此文章已于 11:57:54 2018/3/7 发布到 HealerJean梦想博客

3、表单提交附件和上传多个附件

类别 DDKJ ; 附件上传

## 解释：当选择图片或者是附件的时候，其实就已经准备开始上传了只不过，我们要讲上传之后的实体对象取出相应的字段（也就是文件的存储位置）来发送给前端。然后进行整个表单的提交。这个时候提交的其实不过就是文件的名字和已经上传的文件的路径了

# 1、上传页面以及代码

@RequestMapping("upload")  
@ResponseBody  
public ResponseBean uploadPact(MultipartFile file){  
   
 String suffix = StringUtils.substringAfterLast(file.getOriginalFilename(),".");  
 String fileLink = "admin/crm/contact/" + UUIDGenerator.generate() + "." + suffix;  
  
 try {  
 OSSUploader.upload(fileLink,file.getInputStream());  
 } catch (Exception e) {  
 logger.error(e.getMessage(),e);  
 return ResponseBean.buildFailure("上传失败");  
 }  
  
 return ResponseBean.buildSuccess(fileLink);  
}

# 2、整个表单的提交

### 1、表单对应的实体类为

#### 解释：下面敢于附件的很多东西通过@Transient都不显示到数据库表中去，因为它其实是为了通过对象Contact 进行前台的接收对应的字段

@Entity  
@Table(name = "crm\_contact\_record")  
public class Contact implements Serializable {  
  
 public static final int ***ENABLE*** = 1; //可见  
 public static final int ***DISABLE*** = 0; //不可见  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.***IDENTITY***)  
 private Long id;  
 private Long adminId;  
 private Long customerId;  
 private String title;  
 private String contactName;  
 private Integer type;  
 private String content;  
 private Integer isVisible;  
 private Date contactTime;  
 private Date cdate;  
  
 @Transient  
 private String typeDesc;  
 @Transient  
 private String adminName;  
 @Transient  
 private String name;  
 @Transient  
 private String attachPics; //前端提交的附件的路径，是一个有，号的字符串  
 @Transient  
 private String attachFileNames; //前端提交的附件的名字  
 @Transient  
 private List<ContactRecordAttachment> attachmentList; //在修改实体的时候，查找到附件，放到这个list集合中

## 2、js表单提交

$scope.saveContact = function(){  
 var attachPics = [];  
 var attachFileNames = [];  
 for(var i = 0 ; i < $scope.attachs.length ; i ++ ){  
 if($.trim($scope.attachs[i].attachPic) != ''){  
 attachPics.push($scope.attachs[i].attachPic);  
 attachFileNames.push($scope.attachs[i].contentPicName);  
 }  
 }  
  
 $scope.contact.attachPics = attachPics.join(",");  
 $scope.contact.attachFileNames = attachFileNames.join(",");  
 $http.post(***ctx*** + '/contact/save',$scope.contact)  
 .success(function(data, status) {  
 if(data.success){  
 toastr.success('保存联系记录信息成功','提示');  
 $location.path("/contact#/list").replace();  
 ngDialog.closeMouseDialog();  
 } else {  
 ngDialog.alert(data.message);  
 }  
 })  
 .error(function(data, status) {  
 ngDialog.alert('操作失败');  
 });  
};

# 3、提交表单，后台接收

### 1、controller

@RequestMapping("save")  
@ResponseBody  
public ResponseBean save(Contact contact){  
 try {  
 SysAdminUser user = AppSessionHelper.getSessionUser();  
 if(user != null){  
 contact.setAdminId(user.getId());  
 }  
 contactService.saveContact(contact);  
 return ResponseBean.buildSuccess();  
 } catch (AppException e) {  
 return ResponseBean.buildFailure(e.getMessage());  
 }  
}

### 2、service 保存表单和附件的实体

@Override  
public void saveContact(Long Id, Long adminId, Long customerId, String title, String contactName, Integer type,  
 String content, String linkTime, String attachPics, String attachFileNames) {  
  
 Date lingTimeDate = null;  
 if(linkTime.contains("-")){  
 lingTimeDate = DateHelper.convertString2Date(linkTime,DateHelper.***YYYY\_MM\_DD***);  
 }else{  
 lingTimeDate = DateHelper.convertString2Date(stampToDate(linkTime),DateHelper.***YYYY\_MM\_DD***);  
 }  
 Contact contact = new Contact(adminId, customerId, title, contactName, type,  
 content, Contact.***ENABLE***, lingTimeDate);  
 if(Id == null) {  
 contact.setCdate(new Date());  
 }else {  
 contact = contactDAO.findOne(Id);  
 contact.setContactName(contactName);  
 contact.setType(type);  
 contact.setContent(content);  
 contact.setTitle(title);  
 contact.setContactTime(lingTimeDate);  
 contact.setCustomerId(customerId);  
 contactRecordAttachmentDAO.deleteByContactId(Id); /每次保存数据先将之前的附件删除  
 }  
 Contact con = contactDAO.save(contact);  
  
 String[] pics = attachPics.split(",");  
 String[] fileNames = attachFileNames.split(",");  
 //保存附件  
 for (int i = 0; i < pics.length; i++) {  
 ContactRecordAttachment attachment = new ContactRecordAttachment();  
 attachment.setFileUrl(pics[i]);  
 attachment.setFileName(fileNames[i]);  
 attachment.setContactId(con.getId());  
 attachment.setCdate(new Date());  
 contactRecordAttachmentDAO.save(attachment);  
 }  
  
}

### 3、针对在更新页面的时候，提交附件，替换掉原理的附件，这个原理，个人认为应该像删除数据那样，通，目前原因未知

# 4、更新页面预览、通过id修改实体时，需要得到附件并预览

## 1、前端查找Contact 实体，并找出附件

if($routeParams.id != undefined){  
 $http.get(***ctx*** + '/contact/findContactById?id='+ $routeParams.id)  
 .success(function(data, status) {  
 if(data.success){  
 $scope.contact = data.result;  
 ***console***.log(data.result.attachmentList.length);  
 for(var i = 0 ; i < data.result.attachmentList.length ; i ++ ){  
 $scope.attachs.push({attachPic: data.result.attachmentList[i].fileUrl , btnText: '重新上传'});  
 }  
 $scope.contact.cdate = undefined;  
 } else {  
 ngDialog.alert(data.message);  
 }  
 })  
 .error(function(data, status) {  
 ngDialog.alert('加载联系记录信息失败');  
 });  
}

## 2、后端代码

@Override  
public Contact findContactById(Long id) {  
 Contact contact = ContactDAO.findOne(id);  
 contact.setAttachmentList(contactRecordAttachmentDAO.findByContactId(contact.getId()));  
 return contact;  
}

# 5、前端代码