# pathinfo/pathname风格参数的获取

|  |
| --- |
|  |

## 其实就是动态路由

# 项目实践

# 1.在express\_basic文件夹下面新建一个express\_pathinfo文件夹，然后进入这个文件，使用yarn init -y初始化项目，完成后如图

|  |
| --- |
|  |

# 2.安装express，yarn add express，完成后如图

|  |
| --- |
|  |

# 3.新建一个项目服务主文件01-pathinfo.js，编写脚本框架代码（代码片段：etp），如下

|  |
| --- |
| let express = require('express')  let port = 3000  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+'/');  }) |

# 4.安装art-template并且配置模板引擎

## 1）安装art-template，使用命令yarn add art-template express-art-template，安装后如图

|  |
| --- |
|  |

## 2）在01-pathinfo.js中添加下面的代码

|  |
| --- |
|  |

# 5.在views文件夹里面新建一个list.html,内容如下

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>List demo</title>      <style>          ul {              list-style: none;          }      </style>  </head>  <body>      <h2>新闻列表</h2>      <hr>      <ul>          <li><a href="/detail/1">第1篇新闻标题</a></li>          <li><a href="/detail/2">第2篇新闻标题</a></li>          <li><a href="/detail/3">第3篇新闻标题</a></li>          <li><a href="/detail/4">第4篇新闻标题</a></li>      </ul>  </body>  </html> |

# 6.在服务器文件中添加对这个动态路径如：/detail/:id的响应,注意默认在express中输出中文是会有乱码的，需要在一个全局中间件里面设置一下

|  |
| --- |
| let express = require('express')  let path = require('path')  let port = 3000  let app = express()  //处理中文乱码  app.use(function (req, res, next) {      res.setHeader('Content-Type', 'text/html;charset=utf-8');      next();  });  //配置模板引擎,4步，一个engine和3个set  app.engine('html',require('express-art-template'))  app.set('view options',{      debug:process.env.NODE\_ENV!=='production'  })  app.set('views',path.join(\_\_dirname,'views'))  app.set('view engine','html')  //app.use(express.static(\_\_dirname+'/public'))//静态服务器，根据需要配置  app.get('/',(req,res)=>{       res.end('<a href="/list">new list</a>')  })  app.get('/list',(req,res)=>{ //渲染列表页面       res.render('list')  })  //响应动态路由  app.get('/detail/:id',(req,res)=>{      let {id,type} = req.params      res.end(`<h1>第${id}条新闻标题</h1>`)  })  app.listen(port,()=>{       console.log('server is ready :http://localhost:'+port+'/');  }) |

# 7.那么，如果我们需要不同类型的2级动态路由怎么办？如我们修改一下list.html

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>List demo</title>      <style>          ul {              list-style: none;          }      </style>  </head>  <body>      <h2>新闻列表</h2>      <hr>      <ul>          <li><a href="/detail/1/music">第1篇新闻标题</a></li>          <li><a href="/detail/2/movie">第2篇新闻标题</a></li>          <li><a href="/detail/3/sport">第3篇新闻标题</a></li>          <li><a href="/detail/4/economy">第4篇新闻标题</a></li>      </ul>  </body>  </html> |

## 其实也是一样的道理，我们修改01-pathinfo.js如下

|  |
| --- |
| let express = require('express')  let path = require('path')  let port = 3000  let app = express()  //处理中文乱码  app.use(function (req, res, next) {      res.setHeader('Content-Type', 'text/html;charset=utf-8');      next();  });  //配置模板引擎,4步，一个engine和3个set  app.engine('html',require('express-art-template'))  app.set('view options',{      debug:process.env.NODE\_ENV!=='production'  })  app.set('views',path.join(\_\_dirname,'views'))  app.set('view engine','html')  //app.use(express.static(\_\_dirname+'/public'))//静态服务器，根据需要配置  app.get('/',(req,res)=>{       res.end('<a href="/list">new list</a>')  })  app.get('/list',(req,res)=>{       res.render('list') //渲染列表页面  })  //响应动态路由  app.get('/detail/:id/:type',(req,res)=>{      let {id,type} = req.params      res.end(`<h1>第${id}条新闻标题</h1> <hr> <h3>主题是：${type}</h3>`)  })  app.listen(port,()=>{       console.log('server is ready :http://localhost:'+port+'/');  }) |

## 效果：

|  |
| --- |
|  |
|  |