文昵称后客户端看到的用户昵称为乱码?	7
2.3 为何传入 uid 后会即使不重复 web 用户也会互相踢掉?	7
2.4 嵌入三分屏的 flash 播放器, 但 flash 在当前页面的最前端, 遮盖原页面元素。.....	7
2.5 为何使用第三方认证 (K 值认证) 总是被跳转认证失败地址。.....	9
3 Json API 接口开发调试常见问题.....	10
3.1 Json 调试页面如何使用?	10
3.2 直播 ID (webcastID)、课堂 ID (roomId) 指的什么?	11
3.3 通过 json 创建直播, sec=true 时, 密码 (password) 采用什么加密算法?	12
3.4 为何通过 json 创建直播房间, 开始时间总是 1970 年?	13
3.5 通过 json 接口获取的返回数据是否进行了排序?	13
3.6 课件/已发布点播与录制件有什么区别?	13
3.7 Json 创建直播时, 开始时间与结束时间是什么关系逻辑?	14
3.8 直播录制结束后如何自动获取到点播地址, 并且如何知道是哪个房间录制的?	14
3.9 castmaker 上传的课件如何获取点播地址?	16
3.10 使用 json 修改直播房间配置, 未发生变化的值怎么处理?	16
3.11 如何获取直播、点播的访问记录?	17
3.12 为何 json 数据解析不了?	19
4 WEB SDK 接口开发调试常见问题.....	20
4.1 WEB SDK 的必填参数都有哪些? ownerid 填的什么?	20
4.2 为何 WEB SDK 调试页面无法进入, 怎么用?	21
4.3 为何自己写的 WEB SDK 打开后一直提示 loading?	22
4.4 为何自己写的 WEB SDK 打开后提示 not_found?	22
4.5 为何自己写的 WEB SDK 打开后收不到任何监听事件消息?	23
4.6 使用 WEB SDK 开发后的视频能全屏吗?	24
4.7 是否可以指定 video 的背景图? 设置完 bgimg 参数但是不生效?	25
4.8 当桌面共享时, 文档 (doc)、视频 (video) 位置大小互换 效果如何实现?	26
4.9 如何实现对啊网播放器那种在聊天前增加对应身份图标?	27
4.10 公聊如何实现屏蔽关键字、控制发言间隔?	28
4.11 WEB SDK 聊天 submitChat 如何实现发送表情?	29
4.12 为何 onUserList -- 用户列表事件收不到消息?	33
4.13 为何 onUserOnline -- 用户在线人数事件收不到消息?	34
4.14 为何使用 submitChatTo -- 提交私聊信息发送了私聊, 但 onPriChat -- 收到私聊消息事件收不到任何消息?	36
4.15 为何点播 SDK 中的 onQAList、onChatHistory 收不到任何消息?	36

4.16	如何实现未登陆用户试看一定时间自动弹出购买和注册窗口（广告）？	37
4.17	WEB SDK 如何实现暖场的效果？	37
4.18	聊天是否支持自定义第三方表情？	37
4.19	使用了 WEB SDK 开发了自定义播放器如何实现同一链接不同设备观看？（达到类似 Gensee WEB 端在 PC 和移动端打开效果不同）	37
5	实时回调接口开发调试常见问题	38
5.1	实时回调接口在哪里设置？怎么填写？	38
5.2	实时回调接口已经全部开发调试完成，为何有时候会突然收不到推送的回调消息？ ..	39
6	移动端及其他常见问题	40
6.1	使用 webview 在 IOS APP 上嵌入直播链接，视频自动全屏浮出如何解决？	40
6.2	为何使用 iframe 嵌入 URL 后，移动端打开有界面适配的问题？如何解决？	41
6.3	移动端如果使用 WEB SDK 开发自定义播放器，如何调试？	41
6.4	视频在手机浏览器上观看为何会自动全屏浮出，导致无法看到文档、聊天？	41
6.5	使用 WEB SDK 开发播放器在移动端 H5 界面上视频区域 div 隐藏后就会没有声音。 ..	42

1 介绍

1.1 目的和范围

此文档目的是解决客户研发人员在开发 Gensee 相关接口时遇到的常见问题及疑惑。

目标读者：研发人员

开发前提：为了提高开发和沟通效率，请务必在充分了解产品功能及业务需求情况下进行接口开发，请开发人员在开发前先联系 Gensee 相关人员进行产品功能培训。谢谢。

1.2 接口文档类型和功能介绍

- 1. URL API 文档：**用于使用 `iframe` 方式嵌入 WEB 端 URL，并且可以传递参数，无需手工输入昵称和房间口令。使得直播看上去像是网站的一部分，属于一种简单的集成方式。
- 2. json API 文档：**用于通过接口的方式，完成对 Gensee 后台的操作（如新建直播，编辑直播，获取直播房间 URL，获取点播 URL 等信息）
- 3. WEB SDK 文档：**如果对于我们的标准三分屏播放器、互动增强播放器等样式不满意，可以使用此接口自定义开发直播和点播的播放器。
- 4. APP SDK 文档：**此接口可以用于开发原生 android、IOS 的 APP。
- 5. 实时回调接口文档：**用于实时接收直播课堂状态和用户状态（如直播课堂开启、结束、用户进入、离开、以及当前房间实时人数）

官网接口文档下载地址：

<http://www.gensee.com/document.php>



我司分为三个产品线：webcast（大会直播）、training（教育培训）、meeting（视频会议）
请下载使用对应产品的接口文档（APP SDK 开发工具包为三个产品线公用开发包）

2 URL API 接口开发调试常见问题

2.1 为何 URL 传入昵称后依然无法跳过登陆界面?

答：1.请确认传入昵称参数名是否正确（webcast 产品昵称参数为 nickName，training 产品昵称参数名为 nickname，区分大小写），并且正确传入对应身份口令（token）。

2.取消直播配置中【是否要求用户登录】。

Webcast: 示例: [http://demo.gensee.com/webcast/site/entry/join-](http://demo.gensee.com/webcast/site/entry/join-8a65b8abb3ff4fbb820e0ae3fa3d080c?nickName=test_User&token=333333)

[8a65b8abb3ff4fbb820e0ae3fa3d080c?nickName=test_User&token=333333](http://demo.gensee.com/webcast/site/entry/join-8a65b8abb3ff4fbb820e0ae3fa3d080c?nickName=test_User&token=333333)

直播向导	
➡ 基本信息 邀请嘉宾 插播/暖场 资料/文档 播放器 指定参会者 指定广告	
客户端类型 *	<input type="radio"/> 标准版 <input checked="" type="radio"/> 轻里版
主 题 *	<input type="text" value="wenkai.li-test"/>
开始时间 *	<input type="text" value="2016-12-08 11:22"/>
结束时间	<input type="text"/>
持续时间	<input type="text"/>
最大并发人数	<input type="text"/>
是否公开	<input type="checkbox"/> 要求公开
是否要求用户登录	<input type="checkbox"/> 需要
组织者口令 *	<input type="text" value="111111"/>
嘉宾口令 *	<input type="text" value="222222"/>
普通参加者口令	<input type="text"/>
目 录	<input type="text"/> <input type="button" value="选择目录"/>
客户端信息栏URL	<input type="text"/>

Training: 示例:

http://demo.gensee.com/training/site/s/29160259?nickname=test_User&token=444444

新建课堂
新建课件
课堂管理
课件管理
课程维护

实时课堂
修改实时课堂

课堂类型: 全部 课堂名称 编号 搜索

编号

94070559
22890629
93517965
56895240
57426266
67686109
71187842
04224382
94439590
38929411

* 课堂名称 新东方Mac大讲堂测试

* 所属课程 / 选择

* 开始时间 2016-01-12 17:32:28

失效时间

最大并发数

场景模板选择 ☒ 大讲堂 ☐ 小班课

是否公开 ☐ 要求公开

☒ 是否要求用户登录 ☐ 需要

Web端是否允许切换 ☐ 允许

2.2 为何传入中文昵称后客户端看到的用户昵称为乱码?

答: 传入的中文昵称要统一采用 UTF-8 编码, 并且中文一律要先经过 URL 编码处理。

如 JavaScript 代码示例:

```
var url = http://demo.xxx.com/webcast/site/entry/join-123456?nickName=+encodeURIComponent('中文');
```

2.3 为何传入 uid 后会即使不重复 web 用户也会互相踢掉?

答: uid 要严格开发文档中的取值范围, 超出取值范围会导致实时系统错乱。

1000000000 < UID < 32 位 (4 字节) 即 4294967296。

直播踢出逻辑为, 相同直播房间内, 相同 UID, 后者踢出前者。

2.4 嵌入三分屏的 flash 播放器, 但 flash 在当前页面的最前端, 遮盖原页面元素。

答: 举例: 如下图当页面设计中有横幅或其他需要在 flash 前显示的元素, 直接嵌入后会发现原

本最前面的 div 被 flash 遮盖在后面 (下图已经过处理, 所以访问记录正常)。



问题描述:

<http://www.blueidea.com/tech/web/2009/6469.asp>

解决办法有以下三种:

方法一:

追加透明度参数 `trans=true`。

```
<button data-event="test">open dialog</button>
<iframe id="record_view" frameborder="0" src="http://iflytek.gensee.com/webcast/site/vod/play-b716ce6274564535b07f4fd09741ea61?nickName=xxx&token=KgPpOCgm8nsTgtL8trans=true" width="100%" height="500px"></iframe>
<script src=".../lib/sea.js"></script>
<script>
```

缺点: 三分屏播放器需要用户输入聊天和 Q&A, 如果设置为 flash 透明, 经过测试发现有很多输入法会被影响导致用户无法输入。

方法二:

用 `iframe` 嵌入后, 会发现 flash 覆盖问题, 从技术上, 不是说只有 flash 设置为透明才能解决。客户开发人员可以使用另外一个 `iframe` (层级比较高) 来实现遮盖三分屏播放器。

`iframe` 在页面里面层级是最高的。可以遮挡任何东西 (包括 flash)。只要将客户希望遮挡在三分屏上的内容放在另外一个 `iframe` 里面, 就可以实现这个 `iframe` 盖在其他元素之上。

技术细节请度娘。

方法三:

Web SDK 可以终极解决。Web SDK 设计时候已经考虑到原有的问题, 因此没有键盘输入功能。所以默认 flash 设置即是透明。

2.5 为何使用第三方认证（K 值认证）总是被跳转认证失败地址。

答：通过以往对接经验，有以下几种可能性，供参考。

1. 没有在站点管理后台启用并配置第三方权限认证 URL 地址。
2. 直播/点播 URL 地址没有带上可用的 k 值参数。
3. Gensee 系统收到接口调用请求后，会向第三方认证 url 发送 http post 请求，同时把 k 参数作为 post 数据提交回第三方系统。由第三方系统重新验证 k 值合法性。若第三方认证通过，第三方认证 url 返回字符串” pass”，否则返回” fail”。除了 pass/fail 以外，页面中包含其他 html 标签信息（如 title body 等）。
4. 曾出现过客户方验证 web 服务器不支持 post 请求，Gensee 系统根据第三方认证 URL 返回值判断认证是否成功：当 http response 状态不为 200 且 http response 的内容不为 pass 时，接口将跳转到指定的第三方认证失败地址。若该地址未配置，则跳转到 Gensee 的默认出错页面；若认证成功，则进入直播/点播页面。
5. 自行验证接口方法：使用 postman 或百度 API（<http://apistore.baidu.com/astore/toolshttpproxy>）等工具，模拟向自己开发的验证接口发送 post 请求，调试返回值是否正常。

PS：K 值认证失败可以通过查看后台站点中的 日志监控 来协助定位原因。

Webcast:



Training:



3 JSON API 接口开发调试常见问题

3.1 Json 调试页面如何使用?

答:

4.1 测试工具的使用

如开了一个站点 test 域名是 test.gensee.com 那么测试工具的地址就是 <http://test.gensee.com/webcast/static/integration/tool.html> 域名必须填写正确。在测试工具中选择教育系统就可以选着对应的接口, 测试工具的左边是参数的填写, 点击发送之后的结果会显示在右边。

Gensee 采用多站点运营的 SAAS 模式, 故给每个合作客户均建立了独立站点域名。如 <http://demo.gensee.com> 或 <http://test.gensee.com> 或 <http://support.gensee.com>

故调试页面为 <http://{站点域名}/webcast/static/integration/tool.html>

如开放站点为 <http://demo.gensee.com> 则该客户调试页面地址为:

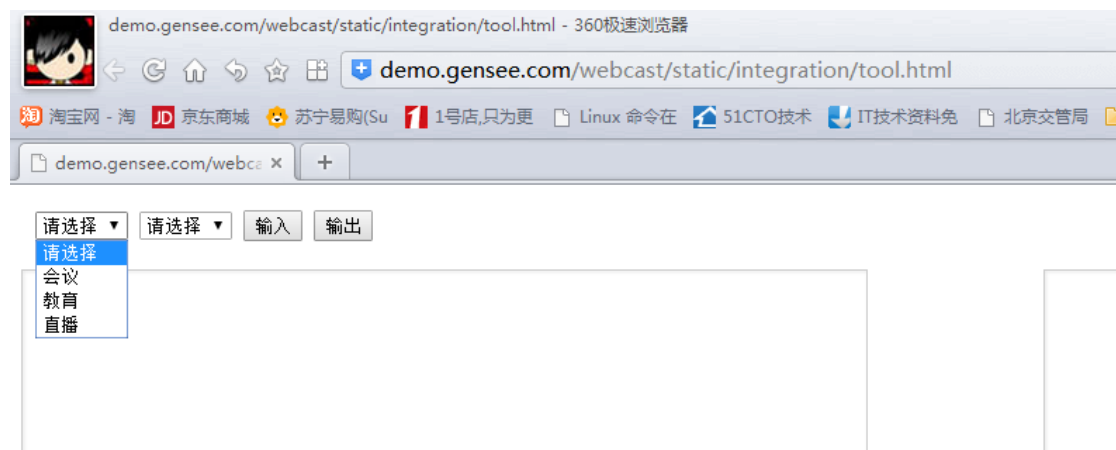
<http://demo.gensee.com/webcast/static/integration/tool.html>

选择对应的产品

会议 (meeting)

教育 (training)

直播 (webcast)



3.2 直播 ID (webcastID)、课堂 ID (roomId) 指的什么?

答: 直播 ID、课堂 ID 为唯一值。

Webcast 产品:

直播信息

基本信息

嘉宾

文档

投票调查

问卷

问答

聊天

录制

主 题	接口开发常见问题FAQ-测试房间		
编 号	51181104		
开始时间	2016-01-18 14:54		
结束时间			
持续时间			
是否公开	不公开		
是否要求登录	否		
Web端是否允许切换	否		
加入URL(web)	http://demo.gensee.com/webcast/site/entry/join-8a65b8abb3ff4fbb820e0ae3fa3d080c		复制链接
加入URL(Client)	http://demo.gensee.com/webcast/site/entry/live-8a65b8abb3ff4fbb820e0ae3fa3d080c		复制链接

直播ID (webcastID)

Training 产品:

实时课堂

实时课堂信息

编号

52597378

29180259

40504279

71440306

71643564

50262215

58857634

70712700

26941461

25800745

加入URL:	http://demo.gensee.com/training/site/r/	复制链接
学生端加入URL:	http://demo.gensee.com/training/site/s/	复制链接
老师口令:		
助教口令:		
学生客户端口令:		
学生WEB端口令:		
所属课程:	/	
是否公开:	不公开	
是否要求用户登录:	不需要	
Web端是否允许切换:	不允许	
无延迟模式:	不允许	
SDK ID:	CD9uw1X9Wz	
演讲人介绍:		

课堂房间ID, roomId, SDK ID,

3.3 通过 json 创建直播, sec=true 时, 密码 (password) 采用什么加密算法?

答: 目前加密算法为 md5、sha-1 二选一, 默认为 md5, 可以在后台进行设置。

Webcast 产品:



Training 产品:



3.4 为何通过 json 创建直播房间，开始时间总是 1970 年？

答：一般出现该问题都是由于误把毫秒时间戳当成秒时间戳来处理导致。请核对检查。

3.3 时间规则

发送参数时间格式: yyyy-MM-dd HH:mm:ss 或者为 long 型(如果时间格式为 long 型那么该值为从 1970 年 1 月 1 日 00:00:00 GMT 以来此时间对象表示的毫秒数)

返回参数时间格式: long, (该 long 型那么该值为从 1970 年 1 月 1 日 00:00:00 GMT 以来此时间对象表示的毫秒数)。

举例：2016 年 1 月 20 日 18:18:18 对应毫秒时间戳为：1453285098000

工具：<http://www.box3.cn/#>



3.5 通过 json 接口获取的返回数据是否进行了排序？

答：目前 json 返回的数据，如：分页同步点播数据、访问记录等为了提高响应性能，均未进行排序处理，如果需要排序显示，请客户方技术通过 json 对应接口，把数据同步到客户自己数据库中才进行排序显示。

3.6 课件/已发布点播与录制件有什么区别？

答：

课件：training 产品直播录制下来或 castmark 上传可直接用于点播的视频。

已发布点播：webcast 产品直播录制下来或 castmark 上传可直接用于点播的视频。

课件、已发布点播与录制件区别：录制件可以理解为存储在服务器上的物理文件，故录制件有诸如存储大小、录制开始时间、录制结束时间等属性。录制件用户是无法直接观看访问的。而课件和已发布点播，抽象理解相当于把录制件开放出一个可供访问的 URL 地址（俗称发布）。一般情况下课件、已发布点播与录制件为一一对应关系，即每个课件 ID/点播 ID，对应一个录制件 ID，也可以粗略把两者理解为一个东西。

3.7 Json 创建直播时，开始时间与结束时间是什么关系逻辑？

客户端类型 *	<input checked="" type="radio"/> 标准版 ? <input type="radio"/> 轻量版 ?
主 题 *	接口开发常见问题FAQ-测试房间
开始时间 *	2016-01-18 14:54
结束时间	
持续时间	

答：开始时间为必填参数，在设置的开始时间之前，用户均可以进入，但在超过结束时间后直播房间就不允许再进入。结束时间为选填参数，可以为空，当为空时，意味着该直播房间永久生效，房间 URL 一直有效，可以重复使用该房间。

3.8 直播录制结束后如何自动获取到点播地址，并且如何知道是哪个房间录制的？

答：有以下两种方式解决。返回值中均由该点播由哪个直播房间 ID 产生。

1.在后台写个定时器，定时（如每天夜里或每隔 30 分钟）通过 json 的分页同步接口把点播地址同步到本地数据库中。

Webcast 产品：

5.24 分页同步点播数据
5.24.1 描述
分页获取本站点中已发布的点播数据，每页最大返回 50 条数据。
在很多场合下，第三方希望能同步 Gensee 的已发布点播数据到自己的系统中，第三方业务系统在同步数据后可以根据自己的业务进行查询，对账等操作。
该 API 要求认证用户具有管理员角色。

31

5.24.2 接口详情

接口路径	/webcast/vod/sync	请求方式	POST/GET
接口说明	分页同步点播数据		
	属性名称	类型	说明
	pageNo	int	指定第几页，默认为 1
	pageSize	String	指定每页返回的记录数，默认为 50

Training 产品：

5.11 分页同步课件数据

5.11.1 描述

分页获取本站点中的课件数据。每页最大返回 50 条数据。
在很多场合下，第三方希望能同步 Gensee 的课件数据到自己的系统中。
第三方业务系统在同步数据后可以根据自己的业务进行查询，对账等操作。
第三方系统可以使用课件进行观看。

该 API 要求认证用户具有管理员角色。

5.11.2 接口详情

接口路径	/training/courseware/syn		请求方式	POST/GET
接口说明	分页同步课件数据			
	属性名称	类型	说明	是否可为空
	pageNo	int	指定第几页，默认为 1	是
	startTime	String	同步创建时间大于等于 startTime 的录制件（详见 3.3）	是
	endTime	String	同步创建时间小于等于 endTime 的录制件（详见 3.3）	是
	sec	String	true: 表示密码是经过加密的。	是
	loginName	String	登录名	
	password	String	密码	

2. 如果不想采用定时的方式，而是想每个直播房间录制结束后就自动获取，则需要使用【实时回调接口】实时监测站点下所有房间的状态码，当回调消息收到停止录制消息时，自动触发 json 的分页同步点播数据接口（因平台数据量巨大，点播数据入库需要时间，建议收到停止录制回调消息后，等待 3-5 分钟后再进行获取，避免立即获取取不到数据情况，具体回调状态码参照【实时回调接口】文档）。

3.9 castmaker 上传的课件如何获取点播地址？

答：依然可以使用 json 分页同步点播/课件 接口，获取 castmark 上传的视频。
当 webcastId（webcast 产品）或 roomId（training 产品）为空时，即该点播并非直播房间录制，而是 castmark 上传。

Webcast 产品：

5.24 分页同步点播数据

5.24.1 描述

分页获取本站点中已发布的点播数据。每页最大返回 50 条数据。
在很多场合下，第三方希望能同步 Gensee 的已发布点播数据到自己的系统中。第三方业务系统在同步数据后可以根据自己的业务进行查询，对账等操作。

该 API 要求认证用户具有管理员角色。

31

5.24.2 接口详情

接口路径	/webcast/vod/sync			请求方式	POST/GET
接口说明	分页同步点播数据				
	属性名称	类型	说明	是否可为空	
	pageNo	int	指定第几页，默认为 1		

Training 产品：

5.11 分页同步课件数据

5.11.1 描述

分页获取本站点中的课件数据。每页最大返回 50 条数据。
在很多场合下，第三方希望能同步 Gensee 的课件数据到自己的系统中。
第三方业务系统在同步数据后可以根据自己的业务进行查询，对账等操作。
第三方系统可以使用课件进行观看。

该 API 要求认证用户具有管理员角色。

5.11.2 接口详情

接口路径	/training/courseware/syn			请求方式	POST/GET
接口说明	分页同步课件数据				
	属性名称	类型	说明	是否可为空	
	pageNo	int	指定第几页，默认为 1	是	
	startTime	String	同步创建时间大于等于 startTime 的录制件（详见 3.3）	是	
	endTime	String	同步创建时间小于等于 endTime 的录制件（详见 3.3）	是	
	sec	String	true:表示密码是经过加密的。	是	
	loginName	String	登录名		
	password	String	密码		

3.10 使用 json 修改直播房间配置，未发生变化的值怎么处理？

答：使用 json 的修改直播课堂接口时，如果参数值较之前未发生变化，仍需把参数和值一并发送过来，如不发送参数名及对应值，则后台会被滞空处理。

3.11 如何获取直播、点播的访问记录？

答：使用分页同步接口完成。

Webcast 产品：

直播观看记录：

5.13 导出直播用户访问历史记录

5.13.1 描述

根据开始时间和结束时间，导出加入时间在此范围内的参会者访问记录
如果没有提供开始时间和结束时间，则找到最近结束的直播场次，导出该场次的访问记录。

注意：开始时间和结束时间间隔不能超过 7 天。

该 API 要求认证用户具有组织者或管理员角色。
管理员可以操作站点内所有直播，组织者只能操作自己创建的直播。

5.13.2 接口详情

接口路径	/webcast/export/history		请求方式	POST/GET
接口说明	导出直播用户访问历史记录			
	属性名称	类型	说明	是否可为空
	webcastId	String	直播 ID	
	startTime	String	导出的数据大等于开始时间	是
	endTime	String	导出的数据小等于结束时间	是
	sec	String	true:表示密码是经过加密的。	是
	loginName	String	登录名	
	password	String	密码	
返回参数	list	JSONArray		

点播观看记录：

5.33 按天同步点播访问记录

5.33.1 描述

为点播导出按天同步访问记录。
该 API 要求认证用户具有管理员角色。

5.33.2 接口详情

接口路径	/webcast/export/vod/history			请求方式	POST/GET
接口说明	按天同步点播访问记录				
	属性名称	类型	说明	是否可为空	
	pageNo	Int	页数（从 1 开始）	是	
	date	String	开始时间（详情见 3.3）		



	hour	int	小时（0-23）不填默认为全天数据	是
	sec	String	true:表示密码是经过加密的。	是
	loginName	String	登录名	
	password	String	密码	

Training 产品：

直播观看记录：

5.18 导出课堂参会记录

5.18.1 描述

根据开始时间和结束时间，为课堂导出加入时间在此区域的学生参课记录。
如果没有提供开始时间和结束时间，则找到最近结束的课堂场次，导出该场次的所有访问记录。

注意：开始时间和结束时间间隔不能超过 7 天。

该 API 要求认证用户具有老师角色。

5.18.2 接口详情

接口路径	/training/export/history			请求方式	POST/GET
接口说明	导出课堂参会记录				
	属性名称	类型	说明	是否可为空	
	roomId	String	课堂 ID		
	startTime	String	开始时间	是	
	endTime	String	结束时间	是	
	sec	String	true:表示密码是经过加密的。	是	
	loginName	String	登录名		
	password	String	密码		
返回参数	list				
	uid	long	学生 ID		
	nickname	String	参课昵称		

点播观看记录:

5.25 按天同步课件访问记录

5.25.1 描述

为课件导出按天同步访问记录。
该 API 要求认证用户具有管理员角色。

5.25.2 接口详情

接口路径	/training/export/study/history			请求方式	POST/GET
接口说明	按天同步课件访问记录				
	属性名称	类型	说明	是否可为空	
	pageNo	Int	页数（从 1 开始）	是	
	date	String	开始时间（详情见 3.3）		
	hour	int	小时（0-23）不填默认为全天数据	是	

29

	sec	String	true:表示密码是经过加密的。	是
	loginName	String	登录名	
	password	String	密码	

3.12 为何 json 数据解析不了?

答：在获取访问记录时，出现 json 格式数据无法解析的情况，主要由于制表符导致。

处理办法：替换制表符。

```
$txt=str_replace(array("\t","\r","\n"),",$txt);
```

`\n`

换行(LF)，将当前位置移到下一行开头

010

`\r`

回车(CR)，将当前位置移到本行开头

013

`\t`

水平制表(HT)（跳到下一个TAB 位置）

009

如：



4 WEB SDK 接口开发调试常见问题

4.1 WEB SDK 的必填参数都有哪些? ownerid 填的什么?

答:

Webcast 产品:

主 题	接口开发常见问题FAQ-测试房间	
编 号	51181104	注: webcast产品则需增加ctx="webcast"
开始时间	2016-01-18 14:54	
结束时间		
持续时间		
是否公开	不公开	
是否要求登录	否	
Web端是否允许切换	否	
加入URL(web)	http://demo.gensee.com/webcast/site/entry/join-8a65b8abb3ff4fbb820e0ae3fa3d080c	复制链接
加入URL(Client)	http://demo.gensee.com/webcast/site/entry/live-8a65b8abb3ff4fbb820e0ae3fa3d080c	复制链接
组织者口令	111111	ownerid, 直播ID
嘉宾口令	222222	
普通参加者口令	333333	authcode即web端口令

```
<gs:video-live id="videoComponent" site="demo.gensee.com" ctx="webcast"
ownerid="8a65b8abb3ff4fbb820e0ae3fa3d080c" uname="user5623" authcode="333333"/>
```

Training 产品:

实时课堂	实时课堂信息
号	
160259	场景模板: 大讲堂
504279	学生登录方式: 所有
140306	加入URL: http://demo.gensee.com/training/site/r/29160259 复制链接
343564	学生端加入URL: http://demo.gensee.com/training/site/s/29160259 复制链接
262215	老师口令: 111111
357634	助教口令: 222222
712700	学生客户端口令: 333333
341461	学生WEB端口令: 444444
300745	所属课程: /
548617	是否公开: 不公开
	是否要求用户登录: 不需要
	Web端是否允许切换: 不允许
	无延迟模式: 不允许
	SDK ID: CD9uw1X9Wz
	演讲人介绍:

```
<gs:video-live id="videoComponent" site="demo.gensee.com" ctx="training"
ownerid="CD9uw1X9Wz" uname="user5623" authcode="444444"/>
```

4.2 为何 WEB SDK 调试页面无法进入，怎么用？

答：请核对 **site**、**ownerid**、**ctx**、**authcode** 是否填写正确。

举例：拿直播举例，点播也类似，只不过在对应的已发布点播（webcast）/课件管理(training)下找。

Webcast 产品：

调试页面举例

<http://demo.gensee.com/sdk/site/test/simple?ownerid=8a65b8abb3ff4fbb820e0ae3fa3d080c&ctx=webcast&authcode=333333>

主 题	接口开发常见问题FAQ-测试房间	
编 号	51181104	注：webcast产品则需增加ctx="webcast"
开始时间	2016-01-18 14:54	
结束时间		
持续时间		
是否公开	不公开	
是否要求登录	否	
Web端是否允许切换	否	
加入URL (web)	http://demo.gensee.com/webcast/site/entry/join-8a65b8abb3ff4fbb820e0ae3fa3d080c	复制链接
加入URL (Client)	http://demo.gensee.com/webcast/site/entry/live-8a65b8abb3ff4fbb820e0ae3fa3d080c	复制链接
组织者口令	111111	
嘉宾口令	222222	ownerid, 直播ID
普通参加者口令	333333	authcode即web端口令

Training 产品：

调试页面举例

<http://demo.gensee.com/sdk/site/test/simple?ownerid=CD9uw1X9Wz&ctx=training&authcode=444444>

实时课堂	实时课堂信息
号	场景模板：大课堂
160259	学生登录方式：所有
304279	加入URL： http://demo.gensee.com/training/site/r/29160259 复制链接
140306	学生端加入URL： http://demo.gensee.com/training/site/s/29160259 复制链接
343564	老师口令：111111
362215	助教口令：222222
357634	学生客户端口令：333333
712700	学生WEB端口令：444444
341461	所属课程：/
300745	是否公开：不公开
346617	是否要求用户登录：不需要
	Web端是否允许切换：不允许
	无延迟模式：不允许
	SDK ID： CD9uw1X9Wz ownerid="CD9uw1X9Wz", 课堂ID
	演讲人介绍：

4.3 为何自己写的WEB SDK 打开后一直提示 loading?

答：一般出现这种情况是由于 site 写错导致，系统找不到该站点导致，请按 [4.1](#) 章节中说明，正确填写相关参数。

Tester

Loading...



接收到消息:

[下午4:31:43] SDK Widget_GS_WIDGET_2_1453105903865) has loaded!

发送消息:

```
<div class="container">
<div class="row-fluid">
  <div class="span6 videoDiv">
    <gs:video-live site="http://demo.gensee.com/webcast" ctx="webcast" ownerid="3"
  </div>
  <div class="span6 docDiv">
```

demo.gensee.com

4.4 为何自己写的WEB SDK 打开后提示 not_found?

答：一般出现这种情况是由于 ctx、ownerid 写错导致，系统找不到该站点下的直播 ID 导致，请按 [4.1](#) 章节中说明，正确填写相关参数。

Tester

not_found



接收到消息:

[下午4:36:17] SDK Widget_GS_WIDGET_2_1453106177651) has loaded!
[下午4:36:17] SDK Widget_GS_WIDGET_4_1453106177651) has loaded!

发送消息:

```
<body>
  <div align="center">Tester</div><br>
  <div class="container">
    <div class="row-fluid">
      <div class="span6 videoDiv">
        <gs:video-live site="demo.gensee.com" ctx="webcast" ownerid="39880c94a99b4856bfa82008efa8a9aa" au
      </div>
      <div class="span6 docDiv">
        <gs:doc site="demo.gensee.com" ctx="webcast" ownerid="39880c94a99b4856bfa82008efa8a9aa" authcode=
      </div>
    </div>
  </div>
</body>
```

4.5 为何自己写的WEB SDK 打开后收不到任何监听事件消息?

答: 请按以下注意事项逐一检查。

1. 确认组织者/老师 客户端已经正确开启直播(小班课需要上讲台)。

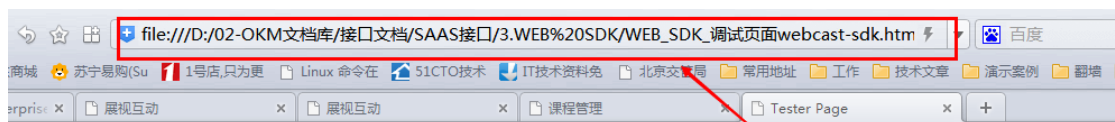
Webcast 产品:



Training 产品:

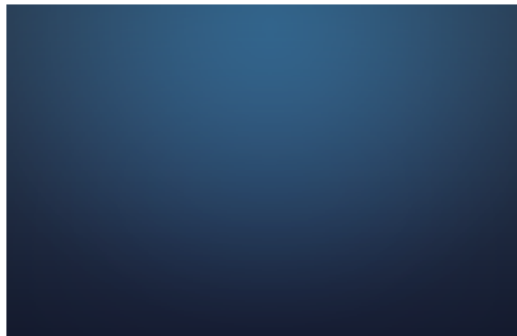


2. 确认已经打开组织者视频或老师视频, 并处于直播视频中。
3. 确认 WEB SDK 页面已经发布到 web server 上进行访问调试, 不能直接双击 html 方式打开。



Tester

错误!! WEB SDK不能用双击打开。



接收到消息:

[下午5:15:44] SDK Widget_GS_WIDGET_2_1453108544325) has loaded!
[下午5:15:44] SDK Widget_GS_WIDGET_1_1453108544323) has loaded!

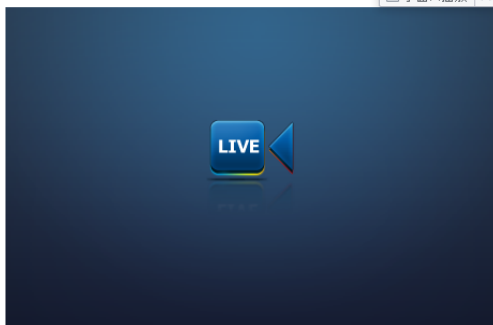
发送消息:

消息类型



Tester

正确, 必须在web server上以URL方式访问调试。



接收到消息:

[下午5:17:26] onStatus { "explain": "", "type": 2 }
[下午5:17:25] SDK Widget_GS_WIDGET_2_1453108645701) has loaded!
[下午5:17:25] SDK Widget_GS_WIDGET_1_1453108645699) has loaded!

发送消息:

消息类型

4.6 使用 WEB SDK 开发后的视频能全屏吗?

答: 如果想视频或文档的 flash 进行全屏, 则 video 需要增加 bar 条, 及文档全屏参数。

如果不使用 bar 控制条, 则 flash 只能通过调整 DIV 大小在浏览器内实现全屏效果, 无法脱离浏览器外整体全屏。



接收到消息:
[下午5:42:35] onStatus {"explain":"","type":2}

发送消息:

- **bar:** 指定是否显示控制栏, 值为 **true** 或者 **false**, 注: **ipad** 和 **android4.0** 及以上版本的移动播放不支持。
- **bgimg:** 设置背景图案, 图片 URL 地址。
- **completed:** video widget 创建之后调用的回调函数名称。若指定回调函数名称, 则在 widget 生成后, 会传递指定数据给该回调函数 (说明见 Part6 可扩展实时数据调用)
- **video:** 设置实时是否使用音频还是视频, 值为: **true** 或者 **false**, **true** 表示视频, **false** 表示音频, 默认为 **true**。

3.4 Doc Widget (可选)

网页 **html** 的 **body** 内, 任意地方可以插入如下代码:

```
<gs:doc id="docComponent" site="192.168.0.168" ownerid="f8625298d18042fbbba7a8"/>
```

属性说明:

- **id:** html 的 element id。 (可选)
- **site:** 站点域名 (必填)
- **ownerid:** 直播 ID。 (必填)
- **group:** 分组名称。作用同 Video Widget 的 Group 属性。 (可选)
- **ctx:** 服务选项。可选值为 **webcast** 或 **training**。无此属性说明默认使用 **webcast** 服务。 (可选)
- **fullscreen:** 是否支持 **flash** 全屏。可选值为 **true** 或者 **false**。默认值为 **false**。 (可选)
- **bgcolor:** 背景色。参考值: **#00ff00**。默认为空。 (可选)

4.7 否可以指定 video 的背景图? 设置完 bgimg 参数但是不生效?

答: 可以指定 video 播放前的背景图, 参数如下:

- `bar`: 指定是否显示控制栏, 值为 `true` 或者 `false`, 注: `ipad` 和 `android4.0` 及以上版本的移动播放不支持。
- `bgimg`: 设置背景图案, 图片 URL 地址。
- `completed`: `video widget` 创建之后调用的回调函数名称。若指定回调函数名称, 则在 `widget` 生成后, 会传递指定数据给该回调函数 (说明见 Part6 可扩展实时数据调用)
- `video`: 设置实时是否使用音频还是视频, 值为: `true` 或者 `false`, `true` 表示视频, `false` 表示音频, 默认情况 `true`。注意: 该值得设置只对移动端设备生效。

如果遇到指定图片 URL 地址后不生效, `video` 区域出现黑屏, 则因为 `flash` 的跨域问题, 解决办法: 图片所在服务器的根目录增加 `crossdomain.xml` 文件。

`crossdomain.xml`:

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="*" secure="false"/>
</cross-domain-policy>
```

PS: 关于 `flash` 的跨域策略文件描述如下 <http://www.2cto.com/Article/201108/100008.html>
官方解释如下:
http://help.adobe.com/zh_CN/as3/dev/WS5b3ccc516d4fbf351e63e3d118a9b90204-7c85.html#WS5b3ccc516d4fbf351e63e3d118a9b90204-7e08

4.8 当桌面共享时, 文档 (`doc`)、视频 (`video`) 位置大小互换 效果如何实现?

答: 需要根据 `onVideoConfig` 事件来进行判断, 当监听到桌面共享时, 文档和视频的 `div` 大小及位置进行互换处理。

4.4 onVideoConfig -- 视频配置更改通知

参数:

```
{
  "hasVideo": [true/false],
  "width": 320,
  "height": 240,
  "type": ["video"/"share"]
}
```

参数	说明
hasVideo	是否有视频数据, true 或者 false
width	流宽度
height	流高度
type	video – 当前 Video Widget 显示视频流; share start – Video Widget 显示共享流; share stop – Video Widget 停止显示共享流

4.9 如何实现对啊网播放器那种在聊天前增加对应身份图标?

答: 对啊网效果 <http://www.duia.com/videoLive/public/govod/multiGovod?sku=1>

实现原理: 因用户先登录客户网站后再进入直播页面, 故可以判断学员是否为付费 VIP 学员, 当为付费学员, 可以在初始化昵称时, 在真实昵称前加入 vip1_****, 即 `uname=vip1_****`, 当播放器监听到公聊信息并且 `uname` 有预设关键字时, 解析显示一个图标, 并把真实昵称、聊天信息正常显示出来。最终达成如下截图效果。(手机图标处理逻辑同理)



4.10 公聊如何实现屏蔽关键字、控制发言间隔?

答：在使用 WEB SDK 的 submitChat-- 提交公聊信息 事件前，先用 js 进行前端关键字及发言间隔判断，当不符合要求时，不继续提交。

对啊网关键字参照如下：

```
83     border-bottom: 1px solid #eee;
84   }
85   </style>
86   <script>
87     var liveFlag = "public";
88     /*屏蔽字*/
89     var keywords = "中共,中公,加我QQ,加我一下,我们一起买一个,合买一个,加我群号,谁转到发我一下,没用,打水漂,大钟,棒子,cso,LOL,猪,jb,正事,吊,屌,闲话,不回,外挂,微信,weixin,穆穆,舒舒,作
90     闹,购物,罗刹,300,讨论群,690,三百万,五百万,讲重点,废话,不客气,扣扣,加群,鄙视,#$%&'()*+,-./:;<=>@,优惠,qd,别招,垃圾,白给,淘宝,无需再用,加Q,不好,奇葩,老,二逼,566,233,真棒,华图,厚颜,苏比,个比,铁子,屌
91     王八,玩意,尼玛,尼玛,宣传手段,营销,不是免费吗,屁,屁,tuo,滚,听腻,熬笔,洗脑,泡,真,做爱,撸,ab,shabi,抱,托,歪歪,胡诌,广告,妹的,Y,马脚,吹牛,仁和,恒企,中华会,投诉,态度不好,东奥,huluo,菊花,欠
92     差,滚不滚,神经,麻痹,fuck,中华网,骗子,无忧考吧,推销,群,q,分期,老逼,喜感,自己学,通过军,自学,骗人,贱人,碰瓷,恶心,搞笑,找不到工作,卖身,坐台,股票,君,羊,吹捧,傻逼,傻逼老师,拉生源,几
93     把,1450,550,1250,不讲,贵,坑,没钱,不贵钱,卡,宣传,糊弄,快讲,赶紧讲课,聊天,轻松过关,梦想成真,中华";
94     var validKeywords = null;
95     if (keywords != null && keywords != "") {
96       validKeywords = keywords.split(",");
97     }
98     var path = "";
99     var zsoDomain = "http://aso.duia.com";
100     var uuid = "";
101     var sku = "1";
102     var userId = "";
103     var senderId = "";
104     var liveId = "";
105     var isUpdateSeekPoint = true;
106     var leftPptjd = 0;
107     var previousMsgLongtime = 0;
108     var webCastId = "937b457b71724d5b89f2f7c11b807e56";
109     var endLong = "1452851361000";
110     var startLong = 1452848620000;
111     var categoryId = 1;
112     var currentUrlIndex = 1;
113     var userName = "";
114     var lectureId = 0;
115     var classId = 0;
116     var videoIsOpen = true;
117     var isUocMain = true;
118     var videoCurrentPointNew = 0;
119     var ownerid = "";
120     var userVip = 2;
121     var vipType = 0;
```

4.11 WEB SDK 聊天 submitChat 如何实现发送表情？

5.7 submitChat -- 提交公聊信息

注意：content 与 richtext 至少提交其中之一，若同时提交 content 和 richtext 时，以 richtext 为准。

如果两个公聊信息的提交有设置间隔时间，那么在间隔内提交公聊信息不会成功的，同时如果提交公聊信息有设置回调函数那么会调用回调函数。

（提议：以 input.textarea 作为聊天信息输入的，使用 content 属性提交信息，以 div[contenteditable] 作为聊天信息输入的，使用 richtext 属性提交信息）

参数：

```
{
  "content": "hello",
  "richtext": ".....",
  "security": "[default|high]"
}
```

参数	说明
content	聊天内容
richtext	聊天内容允许带有表情图标。表情图标参考 6.3
security	设置聊天内容的安全等级权限，default 是原有的较低的安全等级模式，high 为新增较高安全等级模式，缺省该属性默认为 default 安全等级

7.3 关于聊天功能的表情图标

<http://static.gensee.com/webcast/static/emotion/icon.json>

在聊天功能中，提供富文本内容支持。

其中表情图标的 URL 见以上 json 格式中数据。

```
"prefix": "https://static.gensee.com/webcast",
"list": [
  {
    "icon": "emotion\\chat.gif.png", "url": "/static/emotion/chat.gif.png", "zh_cn": "【礼物】", "en": "[gift]", "type": 3, "zh_tw": "【禮物】", "ja": "【ギフト】",
    "icon": "emotion\\emotion_angrily.gif", "url": "/static/emotion/emotion_angrily.gif", "zh_cn": "【愤怒】", "en": "[angry]", "type": 1, "zh_tw": "【憤怒】", "ja": "【怒り】",
    "icon": "emotion\\emotion_his.gif", "url": "/static/emotion/emotion_his.gif", "zh_cn": "【鄙视】", "en": "[despise]", "type": 1, "zh_tw": "【鄙視】", "ja": "【輕蔑】",
    "icon": "emotion\\emotion_cry.gif", "url": "/static/emotion/emotion_cry.gif", "zh_cn": "【伤心】", "en": "[sad]", "type": 1, "zh_tw": "【傷心】", "ja": "【悲しい】",
    "icon": "emotion\\emotion_goodbye.gif", "url": "/static/emotion/emotion_goodbye.gif", "zh_cn": "【再见】", "en": "[bye]", "type": 1, "zh_tw": "【再見】", "ja": "【さようなら】",
    "icon": "emotion\\emotion_laugh.gif", "url": "/static/emotion/emotion_laugh.gif", "zh_cn": "【高兴】", "en": "[happy]", "type": 1, "zh_tw": "【高興】", "ja": "【楽しい】",
    "icon": "emotion\\emotion_is.gif", "url": "/static/emotion/emotion_is.gif", "zh_cn": "【流汗】", "en": "[sweat]", "type": 1, "zh_tw": "【流汗】", "ja": "【汗】",
    "icon": "emotion\\emotion_mod.gif", "url": "/static/emotion/emotion_mod.gif", "zh_cn": "【无聊】", "en": "[bored]", "type": 1, "zh_tw": "【無聊】", "ja": "【退屈した】",
    "icon": "emotion\\emotion_question.gif", "url": "/static/emotion/emotion_question.gif", "zh_cn": "【疑问】", "en": "[doubt]", "type": 1, "zh_tw": "【疑問】", "ja": "【疑問】",
    "icon": "emotion\\emotion_smile.gif", "url": "/static/emotion/emotion_smile.gif", "zh_cn": "【你好】", "en": "[hello]", "type": 1, "zh_tw": "【你好】", "ja": "【こんにちは】",
    "icon": "emotion\\emotion_springs.gif", "url": "/static/emotion/emotion_springs.gif", "zh_cn": "【反对】", "en": "[oppose]", "type": 1, "zh_tw": "【反對】", "ja": "【反対】",
    "icon": "emotion\\feedback_agreed.png", "url": "/static/emotion/feedback_agreed.png", "zh_cn": "【赞同】", "en": "[agree]", "type": 2, "zh_tw": "【贊同】", "ja": "【承認】",
    "icon": "emotion\\feedback_applaud.png", "url": "/static/emotion/feedback_applaud.png", "zh_cn": "【鼓掌】", "en": "[clap]", "type": 2, "zh_tw": "【鼓掌】", "ja": "【エール】",
    "icon": "emotion\\feedback_quickly.png", "url": "/static/emotion/feedback_quickly.png", "zh_cn": "【太快了】", "en": "[too fast]", "type": 2, "zh_tw": "【太快了】", "ja": "【速い】",
    "icon": "emotion\\feedback_slowly.png", "url": "/static/emotion/feedback_slowly.png", "zh_cn": "【太慢了】", "en": "[too slow]", "type": 2, "zh_tw": "【太慢了】", "ja": "【遅い】",
    "icon": "emotion\\feedback_think.png", "url": "/static/emotion/feedback_think.png", "zh_cn": "【值得思考】", "en": "[be consideration]", "type": 2, "zh_tw": "【值得思考】", "ja": "【考え】",
    "icon": "emotion\\rose_down.png", "url": "/static/emotion/rose_down.png", "zh_cn": "【凋谢】", "en": "[wither]", "type": 3, "zh_tw": "【凋謝】", "ja": "【枯れて落ちる】",
    "icon": "emotion\\rose_up.png", "url": "/static/emotion/rose_up.png", "zh_cn": "【绽放】", "en": "[flower]", "type": 3, "zh_tw": "【綻放】", "ja": "【ユズリハ】"
  }
]
```

URL 规则：prefix + url

Type 说明：1- 心情, 2- 反馈, 3- 礼物

答：使用聊天发送表情需要用到 WEB SDK 文档中以上两部分及 JS 配合使用。以【礼物】这个表情为例。

1. 使用 submitChat 提交的数据如下：

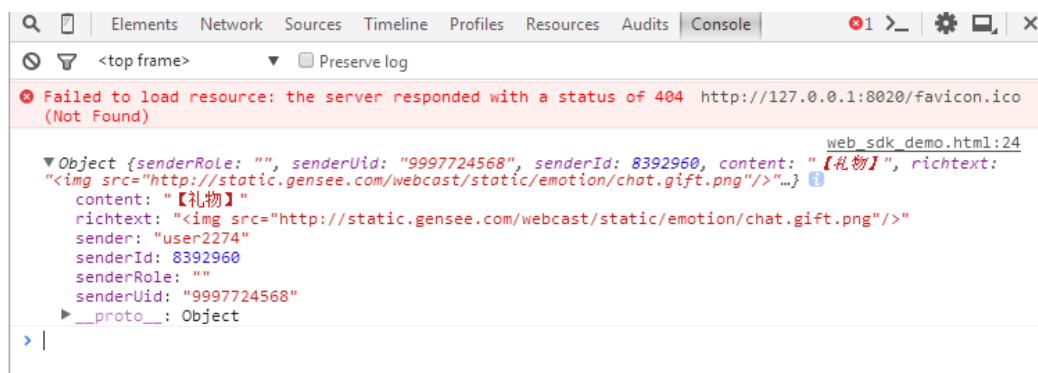
(提议: 以 input、textarea 作为聊天信息输入的, 使用 content 属性提交信息, 以 div[contenteditable] 作为聊天信息输入的, 使用 richText 属性提交信息)

```
c/emotion/chat.gif.png", "zh_CN": "【礼物】", "en": "[gift]", "type": 3, "zh_TW": "【禮物】", "ja": "【ギフト】", "st": "$^lw^"
"/static/emotion/emotion.angerly.gif", "zh_CN": "【愤怒】", "en": "[angry]", "type": 1, "zh_TW": "【憤怒】", "ja": "【怒り】", "st": "$^fn^"
tic/emotion/emotion.bs.gif", "zh_CN": "【鄙视】", "en": "[despise]", "type": 1, "zh_TW": "【鄙視】", "ja": "【輕蔑】", "st": "$^gm^"
atic/emotion/emotion.cry.gif", "zh_CN": "【伤心】", "en": "[sad]", "type": 1, "zh_TW": "【傷心】", "ja": "【悲しい】", "st": "$^sx^"
"/static/emotion/emotion.goodbye.gif", "zh_CN": "【再见】", "en": "[bye]", "type": 1, "zh_TW": "【再見】", "ja": "【さようなら】", "st": "$^zj^"
static/emotion/emotion.laugh.gif", "zh_CN": "【高兴】", "en": "[happy]", "type": 1, "zh_TW": "【高興】", "ja": "【ハッピー】", "st": "$^gx^"
tic/emotion/emotion.lh.gif", "zh_CN": "【流汗】", "en": "[sweat]", "type": 1, "zh_TW": "【流汗】", "ja": "【汗】", "st": "$^lh^"
atic/emotion/emotion.nod.gif", "zh_CN": "【无聊】", "en": "[bored]", "type": 1, "zh_TW": "【無聊】", "ja": "【退屈した】", "st": "$^wl^"

{
  "content": "hello",
  "richText": "$^lw^", //当 security=high 时采用高安全等级, 发送文本为 icon.json 的含义符字符串部分。
  "security": "high"
}
```

2. 在前台把 content 对应的文本或 richText 的 img src 对应解析图标展示在前台即可。

<http://static.gensee.com/webcast/static/emotion/chat.gif.png> 



提供表情转换 function 使用方式如下

使用 channel 你获取的富文本做 emotionNormalize 处理即可 还有使用高安全性 security:"high" 参数

例子如下

```
channel.send("submitChatTo",
{"receiverId": uid, "receiver": uname, "richText": emotionNormalize(richText), security: "high"});
```

```
function emotionNormalize(richText, replace, local){
    richText = richText.replace(/<img.*?emotion\/(.+?)\.gif\/>/gi, function
(match, capture) {
        var note = "";
        var code = "";
        switch(capture){
            case "feedback.quickly":
```

```
code = "$^tk^";  
// note = i18n("emotion.feedback.quickly");  
break;  
case "feedback.slowly":  
code = "$^tm^";  
// note = i18n("emotion.feedback.slowly");  
break;  
case "feedback.agreed":  
code = "$^zt^";  
// note = i18n("emotion.feedback.agreed");  
break;  
case "feedback.against":  
code = "$^fd^";  
// note = i18n("emotion.feedback.against");  
break;  
case "feedback.applaud":  
code = "$^gz^";  
// note = i18n("emotion.feedback.applaud");  
break;  
case "feedback.think":  
code = "$^sk^";  
// note = i18n("emotion.feedback.think");  
break;  
case "emotion.smile":  
code = "$^nh^";  
break;  
case "emotion.goodbye":  
code = "$^zj^";  
break;  
case "emotion.laugh":  
code = "$^gx^";  
break;  
case "emotion.cry":  
code = "$^sx^";  
break;  
case "emotion.angerly":  
code = "$^fn^";  
break;  
case "emotion.nod":  
code = "$^wl^";  
break;  
case "emotion.lh":  
code = "$^lh^";  
break;
```

```

        case "emotion.question":
            code = "$^yw^";
            break;
        case "emotion.bs":
            code = "$^qm^";
            break;
        case "rose.up":
            code = "$^xh^";
            break;
        case "rose.down":
            code = "$^dx^";
            break;
        case "chat.gift":
            code = "$^lw^";
            break;
        default:;
    }

    if((typeof local == "undefined" || !local))match = code;
    if(replace){
        return note;
    }else{
        if(local){
            return match + note;
        }else{
            return match;
        }
    }
    });
    return richText;
}

```

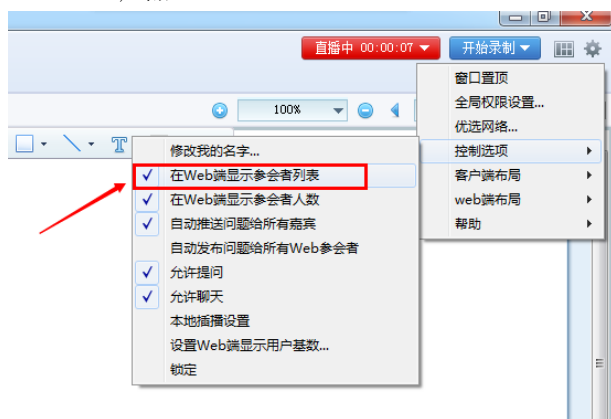

4.12 为何 onUserList-- 用户列表事件收不到消息?

答：需要组织者或老师在客户端中打开【在 WEB 端显示参会者列表】并开启直播后，web 端才能收到消息。

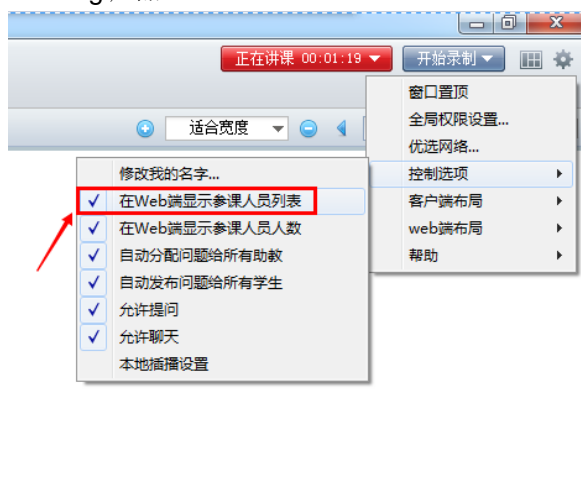
并且 onUserList 事件只会在播放器成功进入直播后收到一次消息（除非断开重连后会再次收到），直播中的用户变动，需要使用 onUserJoin -- 用户加入，onUserLeave -- 用户离开，做相应增减操作，并显示的 UI 前台界面上。

PS：当在线人数超过 200 人时，onUserList 最多只会推送 200 条数据，会优先推送组织者和嘉宾的用户信息。

Webcast 产品：



Training 产品：



如有需要站点后台可以设置每次进入直播房间自动打开。



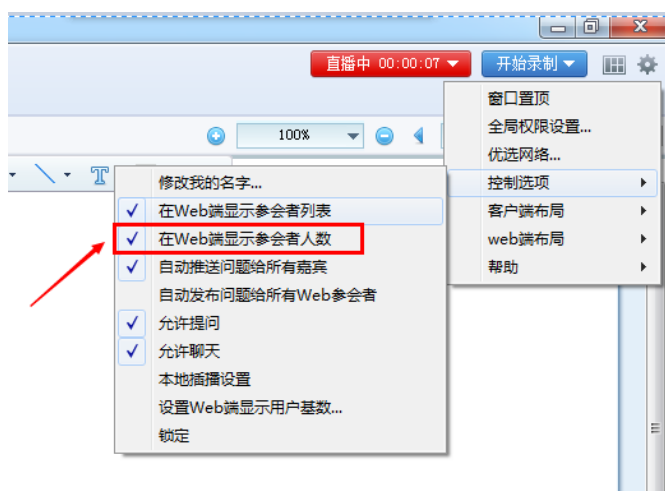
Training 产品:



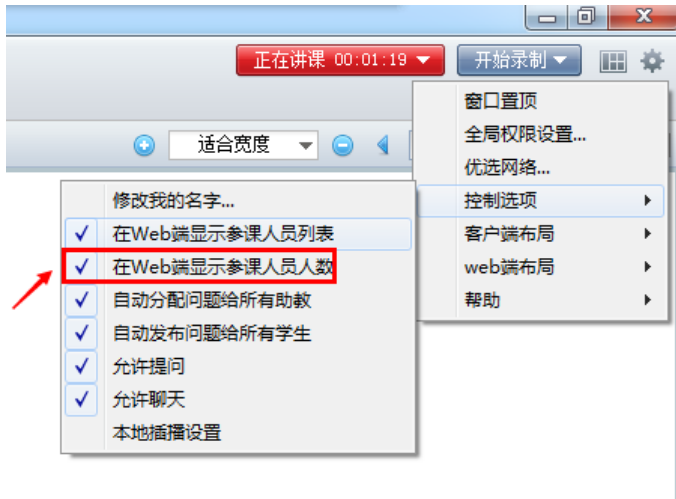
4.13 为何 onUserOnline-- 用户在线人数事件收不到消息?

答: 需要组织者或老师在客户端中打开【在 WEB 端显示参会者人数】并开启直播后, web 端才能收到消息。

Webcast 产品:



Training 产品:



如有需要站点后台可以设置每次进入直播房间自动打开。



Training 产品:



4.14 为何使用 submitChatTo 提交私聊信息发送了私聊，但 onPriChat 收到私聊消息事件收不到任何消息？

答：一般是由于在使用 submitChatTo 提交私聊信息时，误把接收者聊天 ID 理解为接收者 UID 导致。

参数	说明
content	聊天内容
richtext	聊天内容允许带有表情图标。表情图标参考 6.3
receiver	接收者名称，可选参数
receiverId	接收者聊天 ID，必填参数
security	设置聊天内容的安全等级权限，default 是原有的较低的安全等级模式，high 为新增较高安全等级模式。缺省该属性默认为 default 安全等级

4.15 onUserList -- 用户列表

参数：

```
{
  "list": [
    { "id": 1, "name": "Tom", "chatid": 1, "role": "1,2" },
    { "id": 2, "name": "Jason", "chatid": 2, "role": "4" }
  ]
}
```

参数	说明
list	用户列表
id	用户 ID
name	用户名称
chatid	仅用于聊天列表的聊天 ID
role	用户角色以组合的形式出现，可能是单个角色，可能是多个角色，多个角色用逗号分隔，角色对应含义如下：

4.15 为何点播 SDK 中的 onQAList、onChatHistory 收不到任何消息？

答：因聊天和 QA（问答）数据量巨大，故目前点播播放时并非一次性全部获取这些信息，而是通过对应提交事件触发后，分页方式来进行获取。请参照以下文档：

5.10 submitQAList

事件说明：通过接口申请获取生成该点播的直播 Q&A 历史记录。

这是个异步申请接口，Q&A 数据将通过 onQAList 事件返回给第三方。因此，请在调用此方法前，预先监听 onQAList 事件。（见 [4.5](#)）

5.11 submitChatHistory

事件说明：通过接口申请获取生成该点播的直播公共聊天历史记录。

这是个异步申请接口，公共聊天历史数据将通过 onChatHistory 事件返回给第三方。因此，请在调用此方法前，预先监听 onChatHistory 事件（见 4.20）。

4.16 如何实现未登陆用户试看一定时间自动弹出购买和注册窗口（广告）？

答：因 WEB SDK 均使用前台开发技术实现，故只需要在 JS 中写计时器，播放到指定时间后自动触发播放停止事件并前台 UI 做个弹框覆盖视频区域即可。

4.17 WEB SDK 如何实现暖场的效果？

答：暖场实际为开始直播前循环播放的点播。故使用 WEB SDK 则需要按以下几个步骤来实现暖场效果。

1. 用直播 WEB SDK 写个直播播放器页面。
2. 用点播 WEB SDK 写个点播播放器页面。
3. 使用实时回调接口，检测直播房间是否处于开启直播状态，如果未开始直播则用户进入观看页面调用点播播放器播放点播内容，若直播已经开始，则用户进入观看页面调用直播播放器播放内容。
4. 当用户进入观看界面时，直播尚未开始，用户进入点播播放器，观看一段时间后直播开始，则需要点播播放页面收到直播开始消息（客户通过网页技术实现通信），自行刷新处理，进入正常直播播放器界面。

4.18 聊天是否支持自定义第三方表情？

答：目前 gensee web sdk 还不支持第三方表情，正在紧张开发中。

暂时可以通过其他办法间接解决，解决思路类似常规发送表情，只是发送的自己定义的含义符，播放器前台收到这种自定义的含义符时，在前台解析出想要解析的对应图标即可，相当于只是把 gensee server 当做字符串的发布通道而已，表情全部由前台进行解析显示。

4.19 使用了 WEB SDK 开发了自定义播放器如何实现同一链接不同设备观看？（达到类似 Gensee WEB 端在 PC 和移动端打开效果不同）

答：使用 WEB SDK 开发出一套 PC 浏览器的样式，再开发一套适配移动端的样式。通过用户浏览器 UA 来进行判断对方属于 PC 还是移动端，最终加载不同的样式表，达到同一链接，不同设备访问自动适配的效果。

5 实时回调接口开发调试常见问题

5.1 实时回调接口在哪里设置? 怎么填写?

答: 实时 server 会自动发送 get 请求到预设在后台的 URL 地址上。客户方服务器会收到请求如下格式:

`http://*****.***.com/gensee_API/Callback.aspx?ClassNo=3tVIObJeCD&Operator=9911237037&Action=101&Affected=9911237037&totalusenum=7`

PS: ClassNo 等同于 4.1 章节的直播 ID (即 webcast ID, SDK ID)。

Webcast 产品: 例如设置地址: `http://*****.***.com/gensee_API/Callback.aspx`



Training 产品: 例如设置地址: `http://*****.***.com/gensee_API/Callback.aspx`



5.2 实时回调接口已经全部开发调试完成，为何有时候会突然收不到推送的回调消息？

答：因回调接口类似UDP的发送方式，并不一定保证接口URL会收到消息，但只要房间状态发生变化实时server均会进行正常推送。

出现未收到的原因，一般多为server间的网络波动，致使实时server发送的get请求客户方服务器并未收到导致。

6 移动端及其他常见问题

6.1 使用 webview 在 IOS APP 上嵌入直播链接，视频自动全屏浮出如何解决？

答：众所周知，iphone 系统网页看视频，视频会全屏浮出，占用整个屏幕。这是它的默认行为，不受第三方 web 开发人员影响。但是对于 app 应用，由于其有能力将网页内嵌，从而可以实现视频不浮出，不占用全屏。对这块技术有兴趣了解的，请看下面的技术链接。

一是解决方法：

<http://down.51cto.com/data/619584>

二是苹果的官方文档：

https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIWebView_Class/index.html#//apple_ref/occ/instp/UIWebView/allowsInlineMediaPlayback

注意苹果写的：

Managing Media Playback

`allowsInlineMediaPlayback` Property

A Boolean value that determines whether HTML5 videos play inline or use the native full-screen controller.

Declaration

SWIFT

```
var allowsInlineMediaPlayback: Bool
```

OBJECTIVE-C

```
@property(nonatomic) BOOL allowsInlineMediaPlayback
```

Discussion

The default value on iPhone is NO.

In order for video to play inline, not only does this property need to be set on the view, but the `video` element in the HTML document must also include the `webkit-playsinline` attribute.

Import Statement

OBJECTIVE-C

```
@import UIKit;
```

SWIFT

```
import UIKit
```

Availability

Available in iOS 4.0 and later.

因此，对于 app，有些支持该功能，有些不支持。这不由网页决定，由 app 自身决定。

6.2 为何使用 `iframe` 嵌入 URL 后，移动端打开有界面适配的问题？如何解决？



答：

URL_API_iframe嵌入URL适配移动端WEB demo（可用于微信公共号）.zip

双击打开 demo。

6.3 移动端如果使用 WEB SDK 开发自定义播放器，如何调试？

答：因我们采用的判断浏览器 UA 的方式来推送不同流（PC 推 flash，移动端推 H5），如要使用 WEB SDK 开发移动端 web 播放器也同样可是使用 [4.2](#) 的调试页面方式，在手机打开该调试页面 URL 进行调试。

6.4 视频在手机浏览器上观看为何会自动全屏浮出，导致无法看到文档、聊天？

答：iphone 系统网页看视频，视频会全屏浮出，浮出占用整个屏幕。这是 IOS 的默认行为，不受第三方 web 开发人员影响。

目前通过优化，在微信中的特殊处理，已经实现在微信中观看直播，iPhone 不浮出的处理。但 safari 是苹果官方浏览器，是无论如何也无法实现视频不浮出的。

部分安卓设备视频也会自动浮出全屏，原因是某些国产浏览器或者 Android 定制系统自动处理导致。

我们的播放器产品是基于上层的应用，无法对下层的实现方式做处理。

目前我们测试只有以下情况，不会浮出

- 1、安卓设备 使用 CHROME 浏览器
- 2、Iphone 使用微信打开（少量机器仍然会全屏，原因不明）
- 3、微信 6.3.5 以前版本

建议：

- 1、使用支持的设备与浏览器观看。比如在微信右上角选择"在浏览器中打开"
- 2、对于其他情况，建议可以使用部分主流浏览器的小窗功能，能解决文档与视频同时观看的问题
- 3、我们直播的播放器编辑页面中，可以设置为音频文档模式，取消视频的功能

这个问题是所有 web 移动端视频都会碰到的问题。大家可以尝试使用手机浏览器打开优酷、土豆、乐视等视频网站，一样会有这个问题，上层应用无法控制。

6.5 使用 WEB SDK 开发播放器在移动端 H5 界面上视频区域 div 隐藏后就会没有声音。

答：目前移动端对于视频流媒体有个自己的处理方式，即当发现原本播放的 video 内容被隐藏后为了节省手机流量，则会自动断开流媒体播放请求。

故：最好不要用{display: none}或者{width:0;height:0;}的方式，因为这样视频元素会处于未激活的状态。会被移动端自动关掉。最好是采用移出显示界面外的方式进行处理或设置一个小像素放置到界面边缘，让用户察觉不到。

参考链接：<http://www.xuanfengge.com/html5-video-play.html>

这显然不是一个好的视觉体验，那么通常的做法是制作一个模拟的视频播放视图，比如一个封面加一个播放按钮。

而真实的<video>视频元素要隐藏起来，如何隐藏呢？最好不要用 {display: none} 或者 {width:0;height:0;} 的方式，因为这样视频元素会处于未激活的状态，给后续的处理带来麻烦。最佳的方式是将视频设置成1×1像素大小，放在视觉边缘的位置。

```
1 <!--iOS-->
2 <video webkit-playsinline width="1" height="1" class="vplayinside notaplink" x-webkit-
  airplay controls loop="loop" src="<%=src%>"></video>
3
4 <!--Android-->
5 <video width="1" height="1" controls loop="loop" src="<%=src%>"></video>
```