# Node.js

## 创建服务器

const http = **require**('http');

*//http.createServer()表示创建一个服务器*

let server = http.**createServer**(function(){ *//这个回调函数表示一有人访问这个服务器时就会调用这个函数*

**console**.**log**('有人来了');

});

*//监听——把服务器比作超市的话，监听就是超市服务人员等有人来的话对他进行服务*

server.**listen**(8080);

*//http://localost:8080*

## require和response

let server = http.**createServer**(function (require, response) { *//回调函数能接受两个参数*

*//require——客户端对服务器的申请*

*//response——服务器对客户端的响应*

**console**.**log**(require.url);

response.**write**("123"); *//向前台响应东西*

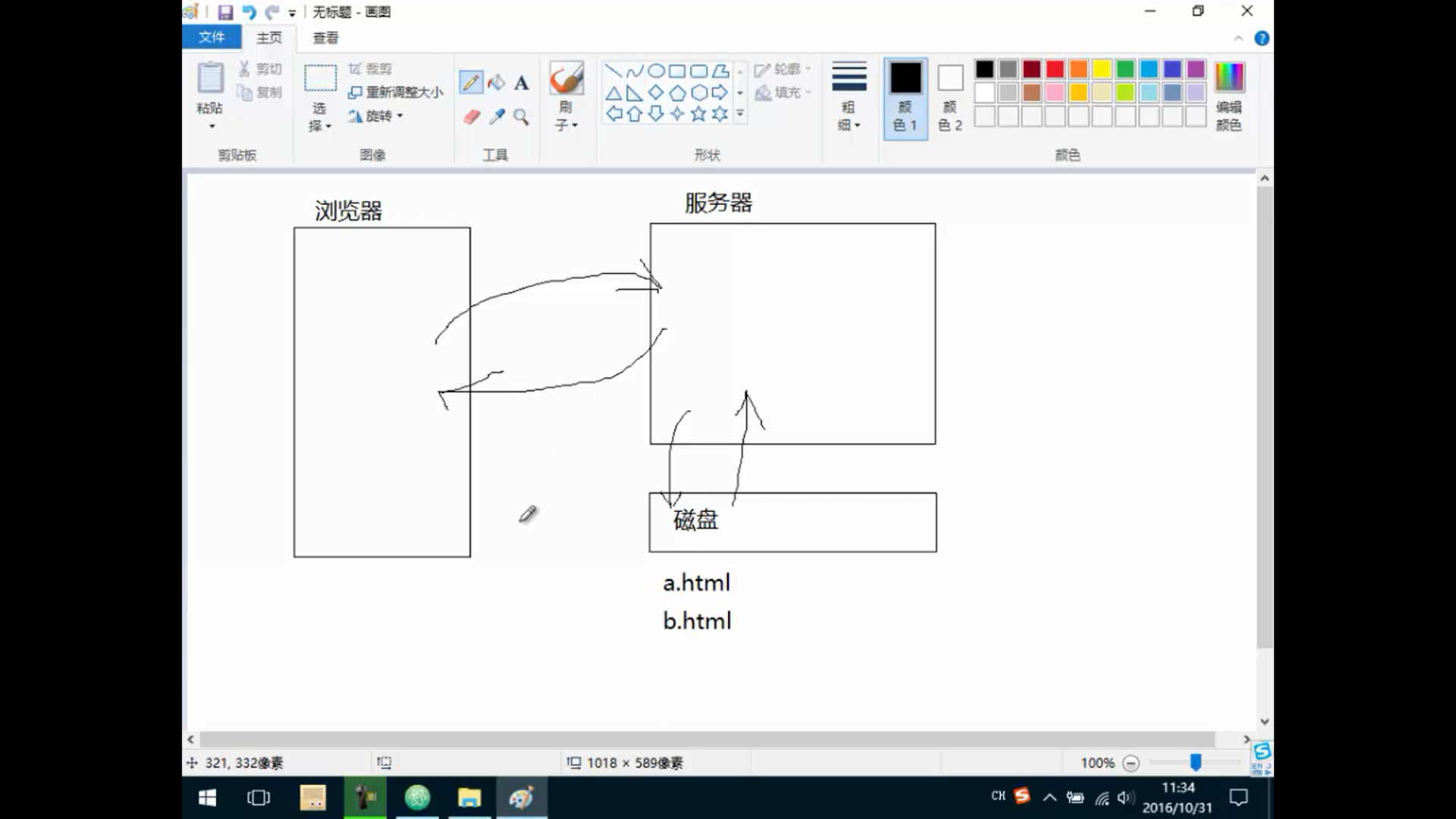
response.**end**(); *//此次服务结束了*

});

//require.url表示请求的url地址，比如说localhost:8080/index.html

那么reuqire.url == ‘/index.html’，它只表示一个url里所请求文件名的那部分

## 文件操作



*//文件操作：fs——fileStream*

const fs = **require**('fs');

*//readFile(文件名, 回调函数)*

fs.**readFile**('./www/1.txt', function(err, data){ *//err——错误信息， data——文件数据*

if(err){

**console**.**log**('读取失败');

}else{

*//console.log(data); //<Buffer 68 65 6c 6c 6f 20 6e 6f 64 65 2e 6a 73>——二进制数据*

**console**.**log**(data.**toString**()); *//toString()方法能把二进制数据转换成字符串数据*

}

}); *//读文件*

*//writeFile(文件名, 内容, 回调函数)*

fs.**writeFile**('./www/2.txt', "hello vue.js", function(err){ *//err——错误信息*

**console**.**log**(err);

}); *//写文件*

## 服务器和文件的综合操作

const http = **require**('http');

const fs = **require**('fs');

let server = http.**createServer**(function (req, res) {

*//假设客户端请求req.url = './index.html'*

*//则文件应该读取的路径名=>./www/index.html*

*//所以路径名应改为 => './www' + req.url*

let file\_name = './www' + req.url;

fs.**readFile**(file\_name, function (err, data) {

if (err) {

res.**write**('404');

} else {

res.**write**(data); *//这里没有使用toString()方法，因为是给浏览器看的，所以不需要把*

*//二进制数据转换为字符串数据*

}

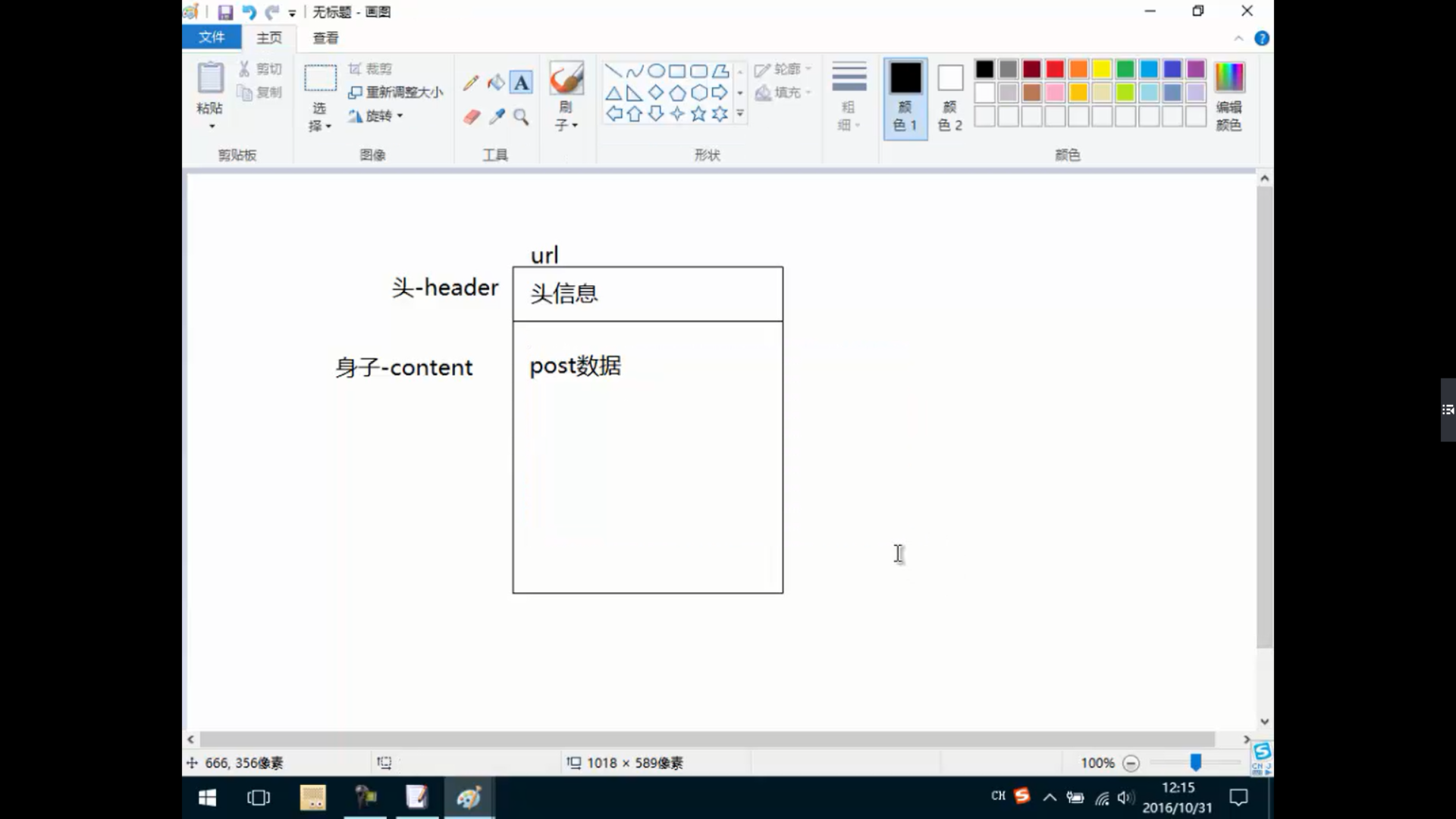
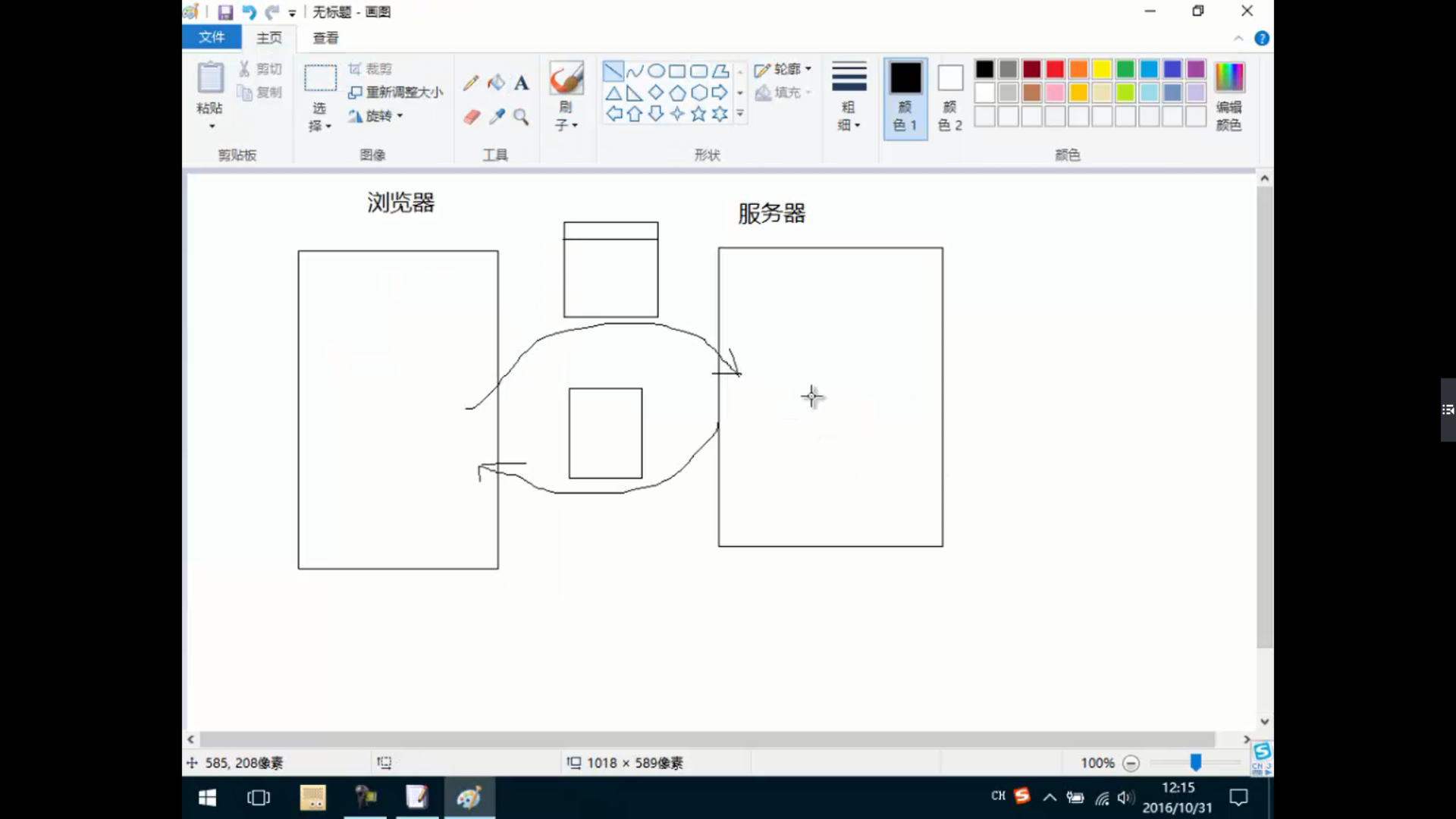
res.**end**();

});

});

server.**listen**(8080);

## 服务器接受前台数据请求

1. 数据请求：
   1. 前台——form，ajax，jsonp
   2. 后台——都一样
   3. 前台<—HTTP请求—>后台
2. 请求方式：
   1. POST 数据不在url里
      1. https://www.taobao.com/
   2. GET 数据在url里
      1. https://www.baidu.com/s?wd=node.&rsv\_spt=1&rsv\_iqid=0x9e2dad4a0001eed9&issp=1&f=8&rsv\_bp=0&rsv\_idx=2&ie=utf-8&tn=baiduhome\_pg&rsv\_enter=1&rsv\_sug3=4&rsv\_sug1=4&rsv\_sug7=101&rsv\_sug2=0&inputT=2280&rsv\_sug4=2751
   3. 
   4. 
3. 服务器读取GET数据方式
   1. html的form表单

<form *action*="http://localhost:8080/resource"，*method*="GET">

用户：<input *type*="text" *name*="user">

密码：<input *type*="password" *name*="password">

<button>提交</button>

</form>

* 1. querystring（parse方法）

const querystring = **require**('querystring');

let json = querystring.**parse**('user=jiang&password=123456&age=20');

**console**.**log**(json);

**//**{ user: 'jiang', password: '123456', age: '20' }

const http = **require**('http');

const querystring = **require**('querystring');

let server = http.**createServer**(function (req, res) {

let GET = {},

url;

if (req.url.**indexOf**('?') != -1) { *//{} '/favicon.ico' 对favicon的处理*

//res.url=>’/resource?user=jiang&password=12345’

let arr = req.url.**split**('?');

url = arr[0];

GET = querystring.**parse**(arr[1]);

} else {

url = req.url;

}

**console**.**log**(GET, url); *//{ user: 'jiang', password: '12345' } '/resource'*

*//req获取前台请求数据*

res.**write**('aaa');

res.**end**();

});

server.**listen**(8080);

* 1. url（parse方法）

const urlLib = **require**('url');

let obj = urlLib.**parse**('http://www.baidu.com:8080/index?user=jiang&password=123456', true); //这个true表示需要解析obj对象的query属性成json对象形式

//query: { user: 'jiang', password: '123456' }

**console**.**log**(obj);

*/\*\**

*\* Url {*

*protocol: 'http:',*

*slashes: true,*

*auth: null,*

*host: 'www.baidu.com:8080',*

*port: '8080',*

*hostname: 'www.baidu.com',*

*hash: null,*

*search: '?user=jiang&password=123456', //字符串形式的请求数据*

*query: 'user=jiang&password=123456',*

*pathname: '/index',*

*path: '/index?user=jiang&password=123456',*

*href: 'http://www.baidu.com:8080/index?user=jiang&password=123456' }*

*\*/*

const http = **require**('http');

const querystring = **require**('querystring');

const urlLib = **require**('url');

let server = http.**createServer**(function (req, res) {

let obj = urlLib.**parse**(req.url, true);

**console**.**log**(req.url); *//=> /resource?user=jiang&password=123456*

let url = obj.pathname,

GET = obj.query;

**console**.**log**(url, GET); *//=> /resource { user: 'jiang', password: '123456' }*

*//req获取前台请求数据*

res.**write**('aaa');

res.**end**();

});

server.**listen**(8080);

* 1. 总结（GET数据解析）
     1. querystring：xxx=xxx&yyy=yyy
     2. url：xxx?yyy=yyy&zzz=zzz

1. 服务器读取POST数据方式
   1. POST数据比GET数据大得多

const http = **require**('http');

const querysting = **require**('querystring');

let server = http.**createServer**(function (req, res) {

*//收到分段数据后需要重新组装*

let str = ''; *//接收数据, 但是无法接受图片,音频等二进制数据*

*//POST数据很大，需要分段发送，data表示有一段数据到达(发生多次)*

req.**on**('data', function (data) {//下面的例子是表单数据通过POST方法上传，所以收到的是表单数据

str += data;

});

*//end表示数据全部到达完毕(发生一次)*

req.**on**('end', function () {

*//console.log(str);//接收到的数据为'user=jiang&password=12356'*

let POST = querysting.**parse**(str);

**console**.**log**(POST); *//{ user: 'jiang', password: '123456' }*

});

res.**write**('收到');

res.**end**();

})

server.**listen**(8080);