

**软件构造课程设计**

**项目文档**

**项目名称：FileToChart**

**项目负责人：黄振洋**

**所在学院：软件学院**

**专业年级： 2015级**

**学 号： 2015141463062**

目录

[一 项目简介 3](#_Toc499995080)

[二 需求说明 3](#_Toc499995081)

[三 概要设计 4](#_Toc499995082)

[四 详细设计 6](#_Toc499995083)

[数据介绍 6](#_Toc499995084)

[处理算法介绍 9](#_Toc499995085)

[五 代码 12](#_Toc499995086)

[六 测试 13](#_Toc499995087)

# 一 项目简介

**项目内容概述**

|  |
| --- |
| 本项目为FileToChart，旨在实现一个小工具，能够保存用户上传的文件，将其以二进制方式打开，并转换成short型数据的数组之后，映射到图表中显示，同时通过用户交互，提升用户体验 |

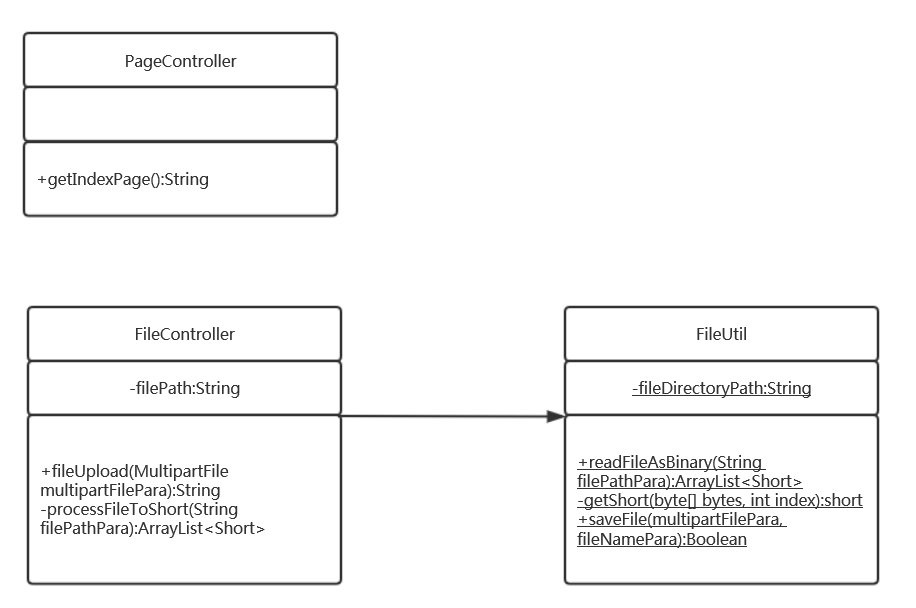
# 二 需求说明

FileToChart项目需求分析如下表所示：

|  |  |
| --- | --- |
| 需求编号 | 需求说明 |
| 01 | 用户上传文件 |
| 02 | 保存文件 |
| 03 | 以二进制方式打开文件 |
| 04 | 将文件二进制数据转为short数组 |
| 05 | 前端能根据后端返回的数据将数据映射到图表中 |
| 06 | 显示文件上传时间 |
| 07 | 显示上传的文件名称 |
| 08 | 显示上传的文件大小 |
| 09 | 能对原始波形进行压缩处理 |
| 10 | 能对原始波形进行扩展处理 |
| 11 | 能对处理过后的波形进行重置操作 |
| 12 | 具有拖动查找功能 |
| 13 | 显示折线图 |
| 14 | 显示柱状图 |
| 15 | 用户未上传文件，点击压缩按钮时对用户做出友好提示 |
| 16 | 用户未上传文件，点击扩展按钮时对用户做出友好提示 |
| 17 | 用户未上传文件，点击重置按钮时对用户做出友好提示 |
| 18 | 对于大小太大的文件，对用户做出友好提示 |

# 三 概要设计

FileTochart为C/S架构，后端的UML图如下所示：



以下为对类的说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 类名 | 说明 | 属性 | 方法 |
| PageController | 页面请求控制器类，处理前端的页面get请求 | 无 | 公有方法getIndexPage()返回index.html页面； |
| FileUtil | 文件工具类 | fileDirectoryPath为文件存储的目录路径 | readFileAsBinary（）方法将文件以二进制方式读取，并返回对应的Short数组；getShort（）是将两个二进制数转成一个byte；saveFile（）函数用于保存用户上传的文件 |
| FileController | 文件请求控制器类 | fileDirectoryPath为文件存储的目录路径 | 公有方法fileUpload()处理文件上传,保存文件,返回处理过的数据;私有方法processFileToShort（）用于调用FileUtil. readFileAsBinary()接口处理文件； |

# 四 详细设计

## 数据介绍

1. FileController fileUpload（）函数数据介绍

public String fileUpload(@RequestParam("file") MultipartFile multipartFilePara)

MultipartFile multipartFile = multipartFilePara; // 用户上传的文件  
String fileName = multipartFile.getOriginalFilename(); // 文件名

long fileSize = multipartFile.getSize(); // 文件大小  
JSONObject jsonObject = new JSONObject(); // 返回的结果  
ArrayList<Short> shortArrayList = null; // 将文件转成short的数组

Boolean saveFileResult = FileUtil.*saveFile*(multipartFile, fileName); // 保存文件的结果

2. FileController processFileToShort（）函数数据介绍

private ArrayList<Short> processFileToShort(String filePathPara)

String filePath = filePathPara; // 文件路径  
ArrayList<Short> shortArrayList = FileUtil.*readFileAsBinary*(filePath); // 文件转成short之后的short数组

**3.FileUtil 静态变量**

private static String *fileDirectoryPath* = "D:\\课件\\软件构造\\filetochart\\src\\main\\resources\\static\\uploadfile\\";//存储文件的目录路径

**4. FileUtil readFileAsBinary()方法**

String filePath = filePathPara; // 文件路径  
File file = new File(filePath); // 文件  
int sizeOfEachRead = 1024; // 每次读取的字节数  
byte[] fileBytesArray = new byte[sizeOfEachRead]; // 一个byte为8字节，两个为16字节， 刚刚好为一个short的大小  
int read\_num = 0; // 每次读取的个数，为DataInputStream.read()函数的返回值  
ArrayList<Short> shortArrList = new ArrayList<>(); // 存储转换后的short  
Short tempShort = 0; // 存储函数返回值的temp变量  
byte b1 = 0; // 需要转化的第一个字节  
byte b2 = 0; // 需要转化的第二个字节

FileInputStream fileInputStream = new FileInputStream(file);// FileInputStream 实例

DataInputStream dataInputStream = new DataInputStream(fileInputStream); // DataInputStream 实例

**5. FileUtil getShort()函数**

byte b1 = b1Para; // 需要转化的第一个字节  
byte b2 = b2Para; // 需要转化的第二个字节

**6. FileUtil saveFile()函数**

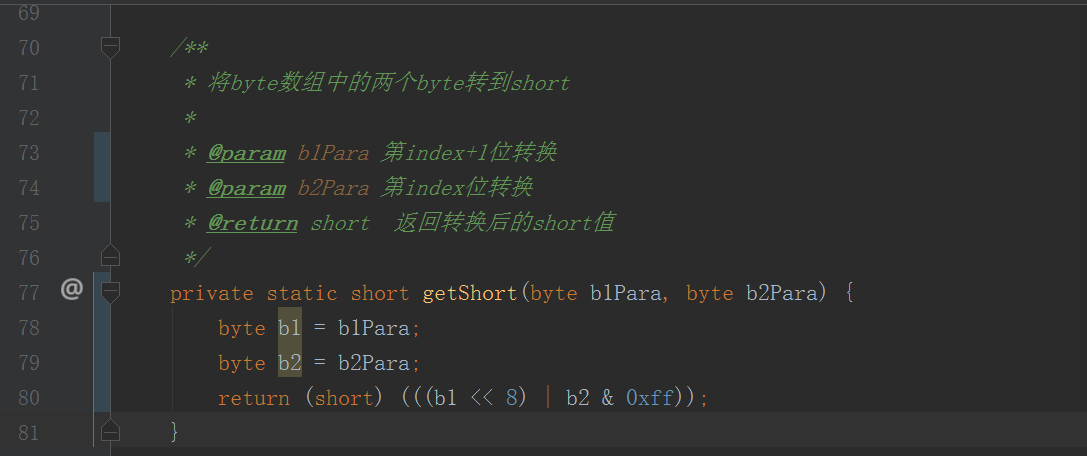
MultipartFile multipartFile = multipartFilePara; // 文件主体  
String fileName = fileNamePara; // 文件名

BufferedOutputStream out = new BufferedOutputStream(  
 new FileOutputStream(new File(*fileDirectoryPath* + fileName))); // 缓冲输出流

## 处理算法介绍

这里主要介绍文件处理成short的算法以及前端对波形进行压缩/扩展的算法：

1. 文件处理成short数组：

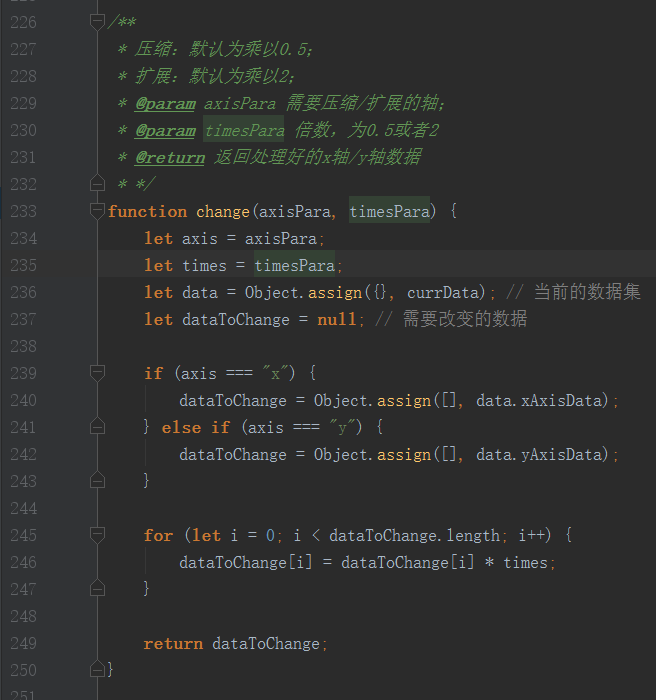


对于文件转成byte数组后，数组中两个连续的byte, b2为第index位，b1位第index+1位，则将b1左移8位，b2与0xff进行与运算；二者相或，再强制转成short;

2. 波形压缩/扩展

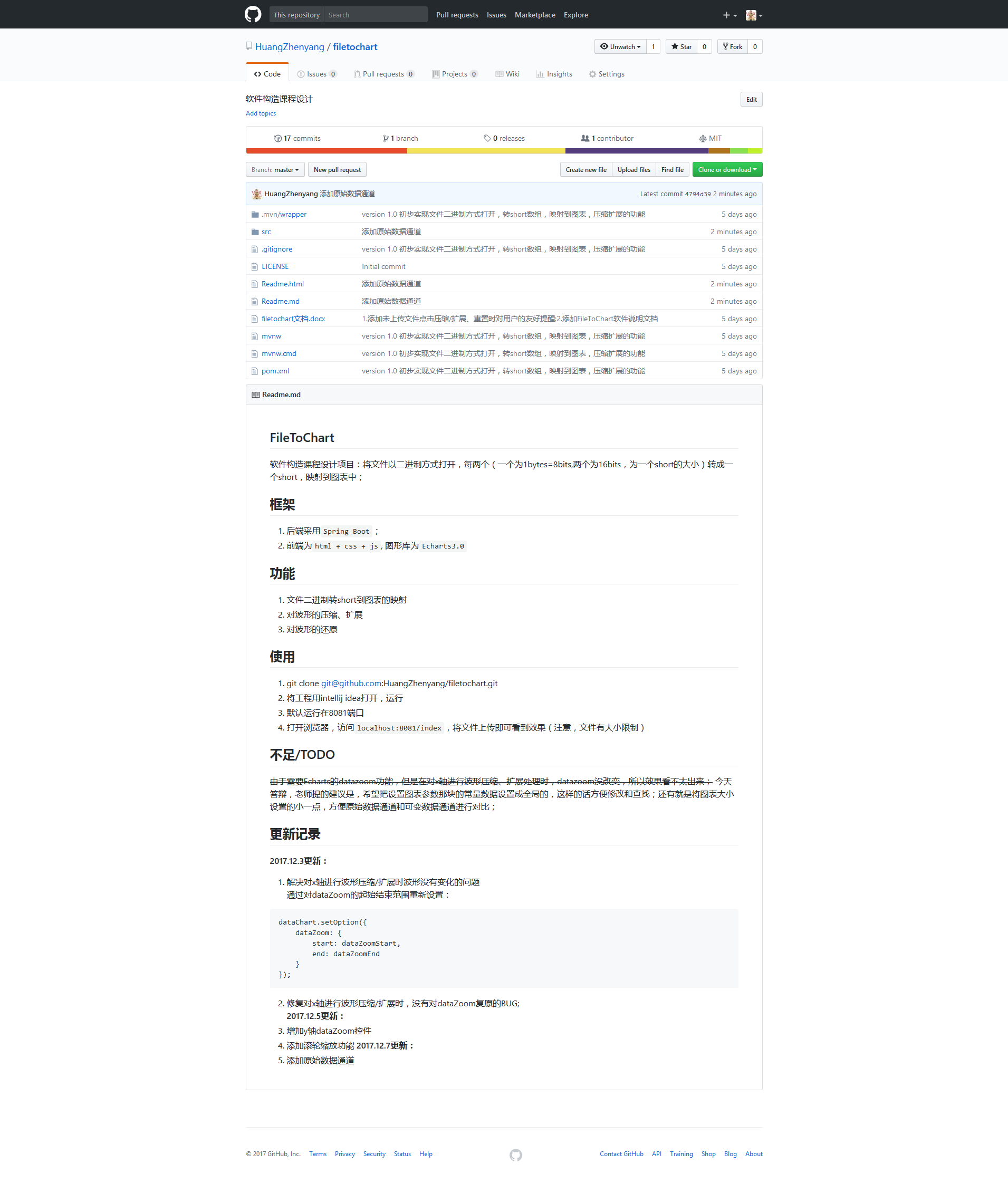
监听按钮点击，当按钮被点击时，获取下拉框（x轴/y轴）的值axisVal；如果点击的按钮为压缩，就将需要需要变换的轴的值及其数据传入change()函数，再判断axisVal的值是x还是y，从而改变数据集中国相应的轴的数据；最后调用setDataToChart()函数将数据置入图表中；





# 五 代码

项目代码在：https://github.com/HuangZhenyang/filetochart



# 六 测试

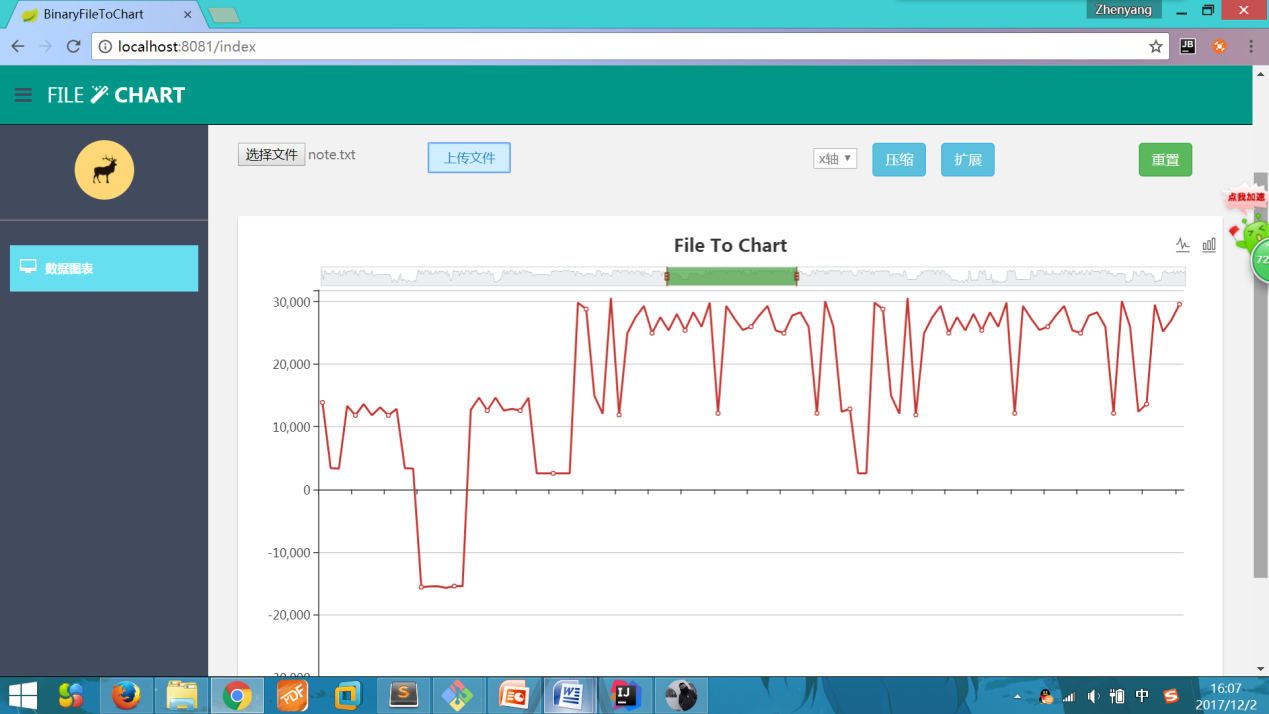
测试1：

测试功能：文件上传

输入：文件

预期输出：图表显示

实际输出：



结果：通过

测试2：

测试功能：文件保存

输入：文件上传

预期输出：文件保存成功

实际输出：



结果：通过

测试3：

测试功能：测试文件上传后，前端显示文件的信息等；

输入：文件上传

预期输出：正常显示文件的信息，上传时间；

实际输出：



结果：通过

测试4：

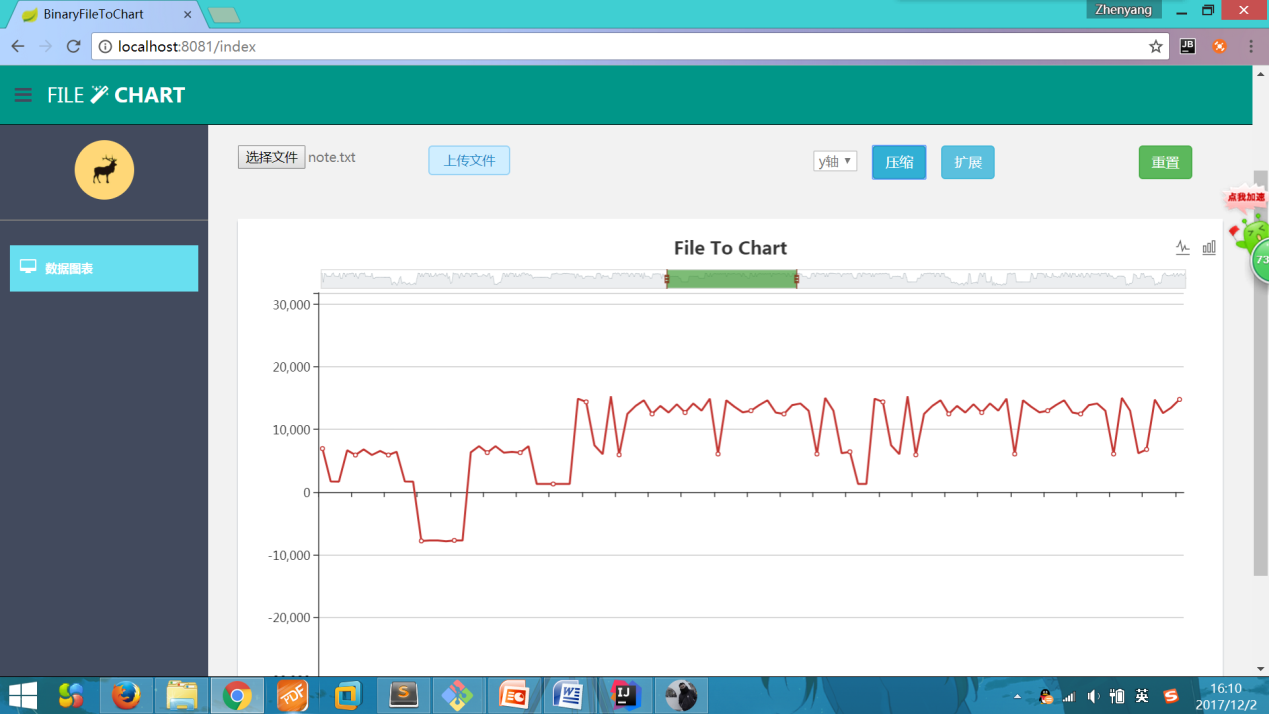
测试功能： 测试波形压缩与扩展

输入： 文件上传，点击压缩/扩展按钮

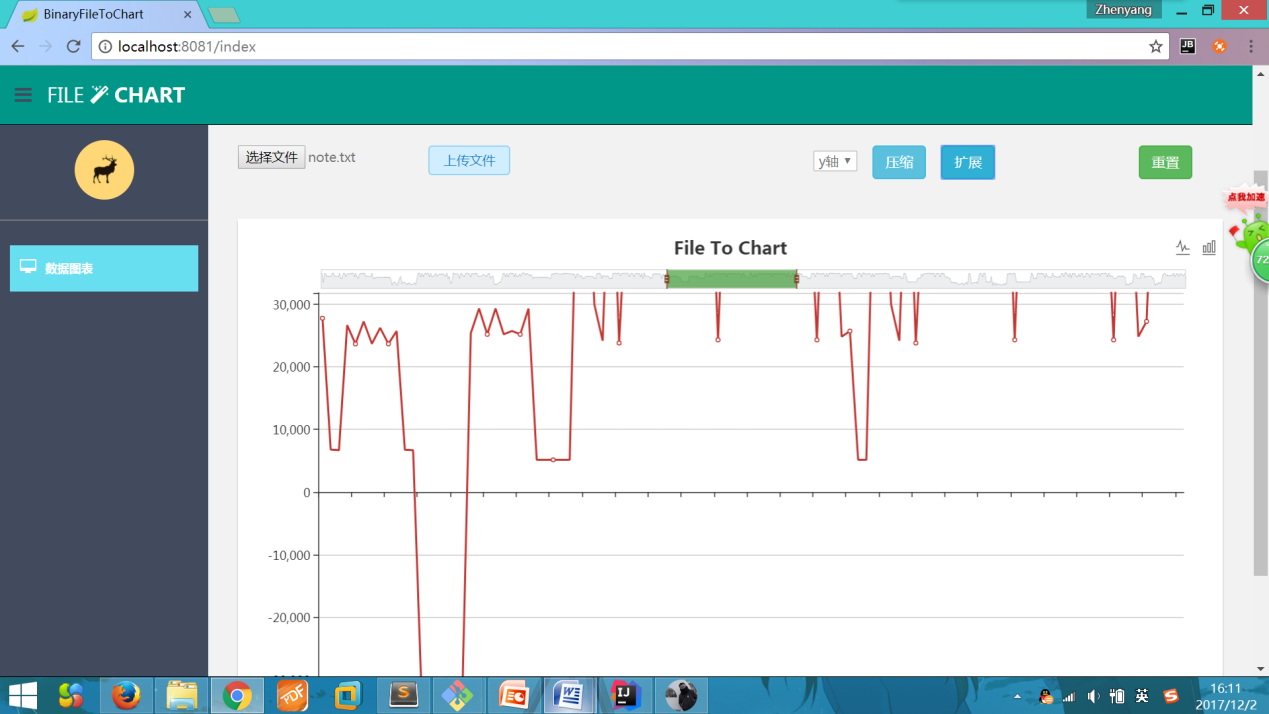
预期输出： 波形被压缩/扩展

实际输出：

压缩：



扩展：



结果：通过

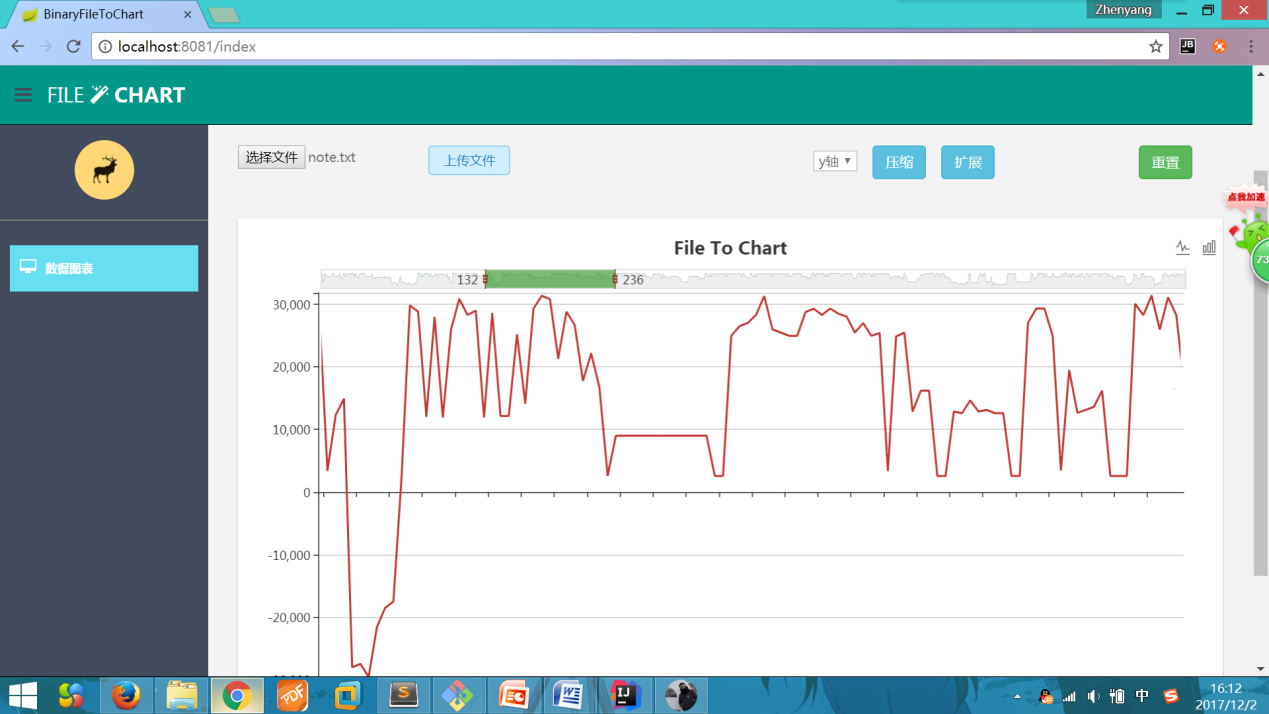
测试5：

测试功能： 图表拖动

输入： 拖动图表

预期输出： 图表显示其他部分

实际输出：



结果：通过

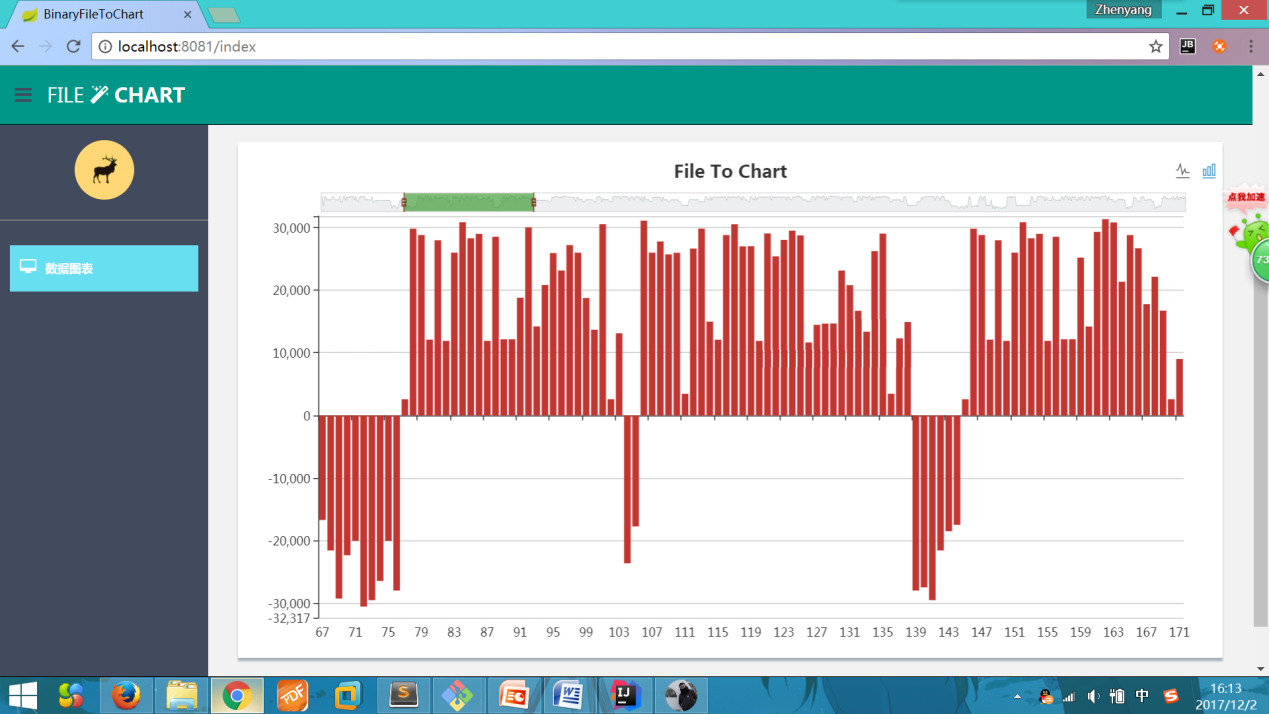
测试6：

测试功能： 折线图、柱状图切换

输入： 点击切换折线图、柱状图按钮

预期输出： 成功切换

实际输出：



结果：通过

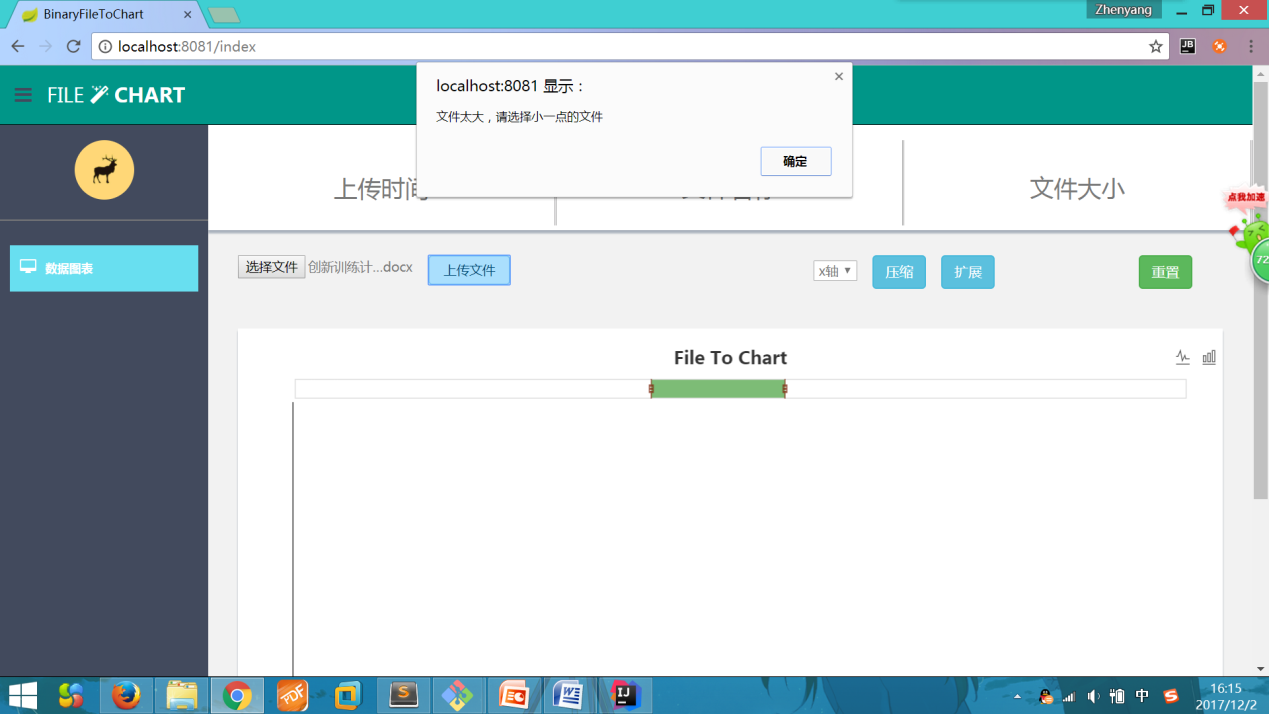
测试7：

测试功能： 文件太大时对用户提示

输入： 上传一个大文件

预期输出： 友好提示用户重新选择

实际输出：



结果：通过

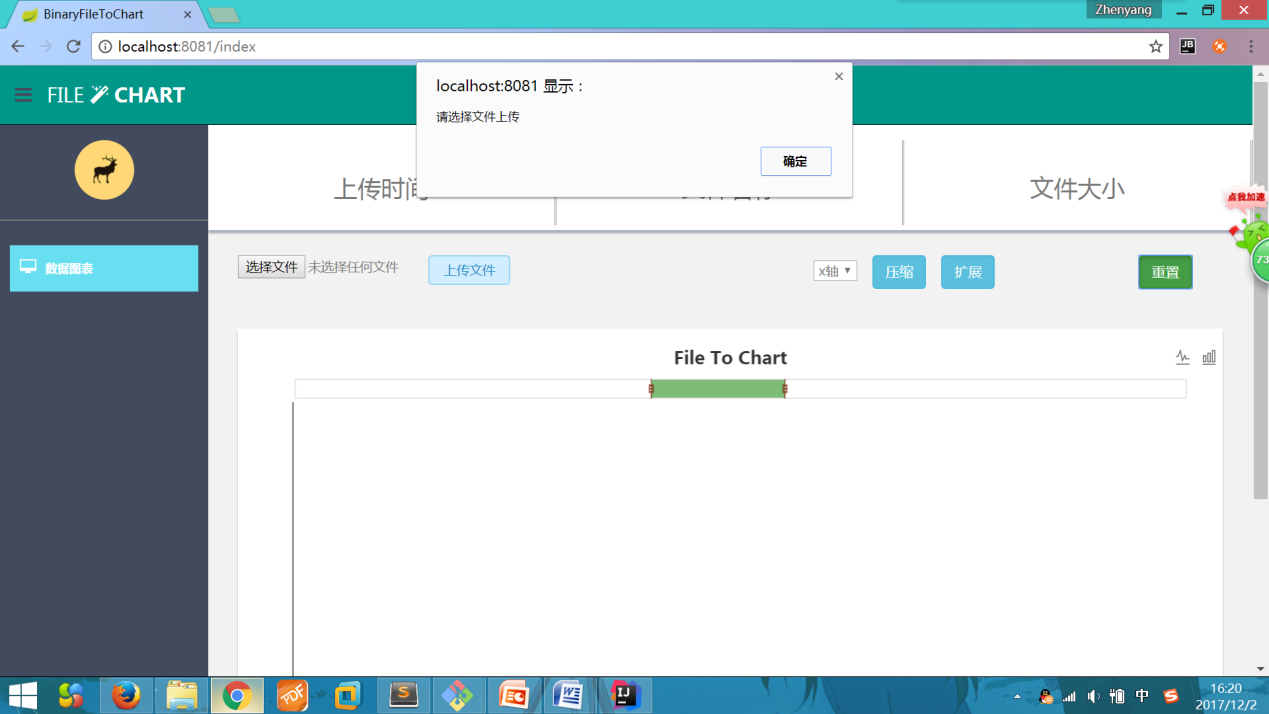
测试8：

测试功能： 测试未上传文件便点击压缩/扩展、重置等按钮时，对用户友好提示

输入： 未选择文件上传，点击压缩/扩展、重置等按钮

预期输出： 对用户友好提示

实际输出：



结果：通过

测试9：

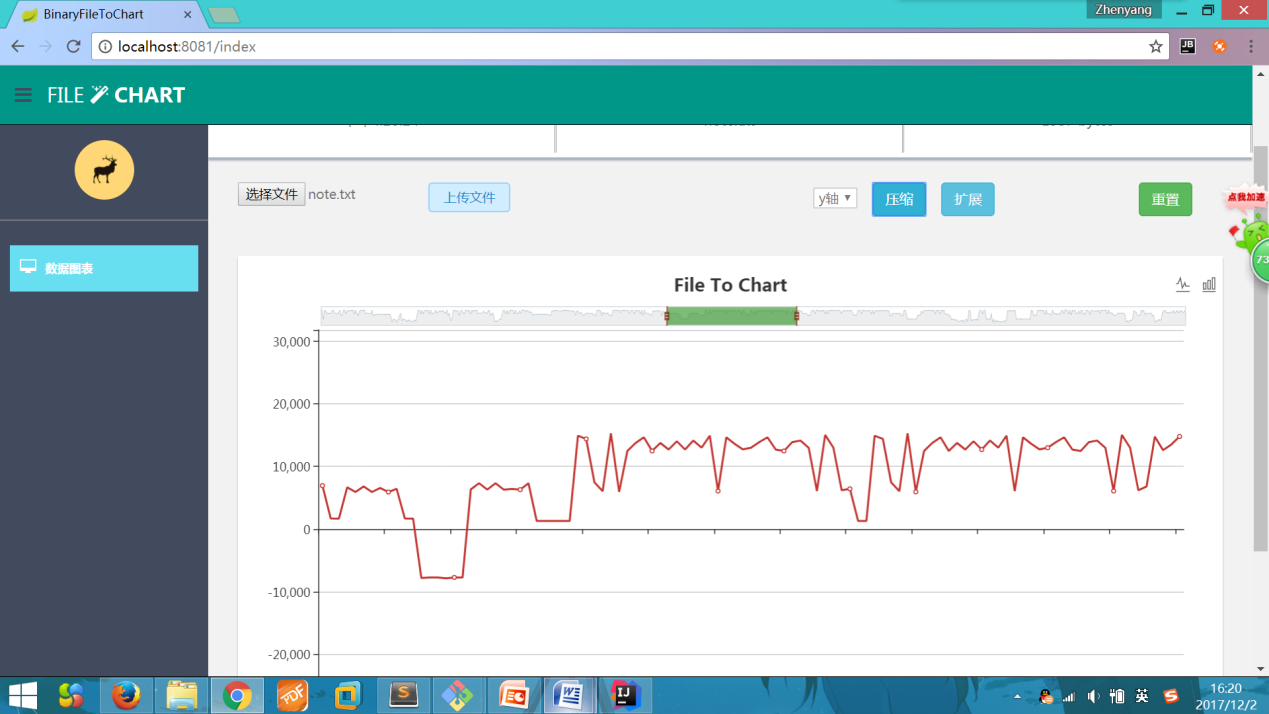
测试功能： 测试图表重置

输入： 对波形进行压缩/扩展处理，然后点击重置按钮

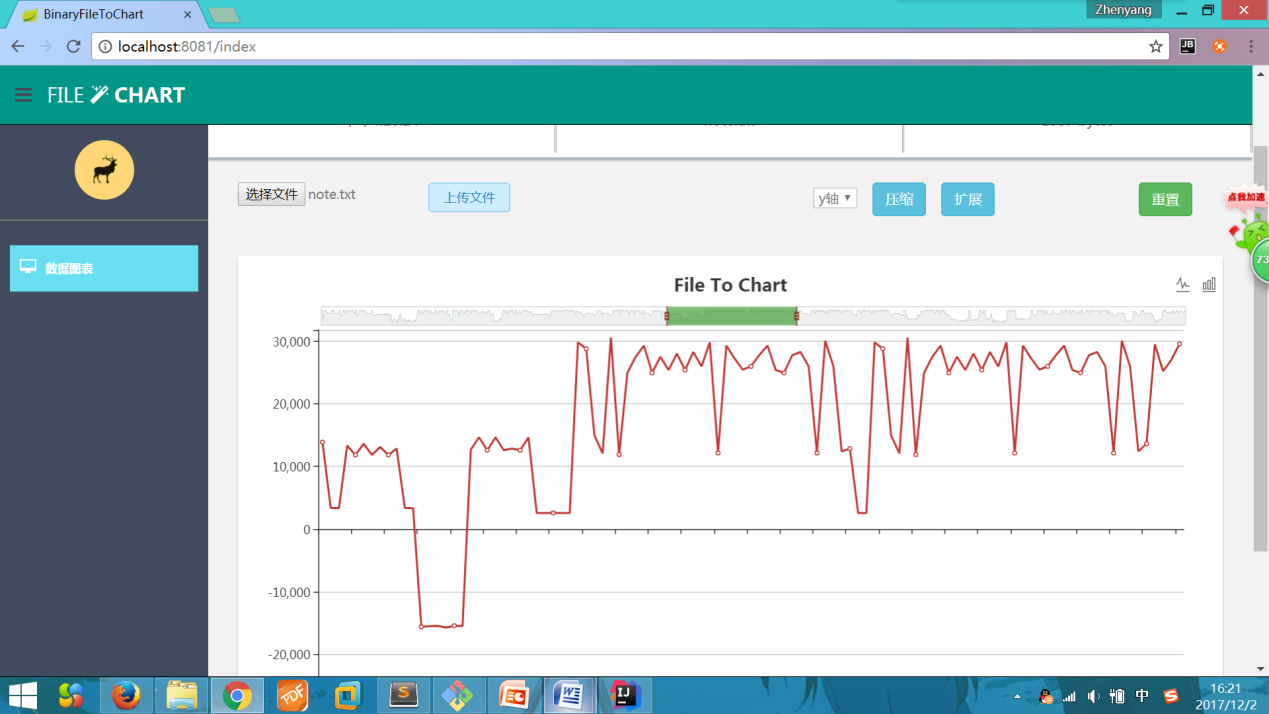
预期输出： 波形被重置

实际输出：

压缩：



重置：



结果：通过