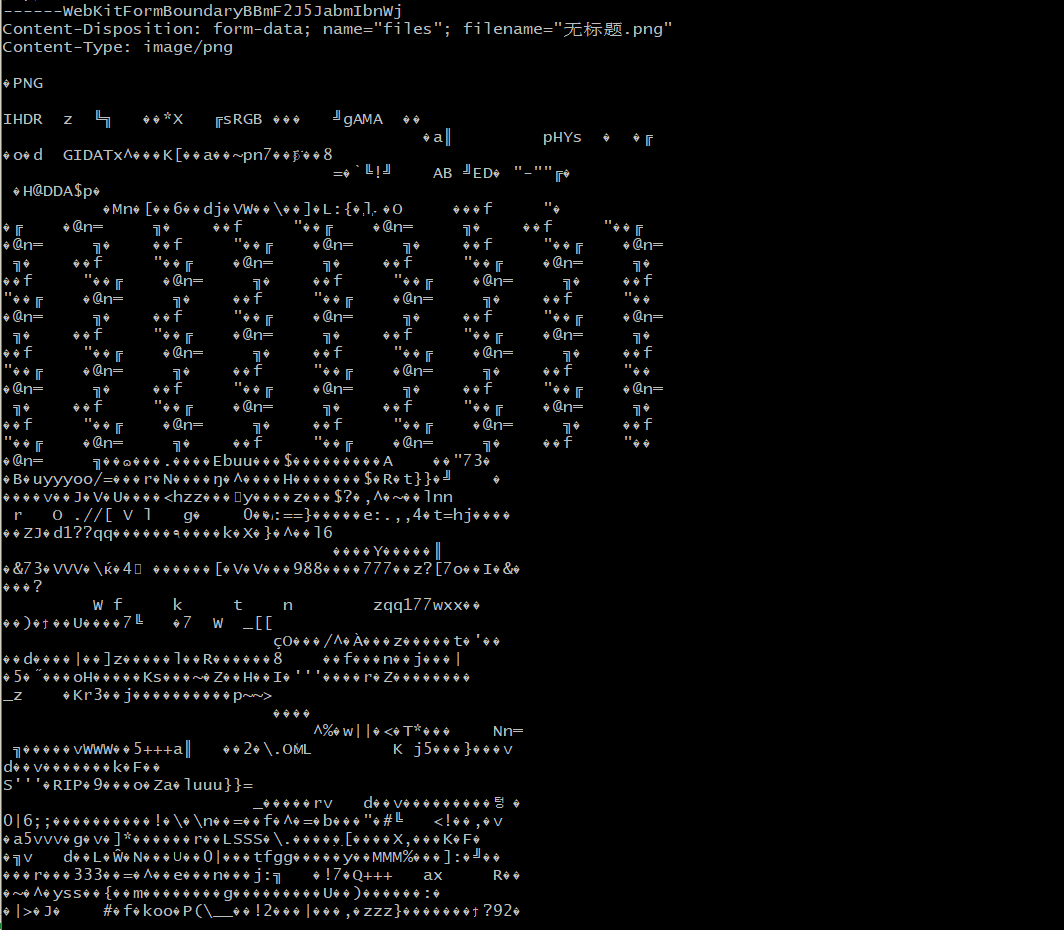
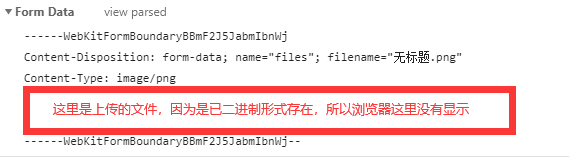
Web端上传文件经过formdata处理过后，node服务端接收到的post数据是这样的

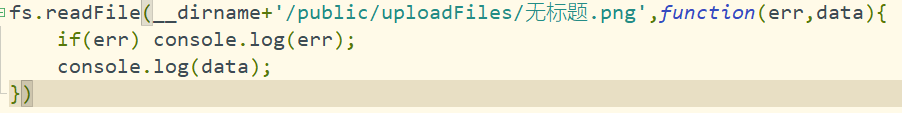


以下是浏览器发送时的样子

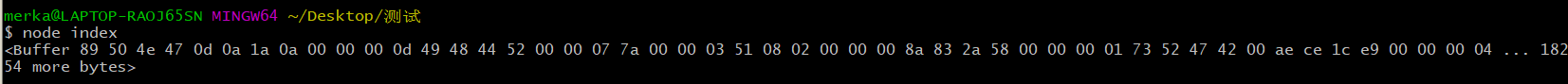


从“浏览器端”和“node服务端”可以看出大体结构没有什么太大变化，主要就是已二进制形式存在的文件被已乱码的形式打印出来了。其实这是因为node在接收时强制将二进制流转换成“编码为utf8的字符串形式”，所以要从新设置node接收数据的编码格式设置为binary。

为了验证上面“编码为utf8的字符串形式”我们用fs.readFile()来读取文件

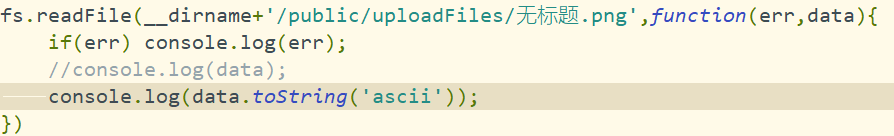


已二进制流形式输出



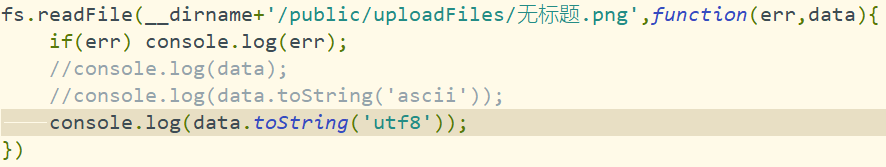
现在我们对代码加工把读取的文件流转换成已“ASCII形式字符串输出”和“utf8形式字符串输出”

ASCII形式字符串输出:





utf8形式字符串输出:





清楚的可以看出两种编码输出的字符串的区别。

用node模拟一个上传的结构

但还是不建议把流数据强转成别的编码格式的数据。

和其它数据一块处理发送过的



