

- Home (/blogs)
- News (/blogs)
- Mooc (/moocs)
- Admin

Search..



Zoemings FOLLOW (<https://github.com/zoemings/>)

Formidable模块的使用

发布时间：1 个月前(2016年7月16日) 作者：

1. `var form = new formidable.IncomingForm()` 创建Formidable.IncomingForm对象
2. `form.encoding = 'utf-8'` 设置表单域的编码
3. `form.uploadDir = "/my/dir";` 设置上传文件存放的文件夹，默认为系统的临时文件夹，可以使用`fs.rename()`来改变上传文件的存放位置和文件名
4. `form.keepExtensions = false;` 设置该属性为true可以使得上传的文件保持原来的文件的扩展名。
5. `form.type` 只读，根据请求的类型，取值'multipart' or 'urlencoded'
6. `form.maxFieldsSize = 2 * 1024 * 1024;` 限制所有存储表单字段域的大小（除去file字段），如果超出，则会触发error事件，默认为2M
7. `form.maxFields = 1000` 设置可以转换多少查询字符串，默认为1000
8. `form.hash = false;` 设置上传文件的检验码，可以有两个取值'sha1' or 'md5'.
9. `form.multiples = false;` 开启该功能，当调用`form.parse()`方法时，回调函数的files参数将会是一个file数组，数组每一个成员是一个File对象，此功能需要html5中multiple特性支持。
10. `form.bytesReceived` 返回服务器已经接收到当前表单数据多少字节
11. `form.bytesExpected` 返回将要接收到当前表单所有数据的大小

12. `form.parse(request, [callback])` 该方法会转换请求中所包含的表单数据，`callback`会包含所有字段域和文件信息，如：

```
form.parse(req, function(err, fields, files) {  
  
    // ...  
  
});
```

- 13) `form.onPart(part)`；你可以重载处理multipart流的方法，这样做的话会禁止field和file事件的发生，你将不得不自己处理这些事情，如：

```
form.onPart = function(part) {  
  
    part.addListener('data', function() {  
  
        // ...  
  
    });  
  
}
```

如果你只想让formidable处理一部分事情，你可以这样做：

```
form.onPart = function(part) {  
  
    if (!part.filename) {  
  
        // 让formidable处理所有非文件部分  
  
        form.handlePart(part);  
  
    }  
  
}
```

14. formidable.File 对象

- A. `file.size = 0` 上传文件的大小，如果文件正在上传，表示已上传部分的大小
- B. `file.path = null` 上传文件的路径。如果不想让formidable产生一个临时文件夹，可以在fileBegin事件中修改路径
- C. `file.name = null` 上传文件的名字
- D. `file.type = null` 上传文件的mime类型
- E. `file.lastModifiedDate = null` 时间对象，上传文件最近一次被修改的时间
- F. `file.hash = null` 返回文件的hash值
- G. 可以使用`JSON.stringify(file.toJSON())` 来格式化输出文件的信息

15. `form.on('progress', function(bytesReceived, bytesExpected) {})`；当有数据块被处理之后会触发该事件，对于创建进度条非常有用。

16. `form.on('field', function(name, value) {})`; 每当一个字段/值对已经收到时会触发该事件
17. `form.on('fileBegin', function(name, file) {})`; 在post流中检测到任意一个新的文件便会触发该事件
18. `form.on('file', function(name, file) {})`; 每当有一对字段/文件已经接收到, 便会触发该事件
19. `form.on('error', function(err) {})`; 当上传流中出现错误便会触发该事件, 当出现错误时, 若想要继续触发request的data事件, 则必须手动调用`request.resume()`方法
20. `form.on('aborted', function() {})`; 当用户中止请求时会触发该事件, socket中的timeout和close事件也会触发该事件, 当该事件触发之后, error事件也会触发
- 21) `form.on('end', function() {})`; 当所有的请求已经接收到, 并且所有的文件都已上传到服务器中, 该事件会触发。此时可以发送请求到客户端。