

---

建立一個名為 myapp 的目錄，

```
$ mkdir myapp
```

切換至該目錄，

```
$ cd myapp
```

並執行 npm init。

```
$ npm init
  - entry point: (index.js) myapp.js
```

將 express 安裝成一個相依關係。

```
$ npm install express --save
```

---

在 myapp 目錄中，建立名為 app.js 的檔案，並新增下列程式碼：

```
var express = require('express');
var app = express();
```

```
app.get('/', function (req, res) {
  res.send('Hello World!');
});
```

```
app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

啟動 app.js

```
$ node app.js
```

然後在瀏覽器中載入 <http://localhost:3000/>，以查看輸出。

---

使用應用程式產生器工具 express，快速建立應用程式架構。

使用下列指令來安裝 express：

```
$ sudo npm install express-generator -g
```

使用 -h 選項來顯示指令選項：

```
$ express -h
```

---

舉例來說，以下是在現行工作目錄中建立一個名為 myapp 的 Express 應用程式：

```
$ express --view=pug myapp
```

然後安裝相依項目：

```
$ cd myapp
$ npm install
```

在 MacOS 或 Linux 中，使用下列指令來執行應用程式：

```
$ DEBUG=myapp:* npm start
```

// - 執行後，欲離開 按按鍵 control + c (不是command鍵)

```
$ DEBUG=express-mysqldtest:* npm start
```

在 Windows 中，使用下列指令：

```
> set DEBUG=myapp:* & npm start
```

然後在瀏覽器中載入 <http://localhost:3000/>，以存取應用程式。

產生的應用程式具有如下的目錄結構：

```
.
├── app.js
├── bin
│   └── www
├── package.json
├── public
│   ├── images
│   ├── javascripts
│   └── stylesheets
│       └── style.css
├── routes
│   ├── index.js
│   └── users.js
└── views
    ├── error.pug
    ├── index.pug
    └── layout.pug
```

7 directories, 9 files

-----  
DEMO package.json

```
{
  "name": "express-mysqldtest",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "node ./bin/www"
  },
  "dependencies": {
    "body-parser": "~1.18.2",
    "cookie-parser": "~1.4.3",
    "debug": "~2.6.9",
    "ejs": "~2.5.7",
    "express": "~4.15.5",
    "morgan": "~1.9.0",
    "serve-favicon": "~2.4.5",
    "jade": "~1.11.0",
    "mongodb": "~2.2.33",
    "mongoose": "~4.13.0"
  }
}
```

