



Web Development Fundamentals

使用 **Azure** 应用服务和
Visual Studio Code
创建和发布 **Web** 应用程序

Oct 2020

Microsoft Reactor | Ryan Chung

```
led by player to  
s.load_image("kg.png")  
  
[self]:  
    initialize Dog object and create Text o  
g, self).__init__(image = Dog.image  
    x = games.mouse.x  
    bottom = games.sc  
  
re = games.Text(value = 0, size = 24  
    top = 5, right = gam  
  
reen.add(self.score)  
1 = games.Text(value = 0, size = 24  
    top = 5, left = gam
```



Ryan Chung

Instructor / DevelopIntelligence
Founder / MobileDev.TW

@ryanchung403 on WeChat
Ryan@MobileDev.TW

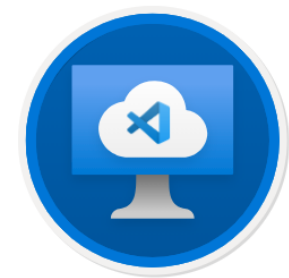




Reactor



developer.microsoft.com/reactor/
@MSFTReactor on Twitter



使用 Azure 应用服务和 Visual Studio Code 创建和发布 Web 应用程序

5 分钟 剩余 • 模块 • 已完成 7 个单元, 共 8 个

★★★★★ 4.8 (19)

对其进行评级

初级

开发人员

Azure

应用服务

Visual Studio Code

使用 Visual Studio Code 创建 Web 应用程序并将它发布到 Azure 应用服务。

学习目标

学完本模块后, 你将能够执行以下操作:

- 在 Visual Studio Code 中创建新的 Web 应用程序
- 将应用发布到 Azure 应用服务
- 在 Visual Studio Code 中更新应用并发布更新

继续 >

🔖 书签

⊕ 添加到集合

<https://aka.ms/PublishAppsAzureAppSvc-8>

流程

介绍

创建 Web 应用程序

发布到 Azure 应用服务

更新应用并发布更新

介绍

目标

1. 使用JavaScript/Node.js/Express.js构建一个显示有倒计时的网站
2. 发布到 Azure 应用服务，了解如何管理网站版本

知识

1. 熟悉 Web 应用概念
2. 初级编程知识
3. Visual Studio Code 和 Node.js 本地安装

Node.js & Express.js

Node.js

在 Web 应用中，可以使用 Node.js 运行服务器端代码。通过这种方法，你可以使用 JavaScript 编写服务器和客户端浏览器中 Web 应用中的所有代码。你无需学习其他语言

Express.js

使用少量自定义代码快速构建 Node.js Web 服务器。Express.js 包含一些函数，可以方便地拦截 HTTP 请求（例如 GET、POST 和 DELETE），然后将它们路由到用于制定响应的代码。它还在本地计算机上实现 HTTP 服务器，可以启动它以便测试代码。

练习：建立第一个Node.js程式

安装Node.js

- 选择稳定版 (Recommend For Most Users)
 - Windows Installer : 下载msi档
 - macOS Installer : 下载pkg档
- 如何确认安装?
 - 打开终端机(命令提示字元) , 输入 : `node -v`

```
[ryan (master) ~ $ node -v  
v10.13.0
```

建立专案资料夹

- 打开VS Code，档案 -> 开启资料夹
- 新增资料夹，命名为HelloNode
- 新增档案 index.js
- 填入以下内容

```
var express = require('express');  
var app = express();  
app.get('/', (req, res) => res.send('Hello World!'));  
app.listen(3000);
```

专案初始化 & 安装需要的套件

- 在VS Code中，按下 Ctrl + ~ 开启终端机
 - 在终端机输入 `npm init`
 - 按下多次 `enter`，使用预设值建立 `package.json`
- 针对特定档案中的需求来安装套件
 - 在终端机输入 `npm install express`


增加Debug与Start Scripts

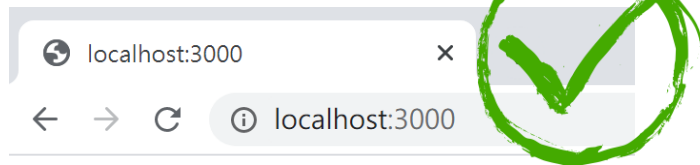
- 在 package.json 新增

```
"version": "1.0.0",  
"description": "",  
"main": "index.js",  
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "start": "node index.js",  
  "debug": "node --inspect index.js"  
},  
"author": "",  
"license": "ISC",
```

- debug : 在地测试、start : 伺服器上线

执行并开启除错模式

- 点击VS Code中左边选单中的第四个(执行) 
- 建立launch.json档案... 选Node.js
- 再一次，新增设定，选择「{} Node.js 透过 NPM 启动」
- 按下Play按钮，VS Code上方出现侦错工具列
- 打开浏览器，输入网址
- <http://localhost:3000/>



Hello World!

练习：使用生成器建立Node.js程式

建立专案资料夹

- 打开VS Code，档案 -> 开启资料夹
- 新增资料夹，命名为HelloNode2
- 在VS Code中，按下 Ctrl + ~ 开启终端机
 - 在终端机输入 `npx express-generator --view=pug -c --css`

选择pug作为View Engine

CSS使用一般CSS语法

安装需要的套件

- 针对特定档案中的需求来安装套件
 - 安装套件侦测软体 `npm install npm-install-all -g`
 - 在终端机输入 `npm-install-all` 一次安装所有套件
 - 若安装失败，可依系统提示讯息单一安装套件
 - `npm install package-name`

增加Debug与Start Scripts

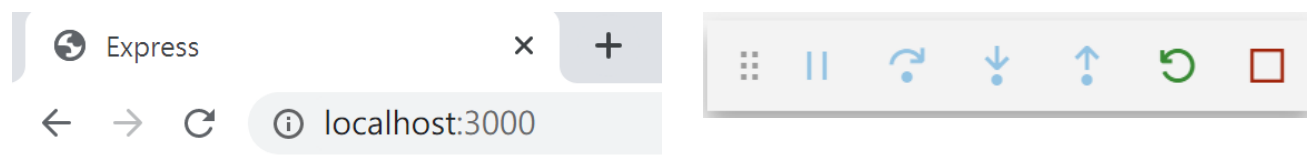
- 在 package.json 新增

```
{  
  "name": "hellonode2",  
  "version": "0.0.0",  
  "private": true,  
  "scripts": {  
    "start": "node ./bin/www",  
    "debug": "node ./bin/www --inspect"  
  },  
}
```

- debug : 在地测试、start : 伺服器上线

执行并开启除错模式

- 点击VS Code中左边选单中的第四个(执行) 
- 建立launch.json档案... 选Node.js
- 再一次，新增设定，选择「{} Node.js 透过 NPM 启动」
- 按下Play按钮，VS Code上方出现侦错工具列
- 打开浏览器，输入网址
 - <http://localhost:3000/>

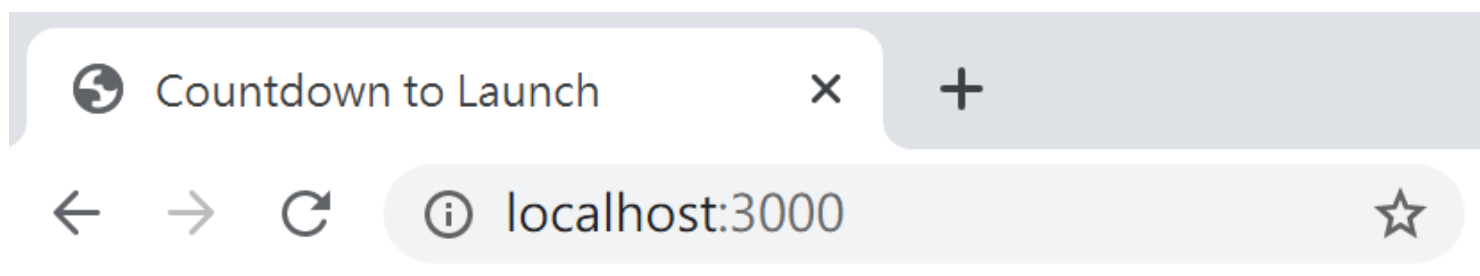


Express

Welcome to Express



练习：延续前例，加上倒数计时秒数显示



Countdown to Launch

Only a few days left until our new game launch!

Countdown: 8117541118 milliseconds

修改 index.js

```
var express = require('express');  
var router = express.Router();
```

```
var launchTime = new Date("12/29/2020 10:00").getTime();
```

```
/* GET home page. */  
router.get('/', function(req, res, next) {  
  //res.render('index', { title: 'Express' });  
  var currentTime = new Date().getTime();  
  var numberOfMilliseconds = parseInt(launchTime-currentTime);  
  res.render('index',{  
    title:'Countdown to Launch',  
    countDown:numberOfMilliseconds  
  });  
});
```

```
module.exports = router;
```


修改 views/index.pug

```
extends layout
```

```
block content
```

```
  h1= title
```

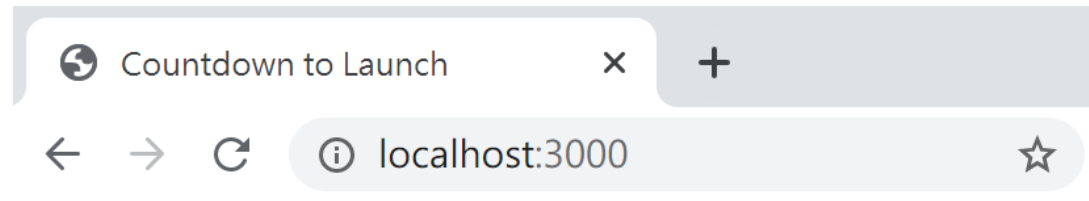
```
  //p Welcome to #{title}
```

```
  p Only a few days left until our new game launch!
```

```
  p Countdown: #{countDown} milliseconds
```

启动侦错，观察网页

- 点击VS Code中左边选单中的第四个(执行)
- 打开浏览器，输入网址
- <http://localhost:3000/>



Countdown to Launch

Only a few days left until our new game launch!

Countdown: 8117541118 milliseconds




练习：部署至Azure

安装Azure App Service

- 左边延伸模组->上方搜寻Azure App Service
- 点击安装





Azure App Service

ms-azuretools.vscode-azureappservice

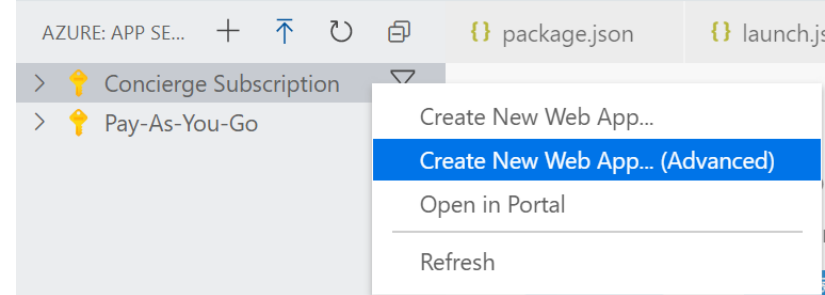
Microsoft | 448,090 | ★★★★★ | 儲存庫 | 授權 | v0.19.0

An Azure App Service management extension for Visual Studio Code.

正在安裝

預覽

进入Azure App Service



- 在线上教学文件中点击激活沙盒，登入Azure帐号
- 在VS Code中左边点击Azure图示 
- 登入Azure，会看见除了原本的Pay-As-You-Go之外，多了一个Concierge Subscription，选这一个
- 右键->Create New Web App...(Advanced)
- 为专案取一个名子，例如launchcountdown-XXX
- 选择该资源组、Linux、Node 12 LTS

<https://docs.microsoft.com/zh-cn/learn/modules/create-publish-webapp-app-service-vs-code/5-exercise-publish-app-azure-app-service-vs-code?pivots=nodeexpress>

进入 Azure App Service



- 新建应用服务计划，取一个名字如LaunchCountPlan
- F1 Free、跳过新建Application Insights资源
- 选择 Central US
- 开始创建应用服务计划和Web应用
- 稍作等待后，右下方点击Deploy
- 开启浏览器进行测试
- 回到线上教学文件，点击检查工作

干得漂亮!

你完成了该任务，获得了相应的经验点数。



<https://docs.microsoft.com/zh-cn/learn/modules/create-publish-webapp-app-service-vs-code/5-exercise-publish-app-azure-app-service-vs-code?pivots=nodeexpress>

练习：在 Visual Studio Code 中更新应用并发布更新

更新 index.js

```
var express = require('express');
var router = express.Router();

var launchTime = new Date("12/29/2020 10:00").getTime();

/* GET home page. */
router.get('/', function(req, res, next) {
  //res.render('index', { title: 'Express' });
  var currentTime = new Date().getTime();
  var numberOfMilliseconds = parseInt(launchTime-currentTime);
  var numberOfDays = parseInt(numberOfMilliseconds/(24 * 3600 * 1000));
  res.render('index',{
    title:'Countdown to Launch',
    countDown:numberOfDays
  });
});

module.exports = router;
```

更新 index.pug

```
extends layout
```

```
block content
```

```
h1= title
```

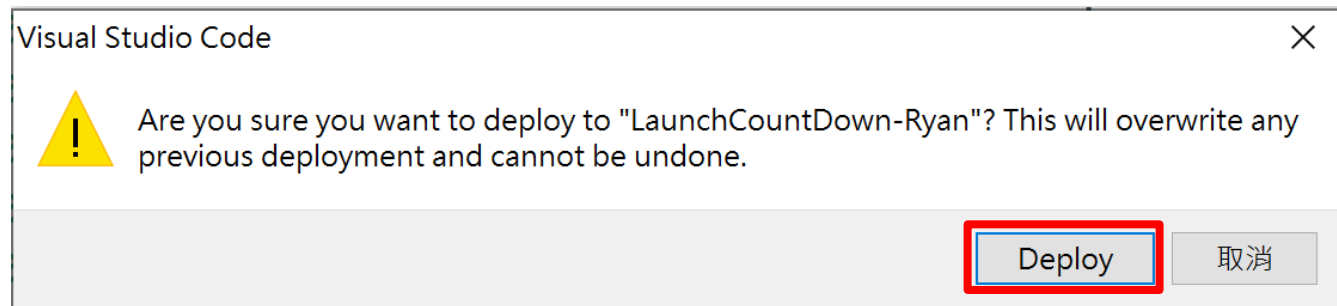
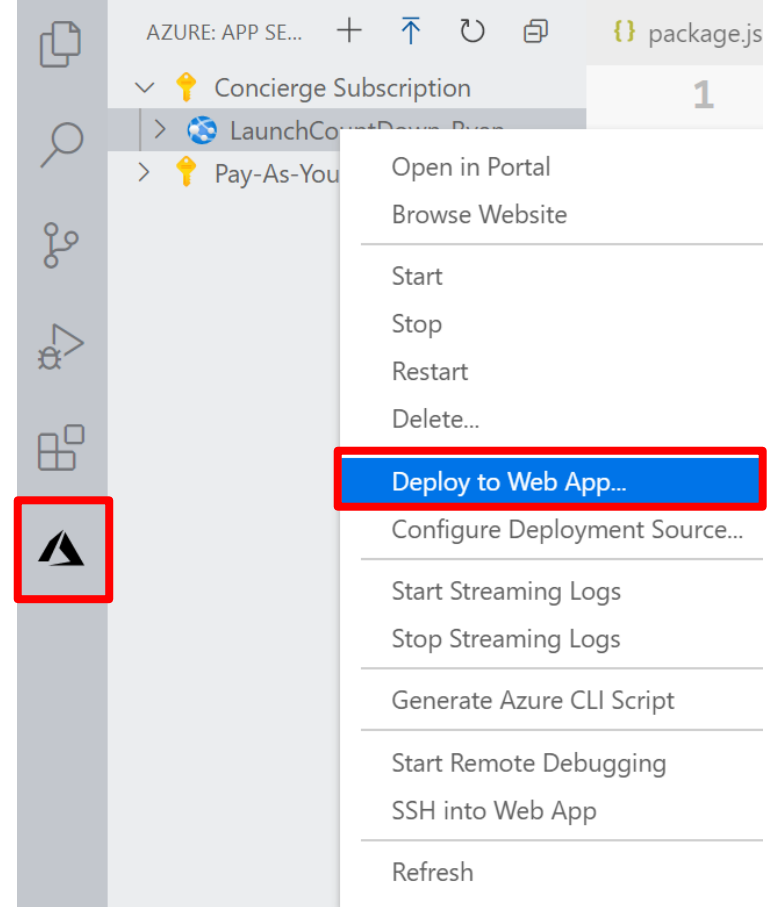
```
//p Welcome to #{title}
```

```
p Only a few days left until our new game launch!
```

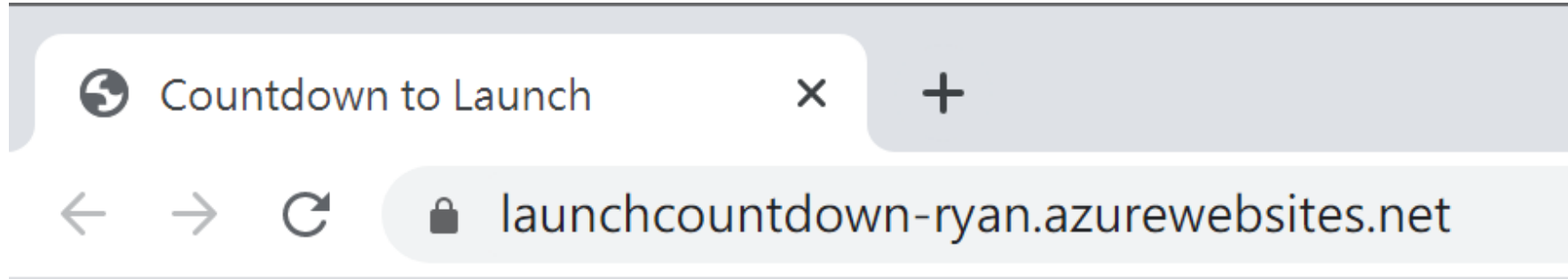
```
p Countdown: #{countDown} days
```

重新部署

- 在VS Code中左边点击Azure图示
- Concierge Subscription 展开
- LaunchCountDown专案，右键
- Deploy to Web App...
- 确认覆写，按下Deploy



打开浏览器，确认网站内容是否已更新



Countdown to Launch

Only a few days left until our new game launch!

Countdown: 93 days

延伸学习资源

- [Visual Studio Code 中的 Node.js 教程](#)
- [应用服务文档](#)
- [Azure 应用服务计划概述](#)
- [使用 Visual Studio Code 部署到 Azure 应用服务](#)
- [适用于 Node.js 的 Azure 远程调试](#)
- [在 Azure 应用服务中设置过渡环境](#)



Reactor



developer.microsoft.com/reactor/
[@MSFTReactor](#) on Twitter

议程结束 感谢聆听



请记得填写课程回馈问卷
<https://aka.ms/Reactor/Survey>

© 2019 Microsoft Corporation. All rights reserved. The text in this document is available under the Creative Commons Attribution 3.0 License, additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are not included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.