# HTML5新特性 -- Unit06

# 1.Multer中间件

## 1.1 概述

在实现文件上传时，表单必须存在以下属性：

· 表单的提交方式只能为POST

· 必须设置表单的enctype="multipart/form-data"属性

<form action="" method="post" enctype="multipart/form-data">
  
 ...
  
</form>

## 1.2 Multer中间件

Multer中间件用于处理存在<input type="file">类型的表单数据。

在通过Multer中间件进行文件上传时，上传文件的信息会自动存储到req.file或req.files属性中。

Multer中间件在文件上传时：

A.为避免文件名称冲突，将自动进行重命名

B.文件上传时没有扩展名

安装Multer

npm install --save multer

配置Multer

//加载Multer中间件
  
const multer = require('multer');
  
   
//创建Multer实例
  
const upload = multer({
  
 //dest:destination,目标
  
 //指定文件上传的位置
  
 dest:'upload'
  
});

自定义存储规则 -- diskStorage()方法

Multer实例.diskStorage({
  
 //用于定制上传目录的相关的规则
  
 //req参数代表当前的HTTP请求对象
  
 //file参数代表当前上传的文件对象
  
 //cb(callback),回调函数,按指定的规则实现上传
  
 destination:function(req,file,cb){
  
   
 },
  
 //用于定制上传文件名称的相关规则
  
 filename:function(req,file,cb){
  
   
 }
  
});

通过自定义规则创建Multer实例

let storage = multer.diskStorage({
  
 destination:function(req,file,cb){
  
   
 },
  
 filename:function(req,file,cb){
  
   
 }
  
});
  
const upload = multer({storage:storage});

Multer实例的方法

· single()方法

single()方法用于实现单文件上传，其语法结构是：

Multer实例.single('浏览框名称')

· array()方法

array()方法用于实现多文件上传，其语法结构是：

Multer实例.array('浏览框名称')

上传文件信息：

originalname:上传的原始名称（包含扩展名称）

filename:上传后的名称

path:上传后的路径及名称

mimetype:上传文件MIME类型

destination:上传文件的位置

size:上传文件的尺寸(以字节为单位)

· UUID

UUID(Universally Unique Identifier)，通用唯一标识符，其目的是为了保证分布式系统中每一个元素都存在唯一的标识信息。

安装

npm install --save uuid

使用

const uuid = require('uuid');
  
//基于时间戳的UUID
  
uuid.v1()
  
   
//基于随机数的UUID
  
uuid.v4()

· DataTransfer对象

DataTransfer对象是通过拖放事件的dataTransfer属性返回的结果，其包含了拖放的数据。其包含了setData()、getData()方法，还包含 files属性。

files属性将返回FileList对象，该对象既可以来源于表单中的 <input type="file">，也可能来源于用户的拖放操作，其语法结构是：

FileList DataTransfer对象.files

FileList对象存在length属性，用于获取包含的<input type="file">或拖放文件的数量。

· FormData对象

FormData对象提供以键/值对表示的表单数据，经过它的数据可以用AJAX提交。

创建FormData对象：

variable = new FormData()

append()方法

append()方法用于添加一个新值到已有的键名上，如果键名不存在，则自动创建，其语法结构是：

FormData.append(key,value)