# 项目结构

|  |
| --- |
|  |

# 1.只是使用Promise还是比较复杂

## 3-promise-join-file-content.js

|  |
| --- |
| let fs = require('fs')  let path = require('path')  let filePath1 = path.join(\_\_dirname,"files/client.txt")  let filePath2 = path.join(\_\_dirname,"files/secret.txt")  let filePath3 = path.join(\_\_dirname,"files/info.txt")  let p1 = new Promise((resolve,reject)=>{      fs.readFile(filePath1,"utf8",(err,data)=>{      if(err){          reject(err)      }      resolve(data)    })  })  //拼接  let content = ''  p1.then(data=>{      // console.log(data);      content += data + '\r\n'      return new Promise((resolve,reject)=>{          fs.readFile(filePath2,"utf8",(err,info)=>{          if(err){              reject(err)          }          resolve(content+info + '\r\n')        })      })  }).then(data2=>{      return  new Promise((resolve,reject)=>{          fs.readFile(filePath3,"utf8",(err,data)=>{          if(err){              reject(err)          }          resolve(data2 + data)        })      })  }).then(data3=>{      console.log("data:" ,data3);      fs.writeFile("content.txt",data3,(err)=>{          if(err){              console.log(err);              return          }          console.log("write file succeeded...");      })  }) |

# 2.其实我们可以使用异步函数async+await的方式+Promise来实现

|  |
| --- |
| let fs = require('fs')  let path = require('path')  let filePath1 = path.join(\_\_dirname,"files/client.txt")  let filePath2 = path.join(\_\_dirname,"files/secret.txt")  let filePath3 = path.join(\_\_dirname,"files/info.txt")  async function joinData(filePath1,filePath2,filePath3){       let str = ''       let data1 = await new Promise((resolve,reject)=>{          fs.readFile(filePath1,"utf8",(err,data)=>{          if(err){              reject(err)          }          resolve(data)        })      }) //这里data1就是文件1的内容       let data2 = await new Promise((resolve,reject)=>{          fs.readFile(filePath2,"utf8",(err,data)=>{          if(err){              reject(err)          }          resolve(data)        })  }) //这里data2就是文件2的内容       let data3 = await new Promise((resolve,reject)=>{          fs.readFile(filePath3,"utf8",(err,data)=>{          if(err){              reject(err)          }          resolve(data)        })      }) //这里data3就是文件3的内容      // console.log(data1);      str = data1 + '\r\n' +data2 + '\r\n' + data3      console.log('joined data:\n',str);      fs.writeFile('./files/result.txt',str,err=>{          if(err){              console.log(err);              return;          }          console.log('saved result in file with success...');      })  }  joinData(filePath1,filePath2,filePath3) |

## 注意：可以用await+一个变量来接收Promise对象得到的是resolve函数的data值

|  |
| --- |
|  |
|  |

# 3.第一种写法的改进版本，也叫做函数版本

## 5-3-promise-join-file-content-better.js

|  |
| --- |
| let fs = require('fs')  let path = require('path')  let filePath1 = path.join(\_\_dirname,"files/client.txt")  let filePath2 = path.join(\_\_dirname,"files/secret.txt")  let filePath3 = path.join(\_\_dirname,"files/info.txt")  function getPromise(filepath){      return new Promise((resolve,reject)=>{          fs.readFile(filepath,"utf8",(err,data)=>{          if(err){              reject(err)          }          resolve(data)        })      })  }  let p1 = getPromise(filePath1)  let p2 = getPromise(filePath2)  let p3 = getPromise(filePath3)  //拼接  let content = ''  p1.then(data=>{      content += data + '\r\n'      return p2  }).then(data=>{      content += data + '\r\n'      return p3  }).then(data=>{      content += data      console.log('final data:\n',content);  }) |

# 4.第二种写法的改进版本

## 6-4-promise-join-file-content2-better.js

|  |
| --- |
| let fs = require('fs')  let path = require('path')  let filePath1 = path.join(\_\_dirname,"files/client.txt")  let filePath2 = path.join(\_\_dirname,"files/secret.txt")  let filePath3 = path.join(\_\_dirname,"files/info.txt")  function getPromise(filepath){    return new Promise((resolve,reject)=>{        fs.readFile(filepath,"utf8",(err,data)=>{        if(err){            reject(err)        }        resolve(data)      })    })  }  async function joinData(filePath1,filePath2,filePath3){       let str = ''       let data1 = await getPromise(filePath1)       let data2 = await getPromise(filePath2)       let data3 = await getPromise(filePath3)      // console.log(data1);      str = data1 + '\r\n' +data2 + '\r\n' + data3      console.log('joined data:\n',str);      fs.writeFile('./files/result2.txt',str,err=>{          if(err){              console.log(err);              return;          }          console.log('saved result in file with success...');      })  }  joinData(filePath1,filePath2,filePath3) |

## 效果

|  |
| --- |
|  |
|  |

# 5.第一种写法的util版本，使用util.promisify函数，这个函数可以把一个异步方法变为一个能返回Promise对象的方法，插入的函数必须是错误优先的异步函数，如fs模块里面的函数

|  |
| --- |
| let fs = require('fs')  let path = require('path')  let util = require('util')  let filePath1 = path.join(\_\_dirname,"files/client.txt")  let filePath2 = path.join(\_\_dirname,"files/secret.txt")  let filePath3 = path.join(\_\_dirname,"files/info.txt")  const readFile = util.promisify(fs.readFile) //util.promisify可以把一个异步函数变为一个返回Promise的函数  let p1 = readFile(filePath1)  let p2 = readFile(filePath2)  let p3 = readFile(filePath3)  //拼接  let content = ''  p1.then(data=>{      content += data + '\r\n'      return p2  }).then(data=>{      content += data + '\r\n'      return p3  }).then(data=>{      content += data      console.log('final data:\n',content);  }) |

## 效果：

|  |
| --- |
|  |

# 6.第二种写法的util版本

|  |
| --- |
| let fs = require('fs')  let path = require('path')  let util = require('util')  let filePath1 = path.join(\_\_dirname,"files/client.txt")  let filePath2 = path.join(\_\_dirname,"files/secret.txt")  let filePath3 = path.join(\_\_dirname,"files/info.txt")  const readFile = util.promisify(fs.readFile) //util.promisify可以把一个异步函数变为一个返回Promise的函数  async function joinData(filePath1,filePath2,filePath3){       let str = ''       let data1 = await readFile(filePath1)       let data2 = await readFile(filePath2)       let data3 = await readFile(filePath3)      // console.log(data1);      str = data1 + '\r\n' +data2 + '\r\n' + data3      console.log('joined data:\n',str);      fs.writeFile('./files/result3.txt',str,err=>{          if(err){              console.log(err);              return;          }          console.log('saved result in file with success...');      })  }  joinData(filePath1,filePath2,filePath3) |

## 效果：

|  |
| --- |
|  |
|  |