# 实验4：异步IO

实验背景

完成实验中的实例，理解Node.js中的导步与同步。如图所示：

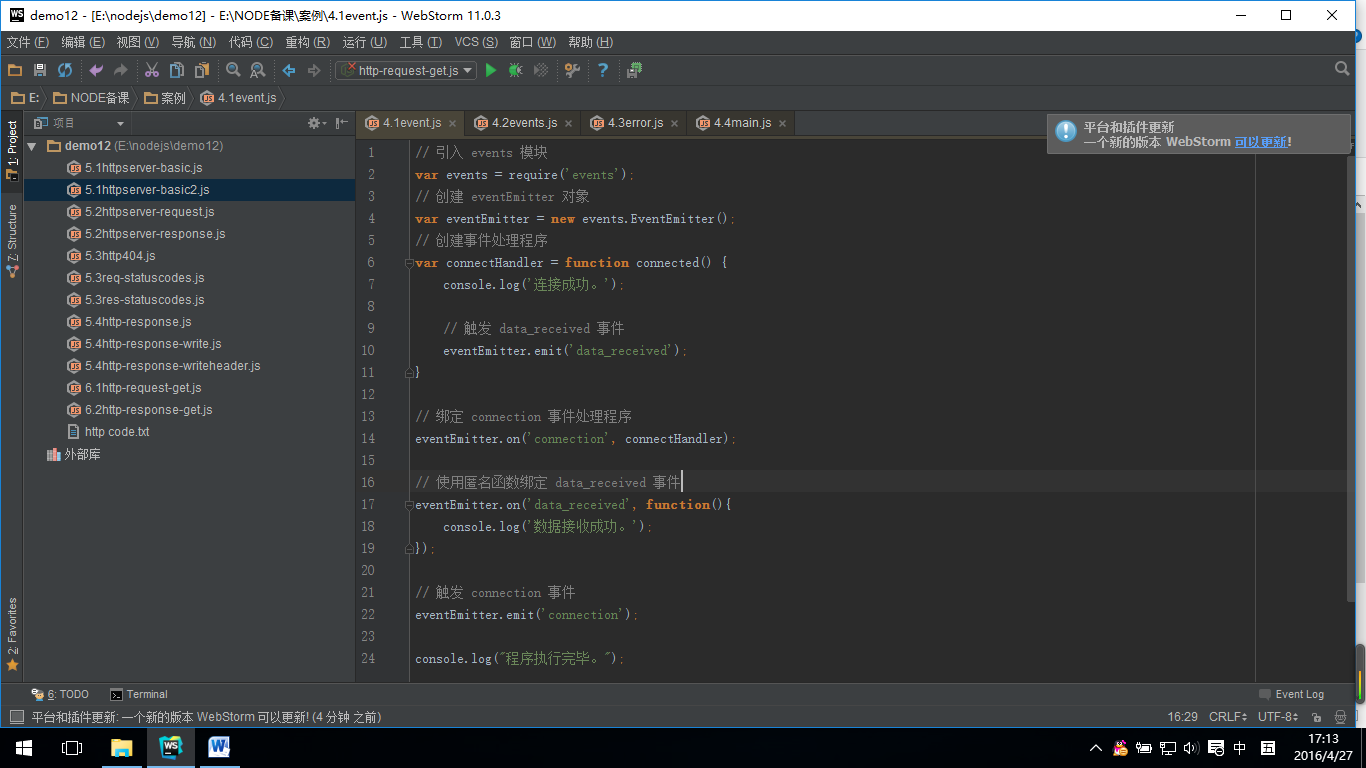


图1-1

实验目的

1．了解异步I/O

2．理解异步操作

3．掌握文件的读取

实验分析

本次实验要完成的有三个方面的知识。第一：完成案例《4.1 event》第二：完成案例《4.2 events》，第三：完成案例《4.3 error》。

实验步骤

1. 相关知识点

Node.js 文件系统

文件系统模块是一个封装了标准的 POSIX 文件 I/O 操作的集合。通过 require('fs') 使用这个模块。所有的方法都有同步和异步两种模式。

异步方法最后一个参数都是回调函数，这个回调的参数取决于方法，不过第一个参数一般都是异常。

如果操作成功，那么第一个参数就是 null 或 undefined。

1、fs.readFile(filename[, options], callback)

filename：{String}

options ：{Object}，可选参数

* encoding： {String | Null} 默认 = null
* flag ：{String} 默认 = 'r'

callback：{Function}

异步读取整个文件的内容。

使用方法详见：步骤3

2、fs.readFileSync(filename[, options])

fs.readFile 的同步版本. 返回整个文件的内容.

使用方法详见：步骤2

1. 创建将要读取的文本

创建一个目录名称为FS，在目录下创建纯文本文件，并命名为：input.txt,

在文件内可随意输入一些文字，如：

*在Node应用程序，任何异步函数接受回调作为最后的参数和回调函数接受错误的第一个参数。*

1. 同步读取文件

在FS目录下创建fs.js文件,实现同步读取文件的功能

参考代码：

var fs=require('fs');

var data=fs.readFileSync('input.txt','utf-8');

console.log(data)

console.log('end.')

代码运行结果:

Contents of the file.//读取的文件内容

end.

1. 异步读取文件

在FS目录下创建fs.js文件,实现同步读取文件的功能

参考代码：

var fs=require('fs');

 fs.readFile('input.txt','utf-8',function(err,data){

if(err){

console.log(err)

}else{

console.log(data)

}

});

console.log('end.')

**代码运行结果:**

end.

Contents of the file.//读取的文件内容

扩展练习

1.参考学习资料：http://www.w3cschool.cn/nodejs/hzpd1iti.html

*备注：文件系统中将会详细讲解fs模块，在本实验中不需要掌握fs的使用，重点理解导步和同步既可。*