

編寫 API ，
建置、部署與監控節點
輕鬆掌握入門技巧



馬上開始吧！

但請先…

1. 註冊

請先在 strongloop.com. 上註冊一個帳號。

2. 安裝 Node.js

請先安裝 [Node.js](https://nodejs.org/)。

3. 安裝 StrongLoop

您可執行以下指令來安裝 StrongLoop：

```
$ npm install -g  
strongloop
```

遇到問題了嗎？請查看 [安裝疑難排解手冊](#)。

您正處於哪一個 開發階段？

API 編寫

請安裝 LoopBack 架構、設計一些 API，然後利用 StrongLoop Arc 圖形式套件來探索與移轉資料模型。還沒建立任何專案嗎？請使用內建的應用程式範例來測試各項功能。



建立 LoopBack 應用程式

現在發出以下指令以使用 LoopBack 架構建立應用程式：

```
$ slc loopback
```

為應用程式與專案目錄命名

當系統出現提示時，請為應用程式以及儲存「arc- getting-started」專案的目錄命名。

執行階段

預備好進入生產階段了嗎？您的應用程式不論是利用 Express、LoopBack 或其他 Node.js 架構建置而成，都可以與我們的執行階段工具相容。還沒建立任何專案嗎？請使用內建的應用程式範例來測試各項功能。



啟動 StrongLoop Arc

```
$ cd your-app-directory
```

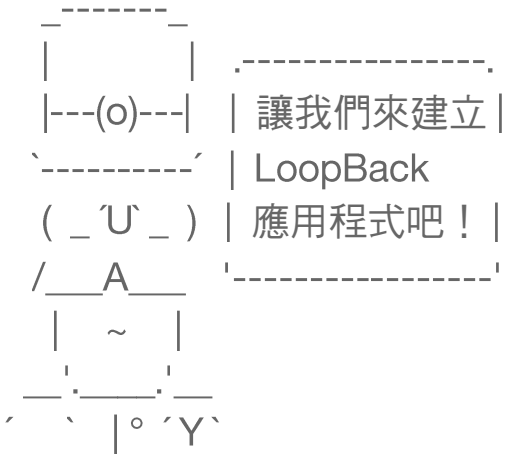
這個目錄內含主要的 JavaScript 檔 server.js。現在輸入：

```
$ slc arc
```

您現在即可利用 StrongLoop Arc 來建置與部署、設定設定檔、監控與管理程序！

建置與部署

不需使用 Sript 即可利用 StrongLoop 的建置功能來安裝相依檔案。



[?] arc-getting-started 應用程式的名稱是什麼？[?] 輸入要儲存 arc-getting-started 專案的目錄名稱

執行 StrongLoop Arc

應用程式已經建立完成，現在請啟動 StrongLoop Arc。但請先將目錄名稱變更為“arc-getting-started”，然後如下執行 Arc：

```
$ cd arc-getting-started
```

```
$ slc arc
```

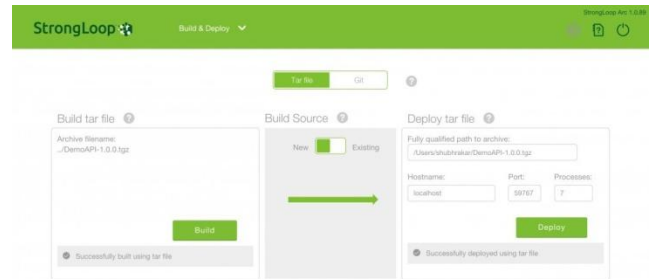
Arc 於此處執行：

```
http://localhost:51230
```

登入 Arc

如果還沒註冊，請先造訪 strongloop.com/register/ 進行註冊。

您可將可發佈的保存檔與相依檔案封包在一起，產生 tar 檔案供 npm pack 與 npm install 使用。



[深入瞭解](#)或[馬上開始](#)。

管理與集中程序

利用叢集可水平擴充多個程序，而使用 Nginx 搭配狀態共用，可水平擴充多部伺服器。

Strong PM	Port	App Status	Count	PIDs
localhost	62506	Active	6	22253 22252 22254 22255
api-server.aws.com	62567	Active	2	22572 22571

[深入瞭解](#)或[馬上開始](#)。

設定檔

用來設定 CPU 與記憶體以及資料堆傾出與交易追蹤的工具。

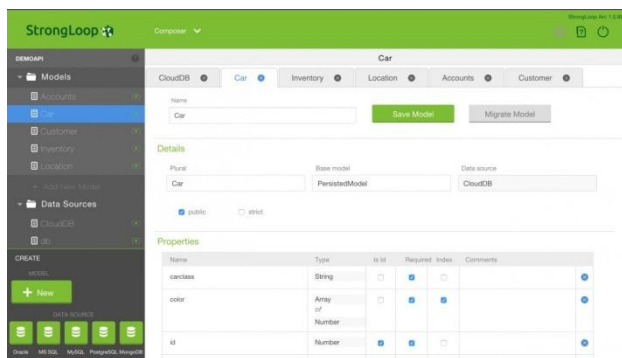
啟動 API 編寫程式

登入之後即可看見 Arc 啟動程式。
按一下「編寫程式」圖示。



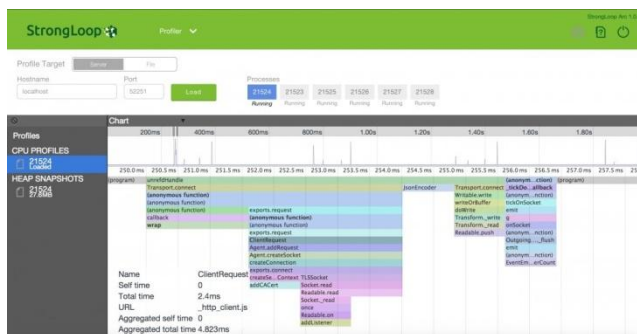
建立與探索模型

完成了！您現在可開始建立與探索模型。
此處提供使用 StrongLoop Arc 編寫 API 的逐步說明。



下個步驟是什麼？

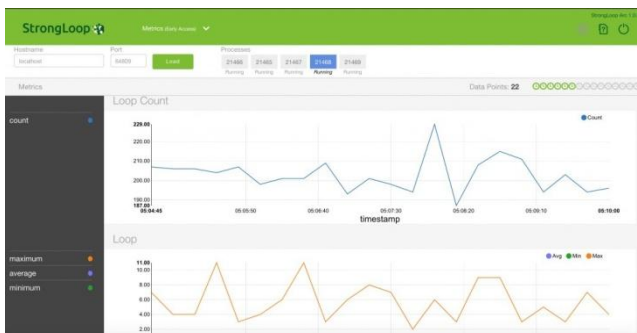
使用 StrongLoop API Explorer 導覽並
測試 API。[深入瞭解](#)...



[深入瞭解](#)或[馬上開始](#)。

監控

可採用託管形式或內部部署形式來使用
APM 的各項功能，如事件迴圈監控、V8
記憶體回收、回應時間、降低端點速度
以及 CPU 與記憶體使用狀況統計資料。



[深入瞭解](#)或[馬上開始](#)。

偵錯

使用 StrongLoop Arc 外部的「節點偵測
程式」在 Chrome 瀏覽器中偵錯程式碼

[深入瞭解](#)或[馬上開始](#)。

Cars		Show/Hide	List Operations	Expand Operations	Raw
POST	/Cars	Create a new instance of the model and persist it into the data source			
PUT	/Cars	Update an existing model instance or insert a new one into the data source			
GET	/Cars	Find all instances of the model matched by filter from the data source			
GET	/Cars/{id}/exists	Check whether a model instance exists in the data source			
HEAD	/Cars/{id}	Check whether a model instance exists in the data source			
GET	/Cars/{id}	Find a model instance by id from the data source			
DELETE	/Cars/{id}	Delete a model instance by id from the data source			
PUT	/Cars/{id}	Update attributes for a model instance and persist it into the data source			
GET	/Cars/findOne	Find first instance of the model matched by filter from the data source			
POST	/Cars/update	Update instances of the model matched by where from the data source			
GET	/Cars/count	Count instances of the model matched by where from the data source			
Customers		Show/Hide	List Operations	Expand Operations	Raw
Inventory		Show/Hide	List Operations	Expand Operations	Raw
Locations		Show/Hide	List Operations	Expand Operations	Raw
notes		Show/Hide	List Operations	Expand Operations	Raw

請致電 0800-016-888 按 1，將有專人為您服務。