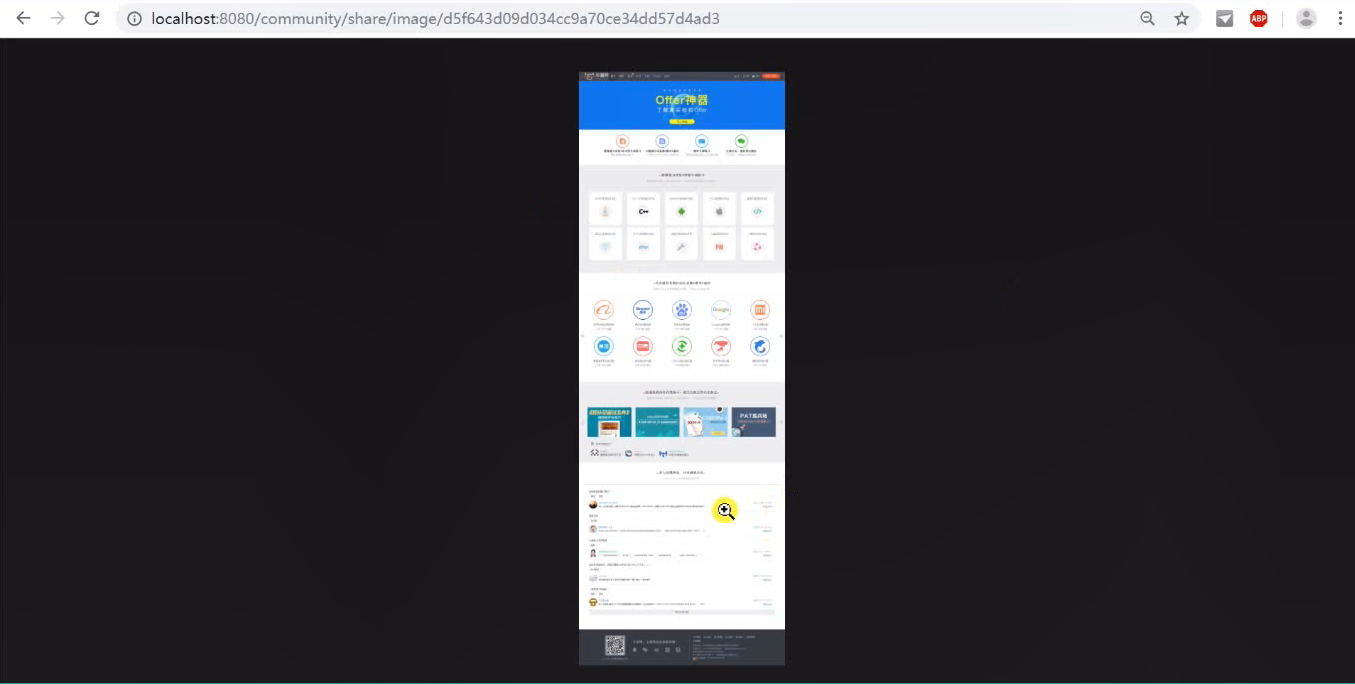
分享时生成图片，相当于截图



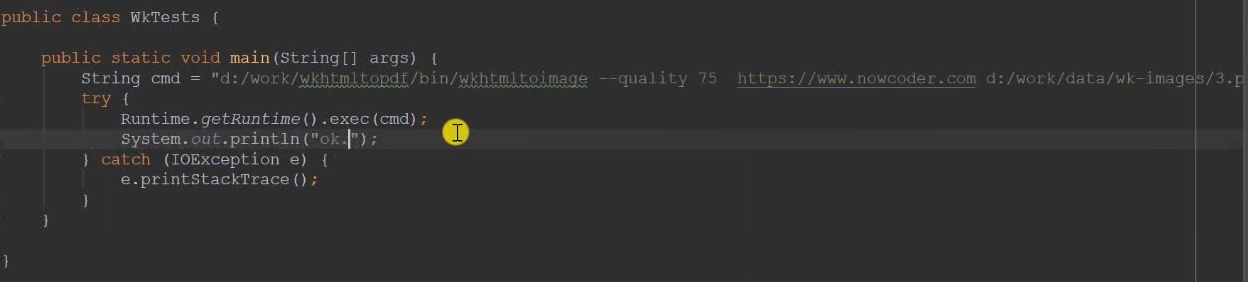
通过wkhtmltopdf工具来实现 将对应的网站生成图片的命令 wkhtmltoimage+域名+生成图片的位置



由于将网站生成图片所产生的图片太大，浪费空间，所以在中间加一个—quality 75参数，意思是要存放图片的质量为原图片的75%

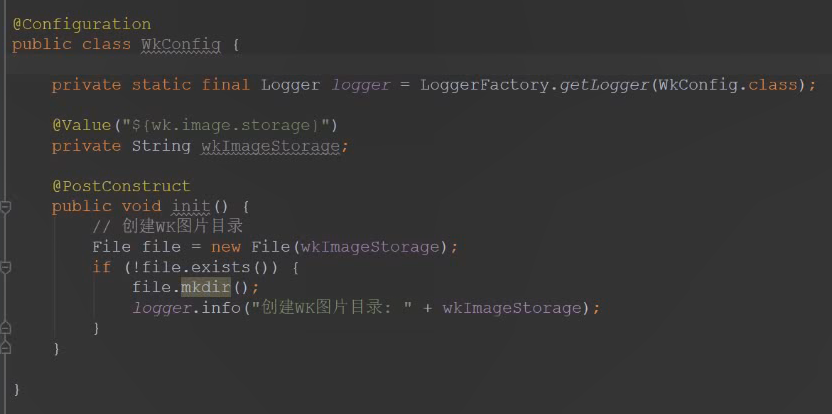


前面是在cmd中操作的步骤，在java中是：通过Runtime.getRuntime.exec（cmd）来实现，其中下图的cmd字符串是在cmd运行这个命令时，也就是上图的执行命令敲的代码，将他以字符串的形式敲入exec（string）的参数中，他会以并发的方式在本地的操作系统中执行这个命令

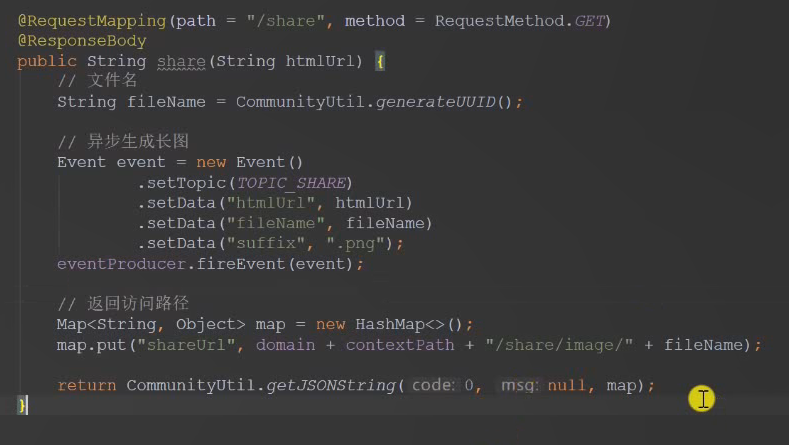


在实际的应用中需要将上面的一些目前固定的参数写入配置文件中，以便于以后更改更改，比如在wkhtmltoimage中的wkhtmltoimage命令在哪个地方，以及生成图片的路径（和上图对着看）

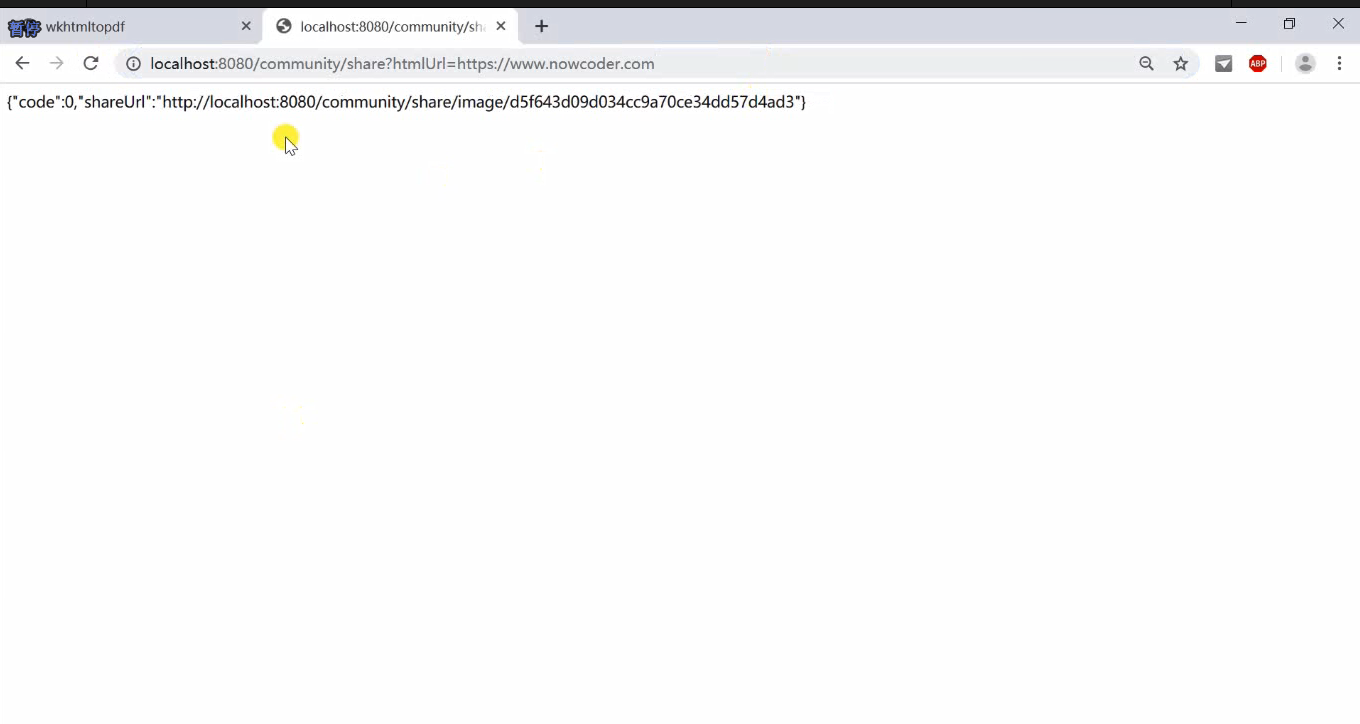
为了防止存放图片的目录没有创建需要在配置类中编写，因为配置类加了@configuration的注解会在服务启动的时候，会先加载，所以我们在配置类中创建这个目录，还需要在创建目录的方法上加@postConstruct注解，表示初始化，这样配置类在实例化的时候会自动调用这个方法（wkImageStorage是我们配置的图片存放目录）



由于生成图片的时间比较长，所以我们采用异步的方式，加入到消息队列中，客户端发送分享请求，随机生成文件名，发送事件给消息队列，由于没有实体类所以不传入entity，而且在event中date是map格式的数据，所以传入多个值，将生成图片所需要的的url，文件名字，以及文件格式加入date中，生成的图片保存在我们服务端中，不保存在客户端，我们将生成的图片在本机的路径保存下来，而且客户想要访问这个图片还需要先定位到我们的域名，将这些拼在一起，传给json字符串，这个请求不用返回一个网页，只需要返回体中给客户返回一个路径就好，让客户复制这个路径，去浏览器访问，得到图片

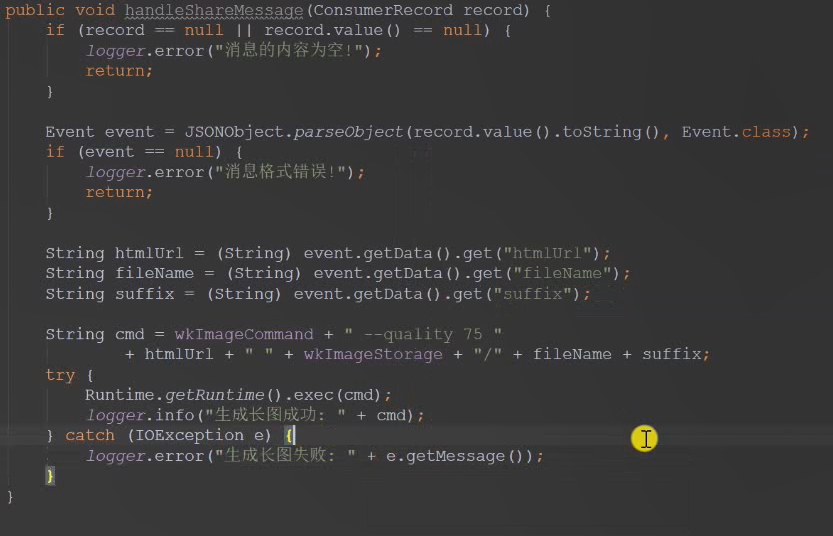


客户在发送请求得到的效果是这样的:需要复制这个路径，继续发送请求，查看图片



消费分享事件：





客户在获取到json字符串之后，想要获取图片，发送的请求：由于这个返回体是图片，不是网站也不是json，所以需要直接调用response

