# 介绍

|  |
| --- |
|  |

# 1.还是使用MySQL-demo1项目，数据库使用qd22，跟老师的名称不一样

|  |
| --- |
|  |

# 2.我们上一节已经用yarn把这个项目初始化了。新建有package.json了，现在我们需要安装mysql，使用命令 yarn add mysql安装，完成后如图

|  |
| --- |
|  |

# 3.需要安装express，yarn add express

# 4.安装body-parser，使用命令yarn add body-parser

# 5.在项目文件夹里面新建一个db目录，在里面创建一个操作数据库的db.js文件，内容如下

|  |
| --- |
|  |
| let mysql = require('mysql')  //创建连接池  let pool = mysql.createPool({      host:'localhost',      user:'root',      password:'root',      database:'qd22'  })  function query(sql,callback){      pool.getConnection(function(err,connection){          connection.query(sql,function(err,rows){              callback(err,rows)              connection.release()          })      })  }  exports.query = query |

# 6.在项目根目录下面新建一个01-mysql-app.js，写入一下基本的express代码，（我在vscode里面设置了代码判断etp，使用他可以快速生成代码）

|  |
| --- |
|  |

## 测试一下，正常运行

|  |
| --- |
|  |
|  |

# 7、成功是成功了但是如果想把数据渲染到页面，我们需要安装art-template和express-art-template，使用命令yarn add art-template express-art-template

|  |
| --- |
|  |

# 8.在项目根目录下面新建一个views文件夹

|  |
| --- |
|  |

# 9.在01-mysql-app.js里面配置模板引擎并且把views设置位模板查找目录

|  |
| --- |
|  |

# 10.中views目录里面新建一个student-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>学生列表</title>  </head>  <body>      <h2>学生列表</h2>      <hr>      {{each students}}      <table>          <tr>              <td>学生姓名：</td>              <td>{{$value.name}}</td>              <td>详情：</td>              <td><a href="/student\_detail/{{$value.id}}">学生详情</a></td>              {{/each}}  </body>  </html> |

# 11.在/get\_data路由中添加渲染模板的代码

|  |
| --- |
|  |

## 测试一下，能够显示学生列表

|  |
| --- |
|  |

# 12.在01-mysql-app.js中添加动态路由/student\_detail/:id来处理学生详细信息

|  |
| --- |
|  |

# 13.在views文件夹下面新建一个detail.html,用来显示学生详情信息

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>学生详情</title>  </head>  <body>      <h2>学生信息</h2>      <hr>      <h3>          <ul style="list-style: none;">              {{each students}}              <li>姓名：{{$value.name}}</li>              <li>性别：{{$value.gender}}</li>              <li>年龄：{{$value.age}}</li>              <li>身高：{{$value.height}}</li>              {{/each}}          </ul>      </h3>  </body>  </html> |

## 效果：

|  |
| --- |
|  |

# 14.为了完善一下我们的应用程序，我们在views文件夹里面新建一个index.html文件，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>首页</title>  </head>  <body>      <h1>首页</h1>      <hr>      <h3><a href="/get\_data">查看学生列表</a></h3>  </body>  </html> |

# 15，然后我们修改 / 路由处理函数、

|  |
| --- |
|  |

## 测试一下：

|  |
| --- |
|  |
|  |
|  |

# 本节到此结束，01-mysql-app.js的完整代码如下

|  |
| --- |
| **let path = require('path')**  **let express = require('express')**  **let db = require('./db/db') //导入db模块**  **let port = 3000**  **let app = express()**  **//app.use(express.static(\_\_dirname+'/public'))//静态服务器，根据需要配置**  **//配置模板引擎和视图路径**  **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.get('/',(req,res)=>{**  **res.render('index')**  **})**  **//获取学生列表的路由**  **app.get('/get\_data',(req,res)=>{**  **//查询数据库，返回到浏览器**  **db.query('select \* from students;',function(err,rows){**  **// console.log(rows);**  **res.render('student-list',{students:rows})**  **})**  **})**  **//显示学生详情信息的路由**  **app.get('/student\_detail/:id',(req,res)=>{**  **let id = req.params.id**  **let sql = 'select \* from students where id=' + id +";"**  **//查询数据库，返回到浏览器**  **db.query(sql,function(err,rows){**  **// console.log(rows);**  **res.render('detail',{students:rows})**  **})**    **})**  **app.listen(port,()=>{**  **console.log('server is ready :http://localhost:'+port+'/');**  **})** |