

# 极速 Angular 开发 效能调校技巧



多奇数位创意有限公司

技术总监 黄保翕 (Will 保哥)

<http://blog.miniasp.com/>

# Will 保哥 (黄保翕)



- 现任多奇数位创意有限公司技术总监
- 台湾 Angular 社群核心成员 (Angular Taiwan)
- 2018 年荣获 [Angular/Web GDE](#) (谷歌开发者专家)
- 2019 年荣获 [Microsoft RD](#) (微软技术社群区域总监)
- 2008 至今连续 12 度获选 Microsoft MVP 最有价值专家
- 每年皆有数十场公开演讲与各式前后端技术培训课程
- 出版著作有 ASP.NET MVC 2 开发实战、Windows Phone 开发实战、ASP.NET MVC 4 开发实战等书籍。



ng-tw

# 效能调校首重观念

我们这行每天有 2000 万人在写 Bugs





天下武功，无坚不破，唯快不破



2019 ngChina 开发者大会





# 效能调校的心法

- 什么叫慢？
  - 用户说慢就是慢
  - 网络慢 or 执行慢
- 量化的数据
  - 善用工具查核数据
  - 善用工具模拟情境
- 什么叫快？
  - 先求不伤身体 (No Bug)
  - 再讲求实际效果 (Speed)
- 快了，然后呢？



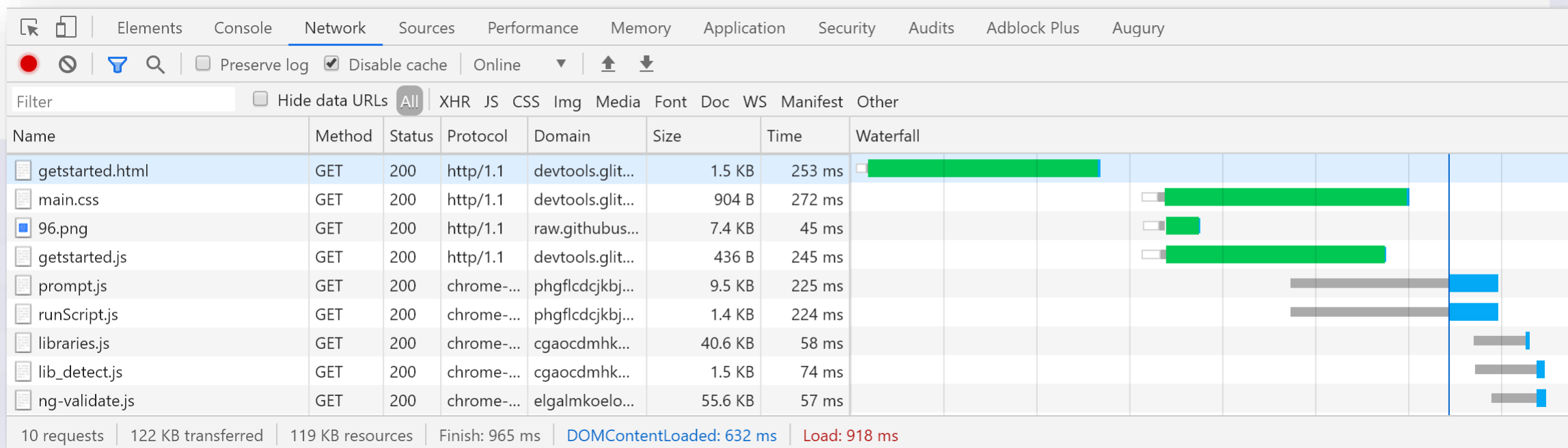
ng-tw

什么叫慢



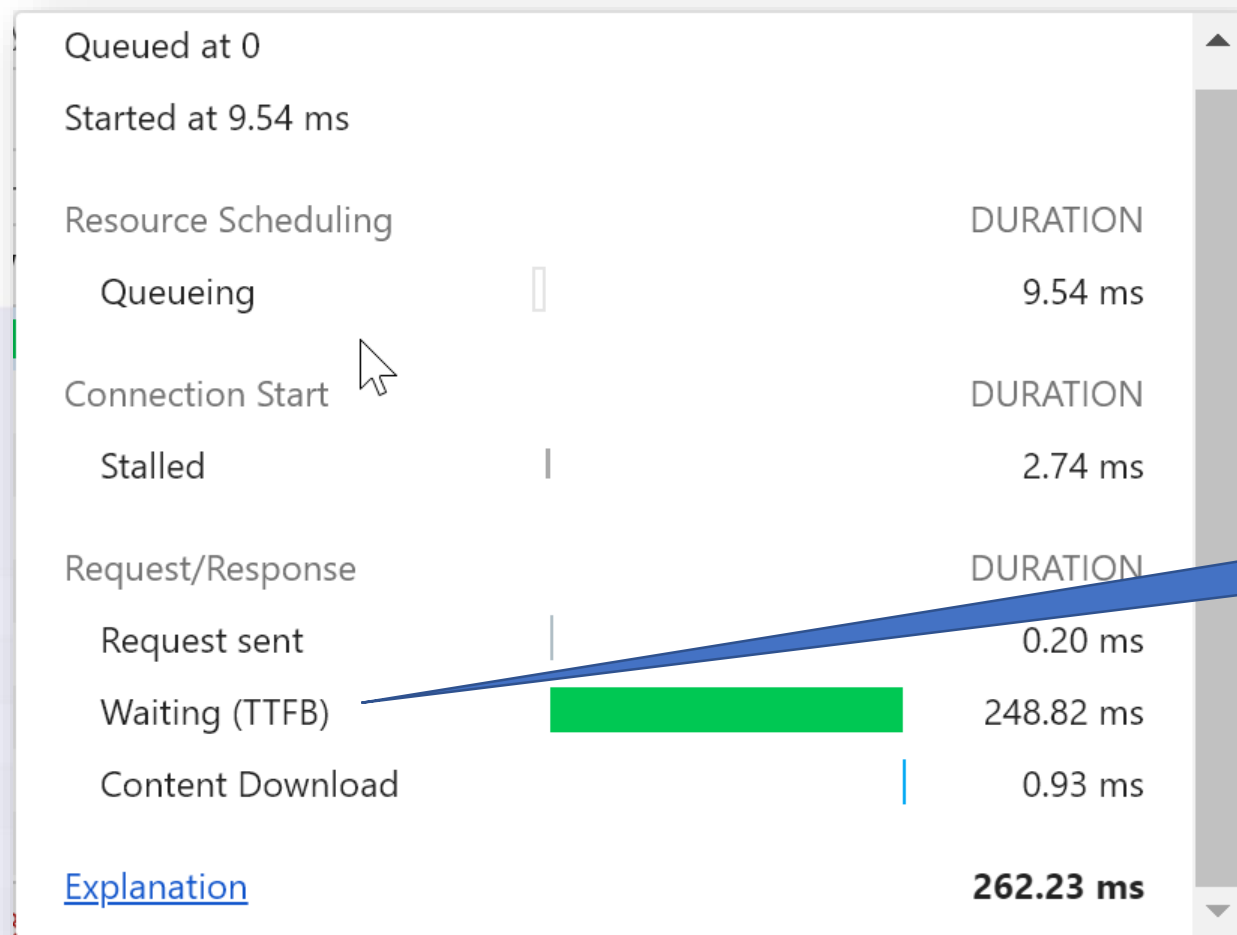
# 网络慢

- Chrome DevTools 开发者工具 > Network 标签





# 每个 HTTP 要求所呈现的网络信息细节

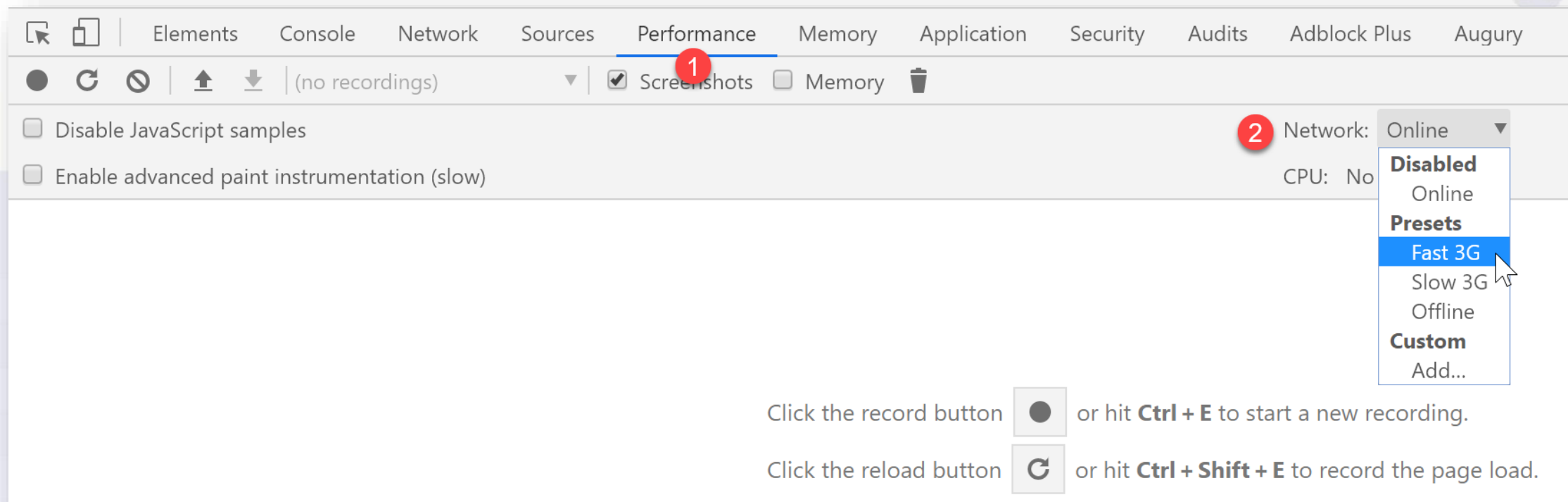


TTFB  
(Time To First Byte)



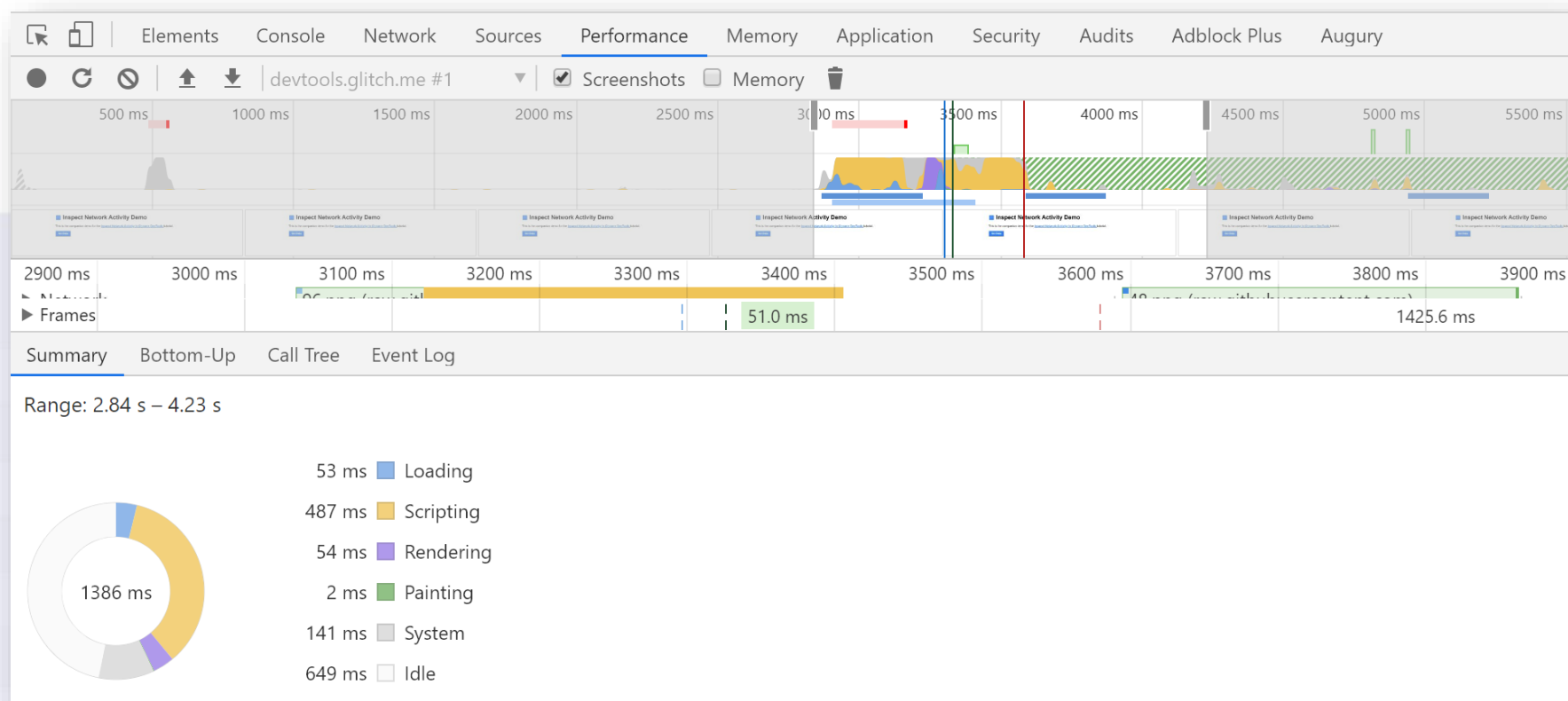
# 模拟网速慢的情境

- Chrome DevTools 开发者工具 > Performance 标签 > Network



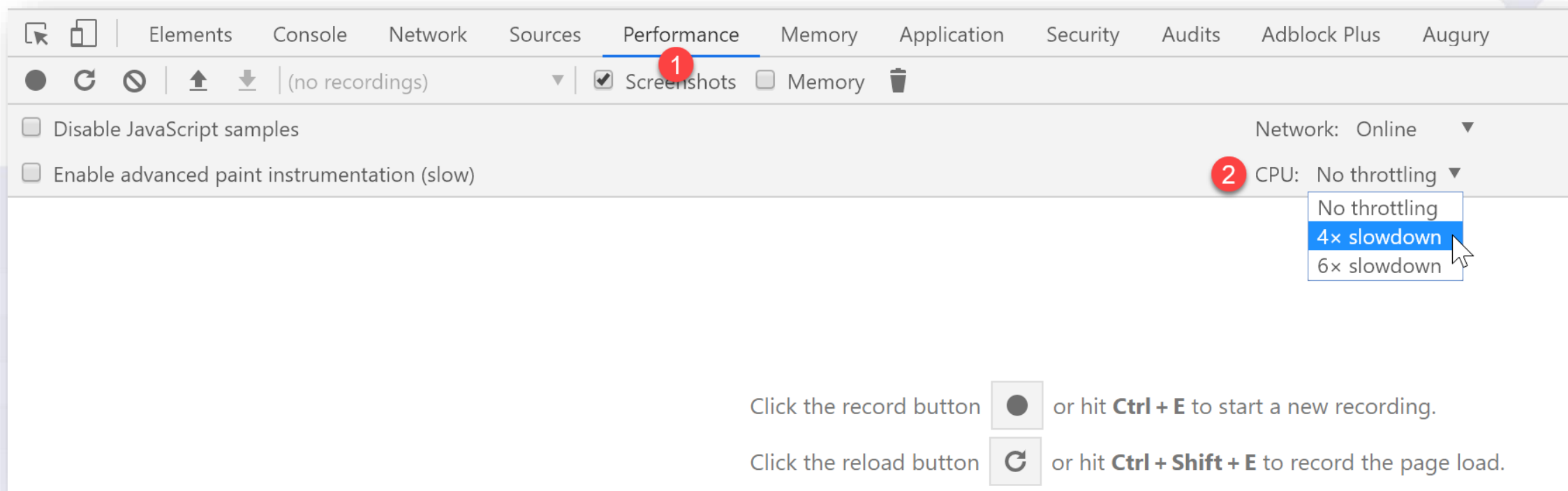
# 执行慢

- 善用 [Chrome DevTools 开发者工具](#) > Performance 标签



# 模拟 CPU 慢的情境

- Chrome DevTools 开发者工具 > Performance 标签 > CPU





# 善用 Lighthouse 审查工具

- [Lighthouse](#) 是一个开源的自动化工具，用于改进网络应用的质量。
- 安装 [Lighthouse Chrome 扩展程序](#)。



# 执行 Lighthouse 审查工具

- Chrome DevTools 开发者工具 > Audits 标签

Learn more'. The panel is divided into three sections: 'Device' with 'Mobile' and 'Desktop' (selected) options; 'Audits' with a list of categories: Performance, Progressive Web App, Best practices, Accessibility, and SEO, all of which are checked; and 'Throttling' with 'Simulated Slow 4G, 4x CPU Slowdown' (selected) and 'Applied Slow 4G, 4x CPU Slowdown' options." data-bbox="65 306 999 1000"/>

Elements Console Network Sources Performance Memory Application Security Audits Adblock Plus Augury

+(new audit)

## Audits

Identify and fix common problems that affect your site's performance, accessibility, and user experience. [Learn more](#)

Device

- ☐ Mobile
- ☒ Desktop

✓ Audits

- ☒ Performance
- ☒ Progressive Web App
- ☒ Best practices
- ☒ Accessibility
- ☒ SEO

⚙ Throttling

- ☒ Simulated Slow 4G, 4x CPU Slowdown
- ☐ Applied Slow 4G, 4x CPU Slowdown

nq-tw

# 掌握变更侦测机制

Change Detection



# 关于 Angular 应用程序状态

- Angular 应用程序是由组件组成的
- 每一个组件里储存着许多内部状态 ( 状态就是属性 )
- 而内部状态可以是任何类型的对象 (原始型别/物件型别)
- 组件的内部状态经常与网页的用户接口 (View) 绑定 (Binding)





# 触发变更侦测的情境

- 常见的状态变更原因
  - 所有的 DOM 事件
    - click, change, input, ...
  - XMLHttpRequest
    - 从远程服务器抓取数据
  - Timers
    - setTimeout(), setInterval()
- 这些都是 异步操作 (async)
- 每当异步操作执行完成，应用程序状态就可能会发生改变
- 此时就可能需要通知 View 更新页面中所有绑定 (Bindings)



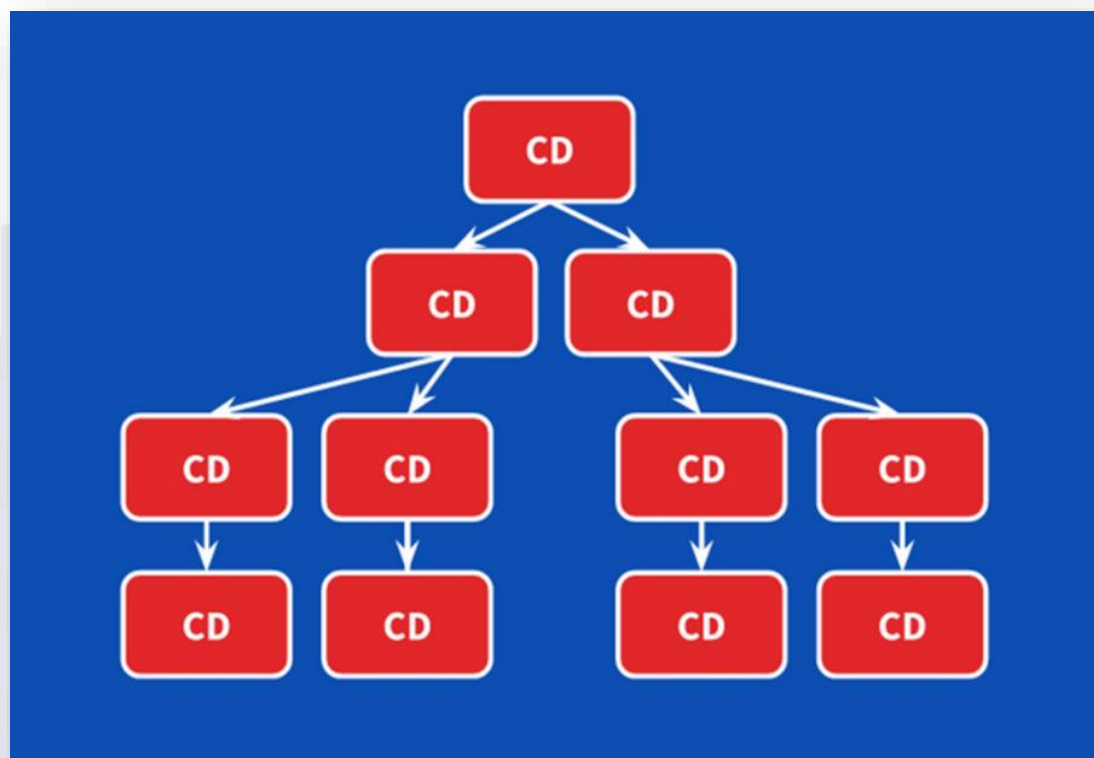
# 关于 Angular 的变更侦测策略

- 每个组件都可以定义自己的[变更侦测策略](#)
  - `ChangeDetectionStrategy.Default`
    - 变更侦测会在每一次异步事件发生时执行
  - `ChangeDetectionStrategy.OnPush`
    - 变更侦测会在当组件中有 `@Input()` 属性的源数据变更时 (参考变更)
- 每次的变更侦测作业都会从根组件开始执行
  - `applicationRef.tick();`

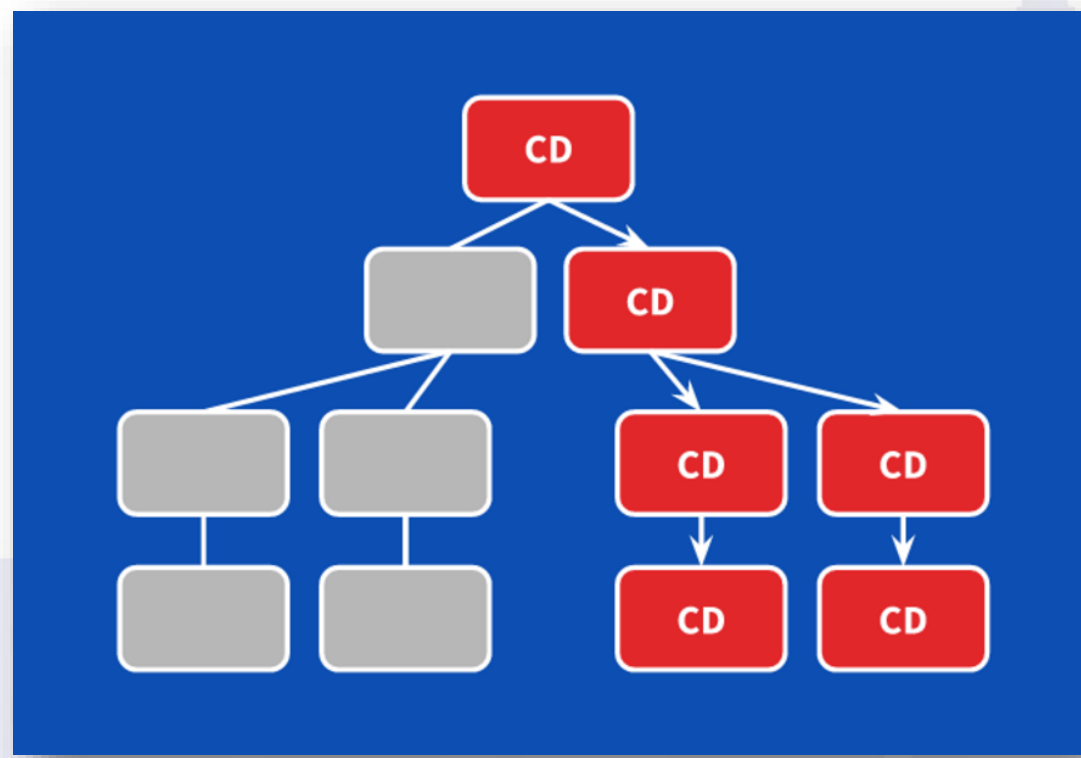


# 变更侦测树 (Change Detection Tree)

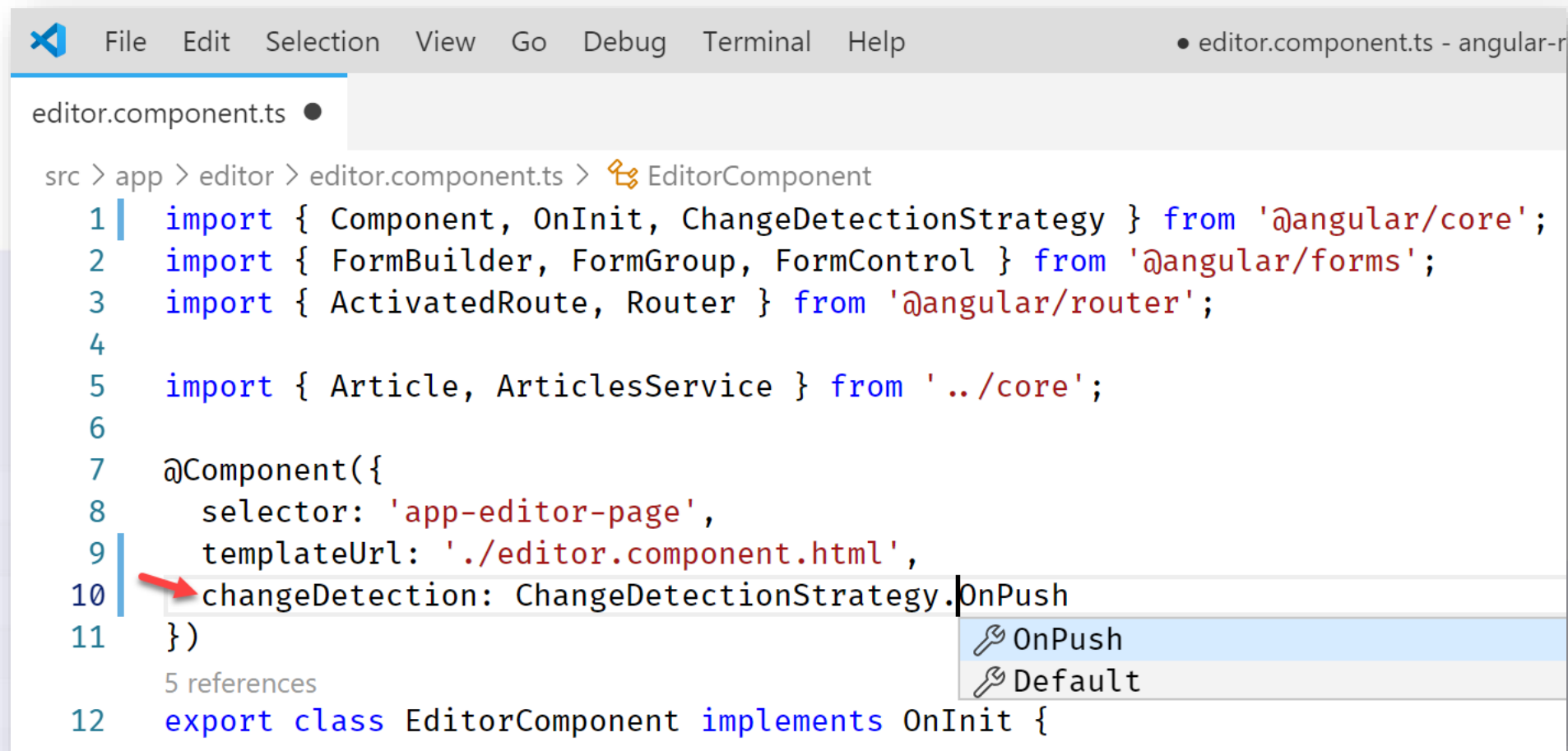
ChangeDetectionStrategy.Default



ChangeDetectionStrategy.OnPush



# 调整组件的变更侦测策略



```
editor.component.ts •
src > app > editor > editor.component.ts > EditorComponent
1 import { Component, OnInit, ChangeDetectionStrategy } from '@angular/core';
2 import { FormBuilder, FormGroup, FormControl } from '@angular/forms';
3 import { ActivatedRoute, Router } from '@angular/router';
4
5 import { Article, ArticlesService } from '../core';
6
7 @Component({
8   selector: 'app-editor-page',
9   templateUrl: './editor.component.html',
10  changeDetection: ChangeDetectionStrategy.OnPush
11 })
12 export class EditorComponent implements OnInit {
```





# 计算变更侦测次数

- <https://stackblitz.com/edit/ng8-change-detection-demo>
- 避免在 Angular 范本中执行函式 (包含 getter 函式)
  - **{{ debug() }}**
  - 计算 Angular Expression 的执行次数
  - 实验 `document.onmousemove` 如何影响效能
- 正式环境记得呼叫 **enableProdMode()**; 函式
  - 变更侦测次数会从 2 降到 1 (效能提升) (Bug 提升)



# 关闭 Change Detection 的多种作法

- 执行在 zone.js 之外

```
this._zone.runOutsideAngular(() => { ... });
```

- 关闭变更侦测

```
this._cd.detach();
```

- 从 main.ts 设定 NgZone 提供者 ( 注意 src/polyfills.ts 也要移除 zone.js )

```
platformBrowserDynamic()  
  .bootstrapModule(AppModule, { ngZone: 'noop' });
```



# 触发 Change Detection 的多种作法

- 执行 [NgZone](#) 的 run() 方法

```
this._zone.run(() => { ... });
```

- 执行 [ChangeDetectorRef](#) 的 detectChanges() 方法

```
this._cd.detectChanges();
```

- 执行 [ApplicationRef](#) 的 tick() 方法

```
this._app.tick();
```



# 完整的变更侦测互动学习接口

- GitHub  
<https://github.com/danielwiehl/edu-angular-change-detection/>
- Live Demo  
<https://danielwiehl.github.io/edu-angular-change-detection/>
- StackBlitz  
<https://stackblitz.com/github/danielwiehl/edu-angular-change-detection/>





nq-tw

# 使用纯管道组件

Pure Pipe



# 使用 Pure Pipe 组件

- 所有 Pipes 组件默认为 [Pure Pipes](#)
  - 唯有属性参考发生改变，才会重新执行 Pipes 转换
  - 考虑使用 [Immutable.js](#)
- 纯管道组件演示
  - <https://stackblitz.com/edit/angular-pure-pipes>



ng-tw

# 路由懒加载 + 预载策略

Router preloading strategy



# 不要将预设路由设定懒加载

// 错误演示

```
const routes: Routes = [  
  { path: '', redirectTo: '/dashboard', pathMatch: 'full' },  
  {  
    path: 'dashboard',  
    loadChildren: () => import('./xxx').then(mod => mod.ooo)  
  },  
  {  
    path: 'heroes',  
    loadChildren: () => import('./ooo').then(mod => mod.xxx)  
  }  
];
```



# 使用 ngx-quicklink 路由预载策略

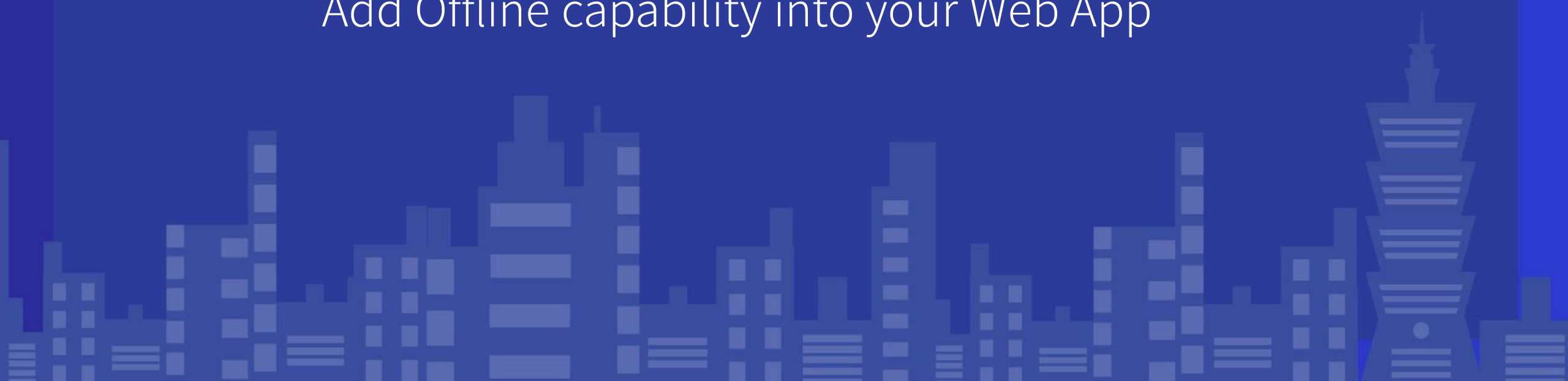
- 安装 [ngx-quicklink](#) 套件  
`npm install ngx-quicklink --save`
- 汇入 **QuicklinkModule** 模块 (AppModule)  
若汇入 (import) 共享模块需记得导出 (export)
- 套用路由预载策略  
`RouterModule.forRoot(routes, {  
 preloadingStrategy: QuicklinkStrategy  
}),`



ng-tw

# 使用 Service Worker

Add Offline capability into your Web App



说一个解决网络慢最好的方案

不上网！





# 替專案加入離線瀏覽的能力

- 安裝 PWA 功能
  - `ng add @angular/pwa`
- 建置 PWA 網站
  - `ng build --prod`
- 只有在 `ng build --prod` 的時候才會產生的檔案
  - `ngsw-worker.js` (Angular Service Worker 主程式)
  - `ngsw.json` (Angular Service Worker 執行時期設定檔)



```
G:\Projects\angular[master]>ng add @angular/pwa
Installing packages for tooling via npm.
Installed packages for tooling via npm.
CREATE ngsw-config.json (620 bytes)
CREATE src/manifest.webmanifest (1081 bytes)
CREATE src/assets/icons/icon-128x128.png (1253 bytes)
CREATE src/assets/icons/icon-144x144.png (1394 bytes)
CREATE src/assets/icons/icon-152x152.png (1427 bytes)
CREATE src/assets/icons/icon-192x192.png (1790 bytes)
CREATE src/assets/icons/icon-384x384.png (3557 bytes)
CREATE src/assets/icons/icon-512x512.png (5008 bytes)
CREATE src/assets/icons/icon-72x72.png (792 bytes)
CREATE src/assets/icons/icon-96x96.png (958 bytes)
UPDATE angular.json (3800 bytes)
UPDATE package.json (1597 bytes)
UPDATE src/app/app.module.ts (923 bytes)
UPDATE src/index.html (821 bytes)
✓ Packages installed successfully.
```



# 重要的配置文件

- ngsw-config.json
  - 主要用来设定 Angular Service Worker 的相关设定
  - 可手动加入可供脱机用的文件名 (Pattern)
- src/manifest.webmanifest
  - 应用程序 Manifest 定义档 (让网站可安装成应用程序)



nq-tw

# 使用虚拟滚动组件

Angular CDK Virtual Scrolling



# 虛擬捲動 (Virtual Scrolling)

- Scrolling | Angular Material

<https://material.angular.io/cdk/scrolling/overview>

- 匯入 ScrollingModule 模組

- PrimeNG 也內建 VirtualScroller ( 內建延遲載入、排序、篩選 )

<https://www.primefaces.org/primeng/#/virtualscroller>



ng-tw

# 效能预算

Performance Budgets



# 效能预算定义在 angular.json

angular.json ×

angular.json > {} projects > {} demo9 > {} architect

```
30      },
31      "configurations": {
32        "production": {
33          "fileReplacements": [ ...
38        ],
39        "budgets": [
40          {
41            "type": "initial",
42            "maximumWarning": "2mb",
43            "maximumError": "5mb"
44          },
45          {
46            "type": "anyComponentStyle",
47            "maximumWarning": "6kb",
48            "maximumError": "10kb"
49          }
50        ],
51        "optimization": true,
```





# 每次建置都可以检查结果與狀態

- 目测建置结果

```
WARNING in budgets, maximum exceeded for initial. Budget 100 kB was exceeded by 824 kB.
```

```
ERROR in budgets, maximum exceeded for initial. Budget 200 kB was exceeded by 724 kB.
```

```
G:\Projects\angular-realworld-example-app-qucklink [master]>echo %ERRORLEVEL%  
1
```

- 命令提示字符 (Command Prompt)

- %ERRORLEVEL% = 0

- PowerShell

- \$LASTEXITCODE = 0 (或判断 \$? 为 True 代表建置成功)



# 相关连结

- [Angular 性能检查清单](#)
- [配置文件大小预算](#)
- [ngx-quicklink](#)
- [EduChangeDetection](#) ([GitHub](#))
- [Angular Service Worker Step-By-Step Guide](#)
- [Precaching with the Angular service worker](#)



< THANK YOU >



2019 ngChina 开发者大会