**IUAP平台使用指南**

**附件组件**

用友网络科技股份有限公司

二○一八年五月

**此文档为保密文档，未经用友软件股份有限公司书面同意，不得向任何单位或个人提供、转让本文档中的任何内容，用友软件股份有限公司将保留对泄漏文档内容的起诉权利。**

**IUAP平台使用指南**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [**√**]正在修改  [] 正式发布 | 文件标识： |  |
| 当前版本： |  |
| 作者： |  |
| 完成日期： |  |

版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 作者 | 参与者 | 起止日期 | 备注 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

审核记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 姓名 | 职位 | 审阅日期 | 备注 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[第一章 概述 1](#_Toc515109650)

[1.1 业务场景 1](#_Toc515109651)

[1.2 开发方案 1](#_Toc515109652)

[1.2.1方案一： 1](#_Toc515109653)

[1.2.2方案二： 2](#_Toc515109654)

[1.3 应用效果 3](#_Toc515109655)

[第二章 示例配置 5](#_Toc515109656)

[2.1 IUAP配置 5](#_Toc515109657)

[2.2 数据库配置 5](#_Toc515109658)

[2.3 开发环境配置 6](#_Toc515109659)

[2.4 Nginx配置 7](#_Toc515109660)

[第三章 代码开发示例 8](#_Toc515109661)

[3.1 后端开发 8](#_Toc515109662)

[3.2 前端开发 8](#_Toc515109663)

# 概述

## 业务场景

开发者经常遇到需要附件上传、下载、删除、查看等开发需求（例如：某条单据录入过程中，需要上传与单据相关的文档附件；需要查询与某条单据相关的所有附件记录……），利用IUAP应用平台，开发者只需编写少量的代码即可方便快捷地实现这类需求。

IUAP提供了三种文件存储位置：Local、FastDFS、Alioss，开发者只需要选择一种做场景开发即可，本示例存储位置为Local。图

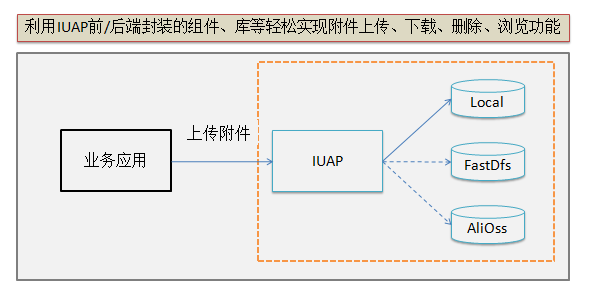


图 1.1.1 IUAP附件服务

## 开发方案

实际业务场景建模中，附件一般会与某一条记录关联，他们之间的关系如下图1.2.1所示。



图 1.2.1 记录与附件关系

也即：一条记录可以包含0-N个附件，一个附件必须属于一条记录。

IUAP提供了附件的管理服务，唯一需要的就是业务系统为附件提供唯一标记，用来记录附件是与那条记录相关联的。因此我们可以构建一张记录表example\_attachment”，用其主键Id字段与附件关联。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 名称 | 类型 | 主键 | 备注 |
| Id | 主键 | Varchar | 是 | Uuid |
| Code | 编码 | Varchar |  | 非空，唯一索引 |
| Name | 名称 | Varchar |  | 可空 |

示例设计功能如下。

* 新增单据
  + 填写表单
  + 上传附件
  + 删除附件
  + 下载附件
  + 查看附件。
* 编辑单据
  + 编辑表单
  + 查询与本记录相关联的附件列表
  + 上传附件
  + 删除附件
  + 下载附件
  + 查看附件。
* 删除单据
* 删除表单
* 删除与记录相关的所有附件

# 示例配置

## IUAP文件上传配置

双击Iuap主目录下“sysconfig”进行文件上传配置，本示例配置为local存储类型，修改完毕后保存配置，然后双击startup启动IUAP。



## 数据库配置

在IUAP的数据库中创建示例表“example\_attachment”。

**CREATE** **TABLE** `example\_attachment` (

`id` **varchar**(36) **COLLATE** utf8\_unicode\_ci NOT NULL,

`code` **varchar**(50) **COLLATE** utf8\_unicode\_ci NOT NULL,

`**name**` **varchar**(50) **COLLATE** utf8\_unicode\_ci **DEFAULT** NULL,

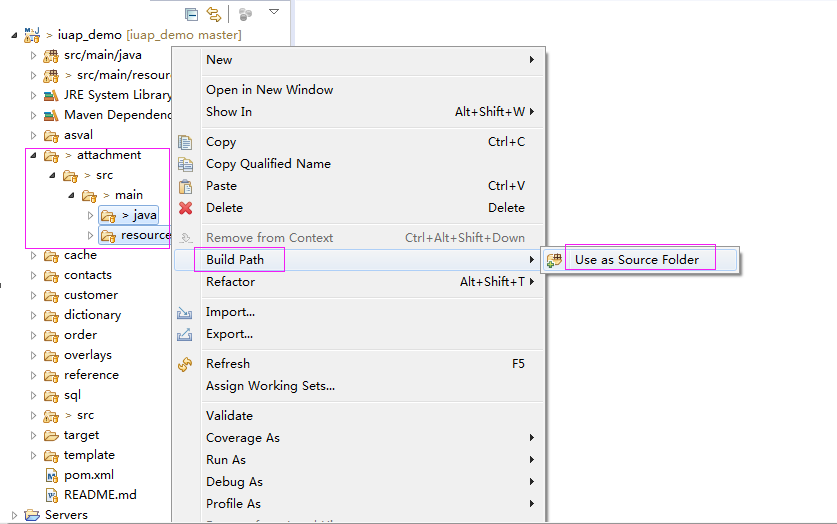
**PRIMARY** **KEY** (`id`),

**UNIQUE** **KEY** `idx\_code` (`code`)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8 **COLLATE**=utf8\_unicode\_ci;

## 开发环境配置

用eclipse载入示例工程，选中attachmen示例下的java和resources目录，然后右键—》Build Path—》Use as Source Folder，将示例工程载入编译环境。



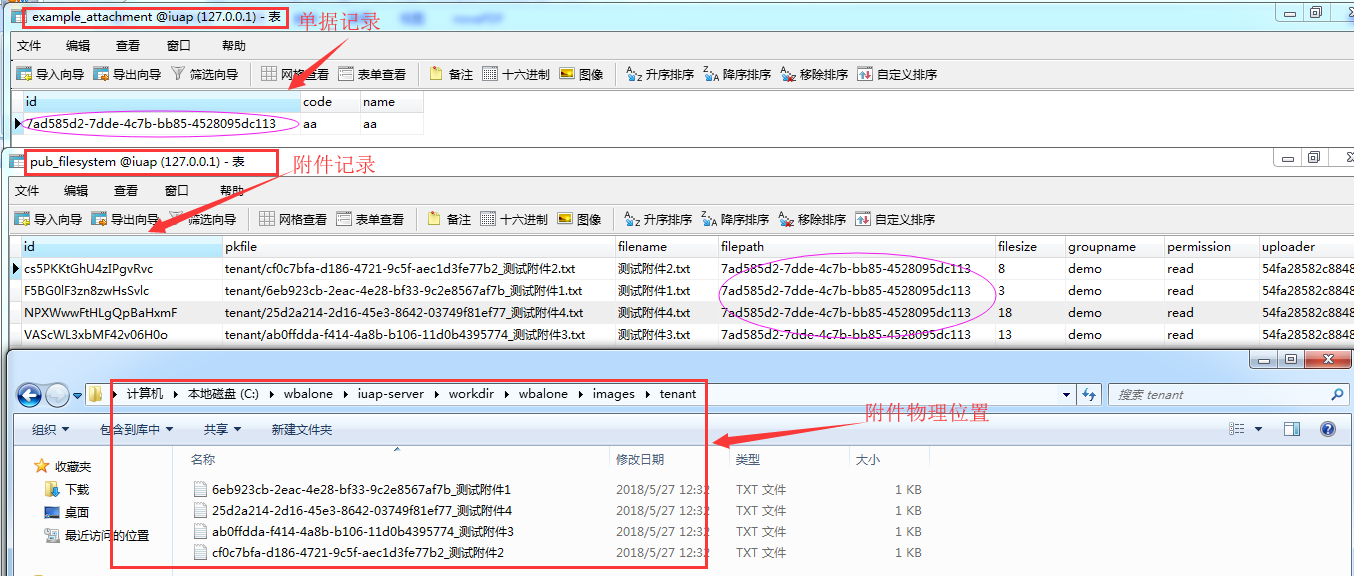
检查项目无问题后，右击iuap\_demo –》Run As —》run on server。

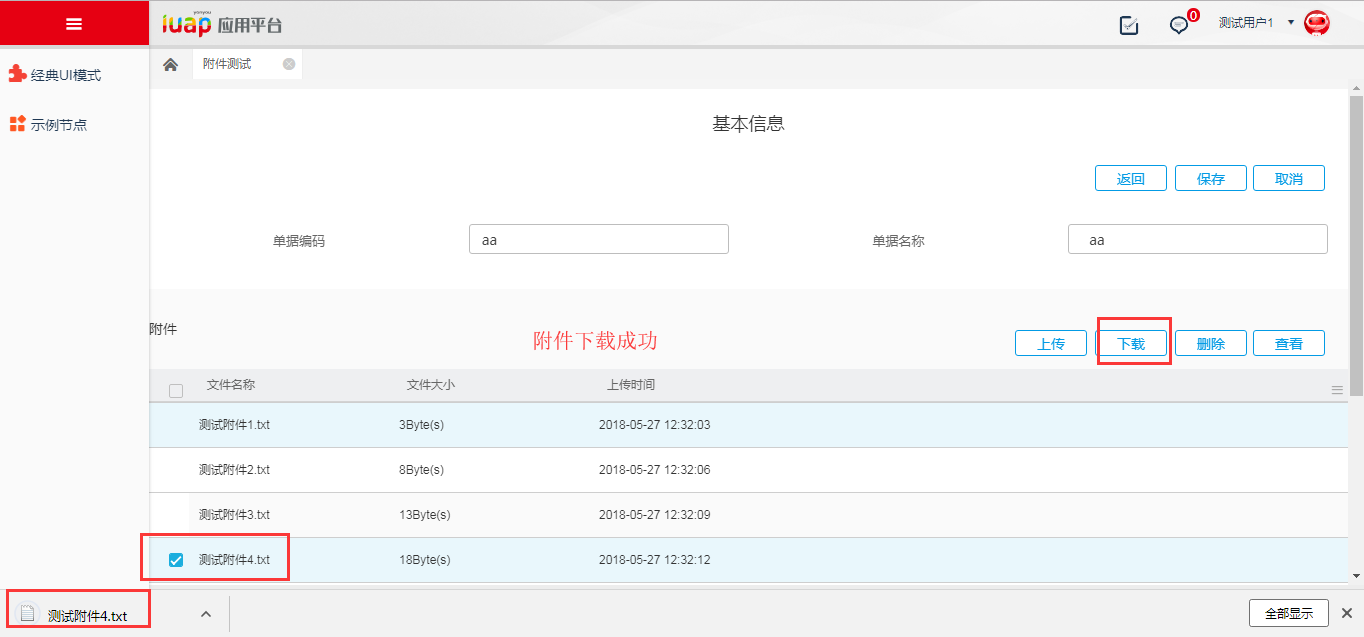
## 功能验证

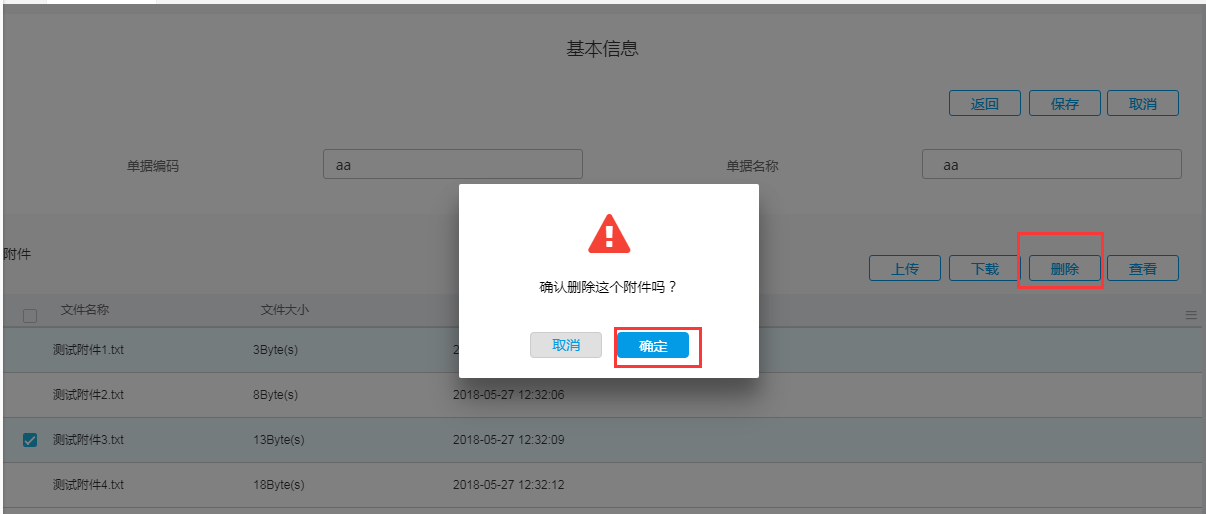














# 代码开发示例

## 后端开发

后端开发请参照单表示例（完成entity，dao，service，controller，xml等文件的编写）。与单表示例不同点是，本示例的controller提供多了两个@RequestMapping。

* “/uuid”提供了一个唯一Id，返回给前端，便于记录id和附件关联。
* “/insert”需要前端先调用uuid，然后给前端对象id，再传给后端，执行记录的insert。

@RequestMapping(value = "/uuid", method = RequestMethod.***GET***)

**public** @ResponseBody Object getUUID() {

**return** buildSuccess(UUID.*randomUUID*().toString());

}

@RequestMapping(value = "/insert", method = RequestMethod.***POST***)

**public** @ResponseBody Object insert(@RequestBody List<ExampleAttachment> list) {

service.insert(list.get(0));

**return** buildSuccess();

}

## 前端开发

### 编写meta.js

meta.js中定义记录对象和附件信息对象，可根据实际业务定义对象。

var meta = { //记录对象

    meta: {

        id: {

            type: 'string',

            required: true,

            nullMsg: 'Id不能为空!'

        },

        code: {

            type: 'string',

            required: true,

            nullMsg: '编码不能为空!'

        },

        name: {

            type: 'string',

            required: true,

            nullMsg: '名称不能为空!'

        }

    }

};

var fileDataMeta = {//附件对象

    meta: {

        id: { type: 'string' },//主键

        filepath: { type: 'string' },//文件名称

        filesize: { type: 'string' },//文件大小

        filename: { type: 'string' },//文件名称

        uploadtime: { type: 'string' },//上传时间

        groupname: { type: 'string' },//

        url: { type: 'string' }//URL

    }

};

### 编写attachment.html文件

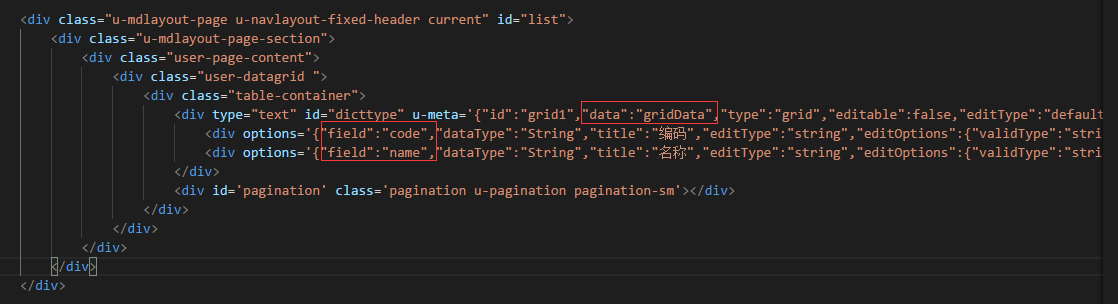
attachment.html 内是一个整体的div

<div class="contain" id="user-mdlayout"></div>

中间包含三部分：

* 默认显示部分：显示记录表格内容
* 表单和附件部分：id="divForm"，默认隐藏，新增或者编辑时候显示。
* 附件上传对话框 <div id="dialog\_uploadfile2">，默认隐藏，上传附件时显示。

开发人员可根据实际业务需求改写表单记录对象，附件对象不用改写。





### 编写attachment.css文件

文件中定义的样式可以根据实际情况修改。

### 编写attachment.js文件

* 添加ajax对上传文件的封装，

'/iuap-saas-filesystem-service/resources/js/ajaxfileupload.js',

'/iuap-saas-filesystem-service/resources/js/ossupload.js',

'interfaceFileImpl'

* 添加iuap文件服务的地址。

window.ctxfilemng = '/iuap-saas-filesystem-service/';

define(['text!./attachment.html',

'./meta.js',

'css!./attachment.css',

'../../config/sys\_const.js',

"../sever.js",

'/iuap-saas-filesystem-service/resources/js/ajaxfileupload.js',

'/iuap-saas-filesystem-service/resources/js/ossupload.js',

'interfaceFileImpl'], function (template) {

window.ctxfilemng = '/iuap-saas-filesystem-service/';

然后定义请求地址和一些模型对象。

var ctx = cookie.appCtx + "/ExampleAttachment";

var viewModel = {

draw: 1,

pageSize: 5,

uuid: "",

insertFlag: true, //默认是新增，点击编辑按钮就变成false

insertStatusFlag: false, //默认没有新增成功，新增成功后变成ture，

listRowUrl: ctx + "/list",  //列表查询URL

saveRowUrl: ctx + "/save",  //修改URL， 有id为修改 无id为新增

delRowUrl: ctx + "/delete",         //刪除URL

insertUrl: ctx + "/insert", //新增表单记录服务地址

uuidRul: ctx + "/uuid", //获取uuid的服务地址

gridData: new u.DataTable(meta), //记录的grid对象

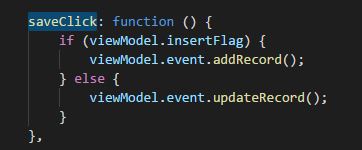
formData: new u.DataTable(meta), //记录的表单对象

fileData: new u.DataTable(fileDataMeta), //附件信息

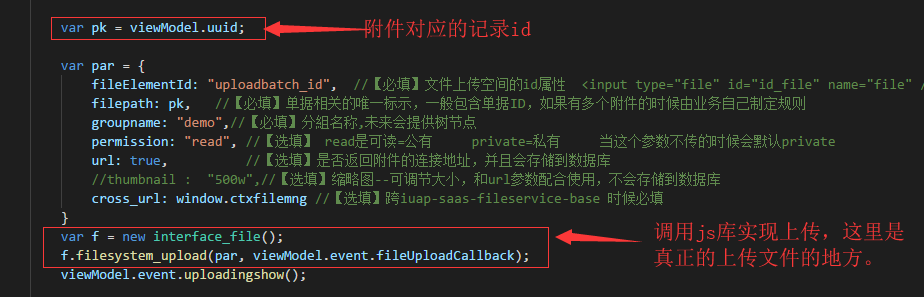
* 页面初始化：initGridDataList，主要是调用 “/list”填充记录的列表控件。
* 新增记录对象：addClick，显示新增div，清除模型，生成模型记录，获取uuid待用。



* 保存记录对象：saveClick，判定是新增还是编辑记录。



* 上传附件：onOpenUploadWin(),先打开上传对话框，选择文件上传，如果记录没有保存，先保存记录。然后上传onFileUpload(), 上传关键是定义上传对象。



上传完毕后执行回调函数fileUploadCallback()。

* 下载附件：每次只能下载一个附件，根据id去执行下载，主要是动态写form对象，注意请求的地址，还有form对象要append到div中，代码如下。

//下载附件

fileDownload: function (rowId, rowIndex, e) {

var row = viewModel.fileData.getSelectedRows();

if (row == null || row.length == 0) {

u.messageDialog({

msg: "请选择要下载的附件",

title: "提示",

btnText: "OK"

});

return;

} else if (row.length > 1) {

u.messageDialog({

msg: "每次只能下载一个附件",

title: "提示",

btnText: "OK"

});

return;

}

var pk = row[0].getValue("id");

var form = $("<form>"); //定义一个form表单

form.attr('style', 'display:none'); //在form表单中添加查询参数

form.attr('target', '');

form.attr('enctype', 'multipart/form-data');

form.attr('method', 'post');

form.attr('action', window.ctxfilemng + "file/download?permission=read&stream=false&id=" + pk);

$('#user-mdlayout').append(form); //将表单放置在web中

form.submit();

},

* 查询某记录下所有的附件

传入记录的id，然后执行查询，回调fileQueryCallBack(), 刷新附件列表。

fileQuery: function (pk) {

var par = {

filepath: pk, //【选填】单据相关的唯一标示，一般包含单据ID，如果有多个附件的时候由业务自己制定规则

groupname: "demo",//【选填】[分組名称,未来会提供树节点]

cross\_url: window.ctxfilemng //【选填】跨iuap-saas-fileservice-base 时候必填

}

var f = new interface\_file();

f.filesystem\_query(par, viewModel.event.fileQueryCallBack);

},

fileQueryCallBack: function (data) {

if (1 == data.status) {//上传成功状态

viewModel.fileData.setSimpleData(data.data);

if (data.data.length) {

$('#noDataFile').hide();

} else {

$('#noDataFile').show();

}

} else {

$('#noDataFile').show();

//没有查询到数据，可以不用提醒

if ("没有查询到相关数据" != data.message) {

u.messageDialog({ msg: "查询失败" + data.message, title: "提示", btnText: "OK" });

} else {

viewModel.fileData.removeAllRows();

}

}

},

* 删除附件，指定需要删除的附件的Id，调用删除方法。

//附件删除

fileDelete: function (rowId) {

var rows = viewModel.fileData.getSelectedRows();

if (rows == null || rows.length == 0) {

u.messageDialog({ msg: "请选择要删除的附件", title: "提示", btnText: "OK" });

return

} else if (rows.length > 1) {

u.messageDialog({ msg: "每次只能删除一个附件", title: "提示", btnText: "OK" });

return

}

u.confirmDialog({

msg:

'<div class="pull-left col-padding u-msg-content-center" >' +

'<i class="fa fa-exclamation-triangle margin-r-5 fa-3x red del-icon" style="vertical-align:middle"></i><br/>确认删除这个附件吗？</div>',

title: '',

onOk: function () {

var pk = rows[0].getValue("id");

var par = {

id: pk,//【必填】表的id

cross\_url: window.ctxfilemng //【选填】跨iuap-saas-fileservice-base 时候必填

}

console.log("par:", par);

var f = new interface\_file();

f.filesystem\_delete(par, viewModel.event.fileDeleteCallBack);

}

});

},

//附件删除回调

fileDeleteCallBack: function (data) {

if (1 == data.status) {//上传成功状态

var rows = viewModel.gridData.getSelectedRows();

var currentData = rows[0].getSimpleData();

viewModel.event.fileQuery(currentData.id);

} else {

u.messageDialog({ msg: "删除失败" + data.message, title: "提示", btnText: "OK" });

}

},

* 查看附件，选中需要查看的附件记录，获取到url，然后open。

//查看

fileView: function (rowId) {

var rows = viewModel.fileData.getSelectedRows();

if (rows == null || rows.length == 0) {

u.messageDialog({ msg: "请选择要查看的附件", title: "提示", btnText: "OK" });

return

} else if (rows.length > 1) {

u.messageDialog({ msg: "每次只能查看一个附件", title: "提示", btnText: "OK" });

return

}

var url = rows[0].getValue("url");

console.log("url:", url);

if (url.indexOf("http://") >= 0 || url.indexOf("https://") >= 0) {

parent.open(url);

} else {

parent.open("http://" + url);

}

}