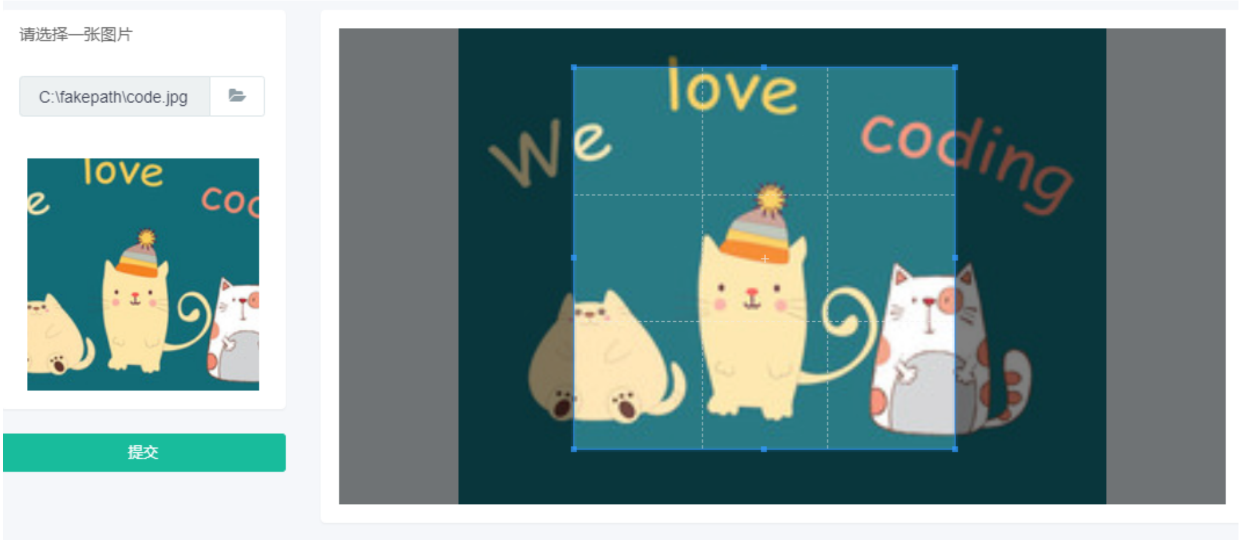
头像，背景图片上传详解

上传头像和上传背景图片的思路都是一样的，所以这里选讲一下上传头像就行。

上传头像页面样式：



上传头像主要分成两大部分：

1. 上传后，将图片存储在本地项目中，然后将图片的url存储到数据库里面，这样以后就可以根据数据库的地址来加载图片了。
2. 图片剪切功能，图片的剪切功能是使用了jquery的开源插件cropper。这个你百度一下就可以去github里面看到了。

当你截取好图片点击上传时，前端页面会将你截取的图片转化为二进制字符串流，再发送给后端，后端先接收二进制字符串流，将其存储下来，然后将存储的路径保存到数据库即可。

html代码：

```
<html xmlns:th="http://www.thymeleaf.org">
<head th:include="lay :: htmlhead" th:with="title='favorites'">
</head>
<body>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script th:src="@{/vendor/toastr/toastr.min.js}"></script>
<section>
    <link rel="stylesheet" th:href="@{/vendor/cropper/cropper.min.css}" />
    <div class="content-wrapper">
        <h3>
            上传头像
            <small>选择一个心仪的头像代表自己</small>
        </h3>
        <div class="container-fluid">
            <div class="row">
```

```
                <div class="col-lg-3 col-md-3">
                    <div class="panel">
                        <div class="panel-heading">
                            请选择一张图片
                        </div>
                        <div class="panel-body">
                            <div class="form-group">
                                <input id="cropper-file" type="file" class="form-control" style="position: absolute; clip: rect(0px, 0px, 0px, 0px);"/>
                                <div class="bootstrap-filestyle input-group">
                                    <input id="cropper-file-url" type="text" class="form-control" disabled=""/>
                                    <span class="group-span-filestyle input-group-btn">
                                    <label for="cropper-file" class="btn btn-default"> <!-- 使用for绑定前面的表单元素input，这样点击label就可以实现操作input-->
                                       <span class="fa fa-folder-open text-muted"></span>
                                    </label>
                                 </span>
                                </div>
                            </div>
                            <br />
                            <div id="cropper-preview" class="cropper-preview" data-text="截取的图像"></div>
                        </div>
                    </div>
                    <button id="cropper-submit" class="btn btn-primary btn-block" disabled="disabled" type="submit">提交</button>
                </div>
                <div class="col-lg-9 col-md-9">
                    <div class="panel">
                        <div class="panel-body">
                            <div class="cropper-area">
                                <img id="cropper-image" src="" alt=""/> </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>
<script th:src="@{/vendor/cropper/cropper.min.js}"></script>
<script>
    $(function(){
        $("body").removeClass("offsidebar-open");

        var $cropperImage = $('#cropper-image');
        var $cropperFile = $('#cropper-file');
        var $cropperFileUrl = $('#cropper-file-url');
        var $cropperSubmit = $('#cropper-submit');
        var $cropperPreview = $('.cropper-preview');
        var URL = window.URL || window.webkitURL;
        var blobURL;
        var croppable = false;

       // alert("变量初始化结束  "+URL);
        $cropperSubmit.prop('disabled', true); //设置disable的属性为true

        //jquery 图片剪辑设置
        $cropperImage.cropper({
            aspectRatio: 1 / 1,
            viewMode: 1,
            preview: $cropperPreview,
            background: false,
            mouseWheelZoom: false,
            built: function () {
                croppable = true;
            }
        });
        //alert("剪辑图片设置结束");

        if(URL){
            //当cropperFile发生变化时，进行改变
            $cropperFile.change(function(){
                var files = this.files;
                var file;
                if(!$cropperImage.data('cropper')){ //取出名为cropper的数据
                    return;
                }
                //alert($cropperImage.data('cropper'));
                if(files && files.length){
                    file = files[0];
                    if(/^image\/\w+$/.test(file.type)){
                        blobURL = URL.createObjectURL(file);
                        $cropperFileUrl.val($cropperFile.val());
                        $cropperSubmit.prop('disabled', false);
                        $cropperImage.one('built.cropper', function(){ //
给选中的元素附加事件
                            URL.revokeObjectURL(blobURL);//释放路径
                        }).cropper('reset').cropper('replace', blobURL);
                        $cropperFile.val('');
                    }else{
                        window.alert('请选择一个图像文件。');
                    }
                }
            });
        }else{
            $cropperFile.prop('disabled', true);
        }
        function getrectangleCanvas(sourceCanvas){
            var canvas = document.createElement('canvas');
            var context = canvas.getContext('2d');
            var width = 200;
            var height = 200;

            canvas.width = width;
            canvas.height = height;
            context.beginPath();
            context.rect(0, 0, width, height);
            context.strokeStyle = 'rgba(0,0,0,0)';
            context.stroke();
            context.clip();
            context.drawImage(sourceCanvas, 0, 0, width, height);

            return canvas;
        }

        $cropperSubmit.on('click', function(){
            var croppedCanvas;
            var rectangleCanvas;
            if(!croppable){
                return;
            }
            croppedCanvas = $cropperImage.cropper('getCroppedCanvas');
            rectangleCanvas = getrectangleCanvas(croppedCanvas);
            var dataUrl =
rectangleCanvas.toDataURL("image/png").replace(/\+/g,'%2B');
            $.ajax({
                url : '/uploadHeadPortrait',
                data : 'dataUrl='+dataUrl,
                async:  false,  //发送同步请求，其他事件必须等待
                type : 'POST',
                dataType : "json",
                error : function(XMLHttpRequest, textStatus, errorThrown)
 {
                },
                success: function(response){
                    if(response.rspCode == '000000'){
                        $("#leftProfilePicture").attr("src",response.dat
a);
                        $cropperFile.prop('disabled', true);
                        $cropperSubmit.prop('disabled', true);
                        $(".cropper-area .cropper-container").hide(); //上
传成功以后，隐藏显示的图片
                        toastr.success('头像上传成功！', '操作成功');//jQuery
提示插件
                    }else{
                        toastr.error(response.rspMsg, '操作失败');
                    }
                },
                error: function(XmlHttpRequest, textStatus, errorThrown){
                    console.log(XMLHttpRequest);
                    console.log(textStatus);
                    console.log(errorThrown);
                }
            });
        });
    });
    /*]]>*/
</script>
</body>
</html>
```

后端代码其实很简单，只需要将前端数据接收就行。

```
@RequestMapping("/uploadHeadPortrait")
    public ResponseData uploadHeadPortrait(HttpServletRequest request,String dataUrl){
        try {
            System.out.println("11");
            String filePath = "C:\\Users\\Administrator\\Documents\\GitHub\\forum\\forum\\src\\main\\resources\\static\\userImage\\";
            //这里的路径需要是绝对路径，如果填了"./resource/static/userImage",那么图片就会存储到idea内置的tomcat中
            String fileName = UUID.randomUUID().toString() + ".png";//指定为png格式,给图片随机起一个名字
            String image = dataUrl;
            String savePath = "userImage/"+fileName;
            String header = "data:image";
            String[] imageArr = image.split(",");
            if (imageArr[0].contains(header)) {
                image = imageArr[1];
                Base64.Decoder decoder = Base64.getDecoder();
                byte[] decodedBytes = decoder.decode(image);
                FileUtil.uploadFile(decodedBytes, filePath, fileName);
                //FileUtil是新写的一个类，代码在后面.
                System.out.println("保存结束");
                User user = getSessionUser(request);
                if(user==null){
                    return new ResponseData(ExceptionMsg.FAILED);
                }
                userServiceImp.updateUserPhoto(savePath,user.getUserId());//存储头像路径
                user.setPhoto(savePath);
                System.out.println(user);
                setSessionUser(request,user);
            }
            System.out.println("头像地址"+savePath);
            System.out.println("上传结束");
            return new ResponseData(ExceptionMsg.SUCCESS,savePath);
        }
        catch (Exception e){
            System.out.println("错误");
            return new ResponseData(ExceptionMsg.FAILED);
        }
    }
```

其实找到这里，并没有结束，因为你会发现，刚刚上传完的图片不能立即显示，需要你重启springboot项目才能显示。此时网页上会显示404图片无法加载。
原因是你将图片放在了static下面 ，而这个时候项目没有重新扫描static，所以是加载不了的。
你需要写一个配置类：

```
package com.example.demo.web;

import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

/**
 * @ClassName MyPicConfig
 * @Description TODO
 * @Auther ydc
 * @Date 2019/2/14 17:36
 * @Version 1.0
 **/
//新增加一个类用来添加虚拟路径映射
@Configuration
public class MyPicConfig implements WebMvcConfigurer {

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("/userImage/**").addResourceLocations("file:C:\\Users\\Administrator\\Documents\\GitHub\\forum\\forum\\src\\main\\resources\\static\\userImage\\");
       registry.addResourceHandler("/userBg/**").addResourceLocations("file:C:\\Users\\Administrator\\Documents\\GitHub\\forum\\forum\\src\\main\\resources\\static\\userBg\\");
//addResourceHandler()是将相对路径填进来，"/userImage/**"表示static下面的userImag里面的全部内容，都映射到绝对路径为"C:\\Users\\Administrator\\Documents\\GitHub\\forum\\forum\\src\\main\\resources\\static\\userBg\\"的文件夹里面去查找，这样就可以立即找到了。
//我这里之所以用来两次是因为我上传头像和背景图的文件夹不是同一个，所以分别映射了两次。注意:file必须要加！！
    }
}
```

FileUtil类：

```java
package com.example.demo.utils;

/**
 * @ClassName FileUtil
 * @Description TODO
 * @Auther ydc
 * @Date 2019/1/12 19:31
 * @Version 1.0
 **/
import java.io.File;
import java.io.FileOutputStream;

/**
 * .
 */
public class FileUtil{

    /**
     * 获取文件类型
     * @param fileName
     * @return
     */
    public static String getFileExtName(String fileName) {
        if (fileName!=null ) {
            int i = fileName.lastIndexOf('.');
            if (i>-1) {
                return fileName.substring(i+1).toLowerCase();
            }else {
                return null;
            }
        }else {
            return null;
        }
    }
    //文件上传工具类服务方法

    public static void uploadFile(byte[] file, String filePath, String fileName) throws Exception{

        File targetFile = new File(filePath);
        if(!targetFile.exists()){
            targetFile.mkdirs();
        }
        FileOutputStream out = new FileOutputStream(filePath+fileName);
        out.write(file);
        out.flush();
        out.close();
    }
}
```

这次的代码有点多，文字有点少，主要是因为这部分基本都是像模板一样使用。