# 阿里云 DataV数据可视化

# 管理可视化应用

文档版本: 20191223

为了无法计算的价值 | [-] 阿里云

### 法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读 或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

- 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用于自身的合法 合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格遵守保密义务;未经阿里云 事先书面同意,您不得向任何第三方披露本手册内容或提供给任何第三方使用。
- 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分 或全部,不得以任何方式或途径进行传播和宣传。
- 3. 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有任何通知或者 提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时发布更新后的用户文档。您 应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
- 4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的"现状"、"有缺陷"和"当前功能"的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
- 5. 阿里云文档中所有内容,包括但不限于图片、架构设计、页面布局、文字描述,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。 非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、 散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人 不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独 为或以组合形式包含"阿里云"、"Aliyun"、"万网"等阿里云和/或其关联公司品牌,上述 品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、 标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
- 6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

# 通用约定

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

### 目录

法律声明	I
通用约定	I
1 使用模板创建可视化应用	1
2 使用马良创建可视化应用	3
3 编辑可视化应用	
4 重命名可视化应用	
5 拷贝可视化应用	10
6 复制可视化应用	
7 删除可视化应用	
8 预览可视化应用	14
9 发布可视化应用	15
10 DataV分享页Token参数签名校验	24

# 1 使用模板创建可视化应用

DataV支持使用模板和马良两种方式创建可视化应用,本文档介绍使用模板创建可视化应用的方法。

#### 背景信息

- <u>!</u>注意:
- ·基础版的DataV最多可以创建5个项目,企业版最多可以创建20个项目,请根据您的需求合理 选择DataV版本。
- · 使用马良创建可视化应用的方法请参见使用马良创建可视化应用。

#### 操作步骤

- 1. 登录DataV控制台。
- 2. 单击我的可视化 > 新建可视化。

在我的可视化页面,可以查看所创建的所有大屏项目及还可以创建的大屏数量。

<b>章</b> 我的可视化	⑦ 我的数据	<b>久</b> 我的组件	G:教程
我的分组	+   全部大师	异 24个/还可创建516个 ⑦ 管	理项目配额
全部大屏	24		
未分组	24	新建可视化 使用 [ 马良 ] 创建	
	(Ai Vi	sLab	ዸ 11 ● 未发布

3. 在左侧的模板列表中,选择一个模版,单击创建。

取消创建		
	选择模板	
空白 > 尺寸目症		
認知医疗大研 (See) 出か 15:5 13:20x1080px		
武政者板 第 三方支持統 世界 16-3 1920×1090px		
全球贸易2 ↔ 比例 15:3 1920x1080px		
「「「「「」」 「「」 「「」 「「」 」 「」 」 「」 「 」 「」 」 「」 」 「」 」 「」 」 「」 」 」 」 」 「 」 」 」 」 」		
################################		
● 全球贸易 比例 15:9 1920x1080px	在空白喜本上尽情感最多的但是明	

- 在创建数据大屏对话框中,输入大屏项目名称,单击创建。
   创建成功后页面会跳转到大屏编辑器页面。
- 5. 单击左上角的 图标,返回我的可视化页面,可查看创建成功的大屏项目。

# 2 使用马良创建可视化应用

DataV支持使用模板和马良两种方式创建可视化应用,本文档介绍使用马良创建可视化应用的方法。

#### 背景信息

!) 注意:

使用模板创建可视化应用的方法请参见使用模板创建可视化应用。

#### 操作步骤

- 1. 登录DataV控制台。
- 2. 单击我的可视化 > 使用马良创建。

😂 我的可视化	3	我的数据	<b>冷</b> 我的组件	
我的分组	+	│ 全部大屏	11个/还可创建19个 꼙	管理
全部大屏	11			
未分组	10		新建可抑化	
test	1			
		(Ai VisL	使用「马良」创建 ab	

#### 3. 在马良页面上方,单击选择创建大屏的模式。

草图绘制	草图上传	
创建大屏模式	说明	
草图绘制	您可以在草图绘制页面的空白画布区 一键生成马良大屏。	〔域内,手动绘制所需的组件,并
	草图绘制页面包括如下几个功能:	
	・ 画笔模式: 単击画布右侧的	图标,单击鼠标并移动,即可在
	空白画布中自定义绘制大屏组件。	
	・橡皮模式: 単击画布右侧的	图标,单击鼠标单击并移动,即
	可擦除绘制过程中错误的图像。	
	・清空画布:単击画布右侧的	图标,即可一键清空画布中所有
	绘制的图像。	
	<b>▲</b> 警告:	
	一键清除后无法恢复内容,请谨	慎操作。
	· 生成大屏: 单击画布右下角的	图标,系统会新开页面生成名
	为马良-智能识图的大屏。	
	您可以在浏览器中直接切换至新的	的大屏页面,或者在马良页面弹出
	的创建成功对话框中,单击确定, 大展	跳转到新的名为马良-智能识图的
	新创建的名称为马良-智能识图的 似的样式,您可以根据生成的大月	大屏中会生成与绘制的草图组件类 屏继续进行其他编辑操作。

创建大屏模式	说明
草图上传	您可以在草图上传页面,上传手绘图,一键生成马良大屏。
	草图上传页面包括如下几个功能:
	・切换手绘图为设计图:在上传&预览区域上方,单击请上传手绘
	图,上传设计图右侧的点此切换,切换到设计图上传页面。
	・ 上传手绘图:系统支持两种方式上传手绘图。
	- 在上传&预览区域中,单击虚线圆内的+号,可以选择本地的手
	绘图,上传到上传&预览区域中。
	- 复制已有手绘图,然后单击上传&预览虚线框,使用CtrlV快捷
	键,将该手绘图上传到上传&预览区域中。
	・ 重新上传:单击上传&预览区域右上角的 〇〇 図标,可重新上传手
	绘图。
	・ 生成大屏:単击上传&预览区域右下角的 긎 图标,系统会新开页
	面生成名为马良-智能识图的大屏。
	您可以在浏览器中直接切换至新的大屏页面,或者在马良页面弹出
	的创建成功对话框中,单击确定,跳转到新的名为马良-智能识图的
	大屏。
	新创建的名称为马良-智能识图的大屏中会生成与绘制的草图组件类似的样式,您可以根据生成的大屏继续进行其他编辑操作。

创建大屏模式	说明
设计图上传	您可以在设计图上传页面,上传设计图,一键生成马良大屏。
	设计图上传页面为默认页面,包括如下几个功能:
	<ul> <li>· 切换设计图为手绘图:在上传&amp;预览区域上方,单击请上传设计</li> <li>稿,上传手绘稿右侧的点此切换,切换到草图上传页面。</li> <li>· 上传设计图:系统支持两种方式上传设计图。</li> </ul>
	<ul> <li>在上传&amp;预览区域中,单击虚线圆内的+号,可以选择本地的设</li> <li>计图,上传到上传&amp;预览区域中。</li> </ul>
	<ul> <li>- 复制已有设计图,然后单击上传&amp;预览虚线框,使用CtrlV快捷</li> <li>键,将该设计图上传到上传&amp;预览区域中。</li> </ul>
	・选择参考素材:在设计图上传页面左侧的参考素材中,单击选择其
	中一个素材,即可在右侧的上传&预览区域进行查看,确定使用该
	素材后,可选择生成以该素材为参考的马良大屏。
	・重新上传:单击上传&预览区域右上角的 [1] 「「「」」「「」」「「」」「」」「」」「」」「」」「」」「」」「」」「」」「」
	计图。
	· 生成大屏: 单击上传&预览区域右下角的 🕞 图标, 系统会新开页
	面生成名为马良-智能识图的大屏。
	您可以在浏览器中直接切换至新的大屏页面,或者在马良页面弹出
	的创建成功对话框中,单击确定,跳转到新的名为马良-智能识图的 大屏。
	新创建的名称为马良-智能识图的大屏中会生成与绘制的草图组件类 似的样式,您可以根据生成的大屏继续进行其他编辑操作。

4. 选择大屏的模式并设计好大屏后,单击 网际,即可生成一个名为马良-智能识图的可视化大





图标,返回我的可视化页面,可查看已经创建成功的马良大屏项目。 5. 单击左上角的



# 3编辑可视化应用

本文档介绍编辑大屏项目的方法,帮助您快速修改大屏项目的内容和配置。

操作步骤

- 1. 登录DataV控制台。
- 2. 单击我的可视化。
- 3. 在大屏管理页面,移动鼠标至需要编辑的大屏项目上,单击编辑。



 在大屏编辑页面,修改组件的布局和配置,或者添加或删除组件,完成大屏的编辑。 编辑完成后,可预览或发布大屏,查看效果。

# 4 重命名可视化应用

本文档介绍如何对一个大屏项目进行重命名,帮助您快速修改大屏项目的名称,便于搜索与管理。

操作步骤

- 1. 登录DataV控制台。
- 2. 单击我的可视化。
- 3. 在大屏管理页面,选择一个大屏,移动鼠标至该大屏项目的标题上。

	2425	000	⊒ ⊒ 4
	编辑		
÷		Ō	Ē
datav_test2	2 dada		
⊿ datav_te	est2		● 已发布

4. 单击大屏名称,进入编辑状态,输入新的大屏名称。



5. 单击空白处或按Enter键,即可完成大屏名称的修改。

大屏名称修改成功后,您就可以完成使用名称搜索大屏,按名称对大屏进行排序等操作。

# 5 拷贝可视化应用

本文档介绍如何将您的大屏项目拷贝给其他用户,帮助您快速地将大屏项目分享给他人,实现与他 人合作共同完成大屏的开发。

🛕 警告:

- ・ 拷贝过去的项目包含完整的数据配置,为避免数据泄露,在拷贝前请仔细核对用户识别码。
- ・拷屏功能只适用于企业版及以上版本。

操作步骤

- 1. 登录DataV控制台。
- 2. 单击我的可视化。
- 3. 在大屏管理页面,移动鼠标至需要拷贝的大屏项目上,单击拷屏图标。



4. 在拷贝项目给他人对话框中,输入对方的用户识别码(区分大小写)。

拷贝项目给他人	
用户识别码	
输入对方的用户识别码,区分大小写	
② 輸入对方的"用户识别码"(用户识别码在首页右上角用户信息 中查看)可直将该项目复制到对方的可视化项目中,拷贝过去的项目包含完整数据配置,为避免数据泄露,请仔细核对用户识别码。(该功能仅企业版及以上用户可用)	息
用户识别码时间	
拷	Q

获取用户识别码:在我的可视化页面,将鼠标移至用户名称上,即可查看并点击复制用户识别 码。



5. 单击拷贝,拷贝当前项目给其他用户。

拷屏完成后,您可以在拷贝项目给他人对话框中的拷屏历史中,查看对方的用户识别码和拷屏时 间,帮助您及时了解拷屏动态。

## 6复制可视化应用

本文档介绍如何复制大屏项目。通过大屏项目复制功能,您可以在现有大屏的基础上,开发类似的 大屏项目,或者作为开发测试环境来进行修改更新,不会影响在线生产应用。

操作步骤

- 1. 登录DataV控制台。
- 2. 单击我的可视化。
- 3. 在大屏管理页面,选择一个大屏项目,单击复制图标。



复制成功后,系统会自动生成一个名为"xxx\_副本"的大屏。您可以修改此大屏的名称,并在 此大屏项目的基础上,修改大屏的布局和配置。

### 7 删除可视化应用

本文档介绍如何删除一个大屏项目,帮助您快速删除一个不需要的可视化大屏,释放大屏空间与配额。



应用删除后不可恢复,请谨慎操作。

操作步骤

- 1. 登录DataV控制台。
- 2. 单击我的可视化。
- 3. 在大屏管理页面,选择一个大屏项目,单击删除图标,删除当前应用。



### 8预览可视化应用

本文档介绍如何预览一个大屏项目,帮助您及时查看开发完成的大屏效果,便于大屏的修改和完 善。

#### 操作步骤

- 1. 登录DataV控制台。
- 2. 单击我的可视化。
- 3. 在大屏管理页面,选择一个大屏项目,单击右上角的预览图标。



预览成功且大屏符合预期后,可将大屏发布到线上环境供其他人员在线观看。

#### 9 发布可视化应用

本文档介绍如何发布一个大屏项目。通过大屏项目的发布功能,可以将已经开发完成的大屏发布到 线上环境,并进行访问权限的设置,供其他人员在线观看。

操作步骤

- 1. 登录DataV控制台。
- 2. 单击我的可视化。
- 3. 在大屏管理页面,选择一个大屏项目,单击右上角的发布图标。



4. 在发布对话框中,打开发布分享开关。

发布	×
发布分享: 💶 开启	查看发布教程
分享链接	
http://datav.aliyuncs.com/sha	are 复制

打开发布分享开关后,请先选择下方的快照,再单击 变更 按钮,会随机生成一个分享链接。

#### 薑 说明:

现在的发布功能变更为表单形式。填写完表单后,单击变更按钮,发布功能才生效。变更完密 码、token、快照后也都是一样,需要单击变更才生效。 5. 单击访问链接右侧的复制。

॑ 说明:

访问链接功能右侧新增了重新生成链接的功能。适用于页面分享后需要重新生成新的分享链接 的场景,重新生成后,旧的分享链接不可用,请用新的分享链接访问。

6. 在浏览器中粘贴复制的链接,即可在线访问您的可视化应用。

大屏分享成功后,您还可以进行访问限制和发布页面的配置。

#### !) 注意:

为了更好地支持分享,DataV大屏分享页域名即将从datav.aliyun.com/share/example切 换为 datav.aliyuncs.com/share/example,此变更可能对您的DataV大屏项目造成的影 响,对您造成的不便敬请谅解。请参考#unique\_15及时排查并解决可能存在的问题。

访问限制

DataV的发布功能提供了三种分享大屏的方式:

- ・公开分享。
- ・访问密码分享。
- ・ Token 验证分享。

请选择合适的方式以方便其他用户进行可视化应用的访问。

密码访问(仅企业版及以上用户可用)

1. 在发布对话框中,勾选访问密码。

访问限制	无限制
验证方法 ②	
🗹 访问密码	<b>请</b> 输入访问密码
Token验证	开启后随机生成验证码
验证有效期(	) 🗩 关闭

2. 在访问密码输入框中,输入您的验证密码。

密码长度为6位以上,且必须具备以下三个条件:

- ・至少包含一个英文大写字母A~Z。
- ·至少包含一个英文小写字母a~z。
- ・至少包含一个数字0~9。

密码设置成功后,系统会提示设置密码成功。

3. (可选) 配置验证有效期。

#### !) 注意:

只有密码设置成功或开启Token验证后,才可配置验证有效期。

- ・ 开启验证有效期,可以设置密码的有效期,最长为32小时。访问者首次输入密码且成功访问
   大屏后,在设置的有效期时间内,可任意访问该大屏而无需输入密码。
- ・ 关闭验证有效期,每次访问都需要输入密码。

密码设置成功后,当您再次访问大屏的分享链接时,系统会提示需要输入密码。

已打开密码保护,请输入密码

Token验证(仅企业版及以上用户可用)

您可以通过Token验证的方式,将大屏访问权限与您的权限体系进行集成。

在发布对话框中,勾选Token验证,即可开启Token验证。Token验证开启后,您可进行如下操作:

· (可选) 配置验证有效期。

- 开启验证有效期,可以设置Token验证的有效期,最长为32小时。访问者首次进行Token验 证并成功访问大屏后,在设置的有效期时间内,可任意访问该大屏而无需再次进行验证。
- 关闭验证有效期,每次访问都需要通过验证。
- ·获取Token验证码。

开启Token 验证功能后,DataV会生成一个Token,如下图所示。您需要记录这个Token,以 备后用。

访问限制	与限制
验证方法 ⑦	
访问密码	开启后可设置访问密码
✓ Token验证	
验证有效期 ②	────────────────────────────────────

Token验证开启后,再次打开您所分享的页面,会收到一个Access Denied消息,表示您的访问 被拒绝了。如果想要打开您的页面,需要完成下面几个步骤:

! 注意:

为了防止重放攻击,请确保您的服务器时间为东8区标准时间。DataV只会提供1分钟的误差,如 果时间误差超过1分钟将会验证失败。

- 1. 发布大屏,记录大屏编码(URL的最后一段)。
- 2. 将编码与当前时间(毫秒)连起来,并用 | (竖线)分隔开。
- 3. 使用Token通过HMAC-SHA256 base64,对上一步得到的Token验证码进行加密。
- 4. 将时间和加密后的签名分别命名为\_datav\_time, \_datav\_signature。
- 5. 将它们依次放入url的querystring中。

#### 

如果您的大屏URL中需要使用Get的方式传递参数,为了安全性,建议您使用DataV提供的Token参数签名校验,详情请参见DataV分享页Token参数签名校验。

示例代码如下:

#### · PHP:

```
<?php
$token = "kBwoX9rFX9v4zbOT0Gjd_wr65DZ3P_WW";
$screenID = "03d1b68faeb09671046d1ef43f588c33";
$time = time()*1000;
$stringToSign = $screenID.'|'.$time;
$signature = urlencode(base64_encode(hash_hmac('sha256', $
stringToSign, $token, true)));
$url = "http://datav.aliyuncs.com/share/".$screenID."?_datav_time
=".$time."&_datav_signature=".$signature;
?>
<iframe width=100% height=100% src="<?=$url?>"/>
```

• Node.js:

```
const crypto = require('crypto');
var token = "Ev97w0USAtJusc3Vsd902ngr_vfVFH67";
var screenID ="14c5448c00ecde02b065c231d1659f38";
var time = Date.now();
var stringToSign = screenID +'|'+ time;
var signature = crypto.createHmac('sha256', token).update(str).
digest().toString('base64');
var url="http://datav.aliyuncs.com/share/"+ screenID +"?_datav_time
="+time+"&_datav_signature="+ encodeURIComponent(signature);
```

• Java:

```
package com.company;
import java.security.*;
import java.util.Date;
import javax.crypto.*;
import javax.crypto.spec.SecretKeySpec;
import org.apache.commons.codec.binary.Base64;
import java.net.URLEncoder;
public class TokenTest {
    public static String getSignedUrl(String screenID, String token)
{
         Date date = new Date();
        Long time = date.getTime();
         String stringToSign = screenID + "|" + time;
         String signature = HMACSHA256(stringToSign.getBytes(), token
.getBytes());
         String url = "http://datav.aliyuncs.com/share/"+ screenID
 +"? datav_time="+time+"&_datav_signature="+ signature;
         return url:
    }
    /**
        使用java原生的摘要实现SHA256加密。
     *
     * @param str加密后的报文。
     * @return
     */
    public static String HMACSHA256(byte[] data, byte[] key)
    ł
         try
              {
             SecretKeySpec signingKey = new SecretKeySpec(key, "
HmacSHA256");
             Mac mac = Mac.getInstance("HmacSHA256");
             mac.init(signingKey);
             return URLEncoder.encode(byte2Base64(mac.doFinal(data
)));
        } catch (NoSuchAlgorithmException e) {
```

```
e.printStackTrace();
           } catch (InvalidKeyException e) {
                e.printStackTrace();
           }
           return null;
       }
       private static String byte2Base64(byte[] bytes){
           return Base64.encodeBase64String(bytes);
       }
       public static void main(String[] args) throws Exception {
           System.out.println(getSignedUrl("screenId", "token"));
       }
  }
• C#:
  using System;
  using System.Security.Cryptography;
  using System.Text.RegularExpressions;
  using System.Collections.Generic;
  using System.Linq;
  using System.Web;
  using System.Text;
  namespace datavToken
   {
       class Program
       ſ
           static void Main(string[] args)
           ł
                var dic = new Dictionary<string, string>(); // 自定义参
  数。
               dic.Add("datav_sign_no", "123998");
                                                                11
  datav_sign_开头, 需要签名。
               dic.Add("datav_sign_lo", "kk");
dic.Add("datav_sign_ao", "xx");
               dic.Add("name", "123"); // 不需要签名。
// 分享页前缀, 屏幕分享id、token, 自定义参数字典。
Console.WriteLine(GenerateUrl("https://datav.aliyun.com
   /share/", "ca74bea5e45503070d607795e0*****", "66DsL2qjrXRHluSJScv_
   flOUhn*****", dic));
           }
           private static string GenerateUrl(string datavBase, string
  screenId, string token, Dictionary<string, string> customeParams)
                string pattern = @"^datav_sign_.*";
                string timestamp = GetTimeStamp();
                // 参数排序。
               Dictionary<string, string>.KeyCollection keyCol =
  customeParams.Keys;
               List<string> signKeys = new List<string>();
                foreach (var item in keyCol.ToList())
                {
                    if (Regex.IsMatch(item, pattern))
                    {
                         signKeys.Add(item);
                    }
                }
                // 按照key排序。
```

```
signKeys = signKeys.OrderBy(k => k).ToList();
            string paramsSignStr = signKeys.Aggregate("", (total,
key) =>
            {
                if (total != "")
                ł
                    total += "&";
                total += key + "=" + customeParams[key];
                return total;
            });
            string signStr = screenId + "|" + timestamp + "|" +
paramsSignStr;
            var encoding = new System.Text.ASCIIEncoding();
            byte[] keyByte = encoding.GetBytes(token);
            byte[] messageBytes = encoding.GetBytes(signStr);
            string signature;
            using (var hmacsha256 = new HMACSHA256(keyByte))
            ł
                byte[] hashmessage = hmacsha256.ComputeHash(
messageBytes);
                signature = Convert.ToBase64String(hashmessage);
            }
            var paramDic = new Dictionary<string, string>();
            paramDic.Add("_datav_time", timestamp);
            paramDic.Add("_datav_signature", signature);
            foreach (var item in customeParams)
            {
                paramDic.Add(item.Key, item.Value);
            }
            return datavBase + screenId + "?" + ParseToString(
paramDic);
        public static string GetTimeStamp()
            TimeSpan ts = DateTime.UtcNow - new DateTime(1970, 1, 1
, 0, 0, 0, 0);
            return Convert.ToInt64(ts.TotalMilliseconds).ToString();
        static public string ParseToString(IDictionary<string,</pre>
string> parameters)
            IDictionary<string, string> sortedParams = new
SortedDictionary<string, string>(parameters);
            IEnumerator<KeyValuePair<string, string>> dem =
sortedParams.GetEnumerator();
            StringBuilder query = new StringBuilder("");
            while (dem.MoveNext())
            ł
                string key = dem.Current.Key;
                string value = dem.Current.Value;
                if (!string.IsNullOrEmpty(key) && !string.IsNullOrEm
pty(value))
                    query.Append(key).Append("=").Append(HttpUtility
.UrlEncode(value)).Append("&");
```

```
}
string content = query.ToString().Substring(0, query.
Length - 1);
return content;
}
}
```

选择要发布的快照(仅企业版及以上用户可用)

设置分享链接后,可以配置快照发布,指定访问者看到的大屏版本(默认为快照发布版本)。屏幕 的内容会锁定在快照创建的那一刻,存档之后,屏幕内容的编辑和修改不会同步到历史快照中,可 作为稳定预览版本的备份。具体操作方式如下:

1. 在发布对话框中,单击选择要发布的快照 > 新增当前时间为快照。

新建后,系统会将单击时的大屏配置项作为快照进行存档。



- 可在注释列,添加注释。
- · 关闭发布分享功能之后,快照的数据会继续保留。
- 2. 再次单击新增当前时间为快照,可新增历史快照。
- 3. 单击快照右侧删除,删除多余快照。

▲ 警告: 快照目前仅支持切换,不具备回滚的功能,删除后将无法恢复,请谨慎操作。 4. 单击快照前的椭圆点,即可完成快照与快照之间的切换。

*选择要发布的快照 ②				
■ 2019/11/8 下午5:27:49 快照 回	[填写注释			
■ 2019/11/8 下午3:01:42 快照 □	[填写注释	2		
+ 新增当前时间为快照				
	取消	变更		

5. 快照设置成功后,单击下方变更键,刷新分享页,即可看到您选中的快照版本的大屏内容。 常见问题

原先已经实时发布的大屏如何迁移?

- ・没有在首页或编辑页单击发布按钮的大屏,保持实时发布。
- ・需要変更发布的大屏,单击进入发布页之前,会对当前在编辑大屏自动打快照。跳转发布页
   后,选择快照,单击变更按钮,即可完成快照发布。

### 10 DataV分享页Token参数签名校验

本文档介绍在发布DataV大屏时,使用Token参数签名校验的方法。通过Token参数签名校验 功能,您可以对大屏交互时传递的参数进行签名鉴权,保证大屏的URL访问链接的参数不能被篡 改,从而提高大屏数据以及用户信息的安全性。

前提条件

在使用Token参数签名校验前,请确保:

- ・大屏使用Token验证 (仅企业版及以上用户可用) 的方式进行发布。
- ・大屏以Get的方式在URL中传递参数(直接在URL后面加参数)。
- ·大屏URL中传递的参数要求不能被篡改。

#### 背景信息

下文以一个示例说明Token参数签名校验的使用背景。

某用户的系统嵌入了DataV大屏, URL通过Token计算, 通过Get方式传递用户的工号给大 屏展示相对应的数据,可以使用https://datav.aliyun.com/share/xxx?\_datav\_tim e=1556022195845&\_datav\_signature=%2BDZFj3QDIla%2F00fBZLdJMgk2Z10cs9 MLL1GiHdYkwa0%3D&workid=123来访问大屏。

其中workid (工号)为大屏传递的参数,存在被篡改的可能。比如工号为123的员工将URL改成 https://datav.aliyun.com/share/xxx?\_datav\_time=1556022195845&\_datav\_sig nature=%2BDZFj3QDIla%2F00fBZLdJMgk2Z10cs9MLL1GiHdYkwa0%3D&workid=124,就 可以看到工号为124的员工的资料。因此需要对用户传递的参数进行签名鉴权,保证计算得到 的URL的参数不能被更改,如果私自更改了传参,页面将无法访问。

#### 签名参数规则

需要加入签名的参数,其参数名需以datav\_sign\_开头,后面可以带任何有效的参数名字符。由此可得此参数名的正则表达式为/^datav\_sign\_.\*/。

不符合签名参数规则的参数,将不会进行参数签名校验,允许修改参数值。

带签名参数的URL计算

Node.js代码示例如下。

```
const crypto = require('crypto');
const querystring = require('querystring');
const signedQueryParamReg = /^datav_sign_.*/; // 符合此正则表达式的参数是
需要签名的。
```

```
const token = "93TWnmeBtxxxxxxx3thGyAgzennsS";
const screenID ="b92xxxxxxxxxxxxxxxx27b4c538cd4";
const time = Date.now();
const customeParams = {
  datav_sign_no: 123998,
  name: 123
};
let signParamsStr = Object.keys(customeParams)
  .filter(paramName => customeParams[paramName] && signedQueryParamReg
.test(paramName))
  .sort()
  .map(param => `${param}=${customeParams[param]}`)
  .join('&');
let stringToSign = [screenID, time];
signParamsStr && stringToSign.push(signParamsStr);
stringToSign = stringToSign.join('|');
let signature = crypto.createHmac('sha256', token).update(stringToSign
).digest().toString('base64');
let queryParams = \{
  _datav_time: time,
  _datav_signature: signature
};
Object.keys(customeParams).forEach(paramName => {
  queryParams[paramName] = customeParams[paramName];
});
let url = `http://datav.aliyun.com/share/${screenID}?${querystring.
stringify(queryParams)}`;
console.log(url);
```

使用以上代码示例得到的URL为: http://datav.aliyun.com/share/b92db8e093 58c82efca0727b4c538cd4?\_datav\_time=1556023246894&\_datav\_signature= GGSbvxlemUeBoRVco8JgrJVWRcmao7NuRYt20rGBC5g%3D&datav\_sign\_no=123998&name =123。在URL的有效期内,如果修改了datav\_sign\_no字段的值,链接将无法访问;如果修改 了name字段的值,链接仍然可以访问,因为datav\_sign\_no符合签名参数规则,参与了签名计 算,而name不符合签名参数规则,不会进行签名计算。

使用流程

- 1. 确定需要签名计算的参数名(即不允许被篡改的参数)。
- 2. 在大屏开发完成后,使用Token验证的方式发布大屏。
- 3. 参考带签名参数的URL计算,计算大屏的URL。
- 使用上一步中计算得到的URL访问大屏,在大屏访问过程中,系统会自动进行参数签名校验。
   如果参数签名校验功能正常,当您修改了签名参数,再次访问此URL时,访问会被拒绝。