网站后台的功能就是根据路径返回字符串

小知识:favicon.ico就是浏览器标签最左边的那个小图标

http://localhost:8888/1
（用git呼出node监听）

```
$node documentname.js
```

CTRL+C 退出

1.

```
var http = require('http');
var fs = require('fs');
var port = process.env.PORT || 8888;
var server = http.createServer(function(request, response){
  response.end('this is my response');

});
server.listen(port);
console.log('success!');
```

2.

```
var http = require('http');
var fs = require('fs');
var port = process.env.PORT || 8888;
var server = http.createServer(function(request, response){
  if(request.url==='/1'){
  response.end('<title>1</title><script src="3.js"></script>你请求1干什么？我给你一个随机数'+Math.random());
  }else if (request.url==='/2'){
  response.end('you are sb!');
  }else if (request.url==='/3.js'){
  response.setHeader('Content-Type','application/javascript');
  response.end('var a=1;alert(a);');
  }
});
server.listen(port);
console.log('success!');
```

加CSS

```
var http = require('http');
var fs = require('fs');
var port = process.env.PORT || 8888;

var server = http.createServer(function(request, response) {

    switch (request.url) {
        case '/index.html':
            response.end('<title>首页</title>' + '<link rel=stylesheet href=/style.css>' + '<h5>hi!</h5>');
            break
        case '/style.css':
            response.end('h5{color:red;}');
            break
    }
});

server.listen(port);
console.log('success!');
```

加JS

```
var http = require('http');
var fs = require('fs');
var port = process.env.PORT || 8888;

var server = http.createServer(function(request, response) {

    switch (request.url) {
        case '/index.html':
            response.end('<title>首页</title><link rel=stylesheet href=/style.css><h5>hi!</h5><button id="button">点我</button><script src="/main.js"></script>');
            break
        case '/style.css':
            response.end('h5{color:red;}');
            break
        case '/main.js':
            response.end('document.querySelector("#button").onclick=function(){alert("点击成功")}');
            break
    }
});

server.listen(port);
console.log('success!');
```

抽象化

index.js:

```
var http = require('http');
var fs = require('fs');
var port = process.env.PORT || 8888;

var server = http.createServer(function(request, response) {

    try {
        var string = fs.readFileSync('.' + request.url);
        response.end(string);
    } catch (ex) {
        response.end('can not find anything');
    }
    // switch (request.url) {
    //     case '/index.html':
    //         var htmlString = fs.readFileSync('./index.html');
    //         response.end(htmlString);
    //         break
    //     case '/style.css':
    //         var cssString = fs.readFileSync('./style.css');
    //         response.end(cssString);
    //         break
    //     case '/main.js':
    //         var jsString = fs.readFileSync('./main.js');
    //         response.end(jsString);
    //         break
    // }
});

server.listen(port);
console.log('success!');
```

main.js

```
document.querySelector("#button").onclick = function() {
    alert("点击成功")
}
```

style.css

```
h5 {
    color: red;
}
```

index.html

```
<title>首页</title>
<link rel=stylesheet href=/style.css>
<h5>hi!</h5><button id="button">点我</button>
<script src="/main.js"></script>
```

抽象完整版

```
var http = require('http');
var fs = require('fs');
var port = process.env.PORT || 8888;

var server = http.createServer(function(request, response) {

    switch (request.url) {
        case '/index.html': //这里及下一行的后缀都不重要
            var htmlString = fs.readFileSync('./index.html');
            response.setHeader('Content-Type', 'text/html'); //这个才重要，而不是后缀
            response.end(htmlString);
            break
        case '/style.css': //后缀不重要
            var cssString = fs.readFileSync('./style.css');
            response.setHeader('Content-Type', 'text/css');
            response.end(cssString);
            break
        case '/main.js': //后缀不重要
            var jsString = fs.readFileSync('./main.js');
            response.setHeader('Content-Type', 'application/javascript');
            response.end(jsString);
            break
        case '/data.json':
            response.setHeader('Content-Type', 'application/json');
            response.end('{"name":"frank","age":18}')
        default:
            response.end('404');
    }
});

server.listen(port);
console.log('success!');
```

根据路径返回字符串（json）

index.html:

```
<title>首页</title>
```

```
<link rel=stylesheet href=/style.css>
<h5>hi!</h5><button id="button">点我</button>
<script src="/main.js"></script>

<script src="//code.jquery.com/jquery-2.1.1.min.js"></script>
<script>
    // $.get('/data.json').then(function(response) {
    //     console.log(response);
    // });
    function reqListener() {
        var object = JSON.parse(this.responseText);
        console.log(typeof this.responseText);
        console.log(this.responseText);
        console.log(typeof object);
        console.log(object);
    }

    var oReq = new XMLHttpRequest();
    oReq.addEventListener("load", reqListener);
    oReq.open("GET", "/data.json");
    oReq.send();
</script>
```

根据路径返回字符串（JQuery处理json）

index.html:

```
<title>首页</title>
<link rel=stylesheet href=/style.css>
<h5>hi!</h5><button id="button">点我</button>
<script src="/main.js"></script>

<script src="//code.jquery.com/jquery-2.1.1.min.js"></script>
<script>
    $.get('/data.json').then(function(response) {
        console.log('response的内容是');
        console.log(response);
        console.log('response的类型是');
        console.log(typeof response);
    });
</script>
```