API Routes 行為:進入店家頁面 API: GET (/store) Middle: CheckLogin API Routes StoreController 行為:新增店家 API: GET (/storeAdd) 1. 資料庫查詢所有店家 Middle: CheckLogin 2. 將店家放進select List裡 3. render store.ejs StoreController I. render storeAdd.ejs 店家頁面 (store.ejs) **API** Routes **API** Routes **API** Routes 行為:選擇店家並確認 | 行為:回到上一頁 StoreController API: POST (/storeAssign) API: GET (/store) (1)storeName | Middle: CheckLogin | Middle: CheckLogin StoreController StoreController 1. 資料庫查詢所有店家 1. redis查詢該Manager目前暫存的編號 2. 將店家放進select List裡 2. 將編號 & storeName 顯示在頁面上 3. render store.ejs 3. render storeAssign.ejs 分配頁面 (storeAssign.ejs) **API** Routes | 行為:按下分配完畢 API: GET (/storeAssignConfirm) | Middle: CheckLogin StoreController **API** Routes 1. redis → 查詢該Manager目前暫存的編號 |行為:回到首頁 2. Mysql → 將暫存的編號放進Store API: GET (/home) 3. 將分配成功編號 & storeName 顯示在頁面上 | Middle: CheckLogin | 4. render storeAssignResult.ejs HomeController 1. render home.ejs 結果頁面 (storeAssignResult.ejs)

新增店家頁面 (storeAdd.ejs)

API Routes

行為:新增店家

API: POST (/storeAdd)

(1)storeName

(2)storeAddress

|Middle: CheckLogin

StoreController

- 1. 檢查有無已存在店家名稱
 - a. 檢查Mysql
- 2. 將店家存入Mysql
- 3. redirect /storeAdd (GET)

API Routes

行為:掃描編號

API: POST (/storeObjectAdd)

(1)cupld

|Middle: CheckLogin

行為:返回店家頁面

API: GET (/store)

| Middle: CheckLogin

1. 資料庫查詢所有店家

3. render store.ejs

2. 將店家放進select List裡

StoreController

- 1. 檢查有無已存在編號
 - a. 檢查redis Manager
 - b. 檢查Mysql Store
- 2. 將編號存進該Manager redis的暫存欄位
- 3. redirect /storeAssign