

# 練習

1. 建議使用Adobe Acrobat Reader DC 軟體閱讀 PDF 文件，可以更方便在左側以書籤查看每個章節。
2. 本練習採用引導方式，專案從無到有，逐步建立程式。除了每個小練習要自行測試，在「階段驗證」的章節，也請稍微停下來驗證看看是否正確。



## 前置安裝

請在練習實作開始之前，先依照「[平台web基礎培訓練習-前置安裝.pdf](#)」安裝。

## 對應的課程範圍

前端課程總時數30.4小時，應該是依序學習，但因時程緊湊，若來不及在練習之前完成學習，建議優先學習下列課程，練習過程也可以參考下列相關文件或上網搜尋知識。第一次練習是考量時程，所以縮小練習範圍，把內容挪到第二次練習，讓學員爭取更多學習時間。其餘的課程請盡量找時間補足學習，以利於第二次的練習。

有網頁設計HTML、CSS、LESS基礎概念者，建議可先跳到後面的課程學習程式，之後再回頭補足網頁設計學習。關於沒有網頁設計基礎概念者，若來不及在第二次練習完成學習，請快速翻閱官方文件，對於有哪些常用HTML標籤可使用、CSS的樣式如何放入HTML中，至少在練習時能看得懂內容。

總計：0.9小時

課程名稱	時間
P1 01 Angular环境搭建 创建项目 运行项目	00:14:56
P2 02 Angular目录结构分析 核心文件分析 以及创建使用组件	00:22:12
P3 03 Angular组件以及组件中的模板合成完整_转	00:34:10
P13 13 Angular视频教程 路由概述 配置路由 路由重定向 路由选中	00:20:04
P15 15 Angular路由的嵌套 父子路由	00:22:46
P17 17 Angular 中内置模块 以及 自定义模块	00:24:59
P18 18 Angular 自定义模块以及配置路由模块懒加载	00:25:51
P14 14 Angular视频教程 get传值 动态路由合成完整	00:26:54

## 參考相關文件

架構概覽 <https://angular.tw/guide/architecture>

NgModule 簡介 <https://angular.tw/guide/architecture-modules>

Component簡介 <https://angular.tw/guide/architecture-components>

定義一個基本路由 <https://angular.tw/guide/router#defining-a-basic-route>

Router Outlet <https://angular.tw/guide/router#add-the-router-outlet>

Routing module <https://angular.tw/guide/router#milestone-2-routing-module>

延遲載入特性模組 <https://angular.tw/guide/lazy-loading-ngmodules>

設定重新導向 <https://angular.tw/guide/router#setting-up-redirects>

CLI 建立module和component <https://angular.tw/cli/generate>

## 1.建立實作模組、組件、路由

知識點：1. 利用 cli 建立模組、組件、路由。2. 父子路由、路由延遲載入(Lazy loading)、路由載入組件。

### 1.1.建立實作Module

請利用 cli 建立模組下列ImplementationModule和他的routing。

```
CREATE src/app/implementation/implementation-routing.module.ts (258 bytes) CREATE
src/app/implementation/implementation.module.ts (312 bytes)
```

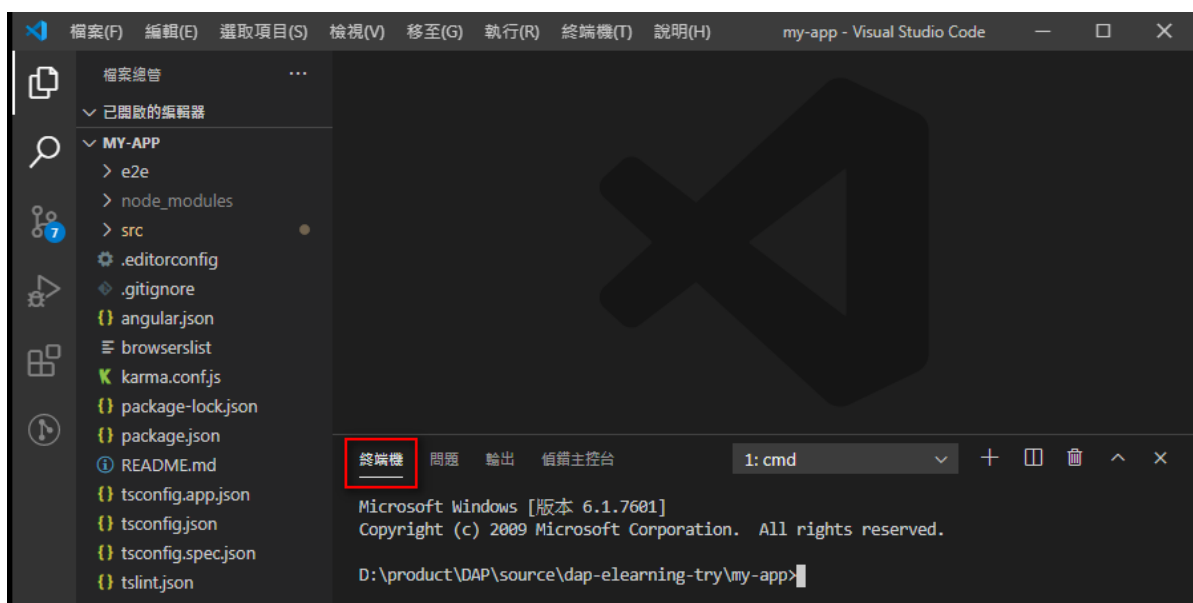
小試身手：

(1)請打開VS Code

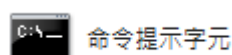
(2)點選VS Code最上方功能列的「檔案」，再選「開啟資料夾」，選擇專案資料夾「my-app」。



(3)開啟終端機



終端機如同系統提供輸入指令的介面



(4)請在終端機輸入下列cli指令，下列二選一即可。

```
1 | ng g module implementation --routing
```

如果在開發者電腦的環境中，同時存在著不同版本的cli，並且你的專案中使用的cli和安裝在global的版本不同時，則可以透過npm run script的方式確保以本專案所安裝的版本進行，如下指令，不論你是裝一個cli版本或多個版本都適用。

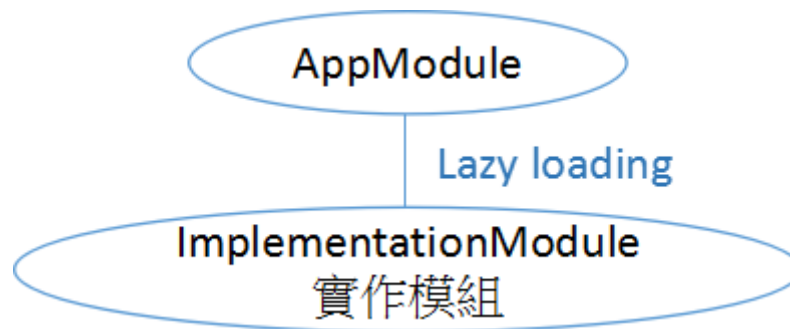
```
1 | npm run ng -- g module implementation --routing
```

(5)恭喜你！瞬間產生了兩個程式囉！

## 1.2.AppModule延遲載入ImplementationModule

在AppRoutingModule路由載入ImplementationModule，在src\app\app-routing.module.ts設定routes，路由path: ''，並以loadChildren延遲載入ImplementationModule。

完成架構：



小試身手：

src\app\app-routing.module.ts

```
1 | const routes: Routes = [
2 |   {
3 |     path: '',
4 |     pathMatch: 'prefix',
5 |     loadChildren: () =>
6 |       import('./implementation/implementation.module').then(m =>
7 |         m.ImplementationModule)
8 |   }
9 | ];
```

別懷疑，這樣你就輕鬆完成了延遲載入！

### 1.3.建立登入頁LoginModule

請利用 cli 建立LoginModule和他的routing。

CREATE src/app/implementation/login/login-routing.module.ts (249 bytes) CREATE src/app/implementation/login/login.module.ts (276 bytes)

### 1.4.建立登入頁LoginComponent

請利用 cli 建立LoginComponent，此component是屬於LoginModule。

CREATE src/app/implementation/login/login.component.html (20 bytes) CREATE src/app/implementation/login/login.component.spec.ts (621 bytes) CREATE src/app/implementation/login/login.component.ts (272 bytes) CREATE src/app/implementation/login/login.component.less (0 bytes) UPDATE src/app/implementation/login/login.module.ts (342 bytes)

小試身手：

```
1 | npm run ng -- g component implementation/login -m implementation/login
```

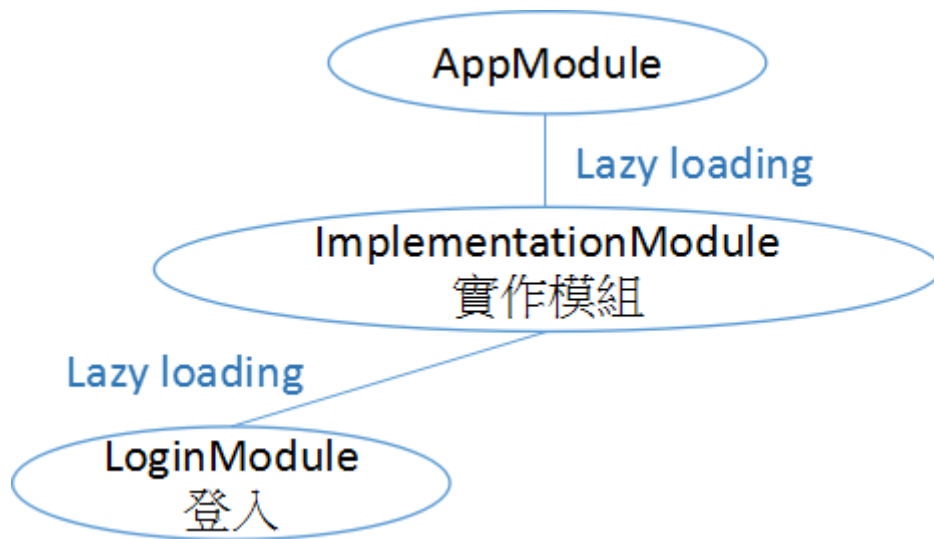
### 1.5.撰寫登入頁畫面

1. src\app\app.component.html 只保留 `<router-outlet></router-outlet>`。
2. css移入src\app\implementation\login\login.component.less。
3. 其它HTML內容移入src\app\implementation\login\login.component.html。
4. src\app\app.component.ts 裡面的title變數，移入LoginComponent

### 1.6.ImplementationModule延遲載入LoginModule

在ImplementationRoutingModule延遲載入LoginModule，路由path: 'login'。

完成架構：



小試身手：

src\app\implementation\implementation-routing.module.ts

```

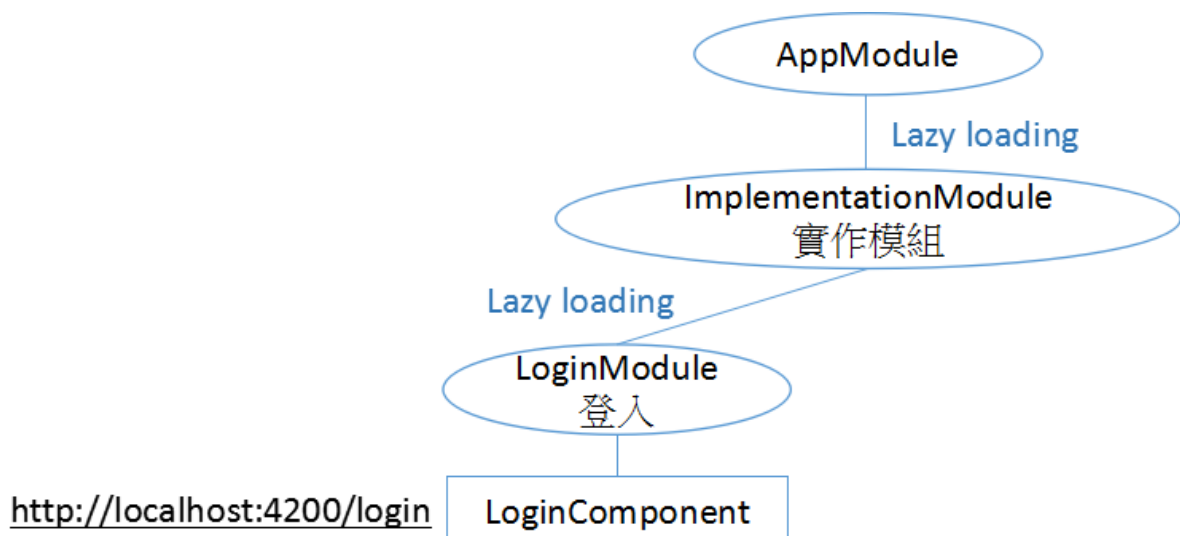
1  const routes: Routes = [
2    {
3      path: 'login',
4      pathMatch: 'full',
5      loadChildren: () => import('./login/login.module').then(m =>
6        m.LoginModule)
7    }
8  ];

```

## 1.7.登入頁路由載入LoginComponent

在LoginRoutingModule載入LoginComponent，路由path: "。

完成架構：



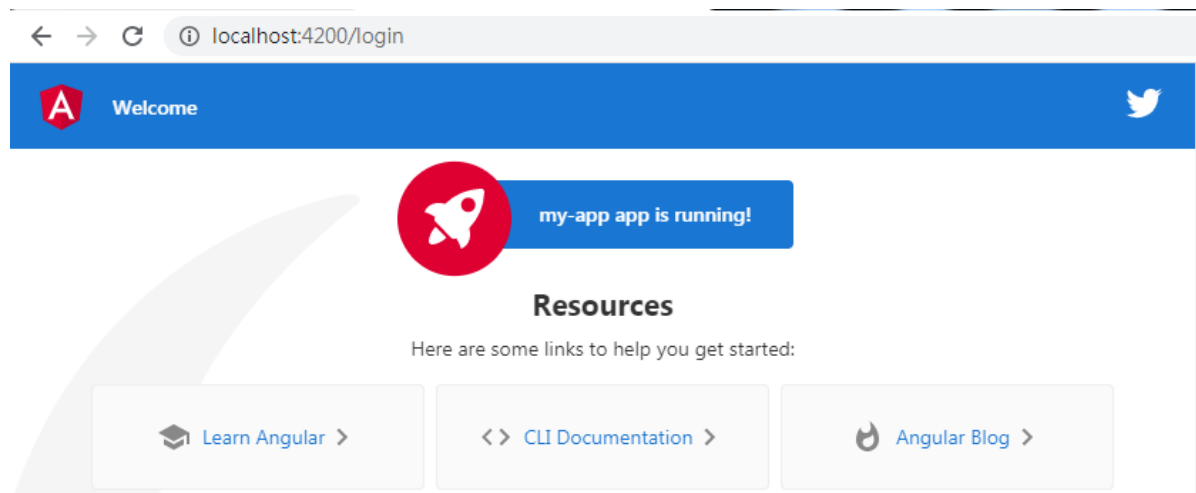
小試身手：

src/app/implementation/login/login-routing.module.ts

```
1 import { LoginComponent } from './login.component';
2
3 const routes: Routes = [
4   {
5     path: '',
6     component: LoginComponent
7   }
8 ];
```

## ► 階段驗證

瀏覽器開啟 <http://localhost:4200/login>



## 1.8.建立視覺佈局LayoutComponent

請利用 cli 建立LayoutComponent，此component是屬於implementation.module。

CREATE src/app/implementation/layout/layout.component.html (21 bytes) CREATE  
src/app/implementation/layout/layout.component.spec.ts (628 bytes) CREATE  
src/app/implementation/layout/layout.component.ts (276 bytes) CREATE  
src/app/implementation/layout/layout.component.less (0 bytes) UPDATE  
src/app/implementation/implementation.module.ts (388 bytes)

## 1.9.在LayoutComponent顯示子路由

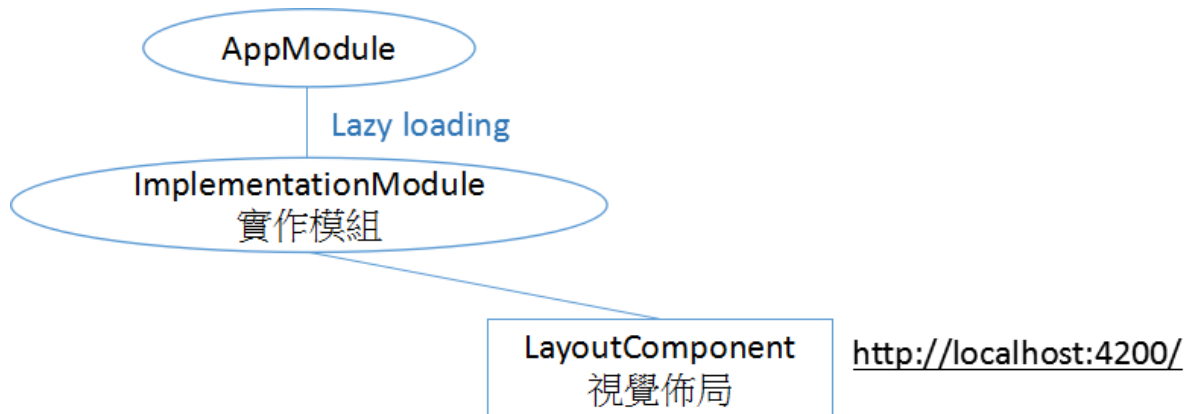
請利用 `<router-outlet>` 讓LayoutComponent顯示子路由的內容。

```
1 <p>layout works!</p>
2 <router-outlet></router-outlet>
```

## 1.10.ImplementationModule路由載入LayoutComponent

在ImplementationRoutingModule載入LayoutComponent，路由path: ''。

完成架構：



小試身手：

```
1 import { LayoutComponent } from './layout/layout.component';
2
3 const routes: Routes = [
4   {
5     path: '',
6     pathMatch: 'prefix',
7     component: LayoutComponent
8   },
9   {
10    path: 'login',
11    pathMatch: 'full',
12    loadChildren: () => import('./login/login.module').then(m =>
13      m.LoginModule)
14  }
15 ];
```

## 1.11.建立首頁HomeModule與路由

請利用 cli 建立HomeModule和他的routing。

CREATE src/app/implementation/home/home-routing.module.ts (248 bytes) CREATE  
src/app/implementation/home/home.module.ts (272 bytes)



## 1.12.建立首頁HomeComponent

請利用 cli 建立 HomeComponent，此 component 是屬於 home.module。

CREATE src/app/implementation/home/home.component.html (19 bytes) CREATE  
src/app/implementation/home/home.component.spec.ts (614 bytes) CREATE  
src/app/implementation/home/home.component.ts (268 bytes) CREATE  
src/app/implementation/home/home.component.less (0 bytes) UPDATE  
src/app/implementation/home/home.module.ts (335 bytes)

## 1.13.HomeModule路由載入HomeComponent

在 HomeRoutingModule 載入 HomeComponent，路由 path: ''。

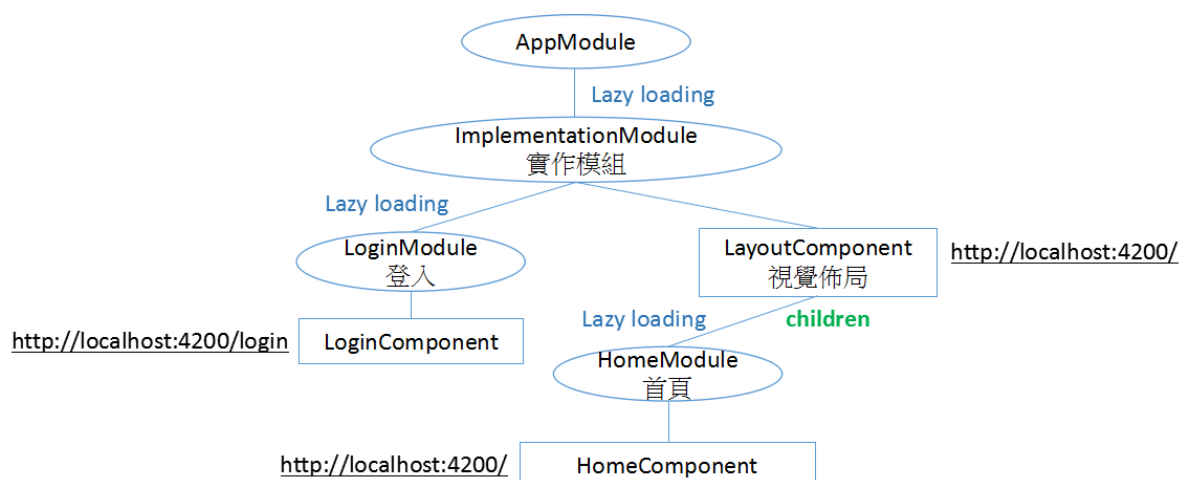
小試身手：

```
1 import { HomeComponent } from './home.component';
2
3 const routes: Routes = [
4   {
5     path: '',
6     component: HomeComponent
7   }
8 ];
```

## 1.14.ImplementationModule延遲載入HomeModule

在 ImplementationRoutingModule 路由 path: '' 的區塊，增加子路由 children，延遲載入 HomeModule，子路由 path: ''。

完成架構：



小試身手：

```

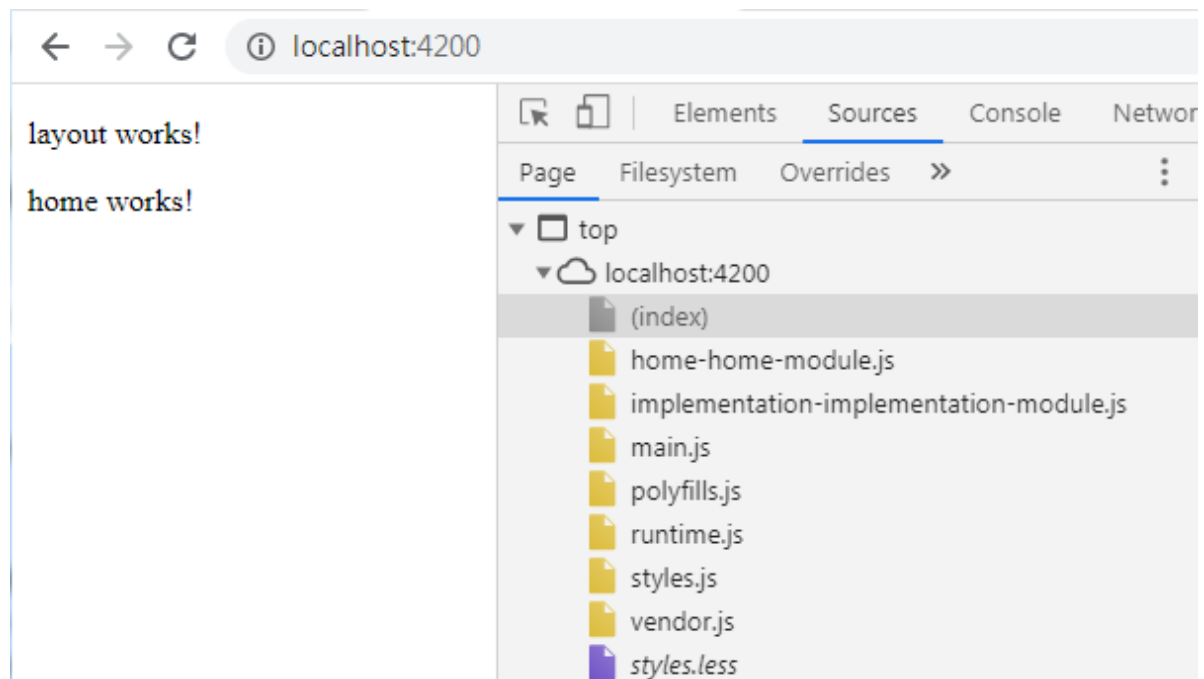
1  const routes: Routes = [
2    {
3      path: '',
4      pathMatch: 'prefix',
5      component: LayoutComponent,
6      children: [
7        {
8          path: '',
9          pathMatch: 'prefix',
10         loadChildren: () => import('./home/home.module').then(m =>
m.HomeModule)
11       }
12     ]
13   },
14   {
15     path: 'login',
16     pathMatch: 'full',
17     loadChildren: () => import('./login/login.module').then(m =>
m.LoginModule)
18   }
19 ];

```

## ► 階段驗證 - Layout和Home

Chrome瀏覽器開啟 <http://localhost:4200/>

1. 畫面有顯示LoginComponent和HomeComponent。
2. 按鍵盤F12，查看如下圖「Sources/Page」。必須延遲載入的LoginModule還不會出現。



## 2.建立應用模組、作業

知識點：1. 利用 cli 建立模組、組件、路由。2. 父子路由、路由延遲載入(Lazy loading)、路由載入組件。3. 路由導航。

應用模組、作業都建立在「src/app/implementation/programs」目錄底下。

新增模組「物料管理material」，新增作業「採購作業purchasing」。

### 2.1.建立應用模組MaterialModule

請利用 cli 建立模組下列Module和他的routing。

CREATE src/app/implementation/programs/material/material-routing.module.ts (252 bytes)  
CREATE src/app/implementation/programs/material/material.module.ts (288 bytes)

### 2.2.建立作業PurchasingModule

請利用 cli 建立下列Module和他的routing。

CREATE src/app/implementation/programs/material/purchasing/purchasing-routing.module.ts  
(254 bytes) CREATE  
src/app/implementation/programs/material/purchasing/purchasing.module.ts (296 bytes)

### 2.3.建立作業主component

請利用 cli 建立PurchasingModule的PurchasingComponent。

CREATE src/app/implementation/programs/material/purchasing/purchasing.component.html (25  
bytes) CREATE  
src/app/implementation/programs/material/purchasing/purchasing.component.spec.ts (656  
bytes) CREATE src/app/implementation/programs/material/purchasing/purchasing.component.ts  
(292 bytes) CREATE  
src/app/implementation/programs/material/purchasing/purchasing.component.less (0 bytes)  
UPDATE src/app/implementation/programs/material/purchasing/purchasing.module.ts (377  
bytes)

小試身手：

```
1 | npm run ng -- g component implementation/programs/material/purchasing -m  
   | implementation/programs/material/purchasing
```

## 2.4.建立作業子component

查詢列表頁：

請利用 cli 建立PurchasingModule的PurchasingListComponent。

```
CREATE src/app/implementation/programs/material/purchasing/purchasing-list/purchasing-list.component.html (30 bytes) CREATE
src/app/implementation/programs/material/purchasing/purchasing-list/purchasing-list.component.spec.ts (685 bytes) CREATE
src/app/implementation/programs/material/purchasing/purchasing-list/purchasing-list.component.ts (311 bytes) CREATE
src/app/implementation/programs/material/purchasing/purchasing-list/purchasing-list.component.less (0 bytes) UPDATE
src/app/implementation/programs/material/purchasing/purchasing.module.ts (489 bytes)
```

修改頁：

請利用 cli 建立PurchasingModule的PurchasingModifyComponent。

```
CREATE src/app/implementation/programs/material/purchasing/purchasing-modify/purchasing-modify.component.html (32 bytes) CREATE
src/app/implementation/programs/material/purchasing/purchasing-modify/purchasing-modify.component.spec.ts (699 bytes) CREATE
src/app/implementation/programs/material/purchasing/purchasing-modify/purchasing-modify.component.ts (319 bytes) CREATE
src/app/implementation/programs/material/purchasing/purchasing-modify/purchasing-modify.component.less (0 bytes) UPDATE
src/app/implementation/programs/material/purchasing/purchasing.module.ts (609 bytes)
```

小試身手：

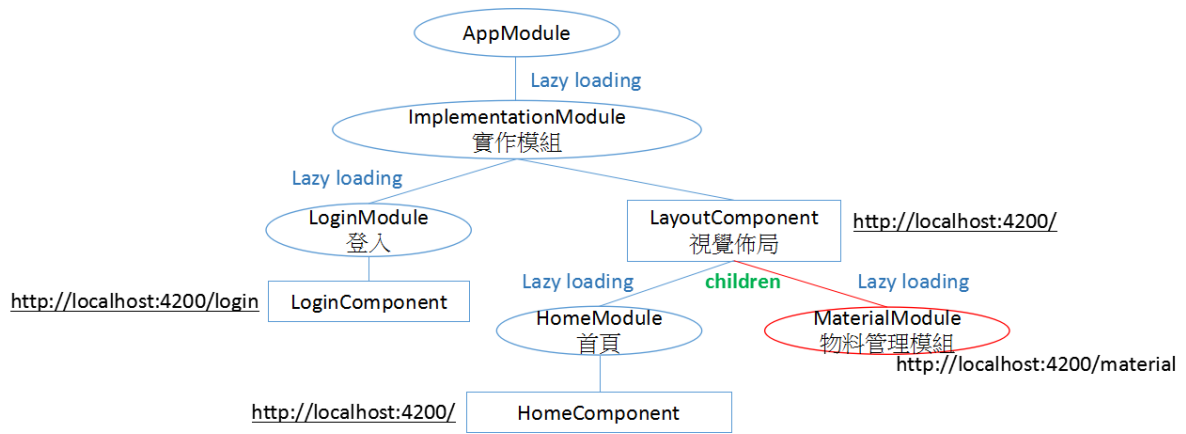
```
1 npm run ng -- g component
  implementation/programs/material/purchasing/purchasing-list -m
  implementation/programs/material/purchasing
```

```
1 npm run ng -- g component
  implementation/programs/material/purchasing/purchasing-modify -m
  implementation/programs/material/purchasing
```

## 2.5.延遲載入應用模組

在ImplementationRoutingModule路由path: "的區塊，增加子路由children，延遲載入MaterialModule，子路由path: 'material'。

完成架構：



小試身手：

```

1  const routes: Routes = [
2    {
3      path: '',
4      pathMatch: 'prefix',
5      component: LayoutComponent,
6      children: [
7        {
8          path: '',
9          pathMatch: 'prefix',
10         loadChildren: () => import('./home/home.module').then(m =>
m.HomeModule)
11        },
12        {
13          path: 'material', // 應用模組Material Management
14          pathMatch: 'prefix',
15          loadChildren: () =>
import('./programs/material/material.module').then(m => m.MaterialModule)
16        }
17      ]
18    },
19    {
20      path: 'login',
21      pathMatch: 'full',
22      loadChildren: () => import('./login/login.module').then(m =>
m.LoginModule)
23    }
24  ];

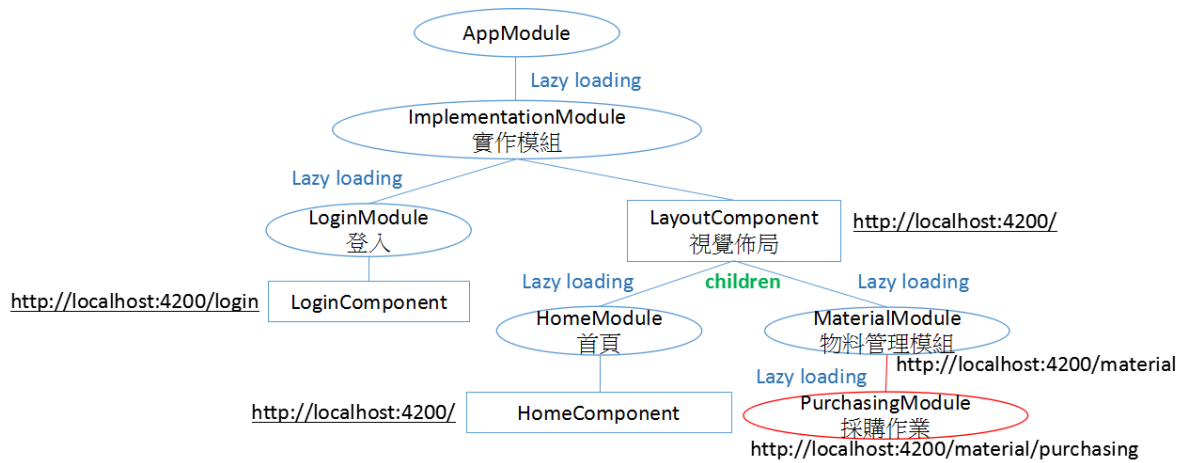
```

## 2.6.應用模組延遲載入作業

(1)在MaterialModule新增路由path: 'purchasing'，延遲載入PurchasingModule。

(2)預設路由path: ''，重新導向path: 'purchasing'。

完成架構：

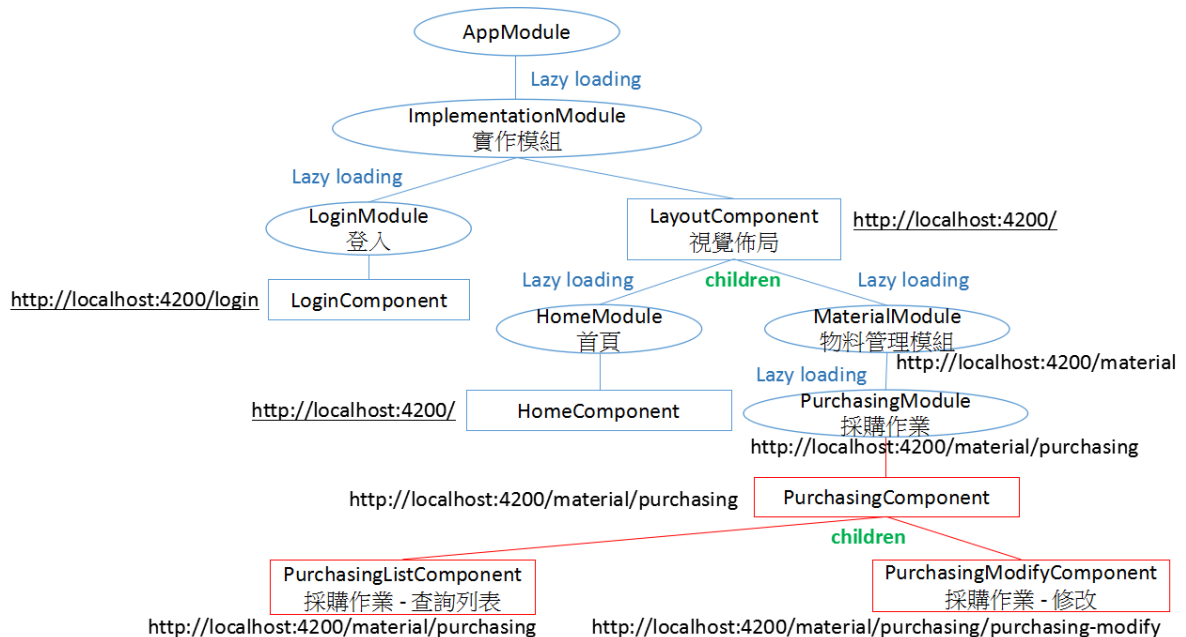


## 2.7.作業主component加子路由載子component

(1)在PurchasingComponent的畫面加上 `<router-outlet>`。

(2)PurchasingRoutingModule路由 `path: ''` 載入PurchasingComponent。並在這底下 `children` 加入子路由`path: ''` 載入PurchasingListComponent。另一個子路由`path: 'purchasing-modify'` 載入PurchasingModifyComponent。

完成架構：



小試身手：

purchasing.component.html

```
1 | <router-outlet></router-outlet>
```

在purchasing-routing.module.ts中設定作業路由routes

```
1 | const routes: Routes = [
2 |   {
```

```

3     path: '',
4     pathMatch: 'prefix',
5     component: PurchasingComponent,
6     children: [
7       {
8         path: '',
9         component: PurchasingListComponent
10      },
11      {
12        path: 'purchasing-modify',
13        component: PurchasingModifyComponent
14      }
15    ]
16  }
17 ];

```

## 2.8. 子頁面之間的超連結

purchasing-list.component.html改為下列內容，並在 `<a>` 標籤加上 `routerLink` 屬性，可連至 PurchasingModifyComponent。

```
1 <a>進入修改頁</a>
```

purchasing-modify.component.html改為下列內容，並加上 `routerLink`，可返回 PurchasingListComponent。

```
1 <a>返回</a>
```

### ►階段驗證 - 作業路由

- (1) 瀏覽器開啟 <http://localhost:4200/material>，預設導向 <http://localhost:4200/material/purchasing>
- (2) 點擊「進入修改頁」，會連結到 <http://localhost:4200/material/purchasing/purchasing-modify>
- (3) 點擊「返回」，會連結到 <http://localhost:4200/material/purchasing>

## 2.9. 導航

src\app\implementation\layout\layout.component.less 加入下列樣式

```

1 .my-layout {
2   width: 100%;
3   height: 100%;
4   display: block;
5   position: fixed;

```

```

6   }
7
8   .my-layout-content {
9     border: 1px #ccc dashed;
10  }
11
12  .menu {
13    height: 100%;
14    width: 150px;
15    background-color: #cdf5a0;
16    position: relative;
17    float: left;
18  }
19
20  .menu-item {
21    cursor: pointer;
22    color: #009688;
23    margin-top: 2px;
24    padding: 5px;
25  }

```

src\app\implementation\layout\layout.component.html改為下列內容，並加上`routerLink` 完成連結和 `<router-outlet>` 顯示網頁內容。

```

1  <div class="my-layout">
2    <div class="menu">
3      <div routerLink="/" class="menu-item">首頁</div>
4      <div class="menu-item">採購作業</div> <!-- 練習：連結
到"/material/purchasing" -->
5      <div class="menu-item">登出</div> <!-- 練習：連結到"/login" -->
6    </div>
7    <div class="my-layout-content">
8      <!-- 網頁內容 -->
9
10   </div>
11 </div>

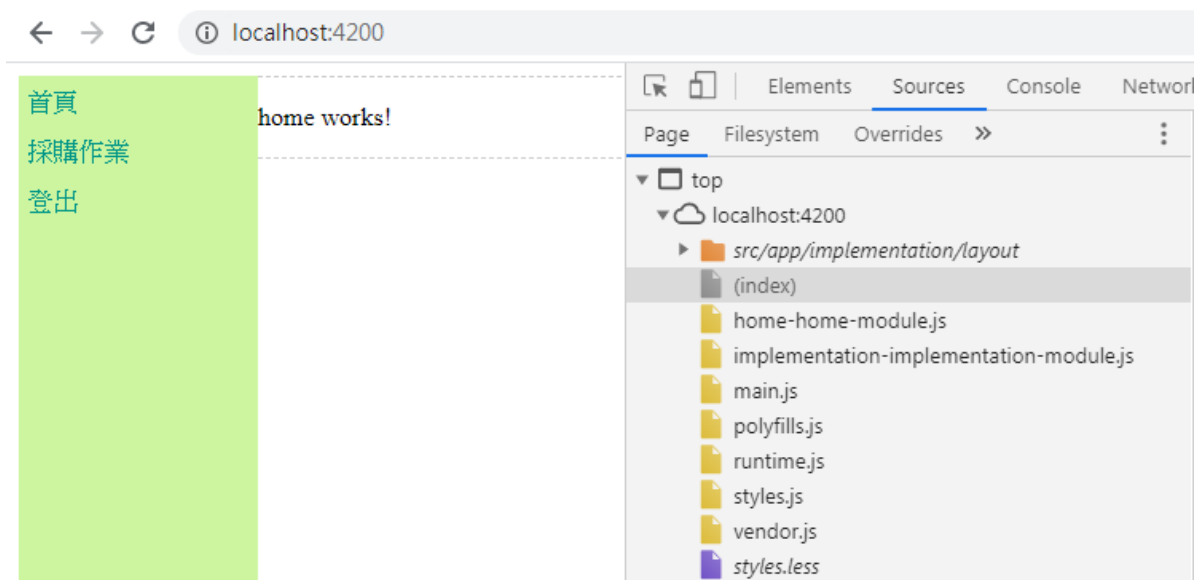
```

## ►階段驗證 - 導航

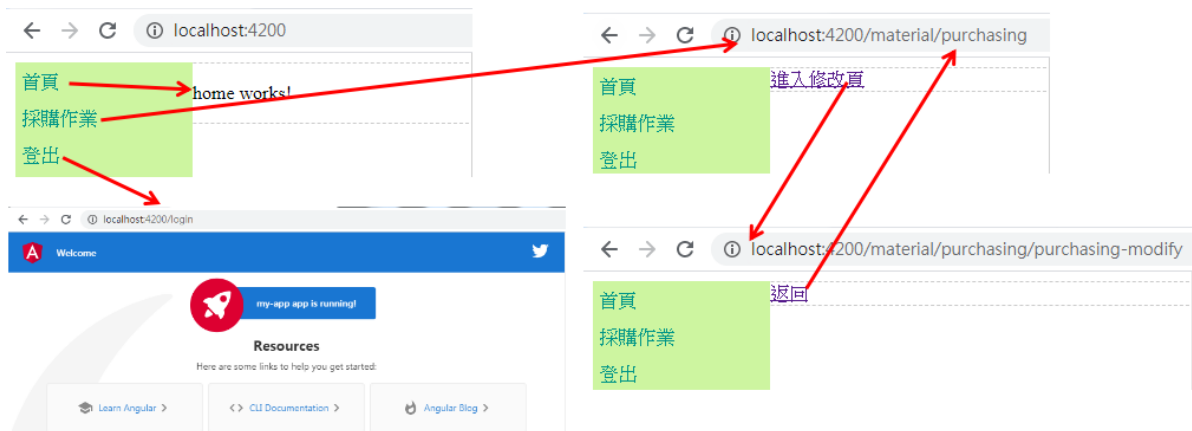
瀏覽器開啟 <http://localhost:4200/>

(1)按鍵盤F12，查看如下圖「Sources/Page」。必須延遲載入的LoginModule和MaterialModule還不會出現。





(2)進行下列路由連結



檔案與目錄

```

1 |-- src/app/
2 |   |-- app-routing.module.ts
3 |   |-- app.component.html
4 |   |-- app.component.less
5 |   |-- app.component.spec.ts
6 |   |-- app.component.ts
7 |   |-- app.module.ts
8 |   |-- implementation
9 |     |-- implementation-routing.module.ts
10 |     |-- implementation.module.ts
11 |     |-- home
12 |       |-- home-routing.module.ts
13 |       |-- home.component.html
14 |       |-- home.component.less
15 |       |-- home.component.spec.ts
16 |       |-- home.component.ts
17 |       |-- home.module.ts
18 |     |-- layout
19 |       |-- layout.component.html
20 |       |-- layout.component.less
21 |       |-- layout.component.spec.ts
22 |       |-- layout.component.ts
23 |     |-- login

```

```
24 | |-- login-routing.module.ts
25 | |-- login.component.html
26 | |-- login.component.less
27 | |-- login.component.spec.ts
28 | |-- login.component.ts
29 | |-- login.module.ts
30 |-- programs
31 | |-- material
32 | |-- material-routing.module.ts
33 | |-- material.module.ts
34 | |-- purchasing
35 | |-- purchasing-routing.module.ts
36 | |-- purchasing.component.html
37 | |-- purchasing.component.less
38 | |-- purchasing.component.spec.ts
39 | |-- purchasing.component.ts
40 | |-- purchasing.module.ts
41 | |-- purchasing-list
42 | |-- purchasing-list.component.html
43 | |-- purchasing-list.component.less
44 | |-- purchasing-list.component.spec.ts
45 | |-- purchasing-list.component.ts
46 | |-- purchasing-modify
47 | |-- purchasing-modify.component.html
48 | |-- purchasing-modify.component.less
49 | |-- purchasing-modify.component.spec.ts
50 | |-- purchasing-modify.component.ts
```

## 練習題交件

請把「**src**」資料夾壓縮成 **zip** 檔，再交件，其他檔案不用交。