lugia产品软件用户手册

不用思考，我帮你想好了

目录

三、组件

四、检查器

4.文本

A.下拉选择

B.文字移动

C.文字编辑器

5.图标

6.移动

五、画板

1.参考线与对齐简述

2.参考线

3.对齐

4.窗体缩放

5.窗体移动

六、上边栏

1.返回项目

2.新建

3.保存

4.预览

5.model管理

6.项目管理

7.设置

A.物料源

B.项目默认配置

C.通用设置

D.关于

一、欢迎使用lugia

欢迎

欢迎使用lugia mega的用户手册，这个手册的内容同时适用于新手和熟练用户。
如果你对于这个手册的内容有任何想法和疑问，请通过邮箱xxx随时联系我们。我们会不断的完善本手册。

介绍

Lugia是一款应用于软件开发、、页面生成的工具。它包含了：Lugia产品、Lugia Design、Lugia Web、Lugiax和lugia mega。

Lugia产品包括一套标准的前端组件库、一套统一的模型管理框架、一套完整的工程构建工具链，以及一套高效并且开箱即用的前端可视化开发工具。

Lugia Design 是一套适合于中后台前端应用企业使用的设计语言。

Lugia Design是一套基于 Lugia Design 的高品质 React 组件库，满足金融行业高性能组件的需求，适用于中后台的web企业应用开发。

LugiaX一个基于 Redux 的前端状态管理工具。提供简单高效的全局状态管理方案、 基于 async/await 的异步操作、快捷的双向绑定。

Lugia Mega 是标准、高效、开箱即用的前端可视化开发工具。无需环境搭建、快速上手的跨平台桌面应用（Mac 和 Windows）。

它的优势在于能够节省项目开发时间，简化开发流程、规划版本记录。让产品、交互、设计、前端、后端合为一个整体。

这不仅对于我们是一个挑战，对于整个行业来讲都是一项艰巨的任务。我们经过一年的努力让lugia在最初的版本不仅完成所有的实用性功能，而且变的最容易理解并上手简单。

为什么选择我们

lugia的终极目的是解决当前行业产品线上人员的杂糅。基于这款产品，产品交互，可以在应用上以简单的拖动就可以拼凑出一套完整的界面原型图。UI设计师可以通过交互模型上的原型图直接调整页面颜色样式字体。而对于研发，仅需要根据设计图作出相应的捆绑调试，就可以直接生成一套完整的项目流程。这样以来，不仅减少了项目的人力成本，还节约了项目时间。

总体来说Lugia产品包括一套标准的前端组件库、一套统一的模型管理框架、一套完整的工程构建工具链，以及一套高效并且开箱即用的前端可视化开发工具。本产品具有无需环境搭建、快速上手，跨平台支持等优点，其贯穿了整个项目的原型设计、效果设计、软件开发测试阶段，为用户极速构建与轻松管理高质量前端项目赋能。支持交互设计师快速迭代原型界面，支持体验设计师灵活迭代效果界面，支持软件开发设计师简单迭代最终界面。最终完成前端项目管理模式重构，使得三者产出成果可以互相传递复用，进而满足云原生大前端的快速迭代开发需求。

二、界面

创建项目

Lugia mega是以项目为基础，页面赋予在本项目中。

这也就意味着，若您开始创建操作时，是先创建一个项目，在有项目的基础上在去创建属于这个项目的各个页面。

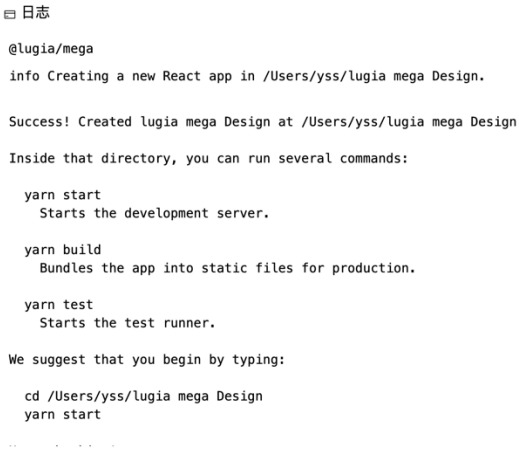
编辑创建

创建项目

当您点击创建新项目时，页面会弹出创建一个项目用户添加的项目信息。我们为了节省您的效率，从而将一些高级设置自定义为默认设置，这样您只需要编写项目名称选择自己所需的页面尺寸就可生成一个真实的项目文件。

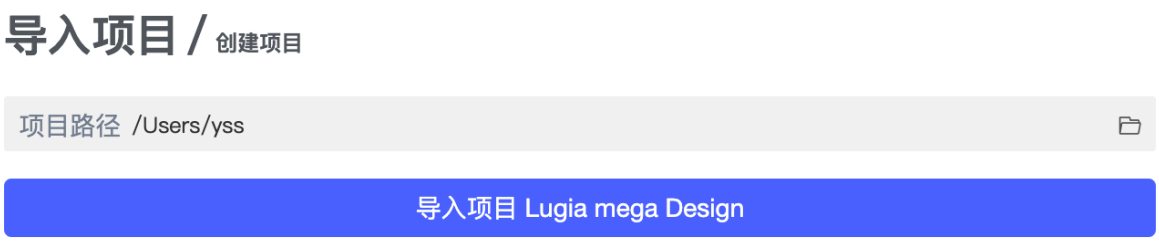
当您录入完项目名称后，点击创建会就会出现一正段的项目日志，在日志项加载完成后创建按钮会变为“打开项目”点击后即可使用。

值得一提的是，若出现一些异常情况，例如：“在创建项目时网络出现问题，或输入项目名称未符合开发规范，创建时可能会出现一些报错信息，这些报错信息在日志中可以查看”。Lugia工程师们已经在努力的避免所有报错的产生，若有影响您使用的情况，还请您与我们联系。我们会以最大的能力解决您的需求。



导入项目

如果您已经有自己本地的lugia项目，您可以点击上方的导入项目切换卡，选择本地的项目路径来启用该项目。



高级设置

高级设置可以精确您的项目明细，您可以在此设定您当前项目的版本，以及对项目的基本描述。

面板

lugia的界面采用简洁化的设计样式。最顶端的工具箱包含了最重要的操作。左侧是应用组件部分，可以在该区域选择您所需要的组件形态在画布中进行编辑。右侧则是对于组件的属性的动态参数调整，不仅满足于设计的静态页面颜色，还可以去修改一个组件的手势状态或者去增加一些双重绑定事件。最后中间的区域就是你正在创作的画布。

Lugia里没有浮动面板，检查器将会根据你选中的组件来显示所选单位，这样你能始终不受打扰的在画布上创作。

工具栏

lugia的工具栏相当于整个项目的控制台，可以在上方对整体的项目进行管理、保存、导出等其他调试服务。

在工具栏中，第一组工具是用来对页面进行基础设置的。其中包含：新建、保存、预览和画布。第二组是对页面的管理。包含：model管理、项目管理、设置和响应式等功能。第三组就是对完成编辑的页面打包构建，既：导出。

左边栏

我们认为一个页面是由4个元素拼合构成的。即：布局、区块、组件和图标。为了方便您更易上手，在左边栏中我们将4个元素按照从左到右、从大到小的方式分类排序。并全部用图像加文案的方式展示出来。

组件面板

Lugia基于对市场上产品的研究，将产品项目单元化。把一个页面所需要的所有组件编译成开源的控件，这样您在也不用去思考一些无意义的形状去构想页面，而是直接通过一些组件的按钮去摆放拼合成一个整体。

为了方便您查找，我们对组件做了两种分类查询您可以直接搜索想要的组件，也可以通过我们的分组整理进行筛选。同样，组件在画板上您也可以通过检查器自定义您想要的属性。

自定义组件

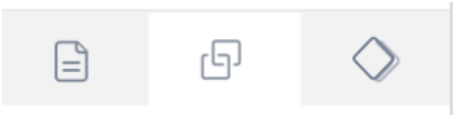
当lugia官方组件不能满足您的需求时，您可以尝试我们的自定义组件功能，可通过左边栏点击自定义组件区域，新增一个适合您的组件。拖拽后放置页面中。

当您需要对自定义组件进行编辑时，您可以点击上边栏中的查看，在您的代码编辑器中打开，您就可以对该组件进行编辑修改了

微应用

当您通过导出创建微应用后，您可以在左边栏中找到您所创建的微应用组件，并在页面中使用，具体创建流程我们会在下文“导出-微应用”流程中找到相关说明文档。

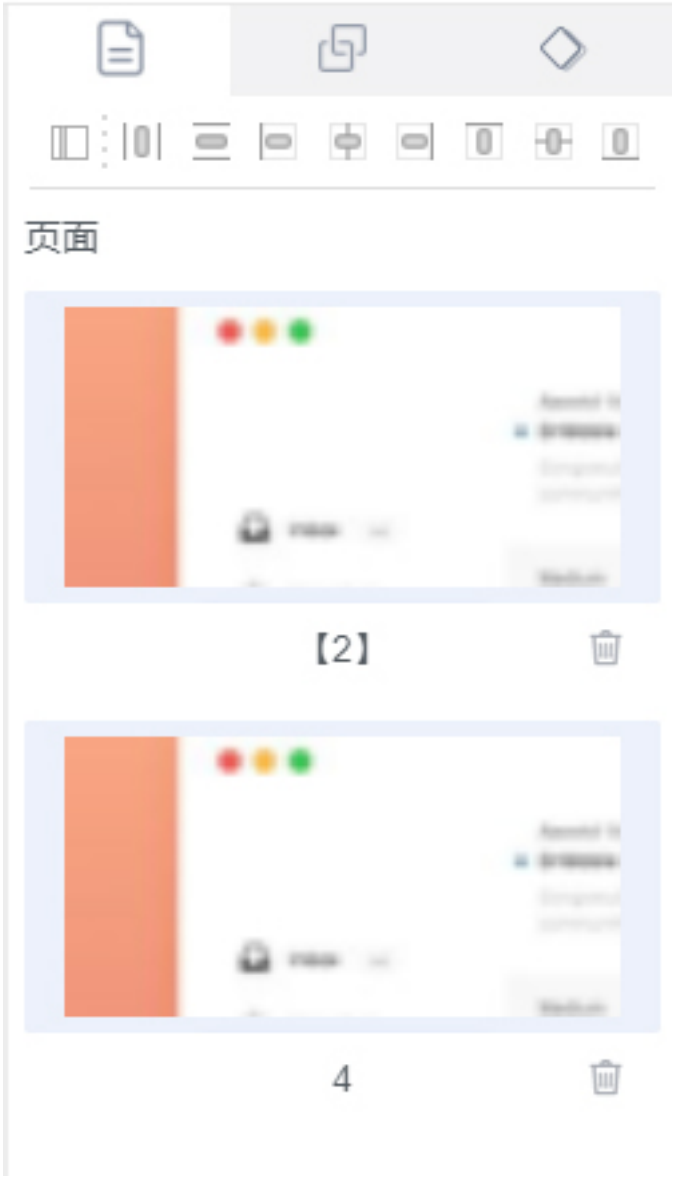
检查器面板



右侧的检查器能让您编辑当前使用的组件属性，进行参数的调整。为了让您更加编辑，当您选择一个组件时，您就会发现我们已经将检查器划分几个区域：“页面区、属性区和事件与模型编辑区域”

页面

在页面区域您可以对您当前的画板进行一个整体的统筹布局。 画板页面是为了让您可以更自由的切换其他应用项目。



属性

在属性区域您可以对当前选择的一个或多个组件进行属性的调整，该区域是一个初级用户需要基础学习才可灵活掌握的页面。在此区域您可以自定义您当前编辑组件的外观。

也可以调整页面中的属性配置

调整页面中整体的布局。

事件与模型

我们为了满足开箱即用的市场需求，在事件与模型页面中可以针对不同的组件进行一系列捆绑。最显著的特点就是响应式和数据驱动，也就是将Model和View进行单向绑定或者双向绑定。

三、组件

组件是lugia中最基本的构成单位。它不同于其他设计产品的图层和对象。我们是将组件通过开源封装成一个个子元素。做为使用者，您在设计页面中思路模式需要从适应的以形状为基本元素去拼接来完整整个页面设计稿，转变成直接思考我需要的页面设计稿需要哪些组件。

例如："当我设计一个简易图表查询功能。常规的设计软件需要自己通过矩形-转变成描边-设置宽高。在生成一个矩形以改变颜色。添加"查询"文本。而通过lugia仅需要思考这是一个"查询"功能，我需要一个图标和一个按钮就可以了"

这种思路上的转变给您带来的改变是不需在绞尽脑汁去做一些无意义的形状，而是直接去思考页面的全局应该有哪些变化。

创建组件

上文已经讲到，在您使用lugia时可以通过左边栏去选择您所需要的组件。您只需要鼠标点击选择你需要的组件，然后拖动鼠标到所需要的区域位置。松开鼠标，即可完成该组件，并开始编辑它。

选择组件

当组件拖拽到页面后，您如果想对它进行调整。您只需要单机来选中组件，这时选择框的四角和边框上会同时出现8个小锚点。这些小锚点会给予您一个提醒："该组件已经被选中"。

shift+option（shift+alt）：组件会以 组件中心为居中点 进行等比例放大缩小。

同时选择多个组件

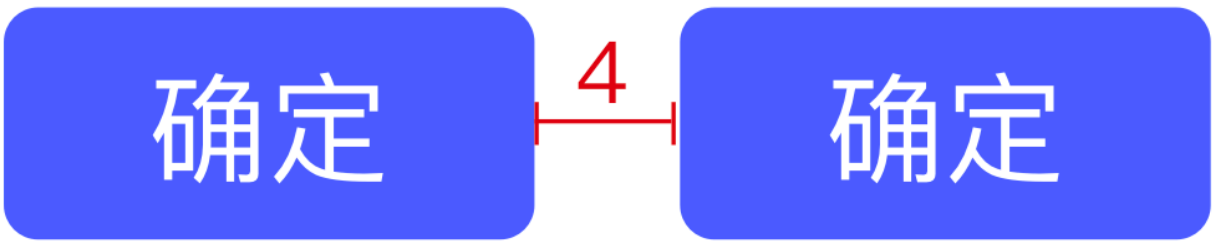
您可以通过鼠标在画板上框选住多个组件，如果您需要单独选择。可以按住键盘上的shift键来同时选择多个组件。同样如果您多选了不需要的组件，您也可以通过shift单机来取消该选择。

值得注意的是，在lugia中涉及到了集合页面的概念（类似于成组）。当您选择一个集合页面的布局，该页面是以父容器为最高选择的。（简单的说，假如有一个盒子和一些组件，把其中的几个组件放到盒子里面。如果我选择了这个盒子和某一个在盒子外的组件，那么盒子内的组件也会被选中。因为它已经在盒子里面了）

移动与复制组件

您可以选择任意组件，并通过拖动鼠标或者键盘上的方向键来移动它们。如果您觉得移动的不够舒适，则可以尝试按住键盘上的shift键来移动它们，这样就会让图层严格按照垂直或者水平方向移动。

Lugia为了满足移动时候的对齐问题。设置了移动状态下防治抖动的功能，当您移动某一个组件时候，lugia会自动帮您将它与相邻的组件根据四周变化做锚点吸附功能。并提供参考线作为吸附依据。



如果您想复制出一个相同的组件，您可以鼠标点击组件并按住使用键盘上的option（alt）键，拖拽到相应的位置即可。当然如果您仅想在原位复制一个组件。可以试试command+c(ctrl+c)进行复制，在按下command+v(ctrl+v)复制即可。

放大缩小组件

如果您想调试组件的尺寸，可以去使用组件的8个小锚点。直接将鼠标滑动到某一个锚点上，然后按住放大缩小就可以感受到该组件带来的变化，这样就省去了在检查器输数值才可得到的状态。而且对齐方式和移动组件一样。我们同样设置了对齐功能。当您放大一个组件与相邻组件某一个属性相邻时，会触发标尺来示意对齐，并带有吸附功能。具体标尺的对齐细节，等后面标尺章节在详细说明。

值得注意的是，如果您习惯了小锚点拖拽操作之后，可以通过一些快捷键，来让您的调整更为精准。

Shift：如果您按住shift键（放大缩小）组件 会锁定组件的宽高 进行等比例变化。

option（alt）：组件会以组件中心为居中点，进行放大缩小。

shift+option(shift+alt)：组件会以 组件中心为居中点 进行等比例放大缩小。

删除与撤销

如果您对拖拽进来的组件感到不满意，您可以试试删除它。您只需要点击或框选一个或多个组件，然后按下键盘上的退格键，您就可以删除它了。

我们都知道在设计页面中会有很多的调试，我们当然知道如果您调试了很久的组件不小心被误删除了该怎么去找回来。lugia mega为您提供了强大的撤销功能。您只需要按键盘上的command+z（ctrl+z）即可复原刚才撤销的项。

编辑组件

如果您需要对一个组件进行更准确的调整，您就需要借助右侧的检查器了。而具体变化成什么形状就取决于您想要比那集的是什么类型的组件了。

和常规的设计软件不同的事，您使用lugia生成的页面不仅仅是平时一个设计师交付的设计稿。而是一个经过代码编译后的页面。所以您的工作不仅仅是完成一个静态页面的调整。您在编辑组件的时候，还要考虑到所编辑的组件鼠标滑过、点击、禁止时候的样式。以及我打开一个下拉菜单，究竟里面有多少字段，多少行选项提供给使用者。又或者，如果您需要对该组件与其他组件，产生一些关系，您还需要通过事件窗口为它们创造一些事件的绑定。

致于检查器有太多的功能。在后面的章节“检查器的功能篇”我们会一一说明。

四、检查器

关于检查器的属性

检查器会显示所选组件的一切属性调试。

用户可以基于自己的喜好在检查器面板进行属性样式的调试。我们对组件所调试的样式种类进行了整理。您可以去配置页面布局的距离，也可以调整基础的宽高以及组件的属性
当然在某些特定的组件上也有一些专属的修改项。

例如：当我们应用滑动滚动条时候，我们要去思考滑动条有多少个节点。

值得注意的是，每个组件是有不同的区域的，我们为了方便您操作，将区域进行外观和属性配置的细分。（以滑动滚动条为例：“滑动滚动条分为滑动条和提示框两个部分，当您编辑该组件时，需要手动切换自定义区域的滑动条和提示框切换选项卡。以便进行对修改）

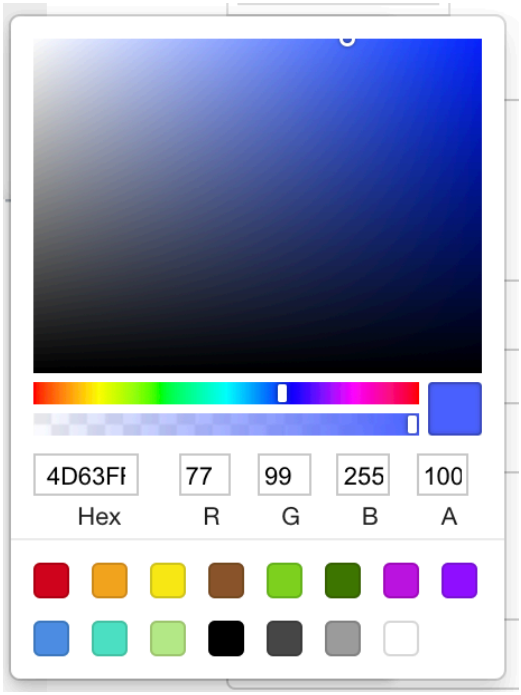
外观

想要改变一个组件的外观？那您可以尝试一下外观界面的参数调整。

组件的外观您可以显而易见的找到它。目前在外观项支持的功能包括：“调整，组件的尺寸和颜色”

尺寸属性目前包含了：“宽和高”。您可以通过它们去修改组件的宽高,也可以当作一个查看工具,能够让您更清楚的知道宽高属性的具体数值。

最后一项为颜色调整。您可以在此去设定组件的颜色。

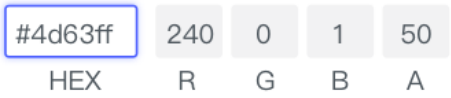


颜色编辑器

颜色编辑器是在颜色窗口点击进入的。在颜色面板有几种调试参数。

颜色识别区：可以直接选取颜色调整颜色属性。

滑动控件：可以选取颜色的范围和透明度。

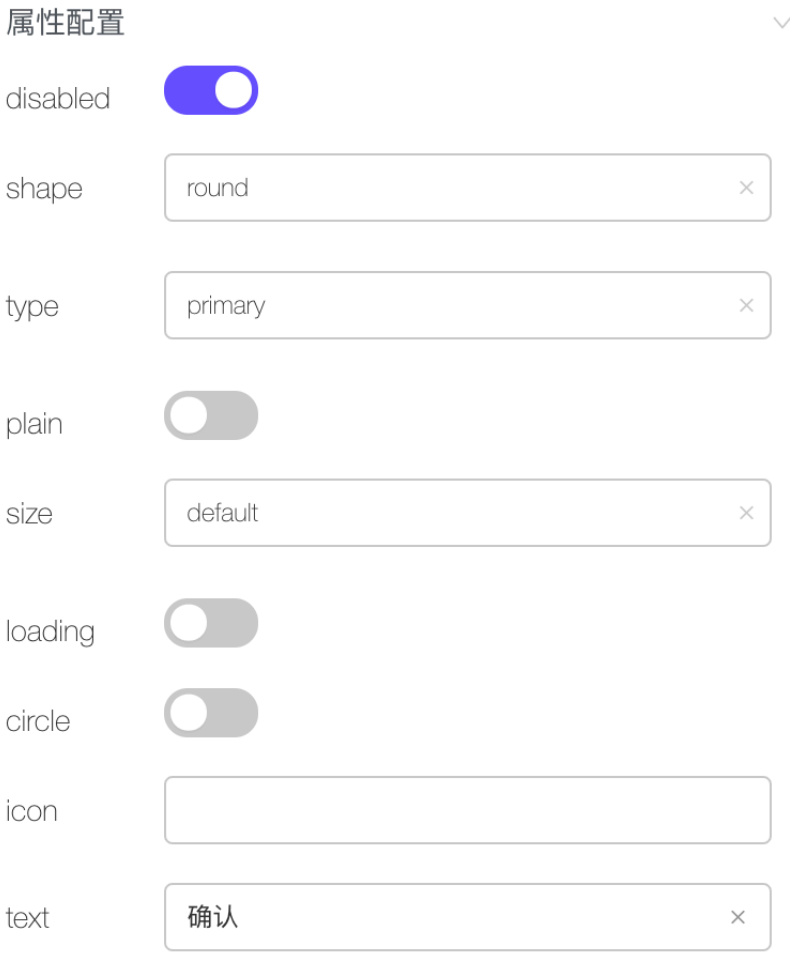


数据输入区：在这里可直接对参数进行调整。目前仅支持RGBA的颜色格式

属性配置

lugia是以一个组件为基本元素，我们基于react开源的组件库将组件的适合的属性进行整理并将它开放出来。当然因为组件的不同，具体所能调整的组件属性也会根据组件相关联。

右侧属性面板是以“按钮button”为例。



Disabled(禁用状态)

可以调整一个组件的状态，当我们在页面中设计一些按钮，在某些情况下会呈现出禁用状态的参数，您可以通过点击开启disable来尝试将它的样式变为禁用。

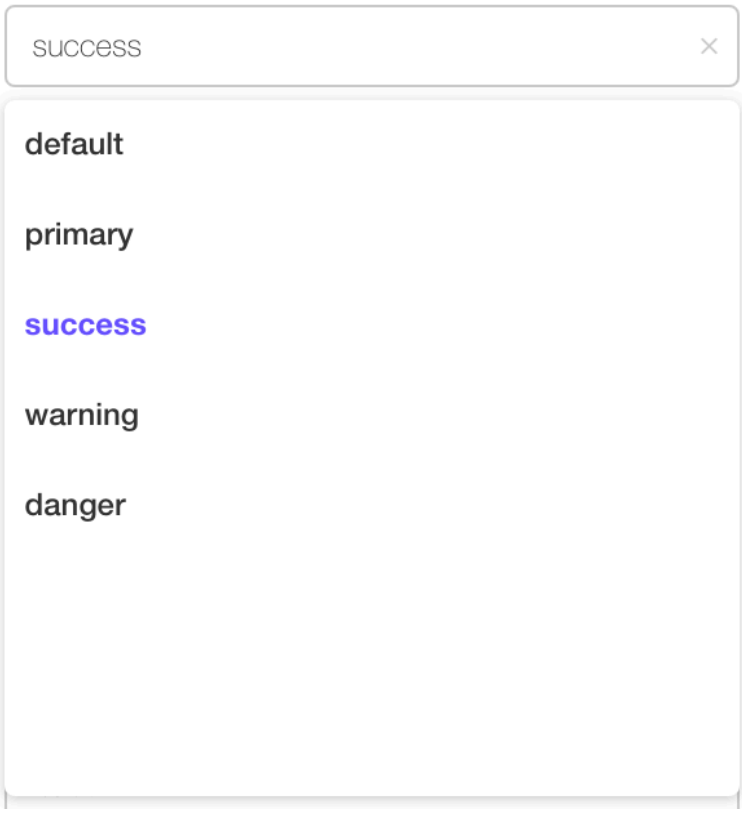
Shape(圆角)

当您开始您的设计图的时候，如果没有一个合适的圆角编辑器，恐怕不能让您满足您的需求。shape可以让您的组件变得与众不同。

目前shape在针对按钮组件开放了两种属性：矩形样式和全圆角样式。您可以根据自己的需求，单机选择所需要的样式就可以了。

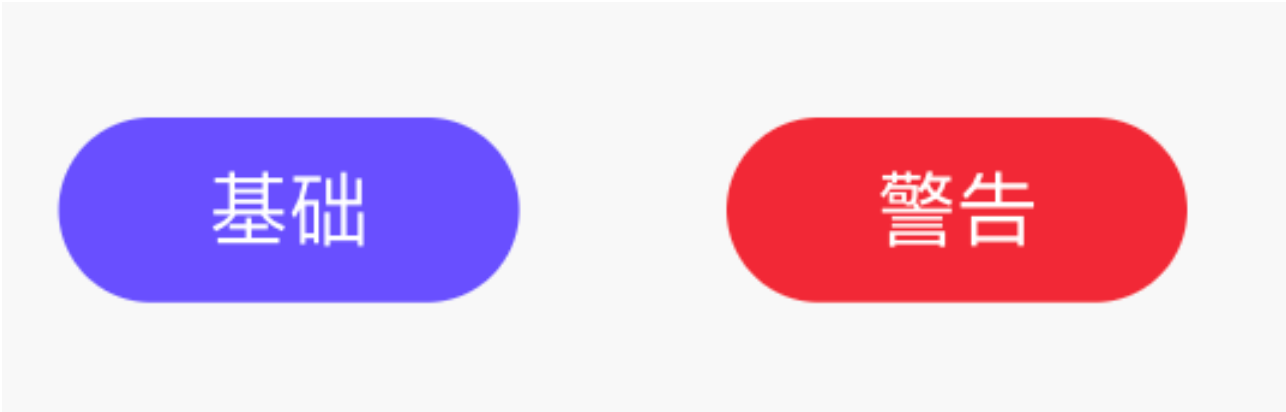
Type（类型）

这个功能可能您在其他类似的设计器中并不知道它的具体作用，type的主要功能是可以让您选择您组件属于什么类型。以按钮组件为例。



您可以type中选择您组件的类型，目前包含的类型：“default、primary、success、warning和danger(默认、基础、成功、警告和危险)”

当您在其中选择一种类型后，
组件会根据您所选择的样式去转变它的形态。
下方图例为：“按钮的基础和警告样式”。

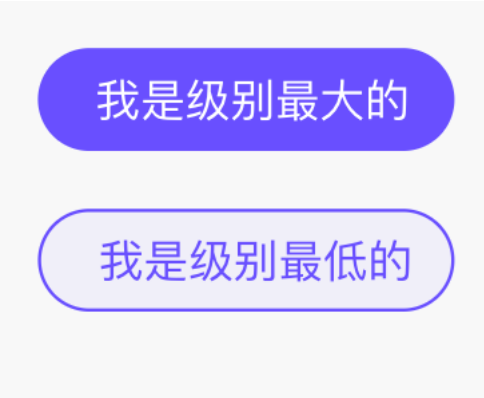


Plain(朴素按钮)

在一个设计师构建页面的时候，对于某些组件会有一个优先级的排列。而朴素按钮的作用就在于您可以通过打开它以降低按钮的重量级。

当然这个级别仅限于表达在样式上。

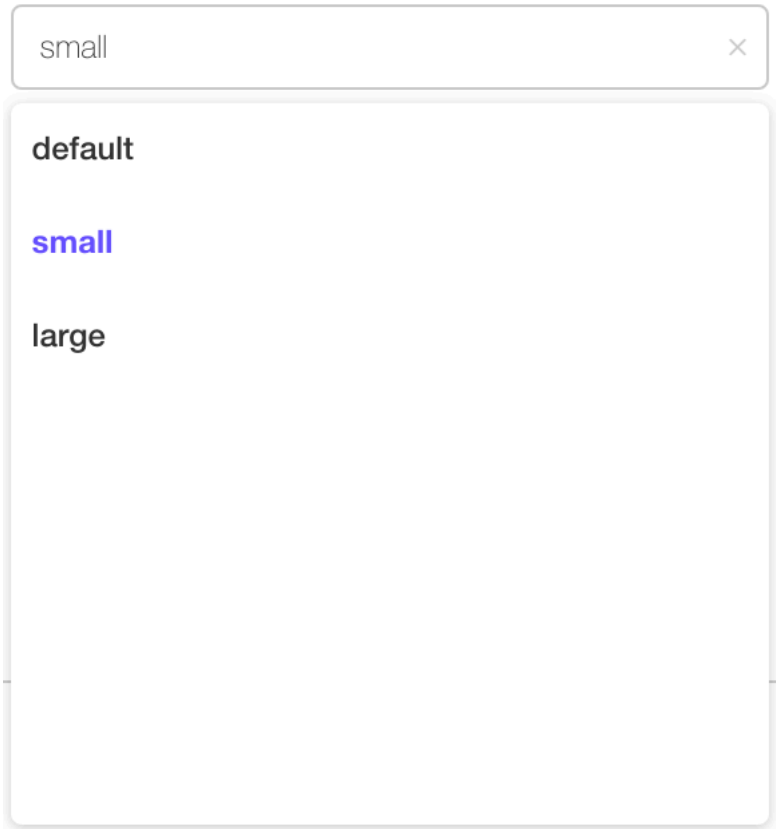
图示为两个同样按钮打开plain之后的状态。



size（尺寸）

我们为您开放了每个组件的几种适配尺寸。

以组件为例：提供了3种尺寸（大、中、小）提供选择。



loading（加载）

在lugia中为组件提供了加载功能，您可以把您的组件提供数据加载功能。
您只需单击勾选加载的选项，就可以看到组件的加载样式。

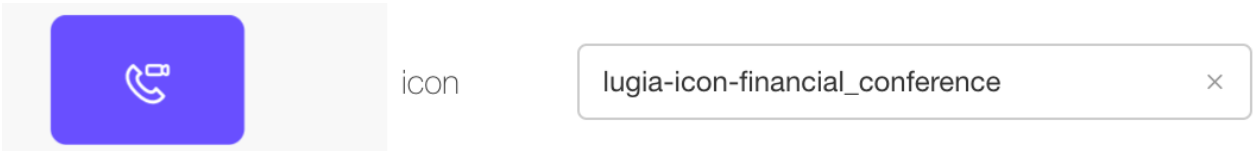
Circle（圆形）

circle的功能是改变您的组件形状为圆形图标形式。当用鼠标点击后组件会自动转换。

以按钮组件为例：点击后变为按钮状。

icon（图标）

您可以在我们的lugia官网图标组件中，选择一个图标。从而复制他的icons名称，然后填入到检查器icons的输入框中。您就可以观察到组件的图标添加样式。
图片以 “lugia-icon-financial_conference”传入为例：



text（文本）

您可以在组件中编辑您的文本，并实施反馈在组件中。
以按钮组件为例：

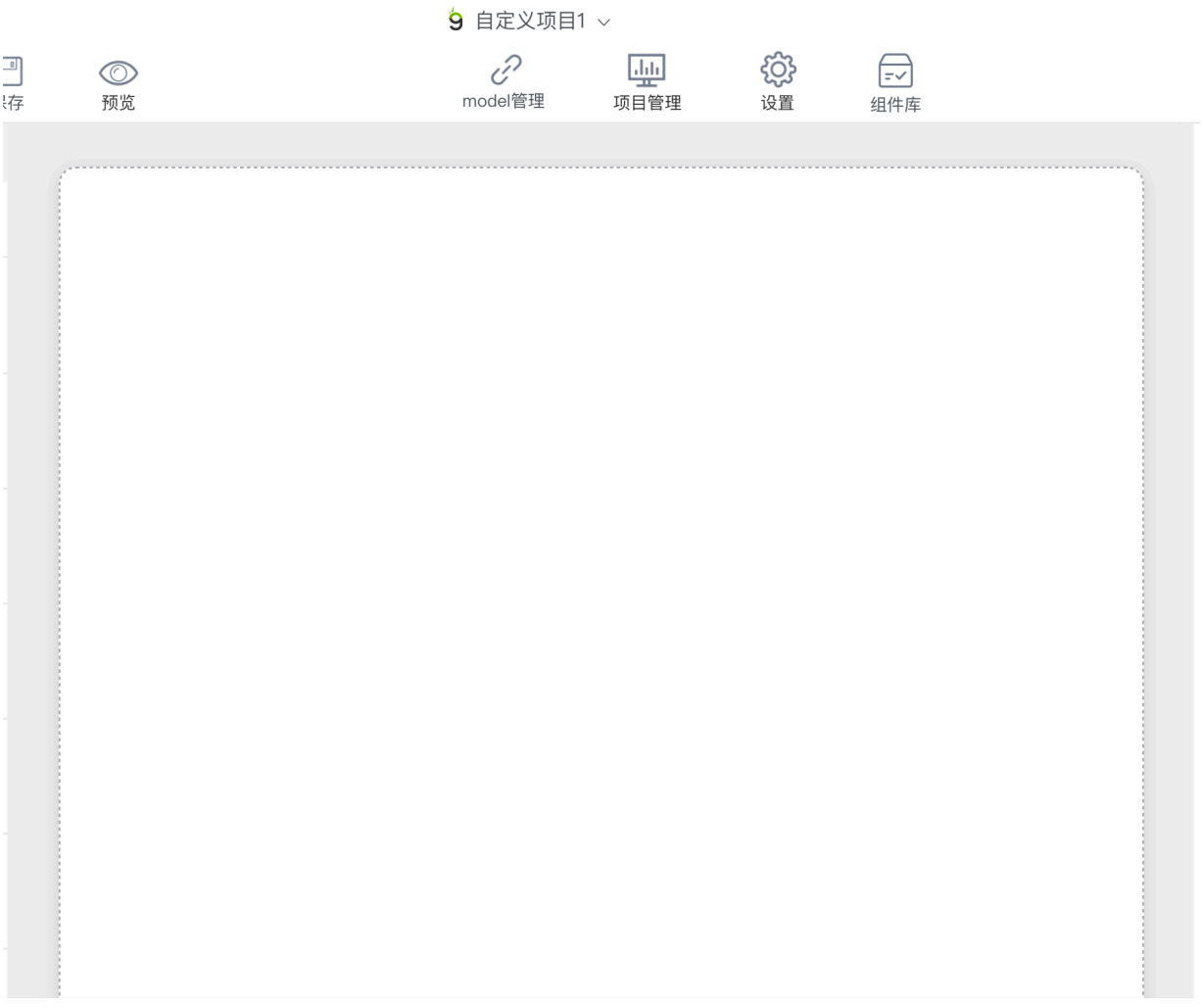
值得一提的是，lugia采用公共字体系统。暂不支持自定义字体上传（我们在稍后的版本中会将自主上传的字体进行开发）。

五、画板

Lugia画板是由两个部分组成，草稿层和主画板。两个部分是在无限画布中的一块固定的画框。当您在文件中创建一个新的画板时，画板之外的部分会将颜色变暗，这样您可以更清晰的看到画板草稿与背景之间的区别。

如果您要进行页面设计，您需要提前预设好您要设计的画面尺寸，注意只有在主画板中才是您最终生成的页面，而草稿层则是为了让您临时不需要的组件有一个可防治的区域。

我们的主画板会有一点像一个特殊的组，它们永远是开放的状态，您不需要点击或双击来查看主画板上的内容。值得注意的是，画板的尺寸也不会因为您添加的内容的数量而自动伸缩。您在预设中设置的画面尺寸，这个尺寸会一直保留，除非您再次在设置中更改它。



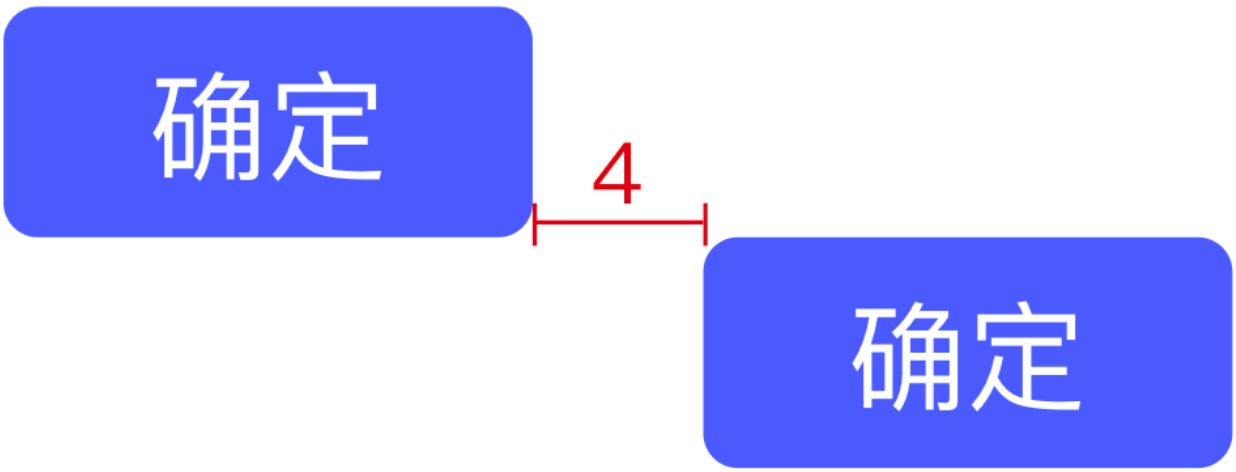
参考线与对齐简述

以上三个功能是为了让您做一些组件拖动位置摆放时变得更加清晰，准确。减少您拖动时候的负担。

参考线

参考线在lugia中默认设置是被打开的。

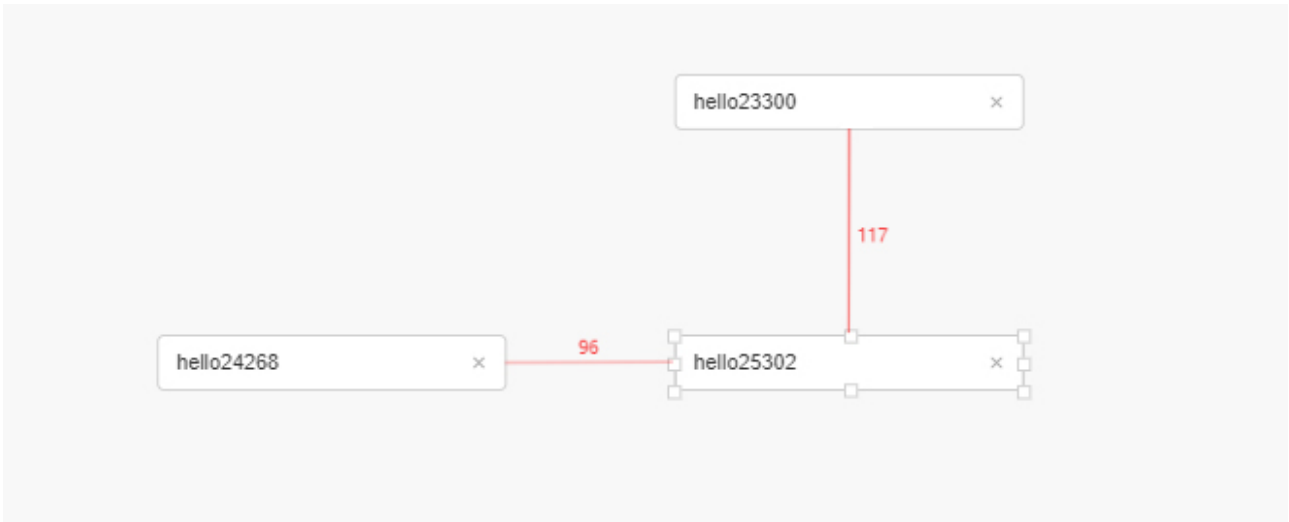
当你在调节一个图层的大小或者移动一个图层的位置时，lugia标尺系统会用自带的吸附功能自动帮你把这个组件与其他临近组件的基点（8个锚点）进行吸附。



如果lugia将某一组件自动与另一图层对齐，你会看见一条红线，两个图层便依据这条红线对齐。

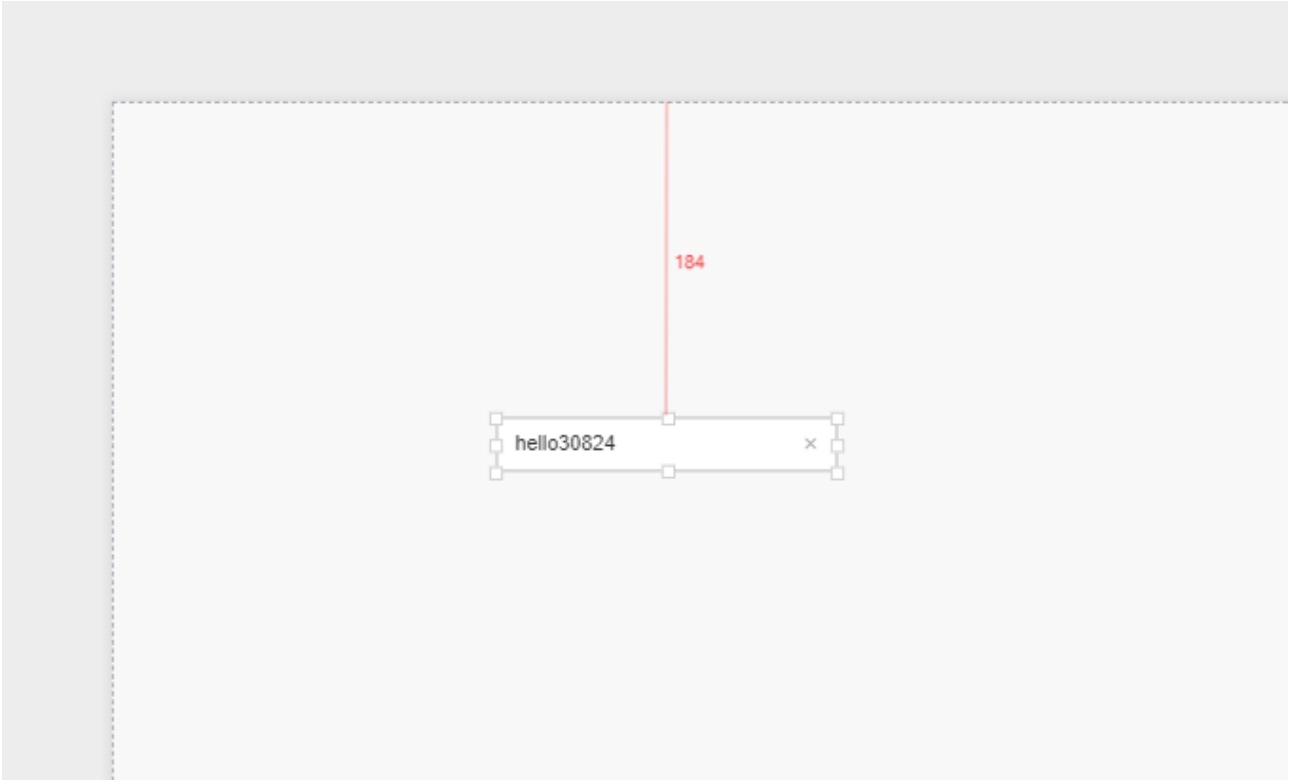
当参考线出现时，会出现一些特定的规则，例如："若当您在画布上需要将两个相同宽高的组件进行对齐，那么则仅显示组件的中心对齐点，且中心对齐点的优先级永远高于其他锚点。"

还有一种更特殊的情况："当画布上对3个组件进临近对齐时，若中线对齐则永远以中线为主，若不对齐则看其他锚点哪个为对齐点。



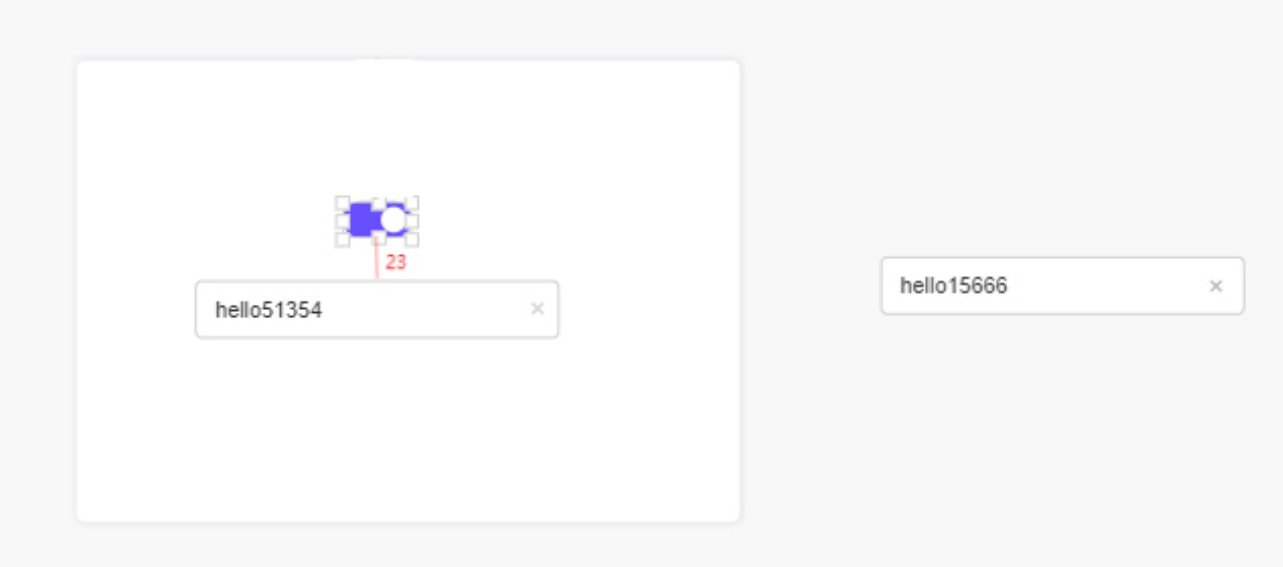
值得一说的是，lugia所附带的参考线不光可以作为您拖动组件时候的准确点，还可以作为一种测量工具。您可以尝试选中一个组件后，按住键盘上的option(alt)键，然后将鼠标滑动到其他的组件上。这时您会看到两个组件之间的距离的具体参数。

另一个值得注意的是，我们设计页面时候，经常会需要知道组件与页面边界的距离。lugia为您研发了组件边界参考线。您只需要拖动一个组件到临机的边界，您就会看到组件到达边界的距离。（该距离以边界的距离）



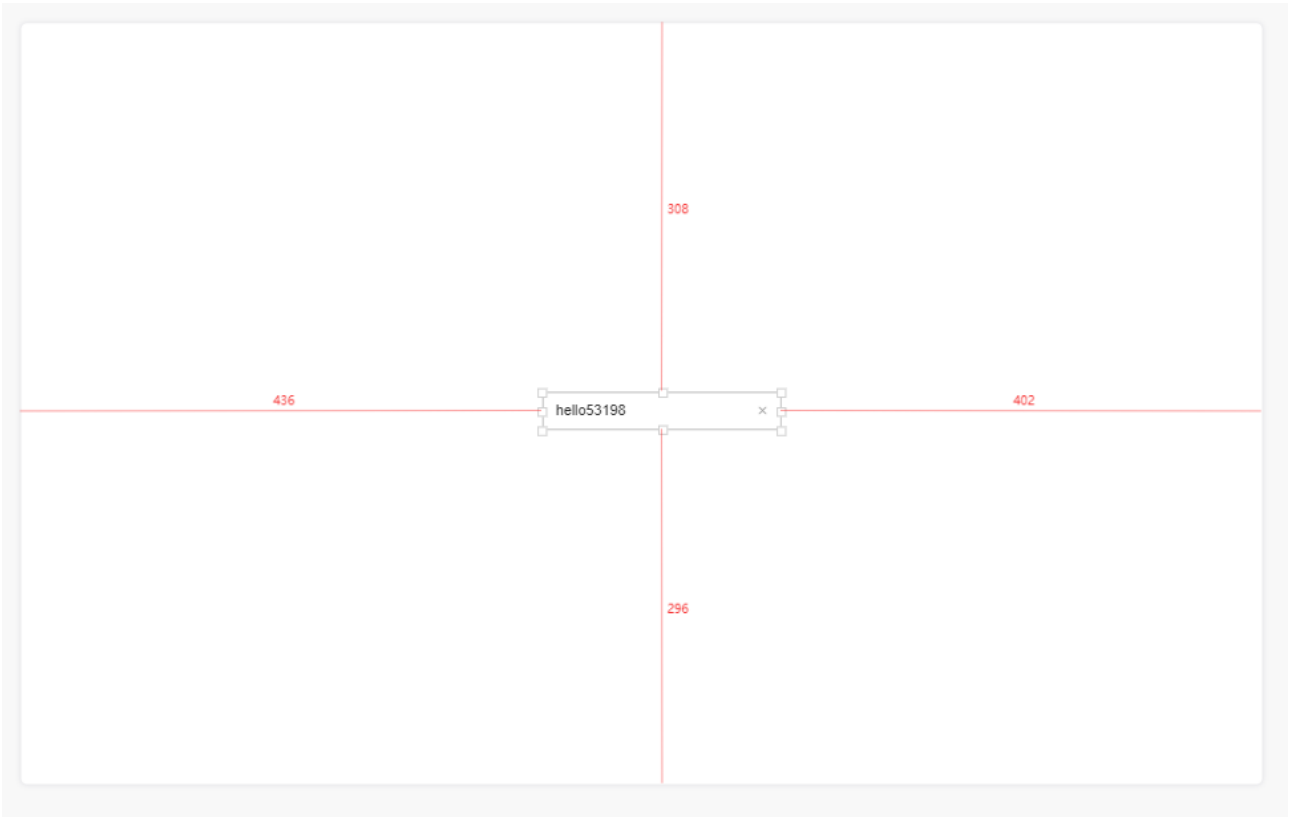
lugia对组件进行分组会有一个容器的概念。在稍后的章节里面会着重讲解父容器的使用。这里仅说父容器参考线与常规样式的区别。

当您拖动几个组件到父容器时，父容器里面的组件仅会以父容器内部的组件之间产生标尺，不会影响到外部的组件。



当然父容器也是一个盒子，使用上面的option（alt）测量距离的功能，您也可以测量组件到父容器的距离。

这时就有一个麻烦的问题如果当您的组件已经在父容器内部，那么在此去测量副容器，那么则显示该组件在父容器的所有距离。



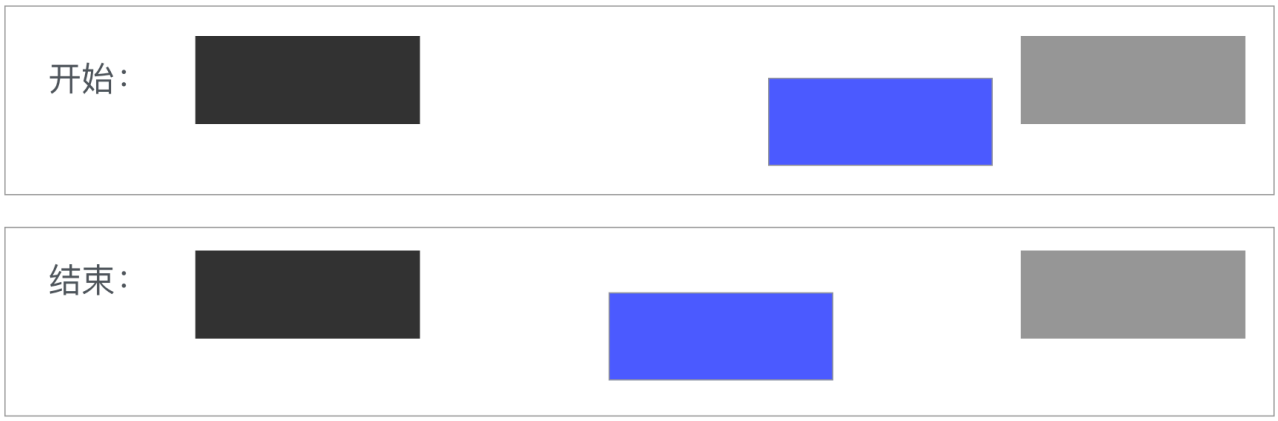
对齐

我们知道如果构建一个组件对齐原则最快捷的办法并不是通过标尺去标记，也不是需要靠辅助线一点点的去测量。而是建立一套完整的对齐系统。

我们为了让您对组件的把控更加灵活，在右侧检查器的上方会看到一排对齐图标。

横向居中对齐（必须组件≥3块才可应用，若<3块则至灰）

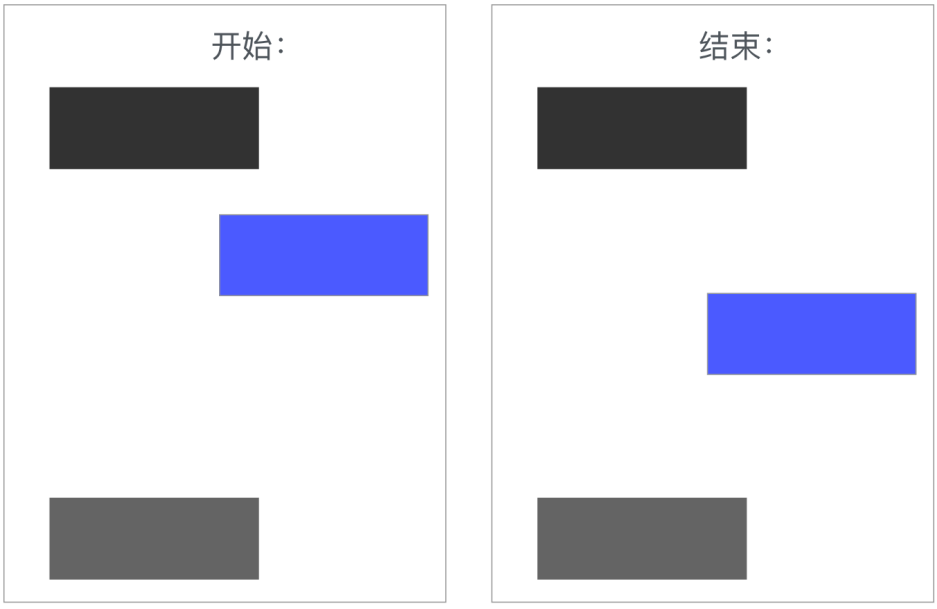
规则：两个组件以前后组件坐标为基准，中间的所有组件，横向居中对齐。
示意图如下：



纵向居中对齐（必须组件≥3块才可应用，若<3块则至灰）

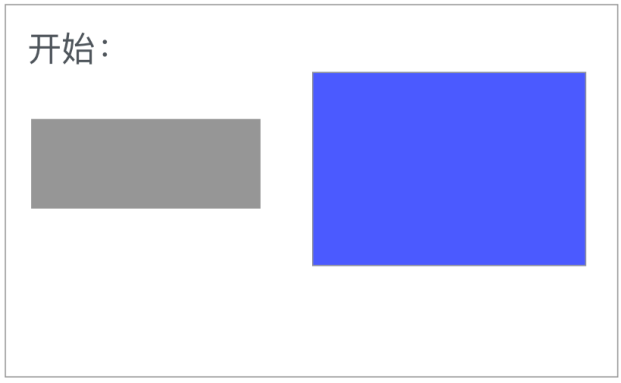
规则：两个组件以前后组件坐标为基准，中间的所有组件，纵向向居中对齐。

示意图如右：



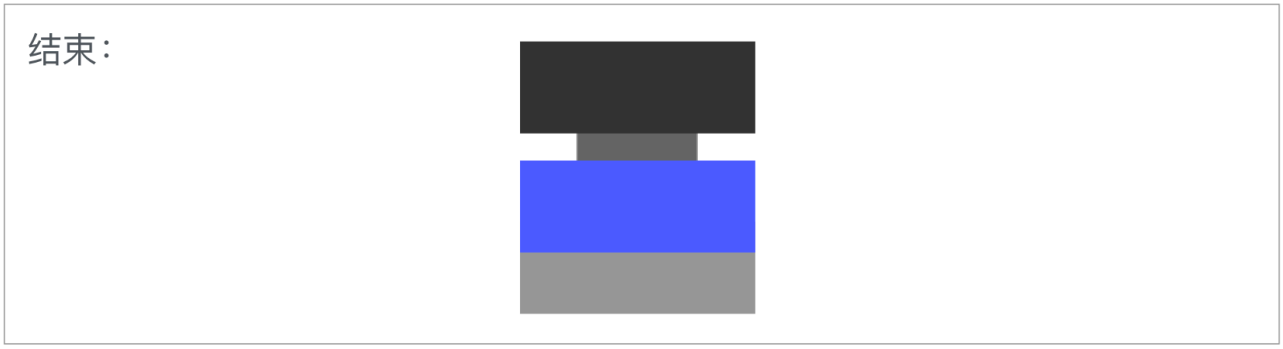
左侧推进对齐（必须组件≥2块才可应用，若<2块则至灰）

规则：以所选组件左侧位置为基准垂直平行移动
示意图如下：



垂直居中对齐（必须组件≥2块才可应用，若<2块则至灰）

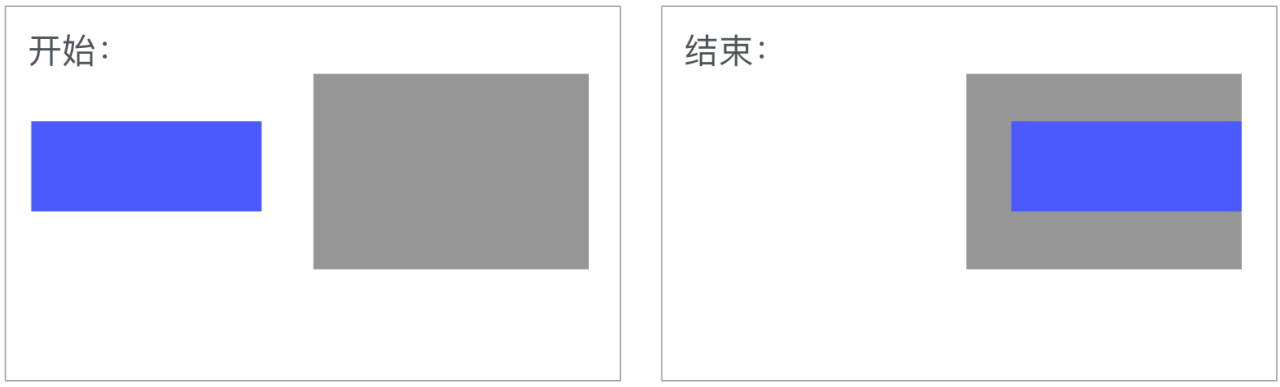
规则：以所选组件中垂线位置 做居中对齐。示意图如下：

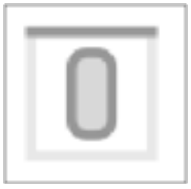


右侧推进对齐（必须组件≥2块才可应用，若<2块则至灰）

规则：以所选组件右侧位置为基准垂直平行移动

示意图如下：

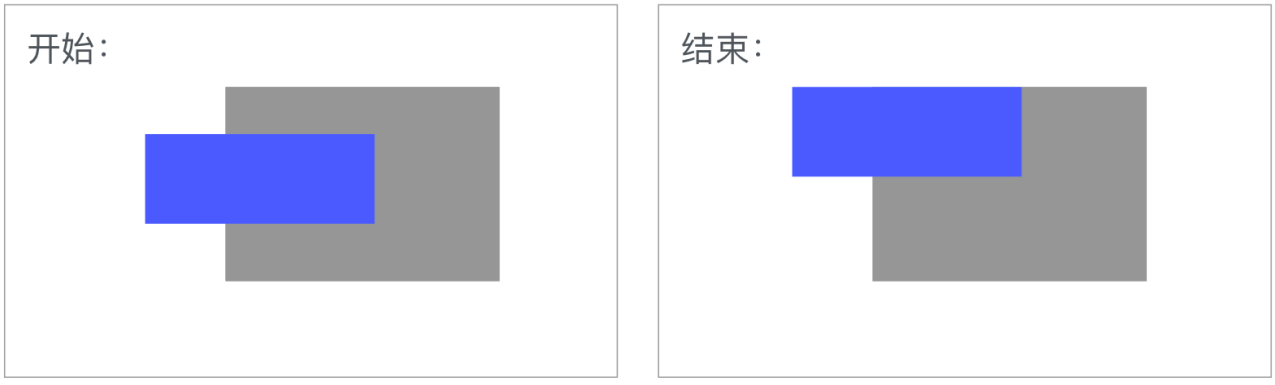




顶部推进对齐（必须组件≥2块才可应用，若<2块则至灰）

规则：以所选组件顶部位置为基准垂直移动

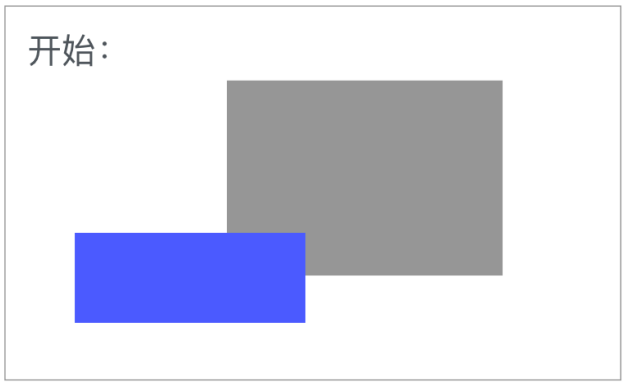
示意图如下：

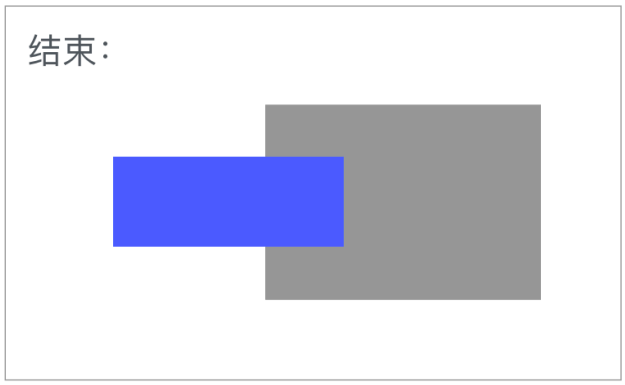


居中对齐（必须组件≥2块才可应用，若<2块则至灰）

规则：以所选组件中线为基准做居中对齐，所有组件根据此基准做向上或向下移动

示意图如下：

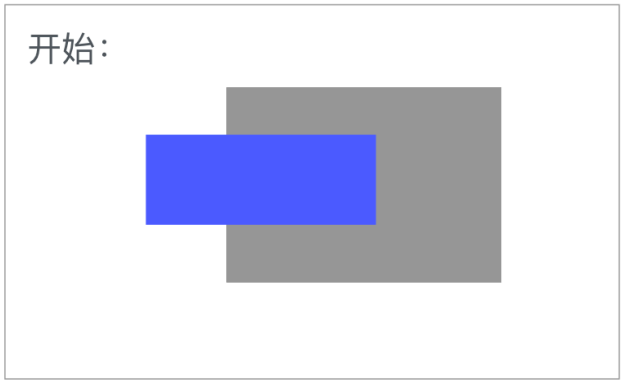


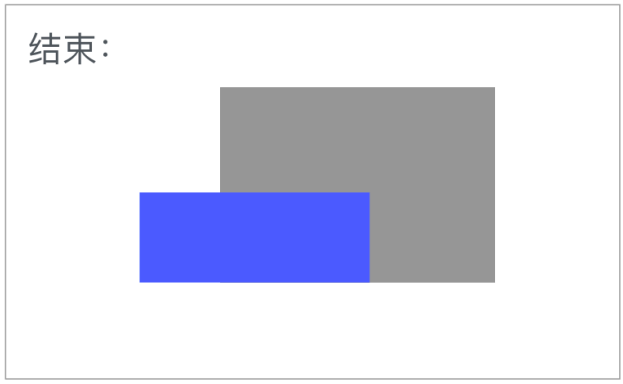


底部推进对齐（必须组件≥2块才可应用，若<2块则至灰）

规则：以所选组件底部位置为基准垂直移动。

示意图如下：





在检查器顶部，我们可以看到除了以上描述的几种对齐按钮。在最左侧开始栏处还有一个类似的画面布局的图标（如左侧所示）。

当您用鼠标点击它的时候，会看到一个间距的弹出窗口。

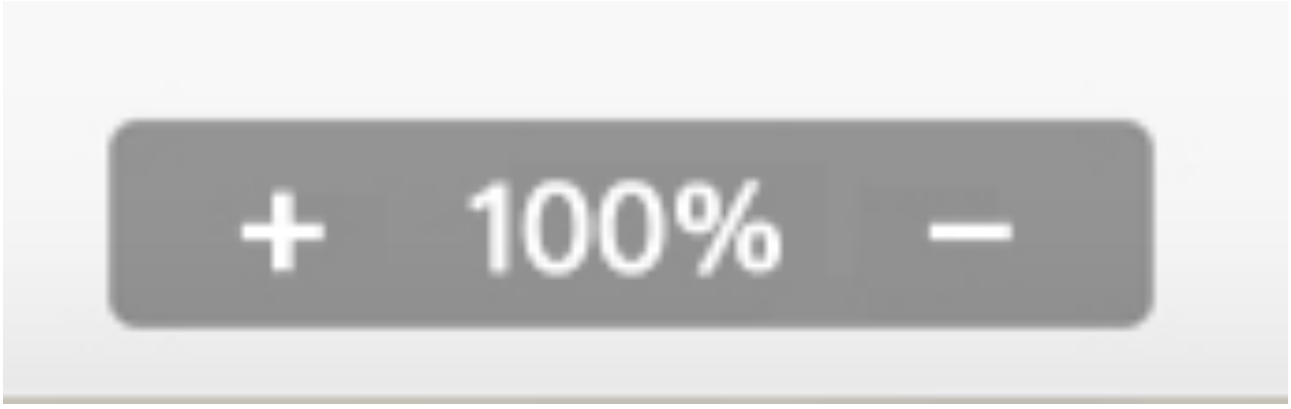
这个功能是为了让您更好的去对齐您的组件。当您选择两个或两个以上组件的时候。您可以使用间距调整功能去让他们保持同一数值排列。

比如说：我选择两个按钮组件。将水平间距设置为20px，那么这两个组件就会按照20像素的间距水平排列。

窗体缩放

因lugia里的画布尺寸是无限延伸的，以至于当页面放大时候也不会因为模糊而看不清具体画布的形态。当您想要放大缩小窗体进行更加合适的操作时候，您可以利用鼠标或键盘的配合。

若您想要放大缩小整个窗体，您可以点击屏幕中下方的浮动条“+或-”以改变当前窗体的位置。注：窗体改变的大小并不会影响实际的组件像素。

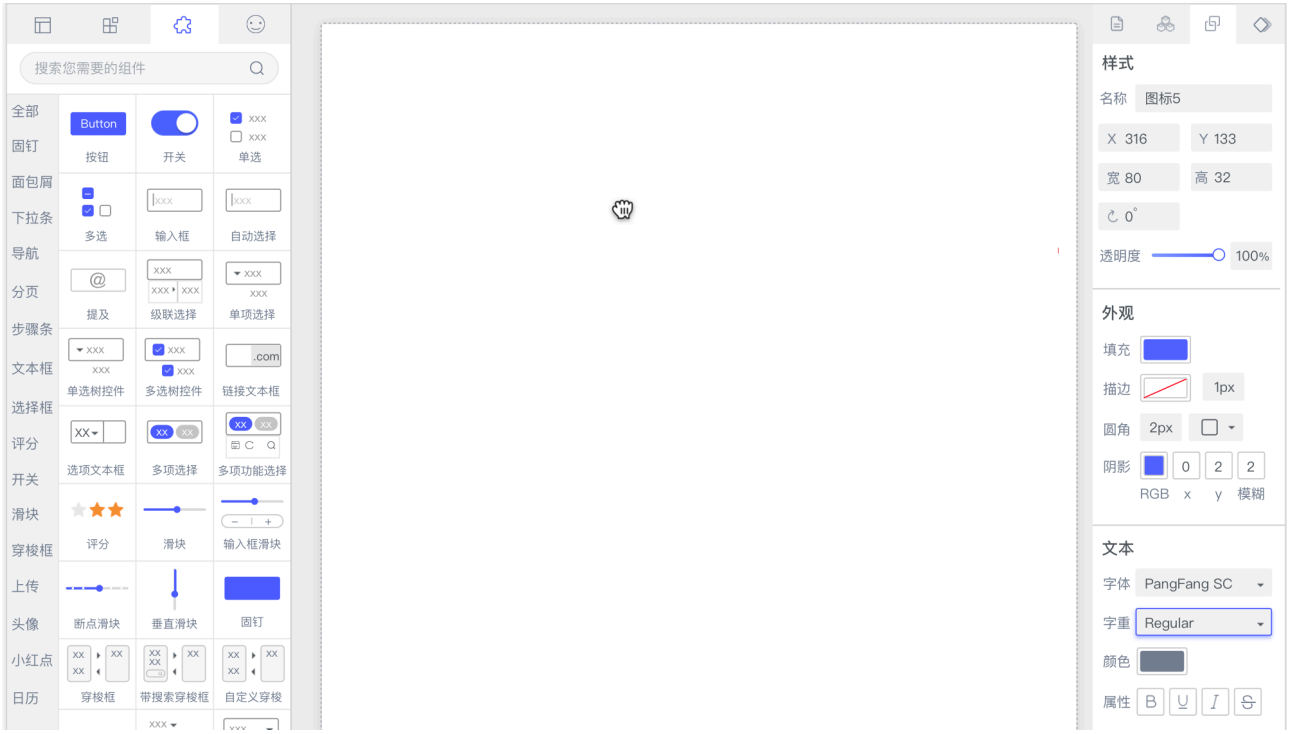


当然对于中级用户来说，每次通过点击才可方法缩小会影响您对页面构建的效率。lugia针对方法缩小独立研发几种关于方法缩小的快捷方式。当您用鼠标在屏幕中，您可以按住command（ctrl）+鼠标滚轮来帮助您放大查看画布或对象。您也可以使用键盘上的command+ “+或-”来放大缩小您的窗体。

窗体移动

当您使用了放大缩小功能去移动您的画布,这时您会觉得画面虽然比例改变了,但是并没有屏幕适应到您想要的位置。这个时候就要借助另一项功能“窗体移动了”

同样窗体最简单的办法就是通过方向键来控制所要移动的方向。但每次一像素的点击过于麻烦。因此我们开发了通过点击鼠标左键+键盘空格键的配合，当您按下空格键后您会发现您的鼠标会变成了一个手的形状，这个时候您只需要配合鼠标左键将画板移动到您想要的位置即可。



六、上边栏

前面说到上边栏相当于页面的主控台。在这里您可以进行一系列的项目管理。

返回项目

Lugia是以一个项目为一个窗体，所以如果在您操作多个项目时，可以通过返回项目去查看，每一个项目为单独的一个lugia.D项目，我们会将所有您最近使用的项目都保存在启动页中。在返回启动页面中，您可以创建新项目或者打开其他的项目。

新建项目

点击新建项目，只需输入页面所需尺寸，就可以开始您的页面设计了。

保存项目

点击保存项目或运用快捷键command+s进行保存。文件格式为lugia专有格式名称xxx.lugiad

预览

预览就是在开发模式下运行应用程序。

您也可以自动在浏览器中打开 http://localhost:3000 去查看它。

注意因为lugia mega不是一个静态页面的设计器而是一个完整的页面构建工具。所以您在操作过程中的任何操作在预览模式下都会试试的显示在浏览器中浏览器会根据您的操作自动重新加载。

而且您还可以看到构建错误和 lint 警告。

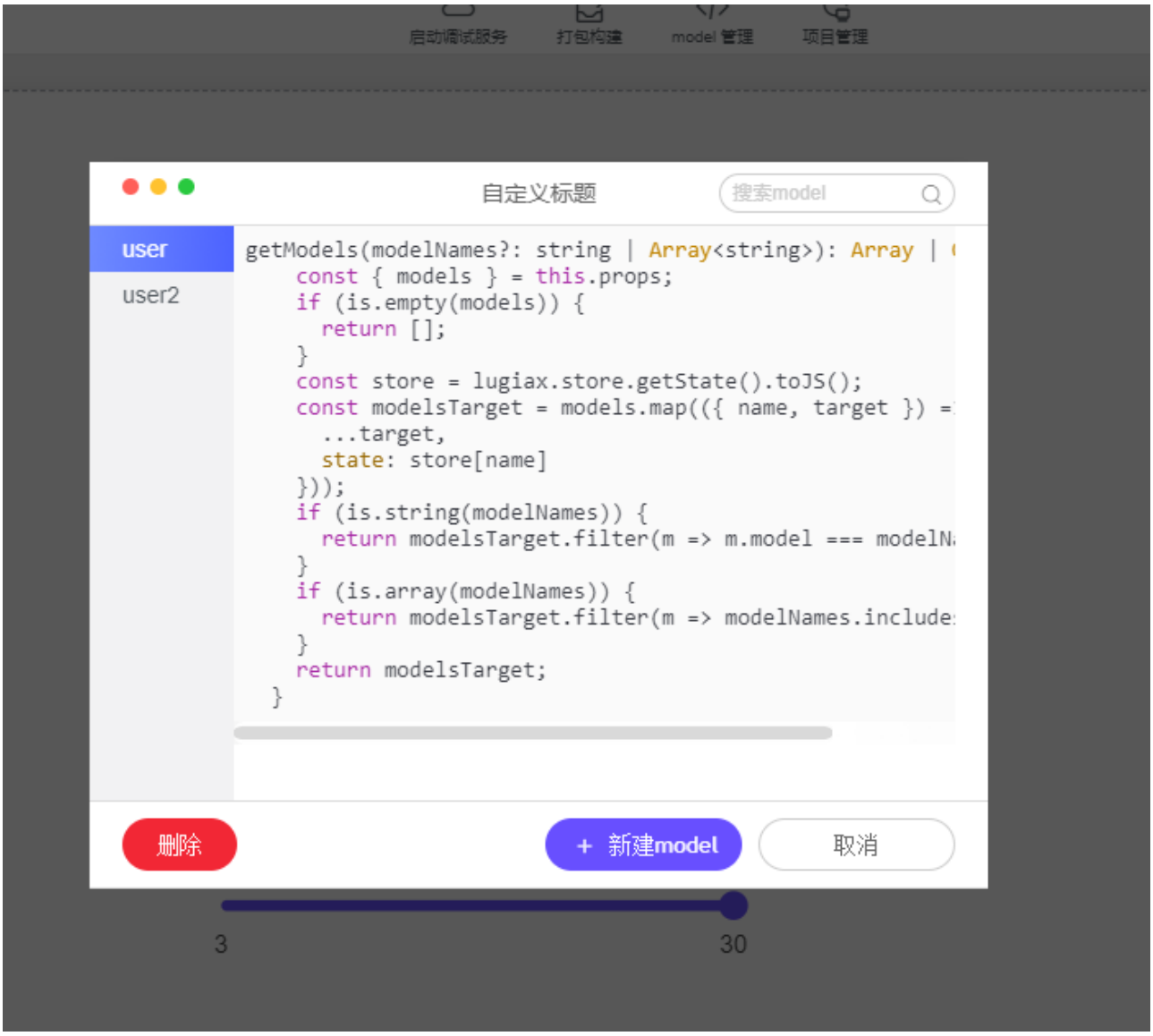
尺寸

用户可以在尺寸控制区域调整当前页面的宽、高、颜色和名称。

Model管理

model管理顾名思义是对项目的model进行管理。

您只需点击上边栏的model管理项目就可看到



项目管理

项目管理是用来管理本项目的配置文件，您可以在此查看当前的项目信息。也可以搜索最新的依赖资源进行安装和更新。

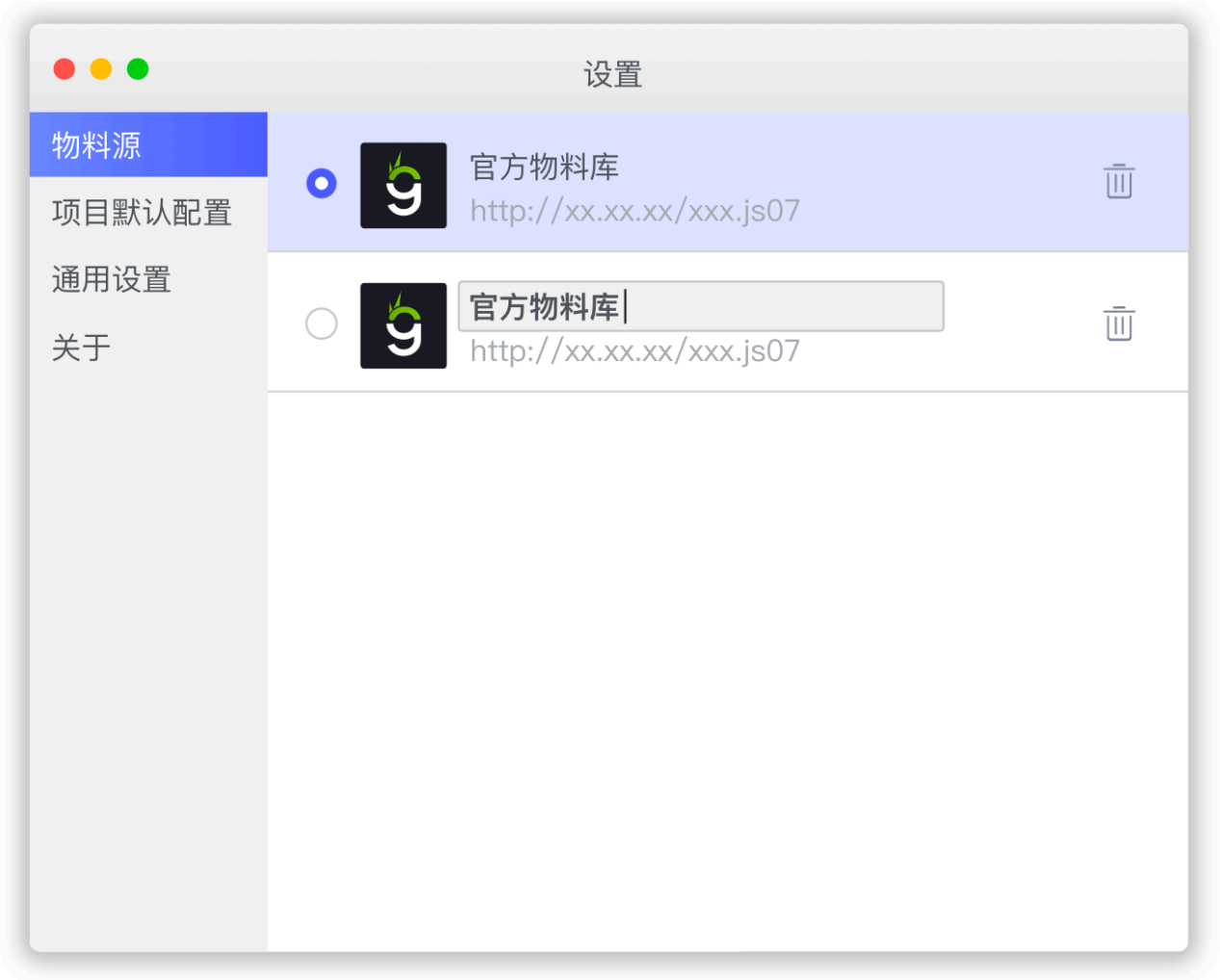
设置

设置功能是对整个lugia页面的一些常规项的更改。您可以根据自己的喜好去配置相对应的属性。

物料源

在 lugia 中，组件、区块、布局、模板等统称为物料，由 lugia 团队维护，在内部有一套完整的开发规范和工具；基于此，你可以参与共建，也可以自建私有的物料库。

在官方物料库中：Lugia 团队开发维护的物料库，质量保证，提供技术支持
在私有物料库：在实际项目中，因为我们提供的物料库和设计风格可能不能满足您所需要的某些业务场景，这时您可以自定义内部的私有物料库。



项目默认配置

项目默认配置是对默认的用户名、端口号、路径和项目版本的管理。更改会影响您的项目初始值。

您也可以在项目配置中去设置，是否在项目开启是打开浏览器。

通用设置

在通用设置中，您可以选择您所需要的终端和编辑器，以及对应用的物料库进行使用选择或删除。

关于

关于页面为lugia当前的版本信息，您可以在此查看lugia mega的相关用户文档，也可以去检测我们最新的lugia mega版本。

如果您对lugia有更多的兴趣，我们欢迎您查看我们的文档和更新的日志。


设置
物料源
项目默认配置
通用设置
关于
Lugia mega v1.0
如果您有什么问题，请查看lugia使用说明
检测版本
升级版本
文档
更新日志

响应式

响应式界面可以对当前页面进行一个响应式添加，您可以选择所需要相应的尺寸作为初始母版尺寸，再次基础上选择相应响应的页面，被响应的页面需要在原页面基础上调整响应的信息。

导出

打包构建

将生产环境需要的应用程序构建到项目 dist 目录。 lugia会正确地打包构建并优化整个应用以提供最佳性能。dist下输出文件将会被压缩并包含哈希，以便更加有效的使用缓存策略。dist 下输出的文件可以部署到服务器上。

Lugia提供开发现代单页 React 应用需要的所有环境：

开箱即用，包含 start、build 和 test 命令
React，JSX，ES6，TypeScript 和 Flow 语法支持
对 Less/Scss、css-modules、postcss、styled-components 的支持
CSS Autoprefixer 前缀自动补全
交互式的单元测试，内置支持覆盖率报告功能；基于 jest，包括 UI 测试（基于 enzyme）
mock 服务，支持引入 json、excel、csv 格式的文件作为数据来源
browser-sync 支持，保持多个浏览器和设备同步
实时调试服务，包含错误警告
打包构建脚本，把 js、css、图片及其他资源构建在一起，添加 hash 值和源码映射
通过一系列的内部优化，整体构建速度提升了 60% ~ 98%，默认开启缓存后速度会再次提升 300% ~ 500%。bebal 编译速度达到 8.58 runs/sec ±8%。
按需加载 Model 和 Router，加快访问速度；同时打包后每个文件体积可以有效控制在 200KB 以内。
灵活的配置支持

曾经的构建配置如下图所示：

```
hanbo@star-PC MINGW64 /f/yssgitlab/lugia-admin (master)
$ yarn build
yarn run v1.12.1
warning package.json: No license field
$ lugia-scripts build
Build completed in 65.482s
```

```
[Lugia Mega] Build complete. The dist directory is ready to be deployed.

Done in 112.51s.
```

Lug mega的打包配置如下所示：

```
hanbo@star-PC MINGW64 /f/yssgitlab/lugia-admin (master)
$ yarn build
yarn run v1.12.1
warning package.json: No license field
$ lugia-scripts build
Build completed in 20.462s
```

```
[Lugia Mega] Build complete. The dist directory is ready to be deployed.

Done in 24.14s.
```

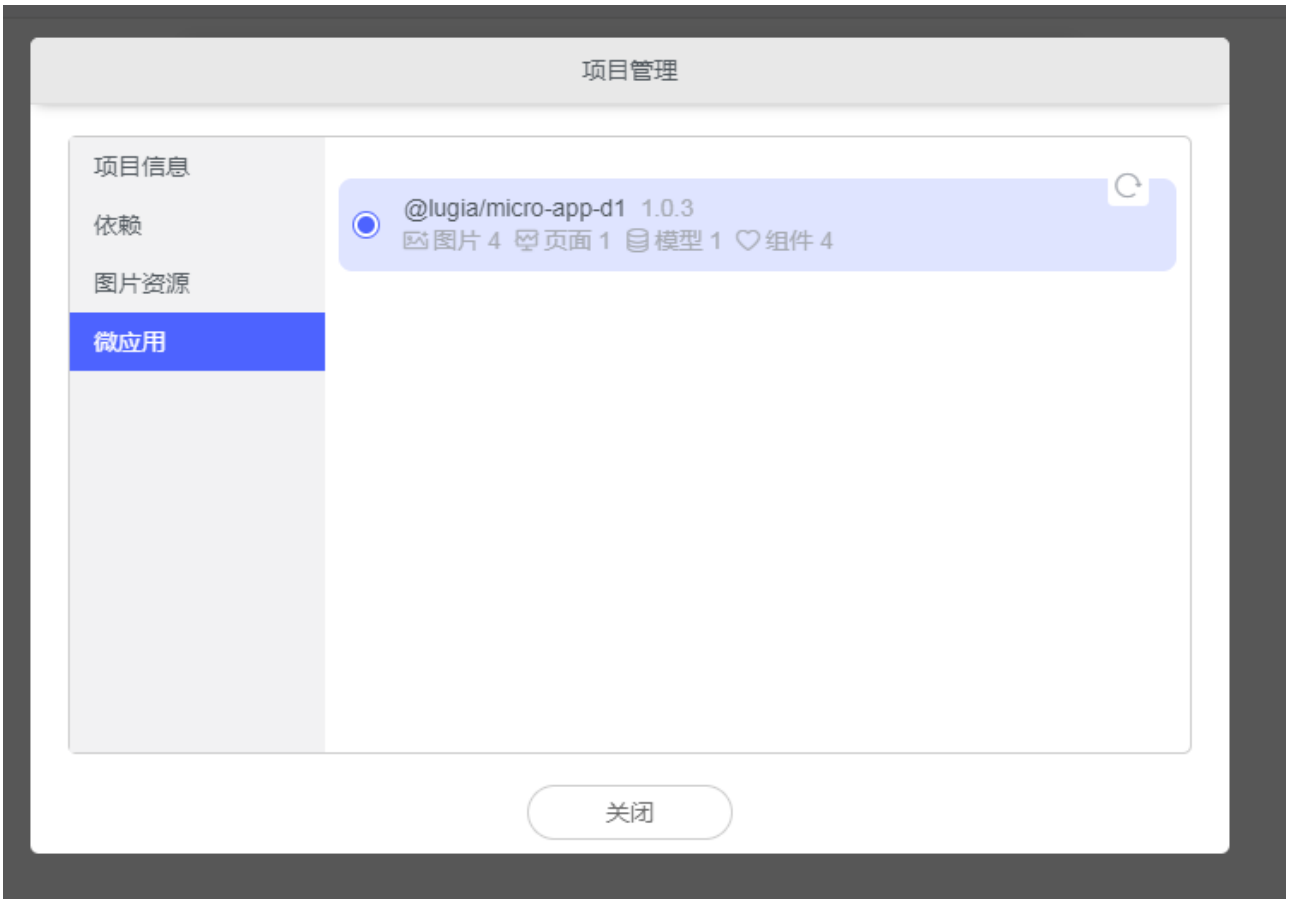
微应用

当开发一个大型项目时，把不同的功能模块以单一、微应用的形式存在，而在运行时，则通过构建系统合并这些应用，组合成一个新的应用

您可以讲所做的项目作为微应用发布到npm上，您的项目成果可以通过包的形式，供其他用户作为依赖进行安装使用。

具体操作：当您点击导出-微应用发布后。您可以配合项目管理进行依赖的安装，成功后项目管理微应用页面会显示您当前安装的微应用。您的微应用也自动应用到您的项目当中，您可以通过左边栏找到它。

值得注意的是，如果提示发布失败，有可能您没有微应用的发布权限，您可以登陆您的npm账户，或可以在项目的package.json更改您的项目配置。


项目管理
项目信息
依赖
图片资源
微应用
@lugia/micro-app-d1 1.0.3
图片 4 页面 1 模型 1 组件 4
关闭

代码拆分（Code Splitting）

与用户使用前下载整个应用程序不同，代码分割允许你将代码分割成小块，然后按需加载。lugia会自动对每一个页面做代码切分，以保证更加快速的加载应用。这里使用了动态import() 进行代码拆分。 它是在第3阶段的 提案 。 import() 函数表单接受模块名作为参数，并返回一个 Promise ，该 Promise 总是 resolves 到模块的命名空间对象。

```
[Lugia Mega] File sizes after gzip:

  131.15 KB  dist\vendors.c5baa2a0.js
  117.98 KB  dist\index.edf7c3aa.js
  80.74 KB   dist\0.db461030.async.js
  58.3 KB    dist\10.734b9910.async.js
  34.6 KB    dist\1.de472296.async.js
  26.01 KB   dist\2.7e11a3dd.async.js
  12.84 KB   dist\index.04f8f069.css
  9.45 KB    dist\5.0d79e5a6.async.js
  9.01 KB    dist\23.88d77251.async.js
  6.77 KB    dist\6.cfb68f28.async.js
  5.8 KB     dist\19.899fe030.async.js
  4.04 KB    dist\vendors.39e1dd28.css
  3.71 KB    dist\21.52a23fde.async.js
  2.7 KB     dist\33.f781dcd8.async.js
  2.61 KB    dist\31.e63d57a2.async.js
  2.47 KB    dist\30.58b9da35.async.js
  2.09 KB    dist\29.9b7a4aa9.async.js
  1.9 KB     dist\16.f23e405a.async.js
  1.55 KB    dist\11.04082d3a.async.js
  1.55 KB    dist\4.752f57d3.async.js
  1.54 KB    dist\9.8d43b01d.async.js
  1.54 KB    dist\8.2e1bd9db.async.js
  1.31 KB    dist\7.8bcfab88.async.js
  1002 B     dist\17.e109c584.async.js
  732 B      dist\15.bc32fac0.async.js
  730 B      dist\14.d98419ef.async.js
  730 B      dist\13.197ff155.async.js
  729 B      dist\12.2bf6e00f.async.js
  666 B      dist\18.a51e732c.async.js
  112 B      dist\25.9f28512a.async.js
  108 B      dist\32.f24cec27.async.js
  107 B      dist\22.dbde9373.async.js
  106 B      dist\24.ed4ee2e9.async.js
  106 B      dist\20.cc226236.async.js
  105 B      dist\28.b4e94f5f.async.js
  105 B      dist\27.f6c16b40.async.js
  103 B      dist\26.5095163d.async.js
  62 B       dist\34.d5cbcecd.async.js

  Images and other types of assets omitted.
```

七、父容器（编组）

父容器，简单的说就是一个组合的功能。该功能是为了更好的将组件拼合成一个整体。您可以以拖拽的方式同时将多个图层放置到一个容器内，这样您就可以随意的移动和缩放，同时也可以在容器内修改每一个独立的组件。

Lugia mega中的父容器功能是非常强大的，因为对于逻辑层来讲，一个父容器就相当于一个简易的画布。您新创建画布上可实现的功能在父亲容器上都可以实现。

为了防止您的组件因放入父容器而变得不可控（例如sketch：放大缩小组内部图层也会跟随变化）。lugia在一定程度上减少了父容器对组件的占有权。也就是说父容器的形状改变并不会影响到内部组件的变化，且父容器内部组件在您的操控范围内也是可以随意拖拽、框选的。它并不需要在进入到一个相对应的父容器页面。（例如:adobe illustrator成组后想修改其中内容，需要点击进入）这样您就可以自由灵活的操控父容器内部的组件系统。

您可能会有疑问，将组件放入父容器，组件与父容器有什么影响呢？

还是回到之前描述的逻辑层关系，一个父容器与内部组件的关系与一个画布上新建一个组件的关系一致。可同时移动，可同享一个标尺系统而已。

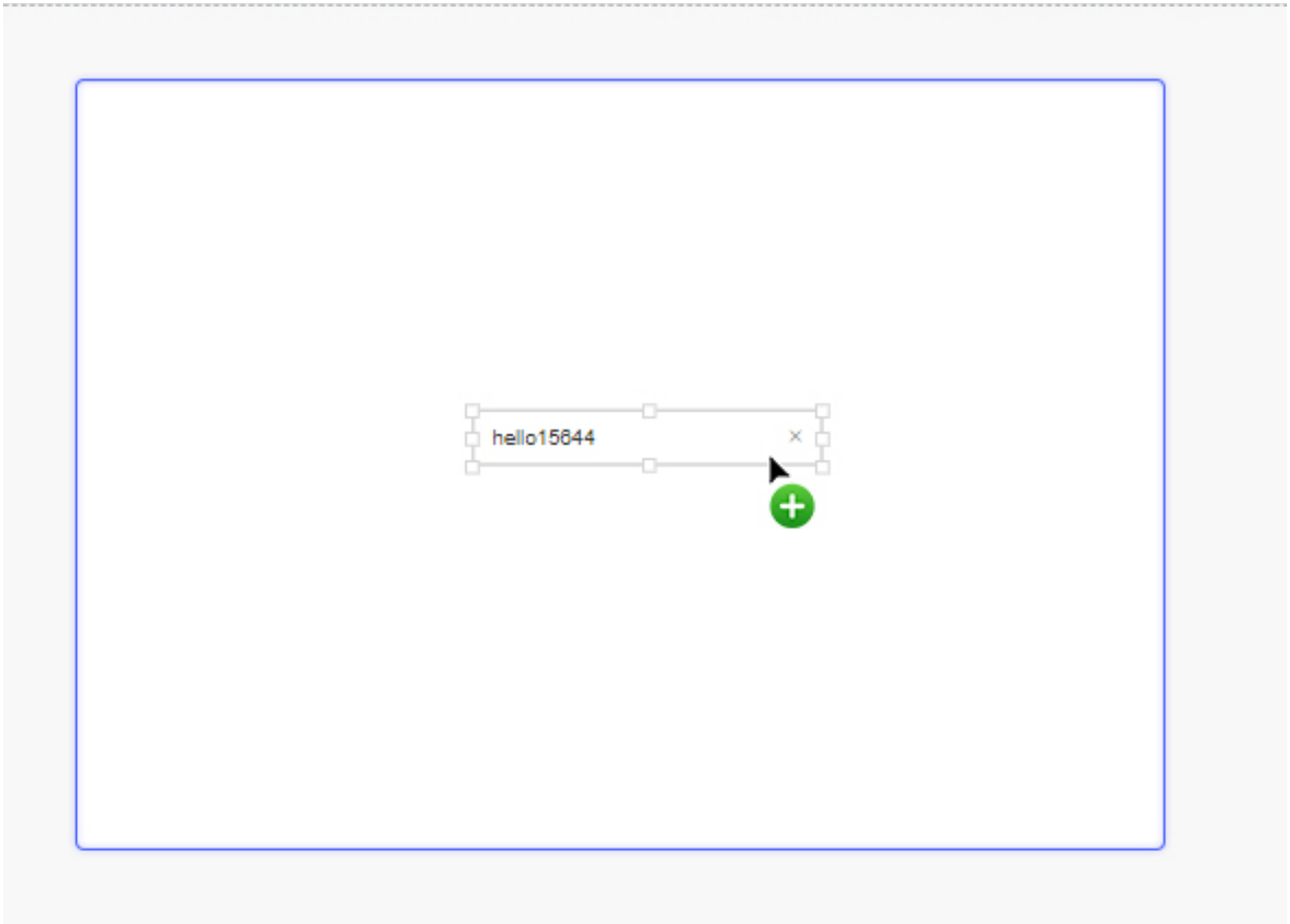
创建父容器

若您想要创建一个父容器，您只需要在左侧组件中找到一个命名为盒子的组件。将盒子组件拖拽到画板上，这时页面中就会自动生成一个父容器层。

值得一说的是，您也可以在右侧检查器中修改父容器的属性。将它变成您需要的样式。

编组父容器

若您想将组件拖拽到容器中，您只需要直接将画布上的组件用鼠标拖拽到已经生成好的副容器中就可以了。当把组件放置在父容器中，父容器会有颜色的反馈。且鼠标的光标显示为可添加状态。



八、模型绑定

我们针对lugia组件，给予组件一个mode来进行绑定，l以便与管理组件的状态。

我们提供了3个步骤：“首先您要选择一个您要绑定的组件,为组件指定绑定的模型。

其次，通过组件内部触发的事件来更新model mutation，实时更新model状态。

最后您可以通过模型绑定来将组件的部分属性与模型相关联从而达到模型更新组件的目的。

我们通过实时读取lugia x的模型文件，来实时加载到lugia内部。

关于Lugia x

Lugia x是一个基于 Redux 的前端状态管理工具。提供简单高效的全局状态管理方案、 基于 async/await 的异步操作、快捷的双向绑定。LugiaX 内置路由库，对 react-router 做了轻量封装，使用起来更加简单明了。

设计思想

基于`redux` + `redux-saga` 封装出更加简单的状态管理工具。我们引入了 `mutation` 的概念 (`mutation` + `state`) ， 简化了 `redux`。lugiax 的 state 是不可变类型的数据， 可参看 [Immutable](https://facebook.github.io/immutable-js/docs/#/)； Immutable数据一旦创建，就不能更改。而 `mutation` 就是修改 `state` 的唯一途径。state 被修改后， 并不会通知全局来进行更新，而是通知所绑定的对应的 Component 来进行更新。

state
`state` 是单独的，每个 model 都有自己的 `state`，并且是不可变类型的

mutation
`mutation` 是一个标准函数，是唯一修改 `state` 的途径，修改方式是通过返回一个新的 state ，然后通过 state.set 来修改 state。并且 `mutation` 只能修改自己域下面的 `state`。`mutation` 提供了 async 和 sync 两种不同的操作方式。

mutation 进阶
可以通过 `wait` 等待一个mutation结束，然后处理返回新的 state
可以通过 `lugiax.on` 进行全局监听，被监听状态改变后，会执行 `lugiax.on`
可以通过 `lugiax.getState` 获取其他 model 状态

wait:

```
const mutation = {
  async: {
    async changePwd(data, inParam, { mutations,  }) {
      return data.set('pwd', inParam.pwd);
    },
    async changeName(data, inParam, { mutations, }) {
      return data.set('name', inParam.name);
    },
    async start(data, inParam, { mutations, wait,}) {
      await wait(asyncChangeName);
      state.set('pwd','333')
    }
  }
};
lugiax.register({model:'user',state:{name: 'li',pwd: '12345'},mutation});
```

```
const getAsyncResult = new Promise((resolve,reject) => {
  const asyncResult = [];
  lugiax.on(async (mutation, params, { mutations, wait, }) => {
    if(true){ // 做一些判断
      asyncResult.push(params)
    }
    if(asyncResult.length === 2){ // mutation 全部响应后放回
      resolve(asyncResult);
    }
  })
})
```

lugiax.on:

lugiax.getState:

```
import lugiax from "@lugia/lugiax";
const userModel = lugiax.register({
  model: 'user',
  state: {name: 'user'},
  mutations: {}
});
const loginModel = lugiax.register({
  model: 'login',
  state: {login: ' '},
  mutations: {}
});

lugiax.getState().get('user').get('name'); // user
```

lugiax–router 路由

lugiax 对 react-router 做了轻量的封装，createApp 和 createRoute 供你创建路由使用。

createRoute

```
createRoute({
  [path: string]:{
     render?: Function,
     exact?: boolean,
     strict?: boolean,
     component?: Function,
     onPageLoad?: Function,
     onPageUnLoad?: ?Function
  }
})
```

除了 component 的静态打包外，还提供了 render 动态打包，用于代码分割。

页面生命周期函数

onPageLoad 页面加载完成后执行

onPageUnLoad 页面卸载时执行

createApp

```
createApp({
  routerMap: RouterMap, // type: Object, 通过 createRoute 创建后的route
  history: Object,
  param?: CreateAppParam = {}
})
```

param 提供 loading 和 onBeforeGo 两个api；

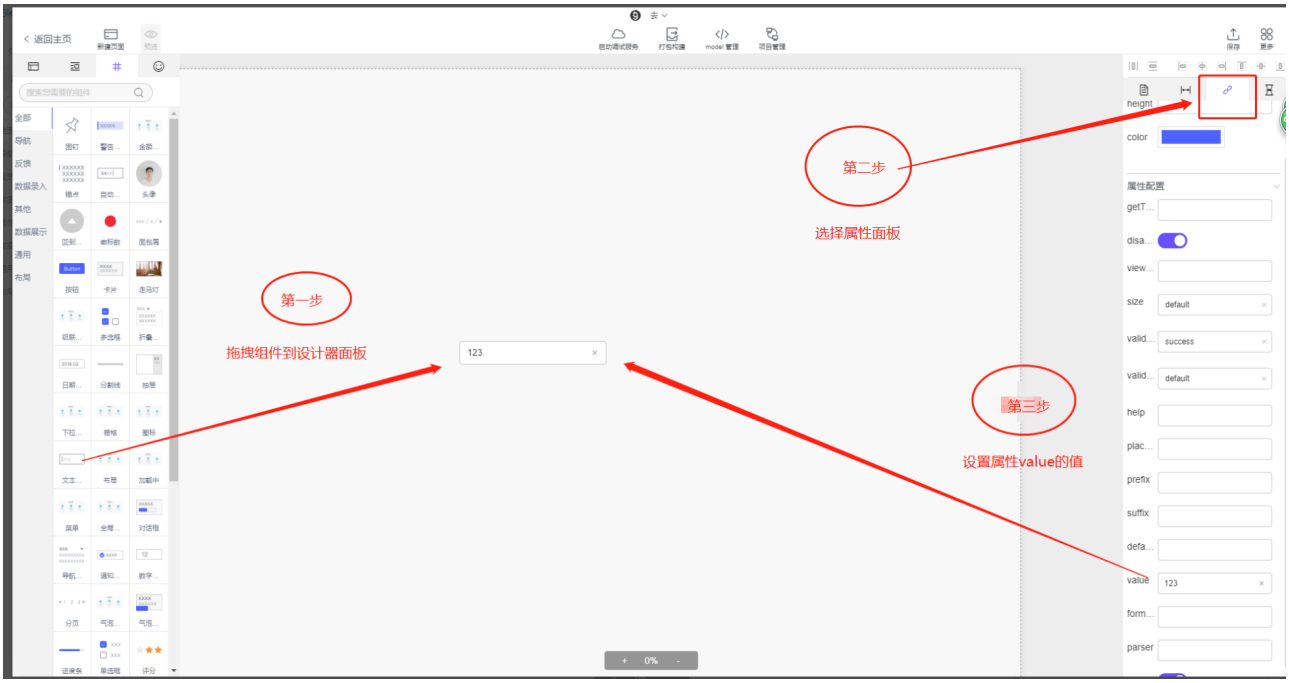
loading: 路由切换的加载页面，可配置 Component 组件

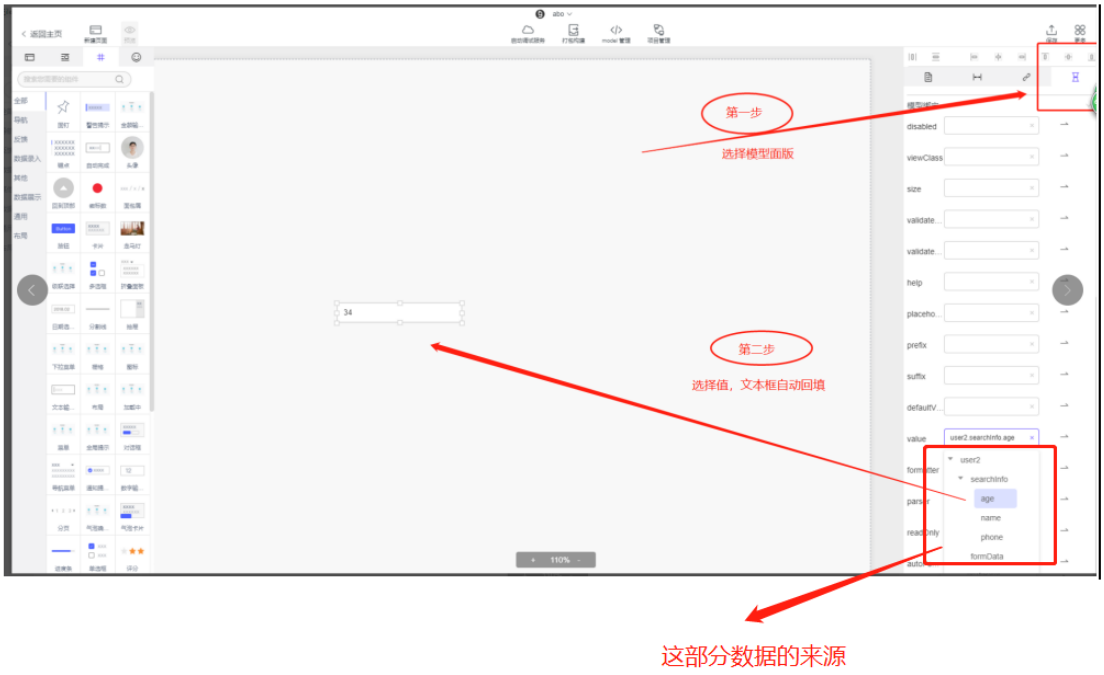
onBeforeGo 跳转之前的回调，可做权限处理

具体绑定应用

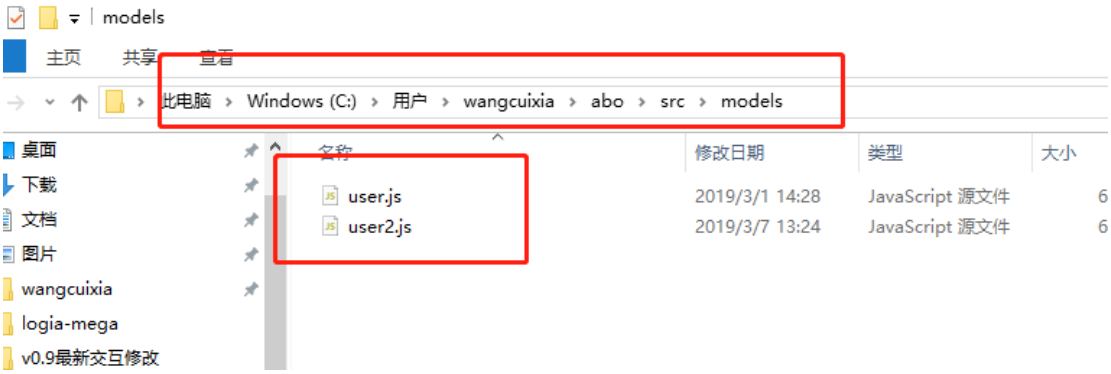
当您选择一个组件的时候，您可以在检查器上选择绑定窗口。

具体操作步骤如图所示





这个文件所在的位置是新建项目时，根据设置的路径存放的；



九、快捷键

lugia mega有一部分为数不多的快捷键，来为您服务。当您掌握之后可以在一定时间内提高您的工作效率。

通用快捷键

Esc取消选中
Space：抓手工具
command+c：复制
command+x： 剪切
command+v：复制
Option+鼠标左键：拖动复制
option+shift+鼠标左键：拖拽平移复制
command+z：撤销操作
Backspace：删除

移动图层 & 更改尺寸

Shift+鼠标左键移动：基于拖动的方向锁定相对应的x\y轴坐标
Shift+鼠标单击组件：可以进行多选
按住锚点+shift：等比例缩放
按住锚点+option：组件以中心为居中点进行缩放
按住锚点+shift+option：组件以中心为居中点进行等比例缩放