Atitit 前端测试最简化内嵌web服务器 php 与node.js

目录

[1.1. Java 嵌入式tomcat 1](#_Toc31117)

[1.2. php内置Web Server 1](#_Toc8670)

[1.3. Node的 2](#_Toc9568)

## Java 嵌入式tomcat

Atitit 嵌入式 tomcat的使用

Node的比较麻烦些。。Php更加简单

## php内置Web Server

D:\wampv2\bin\php\php5.6.15\php.exe -S localhost:8000 -t D:\wampv2\www

PHP 5.4.0起， CLI SAPI 提供了一个内置的Web服务器。

这个内置的Web服务器主要用于本地开发使用，不可用于线上产品环境。

URI请求会被发送到PHP所在的的工作目录（Working Directory）进行处理，除非你使用了-t参数来自定义不同的目录。

如果请求未指定执行哪个PHP文件，则默认执行目录内的index.php 或者 index.html。如果这两个文件都不存在，服务器会返回404错误。

当你在命令行启动这个Web Server时，如果指定了一个PHP文件，则这个文件会作为一个“路由”脚本，意味着每次请求都会先执行这个脚本。如果这个脚本返回 **FALSE** ，那么直接返回请求的文件（例如请求静态文件不作任何处理）。否则会把输出返回到浏览器。

<?php

// router.php D:\wampv2\bin\php\php5.6.15\php.exe -S localhost:8000 -t D:\0db\webserver D:\0db\webserver\router.php

// http://localhost:8000/device/get

//echo $\_SERVER["REQUEST\_URI"];

if ($\_SERVER["REQUEST\_URI"]=="/api2")

{

header("Access-Control-Allow-Origin: \*");

echo ( file\_get\_contents('d:\0db\list.json'));

return true;

}

if (strstr($\_SERVER["REQUEST\_URI"],"/device/get") )

{

header("Access-Control-Allow-Origin: \*");

echo ( file\_get\_contents('d:\0db\iot\_part.json'));

return true;

}

// return false; // 直接返回请求的文件

?>

## Node的

var http = require('http');

var fs = require('fs');

var url = require('url');

//D:\0workspace\nodejs\node.exe D:\0db\nodewebserver.js

// 创建服务器

port=1314

http.createServer( function (request, response) {

console.log(request)

if(request.url==="/api2")

{

response.writeHead(200, {'Content-Type': 'text/html',"Access-Control-Allow-Origin":"\*"});

// 响应文件内容

response.write(fs.readFileSync('d:\\0db\\list.json','utf8'));

response.end();

}

}).listen(port);

console.log('Server running at http://localhost:/'+port);

# 问题

Php的uri会吧querystr也弄到。。所以只好判断包含uri就可

if (strstr($\_SERVER["REQUEST\_URI"],"/device/get") )