

海报生成器

bigezhang

2017-11-02

需求描述

根据用户输入的字段或上传的图片生成海报

实现效果



实现原理

1. 初始化 canvas

<canvas> 元素创造了一个固定大小的画布，它公开了一个或多个渲染上下文，海报生成主要使用2D 渲染上下文。

canvas起初是空白的。为了展示，首先脚本需要找到渲染上下文，然后在它的上面绘制。

<canvas> 元素有一个叫做 getContext() 的方法，这个方法是用来获得渲染上下文和它的绘画功能。

1.初始化 canvas

```
this.cvs = document.createElement('canvas')  
this.ctx = this.cvs.getContext('2d')  
this.cvs.width = options.width  
this.cvs.height = options.height  
this.ctx.save()
```


实现原理

2. 绘制文本

canvas 提供了两种方法来渲染文本:

`fillText(text, x, y [, maxWidth])` | 在指定的(x,y)位置填充指定的文本, 绘制的最大宽度是可选的.

`strokeText(text, x, y [, maxWidth])` | 在指定的(x,y)位置绘制文本边框, 绘制的最大宽度是可选的.

2.绘制文本

```
this.font = function (data, param) {  
    var self = this  
    self.ctx.save()  
    self.ctx.fillStyle = param.color  
    self.ctx.font = [  
        param.fontStyle,  
        param.fontWeight,  
        param.fontSize + 'px',  
        param.fontFamily  
    ].join(' ')  
    self.ctx.textAlign = param.textAlign  
    self.ctx.textBaseline = param.textBaseline  
    self.ctx.fillText(data, param.left, param.top)  
    self.ctx.restore()  
    self.trigger()  
    return self  
}
```


实现原理

3. 绘制图片

获取图片源对象

使用 `drawImage` 方法将它渲染到 `canvas`

3.绘制图片

```
ctx.drawImage(img, 0, 0, param.width, param.height)
```

```
this.toDataURL = function (param) {  
    return this.cvs.toDataURL(param.type, param.quality)  
}
```

一些注意事项

图片如果使用线上的地址，需保证图片在服务端设置允许跨域或着将图片格式转换成 base64

canvas 生成的图片是 base64 格式，如果使用 FormData API 上传文件，可以先将其转换成 Blob

图片下载：pc 端源背景底图不是太大，可以实现下载

还是看代码吧 <https://github.com/clteam/createposter>