ThingJS

CampusBuilder 用户手册

北京优锘科技有限公司

2019年4月16日

目录

第1章	引言		6		
第2章	产品说明				
2.1	特性说明]			
2.2	功能简介	·			
2.3	下载安装	ē	9		
第3章	主界面功	台上 月七			
3.1	主界面介	~绍			
	3.1.1	标题栏.			
	3.1.2	菜单栏.			
	3.1.3	功能展为	示栏13		
	3.1.	3.1	工具栏14		
3.2	我的场景	<u>.</u>			
3.3	场景集市	ī			
	3.3.1	购买场量	쿛18		
3.4	.4 DIY 模型库1				
	3.4.1	贴图上住	专21		
	3.4.2	OBJ 模型	型上传21		
	3.4.3	全景球期	照片上传22		
第4章	uBuilder	功能	25		
4.1	4.1 搭建功能				
	4.1.1	界面介约	召25		
	4.1.	1.1	uBuilder 菜单栏		
	4.1.	1.2	uBuilder 工具栏		
	4.1.	1.3	自定义建筑工具栏27		
	4.1.	1.4	绘图工具栏		
	4.1.	1.5	视图工具栏29		

	4.1.	1.6	场景内模型搜索
	4.1.2	编辑功能	
	4.1.2	2.1	场景编辑
	4.1.2	2.2	模型属性编辑35
	4.1.2	2.3	组合模型编辑40
	4.1.2	2.4	参考图导入41
	4.1.2	2.5	模型贴图43
	4.1.2	2.6	传送点设置44
	4.1.2	2.7	距离测量45
	4.1.3	模型中心	
	4.1.3	3.1	模型库46
	4.1.3	3.2	模型检索47
	4.1.3	3.3	模型属性48
	4.1.3	3.4	模型材质52
	4.1.3	3.5	OBJ 模型上传52
	4.1.4	自定义建	筑
	4.1.4	4.1	自定义建筑绘制
	4.1.4	4.2	编辑建筑外立面
	4.1.4	4.3	建筑结构隐藏
	4.1.4	4.4	自定义建筑工具栏
4.2	预览功能	1	57
	4.2.1 行走模式		57
	4.2.2 楼层展开		
	4.2.3 视点编辑		58
	4.2.4 视点动画.		
	4.2.5	GIF 录制	60
4.3	API 增强		60
4.4	快捷操作		

4.5	文件格式	<u>.</u>		65
第5章	数据源接	入		67
5.1	DIX 系统.			67
	5.1.1	DIX 介绍	J	67
	5.1.2	数据接入	、	68
	5.1.2	2.1	接收 MySQL 数据	68
	5.1.2	2.2	接收 ActiveMq 数据	71
	5.1.2	2.3	接收 Syslog 数据	74
	5.1.3	数据处理	<u>I</u>	77
	5.1.	3.1	处理 MySQL 数据	77
	5.1.	3.2	处理 activemq 数据	81
	5.1.	3.3	处理 syslog 数据	86
	5.1.4	数据输出	1	87
	5.1.4	4.1	输出到 MySQL	87
	5.1.4	4.2	输出 ActiveMq	88
	5.1.5	DIP 启动	l	89
	5.1.	5.1	保存	89
	5.1.	5.2	启动	90
	5.1.	5.3	运行	90
	5.1.	5.4	查看日志	90
5.2	场景接入	数据源		91
	5.2.1	CampusE	Builder API 调试环境实时数据接口接入	91
	5.2.	1.1	CampusBuilder 调试环境使用介绍	91
	5.2.	1.2	接入 MySQL 数据	93
	5.2.	1.3	接入 ActiveMq 数据	99
	5.2.	1.1	发布模模搭场景1	04
	5.2.2	ThingJS	调试环境接入1	05
	5.2.2	2.1	ThingJS 调试环境使用介绍1	05

	接入 MySQL 数据	5.2.2.2	
<u>ع</u> 111	接入 ActiveMq 数	5.2.2.3	
	发布 ThingJS 场景	5.2.2.4	
		关于我们	第6章

第1章 引言

伴随着物联网概念的兴起,物联网技术与传统计算机系统,通讯系统的不断融合,其应 用场景已经延伸到了各行各业。在一个完全智能化的应用场景里,智能硬件通过自身携带的 传感器,依靠无线或有线网络进行数据传输,根据环境的变化自动响应,自动调节,提高设 备的效率和准确性。整个过程无需人为参与,却能为人们节省金钱和时间。在一个 万物互 联"的网络环境中,软件与硬件高度融合将生产出大量数据,并在无数条无形的高速公路上 进行实时传输,如此庞大的数据量以及多元的 IT 设备对于运维管理提出了全新的挑战。传 统 IT 运维系统运用计算机技术通过图表、数字、图形、2D 平面图等表达方式把真实世界中 存在的实物和信息抽象化,用于管理与维护应用场景中的 IT 资源和数据。这些表现形式在 信息传达方面有很强的局限性,已经无法满足像物联网这样庞大且复杂的网络环境。因此, 物联网可视化需要一种更直观,更立体的 3D 数据展示平台。



图 1.3D 物联网监控系统-地铁枢纽

3D 物联网可视化平台几乎可以适用于任何场景,真实世界本身就是以 3D 的形式存在, 人类肉眼捕捉到的信息几乎全部是 3D 信息,因此直观的 3D 数字化场景能够承载更多、更直 接的实时数据,更容易被察觉,更容易被捕获。尤其应用到复杂且多变的物联网环境中能够 很大程度上降低误操作事件的概率,减少人员部署,提高管理效率,节省管理成本。

针对 3D 场景的制作,目前市场上主要依赖 3DSMax, Maya, AutoCAD, Sketchup 等主流 建模软件,这些人们耳熟能详的建模平台功能强大,能够实现大量的建模需求,例如:工业 产品设计、家具设计、建筑结构建模、力学测试等。但这些软件的使用者多为各行各业的专业人士,不经过培训,普通人很难上手进行操作,而过于细节化的建模工具,大多以点、线、面等矢量模型单位进行建模,使得建模过程繁琐,建模效率低下。如果选用这些传统建模软件搭建场景将耗费大量人力物力,使原本为了节约成本,提高效率所开发的 3D 物联网平台变得既不经济也不高效。因此,物联网 3D 场景的制作,需要一个操作简单,易于上手且搭建效率高的场景搭建平台。

CampusBuilder(也叫"模模搭")是优锘科技推出的一款专门针对物联网设计的 3D 可视化软件。同时,CampusBuilder也是优锘科技物联网 3D 可视化 ThingJS 平台的一个组 件。CampusBuilder 致力于帮助非专业 3D 建模人员快速搭建 3D 场景。CampusBuilder 不仅 提供了一个使用简单易用、功能强大的 3D 搭建工具,同时配合提供了丰富的 3D 模型库,即 使是没有任何图形编辑经验的普通用户也可以快速搭建起例如:商场、医院、学校、仓库、 公交车站等常见 3D 场景,大大提升了场景搭建效率。搭建好的 3D 场景通过简单的操作就可 以上传到 ThingJS 在线平台上进行物联网可视化系统开发。

本手册提供了使用 CampusBuilder 进行场景搭建的详细指导,将帮助用户有效使用 CampusBuilder,尽快完成场景搭建工作。



图 2. CampusBuilder 搭建出的场景(优锘科技办公楼)

第2章 产品说明

2.1 特性说明

CampusBuilder 的特性主要有:

1) 快速上手

CampusBuilder 依托先进的 3D 引擎技术和高效的人机交互设计,提供低门槛场景搭建能力,任何具有基本计算机操作能力的普通用户,都可轻松在数字化世界中构建一个真实场景。

2) 快速搭建

CampusBuilder 具有快速场景搭建能力,建模人员可通过鼠标快速"画"出楼层墙体结构,快速"摆"放物体在场景中的位置,依托丰富的 3D 模型库,5 分钟即可"搭建"一个场景。

3) 模型扩展

CampusBuilder 支持模型扩展,提供海量 3D 模型,用户也可以上传自定义模型至 CampusBuilder 中,即可像官方 3D 模型一样使用它们。

4) ThingJS 支持

通过 CampusBuilder 搭建的 3D 场景,可直接用于 ThingJS 开发,快速实现 3D 物联网行业应用。

2.2 功能简介

CampusBuilder 的主界面包括我的场景、场景集市、DIY 模型库。

- 【我的场景】用于存储,查看用户所创建的场景,可以对每个场景进行搭建、调试、 下载。点击【搭建】按钮,打开 uBuilder 进行场景编辑, uBuildr 是面向非图形
 图像专业人士提供的设计模块,使用便捷、功能强大;通过鼠标点击即可编辑、调试场景脚本,完成动态场景制作并将场景下载到本地。
- 【场景集市】提供了各种现成场景,用户可以在此购买需要的场景,购买成功后, 即可在【我的场景】中查看、编辑该场景。

【DIY 模型库】提供了搭建场景所需要的模型、贴图等,方便用户根据自己的需求
 任意挑选并添加到场景中,提高搭建效率。

2.3 下载安装

"模模搭"安装包可以在"模模搭"官网以及 ThingJS 官网下载,并提供多种下载方式可以选择,下载链接如下所示:

"模模搭"官网: <u>www.3Dmomoda.com;</u>

ThingJS 官网: http://www.thingjs.com/guide/?m=campus;

第3章 主界面功能

3.1 主界面介绍

客户端主界面展示如下:



图 1. 客户端主界面

主界面由四部分组成:

- 标题栏: 左侧展示客户端的全称及版本号, 右侧为窗口操作按钮。
- **菜单栏**:包含我的场景、场景集市、DIY 模型库、主菜单按钮等。

功能展示栏:主要由三部分构成,分别为搜索排序工具栏、场景列表区和场景信息
 展示区。

● **底部信息栏:**展示用户未同步场景、模型、贴图数量。

CampusBuilder-用户手册 VO.7

3.1.1 标题栏

o CampusBuilder(模提塔) V1.2.1.204

압 - 미 ×

图 2. 主程序标题栏

在标题栏中,左侧展示了客户端的 Logo、全称和版本号,右侧为窗口的操作按钮,可 放大、缩小、关闭客户端。点击 [℃]可更换客户端的皮肤,系统默认为【简约白】皮肤。现 有两种皮肤展示如下:



图 3. 皮肤-简约白

CampusBuilder-用户手册 VO.7



图 4. 皮肤-经典黑

3.1.2 菜单栏

在菜单栏中,主要分为两个部分,左侧是三个主页签,我的场景、场景集市、DIY 模型 库,右侧为四个功能按钮。

 我的场景
 场景集市
 DIV模型库
 + 新建
 ♀ 立即同步
 Ξ
 ● 张三

图 5. 主程序菜单栏

菜单栏左侧主页签对应客户端主程序的三大模块:

● 我的场景:储存场景,包括了用户自己搭建的场景以及复制、购买的场景。

场景集市:场景买卖平台,用户可将自己搭建的场景在此售出,也可购买他人出售的场景。

DIY 模型库:提供创建场景所需的模型、贴图等,也可以自定义上传用其它软件制作的模型。菜单栏右侧有四个功能按钮,功能描述见下表:

序号	功能图标	功能名称	描述
1	╋ 新建	新建	新建场景;

2	G 立即同步	立即同步	与云服务器同步本账号的状态和信息;
3	≡	主菜单	主菜单功能入口;
4	张三 登陆		账户切换入口;

表格 1. 工具栏功能表

主菜单内含有五个功能按钮,功能描述见下表:

序号	功能图标	功能名称	描述
1	8 我的账户	我的账户	用户账户的明细列表;
2	⑦ 系统设置	系统设置	可设置客户端的导出、同步、调试等功能;
3	☶ 用户手册	用户手册	CampusBuilder 的用户操作手册;
4	🕞 官方论坛	官方论坛	CampusBuilder 的官方论坛;
5	 关于我们 	关于我们	CampusBuilder的相关信息,可在此查看客户端的更新日志;

表格 2. 主菜单功能表

3.1.3 功能展示栏

主界面的三大模块由搜索栏、筛选排序栏等组成的工具栏、场景列表、场景详细信息栏 构成,界面展示如下: CampusBuilder-用户手册 VO.7



图 6. 功能展示栏

3.1.3.1 工具栏

如图 9 所示工具栏分为两部分, 左侧为搜索栏, 右侧工具栏分为筛选条件和排序条件两部分。

场景名/场景ID	Q 筛选条件	∨│排序条件 📛 🛈
	图 9. 右侧筛选排序工具栏	<u>.</u>
● 搜索栏:		通过场景名称及 ID 可

找到对应的场景。

筛选条件:可根据拥有的场景类型和状态进行场景筛选,若勾选了筛选条件,则场景列表只展示符合条件的场景;若不勾选,则默认展示所有场景。

排序条件:提供两种排序方式,本地修改时间和历史创建时间。默认按照历史创建
 时间倒序排列。

3.2 我的场景

我的场景包括了用户创建,购买以及复制的所有场景,用户点击某一场景,该场景的全部信息将显示到右侧【场景详情】栏中。每一个场景都含有一个场景菜单项,如图 10 所示:





销售次数:0 场景ID:1551862379742 场景名:hc1521 创建时间:2019-03-0616:54:02 修改时间:2019-03-0616:56:49 本地创建时间:2019-03-0616:57:02 模型种类:114 贴图种类:21 对象/三角面/贴图:56168/7679246/1535... 代码:0行

图 10. 购买场景以及场景详情

场景菜单及其子菜单项展示如下:



图 11. 场景菜单

场景菜单中各个功能子项参见图表 3-6:

序号	功能名称	描述
1	CB 预览	使用 uBuilder 预览;
2	插件预览	加载插件,在浏览器端预览;
3	ThingJS 预览	通过 Thing JS 模式预览场景;

表格 3. 场景预览子项功能列表

序号	功能名称	描述
1	离线部署包	下载该场景的模模搭 3D 资源部署包;
2	单机部署包(EXE)	下载后直接运行即可离线预览该场景;
3	单机部署包(EXE+VR)	暂不可用;

表格 4	4. T	载场景子项功能列表
------	------	-----------

序号	功能名称	描述
1	CB 调试	使用 CampusBuilder 内置插件调试场景;
2	ThingJS 调试	以ThingJS 调试场景;
3	ThingJS 第三方调试	以 ThingJS 第三方调试场景;

表格 5. 调试场景子项

序号	功能名称	描述
1	以 VR 打开	以 VR 模式打开场景;
2	同步场景	同步本场景到云服务器;
3	设为私有/公有	设置场景为私有/公开;
4	出售场景	将场景在场景集市中出售(该场景作者为用户本身时,才可以 出售场景);
5	删除场景	点击后,删除该场景;

表格 6. 其它操作子项功能列表

3.3 场景集市

场景集市展示如图所示:

CampusBuilder-用户手册 VO.7



图 12. 场景集市

场景集市是一个场景买卖平台,点击任意场景时,右侧展示该场景的详细信息,包括该场景售价、ID、名称等。

3.3.1 购买场景

场景集市中出现的场景,均由场景作者自主出售,并制定售价。用户选择需要购买的场景,点击 **②**【按钮】,出现付款界面。用户可选择支付宝或微信扫码付款,也可使用优惠券进行付款。用户购买场景后,将获得该场景的副本,并拥有对此副本的编辑、预览权限,可在【我的场景】中查看购买的场景。如图所示:

CampusBuilder-用户手册 V0.7



图 13. 场景集市-场景截图

3.4 DIY 模型库

DIY 模型库主要由系统目录和用户目录(非系统目录)组成。其中,共有7个系统目录, 分别为已购买的模型、已上传的模型、已复制的模型、照片建模的模型、已上传的全景球、 已添加的 0BJ 模型、垃圾箱。

DIY 模型库目前仅支持 3 层目录结构, 第一层是系统根目录, 只能存放系统目录及用户 自己创建的目录; 第二层可存放文件夹目录、贴图或模型; 第三层只可存放贴图、模型。 CampusBuilder-用户手册 V0.7



图 14. DIY 模型库展示

序号	功能图标	功能名称	描述
1	◆ 新建目录	新建目录	新建用户目录;
2	◆ 上传资源	上传资源	上传贴图、Obj 模型、全景球,建模照片入口;

表格 7. DIY 模型库功能列表

点击上传资源,可上传贴图、OBJ模型、全景球照片等。界面展示如下:

上传资源		×
	贴图上传 Obj模型上传 全景球照片上传 照片建模	
	确定取消	

图 15. 上传资源

3.4.1 贴图上传

用户可在非系统目录上传自定义贴图,在系统根目录中不可上传贴图。贴图支持 JPG 和 PNG 格式。上传的贴图可在 uBuilder 右侧模型列"个人"中,与 DIY 模型库对应的文件 夹内找到,与万能物体配合使用。

上传贴图		×
贴图名称	支持jpg,png格式,分辨率控制在10240以内	浏览
	上传取消	

图 16. 上传贴图

3.4.2 OBJ 模型上传

在【上传资源】处选择【OBJ 模型上传】即可打开以下界面:

上传Obj模型	2	×
Obj模型	模型路径	选择模型
	确定取消	

图 17. OBJ 模型上传

可在输入框中输入 OBJ 模型所在路径,或点击【选择模型】浏览模型所在路径,选择需要的模型,点击确定即可上传。要求上传的 OBJ 模型为".zip"格式。上传成功的模型存放 在系统目录"已添加的 OBJ 模型"中,同时可在场景搭建时使用上传成功的模型。uBuilder 内上传 OBJ 模型参见 4.1.3.5。

OBJ 模型导入规则

- 1) OBJ 模型文件需打包导入,格式为*.zip。文件包中含有模型、材质、贴图、缩略 图文件。
- 2) 文件包组成(注:模型、贴图、材质、缩略图,需在同级目录) 模型文件:.obj格式
 贴图文件:.png.jpg.dds.tga格式
 材质文件:.mtl格式
 缩略图文件:.png格式(文件名须与obj文件一致)

注意事项

1) OBJ 模型的每个节点的顶点数量不能超过 65000, 否则会报错。

2) 模型文件、贴图文件、材质文件、缩略图文件必须以英文名命名。

其他建议

使用 3ds Max 制作 OBJ 模型,在导入 CampusBuilder (模模搭)前建议预先缩小模型尺 寸为原尺寸的 1/100,以便在 "基于 unity3d 引擎的 CampusBuilder"中能按原尺寸显示该 模型。关于 3ds Max、unity3d 两种平台间的尺寸体系差异,若有更多疑问,可自行百度搜 索了解。

上传 OBJ 模型可参考下图:

	🦻 퉬 \sofa.zip\sofa\		-
	名称	大小	
	📄 sofa.obj	574 990	模型
	sofa.mtl	1 834	材质
	🔝 sofa.png	351 305	缩略图
	sofa_03_01.jpg	388 994	
	🔤 sofa_03_03.jpg	520 549	
a a fa alia	sofa_03_02.jpg	389 160	贴图
sora.zip	🔤 color.jpg	18 827	~

图 70BJ 模型上传示例

3.4.3 全景球照片上传

在"上传资源"处选择【全景球照片上传】即可打开以下界面:

建模进度(1/2)		×
上传照片: _{请确保图片满足以下条件:}		
 ✓ jpg格式 ✓ 全景照片 	✓ 单张横竖比例2:1✓ 照片大小低于10MB	
 提示 请上传专用相机拍摄的全景图; 目前仅支持上传单张全景图照片; 请上传小于8192*4096(像素)的照片; 上传过程中请耐心等候,切勿中途关闭 选择全景图后,点击"开始创建"系统 	窗口防止上传失败; 将目动生成全景图,生成后可点击预览。	
点	↔ 击选择文件	
进度:未上传		
		开始创建

图 18. 全景球照片上传 1



图 19. 全景球照片上传 2

选择全景图照片后,会自动上传照片,照片上传成功后,即可点击【开始创建】按钮,页面跳转只下一步,如图 19 所示。点击【查看全景图】或复制全景图的 url,可以看到转换后的全景图。同时,可在场景搭建中,通过【插件助手】中的【全景球助手】中使用。具体参见 4.1.2.1-5)。

CampusBuilder-用户手册 VO.7

第4章 uBuilder 功能

uBuilder 主要分为两个功能,搭建功能和预览功能。

搭建功能:即编辑场景,用户可根据需求进行场景编辑,包括模型属性编辑、组合模型编辑、参考图导入、CAD导入、模型贴图等。

预览功能:含有行走模式、楼层展开、视点编辑、视点动画、GIF 录制几个功能。

4.1 搭建功能

菜单栏右侧,点击【新建】打开 uBuilder。



4.1.1 界面介绍

打开 uBuilder 界面会有数秒加载过程,加载完毕后的界面如下:



图 19. uBuilder 菜单

4.1.1.1 uBuilder 菜单栏

文件 编辑 选择 工具 视角 插件 帮助

图 20. uBuilder 菜单栏

uBuilder 菜单栏共分 7 个菜单项,每个菜单项又分别含有不同的子菜单项。各菜单项 描述见下表:

序号	功能名称	描述
1	文件	进行打开、保存/另存为、导出以及导入 CAD 等操作;
2	编辑	撤销/恢复对当前场景的操作;
3	选择	选择同类型模型(暂未开放);
3	工具	参考图、场景截图;
4	插件	支持官方插件;
5	视角	切换 2/3D 视角,聚焦场景/物体;
6	帮助	uBuilder 场景搭建快捷键说明(暂未开放);

表格 8. uBuilder 菜单项功能表

4.1.1.2 uBuilder 工具栏



图 21. 菜单栏

uBuilder 工具栏共有7个功能按钮, 描述见下表:

	功能图标	功能名称	描述
1		打开	导入 cb1 文件、全量场景包件;
2		保存	保存当前场景;
3	Ō	场景截图	分高中低三种画质对场景进行截图;
4		参考图管理	上传建筑物布局图等用于辅助场景搭建;
5	140		对整体场景进行设置,包括场景光影及色彩特效,是否提交烘
5	342		焙,是否开启右键旋转;
6	?	帮助	uBuilder 场景搭建快捷键说明;

表格 9. 菜单栏功能列表

4.1.1.3 自定义建筑工具栏

自定义建筑工具栏主要用于对自定义建筑进行控制。

く返回	建筑

图 22. 自定义建筑工具栏

自定义建筑工具栏的功能显示会根据不同的图层变化(如外部场景、自定义建筑内、楼 层内),功能描述见下表:

序号	功能图标	功能名称	描述
1	< 返回	返回	返回上一图层;
2	▲ 建筑	建筑	场景中的自定义建筑列表;

表格 10. 自定义建筑工具栏功能表

4.1.1.4 绘图工具栏

绘图工具栏提供场景内模型绘图工具。



图 23. 绘图工具栏

通过这些工具,可以更便捷的对场景内的模型进行编辑,具体功能描述见下表:

序号	功能图标	功能名称	描述
1	4	选择	选中场景中任意对象,并对其进行编辑
			该功能即是选中场景一个模型,可以进行的操作,选中后有:
			(属性) 设置、自定义属性编辑、批量复制、镜像、删除。
2	1	在 选	该功能即是用框选的方式择场景中的几个模型,选中后操作有:
			(属性) 设置、 自定义属性设置、组合、复制、镜像、删除。
3	÷	无地板画墙	开启本功能后,将以无地板模式画墙。
4		一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	该功能即是用直线的方式在场景中对墙体进行绘制。主要用于
		四旦线垣	户不规则形状的墙体。注意,绘制完的墙体若未闭合,则不会

			自动填充地板。
E		画矩形墙	该功能即是用直线的方式在场景中对墙体进行绘制,主要用于
5			用户规则形状的墙体。
G	~		如果需要将墙体打断,可以通过该功能实现,添加过墙点的墙
6	ν_{\pm}		将不再是连续的整体,可以对两个点间的墙体进行其他的
7		擦除	可以删除给墙的加点(及点两边的墙体)。

表格 11. 绘图工具栏功能表

4.1.1.5 视图工具栏

视图工具栏主要用于对当前场景的展现方式进行设置,包括 2D/3D 视角切换、聚焦特定模型、是否展示室外场景等功能。



图 24. 视图工具栏

具体功能描述见下表。

序号	功能图标 功能名称		描述	
1	2D	看 2D/3D(空格键)	点击或按空格在 2D、3D 视角中切换;	
2		聚焦物体	聚焦至当前的物体或建筑;	
3	•	行走模式	以第一人称进行行走,按【Esc】按钮返回;	

表格 12. 视图工具栏功能表

4.1.1.6 场景内模型搜索

当 uBuilder 场景中模型数量众多时,寻找特定模型较为困难。uBuilder 提供场景搜索功能,可以搜索当前场景中的模型。搜索关键字可以是模型的自定义 ID,也可以是模型名称。搜索框在右侧模型库的搜索页签中。



图 25. 场景搜索

输入搜索内容时,搜索框下方会即时展示符合搜索条件的模型列表,场景中的对应模型 上方也会出现提示气泡。选中其中一个模型,则场景视角自动聚焦至该模型。



图 26. 待选模型列表及模型提示气泡

4.1.2 编辑功能

4.1.2.1 场景编辑

在 CampusBuilder 中可以非常方便的搭建和管理场景。用户点击主界面的【新建】按钮, 即可打开新建场景的页面,等待数秒加载完毕后,即可开始搭建场景。 在 CampusBuilder 场景中,按住鼠标左键可旋转场景,鼠标左键还可从模型库中拖拽模型至场景中、绘制建筑结构、对模型进行选中并进行下一步操作等,按住并拖动鼠标右键可以平移场景,滑动鼠标滚轮可以进行场景缩放。

通过菜单栏可以对场景进行保存、另存,恢复,对场景的设置参见 4.1.1.1 所列功能。 详细说明如下:

1) **打开**:

a) 打开:用户可以从外部导入 cb1 文件或全量场景包。

b) 打开 CAD 文件:

点击【打开 CAD 文件】,首先出现 CAD 导入的前导页,用户导入 CAD 文件前,可参考前导页图纸导入标准,点击下一步,进入图 31,选择需 要上传 CAD 文件。文件上传成功后,可按照 CAD 图纸进行识别墙、合并 墙、识别门、添加门等操作。



图 27. CAD 导入

2) 保存:

用户保存当前场景,在保存时可更改场景名称、另存场景、添加场景说明。

CampusBuilder-用户手册 V0.7

保	存场景	3
场景封面	3	
不满意?换个角度再点	氯保存试试!	
请输入场景名称	•	
请输入场景描还		APR -
另存	保存	

图 28. 保存场景

保存时,可选择保存的格式,格式说明如下:

序号	文件格式	描述
1	.cb1	由场景文件和自定义模型组成,场景文件包括场景的信息、通用模型等。
2	.tjs	由 ThingJS 场景文件和自定义模型组成。".tjs"格式的文件只能由 ThingJS 打开。

表格 13. 场景文件格式说明

3) 场景截图:

场景截图提供三种保存清晰度,分别是默认、高清(3K)和超清(4k),清晰度为高清 和超清的截图对电脑配置有一定要求,低配置机器慎用。

4) 设置:

此功能是设置当前场景的一些光影及特效,点击【设置】按钮,弹出如下界面:

特效设置	操作设置 烘焙
阴影 🧲	● 色彩校正 ●
时间轴	14:00
太阳方位	-,
曝光	
大气散射	<u> </u>
龗	
阳光系数	j
太阳尺寸	
红	
绿	
蓝	
	重置

图 29. 设置面板-特效设置

操作设置是对是否开启右键旋转进行设置。

特效设置	操作设置	烘焙	×
右键旋转 第一人称变 鼠标操作提	· · · · · · · · · · · · · ·	鼠标 右键	

图 30. 设置面板-操作设置

烘焙是优化场景显示,烘焙会对场景进行固化,若对烘焙过的场景进行修改,预览时会 出现错误,可删除烘焙修改场景后,再次提交烘焙。

特效设置	操作设置	烘焙	×				
烘焙	烘焙效果会在预览时候生效						
烘焙		删除					

图 31. 设置面板-烘焙

5) 插件管理-全景球助手

此功能和 DIY 模型库中"全景球照片上传"配合使用,需要预先上传全景球照片,然后 才可使用此功能。"全景球照片上传"使用参见 3.4.3。点击【全景球助手】,弹出如下界 面:

	全景球助手 V0.12	?⊗
当前层级:world		⊕⊖⊖G
场景中全景球模型	场景中的关系	
模型uid 关联次数	 关系序列 全景球A 关系 → → 	类型 全景球B
< 1/1 >	< 1/1	
		部署

图 32. 全景球助手

使用步骤:

- 1) 创建新的全景图模型,中央绘制区域会出现新的全景图模型;
- 2) 点击 2 后, 弹出右侧 DIY 模型库中上传的全景图列表, 选择需要的全景图;
- 3) 点击确定后,全景图和全景图模型相对应。剩余模型按上述方法创建完毕;

CampusBuilder-用户手册 V0.7

点击 4,在面板左侧"场景中的全景球模型"列表中,选择需要填入关系的模型。
 点击 ()按钮,可以将上述选择的关系和模型对应起来。当一个关系的两端全部配置完成后,场景中会出现相应的箭头标识;

5) 点击【部署】按钮, 部署完成后可在当前场景双击预览,或者在预览场景模式下 查看。



图 33. 全景球助手使用步骤

面板右上角有【使用教程】,点击 22 按钮,即可获取全景球使用教程。

4.1.2.2 模型属性编辑

在模型中心中,有模型库、模型信息、调色盘和搜索四个子页面。在模型库中所有模型 按照使用场景及类别分类,点击场景中的模型,打开模型信息页面可查看该模型的属性信息, 点击调色盘可更改该模型的材质及颜色。

模型中心里有两个搜索入口,模型库上方的搜索框可通过关键词搜索模型库中的模型, 搜索子页面则用来搜索当前场景中的模型。

CampusBuilder 中所有的模型都已全面对象化,均带有自有参数编辑和驱动,并可任意修改除 ID、模型名外的其它属性。如下图所示:



图 34. 模型菜单

上方弧形从左至右依次为:设置、自定义属性设置、编辑信息牌、编辑、批量复制、复制、镜像、删除。

下方三个按钮从左至右依次为:拖动调整旋转角度、拖动调整位置、拖动调整上下位置。

1) 设置:

可以用来修改物体的基本属性,编辑自定义 ID、为物体重命名、或修改长宽高等参数。 勾选【预览隐藏】,则该模型在预览模式下不显示,勾选【预览时可选】,则该模型在预览 模式中可以被选中。勾选【接收数据】后,该模型可被外部数据驱动。

_ Q	6	19,66			Û	
模型ID:	872	9B4	E2'	74F24	293BEE	
自定义IC	请新	请输入自定义的ID				
名称:	物体					
长宽高:	0.3	4	0	.34	0.29	m
悬空: 0.3		5	m	旋转:	270.0	°
□ 预览隐藏 □ 预览时可选						
接收	ŝ		保存	₹视点]	

图 35. 模型-设置
2) 自定义属性设置:

用户可以根据实际搭建场景的需求,自定义属性;



图 36. 自定义属性设置

3) 编辑信息牌:

每个物体可以增添一个信息牌,点击【保存】只保留最近一次的编辑记录。用户可以在 样式选择中选择自己需要样式,信息牌中的文本均可再编辑。若勾选【3D 模型化】,则该 信息牌会变成普通 3D 模型,若不勾选,则信息牌始终正对屏幕。勾选【编辑时可见】,则 信息牌在也会在编辑的同时显示。 CampusBuilder-用户手册 V0.7

样式选择:	样式1 ~
分类命名:	默认 ~
	3D模型化 编辑时可见
言息牌设置	预览
_	
Λ	孵化大楼
A	花家地南街
/	
/	

图 37. 编辑信息牌

4) 编辑:

沿 x, y, z 轴三个方向对模型进行平移、旋转、缩放,按住 shift 键进行旋转时,则模型每次固定旋转 15°角。



图 38. 编辑模型

5) 批量复制:

连续复制选中模型并移动鼠标,按鼠标移动方向连续复制,按方向键可对模型方向进行 调整,按 ctrl 滑动鼠标滚轮可对模型间距进行调整,随光标移动显示当前复制的模型数量 及模型间距。



图 39. 批量复制

6) 复制:

复制选中模型(单个)。

7) 镜像:

原地 180°旋转当前选中模型,模型位置不变。



图 40. 模型镜像

8) 删除:

删除选中模型。

9) 拖动调整旋转角度:

按住该按钮移动鼠标,旋转选中模型,并显示旋转角度,松开左键完成旋转。

10) 拖动调整位置:

按住该按钮移动鼠标,在水平方向上移动选中模型,显示模型位移(x轴位移,y轴位移)。

11) 拖动调整上下位置:

按住该按钮上下移动鼠标,可延竖直方向移动选中模型,显示离地高度。



图 41. 位移及角度发生变化时显示变化数值

12) 模型吸附对齐:

在编辑场景时,有时会需要对若干个模型进行对齐摆放。uBuilder 提供吸附对齐功能。 按住 V键不要松开,移动光标,附近的模型端点会出现可选择的圆圈,选中后拖动至另一个 模型附近,即可用选中的模型端点去对齐另一个模型的端点。



图 42. 模型吸附对齐

4.1.2.3 组合模型编辑

选中若干个模型后,点击【组合】按钮,则多选中的模型会组合成一个模型,对应的按钮图标也会变为【解组】图标,该操作是可逆的。

自定义ID: 请输入自定义的ID
名称: 组
● 预览隐藏 ● 预览时可选

图 43. 组合模型

组合后的模型可以进行复制、镜像等编辑操作。

4.1.2.4 参考图导入

此功能主要针对用使用 CAD 或其他建筑物布局图纸进行平面图设计。用户可将这些图纸 导入到当前场景中,辅助用户创建 3D 场景。首先点击【参考图管理】按钮,弹出参考图管 理框:

			-	- x
查看管理	设置比例	透明度	J (

图 44. 参考图管理

点击【查看管理】,点击选择图片文件上传参考图,参考图只可上传一张,需要替换时 点击查看管理重新选择图片即可。参考图将自动定位在当前所处楼层的地面高度。

参考图上传完毕后,点击设置比例可对参考图进行调整:



图 45. 设置参考图比例

点击【应用】后点击【完成】结束设定。



用户还可以设置参考图透明度,点击右上角【关闭】按钮后隐藏参考图及参考图管理窗 口,再次点击参考图管理可调出。

点击右侧 开关按钮也可显示或隐藏参考图,但参考图管理窗口不会关闭。

4.1.2.5 模型贴图

用户可以自行上传贴图到 DIY 模型库中,编辑时通过右侧模型中心-个人选择需要的贴图。如果 DIY 模型库中的贴图模型有变动,可在编辑界面内通过【刷新】功能进行同步。

贴图的使用方法:选中贴图模型后,光标变为油漆桶形状,点一下需要修改的墙面(或 者地板、天花板、屋顶这类本身具有贴图性质的模型),该贴图自动平铺所选墙面,用户可 以通过调整模型的高度和宽度来修改平铺时单张贴图大小,从而调整贴图覆盖的纵向及横向 密度。如果用自定义贴图修改其他模型,则会根据贴图中的颜色更改该模型的配色方案。



图 47. 万能物体贴图效果

此外,模型库中的墙面、地板、天花板、屋顶等图案,属于贴图类型,用户同样可以调 整这些贴图的横向密度和纵向密度。 CampusBuilder-用户手册 V0.7



图 48. 修改墙壁贴图宽、高度前后对比

4.1.2.6 传送点设置

在编辑场景时,可以通过模型的自定义属性配置传送点。在预览场景时的第一人称行走 状态下,行走到传送点的位置会传送到配置的地点。87x6=522

在自定义属性中键入 key:_Portal_, value:要传送物体的自定义 ID,则预览时候走到物体上就会传送到指定 ID 的位置。

		(i) (P) (Q)
		物品属性设置
	模型ID	05D669BBDA97440DA482FAC
	自定义ID	2
	名称	物体
	尺寸	X 0.60 Y 2 Z 1.21
	悬空	0.01
	旋转	359.88
	预览时隐藏	
	预览时可选	
NV)	接收数据	
	自定义属性	Size : 1 +
AN AN	X	_Portal_ 3
		保存视点

图 49. 传送点设置

4.1.2.7 距离测量

在编辑场景时,按下【B】即进入测量状态。鼠标连续点击场景,创建测量路径。测量 状态中,按住 shift 键可竖直向上绘制路径(垂直于平面),按住 ctrl 键可在平面内沿水 平、垂直或 45°方向绘制路径。

点击鼠标右键停止绘制测量路径。

如需删除,点击路径末尾的删除按钮,即可删除该测量路径。



图 50. 绘制测量路径

4.1.3 模型中心

模型中心在界面右侧,分为四个页签,分别是模型库、模型信息、模型调色板及场景搜索。



4.1.3.1 模型库

模型库中的模型分为室外、室内、效果、生物、生活、交通、消安、机房、其它、抽象、 遗弃、个人、功能 14 个子库,可通过最右侧的列表菜单切换,以下对抽象、个人、功能 3 个子库进行简要介绍。

抽象:点击打开【抽象】模型子库,查看可以进一步摆放的要素包括万能物体、物体、 虚拟辅助物体三部分。若模型库中没有需要的模型,则可以使用万能物体打组拼接。

个人: 该功能是针对当前登录的用户 DIY 的模型。如果用户已经 DIY 了自己的模型并且 需要用到当前搭建的场景中,可以通过下图中的页面进行模型选择:

添加	C
ⅲ 垃圾箱	
ⅲ 已添加的OBJ模型	
ⅲ 已上传的全景球	
ⅲ 照片建模的模型	
ⅲ 已复制的模型	
ⅲ 已上传的模型	
ⅲ 已购买的模型	

图 52. 个人模型库

如果用户是第一次使用该功能,需要点击上图中的【添加】按钮,在主界面 DIY 模型库 中新建模型目录及上传新建模型。

功能:功能模型包括线路(三种颜色)、管线、漏水线、虚拟物体、曲线、数据六类模型,主要用于在场景中进行功能标注。除虚拟物体外,均可设置预览隐藏。

功能模型不可修改模型材质和颜色,但虚拟物体(文字)可通过 html 代码修改文字颜 色。举例: < color=yellow>CampusBuilder< /color>,将显示黄色的 "CampusBuilder" 三 个字。

曲线模型是有一定宽度、无厚度的线条,默认线宽为 1m,有带箭头和不带箭头两种线

条样式。曲线及曲线上的节点均可命名。曲线可以修改颜色,曲线上可贴图,贴图可流动, 流动速度可通过设置中的"贴图重复"调整。曲线宽度可通过"线宽"调整。



图 53. 曲线-设置

曲线可跨楼层绘制,在绘制曲线的过程中,Ctrl+Q,显示其他楼层,Ctrl+W 隐藏当前楼层。

4.1.3.2 模型检索

在模型库上方有模型库搜索窗口,可通过关键词搜索模型库中的模型,默认显示当前所 在子库的搜索结果,按该模型所属类别分类显示。用户可切换其它子库查看该关键词的搜索 结果。



4.1.3.3 模型属性

uBuilder 模型库中的模型根据特性可分为以下几类: 自定义建筑模型、门窗模型、墙模型、动态效果模型、动画模型、摄像头模型、贴图模型、普通模型。用户可以通过右侧模型中心的模型信息面板查看及编辑模型属性。

自定义建筑模型:用于创建自定义建筑,具体操作方法详见 4.1.4 节"自定义建筑"。

对于自定义建筑,用户可修改自定义建筑的尺寸(单位:米)和下沉距离,还可对其进行复制以及删除操作。需要注意一点,自定义建筑的尺寸会被实际建筑结构的尺寸覆盖显示,但属性里的长宽数据并未更改。

	i		Q
	建筑属性	主设置	
自定义ID			
名称	建筑		
宽度	10		
长度	30		
下沉	0		

图 55. 自定义建筑-模型属性

门窗模型:只能放置在墙模型上,若放置门窗模型的墙壁宽度不够,门窗模型会自动在 视觉上调整所处墙壁宽度(墙壁实际宽度未改变),但仍会造成门窗模型不能正常显示。用 户可以修改门窗的尺寸(单位:米)、开门方向及是否开门,目前 uBuilder 的门窗模型中 暂时只有开门的效果,没有打开窗户的效果。

	i		Q
	门窗属	生设置	
自定义ID			
门窗尺寸	X 0.56	1.5 Z	0.06
离地高度	0.6		
	改变开门	门方向	
是否打开			

图 56. 门、窗-模型属性

墙模型:场景中的【摆围栏】和【拉墙壁】创建的本质上都是创建的墙模型。自定义建 筑中的所有墙组成了该自定义建筑的建筑结构。用户可以通过画直线墙和矩形墙两种方式编 辑建筑结构,画好的墙可在选中点打断,也可直接在中点打断。墙的各种属性也可通过【选 中-设置】或右侧的模型属性面板进行编辑。

	(i) (i) Q
	墙属性设置
墙体高度	3
墙体宽度	0.25
水平倾斜	0
垂直倾斜	0
材质宽度	1
材质高度	1

图 57. 墙-模型属性

动态效果模型:带有动态效果的模型,一般用于丰富场景效果,当前有的动态效果模型 有喷水、火焰、闪电、水面(水面也属于贴图模型)等。与带动画的模型不同点在于,动态 效果不可暂停、不可更换动画。动态模型的模型菜单及模型属性与普通模型一致。

	(i) (P) (Q)
	物品属性设置
模型ID	9f6560d5-8fc5-411f-983b-7bbd{
自定义ID	
名称	物体
尺寸	X 6.34 Y 1.45 Z 1.51
悬空	0.01
旋转	180
预览时隐藏	
预览时可选	
接收数据	
自定义属性	Size: 0 +
	保存视点

图 58. 动态效果模型-模型属性

动画模型:模型库中,有一些模型带有动画效果,载入新场景后,场景中鞠躬欢迎的女性模型就是这样的模型。动画模型的动画可以在右侧模型属性面板中进行控制,可选择是否播放动画,也可以从下拉列表中更换动画。

	(i) (P) Q	
物品属性设置		
模型ID	59C64077035C4DDAA7E4544I	
自定义ID	object01	
名称	People	
尺寸	X 0.70 Y 1.72 Z 0.56	
悬空	0	
旋转	180	
动画	_defaultAnim_	
播放动画		
预览时隐藏		
预览时可选		
接收数据		
自定义属性	Size: 0 +	
	保存视点	

图 59. 动画模型-模型属性

摄像头模型:这是一类特殊模型,默认携带一条自定义属性用于规定物体类型为摄像头。 摄像头模型在预览时可以打开监控范围(视锥),可用于查看监控盲区。通过 API 二次开发, 输入摄像头 IP 地址与现实中的摄像头相连接,可在 uBuilder 场景中实时显示当前的监控画 面。摄像头模型的这些功能主要用于安防领域应用。

	(i) (P) Q
	物品属性设置
模型ID	2DBD1DCD7B5F49BDA43CC6
自定义ID	
名称	监控摄像头
尺寸	X 0.120 Y 0.21 Z 0.19
悬空	2.71
旋转	180
预览时隐藏	
预览时可选	
接收数据	
自定义属性	Size : 1 +
×	物体类型摄像头
	保存视点

图 60. 摄像头-模型属性

贴图模型:贴图模型包括模型库现有贴图及用户上传的自定义贴图,场景中的墙壁图案、 围墙效果、天花板图案、地板图案、水面效果均属于贴图。使用贴图时,光标变为油漆桶形 状,可改变选中的模型的外观。贴图模型的具体使用方式详见 4.1.2.5 节模型贴图。

普通模型: 普通模型是模型库中数量最多的。对于普通模型,在4.1.2.2 节中有详细描述,于此不再赘述。

	(i) (P) Q
	物品属性设置
模型ID	C01C935EEC6C47A08E5F382
自定义ID	
名称	物体
尺寸	X 2.680 Y 1.82 Z 2.08
悬空	0.0
旋转	180
预览时隐藏	
预览时可选	
接收数据	
自定义属性	Size: 0 +
	保存视点

图 61. 普通模型-模型属性

4.1.3.4 模型材质

选中模型后,在右侧【模型中心-调色板】面板中可以修改模型的材质及颜色。部分模型由于预设属性原因不支持修改。当前修改模型材质是整体修改,之后会对模型材质部分做进一步完善。

在调色板中,需要先将模型材质从"默认"修改为其它材质才能更改颜色。更改模型材 质后,还可调整模型的透明度,下方三条色值属性默认为最高值。



图 62. 更改模型材质及颜色

4.1.3.5 OBJ 模型上传

在模型中心"个人"中,点击【添加】,选择需要上传的 OBJ 模型,上传成功后,可在 "已添加 OBJ 模型"文件中找到并使用。此外,在 DIY 模型库中上传的 OBJ 模型,也存放"已 添加的 OBJ 模型"中。

4.1.4 自定义建筑

自定义建筑是uBuilder 提供的最灵活的一个模型,用户进入自定义建筑内部后,可以 自行绘制想要的建筑结构、摆放模型等。

4.1.4.1 自定义建筑绘制

以下是自定义建筑的具体绘制过程。



图 63. 自定义建筑模型

首先将【室内-楼】中的自定义建筑拖拽到场景中,放置在合适的位置。自定义建筑的 大小可以通过模型设置进行调整。

双击进入自定义建筑内部编辑更多细节。

在楼层搭建时,有两种墙体绘制方式,一个是画直线墙,一个是画矩形墙。绘制出的墙体默认为白墙,墙体高度 3m,可根据需要选中要修改的墙进行属性修改。



图 64. 墙壁设置菜单

墙面的高度厚度等属性均可通过设置进行调节。

更换墙面图案:点击刷内/外墙壁中的图案,当出现油漆桶图标时点击要刷的墙面改变 图案(刷内外墙壁时建议更换到 3D 视角)。

编辑完一层后,可以在左上角的菜单栏中点击【复制当前楼层】按钮,将复制该层的房间结构及所有物品;也可以点击【新楼层】按钮新增空楼层。

如果未进入自定义建筑,直接在场景中编辑建筑结构,则该"建筑"无法封顶,也无法 在一层基础上新建楼层,需要将处于"室外"的墙、模型等,移动进自定义建筑内变为室内 场景。具体操作是,先在场景中放置一个自定义建筑,双击进入,在左上方的菜单栏中点击 【属性设置】,在弹出窗口中勾选【移动室外场景到本层】点击确定,室外的墙和所有模型 就被移动到了自定义建筑内的当前楼层。

4.1.4.2 编辑建筑外立面

自定义建筑的默认外观由用户编辑的建筑结构决定,若觉得不好看希望更改外观,则可 通过编辑建筑外立面实现。

进入自定义建筑的建筑层面,点击【开始编辑建筑外立面】,从【室内-楼】中,选择 想要的楼宇模型拖拽到自定义建筑中,点击【返回】回到室外,即可看到该自定义建筑的外 观变成了所选楼宇模型的外观。



若想更换自定义建筑外观,删除原先的楼宇模型后重新拖拽新的楼宇模型即可。

图 65. 拖拽目标楼宇模型至自定义建筑内

4.1.4.3 建筑结构隐藏

建筑结构隐藏功能与前一节的编辑建筑外立面功能相关联。编辑建筑外立面后,室外场 景中,该建筑的外观变为选定楼宇模型的外观,建筑结构默认被隐藏。如果想在显示外立面 的同时显示建筑结构,可进入建筑后点击【建筑结构隐藏】状态按钮,此时状态变为【建筑 结构显示】,去除了默认的隐藏效果,再次返回室外场景,即可看到自定义建筑原本的建筑 结构。



图 66. 建筑结构显示

4.1.4.4 自定义建筑工具栏

自定义建筑工具栏主要用于对自定义建筑进行控制。由于所处不同场景层次时显示菜单 不同,故以下按自定义建筑层次对功能进行描述。

1) 建筑

点击建筑弹出自定义建筑列表,列表中背景色显示当前所在建筑,点击列表中的建筑名称,切换至该建筑。在建筑层面可以开始/关闭编辑外立面,可以显示/隐藏建筑结构。点击 自定义建筑中的楼层,弹出菜单:



图 67. 自定义建筑-菜单

功能从左至右依次为:

进入内部:进入选中楼层;

复制:在当前楼层上复制该层,包括楼层结构和该层所有模型;

新建:在当前楼层上新建空楼层;

向上移动楼层:将当前楼层向上移一层;

向下移动楼层:将当前楼层向下移一层;

删除:删除选中楼层,上面的楼层自动下移。

2) 楼层

CampusBuilder					B	0	\$	*	
< 返回	▲ 建筑	€楼层	ø	، چ					
		F1 1层							
		F2 2层							
		F3 3层	- 1						
		F4 4层	1						
		F5 5层							
		F6 6层							
		F7 屋顶							

图 68. 楼层菜单

点击【楼层】弹出楼层列表,列表中背景色显示当前所在楼层,点击列表中的楼层名称, 切换至该楼层。进入楼层后,左上角工具栏中增加显示以下按钮:

复制当前楼层:复制选定楼层至自定义建筑顶层上,复制内容包括楼层结构和该层所有

模型;

新楼层:在自定义建筑顶层上新建空楼层;

显示/隐藏天花板:显示天花板后可通过贴图工具改变天花板颜色,隐藏天花板后方便 修改房间结构及模型;

属性设置:设置楼层名称,墙的高度,勾选【生效到所有墙】则该层所有墙壁变为设置 高度,不勾选则之后新建的墙为设置高度,勾选在室外显示本层模型,则在场景中可通过打 开的门等看到该层内的模型,勾选【移动室外场景到本层】,室外所有模型、墙均移动到室 内,移动至室内后不可移除,若误操作,可通过撤销恢复;

删除:只能删除该层的所有墙和模型,不能删除楼层;

3) 房间

房间列表:点击列表中的房间名称,聚焦至该房间。

4.2 预览功能

uBuilder 预览功能是后续重点发展功能。对行业 3D 可视化而言,多数情况,3D 预览是 核心功能,而 3D 场景编辑只是生产过程,因此预览功能非常重要。

4.2.1 行走模式

在场景预览中,点击右上角的行走图标,将红色光标放置在闭合墙体构成的地面上,或 自定义建筑的室内,则会进入第一人称行走模式。在行走模式中,可按 shift 键进行奔跑加 速,也可以按空格键进行跳跃。需注意,当所在位置从闭合场景离开时,会造成无限掉落。 按 esc 键可退出行走模式。

自定义建筑的绘制详见 4.1.4 部分。

4.2.2 楼层展开

在有自定义建筑的场景内,双击进入自定义建筑,点击右上角的楼层展开按钮,可以对 自定义建筑进行楼层展开。 CampusBuilder-用户手册 VO.7



图 69. 楼层合并状态



图 70. 楼层展开状态

4.2.3 视点编辑

通过界面左上方菜单中的【工具-视点编辑】打开视点编辑面板。视点编辑中包括新建、 插入视点及编辑视点动画。

196 点 49 回				7
+ da				
	名称	视点	ī,1	
Flet	操作	-		1
A.		● 浏	览时显	示

图 71. 视点编辑

视点其实是特定位置的屏幕快照, 在视点展示时点击该视点即可直接转换场景角度、远近至该视点设定的位置。

用户在视点编辑中可以新建视点,插入视点,给视点命名,重新捕捉视点快照。并可设 定该视点是否在预览时展示。

4.2.4 视点动画

视点动画其实是将多个视点快照串起来形成连续的场景变化,场景变化时的视角飞行时间及各个场景的停留时间均可自行设置,默认飞行时间 2s,停留时间 5s。

视点动画的具体操作步骤是:从【工具-视点编辑】打开视点编辑面板,点击【视点动 画】页签切换,点击【插入】创建新的视点动画,点击【操作】,将场景旋转调整至满意的 视角,再点击【插入】,完成视点动画中第一个视点的编辑,重复旋转场景并【插入】视点 动画的操作,即得到多个视点连续切换变化的视点动画。点击右下角的播放按钮播放视点动 画,将从当前所处场景视角按设定的飞行时间变换至视点动画中的快照场景,并按当前排列 的视点快照的次序依次循环播放。

CampusBuilder-用户手册 VO.7



图 72. 视点动画

4.2.5 GIF 录制

GIF 录制功能基于视点动画功能。具体操作是,编辑好视点动画后,勾选屏幕右下角的 【录制 GIF 动画】按钮,点击【播放视点动画】,播放的同时将录制 GIF 动画并保存至【我 的截图】。视点动画默认连续循环播放,录制的 GIF 仅录制第一次播放的视点动画。

4.3 API 增强

API 增强了 CampusBuilder 的搭建能力,以下是 API 列表引用,包含 camera、object、gui、BaseObject、ScriptObject、util、input、console 类,覆盖了物体控制、资源、效果、动画、界面等功能。

皮	对象	函数	说明	说明	
/丁			函数说明	返回值	
1		changeTo2D	场景模式改变至2维(顶视);	none	
2		changeTo3D	场景模式改变至3维;	none	
3	camera	getEyePos	获取当前摄影机位置;	none	
4		getTargetPos	获取摄影机的注视点;	none	

5	fit		摄影机聚焦在给定物体上;	none
6		flyTo	摄影机以给定的时间飞向给定的位置 和注视点,完成后执行给定函数;	none
7	lookAt		设置摄影机的注视点;	none
8		setPosition	设置摄影机位置;	none
9		stopFlying	摄影机停止在触发该函数时所在位置、 注视点,通常与 camera.flyTo 结合使 用;	none
10		create	创建一个物体;	object 类型的已创 建物体
11		createArrowLine	创建一条箭头曲线;	object 类型的已创 建曲线
12	object	createCurveLine	创建复杂曲线;	object 类型的已创 建曲线
13		destroyAll	删除由脚本创建的所有对象;	none
14		find	按 uid 查找物体;	object 类型的已存 在物体
15		createBox	创建一个矩形 GUI 对象;	object 类型的己创 建矩形
16		createButton	创建一个按钮 GUI 对象;	object 类型的已创 建按钮
17	gui	createLabel	创建一个标签 GUI 对象;	object 类型的已创 建标签
18		createToggle	创建一个复选框 GUI 对象;	object 类型的已创 建复选框
19		load	载入外部的 GUI 资源;	none
20	Base Oh isst	addGravity	为指定对象添加重力(质量);	none
21	Daseobject	addScript	为指定对象添加脚本;	ScriptObject 类型

			的已添加脚本	
		为指定对象添加光尾,通常与		
22	addTail	movePath 万法协问使用, 可用米增强	none	
		初冲枕见双木;	BaseObject 类型的	
23	clone	复制一个已存在的对象;	己复制对象	
			BaseObject 类型的	
24	destroy	删除一个已存在的对象;	已复制对象	
			Vector3 类型的空	
25	getPosition	获取指定对象的位置;	间向量	
			Vector3 类型的空	
26	getScale	获取指定对象的尺寸;	间向量	
27	movePath	使指定对象按参数给定的方式移动;	none	
28	movoTo	使指定对象以给定时间移动至给定空	nono	
20	movero	间位置;	none	
29	pitch	使指定对象按 X 轴正向旋转给定角度;	none	
30	playAnim	使指定对象执行物体动画;	none	
31	removeScript	移除指定对象上的脚本;	none	
32	roll	使指定对象按 Y 轴正向旋转给定角度;	none	
33	setAnimSpeed	设置指定对象的物体动画速度;	none	
34	setColor	设置指定对象的颜色;	none	
35	setColorFlash	设置指定对象闪烁状态和方式;	none	
36	setPickFnahled	设置指定对象的选取可用状态,通常配	none	
50	Seti Tekilabieu	合鼠标的拖拽事件使用;	none	
37	setPosition	设置指定对象的空间位置(x, y, z 坐	none	
		标);	none	
38	setPositionXZ	设置指定对象的水平位置(x, z 坐标);	none	
39	setPositionY	设置指定对象的垂直位置(y坐标);	none	

40		setScale	设置指定对象的尺寸;	none	
41		setTransparent	设置指定对象的透明度;	none	
42		stopAnim	停止指定对象的物体动画;	none	
43	stopMoving		停止指定对象的移动状态;	none	
4.4		twongformDoint	萨梅 地宫对鱼竹根对似行为绝对似行	Vector3 类型的绝	
44			转供指足对家的相对坐称为把对坐标;	对坐标点	
45		translate	使指定对象移动相对位移;	Vector3 类型	
46		yaw	使指定对象按 Y 轴正向旋转给定角度;	none	
477		Start	目前使用范围局限在物体脚本中,脚本		
47	Consisted is at	Start	执行时仅执行一次该方法;	none	
40	ScriptObject	II. J. t.	目前使用范围局限在物体脚本中,脚本		
48		Update	执行时每帧执行一次该方法;	none	
49		addEventListener	添加一个事件;	none	
50		clearAllEvents	清除场景内的所有事件;	none	
F 1			清除所有的计时器,无需指定计时器		
91		clearAllimers	ID;	none	
5.9		al a an Indamual	清除 setInterval 方法生成的计时器,		
52		clearinterval	需指定计时器 ID;	none	
5.2		alaanGanintOhiaata	清除场景内由脚本创建的对象,包含物	2020	
55		clearscriptobjects	体,GUI 等;	none	
54	utii	alaanTimaaut	清除 setTimeout 方法生成的计时器,	nono	
94		creariimeout	需指定计时器 ID;	none	
55		downloadToxtuno	将一个外部 url 材质图片载入至场景	nono	
00		dowintoautexture	内;	none	
56		downloadToxtures	将多个外部 url 材质图片载入至场景	nono	
50		uownioaurextures	内;	none	
57		randomColor	生成一个 RGBA 类型的随机颜色值;	RGBA 类型的颜色值	
58		randomFloat	生成给定范围内一随机浮点数;	float 类型随机结	

				果	
59		randomInt	生成给定范围内一随机整数;	int 类型随机结果	
		1 1 1 0	生成一个 Vector3 类型的随机空间向	Vector3 类型的空	
60	randomVector3		量;	间向量	
			计时器记满后执行回调函数,而后计时	;nt 米刑的计时器	
61		setInterval	器清零再次计时,循环往复,返回	IIIL 尖型的口时奋	
			number 类型的计时器 ID;	ID	
62		setRenderCallback	每帧执行一次回调函数;	none	
60		a a t T i ma a su t	计时器记满后执行回调函数,返回	int 类型的计时器	
03		Setlimeout	number 类型的计时器 ID;	ID	
64		getKey	获取键盘给定键点击状态;	boolen 类型	
65		getKeyDown	获取键盘给定键按下状态;	boolen 类型	
66		getKeyUp	获取键盘给定键弹起状态;	boolen 类型	
67	input	getMouseButton	获取鼠标给定按钮点击状态;	boolen 类型	
68		getMouseButtonDown	获取鼠标给定按钮按下状态;	boolen 类型	
69		getMouseButtonUp	获取鼠标给定按钮弹起状态;	boolen 类型	
70		clear	清除控制台文本内容;	none	
71	console	log	在控制台打印指定对象的信息;	none	
72		show	控制台显示或隐藏;	none	

表格 14. API 列表

4.4 快捷操作

为了使操作更加简便快捷,uBuilder 设置了若干快捷键。描述如下:

1) 墙

Ctrl: 按住取消吸引。

Shift: 按住画垂直/水平的墙。

Delete: 删除墙。

2) 物体

Ctrl:按住单选物体;批量复制的时候,按住Ctrl加滚轮调整距离。同时按键盘的上下左右可以调整方向。

Shift: 按住可在 3D 场景下旋转物体角度,点要旋转物体时方向的小球有 3 个点,代表 3 个坐标轴的方向,可以固定每次 15 度角旋转。

Alt: 按住能选中同一模型的所有物体

F: 按住。如有选中就是聚焦,无选中就是聚焦整个场景

空格: 2/3D 切换

3) 参考图设置比例

Shift: 按住点鼠标左键能快速设置中心点。

ctrl: 按住加鼠标左键比例尺的第一个点。

Alt: 按住加鼠标左键是比例尺的第二个点。

Ctrl+z: 撤销操作

Ctrl+y: 重复操作

Shift: 多物体选择之后按住 Shift 再点击物体可减选

4) 预览

ASDWQZ: 有视角移动效果

空格: 2D/3D 切换

F: 物体聚焦

4.5 文件格式

uBuilder 在场景导出中支持了更多的文件格式,现对所有文件格式介绍如下:

序	分类	格式	格式名	格式说明
1		430	ThingIC 招导句	由 ThingJS 场景文件和自定义模型组成。该".tjs"格式
1	ThingTS	.tjs	IningJS 切京包	的文件只能由 ThingJS 打开。
2	IIIIIgjo	tic	际田 LoT 咨酒句	专为阿里物联网云打造。该".tjs"格式的文件可上传至
2		. ເງຣ	两主 101 页称已	阿里云物联网平台中,也可由 ThingJS 打开。
3		ch1	CB 场暑包(* ch1)	由场景文件和自定义模型组成,场景文件包括场景的信息、
0	CB	.001		通用模型等。
4	0.0	. ch2	CB 场景包(*. ch2)	由 UI 资源、场景文件和自定义模型组成,场景文件包括场
1		.001		景的信息、通用模型等。
5		. ison	uDCV 完整场景文件	生成可供 uDCV 产品使用的建筑模型信息及设备位置的数据
_	-			文件。
6	uDCV	.json	uDCVC 场景建筑文件	生成可供 uDCV 产品使用的建筑模型信息的数据文件。
7		.json	uDCVC 场景布局文件	生成可供 uDCV 产品使用的设备位置的数据文件。
0			жис (д	由 3DSMax 软件生成的源文件格式,安装了 3DSMax 软件后
8		.max	max 贠你包	即可打开此文件。同时也可以将此文件转换成其它格式。
0		Cl.	01. 次逝与	由 3DSMax 软件生成的模型文件格式,安装 3DSMax 软件后
9		.10X	10x 页까巴	即可打开此文件,同时也可以将此文件转换成其它格式。
10		3de	2da 2da 次派句	由 3DSMax 软件生成的模型文件格式,安装 3DSMax 软件后
10		. 505	JUS 页你已	即可打开此文件,同时也可以将此文件转换成其它格式。
		.obj		.obj文件是一种 3D 模型文件格式,适合用于 3D 软件模型
11			obj obj资源包	之间的互导,安装 3DSMax 或 Maya 软件既可打开此文件,
	3DSMax			同时也可以将文件转化成其他格式。
				. dae 是一种 3D 模型文件格式,可由 3DSMax、Maya 等软件
12		.dae	dae 资源包	输出,安装 3DSMax 或 Maya 软件即可打开此文件,同时也
	-			可以将此文件转换成其它格式。
13		st]	st1 资源包	.stl 是一种表示三角形网格的文件格式,安装 3DSMax 软件
10	-	. 5 01		后既可打开此文件,同时也可以将文件转化成其他格式。
				.wrl 是一种虚拟现实文本格式,可通过 VRML 浏览器或
14		.wrl	wrl 资源包	3DSMax 等软件打开,同时也可通过 3DSMax 软件后转化成
				其他格式。
				glTF资源包导出的压缩文件中包含了.json、.bin 和.gltf
15	GLTF	.zip	g1TF 资源句	格式的文件,其中.gltf格式是一种最新的实时 3D 资源交
	5511			换格式,用于互联网或移动设备上展现 3D 内容,充分支持
				opengl,webgl,opengles 图形标准。

表格 15. 文件格式列表

第5章 数据源接入

5.1 DIX 系统

5.1.1 DIX 介绍

DIX 是 CampusBuilder 中用于实时对接入数据进行处理后输出到指定目的地的系统。可以用来接入 Mysql、ActiveMq, syslog 等数据源中存储的数据,可以在接入这些数据源存储的数据后,利用 DIX 对数据进行简单处理后,输出到指定的数据源,目前支持 MySql、ActiveMq 等。

通过访问 DIX 服务器进入 dip 配置页面,选择 JavaScript,点击立即接入按钮之后进入了 dip 的配置页面,在该页面进行 dip 的属性配置、脚本配置。

配置完成上述信息之后,点击完成并保存,一个 dip 便生成了,然后回到数据集成页面 找到对应的 dip,启动即可。

Javascript dip 具体的使用方法,请看以下几小节。

← → C ① ③ 不安金 192.168.137. … □ <		[作] 19時] 学习] 68接] 出行]]	2) C C C C C C C C C C C C C C C C C C C				
支持厂商							
ZABBIX	НЗС	英形 科技	LONCOM	Nêtis₹⊒	Java JDBC		
立即被入	立即被入	立即接入	立即接入	立即被入	立即被入		
iManager NetEco	sysl ∕g	JavaScript	医 释世纪 CS13000-		Tarsier		
立即输入	立即输入	立即输入	立即输入	立即接入	暂不支持		
eSight	webservices**	vmware	Neusoft [®] 东软		ぞ 神州数码 Digital China		
留不支持	留不支持	留不支持	留不支持	留不支持	留不支持		

图 73. DI 主页

5.1.2 数据接入

5.1.2.1 接收 MySQL 数据

点击 JavaScript,进入配置页面,在1处填入 dip 名字,2处的地址为宿主机地址,默认为 dip 系统部署机器地址,点击下一步,填写脚本配置。

dip 名称:使用英文且见名知义(最好使用驼峰命名法)。

dip 的宿主机:运行该 dip 的服务器 ip 地址,也是我们的数据处理服务器。

脚本 -	
1、填入dip名称,自定义	
宿主机* ● 192.168.137.128:8082 ● 2、宿主机地址为部署dip系统的机器ip	
上一步	
2 脚本配置	

图 74. 新建 MySQL dip

进入如下界面,在黑窗口的 run 函数中编写 JavaScript 脚本,将获取到的数据转换成 产品需要的数据结构,最后输出。

首先编写脚本获取数据,获取数据有2种方式,第一种是主动获取,第二种是被动接收, 根据实际情况选择对应的获取数据方式。而每一种方式又有多种类型。 2 脚本配置

脚	本*	
	Mysql	输入-被动接收
	SqlServerJdbcAdapter	
fı	OracleJdbcAdapter	
	DB2JdbcAdapter	<pre>Ltype(com.unnova.dl.dlcom.jdbc.SqlServerjdbcAdapter / bcAdapter("jdbc:jtds:sqlServer://192.168.10.234:1433;Da 'able("", "select mgrobiid as ciCode, ch as metric, id valu</pre>
	SybaseJdbcAdapter	:e());
	PostgreSqlJdbcAdapter	·) [
	HttpGet	de^); >tCIbyCICode(obj.ciCode); ○.
	HttpPost	
	obj.value = temp.get(~valu obj.timestamp =JSON.string	ic"); e"); ify(new Date().getTime());

图 75. 主动获取



图 76. 被动接受

对接 MySQL 我们选择主动获取方式,点击输入-主动获取,选择第一项 MySQL,相应的 接入 MySQL 的代码会自动填写在 function run() 里面, in 为接入语句, executes 为数据 处理逻辑, outs 处填写数据输出语句。

如下图,在1处添加接入 MySQL 数据源 url, username, password,在2处添加查询简单的查询 sql 语句,查询自己需要的数据。

 ● DP配置 ● 単本配置 				
脚本。 输入:主动获取 工具	输入被动接收	数据处理	8 033	
<pre>section model if a section = junct type("section on thread if the section = junct type("section "yes") // function // section = section = section = section // section = section = section = section // section = section = section = section // section = section = section = section = section // section = section = section = section = section // section = section</pre>	razera, di, di na jako (graf) (kolakytera'); eneret generati skole - tras	etters (not)		
上一步 完成 贫弃损	H作,返回主页			

对接收到的 MySQL 数据进行处理,参考 5.1.3.1 小节。



代码示例:

<pre>function run() {</pre>
var MysqlJdbCAdapter =
<pre>Java.type("com.uinnova.di.dicom.jdbc.MysqlJdbcAdapter");</pre>
while (true) {
try {

```
var adapter = new
MysqlJdbcAdapter("jdbc:mysql://127.0.0.1:3306/database?useUnicode=t
rue&characterEncoding=UTF-8", "root", "root");
var data = adapter.getDataByTable("select * from ...");
heartBeat.addInCount(data.size());
//executes...
//outs...
sleep(60000);
} catch (error) {
logger.error("脚本执行错误: ",error);
}
}
```

5.1.2.2 接收 ActiveMq 数据

点击 JavaScript,进入配置页面,在1处填入 dip 名字,2处的地址为宿主机地址,默认为 dip 系统部署机器地址,点击下一步,填写脚本配置。

dip 名称:使用英文且见名知义(最好使用驼峰命名法)。

dip 的宿主机:运行该dip 的服务器ip 地址,也是我们的数据处理服务器。



图 78. 新建 ActiveMq

进入如下界面,点击输入-被动接受,选择第一项,相应的接入 ActiveMq 的代码会自动填写在 function run() 里面 in 为接入语句, executes 为数据处理逻辑, outs 处填写数据输出语句。

如下图,在1处依次填入ActiveMq数据源的url,queue,username,pasword,最后一个参数为true,表示接受queue,为false表示接受topic,

URL: MQ 的地址,格式为tcp://\${ip}:\${port},例如tcp://192.168.10.44:61616

对列名 queue: 产品接收的指定队列 noah_perfs.

用户名 username: admin

密码 password: admin


```
startReceiver(receiver);
while (true) {
    try {
        var data = receiver.getData();
        if (data!=null) {
            heartBeat.addInCount(1);
            //executes...
            //outs...
        }
        catch (error) {
            logger.error("脚本执行错误: ",error);
        }
    }
}
```

5.1.2.3 接收 Syslog 数据

点击 JavaScript,进入配置页面,在1处填入 dip 名字,2处的地址为宿主机地址,默认为 dip 系统部署机器地址,点击下一步,填写脚本配置。

dip 名称:使用英文且见名知义(最好使用驼峰命名法)。

dip 的宿主机:运行该 dip 的服务器 ip 地址,也是我们的数据处理服务器。



图 80. 新建 syslog

进入如下界面,在1处点击输入-被动接受,选择第一项,相应的接入 Syslog 的代码 会自动填写在 function run() 里面 in 为接入语句, executes 为数据处理逻辑, outs 处 填写数据输出语句。

IN 配置是指接收 syslog 告警数据必须的一些信息,根据具体的需求调研填写

如下图,在2处依次填入Syslog数据源的url,port, protocal,

主机 url: 部署 DIP 服务端程序的主机 IP

端口 port: 与第三方厂商 syslog 确认的 syslog 数据推送端口,建议填写大于 1600 的未被其他服务占用的端口,

协议 protocal: 与第三方厂商 syslog 确认进行 syslog 数据传输协议,可选 UDP 或 TCP

接入 syslog 数据库,对数据进行处理和输出到指定的数据源参考 5.1.3 数据处理和 5.2.4 数据输出章节所述。

脚本	•		
0	DIP配置		
2	脚本配置		1
	脚本*		
	输入-主动获取	输入-被动接收	
	工具		
	<pre>function run() {</pre>	iver. SyslogReceiver") :	
	startReceiver (receiver);		
	<pre>while (true) { try { var data = receiver.getData(); if (data!=null) { heartBeat.addInCount(1); 3</pre>		
	//executes 4		
	} } estch (error) { logger.error("脚本执行错误: ",error): } } }		

图 81. 接入 syslog

```
代码示例:
```

function run() {
var SyslogReceiver =
<pre>Java.type("com.uinnova.di.dip.base.receiver.SyslogReceiver");</pre>
//host, port, protocol(TCP or UDP)
<pre>var receiver = new SyslogReceiver("0.0.0.0", 1501, "UDP");</pre>
<pre>startReceiver(receiver);</pre>

```
while (true) {
    try {
        var data = receiver.getData();
        logger.info("[原始数据]" + data);
        if (data != null) {
            heartBeat.addInCount(1);
            //executes...
            //outs...
        }
        catch (error) {
            logger.error("脚本执行错误: ", error);
        }
    }
}
```

5.1.3 数据处理

5.1.3.1 处理 MySQL 数据

参考 5.1.2.1 章节所述,我们可以取出 MySQL 中的数据进行处理,将他们转为我们需要的格式,在进行输出。

在编写脚本数据转换的时候可以点击数据处理, 就会出现如下图所示的子菜单, 可以 根据接收到的数据类型(告警/性能)选择对应的菜单, 脚本框中会出现对应的模板代码, 如下图所示。

脚本配置							
脚本*							
输入-主动获取	输入-被动接收	告警数据映射					
工具		性能数据映射					
function runO {	function runO (
<pre>var SqlServerJdbCAdapter = Jat while (true) { var adapter = new SqlServer, var data = adapter.setDataB</pre>	va.typet com.uinnova.di.dicom.jndc.aqlserverjdbcAdapter);]dbcAdapter ('jdbc:jlds:sqlserver://192,108,10,234:1433,DatabaseNams=centerobj']vdble(''select merobid as ciCode.ch as metric.id value.unit.devnams from x	根据属性查询CI					
heartBeat.addInCount(data.s //executes var list = new Array();	ize());	根据索引规则查询CI					
<pre>for(var i=0;i<data.size();i <="" =="" data.get(i);="" obj="new" object();="" pre="" temp="data.get(i);" var="" vi="temp"></data.size();i></pre>	<pre>for (var i=0;iCdata.size();i++) [var obj = new Object(); var temp = data.get(i);</pre>						
var ciCode = temp.get(ci var ciCode = tarsierTool. if (ciCode=="{}") {	obj.ciCode = temp.set(ciCode); var ciCode = tarsierTool.getClbyClCode(obj.ciCode); if (ciCode=="0") {						
obj.CIObject = new Obje } else {	ct();						
<pre>obj.CIObject = ciCode; } </pre>							
obj.metric = temp.get(me obj.value = temp.get("val obj.timostame =ISON strip	tric /; ue ⁽⁷); us ⁽⁷⁾ : Tato() gotTime());						

图 82. 数据处理

以 JavaScript 处理接入 MySQL 为例,准备数据接入。

样例数据:

如下图,1处接入 MySQL 数据,2处对数据进行处理,可以参考下面数据处理中的性能数据映射代码,3处把处理后的数据同样输出到另外一个 MySQL 中去,对其他数据源的处理可以参照此处进行处理。

数据处理后的输出参考 5.1.1 小节,输出到 MySQL 数据。

	Maxing and (ad cut size ());				
	_istData = json2JavaList(data);				
var list	new Array();				
for(var i	;i <data.size();i++){< td=""><td></td><td></td><td></td><td></td></data.size();i++){<>				
var ot	j = new Ubject();				
var te	= town got("id");				
obj.10	onoid = town got("sconoid"):				
obj.st	i key = temp get("kni key");				
obj.kr	i value = temp.get("kpi value")				
logger	.info("obi处理后数据"+obi.toStri	, ng());			
list.	ush(obj);				
//logger.i	nto("listlength="+list.length);				
	F数据库的儿个万法,传入参数sql,		失败返回0		
	teTable(sql);创建表				
	teTable(sql); 修改表数据 teTable(sql);删除表				
	teTable(sql); 修改表数据 teTable(sql); 删除表 teTableStructure(sql); 修改表结 *tTableCcal): 李海表				
//out.upda //out.dele //out.upda //out.sele	teTable(sql); 修改表数据 teTable(sql); 删除表 teTableStructure(sql); 修改表结 ttTable(sql); 音调表 effueter = Java ture("com uiopo	تم va di din customou	3		
//out.upda //out.dele //out.upda //out.sela var Mysql0	teTable(sql); 修改表数据 teTable(sql); 働於表 teTableStructure(sql); 修改表结H ctTable(sql); 查询表 rtCustom = Jaxa.type("com.winnor mame_nescound		3 tmxsql.MysqlOutCusto		
//out.upda //out.dele //out.sele var Mysql0 //url, use var out =	teTable(sql); 修改表数据 teTable(sql);删除表 teTableStructure(sql); 修改表结 <u>tTable(sql): 查询表</u> utCustom = Jaxa.type("com. <u>uinna</u> "mame, <u>passound</u> ewe WosqlOutCustom("idbc:mysal:		3 Ltwy.sql.MysqlOutCusto		
//out.upda //out.upda //out.seld //out.seld //url, use var Out = //传入教据	teTable(sq1); 傍次表数据 teTable(sq1);删除表 teTableStruture(sq1); 修改表语; ttTable(sa1): 查違表 utCustom = <u>laxa</u> .type("com.winnog orname, passowicd maw Mysq10utCustom("jdbc:mysq1: 教留臣意名, 14: 具種(M)		3 Itmysal.MysqlOutCusto 3306/outputDB?useUni		
<pre>//out.upda //out.deld //out.seld //out.seld //out.seld //url, usd var MysqlQ //url, usd var out = //传入数据 out.out(co</pre>	teTable(sql); 傍次未数据 teTable(sql); 勝次表 teTableStructure(sql); 修改表结 :tTable(sol): 查過表 :tTuble(sol): 查過表 itCustom = Jaya.type("com. <u>winne</u> name, passSound new MysqlOutCustom("jdbc:mysql: 教授臣完美者,王徳,其他列 wert)s2Jaya(list),"cout","id"	≿j yg.di.dip. <u>customo</u> l //192.168.137.138: ,"sceneid,kpi kev.	3 .tmx.sql.MysqlOutCusto .3306/outputDB?useUni kpi value");		
<pre>//out.upda //out.upda //out.upda //out.seld var Mysql(//url, use var out = //传入数据 out.out(cg heartBeat.</pre>	teTable(sql); 傍次表数提 teTable(ture(sql); 歸除表 teTableStructure(sql); 修改表语; ttTable(sql): 查请表 ttTuble(sql): 查请表 ttCustom = Jaya, type("com.yinnon mame, passownd new MysqlOutCustom("jdbc:mysql: 数据译表名,主律,其他列 MymerL32Jaya(list),"c_out","id" addoutCount(list.length);		3 (tmysgl.MysqlOutCusto 3306/outputDB?useUni kpi_value");		
<pre>//out.upda //out.upda //out.upda //out.sela var Mysql(//url, use var out = //传入数据 out.gut(cs heartBeat.</pre>	teTable(sq1); 傍次表数据 teTable(sq1); 層於表 teTableStructure(sq1); 修改表培 ttTable(sal): 查違法 stCustom = <u>lava</u> .type("com.uinns mmame, <u>passowrd</u> mew Mysq1OutCustom("jdbc:mysq1: 费超度表名, 王徒, 其他例 by <u>ertJs2lava</u> (list),"c_out","id" ddOutCount(list.length);		3 (tmysql.MysqlOutCusto 3306/outputDB?useUni (kpi_value");		
<pre>//out.upda //out.deld //out.seld var Mysql0 //url,uss var out = //f{入数据 out.gut(gg heartBeat. sleep(5000)</pre>	teTable(sql); 傍次未数据 teTable(sql); 傍次未结 teTableStructure(sql); 停次未结 ;tTable(sol): 查過表 ;tTable(sol): 查過表 itCustom = Jaya-type("com. <u>winno</u> mame, passSound new MysqlOutCustom("jdbc:mysql: 對据庫表言,王雄,其他列 myertJs2Jaya(list),"c_out","id" addOutCount(list.length); b);		3 LTMYSQL-MysqlOutCusto 3306/outputDB?useUni kpi_value");		

图 83. 处理 MySQL 数据

DIP 代码参考如下:

<pre>function run() {</pre>
var MysqlJdbcAdapter =
<pre>Java.type("com.uinnova.di.dicom.jdbc.MysqlJdbcAdapter");</pre>
while (true) {
try {
var adapter = new
MysqlJdbcAdapter("jdbc:mysql://192.168.137.128:3306/inputDB?useUni
<pre>code=true&characterEncoding=UTF-8", "root", "root");</pre>
<pre>var data = adapter.getDataByTable("SELECT</pre>
<pre>id,sceneid,kpi_key,kpi_value FROM c_in");</pre>
<pre>heartBeat.addInCount(data.size());</pre>

```
//executes...
       //var javaListData = json2JavaList(data);
      var list = new Array();
       for(var i=0;i<data.size();i++){</pre>
        var obj = new Object();
        var temp = data.get(i);
        obj.id = temp.get("id");
        obj.sceneid = temp.get("sceneid");
        obj.kpi_key = temp.get("kpi_key");
        obj.kpi value = temp.get("kpi value");
        logger.info("obj 处理后数据"+obj.toString());
        list.push(obj);
       }
       //logger.info("listlength="+list.length);
       //outs...
       //下列为操作数据库的几个方法,传入参数 sq1,执行成功返回 1,执行失败返
日 0
       //out.createTable(sql);创建表
       //out.updateTable(sql); 修改表数据
       //out.deleteTable(sql);删除表
       //out.updateTableStructure(sql); 修改表结构
       //out.selectTable(sql); 查询表
```

```
var MysqlOutCustom =
Java.type("com.uinnova.di.dip.customoutmysql.MysqlOutCustom");
       //url, username, passowrd
       var out = new
MysqlOutCustom("jdbc:mysql://192.168.137.138:3306/outputDB?useUnic
ode=true&characterEncoding=UTF-8", "root", "root");
       //传入数据,数据库表名,主键,其他列
out.out(convertJs2Java(list),"c out","id","sceneid,kpi key,kpi val
ue");
       heartBeat.addOutCount(list.length);
       sleep(60000);
     } catch (error) {
      logger.error("脚本执行错误: ",error);
     }
   }
  }
```

5.1.3.2 处理 activemq 数据

参照 5.1.2.2 章节所述,我们可以取出输入 ActiveMQ 中的数据进行处理,将他们转为我们需要的格式,在进行输出。

以 JavaScript 处理接入 ActiveMQ 为例,准备数据接入。

连接配置信息:

url: tcp://{ip}:{port}

用户名: username

密码: password

对列名称: inTopicName

队列数据格式: JSON 格式的字符串

数据属性说明:

参数名	必选/可选	参数位置	参数类型	参数名字
ciCode	必选	参数列表	String	配置项唯一标识
Metric	必选	参数列表	String	性能数据所属的指标名 称
instance	可选	参数列表	String	性能数据所属的指标实 例,若无则填英文字符 下划线,"_"
Value	必选	参数列表	String	性能数据的值
timestamp	可选	参数列表	String	数据采集时间,若为空则 默认当前时间,时间戳 格式

样例数据:

Ι

{

"ciCode": "P211-E1-01",

```
"metric": "CPU 使用率",
"instance": "cup0",
"value": "50",
"timestamp": "1500790703007"
}
```

如下图,1处接入 MQ 数据,2处对数据进行处理,3处把处理后的数据同样输出到另外 一个 MQ 中去,对其他数据源的处理可以参照此处进行处理。

输出到 ActiveMq 参考 5.1.3.2 章节所述。



图 84. 处理 Active Mq 数据

Dip 代码参考示例:

function run() {

```
var ActivemqReceiver =
Java.type("com.uinnova.di.dip.base.receiver.ActivemqReceiver");
     //url, queue, username, password, queue(true) or topic(false)
     var receiver = new
ActivemqReceiver("tcp://192.168.137.128:8094", "testIn123",
"admin", "admin", true);
     startReceiver(receiver);
     while (true) {
      try {
        var data = receiver.getData();
        if (data!=null) {
          heartBeat.addInCount(1);
          //executes...
          var listData = JSON.parse(data);
          var list = new Array();
          for(var i=0;i<listData.length;i++) {</pre>
           var obj = new Object();
            var temp = listData[i];
            obj.ciCode = temp["ciCode"];
            obj.metric = temp["metric"];
            obj.value = temp["value"];
            obj.timestamp = temp["timestamp"];
```

```
obj.unit = temp["instance"];
           list.push(obj);
          }
          //logger.info("listlength="+list.length);
          //outs...
          var
mqurl="failover://(tcp://192.168.137.138:8094)?randomize=false&tim
eout=5000&maxReconnectAttempts=-1";
          var ActiveMqOut =
Java.type("com.uinnova.di.dicom.amq.AMQProducer");
          //url, 用户名, 密码, topic=1 或 queue=2, Topic 或者 Queue 名称,
非持久化=1 或持久化=2, 最晚提交时间秒
          var out =
ActiveMqOut.getInstance(mqurl,"admin","admin","1","testOut123",
2,1);
          out.sendList(convertJs2Java(list),1);
          heartBeat.addOutCount(list.length);
        }
      } catch (error) {
        logger.error("脚本执行错误: " + error.message);
       }
     }
   }
```

5.1.3.3 处理 syslog 数据

参照 5.1.2.3 章节所述,我们选择发送 syslog 数据,

Syslog 消息体

属性分隔符:|

数据定义: sceneid|kpi_key|kpi_value|value|id

样例数据: 20170109153114611633853 | temperature | 56 | 1

说明: syslog 在发送数据时会在数据前自动添加发送时间, 主机名称等内容, 需要根据产品是需要该数据进行处理,实际发送数据如下所示:

Feb1909:32:16 DESKTOP-PDH3088 20170109153114611633853 |

temperature|56|12

在 executes... 对输入数据进行解析,获得所需要的数据格式,输出到指定的 MySQL 或者 ActiveMq 数据源中,如下图所示

	nost, port, protocol(TCP or UDP) receiver = new SyslegReceiver("192.168.137.128", 1601, "UDP");
sh	understrong(receiver); Ede (true) { try {
	<pre>//20170109153114611633853[temporature[56]1 var data = receiver.getDat(); logget.ndv(][NBNRM]" + data); if (data1=mul) (</pre>
	//outs //下列列錄作数据库的几个方法,传入参数sq1,执行成功返回1,执行失数返回e //out.updateTable(sq1);创建表 //out.updateTable(sq1);创造表 //out.updateTableStructure(sq1); 答论表结构 //out.updateTableStructure(sq1); 答论表结构
	<pre>v=r/MysQDutCustom = Java_type("com.ulnnova.dl.dlp.customoutmysql/MysqlDutCustom"); //url, username, basSandd var out = new MysQDutCustom("jdbc:mysql://122.168.137.138.3306/outputDB?useUnicode=true&characterEncoding=UTF-8", "root", "root"); //(0, VBR: Bigmarks; = tdt, HRM out.out(convectJs2Java(11st), "sceme_data", "id", "sceneid, kpi_key,kpi_value"); heartEdest.addbutCustom(fist.length);</pre>
	} } catch (error) { logger.error("即本执行错误: ",error);

图 85. 处理 syslog 数据

5.1.4 数据输出

2 脚本配置

5.1.4.1 输出到 MySQL

进入如下界面,在编写脚本数据输出的时候可以点击输出, 就会出现如下图所示的子 菜单,可以根据发送方式选择对应的菜单,脚本框中会出现对应的模板代码,只需要修改一 下连接信息配置参数,如下图所示。

 max is a start of the s

图 86. 数据输出

点击输出,选择第二项 MySQL,相应的接入 MySQL的代码会自动填写在 function run() 里面 outs 处填写数据输出语句, in 为接入语句, executes 为数据处理逻辑,

如图,展示了关键处理逻辑:

1、为查询 MySQL 数据源所得数据,

2、对数据进行处理,

3、是一些操作数据库的函数,可以对输出数据库进行创建表,更新表结构等操作,4 处填写输出到目的地 MySQL 的配置信息。



5.1.4.2 输出 ActiveMq

进入如下界面,点击输出,选择第一项,相应的接入ActiveMQ的代码会自动填写在function run()里面 outs处填写数据输出语句, in 为接入语句, executes 为数据处理逻辑。

如下图,在1处为填入参数的含义,2处为要填写参数的值。



填写完成后参考 5.1.5.1 小节, 保存 dip。



5.1.5 DIP 启动

5.1.5.1 保存

填写完以上配置信息后,点击完成,进入该 dip 的保存界面,如下图所示。



图 88. 保存 dip

然后点击保存会在 di 主页看到接入 javascript 的 dip 已经存在,如下图所示。点击 启动按钮,便可以获取数据了。

javascript 192.168.1.183:1552			
	没有监控数据	2	
● 已停止	启动	配置	删除
	图 89. 启动 dip		

5.1.5.2 启动

配置完 dip 后,点击完成,保存好,回到 dip 主界面,显示所有的 dip 列表,如下图所示,点击启动,启动一个 dip。

已接入 🗹			帮助文档
1-mysql» mysql	2-mysqlmq	3-mysqlmq	syslog1 192.168.137.128.8082
没有监控数据	没有监控数据	没有监控数据	没有监控数据
● 已得止 2月得止	●	●	
接入ActiveMQ 192.168.137.128.8082	syslog222 192.168.137.128.8082		
没有监控数据	没有监控数据		
● #30 配置 母51÷			
년주止	Life)도		
支持厂商			

图 90. 启动

5.1.5.3 运行

刚启动时状态,已接入、已处理、待发送、已发送为0,启动一会后,如果已接入和已 发送有数值,则表示 dip 正在传输数据。

1-mysql> mysql	2-mysqlmq 192.168.137.128.8082	3-mysqlmq 192.168.137.128.8082	syslog1 192.168.137.128.8082
没有监控数据	没有监控数据	没有监控数据	没有监控数据
0 800 800 900 841	●	● 100 82 898 89±	800 82 899 291
接入ActiveMQ 192.168.137.128.8082	syslog222 192.168.137.128.8082		
EMA-1 19508-0 19508-0 EX88.1	没有监控数据		
日前約 日前約	● 已停止 ———————————————————————————————————		

图 91. 运行

5.1.5.4 查看日志

Dip运行后点击监控按钮,进入 dip 详细页面,点击日志下载,查看 dip运行日志。

DIP状态	Baa de Bata	E
DIP基本信息		
DIP名称	Javascript,SoripsDp)请示使用	
DIP唯一标识	Javascript_Javascript_Script_1_0_518384179880004	
DIP类型		
DIP产品名称	javascript_1_0	
DIP主机	172.16.2.20.8082	
DIP监控		日達入-124 (二447年)(1) (二4)(1)

图 91. 查看运行日志

5.2 场景接入数据源

5.2.1 CampusBuilder API 调试环境实时数据接口接入

第三方行业应用如需驱动特定 3D 场景各类信息动态变化,实现实时状态可视化展示, 需使用"实时数据"接口向场景内推送数据。

5.2.1.1 CampusBuilder 调试环境使用介绍

可以客户端中点击 CB 调试进入客户端离线调试环境,

o CampusBuilder (模模搭) V1.2.1.204			Ƴ - □ ×
我的场景 场景集市 DIV模型角	2 2	+	新建 🖸 立即同步 😑 🌑 xuyy 😒 🕬
场景名/场景ID Q		筛选条件 > 排序条件 🗒 🕚	
xuyy1 @6.00	編輯场展 預定场展 > 下載场展 > 隆磁振 > 隆世週武 ThingJS碼就 ThingJS第三方调试	1) ① ② ② ② ② ② ② ② ② ② ② ② ② ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ● ③ ③ ③ ④ ③ ● ④ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	・ 坊房详情
◎1●0★0	◎ 望和桥 ◎ 0 ● 0 ★ 0	◎ 望和桥 ⑧ 0 ● 0 ★ 0	贴图种类: 0/0
			对象/三角面/站图:4/1096/6 代码:0行 描述:
	● 元应均依 ● 1 ● 0 ★ 0	◎ 1 2 0 ★ 0	
	« 1 2 »		
		1 场景: 0/0	模型: 0/0 贴图: 0/0

图 92. 进入 CampusBuilder 离线调试环境

也可访问模模搭官网(www.3dmomoda.com),通过点击开发、在线调试进入在线调试界

面。

< > C 🛆 🗅 🙀 🤠 www.3dmomoda.com			Ŧ
	开发 - 下载中心	ThingJS平台 关于我们~	🥝 🕐 🔔 18
(ThingJS3D	▶ 介绍	12月20日 20:00 📀 腾讯课堂 在线直播!	免费报名 →
	88 教程 图 API		
3	<♪ 在线调试		

图 93. 在线调试

也可以直接访问如下网址进入调试界面,下载链接如下所示:

http://www.3dmomoda.com/mmdclient/script/examples/examples.html



图 94. 场景预览

如上图,进入调试页面后,在右侧输入自己想要调试的场景 ID,点击场景预览按钮, 右侧会显示场景,在左侧点击官方列表,在头顶牌子选择固定位置物体,接下来我们选择这 个官方示例代码来进行场景实时接入数据的演示。

点击执行代码后,会在右侧场景中场景一个头顶牌,显示 hello!我们可以利用数据接入接口接入所需要得数据,在调试代码中把数据赋值给对应的物体,达到头顶牌中的数据可以根据接入数据的变化而变化。

5.2.1.2 接入 MySQL 数据

进入调试环境,选择官方列表中头顶牌,选择固定位置,使用如下图所示的 Ajax 数据 对接(支持跨域请求)。

请求参数说明:

参数名	必选/可选	参数类型	参数说明	参数示例
requestUrl	必选	String	请求接口	http://192.168.10.67:8090/ajax/getDbData

username	必选	String	数据库用户 名	root
password	必选	String	数据库密码	root
dburl	必选	String	数据库地址	jdbc:mysql://192.168.137.138:3306/outputDB
sql	必选	String	查询 sql	SELECT id , sceneid, kpi_key,kpi_value from scene_data WHERE id = 1

样例数据:

返回结构说明:

```
{
    "state": true,
    "data": [
```

```
CampusBuilder-用户手册 VO.7
```

```
{
    "kpi_value": "253",
    "sceneid": "20170109153114611633853",
    "kpi_key": "temperature",
    "id": "1"
  }
}
```

调试代码示例:

```
ui.setImageColor("Button", Color.red);
      ui.setTextColor("Button/Text", Color.yellow);
      ui.regButtonEvent("Button", function() {
         console.log("click ui button");
      });
   };
   util.download({
      "url":
"http://www.3dmomoda.com/mmdclient/script/examples/demos/message_b
utton.bundle",
      "success": init balloon
   });
   // 测试显示隐藏:
   // util.setInterval(function() {
   // if (ui != null) {
   // ui.show(!ui.isShow);
   // }
   // }, 1000);
```

```
//设置定时器
   util.setInterval(function() {
          //请求参数
   var requestUrl= "http://192.168.10.67:9090/ajax/getDbData";
   var dburl="jdbc:mysql://192.168.137.138:3306/outputDB";
   var username="root";
   var password="root";
 var sql="SELECT id, sceneid, kpi key, kpi value from scene data WHERE
id = 1";
util.download({
      "url":
requestUrl+"?dburl="+dburl+"&username="+username+"&password="+pass
word+"&sql="+sql,
      "type":"text",
      "success": function(res) {
                 ////打印返回结果
               console.log(res);
                            //json 转换为 table 类型
               var tableres = json.decode(res);
```

```
// 取得自己需要的值
var kpi_value=tableres.data[0].kpi_value;
console.log("kpi_value:"+kpi_value);
//把结果复制给场景中头顶牌中
ui.setText("Button/Text",kpi_value);
},
"error":function(error){
    console.log(error)
    }
});
},3000); //每三秒
```

调试结果如下图所示:

在1处是请求数据实时渲染到场景中,2处为接口地址和请求参数,3处为通过图中方 法获得请求结果,请求成功后对于返回数据可以使用标准 API 语法解析,获取需要的指标参 数,赋值到场景对象中。



图 95. 接入 MySQL 数据源

5.2.1.3 接入 ActiveMq 数据

对于 ActiveMq 数据,同样采用 Ajax 请求获取数据源数据,支持跨域请求,请求地址可以是自己部署的 ActiveMq 服务器地址。

请求参数说明:

参数名	必选/可选	参数类型	参数说明	参数示例
requestUrl	必选	String	请求接口地址	http://192.168.10.67:9090/aj ax/getMqData
mqurl	必选	String	ActiveMq 地址	tcp://192.168.137.138:8094
username	必选	String	ActiveMq 用户名	admin
password	必选	String	ActiveMq 密码	admin

chanelMode	可选, 默认1	String	订阅类型,topic=1 queue=2	2
topicOrQueu e	必选	String	待接收 topic 或 queue 名 字	testOut789
timeout	可选,默认 3000	String	收到下一条消息时间,单位 ms	3000

请求示例数据:



返回结构说明:

```
{
    "state": true,
    "data": "[\n {\n \"ciCode\":\"P211-E1-01\",\n
\"metric\":\"CPU 使用率\",\n \"instance\":\"cup0\",\n
\"value\":\"50\",\n \"timestamp\":\"1500790703007\"\n }\n]\n"
```

100

}

调试代码示例:

```
// ------
// 创建牌子, 跟踪位置、设置文字、改变颜色、注册事件、显示隐藏
// -----
var ui = null;
function init_balloon(res) {
  ui = gui.create(res);
  ui.trackPos(0, 2, 0);
  ui.setScale(0.3, 0.3);
  ui.setText("Button/Text", "Hello!");
  ui.setImageColor("Button", Color.red);
  ui.setTextColor("Button/Text", Color.yellow);
  ui.regButtonEvent("Button", function() {
    console.log("click ui button");
  });
```

```
};
   util.download({
       "url":
"http://www.3dmomoda.com/mmdclient/script/examples/demos/message b
utton.bundle",
       "success": init balloon
   });
   // 测试显示隐藏:
   // util.setInterval(function() {
   // if (ui != null) {
   // ui.show(!ui.isShow);
   // }
   // }, 1000);
   //设置定时器
   util.setInterval(function() {
          //请求参数
   var requestUrl= "http://192.168.10.67:9090/ajax/getMqData"; //
请求 url
   var mqurl="<u>tcp://192.168.137.138:8094</u>"; //ActiveMq 地址
   var username="admin"; // ActiveMq 用户名
```

```
var password="admin";//topic=1 queue=2
   var chanelMode="1"; // topic=1 queue=2
   var topicOrQueue="testOut123";//待接收 topic 或 queue
   var timeout="3000"; //设置收到下一条消息时间,单位 ms,默认 3000ms
   util.download({
        "url":
requestUrl+"?mqurl="+mqurl+"&username="+username+"&password="+pass
word+"&chanelMode="+chanelMode
            +"&topicOrQueue="+topicOrQueue+"&timeout="+timeout,
        "type":"text",
        "success": function(res) {
                   ////打印返回结果
                 console.log(res);
                             //json 转换为 table 类型
                var tableres = json.decode(res);
                 console.log("tableres : "+tableres);
                            // 取得自己需要的值
                var value=json.decode(tableres.data)[0].value;
                 console.log( "value:
"+json.decode(tableres.data)[0].value);
                   //把结果复制给场景中头顶牌中
```

```
CampusBuilder-用户手册 VO.7
```

调试结果如下图所示:

5.2.1.1 发布模模搭场景

代码调试完成后,点击下图1处发布按钮,复制2处地址可以在浏览器中访问,本机 地址是指调试场景的机器所在地址,局域网地址是该场景变化处于同一个局域网内的用户可 以访问



图 96. 发布模模搭场景



图 97. 接入 ActiveMq 数据源

5.2.2 ThingJS 调试环境接入

5.2.2.1 ThingJS 调试环境使用介绍

打开客户端,点击 ThingJS 调试进入 ThingJS 调试环境

🄰 CampusBuilder (模模搭) V1.2	2.1.204			Ϋ́	- 0	×
我的场景 场景集市 DI	IV模型库		➡新建 2	;立即同步 😑	🔵 хиуу	SUIP
场景名/场景ID	٩	/ 点击进入丁	hing」S调试环境			
€ xuyy1 @6 €	 ● 編輯场景 ● 預算场景 > ● 下載场景 > ● 「載坊易県 > ○ (2) 禰武场景 > ○ (2) 禰武场景 > ○ (2) 確認(1) ○ ○ (2) ○	0 • 0 0 • 0 0 • 0 0 • 0 0 • 2 0 • 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2	●1 € 0 ★ 0 ● 1 € 0 ★ 0 「 场界は 「 场界は 一 「 場合 一 一 一 一 一 一 一 一 一 一	¥	• 39313933 4:11 4:11 16:59:21	
• 望和桥 • 1 9	•0★0 ◎ 望和桥 ④	● 0 ● 0 ★ 0 ● 0 ● 0 ● 0 ● 0 ● 0 ● 0 ● 0	●0●0★0 點图种	类: 0/0		
			5/象灯 代码: : 远逝	三角面/贴图: 4/1096/6 0行	5	
• 望和桥 • 0 •	0★0	●1●0★0 ● 望和桥	●1●0★0			
	« 1 2	»				
			1 场景: 0/0 模型	型: 0/0 贴图	9: 0/0	

图 98. 进入 ThingJS 调试环境



选择右侧官方列表中的数据,点击第一项数据对接。

Lampusbullder-用户于册 VU.	isBuilder-用尸手册 VO.7
------------------------	---------------------



图 99. 数据对接

5.2.2.2 接入 MySQL 数据

进入调试环境,选择官方列表中的数据,选择数据对接,使用如下图所示的 Ajax 数据 对接(支持跨域请求)。

在1处填写接口地址,目前为CampusBuilder 官方提供,在2中的填写请求参数,请求 参数规则如表格所示

参数名	必选/可选	参数类型	参数说明	参数示例
username	必选	String	数据库用户名	root
password	必选	String	数据库密码	root
dburl	必选	String	数据库地址	jdbc:mysql://192.168.137.138:3306/outp utDB

sql	必选	String	查询 sql	SELECT id , sceneid, kpi_key,kpi_value
				from scene_data WHERE id = 1

请求成功后对于返回结果可以使用标准 JS 语法解析,获取需要的指标参数,使用 ThingJS 特有的 API 语法,赋值到场景对象中。



图 100. 接入 ActiveMq

调试代码示例:

/** * 说明:为模拟数据对接,ThingJS平台部署单独部署了一台服务器 3dmmd.cn, 模拟合作商的后端服务器 * 端口 82 为 websoket 接口,优势实时性好,每秒上传一个温度数值(-10 - 30 的随机数) * 端口 83 为 a jax 接口,优势服务器压力小,每次读数的时候返回一个温度数值 (-10 - 30 的随机数)
```
* 本例子演示如何做 ajax 和 websoket 数据对接
   */
  var app = new THING.App({
     url: 'models/storehouse'
  });
  // 界面组件
  var panel = new THING.widget.Panel({ name: '温度面板', hasTitle:
true, });
  var dataObj = {
     t1: '21°C',
     t2: '21°C',
  };
  panel.add(dataObj, 't1').name('水温').isChangeValue(true);
  panel.add(dataObj, 't2').name('气温').isChangeValue(true);
   // 接收服务端数据时触发事件
  update = function (evt) {
     // ***如设置服务器的 CORS,实现跨域访问。会更简单***
     // ***下例采用的不设置 CORS 的跨域实现方案***
     $.ajax({
        type: "get",
        url: "http://192.168.10.67:9090/ajax/getDbData",
```

```
data: {
```

```
"dburl":"jdbc:mysql://192.168.137.138:3306/outputDB?useUnicode=tru
e&characterEncoding=UTF-8", //mysql 地址
             "username":"root", //mysql 用户名
            "password":"root", //mysql 密码
             "sql":"select id, sceneid, kpi key, kpi value from
scene data where id=1",//设置收到下一条消息时间,单位 ms, 默认 3000ms
            },
         dataType: "jsonp",
         jsonpCallback: "callback",
         success: function (d) {
            console.log('拉取数据: ' +d);
             console.log('拉取数据: ' + JSON.stringify(d));
             console.log('拉取数据: ' + d.data[0].kpi value);
            // ***将获得的数据赋值到场景中***
            dataObj.t2 = d.data[0].kpi_value + '°C';
         }
      });
   };
   //每1秒更新一个数据
```

setInterval(update, 5000);

5.2.2.3 接入 ActiveMq 数据

进入调试环境,选择官方列表中的数据,选择数据对接,使用如下图所示的 a jax 数据 对接(支持跨域请求)。

目前对接 ActiveMq 支持使用 ajax 方式,在1处添加请求接口地址,2处对返回数据进行解析,接口格式定义如下

调试接口请求地址: http://192.168.10.67:9090/ajax/getMqData

如进行网络部署则为服务器实际 IP 及端口。

请求参数	必选/可选	参数类型	参数说明	参数示例
mqurl	必选	String	Mq 地址	tcp://192.168.137.138:8094
username	必选	Int	ActiveMq 用户名	9090
password	必选	String	ActiveMq 密码	ws
chanel Mode	必选	String	topic=1 queue=2	1
topicOrQueue	必须	String	订阅 topic 或 queue 名 字	testOut123
timeout	必须	String	设置收到下一条消息 时间,单位 ms,默认 3000ms	3000

请求数据示例:

[

```
{
    "ciCode": "P211-E1-01",
    "metric": "CPU使用率",
    "instance": "cup0",
    "value": "50",
    "timestamp": "1500790703007"
}
```

返回数据示例:

[{"kpi_value":"251","sceneid":"20170109153114611633853","kpi_key":"temperature","id":"

1"}]



图 101. 接入 MySQL

调试代码示例;

1、ajax 请求数据:

```
* 说明:为模拟数据对接,ThingJS平台部署单独部署了一台服务器 3dmmd.cn,
模拟合作商的后端服务器
   * 端口 82 为websoket 接口,优势实时性好,每秒上传一个温度数值(-10 - 30
的随机数)
   * 端口 83 为 a jax 接口,优势服务器压力小,每次读数的时候返回一个温度数值
(-10 - 30 的随机数)
   * 本例子演示如何做 ajax 和 websoket 数据对接
   **/
  var app = new THING.App({
     url: 'models/storehouse'
  });
  // 界面组件
  var panel = new THING.widget.Panel({ name: '温度面板', hasTitle:
true, });
  var dataObj = {
     t1: '21°C',
     t2: '21°C',
  };
  panel.add(dataObj, 't1').name('水温').isChangeValue(true);
  panel.add(dataObj, 't2').name('气温').isChangeValue(true);
```

```
// 接收服务端数据时触发事件
  update = function (evt) {
     // ***如设置服务器的 CORS,实现跨域访问。会更简单***
     // ***下例采用的不设置 CORS 的跨域实现方案***
     // 服务器返回的数据格式为
callback({"state":"success","data":29})
     $.ajax({
        type: "get",
        url: "http://192.168.10.67:9090/ajax/getMqData",
        data: {
           "mqurl":"tcp://192.168.137.138:8094", //mq地址
           "username":"admin", //ActiveMq 用户名
           "password":"admin", //ActiveMq 密码
           "chanelMode":"1", //topic=1 queue=2
           "topicOrQueue":"testOut123",// 订阅 tppic 名字
           "timeout":"3000",//设置收到下一条消息时间,单位 ms,默认
3000ms
           },
        dataType: "jsonp",
        jsonpCallback: "callback",
```

```
success: function (d) {
    console.log('拉取数据: ' +d);
    console.log('拉取数据: ' + JSON.stringify(d));
    console.log('拉取数据: ' +
JSON.parse(d.data)[0].value);
    // ***将获得的数据赋值到场景中***
    dataObj.t2 =JSON.parse(d.data)[0].value+ '°C';
    }
  };
  //每1秒更新一个数据
  setInterval(update, 6000);
```

2、 ws 请求数据:

/**

* 说明:为模拟数据对接,ThingJS平台部署单独部署了一台服务器 3dmmd.cn, 模拟合作商的后端服务器

* 端口 82 为 websoket 接口,优势实时性好,每秒上传一个温度数值(-10 - 30 的随机数)

* 端口 83 为 ajax 接口,优势服务器压力小,每次读数的时候返回一个温度数值 (-10 - 30 的随机数)

* 本例子演示如何做 ajax 和 websoket 数据对接

*/

```
var app = new THING.App({
     url: 'models/storehouse'
  });
   // 界面组件
  var panel = new THING.widget.Panel({ name: '温度面板', hasTitle:
true, });
  var dataObj = {
     t1: '21°C',
     t2: '21°C',
  };
  panel.add(dataObj, 't1').name('水温').isChangeValue(true);
  panel.add(dataObj, 't2').name('气温').isChangeValue(true);
   ******
   // 对接自有 websoket 服务器
  var ws = new WebSocket('ws://192.168.10.67:9090/ws/testOut789');
  // 建立 web socket 连接成功触发事件
  ws.onopen = function () {
     console.log("websoket 服务器连接成功...");
  };
   // 接收服务端数据时触发事件
  ws.onmessage = function (evt) {
```



5.2.2.4 发布 ThingJS 场景

代码调试完成后,点击下图 1 处演示按钮,2 处电脑演示会弹出新的浏览器窗口, 复制下图 3 处地址栏中的 url 地址给其他人就可以看到场景中的数据变化



图 102. 发布 ThingJS 项目



图 103. 演示地址

第6章 关于我们

北京优锘科技有限公司(英文名 uinnova Technology,以下简称"优锘科技")是 IT 可视化和数据可视化领域的领导厂商。模模搭是优锘科技从"让 IT 可视"走向"万物可视" 的重要一步,优锘科技相信未来的世界是"万物互联"到"万物可管"的世界,在新的世界 中,"万物可视"将是实现"万物可管"的关键一环。

如有疑问或想了解更多信息,欢迎访问优锘科技网站(www.uinnova.cn),或直接访问 ThingJS 官网(<u>www.ThingJS.com</u>)。

联系我们

公司地址:北京市朝阳区花家地北里北京国际画材中心5楼

电子邮箱: momoda@uinnova.com

客服电话: 400-051-0152

邮政编码: 100102