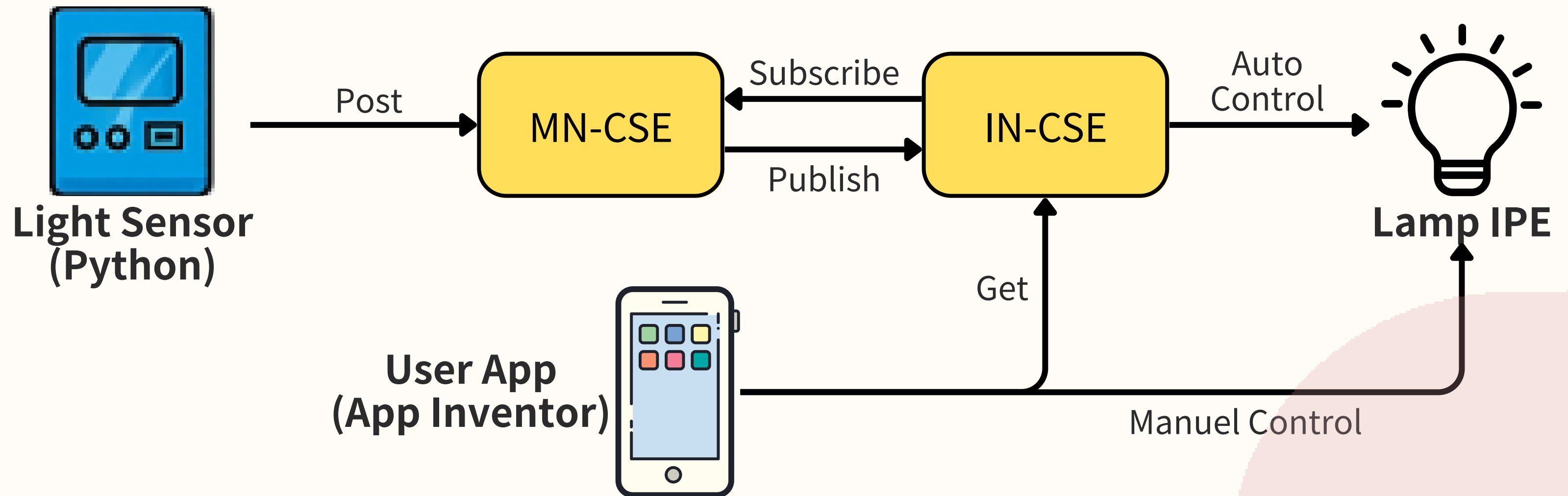


智慧電燈系統

王郁誠

Architecture



APP Inventor

- 分為 Auto、Manuel mode。使用者可以隨時切換模式。
- Auto mode：使用者無法自行控制電燈開關，系統會根據偵測到的光強度來自動控制電燈開關。
- Manuel mode：使用者可以自行控制電燈開關。

Smart Lamp APP

Current mode : auto

Auto Mode

Manuel Mode



Current light val : 70 lux

On

Off

Auto mode

Smart Lamp APP

Current mode : manuel

Auto Mode

Manuel Mode



Current light val : 60 lux

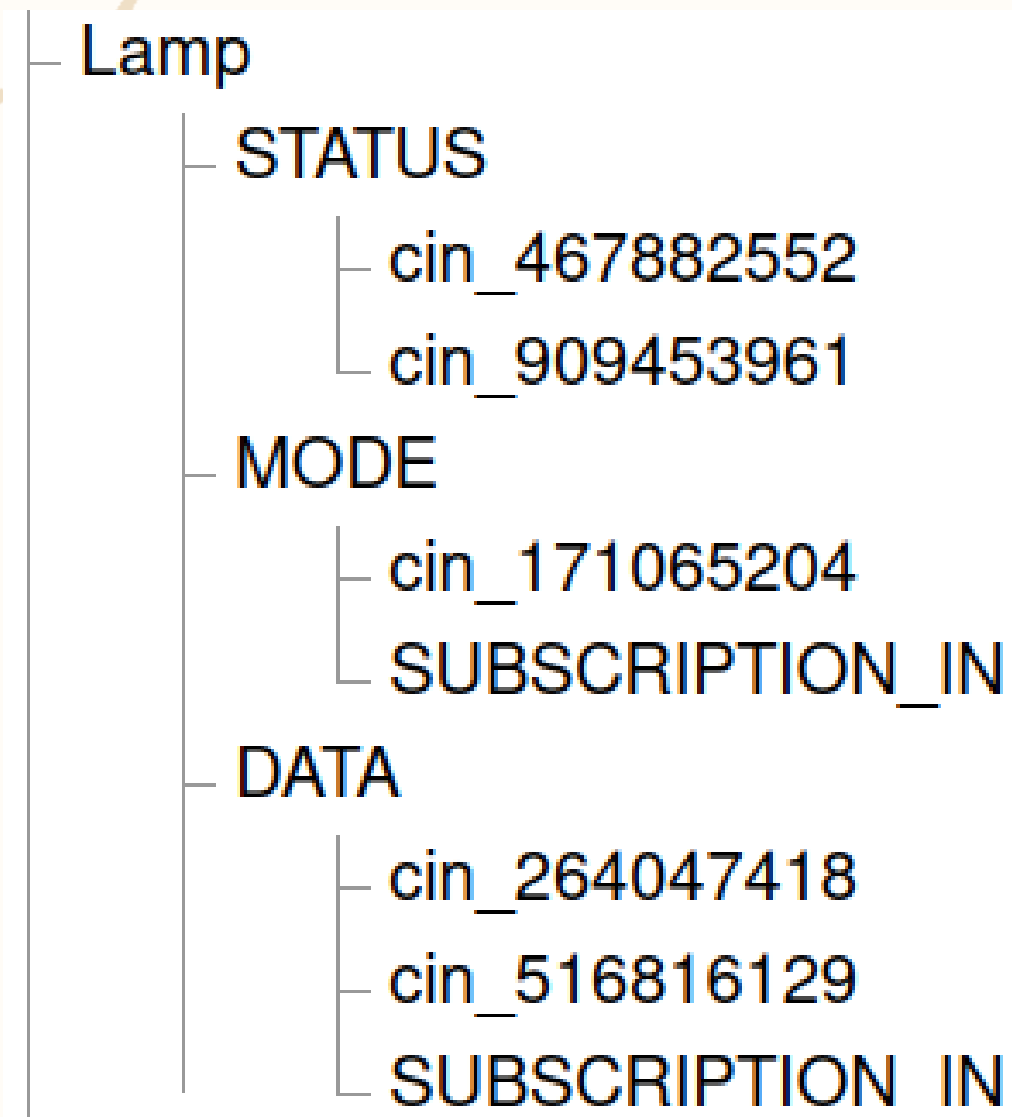
On

Off

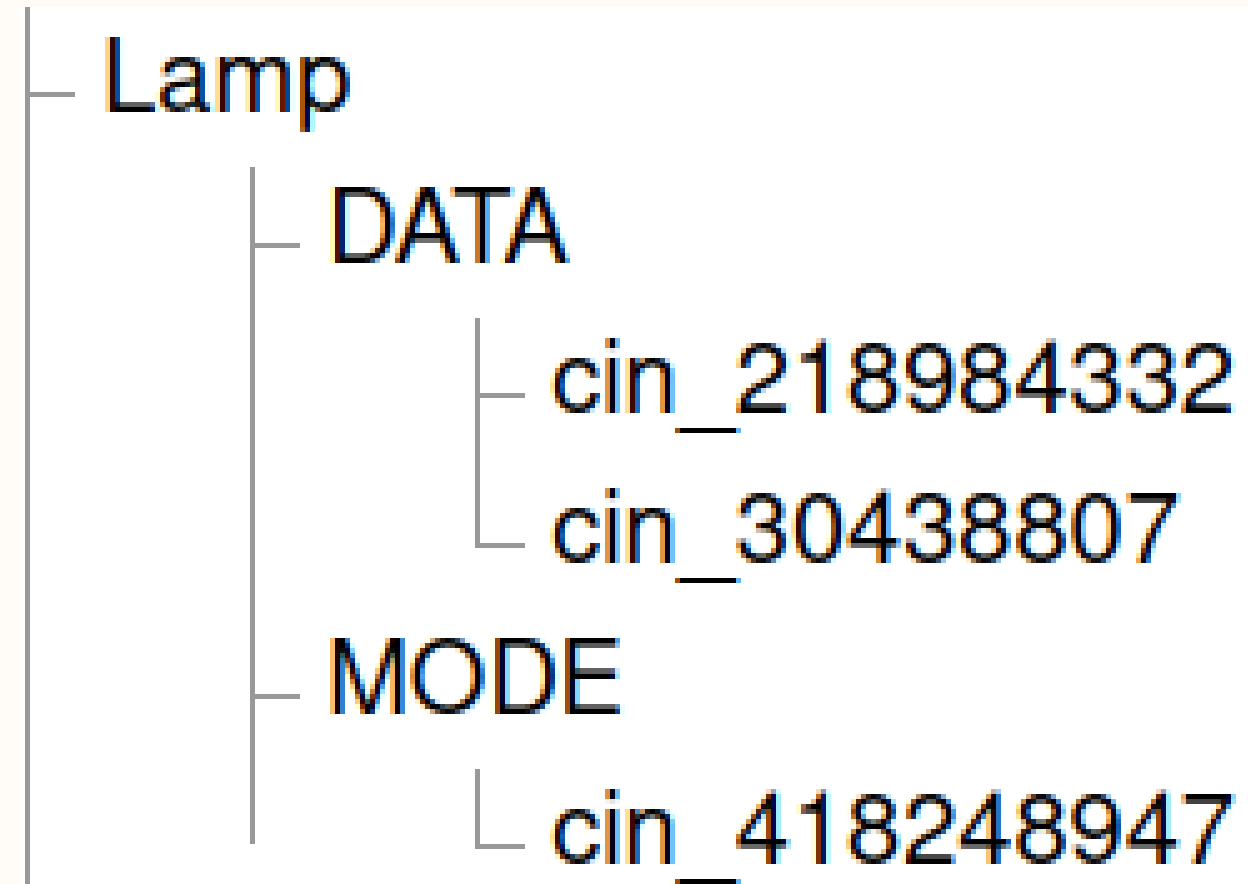
Manuel mode

Resource Tree

- MN的Lamp有DATA/STATUS/MODE 三個 container。
- IN的Lamp僅有DATA/MODE 兩個container，皆為訂閱MN而來。



MN-CSE



IN-CSE

Resource Tree

- Container instance內容
 - STATUS：
燈泡的狀態，e.g. on、off
 - MODE：
燈泡的模式，e.g. manuel、auto
 - DATA：
光感測器收到的光強度，e.g. 88 lux

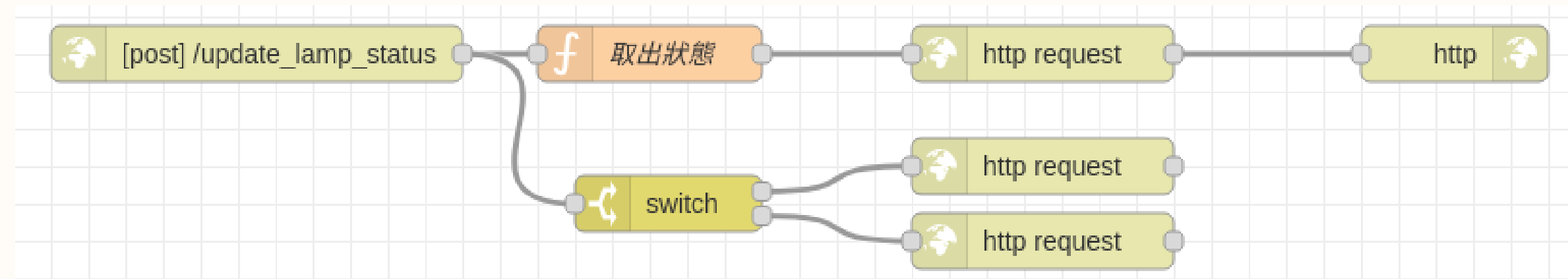
Attribute	Value
status	on

Attribute	Value
mode	manuel

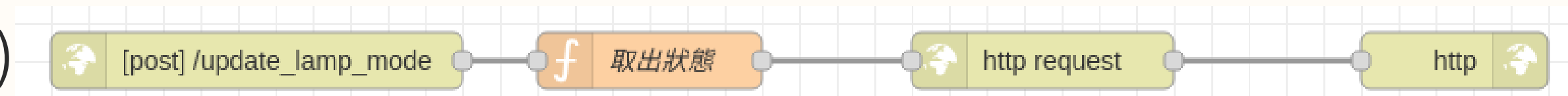
Attribute	Value
light	88

MN-CSE 更新流程

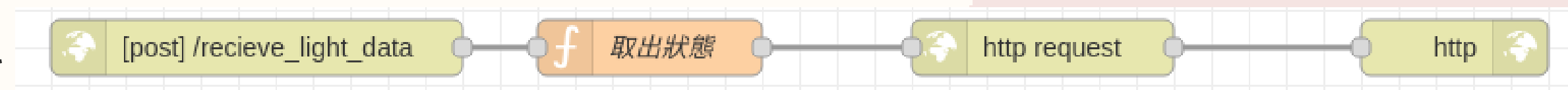
- 手動模式中，更新 Lamp 狀態 (on ,off) ，並在IPE上反映。



- 更新 Lamp 模式(manuel, auto)

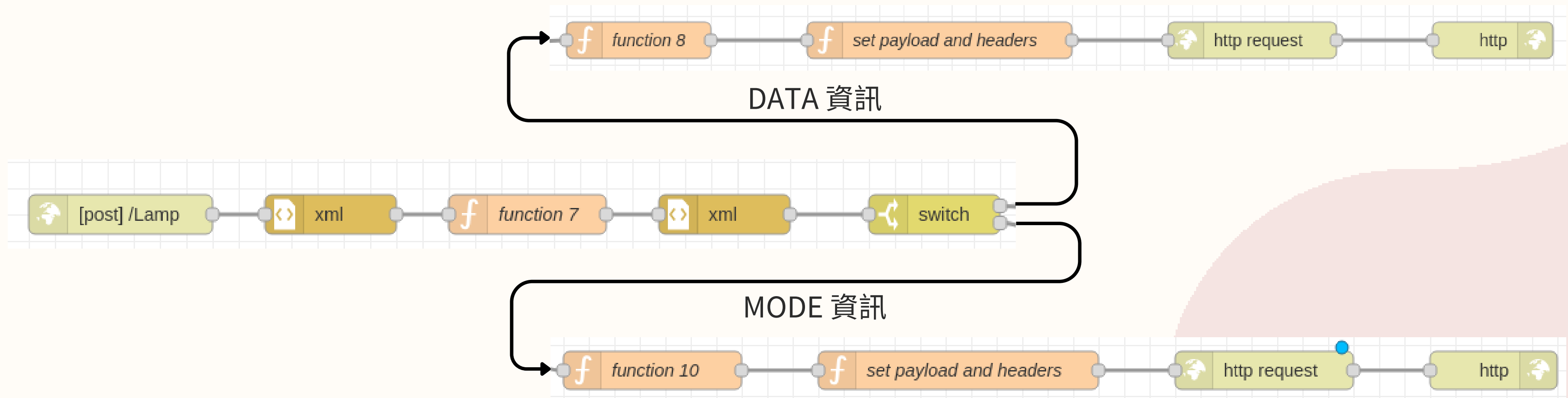


- 接收 Python 模擬的光強度資料



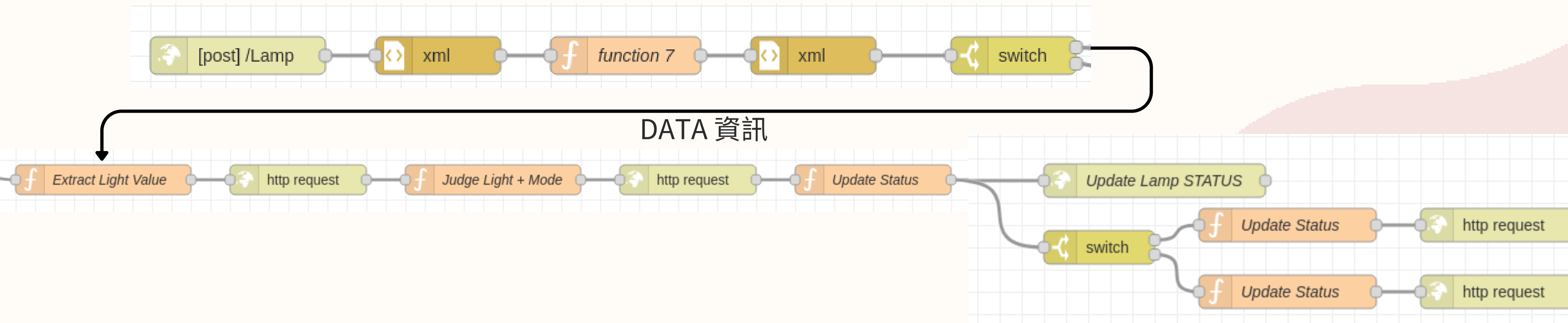
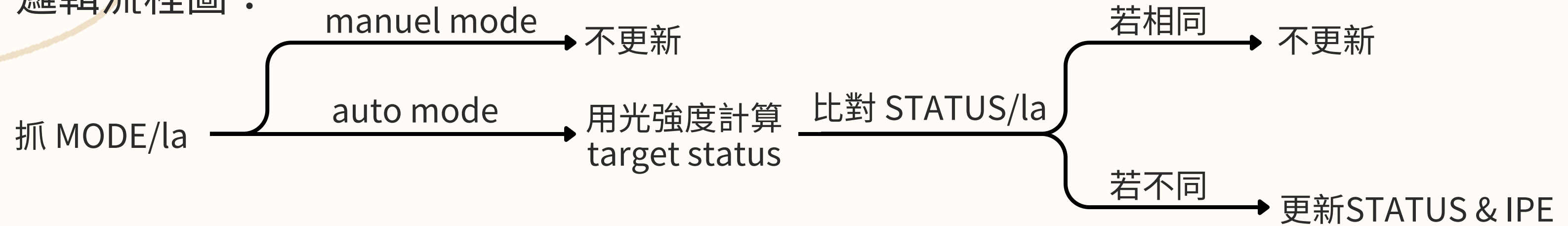
IN-CSE 更新流程

- IN 訂閱 MN 的 Lamp/DATA & Lamp/MODE。
- 當 MN 有新資訊時，會進 nodered 的 /Lamp處理，/Lamp會分辨是 DATA 還是 MODE 更新，並將這些資訊推到 IN 那邊。



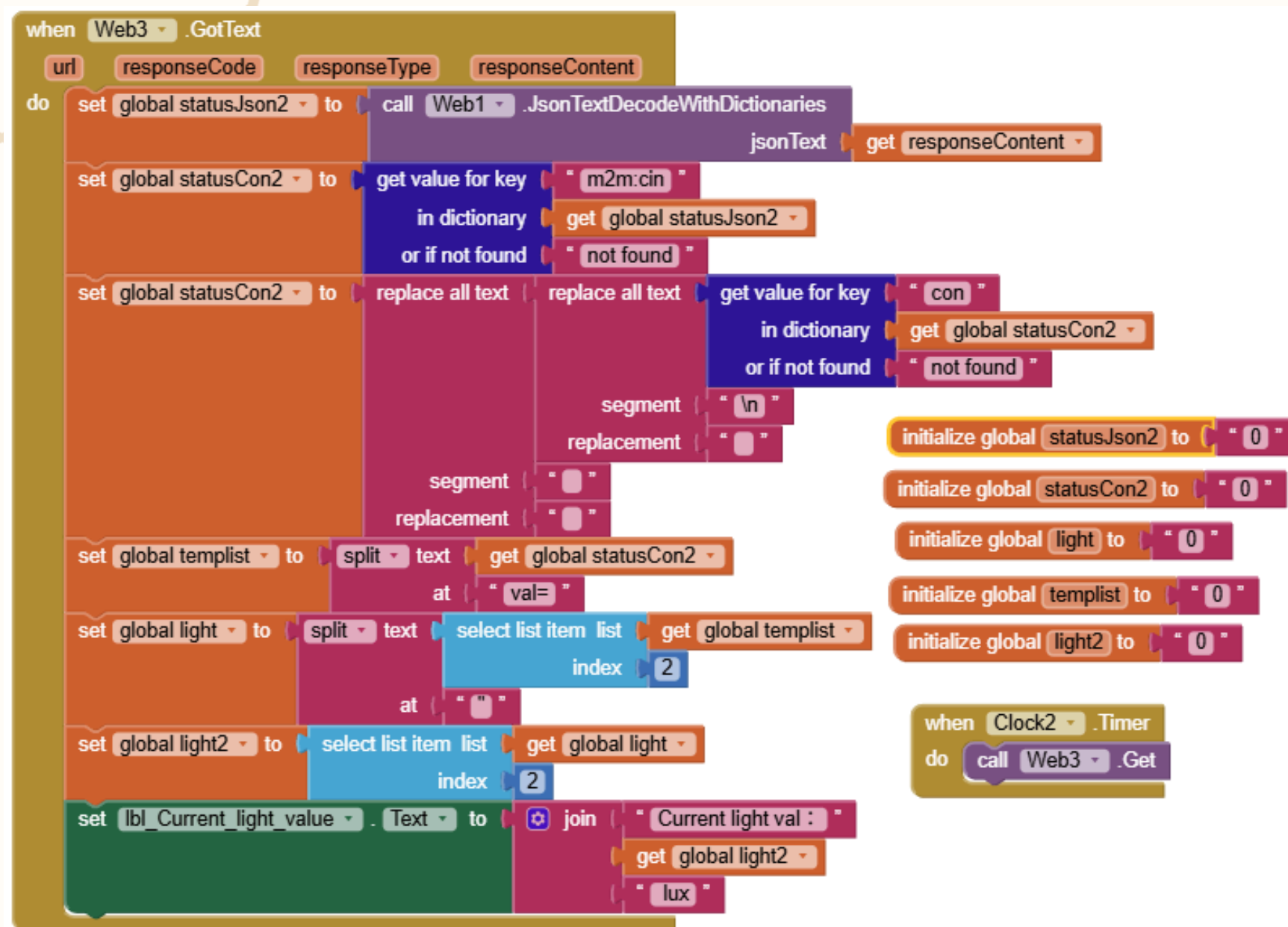
Auto Mode 判斷部分

- 同樣在 nodered 的 /Lamp 處理。
- 若發現 MN 收到 DATA 資料，除了更新 IN 外，進一步判斷是否要更新燈泡狀態。
- 邏輯流程圖：

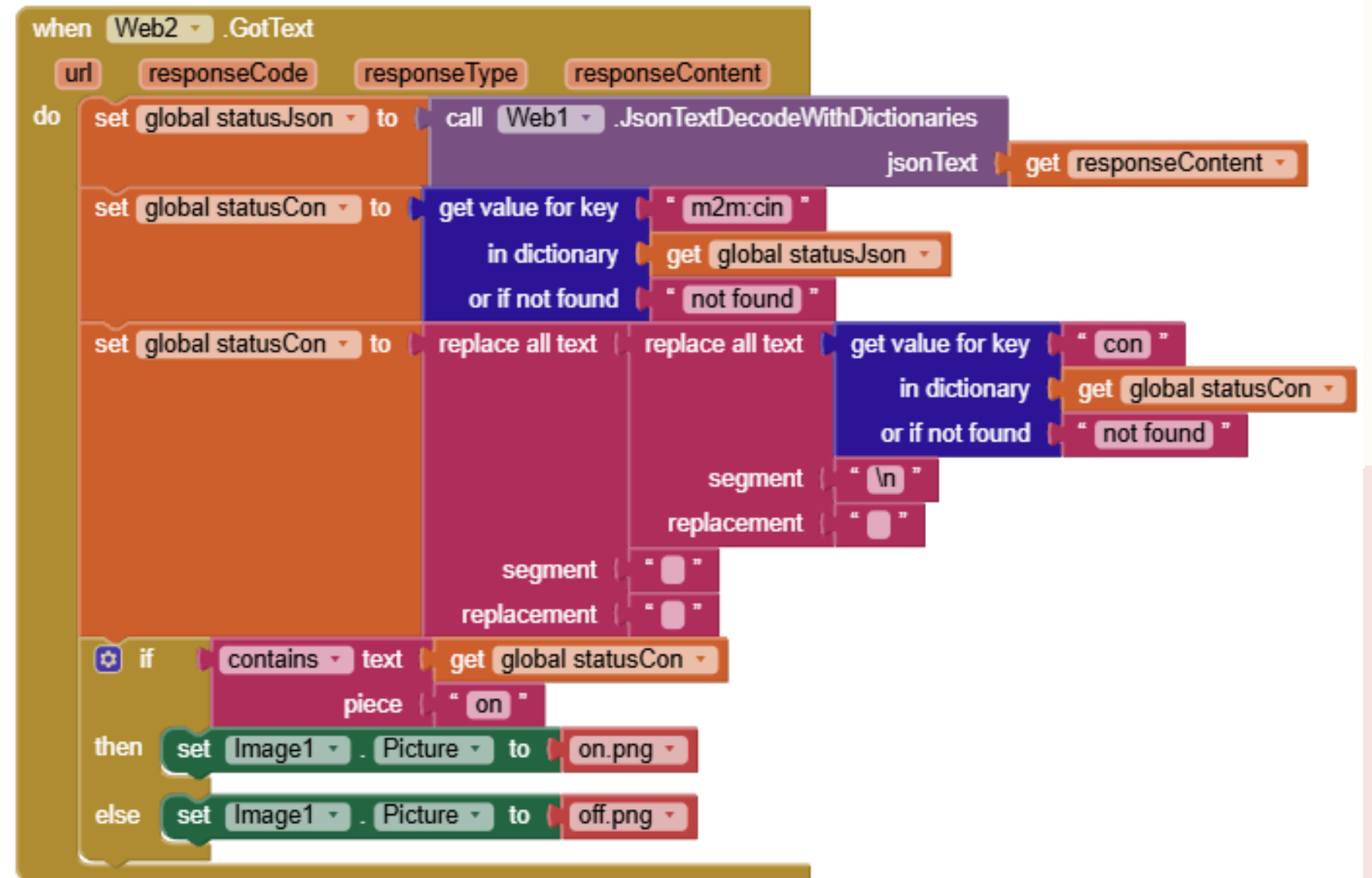


App Inventor 抓資料流程

- 在Auto mode 中，APP inventor 每秒去 mn 抓一次資料 (DATA、STATUS)



抓 DATA



抓 STATUS