

实验三 Node.js 实现 HTTP 服务

一、实验目的：

1. 了解 HTTP 请求响应的流程
2. 了解 HTTP 的请求报文和响应报文
3. 掌握如何使用 HTTP 模块构建 Web 服务器
4. 掌握使用 HTTP 服务处理请求。

二、实验内容

1. 第一个 web 服务器程序

```
var http=require("http")
var server=http.createServer()
server.on("request",function(req,res){
    console.log("本次 url 地址为"+req.url)
    console.log("收到了请求")
    res.write("hello")
    res.write("node.js")
    res.end("this is end ")
})
server.listen(2000,function(){
    console.log("服务器启动成功，请在浏览器中输入：http://127.0.0.1:2000 执行")
})
```

在命令行窗口执行启动服务器 : node XXXXX.js

然后在浏览器地址栏分别输入以下地址后回车，在浏览器窗口和命令行窗口查看结果：

<http://127.0.0.1:2000>

<http://127.0.0.1:2000/login>

<http://127.0.0.1:2000/register>

(1) 代码中的 url 是什么

(2) Server.on () 方法中的回调函数什么时候执行

2. http 服务器根据不同的 URL 做出不同的响应

```
var http=require("http")
var server=http.createServer()
server.on("request",function(req,res){
    res.setHeader("Content-Type", "text/plain;charset=utf-8")
    if(req.url==="/"){
        res.write("这是首页")
        res.end("this is index page")
    }
    else if(req.url==="/login"){
        res.write("这是登录页")
        res.end("this is login page")
    }
    else if(req.url==="/register"){
        res.write("这是注册页")
        res.end("this is register page")
    }else{
        res.write("没有这页")
        res.end("no this page")
    }
})
server.listen(2000,function(){
    console.log("服务器启动成功，请在浏览器中输入：http://127.0.0.1:2000 执行")
})
```

在命令行窗口执行启动服务器 : node XXXXX.js

然后在浏览器地址栏分别输入以下地址后回车，在浏览器窗口和命令行窗口查看结果：

<http://127.0.0.1:2000>

<http://127.0.0.1:2000/login>

<http://127.0.0.1:2000/register>

(1) 把 `res.setHeader("Content-Type", "text/plain;charset=utf-8")` 注释掉，再重新执行，有什么现象

(2) 服务器如何分辨不同的 URL 并做出不同的响应

(3) 改写代码，增加响应如下 URL

<http://127.0.0.1:2000/about>

<http://127.0.0.1:2000/detail>

3. 服务器向客户端浏览器响应一个 HTML 页面

(1) 先创建一个 HTML 页面命名为 `index.html`，在页面里添加一些标记

(2) 创建 js 文件，读出上述页面，并响应到浏览器

```
var http = require("http")
var fs = require("fs")
var server = http.createServer()
server.on("request", function (req, res) {
  if (req.url === "/index") {
    fs.readFile("./index.html", "utf-8", function (err, data) {
      res.setHeader("Content-Type", "text/html;charset=utf-8")
      res.end(data)
    })
  }
  else{
    res.write("没有这页")
    res.end("no this page")
  }
})
server.listen(3000, function () {
  console.log("服务器已经启动")
})
```

执行后在浏览器地址栏输入如下 URL 查看结果：

<http://127.0.0.1:2000/index>

(4) 再分别创建 `login.html` 、 `register.html`.改写上述程序，使得在浏览器输入

<http://127.0.0.1:2000/login>

<http://127.0.0.1:2000/register>

能分别响应对应的页面

4. 再响应 2 个请求，

<http://127.0.0.1:2000/kangyiimage>

<http://127.0.0.1:2000/mp4>

能在浏览器中分别显示一幅图片，播放一个视频

三、 实验收获或心得体会