# 練習01-1PC1

> 學號：112111122 (學號和姓名都要寫)
> 姓名：周廉恆

1. 第一部分：建立MIME類型模組

Ans:

```
const contentTypes = {
  '.html': 'text/html; charset=utf-8',          // HTML 網頁文件
  '.ejs': 'text/html; charset=utf-8',           // EJS 模板（渲染後輸出為 HTML）
  '.js': 'text/javascript; charset=utf-8',      // JavaScript 腳本文件
  '.css': 'text/css; charset=utf-8',            // CSS 樣式表文件
  '.json': 'application/json',                  // JSON 資料格式
  '.png': 'image/png',                          // PNG 圖片格式
  '.jpg': 'image/jpg',                          // JPG/JPEG 圖片格式
  '.gif': 'image/gif',                          // GIF 動畫圖片
  '.svg': 'image/svg+xml',                      // SVG 向量圖形
  '.ico': 'image/x-icon'                        // 網站 favicon 圖示
};
function getContentType(extname) {
    const contentType = contentTypes[extname] || 'text/plain';// 這個||「如果左邊沒東西，就用右邊的」
    return contentType;
}
export default getContentType;
```

- 這是`utils/mimeTypes.js`內容，主要工作在於給程序有一個能資源篩選器，判斷請求的資源「附檔名」是否在字典裏面，沒有的話就判斷為`'text/plain'`。

---

2. 第二部分：建立模板渲染模組

Ans:

```
import fs from 'fs';
import ejs from 'ejs';
function renderTemplate(res, filePath, data = {}) {
    fs.readFile(('.' + filePath), 'utf8', function(err, template)  {
          //'.' ：代表 「當前目錄」
          //這行程式碼 '.' + filePath 的動作非常單純，就是 「字串拼接」
          if (err) {
            res.writeHead(500, { 'Content-Type': 'text/html; charset=utf-8' });
            res.end('錯誤：無法讀取模板文件 - ' + err.message);
            return;
          }
          const html = ejs.render(template, data);//這裡用了EJS的render方法來渲染模
```

```
板，並傳入資料
          res.writeHead(200, { 'Content-Type': 'text/html; charset=utf-8' });//這
串文字叫做 HTTP Header (回應標頭)，它是瀏覽器的「使用說明書」。
          //直接寫死 'text/html; charset=utf-8' 反而更合理，因為 EJS 模板渲染後的內容
本質上是 HTML，所以不需要受到 mimeTypes.js 的影響。
          res.end(html);
        });
}
export default renderTemplate;// 這裡改成 ES6 語法
```

- 這是utils/templateRenderer.js內容，主要工作是渲染動態資源，也就是EJS模板。

---

3. 第三部分：建立靜態文件處理模組

Ans:

```
import fs from 'fs';
import ejs from 'ejs';
import getContentType from './mimeTypes.js';
function  handleStaticFile(res,fileOtherFile,extname) {
    const staticFilePath = '.' + fileOtherFile;
    const contentType = getContentType(extname);
    fs.readFile(staticFilePath, 'utf8' ,(err, content) => {
      if (err) {// 讀取靜態資源失敗，回傳 404
        fs.readFile(('.' + '/index3.ejs'), 'utf8', (err, template) => {
          const html = ejs.render(template);
          res.writeHead(200, { 'Content-Type': 'text/html; charset=utf-8' });
          res.end(html);
        });
      } else {
        res.writeHead(200, { 'Content-Type': contentType });
        res.end(content);
      }
    });
}
export default handleStaticFile;
```

```
import path from 'path';
import handleStaticFile from './utils/staticFileHandler.js';
import renderTemplate from './utils/templateRenderer.js';
// 這個 Controller 負責接收路由決定的參數，然後執行實際的商業邏輯
function userController(res, filePath, fileOtherFile, data){
  const extname = (fileOtherFile === '') ? path.extname(filePath) :
path.extname(fileOtherFile);
  if (extname === '.ejs') {
    renderTemplate(res, filePath, data)
  } else {
    handleStaticFile(res, fileOtherFile,extname)
  }
```

```
}
export default userController;
```

**這裡我分成兩個檔案**

- 第一個程序`utils/staticFileHandler.js`是拿來判讀靜態資源。
- 第二個程序`userController.js`是用來做商業邏輯Controller的工作。

---

4. 第四部分：重構主檔案

Ans:

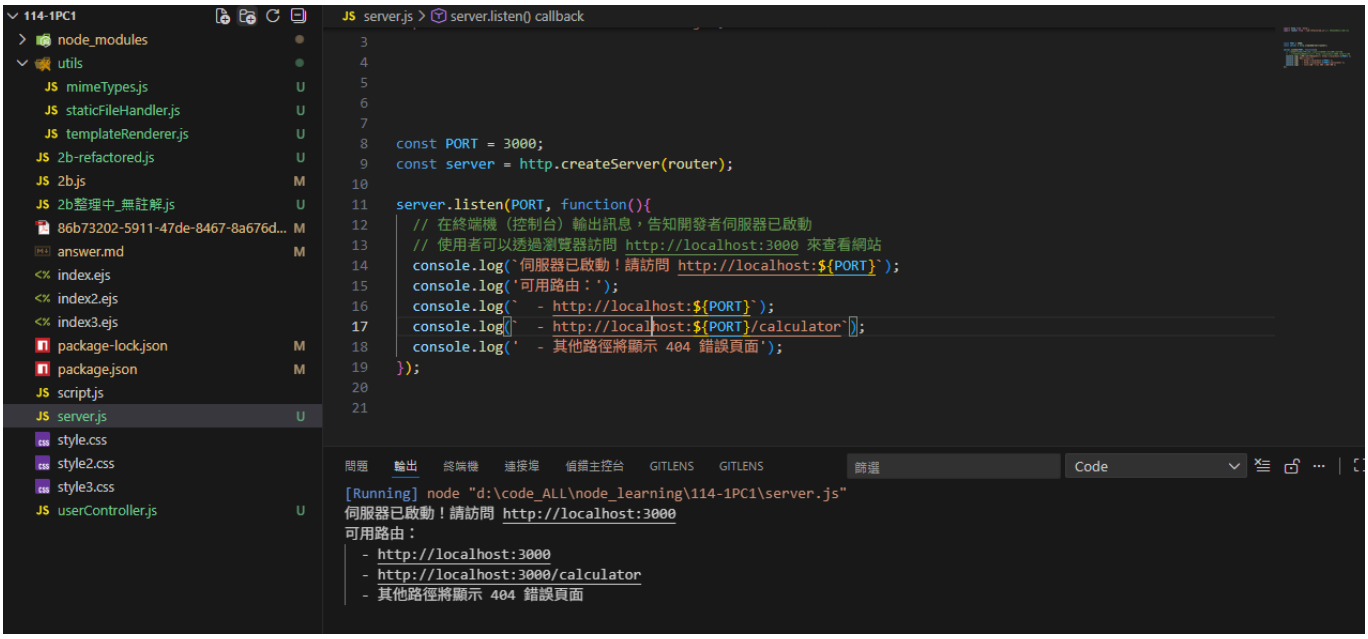
```
import userController from './userController.js';
function router(req, res) {
  let filePath = '';//創建filePath變數，儲存要渲染的 EJS 模板文件路徑
  let fileOtherFile = '';//儲存靜態資源（CSS、JS 等）的路徑
  var data={};
  switch (req.url) {
    case '/':
      //我增加了篩選請求方法的功能
      if (req.method === 'GET') {
      filePath = '/index.ejs';
      data = "您好 xxx";
    } else if (req.method === 'POST') {
      console.log('接收到 POST 請求');
    }
      break;
    case '/calculator':
      //我增加了篩選請求方法的功能
      if (req.method === 'GET') {
        filePath = '/index2.ejs';
        break;
      } else if (req.method === 'POST') {
        console.log('接收到 POST 請求');
      }
    default:// 如果都不是上面定義的
      filePath = req.url;// 直接把網址當作檔案路徑
      fileOtherFile = filePath;// 標記這可能是一個靜態檔案
      break;
  }
  userController(res, filePath, fileOtherFile, data);
}
export default router;//  這裡改成 ES6 語法
```

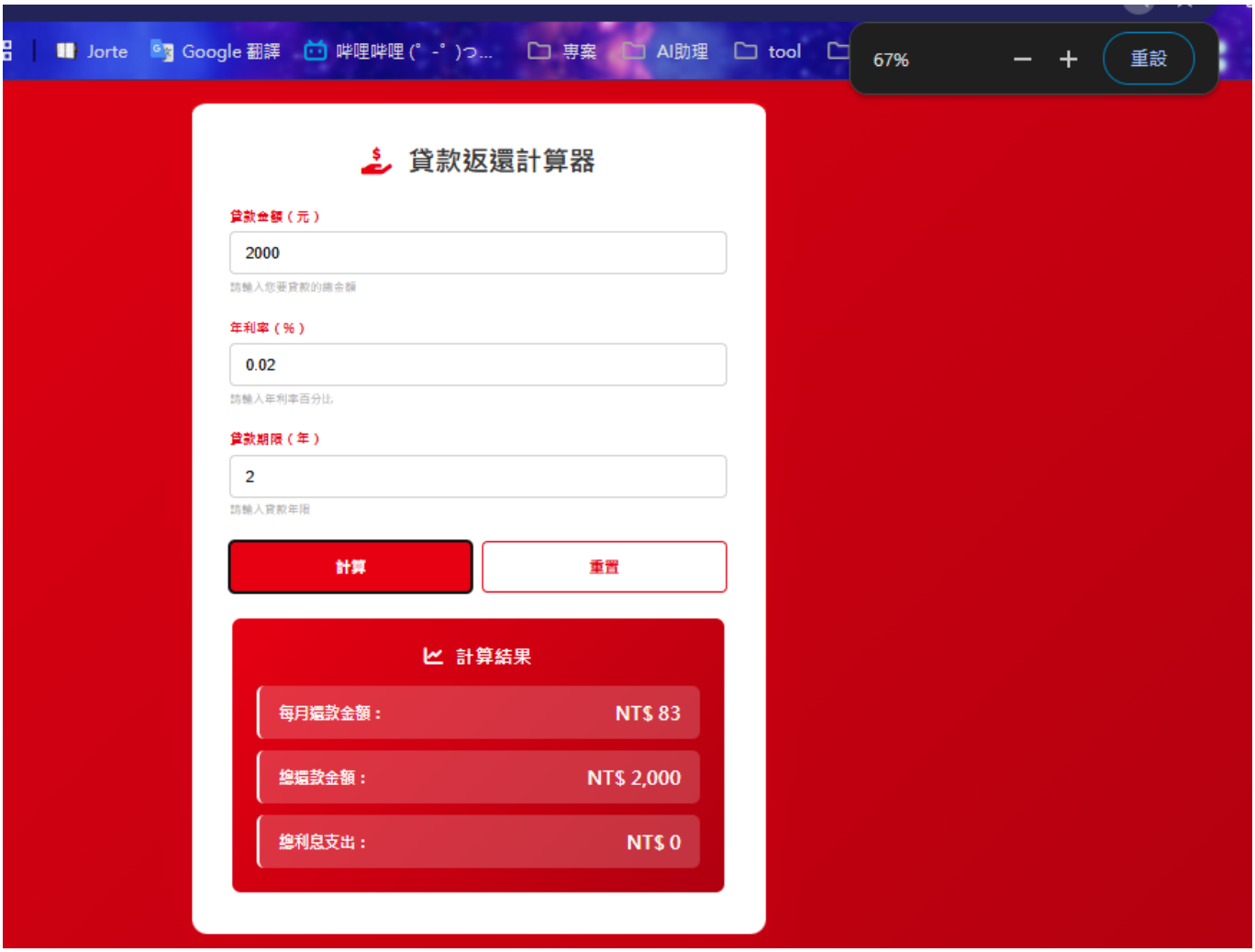
```
import http from 'http';
import router from './2b-refactored.js';// 引入手動定義的路由
const PORT = 3000;
const server = http.createServer(router);
```

```
server.listen(PORT, function(){
  // 在終端機（控制台）輸出訊息，告知開發者伺服器已啟動
  // 使用者可以透過瀏覽器訪問 http://localhost:3000 來查看網站
  console.log(`伺服器已啟動！請訪問 http://localhost:${PORT}`);
  console.log('可用路由：');
  console.log(`  - http://localhost:${PORT}`);
  console.log(`  - http://localhost:${PORT}/calculator`);
  console.log('  - 其他路徑將顯示 404 錯誤頁面');
});
```

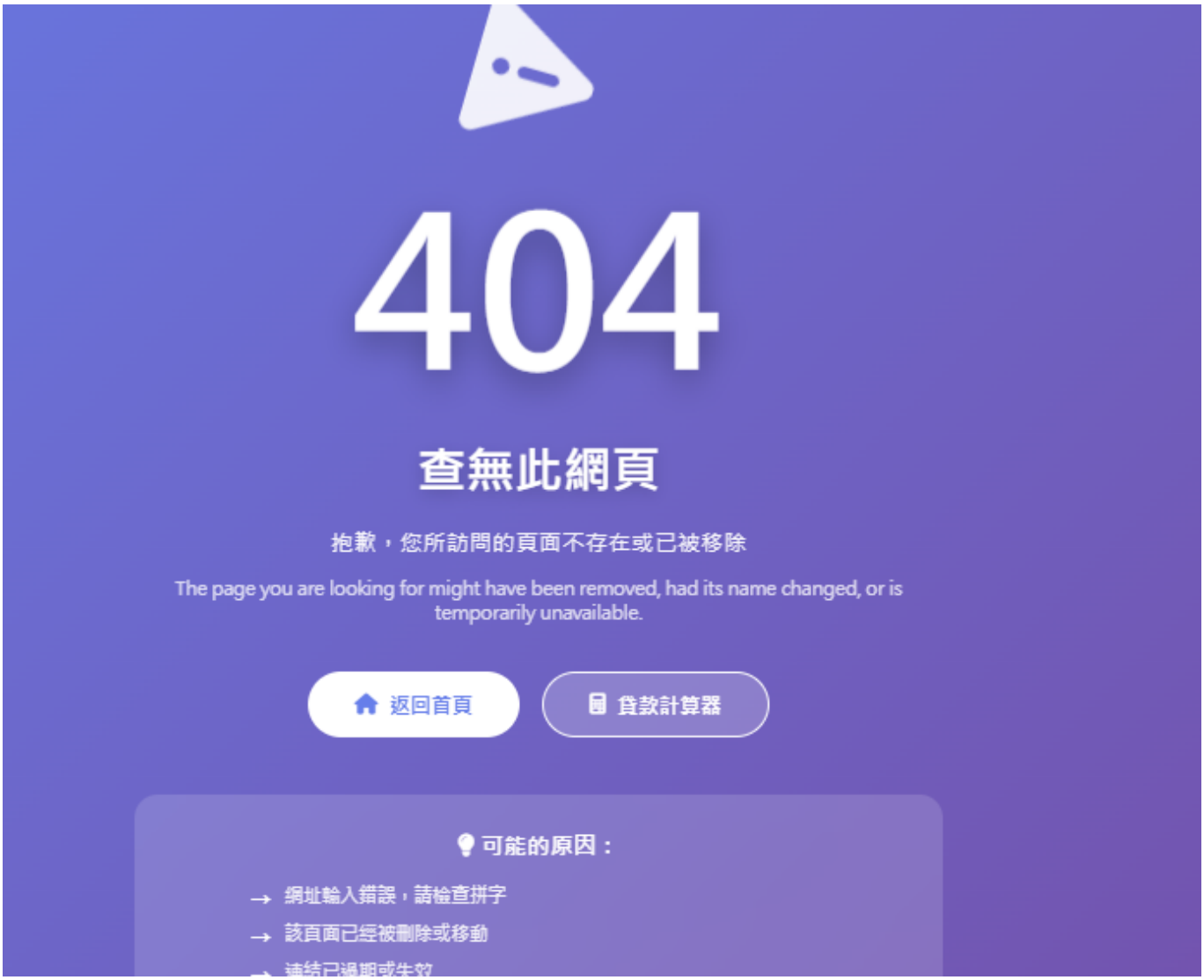
**這裡我分成兩個檔案**

- 第一個程序`2b-refactored.js`是拿來作路由工作。
- 第二個程序`server.js`這裡就是總部，工作是開啟伺服器。


114-1PC1
node_modules
utils
mimeTypes.js
staticFileHandler.js
templateRenderer.js
2b-refactored.js
2b.js
2b整理中_無註解.js
86b73202-5911-47de-8467-8a676d...
answer.md
index.ejs
index2.ejs
index3.ejs
package-lock.json
package.json
script.js
server.js
style.css
style2.css
style3.css
userController.js
server.js > server.listen() callback
const PORT = 3000;
const server = http.createServer(router);
server.listen(PORT, function(){
// 在終端機（控制台）輸出訊息，告知開發者伺服器已啟動
// 使用者可以透過瀏覽器訪問 http://localhost:3000 來查看網站
console.log(`伺服器已啟動！請訪問 http://localhost:${PORT}`);
console.log('可用路由：');
console.log(` - http://localhost:${PORT}`);
console.log(` - http://localhost:${PORT}/calculator`);
console.log(' - 其他路徑將顯示 404 錯誤頁面');
});
問題 輸出 終端機 連接埠 偵錯主控台 GITLENS GITLENS
篩選
Code
[Running] node "d:\code_ALL\node_learning\114-1PC1\server.js"
伺服器已啟動！請訪問 http://localhost:3000
可用路由：
- http://localhost:3000
- http://localhost:3000/calculator
- 其他路徑將顯示 404 錯誤頁面


Jorte
Google 翻譯
哔哩哔哩 (゜-゜)つ...
專案
AI助理
tool
67%
重設
貸款返還計算器
貸款金額（元）
2000
請輸入您要貸款的總金額
年利率（%）
0.02
請輸入年利率百分比
貸款期限（年）
2
請輸入貸款年限
計算
重置
計算結果
每月還款金額：
NT$ 83
總還款金額：
NT$ 2,000
總利息支出：
NT$ 0


貸款返還計算器
貸款資訊輸入
$ 貸款金額（元）
2000
輸入您需要申請的貸款金額
% 年利率（%）
0.03
輸入貸款方案的年利率
貸款期限（年）
3
輸入您的還款期限
立即計算
重新輸入
計算結果明細
每月應繳金額
NT$ 56
總繳款金額
NT$ 2,001
總利息費用
NT$ 1


404
查無此網頁
抱歉，您所訪問的頁面不存在或已被移除
The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.
返回首頁
貸款計算器
可能的原因：
網址輸入錯誤，請檢查拼字
該頁面已經被刪除或移動