任务：GameManager

用途：成为所有游戏的模板

前端语言：react-JS

后端语言：node-js

后端模板：express

前后端路由：react-router

JS标准：ES6

学习内容：react（16）, ES6，react-router（4.2.0），node-js（8.9.1），express（4.12.3），

Webpack（提供前端打包服务，显著减少包大小），react-script

任务1（18.2.8）express架构下的**静态资源管理器**

功能：上传文件+显示

疑问：有了express，那react-router在哪里运用？前后端的交互？前后端的分离？[中间件到底是个什么概念？](#中间件是什么)express和express.router有什么区别？

前后端交互：通过create-react-app explorer\_client生成前端代码框架

通过express –t [] explorer\_server生成后端代码框架

前端框架结构分析：package.json为配置文件，定义了这个项目所需要的各种模块，以及配置信息。Scripts部分为运行脚本命令的npm命令行缩写，比如yarn build可以将依赖库重新initiate。Dependencies(项目运行依赖模块)/DevDependencies（项目开发依赖模块）分指定版本1.1.2，~1.2.2=1.2.x，^1.2.2=1.x.x。bin字段指定内部指令对应的可执行文件位置。

细节： 1. 支持不同文件格式（.jpg,.doc,.txt）的上传

2. 包括一定方式的浏览（图片浏览+文字浏览）

3. 支持刷新（上传后自动刷新）

4. 资源浏览器界面（界面设计）

代码描述：

**Package.json代码示例：**

"scripts": {

"preinstall": "echo here it comes!",

"postinstall": "echo there it goes!",

"start": "node index.js",

"test": "tap test/\*.js"

}

"devDependencies": {

"browserify": "~13.0.0",

"karma-browserify": "~5.0.1"

}

"name" : "foo",

"config" : { "port" : "8080" },

# Node Express获取参数的几种方式

Node配合Express框架获取参数主要有以下4种形式:

* req.body
* req.params
* req.param()
* req.query

## req.body

req.body在[官方文档](http://expressjs.com/4x/api.html#request)是没有提供的，但是它也是很好用的获取表单参数的方法。req.body是**配合Express中间件bodyParser()中间件**提供的。当bodyParser()中间件使用后，这个对象默认为 {},req.body可以获取到post到body中的内容。  
举个栗子：

|  |  |
| --- | --- |
| 1  2  3 | app.post('/user/signup',function(req,res){  var \_user = req.body.user;  }) |

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | // POST user[name]=tobi&user[email]=tobi@learnboost.com  req.body.user.name  // => "tobi"  req.body.user.email  // => [tobi@learnboost.com](mailto:tobi@learnboost.com)  // POST { "name": "tobi" }  req.body.name  //=> "tobi" |  |

## req.param(name)

req.param()是req.query、req.body、以及req.params获取参数的三种方式的封装。req.param(name)返回name参数的值。  
举个栗子：  
和上面的req.body方法类似

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | // POST name=tobi  app.post('/user?name=tobi',function(req,res){  req.param('name');  // => "tobi"  })  // ?name=tobi  req.param('name')  // => "tobi"  // /user/tobi for /user/:name  req.param('name')  // => "tobi" |

**Web开发中前端，后端和中间件之间的区别**

这是一个细节：

前端层 - >用户界面层通常由HTML，Javascript，CSS，Flash以及ASP.Net，经典ASP，PHP等各种服务器端代码组成。将其视为最接近用户在代码方面。

中间件，中间层 - >一层，通常被称为系统的“管道”部分。 Java和C＃是编写此部分的通用语言，可以被视为UI和数据之间的粘合，可能是Web服务或WCF组件或其他SOA组件。

后端层 - >数据库和其他数据存储通常处于此级别。 Oracle，MS-SQL，MySQL，SAP以及各种现成的软件，都是为数据最后处理的软件而设计的。

任何这些之间可能存在重叠，因为您可以将所有内容都像ASP.Net网站一样倾斜，ASP.Net网站使用内置的AJAX功能生成Javascript，而后面的代码可能包含数据库命令，使代码背后包含中间和后面-end层。 或者，可以使用VBScript作为使用ADO对象的所有层并将所有三个层合并成一个。