在express中使用cookie

# 一、express设置cookie

## 1.在nodejs-code文件夹下面新建一个session-control文件夹，在里面新建一个01-cookies.js文件

|  |
| --- |
|  |

## 2.使用前面我们定义的代码片段etp输入基本结构，因为我们在nodejs-code里面安装了express，所以这里不需要安装express

|  |
| --- |
| let express = require('express')  let port = 8000  let app = express()  //app.use(express.static(\_\_dirname+'/public'))//静态服务器，根据需要配置  app.get('/',(req,res)=>{       res.end('<h1>welcome</h1>')  })  app.listen(port,()=>{       console.log('server is ready :http://localhost:'+port+'/');  }) |

### 测试一下，可以正常运行

|  |
| --- |
|  |
|  |

## 3.把路由规则修改一下，并且添加代码

|  |
| --- |
| let express = require('express')  let port = 8000  let app = express()  //app.use(express.static(\_\_dirname+'/public'))//静态服务器，根据需要配置  app.get('/set-cookie',(req,res)=>{       res.cookie("name","admin")       res.end('<h1>welcome</h1>')  })  app.listen(port,()=>{       console.log('server is ready :http://localhost:'+port+'/');  }) |

## 4.使用nodemon确定服务器，然后输入/set-cookie，在调试工具中查看，效果如下

|  |
| --- |
|  |

### 可以看到响应头里面又一个Set-Cookie 内容是name=admin，这就是服务器返回的cookie。此时请求头是没有这个cookie的，然后我们刷新页面，发现请求头以及携带了这个cookie

|  |
| --- |
|  |

### 上面设置的cookie会在浏览器关闭的时候销毁，如果需要它能保存一段时间，需要设置第三个参数，如下面的代码把cookie的过期时间设置为1分钟

|  |
| --- |
| let express = require('express')  let port = 8000  let app = express()  //app.use(express.static(\_\_dirname+'/public'))//静态服务器，根据需要配置  app.get('/set-cookie',(req,res)=>{      //  res.cookie("name","admin") //这样子设置的cookie会在浏览器关闭的时候销毁       res.cookie("name","supervisor",{maxAge:60\*1000})//maxAge的时间的单位是毫秒60\*1000就是60秒       res.end('<h1>welcome</h1>')  })  app.listen(port,()=>{       console.log('server is ready :http://localhost:'+port+'/');  }) |

# 二、express删除cookie

## 1.还是上面的案例，我们添加一条删除cookie的路由，为了方便看到结果，我们在设置一个theme cookie，值是blue代码如下

|  |
| --- |
| let express = require('express')  let port = 8000  let app = express()  //app.use(express.static(\_\_dirname+'/public'))//静态服务器，根据需要配置  app.get('/set-cookie',(req,res)=>{      //  res.cookie("name","admin") //这样子设置的cookie会在浏览器关闭的时候销毁       res.cookie("name","supervisor",{maxAge:60\*1000})//maxAge的时间的单位是毫秒60\*1000就是60秒       res.cookie("theme","blue",{maxAge:60\*1000})//maxAge的时间的单位是毫秒60\*1000就是60秒       res.end('<h1>welcome</h1>')  })  app.get('/remove-cookie',(req,res)=>{       res.clearCookie("name")  })  app.listen(port,()=>{       console.log('server is ready :http://localhost:'+port+'/');  }) |

## 测试一下，我们先访问set-cookie，是有值的

|  |
| --- |
|  |

### 然后我们访问remove-cookie，name这个cookie就被删除了

|  |
| --- |
|  |

### 什么时候需要删除cookie？就是用户推出登录的时候

# 三、express读取cookie数据

## 1.需要安装一个插件cookie-parser：npm i cookie-parser

|  |
| --- |
|  |

## 2.使用：这个包其实是一个中间件。可以全局设置，然后我们添加一个获取cookie的路由规则

|  |
| --- |
| let express = require('express')  let port = 8000  let app = express()  //导入cookie-parser  let cookieParser = require('cookie-parser')  //全局注册cookie-parser中间件  app.use(cookieParser())  //app.use(express.static(\_\_dirname+'/public'))//静态服务器，根据需要配置  app.get('/set-cookie',(req,res)=>{      //  res.cookie("name","admin") //这样子设置的cookie会在浏览器关闭的时候销毁       res.cookie("name","supervisor",{maxAge:60\*1000})//maxAge的时间的单位是毫秒60\*1000就是60秒       res.cookie("theme","blue",{maxAge:60\*1000})//maxAge的时间的单位是毫秒60\*1000就是60秒       res.end('<h1>welcome</h1>')  })  app.get('/remove-cookie',(req,res)=>{       res.clearCookie('name')       res.send('ok')  })  //获取cookie的路由规则  app.get('/get-cookie',(req,res)=>{      //获取cookie并且输出      console.log(req.cookies);      res.send("ok")  })  app.listen(port,()=>{       console.log('server is ready :http://localhost:'+port+'/');  }) |

### 我们先调用set-cookie路由，然后访问get-cookie路由，就会在控制台输出cookie效果如下

|  |
| --- |
|  |

### 还可以把获取到的cookie值返回给浏览器

|  |
| --- |
|  |