**第八章 資料庫設計**

**8-1 資料庫關聯表**

**8-2 表格及其 Meta data**

表 8-2-1資料表描述：會員

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 資料表名稱: member | | | | | |
| 欄位名稱 | 中文名稱 | 資料類型 | 長度 | 允許空值 | 備註 |
| memberphone | 會員手機號碼 | varchar | 20 | 否 | PK |
| membername | 會員姓名 | varchar | 50 | 否 |  |
| gender | 性別 | varchar | 1 | 是 | 男(M)，女(F) |
| birthday | 出生日期 | date | 0 | 是 |  |
| points | 剩餘點數 | int4 | 32 | 是 |  |
| creationdate | 建立日期 | date | 0 | 是 |  |
| password | 密碼 | varchar | 20 | 是 |  |

表 8-2-2資料表描述：員工

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 資料表名稱：staff | | | | | |
| 欄位名稱 | 中文名稱 | 資料類型 | 長度 | 允許空值 | 備註 |
| staffphone | 員工電話 | varchar | 20 | 否 | PK |
| username | 員工姓名 | varchar | 30 | 否 |  |
| password | 密碼 | varchar | 20 | 否 |  |
| nickname | 員工綽號 | varchar | 20 | 是 |  |

表 8-2-3資料表描述：店家資訊

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 資料表名稱：storeinformation | | | | | |
| 欄位名稱 | 中文名稱 | 資料類型 | 長度 | 允許空值 | 備註 |
| storeID | 店家編號 | int4 | 32 | 否 | PK（自動遞增） |
| storename | 店家名稱 | varchar | 255 | 否 |  |
| storeaddress | 店家地址 | varchar | 50 | 否 |  |
| phoneno | 店家電話 | varchar | 30 | 否 |  |
| vacanttable | 空桌數 | int4 | 32 | 否 |  |
| businesshours | 營業時間 | varchar | 30 | 否 |  |
| wifi | 提空Wi-Fi | varchar | 1 | 否 |  |
| socket | 提供插座 | varchar | 1 | 否 |  |
| providemeals | 提供餐點 | varchar | 1 | 否 |  |
| outsidefood | 攜帶外食 | varchar | 1 | 否 |  |
| chargingstandards | 收費標準 | varchar | 255 | 否 |  |
| atime | 1小時 | int4 | 32 | 否 |  |
| apoint | 1小時點數 | int4 | 32 | 否 |  |
| lessatime | 小於1小時 | int4 | 32 | 否 |  |
| addapoint | 額外點數 | int4 | 32 | 否 |  |

表 8-2-4資料表描述：菜單

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 資料表名稱：food | | | | | |
| 欄位名稱 | 中文名稱 | 資料類型 | 長度 | 允許空值 | 備註 |
| foodid | 餐點編號 | int4 | 32 | 否 | PK（自動遞增） |
| foodname | 餐點名稱 | varchar | 255 | 否 |  |
| foodpoint | 餐點價格 | int4 | 32 | 否 |  |
| foodimg | 餐點圖片 | varchar | 255 | 是 |  |
| itemname | 餐點類別 | varchar | 255 | 否 |  |

表 8-2-5資料表描述：餐點類別

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 資料表名稱：item | | | | | |
| 欄位名稱 | 中文名稱 | 資料類型 | 長度 | 允許空值 | 備註 |
| itemID | 類別編號 | int4 | 32 | 否 | PK（自動遞增） |
| itemname | 類別名稱 | varchar | 255 | 否 |  |

表 8-2-6資料表描述：會員訂單明細

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 資料表名稱：orderdetail | | | | | |
| 欄位名稱 | 中文名稱 | 資料類型 | 長度 | 允許空值 | 備註 |
| orderdetailid | 訂單編號 | int4 | 32 | 否 | PK（自動遞增） |
| foodid | 餐點編號 | int4 | 32 | 否 | FK |
| foodno | 餐點數量 | int4 | 32 | 否 |  |
| customized | 備註 | varchar | 50 | 是 |  |
| memberphone | 會員電話 | varchar | 20 | 否 |  |
| tableno | 桌號 | int4 | 32 | 否 |  |
| total | 合計 | int4 | 32 | 是 |  |
| ordtime | 訂單建立時間 | timestamp | 6 | 是 |  |
| foodname | 餐點名稱 | varchar | 255 | 否 |  |
| sumtotal | 餐點總金額 | int4 | 32 | 是 |  |

表 8-2-7資料表描述：計算時間

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 資料表名稱：calculatingtime | | | | | |
| 欄位名稱 | 中文名稱 | 資料類型 | 長度 | 允許空值 | 備註 |
| timeserNo | 到達及結束時間流水號 | int4 | 32 | 是 | PK（自動遞增） |
| memberphone | 會員電話 | varchar | 20 | 否 | FK |
| arrivaltime | 到達時間 | timestamp | 6 | 否 |  |
| endtime | 結束時間 | timestamp | 6 | 否 |  |
| staymins | 停留時間 | int4 | 32 | 否 |  |
| ttotal | 遊玩花費時間 | int4 | 32 | 否 |  |
| confirm | 是否結帳 | bpchar | 1 | 是 |  |

表 8-2-8資料表描述：會員結帳明細

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 資料表名稱：checkout | | | | | |
| 欄位名稱 | 中文名稱 | 資料類型 | 長度 | 允許空值 | 備註 |
| serNo | 結帳編號 | int4 | 32 | 否 | PK（自動遞增） |
| staffphone | 員工電話 | varchar | 20 | 否 | FK |
| memberphone | 會員電話 | varchar | 20 | 否 | FK |
| totalpoint | 消費點數 | int4 | 32 | 否 |  |
| billtime | 結帳時間 | timestamp | 6 | 否 |  |

表 8-2-9資料表描述：會員儲值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 資料表名稱：topup | | | | | |
| 欄位名稱 | 中文名稱 | 資料類型 | 長度 | 允許空值 | 備註 |
| serNo | 儲值編號 | int4 | 32 | 否 | PK（自動遞增） |
| memberphone | 會員電話 | varchar | 20 | 否 | FK |
| staffphone | 員工電話 | varchar | 20 | 否 | FK |
| topup | 儲值金額 | int4 | 32 | 否 |  |
| topuppoints | 儲值點數 | int4 | 32 | 否 |  |
| topuptime | 儲值時間 | timestamp | 6 | 否 |  |

**第九章 程式**

**9-1 元件清單及其規格描述**

**員工端**

表 9-1-1首頁功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：routes/ | | |
| 編號 | 函式名稱 | 功能 |
| 1-1-1 | index.js | 顯示各功能之按鈕 |

表 9-1-2員工功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-2-1 | staffone.js | 呼叫搜尋員工資料服務，並顯示登入中之員工資料 |
| 1-2-2 | Staffupdate.js | 建立員工更新資料物件，並呼叫更新服務 |
| 1-2-3 | staffupdateno.js | 導至尋找員工資料之頁面 |
| 1-2-4 | Staffupdateform.js | 顯示員工原本資料於更新頁面欄位 |
| 檔案路徑：routes/utility/ | | |
| 1-2-5 | staff.js | 回應上述函式呼叫服務 |

表 9-1-3店家資訊功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-3-1 | storeadd.js | 建立店家資料物件並呼叫新增服務 |
| 1-3-2 | storeaddform.js | 導至店家資料新增頁面 |
| 1-3-3 | storelist.js | 呼叫搜尋店家資料之服務，並回傳及顯示店家資料 |
| 1-3-4 | storeremove.js | 取得欲刪除店家之編號並呼叫刪除服務 |
| 1-3-5 | storeremoveform.js | 導至店家資料刪除頁面 |
| 1-3-6 | storeupdate.js | 建立店家更新資料物件並呼叫更新服務 |
| 1-3-7 | storeupdateno.js | 導至搜尋店家編號之頁面 |
| 1-3-8 | storeupdateform.js | 將原有店家資料回傳至更新頁面之欄位 |
| 檔案路徑：routes/utility/ | | |
| 1-3-9 | store | 回應上述函式呼叫服務 |

表 9-1-4菜單資訊功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：routes | | |
| 編號 | 檔案名稱 | 功能 |
| 1-4-1 | food\_add.js | 建立菜單餐點物件並呼叫新增服務 |
| 1-4-2 | food\_add\_form.js | 導至餐點資料新增頁面 |
| 1-4-3 | food\_list.js | 呼叫搜尋餐點資料之服務，並回傳及顯示餐點資料 |
| 1-4-4 | food\_remove.js | 取得欲刪除餐點之編號並呼叫刪除服務 |
| 1-4-5 | food\_remove\_form.js | 導至餐點刪除頁面 |
| 1-4-6 | foodupdate.js | 建立餐點更新資料物件並呼叫更新服務 |
| 1-4-7 | foodupdateno.js | 導至搜尋餐點編號之頁面 |
| 1-4-8 | foodupdateform.js | 將原有餐點資料回傳至更新頁面之欄位 |
| 檔案路徑：routes/utility/ | | |
| 1-4-9 | food.js | 回應上述函式呼叫服務 |

表 9-1-5類別功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-5-1 | itemadd.js | 建立類別資料物件並呼叫新增服務 |
| 1-5-2 | itemaddform.js | 導至類別新增頁面 |
| 1-5-3 | itemlist.js | 呼叫搜尋類別資料之服務，並回傳及顯示類別資料 |
| 1-5-4 | itemremove.js | 取得欲刪除類別之編號並呼叫刪除服務 |
| 1-5-5 | itemremoveform.js | 導至類別刪除頁面 |
| 1-5-6 | itemupdate.js | 建立類別更新資料物件並呼叫更新服務 |
| 1-5-7 | itemupdateno.js | 導至搜尋類別編號之頁面 |
| 1-5-8 | itemupdateform.js | 將原有類別資料回傳至更新頁面之欄位 |
| 檔案路徑：routes/utility/ | | |
| 1-5-9 | item.js | 回應上述函式呼叫服務 |

表 9-1-6餐點訂單功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-6-1 | orderlist.js | 呼叫搜尋會員訂單之服務，並回傳及顯示當日會員訂單 |
| 檔案路徑：routes/utility/ | | |
| 1-6-2 | order.js | 回應上述函式呼叫服務 |

表 9-1-7會員消費明細功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-7-1 | checkoutlist.js | 呼叫搜尋會員結帳明細之服務，並回傳及顯示會員當日結帳明細 |
| 檔案路徑：routes/utility/ | | |
| 1-7-2 | checkout.js | 回應上述函式呼叫服務 |

表 9-1-8儲值點數功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-8-1 | topupadd.js | 建立會員儲值資料物件並呼叫新增服務 |
| 1-8-2 | topupaddform.js | 導至會員儲值新增頁面 |
| 檔案路徑與名稱：routes/utility/topup.js | | |
| 1-8-3 | topup | 回應上述函式呼叫服務 |

表 9-1-9使用者功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-9-1 | usershow.js | 顯示登入中員工名稱 |
| 1-9-2 | userlogout.js | 將員工登出 |
| 1-9-3 | userloginform.js | 判斷員工是否登入 |
| 1-9-4 | userlogin.js | 呼叫搜尋員工帳號密碼服務，並判斷是否存在 |
| 1-9-5 | checkAuth.js | 檢查登入權限 |
| 檔案路徑：routes/utility/ | | |
| 1-9-6 | user.js | 回應上述函式呼叫服務 |

**會員端**

表 9-1-10首頁功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：routes/ | | |
| 編號 | 函式名稱 | 功能 |
| 1-10-1 | index.js | 顯示各功能之按鈕 |

表 9-1-11使用者功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-11-1 | user\_show.js | 顯示登入中會員名稱 |
| 1-11-2 | user\_logout.js | 將會員登出 |
| 1-11-3 | user­\_login\_form.js | 判斷會員是否登入 |
| 1-11-4 | user\_login.js | 呼叫搜尋會員帳號密碼服務，並判斷是否存在 |
| 1-11-5 | checkAuth.js | 檢查登入權限 |
| 檔案路徑：routes/utility/ | | |
| 1-11-6 | user.js | 回應上述函式呼叫服務 |

表 9-1-12查看店家資訊功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-12-1 | storelist.js | 呼叫搜尋店家資料之服務，並回傳及顯示店家資料 |
| 檔案路徑：routes/utility/ | | |
| 1-12-2 | store.js | 回應上述函式呼叫服務 |

表 9-1-13訂單明細功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-13-1 | orderdetail\_one.js | 取得某一位會員的訂單明細 |
| 1-13-2 | orderdetail\_add.js | 建立訂單明細物件並呼叫新增服務 |
| 1-13-3 | orderdetail\_add\_form.js | 導至訂單明細新增頁面 |
| 1-13-4 | orderdetail\_remove.js | 取得欲刪除訂單明細之編號並呼叫刪除服務 |
| 1-13-5 | orderdetail\_remove\_form.js | 導至訂單明細刪除頁面 |
| 1-13-6 | orderdetail\_update.js | 建立訂單明細更新資料物件並呼叫更新服務 |
| 1-13-7 | orderdetail\_update\_no.js | 導至搜尋訂單明細編號之頁面 |
| 1-13-8 | orderdetail\_update\_form.js | 將原有訂單明細回傳至更新頁面之欄位 |
| 檔案路徑：routes/utility/ | | |
| 1-13-9 | orderdetail.js | 回應上述函式呼叫服務 |

表 9-1-14會員資料功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-14-1 | member\_one.js | 取得某一位會員的資料 |
| 1-14-2 | member\_add.js | 建立會員資料物件並呼叫新增服務 |
| 1-14-3 | member\_add\_form.js | 導至會員資料新增頁面 |
| 1-14-4 | member\_list.js | 呼叫搜尋會員資料之服務，並回傳及顯示會員資料 |
| 1-14-5 | member\_remove.js | 取得欲刪除會員之帳號並呼叫刪除服務 |
| 1-14-6 | member\_remove\_form.js | 導至會員資料刪除頁面 |
| 1-14-7 | member\_update.js | 建立會員更新資料物件並呼叫更新服務 |
| 1-14-8 | member\_update\_no.js | 導至搜尋會員帳號之頁面 |
| 1-14-9 | member\_update\_form.js | 將原有會員資料回傳至更新頁面之欄位 |
| 檔案路徑：routes/utility/ | | |
| 1-14-10 | member.js | 回應上述函式呼叫服務 |

表 9-1-15查看菜單功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-15-1 | food\_list.js | 呼叫搜尋菜單之服務，並回傳及顯示菜單 |
| 檔案路徑：routes/utility/ | | |
| 1-15-2 | food.js | 回應上述函式呼叫服務 |

表 9-1-16計時及結帳功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑：routes/ | | |
| 編號 | 檔案名稱 | 功能 |
| 1-16-1 | caltime\_add.js | 建立計時資料物件並呼叫新增服務 |
| 1-16-2 | caltime\_add\_form.js | 導至計時資料新增頁面 |
| 1-16-3 | caltime\_addend.js | 建立結帳資料物件並呼叫新增服務 |
| 1-16-4 | caltime\_addend\_form.js | 導至結帳資料頁面 |
| 檔案路徑：routes/utility/ | | |
| 1-16-5 | caltime.js | 回應上述函式呼叫服務 |

**員工端**

表 9-1-17首頁功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-1 | 檔案名稱 | index.js |
| 功能 | 顯示各功能之按鈕 | | |
| var express = require('express');  var router = express.Router();  /\* GET home page. \*/  router.get('/', function(req, res, next) {  res.render('index', { title: 'Express' });  });  module.exports = router; | | | |

表 9-1-18搜尋員工資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-2-1 | 檔案名稱 | staffone.js |
| 功能 | 呼叫搜尋員工資料服務，並顯示登入中之員工資料 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  var moment = require('moment');  const staff = require('./utility/staff');  //接收GET請求  router.get('/', function(req, res, next) {  var staffphone = req.session.staffphone;  staff.one(staffphone).then(data => {  if (data==null){  res.render('error'); //導向錯誤頁面  }else if(data==-1){  res.render('notFound'); //導向找不到頁面  }else{  res.render('stafflist', {item:data}); //將資料傳給顯示頁面  }  })  });  module.exports = router; | | | |

表 9-1-19 建立員工更新資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-2-2 | 檔案名稱 | staffupdate.js |
| 功能 | 建立員工更新資料物件，並呼叫更新服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const staff = require('./utility/staff');  //接收POST請求  router.post('/', function(req, res, next) {  var username = req.body.username; //取得員工姓名  var newData={  username:username, //員工姓名  staffphone: req.body.staffphone, //取得員工電話  nickname: req.body.nickname, //取得暱稱  password: req.body.password //取得密碼  }    staff.update(newData).then(d => {  if (d>=0){  res.render('updateSuccess', {results:d}); //傳至成功頁面  }else{  res.render('updateFail'); //導向錯誤頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-20導至尋找員工資料之頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-2-3 | 檔案名稱 | staffupdateno.js |
| 功能 | 導至尋找員工資料之頁面 | | |
| var express = require('express');  var router = express.Router();  /\* GET home page. \*/  router.get('/', function(req, res, next) {  res.render('staffupdateno');  });  //匯出  module.exports = router; | | | |

表 9-1-21 顯示員工原本資料於更新頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-2-4 | 檔案名稱 | staffupdateform.js |
| 功能 | 顯示員工原本資料於更新頁面欄位 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  var moment = require('moment');  const staff = require('./utility/staff');  //接收GET請求  router.get('/', function(req, res, next) {  var no = req.session.staffphone;  staff.query(no).then(d => {  if (d!=null && d!=-1){  var data = {  username: d.username,  staffphone: d.staffphone,  nickname: d.nickname,  password: d.password,  }  res.render('staffupdateform', {item:data}); //將資料傳給更新頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-22 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-2-5 | 檔案名稱 | staff.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //------------------------------------------  //執行資料庫動作的函式-取出單一員工  //------------------------------------------  var one = async function(staffphone){  var result={};    await sql('SELECT \* FROM staff WHERE staffphone = $1', [staffphone])  .then((data) => {  if(data.rows.length > 0){  result = data.rows[0];  }else{  result = -1;  }  }, (error) => {  result = null;  });    return result;  }  //------------------------------------------  //執行資料庫動作的函式-取出單一員工  //------------------------------------------  var query = async function(staffphone){  var result={};    await sql('SELECT \* FROM staff WHERE "staffphone" = $1', [staffphone])  .then((data) => {  if(data.rows.length > 0){  result = data.rows[0];  }else{  result = -1;  }  }, (error) => {  result = null;  });    return result;  }  //----------------------------------  // 更新員工資料  //----------------------------------  var update = async function(newData){  var results;  await sql('UPDATE staff SET "username"=$2, "nickname"=$3, "password"=$4 WHERE "staffphone" = $1', [newData.staffphone, newData.username, newData.nickname, newData.password])  .then((data) => {  results = data.rowCount;  }, (error) => {  results = -1;  });    return results;  }  module.exports = {one, query, update} | | | |

表 9-1-23 建立店家資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-1 | 檔案名稱 | storeadd.js |
| 功能 | 建立店家資料物件並呼叫新增服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const store = require('./utility/store');  //接收POST請求  router.post('/', function(req, res, next) {  var storename = req.body.storename;  var storeaddress = req.body.storeaddress;  var phoneno = req.body.phoneno;  var vacanttable = Number(req.body.vacanttable);  var businesshours = req.body.businesshours;  var wifi = req.body.wifi;  var socket = req.body.socket;  var providemeals = req.body.providemeals;  var outsidefood = req.body.outsidefood;  var chargingstandards = req.body.chargingstandards;  var atime = Number(req.body.atime);  var apoint = Number(req.body.apoint);  var lessatime = Number(req.body.lessatime);  var addapoint = Number(req.body.addapoint);  // 建立一個新資料物件  var newData={  storename:storename,  storeaddress:storeaddress,  phoneno:phoneno,  vacanttable:vacanttable,  businesshours:businesshours,  wifi:wifi,  socket:socket,  providemeals:providemeals,  outsidefood:outsidefood,  chargingstandards:chargingstandards,  atime:atime,  apoint:apoint,  lessatime:lessatime,  addapoint:addapoint  }    store.add(newData).then(d => {  if (d==0){  res.render('addSuccess'); //傳至成功頁面  }else{  res.render('addFail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-24導至店家資料新增頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-2 | 檔案名稱 | storeaddform.js |
| 功能 | 導至店家資料新增頁面 | | |
| var express = require('express');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  res.render('storeaddform');  });  module.exports = router; | | | |

表 9-1-25 搜尋店家資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-3 | 檔案名稱 | storelist.js |
| 功能 | 呼叫搜尋店家資料之服務，並回傳並顯示店家資料 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const store = require('./utility/store');  //接收GET請求  router.get('/', function(req, res, next) {  store.list().then(data => {  if(data==null){  res.render('error'); //導向錯誤頁面  }else if(data.length > 0){  res.render('storelist', {items:data}); //將資料傳給顯示頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  module.exports = router; | | | |

表 9-1-26 取得欲刪除店家之編號列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-4 | 檔案名稱 | storeremove.js |
| 功能 | 取得欲刪除店家之編號並呼叫刪除服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const store = require('./utility/store');  //接收POST請求  router.post('/', function(req, res, next) {  var storeID = req.body.storeID; //取得店家編號    store.remove(storeID).then(d => {  if(d>=0){  res.render('removeSuccess', {results:d}); //傳至成功頁面  }else{  res.render('removeFail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-27導至店家資料刪除頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-5 | 函式名稱 | storeremoveform.js |
| 功能 | 導至店家資料刪除頁面 | | |
| var express = require('express');  const store = require('./utility/store');  var router = express.Router();  router.get('/', function(req, res, next) {  store.getDropdownData().then(d => {  if (d!=[]){  res.render('storeremoveform', {result:d}); //轉至新增頁面  }else{  res.render('removeFail'); //導向錯誤頁面  }  });  });  module.exports = router; | | | |

表 9-1-28 建立店家更新資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-6 | 檔案名稱 | storeupdate.js |
| 功能 | 建立店家更新資料物件並呼叫更新服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const store = require('./utility/store');  //接收POST請求  router.post('/', function(req, res, next) {  var storeID = req.body.storeID; //取得店家編號  var newData={  storeID:storeID, //店家編號  storename: req.body.storename, //取得店家名稱  storeaddress: req.body.storeaddress, //取得店家地址  phoneno: req.body.phoneno, //取得電話號碼  vacanttable: Number(req.body.vacanttable), //取得空桌數  businesshours: req.body.businesshours, //取得營業時間  wifi: req.body.wifi, //取得提供wifi  socket: req.body.socket, //取得提供插座  providemeals: req.body.providemeals, //取得提供餐點  outsidefood: req.body.outsidefood, //取得可帶外食  chargingstandards: req.body.chargingstandards, //取得收費標準  atime: Number(req.body.atime),  apoint: Number(req.body.apoint),  lessatime: Number(req.body.lessatime),  addapoint: Number(req.body.addapoint)  }    store.update(newData).then(d => {  if (d>=0){  res.render('updateSuccess', {results:d}); //傳至成功頁面  }else{  res.render('updateFail'); //導向錯誤頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-29導至搜尋店家編號之頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-7 | 檔案名稱 | storeupdateno.js |
| 功能 | 導至搜尋店家編號之頁面 | | |
| var express = require('express');  var router = express.Router();  /\* GET home page. \*/  router.get('/', function(req, res, next) {  res.render('staffupdateno');  });  //匯出  module.exports = router; | | | |

表 9-1-30 原有店家資料回傳至更新頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-8 | 檔案名稱 | storeupdateform.js |
| 功能 | 將原有店家資料回傳至更新頁面之欄位 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  var moment = require('moment');  const store = require('./utility/store');  //接收GET請求  router.get('/', function(req, res, next) {  var no = req.query.storeID;  store.query(no).then(d => {  if (d!=null && d!=-1){  var data = {  storeID: d.storeID,  storename: d.storename,  storeaddress: d.storeaddress,  phoneno: d.phoneno,  vacanttable: d.vacanttable,  businesshours: d.businesshours,  wifi: d.wifi,  socket: d.socket,  providemeals: d.providemeals,  outsidefood: d.outsidefood,  chargingstandards: d.chargingstandards,  atime: d.atime,  apoint: d.apoint,  lessatime: d.lessatime,  addapoint: d.addapoint,  }  res.render('storeupdateform', {item:data}); //將資料傳給更新頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-31 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-9 | 檔案名稱 | store.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //------------------------------------------  //執行資料庫動作的函式-傳回所有店家資訊  //------------------------------------------  var list = async function(){  var result=[];  //console.log("查詢店家資訊");  await sql('SELECT \* FROM storeinformation')  .then((data) => {  result = data.rows;  console.log(result) ;  }, (error) => {  result = null;  //console.log("除去錯誤") ;  });    return result;  }  //------------------------------------------  // 新增店家資訊  //------------------------------------------  var add = async function(newData){  var result;  console.log(newData)  await sql('INSERT INTO storeinformation ("storename", "storeaddress", "phoneno", "vacanttable", "businesshours", "wifi", "socket", "providemeals", "outsidefood", "chargingstandards", "atime", "apoint", "lessatime", "addapoint") VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9, $10, $11, $12, $13, $14)', [newData.storename, newData.storeaddress, newData.phoneno, newData.vacanttable, newData.businesshours, newData.wifi, newData.socket, newData.providemeals,newData.outsidefood, newData.chargingstandards, newData.atime, newData.apoint, newData.lessatime, newData.addapoint])  .then((data) => {  result = 0;  }, (error) => {  result = -1;  });    return result;  }  //----------------------------------  // 刪除員工  //----------------------------------  var remove = async function(storeID){  var result;  await sql('DELETE FROM storeinformation WHERE "storeID"= $1', [storeID])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });    return result;  }  //------------------------------------------  //執行資料庫動作的函式-取出單一店家資訊  //------------------------------------------  var query = async function(storeID){  var result={};    await sql('SELECT \* FROM storeinformation WHERE "storeID" = $1', [storeID])  .then((data) => {  if(data.rows.length > 0){  result = data.rows[0];  }else{  result = -1;  }  }, (error) => {  result = null;  });    return result;  }  //------------------------------------------  // 取出型態資料  //------------------------------------------  var getDropdownData = async function(){  //儲存下拉式選單資料  var storeinformation;    //取回protype資料  await sql('SELECT \* FROM storeinformation ORDER BY "storeID"')  .then((data) => {  storeinformation = data.rows;  }, (error) => {  result = [];  });    //設定回傳資料  var result = {};  result.storeinformation = storeinformation;  //回傳  return result;  }  //----------------------------------  // 更新店家資料  //----------------------------------  var update = async function(newData){  var results;  await sql('UPDATE storeinformation SET "storename"=$2, "storeaddress"=$3, "phoneno"=$4, "vacanttable"=$5, "businesshours"=$6, "wifi"=$7, "socket"=$8, "providemeals"=$9, "outsidefood"=$10, "chargingstandards"=$11, "atime"=$12, "apoint"=$13, "lessatime"=$14, "addapoint"=$15 WHERE "storeID" = $1', [newData.storeID, newData.storename, newData.storeaddress, newData.phoneno, newData.vacanttable, newData.businesshours, newData.wifi, newData.socket, newData.providemeals, newData.outsidefood, newData.chargingstandards, newData.atime, newData.apoint, newData.lessatime, newData.addapoint])  .then((data) => {  results = data.rowCount;  }, (error) => {  results = -1;  });    return results;  }  module.exports = {list, add, remove, query, getDropdownData, update} | | | |

表 9-1-32 建立菜單餐點物件列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-4-1 | 檔案名稱 | food\_add.js |
| 功能 | 建立菜單餐點物件並呼叫新增服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const food = require('./utility/food');  //---------------------------  // 引用multer外掛  //---------------------------  const multer = require('multer');  // 宣告上傳存放空間及檔名更改  var storage = multer.diskStorage({  destination: function (req, file, cb) {  cb(null, 'public/imgs');  },  filename: function (req, file, cb) {  cb(null, Date.now()+"--"+file.originalname);  }  })  // 產生multer的上傳物件  var maxSize=1024\*1024; //設定最大可接受圖片大小(1M)  var upload = multer({  storage:storage  })  //---------------------------  //接收POST請求  router.post('/',upload.single('foodimg'), function(req, res, next) {    var itemname = req.body.itemname;  var foodname = req.body.foodname;  var foodpoint = Number(req.body.foodpoint);  var foodimg;  // 如果有選擇圖片  if (typeof(req.file) != 'undefined'){  foodimg=req.file.filename; //取得上傳照片名稱  }  // 建立一個新資料物件  var newData={  itemname:itemname,  foodname:foodname,  foodpoint:foodpoint,  foodimg:foodimg  }    food.add(newData).then(d => {  if (d==0){  res.render('addSuccess'); //傳至成功頁面  }else{  res.render('addFail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-33導至餐點資料新增頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-4-2 | 檔案名稱 | food\_add\_form.js |
| 功能 | 導至餐點資料新增頁面 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const food = require('./utility/food');  //接收GET請求  router.get('/', function(req, res, next) {  food.getDropdownData().then(d => {  if (d!=[]){  res.render('food\_add\_form', {result:d}); //轉至新增頁面  }else{  res.render('addFail'); //導向錯誤頁面  }  });  });  module.exports = router; | | | |

表 9-1-34 搜尋餐點資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-4-3 | 檔案名稱 | food\_list.js |
| 功能 | 呼叫搜尋餐點資料之服務，並回傳並顯示餐點資料 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const food = require('./utility/food');  //接收GET請求  router.get('/', function(req, res, next) {  food.list().then(data => {  if(data==null){  res.render('error'); //導向錯誤頁面  }else if(data.length > 0){  console.log(data);  res.render('food\_list', {items:data}); //將資料傳給顯示頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  module.exports = router; | | | |

表 9-1-35 取得欲刪除餐點之編號列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-4-4 | 檔案名稱 | food\_remove.js |
| 功能 | 取得欲刪除餐點之編號並呼叫刪除服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const food = require('./utility/food');  //接收POST請求  router.post('/', function(req, res, next) {  var foodid = req.body.foodid; //取得餐點編號    food.remove(foodid).then(d => {  if(d>=0){  res.render('removeSuccess', {results:d}); //傳至成功頁面  }else{  res.render('removefoodFail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-36導至餐點刪除頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-4-5 | 檔案名稱 | food\_remove\_form.js |
| 功能 | 導至餐點刪除頁面 | | |
| var express = require('express');  var router = express.Router();  const food = require('./utility/food');  /\*  //接收GET請求  router.get('/', function(req, res, next) {  res.render('food\_remove\_form');  });  module.exports = router;  \*/  //接收GET請求  router.get('/', function(req, res, next) {  food.getfoodnameData().then(d => {  if (d!=[]){  res.render('food\_remove\_form', {result:d});  }else{  res.render('removefoodFail'); //導向錯誤頁面  }  });  });  module.exports = router; | | | |

表 9-1-37建立餐點更新資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-4-6 | 檔案名稱 | foodupdate.js |
| 功能 | 建立餐點更新資料物件並呼叫更新服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const food = require('./utility/food');  //接收POST請求  router.post('/', function(req, res, next) {  var foodid = req.body.foodid; //取得餐點編號  var newData={  foodid:foodid, //餐點編號  itemname: req.body.itemname, //取得類別編號  foodname: req.body.foodname, //取得餐點名稱  foodpoint: req.body.foodpoint, //取得餐點點數  }    food.update(newData).then(d => {  if (d>=0){  res.render('updatesuccesss', {result:d}); //傳至成功頁面  }else{  res.render('updateFail'); //導向錯誤頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-38導至搜尋餐點編號之頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-4-7 | 檔案名稱 | foodupdateno.js |
| 功能 | 導至搜尋餐點編號之頁面 | | |
| var express = require('express');  var router = express.Router();  const food = require('./utility/food');  /\* GET home page. \*/  /\*  router.get('/', function(req, res, next) {  res.render('foodupdateno');  });  //匯出  module.exports = router;  \*/  //接收GET請求  router.get('/', function(req, res, next) {  food.getfoodnameData().then(d => {  if (d!=[]){  res.render('foodupdateno', {result:d}); //轉至新增頁面  }else{  res.render('notFound'); //導向錯誤頁面  }  });  });  module.exports = router; | | | |

表 9-1-39 原有餐點資料回傳至更新頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-4-8 | 檔案名稱 | foodupdateform.js |
| 功能 | 將原有餐點資料回傳至更新頁面之欄位 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  var moment = require('moment');  const food = require('./utility/food');  //接收GET請求  router.get('/', function(req, res, next) {  var no = req.query.foodid;  food.query(no).then(d => {  if (d!=null && d!=-1){  res.render('foodupdateform', {result:d}); //將資料傳給更新頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-40 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-4-9 | 檔案名稱 | food.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //------------------------------------------  //執行資料庫動作的函式-傳回所有餐點資料  //------------------------------------------  var list = async function(){  var result=[];  console.log("查詢餐點");  await sql('SELECT \* FROM food')  .then((data) => {  result = data.rows;  console.log(result) ;  }, (error) => {  result = null;  //console.log("除去錯誤") ;  });    return result;  }  //------------------------------------------  // 取出型態資料  //------------------------------------------  var getDropdownData = async function(){  //儲存下拉式選單資料  var item;    //取回protype資料  await sql('SELECT \* FROM item ORDER BY "itemID"')  .then((data) => {  item = data.rows;  }, (error) => {  result = [];  });    //設定回傳資料  var result = {};  result.item = item;  //回傳  return result;  }  //------------------------------------------  // 取出型態資料  //------------------------------------------  var getfoodnameData = async function(){  //儲存下拉式選單資料  var food;    //取回protype資料  await sql('SELECT \* FROM food ORDER BY "foodid"')  .then((data) => {  food = data.rows;  }, (error) => {  result = [];  });    //設定回傳資料  var result = {};  result.food = food;  //回傳  return result;  }  //------------------------------------------  // 新增餐點  //------------------------------------------  var add = async function(newData){  var result;  console.log(newData)  await sql('INSERT INTO food ( "itemname", "foodname", "foodpoint", "foodimg") VALUES ($1, $2, $3, $4)', [newData.itemname, newData.foodname, newData.foodpoint, newData.foodimg])  .then((data) => {  result = 0;  }, (error) => {  result = -1;  });    return result;  }  //----------------------------------  // 刪除餐點  //----------------------------------  var remove = async function(foodid){  var result;  await sql('DELETE FROM food WHERE "foodid" = $1', [foodid])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });    return result;  }  //------------------------------------------  //執行資料庫動作的函式-取出單一餐點  //------------------------------------------  var query = async function(foodid){  var result={};  await sql('SELECT \* FROM food WHERE "foodid" = $1', [foodid])  .then((data) => {  if(data.rows.length > 0){  result.food = data.rows[0];  }else{  result.food = -1;  }  }, (error) => {  result.food = null;  });    //取回protype資料  await sql('SELECT \* FROM item ORDER BY "itemID"')  .then((data) => {  result.item = data.rows;  }, (error) => {  result.item = [];  });  console.log(result)      return result;  }  //----------------------------------  // 更新餐點資料  //----------------------------------  var update = async function(newData){  var results;  console.log(newData)  await sql('UPDATE food SET "itemname"=$2, "foodname"=$3, "foodpoint"=$4 WHERE "foodid" = $1', [newData.foodid, newData.itemname, newData.foodname, newData.foodpoint])  .then((data) => {  results = data.rowCount;  }, (error) => {  results = -1;  });    return results;  }  module.exports = {getDropdownData, list, add, remove, query, update, getfoodnameData}food\_remove.js | | | |

表 9-1-41 建立類別資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-1 | 檔案名稱 | itemadd.js |
| 功能 | 建立類別資料物件並呼叫新增服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const item = require('./utility/item');  //接收POST請求  router.post('/', function(req, res, next) {  var itemname = req.body.itemname; //取得類別名稱    // 建立一個新資料物件  var newData={  itemname:itemname,  }  console.log(newData)  item.add(newData).then(d => {  if (d==0){  res.render('addSuccess'); //傳至成功頁面  }else{  res.render('addFail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-42導至類別新增頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-2 | 檔案名稱 | itemaddform.js |
| 功能 | 導至類別新增頁面 | | |
| var express = require('express');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  res.render('itemaddform');  });  module.exports = router; | | | |

表 9-1-43搜尋類別資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-3 | 檔案名稱 | itemlist.js |
| 功能 | 呼叫搜尋類別資料之服務，並回傳並顯示類別資料 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const item = require('./utility/item');  //接收GET請求  router.get('/', function(req, res, next) {  item.list().then(data => {  if(data==null){  res.render('error'); //導向錯誤頁面  }else if(data.length > 0){  console.log(data);  res.render('itemlist', {items:data}); //將資料傳給顯示頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  module.exports = router; | | | |

表 9-1-44 取得欲刪除類別之編號列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-4 | 函式名稱 | itemremove |
| 功能 | 取得欲刪除類別之編號並呼叫刪除服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const item = require('./utility/item');  //接收POST請求  router.post('/', function(req, res, next) {  var itemID = req.body.itemID; //取得類別編號    item.remove(itemID).then(d => {  if(d>=0){  res.render('removeSuccess', {results:d}); //傳至成功頁面  }else{  res.render('removeFail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-45導至類別刪除頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-5 | 檔案名稱 | itememoveform.js |
| 功能 | 導至類別刪除頁面 | | |
| var express = require('express');  const item = require('./utility/item');  var router = express.Router();  /\*  //接收GET請求  router.get('/', function(req, res, next) {  res.render('itemremoveform');  });  module.exports = router;  \*/  //接收GET請求  router.get('/', function(req, res, next) {  item.getDropdownData().then(d => {  if (d!=[]){  res.render('itemremoveform', {result:d}); //轉至新增頁面  }else{  res.render('removeFail'); //導向錯誤頁面  }  });  });  module.exports = router; | | | |

表 9-1-46建立類別更新資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-6 | 檔案名稱 | itemupdate.js |
| 功能 | 建立類別更新資料物件並呼叫更新服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const item = require('./utility/item');  //接收POST請求  router.post('/', function(req, res, next) {  var itemID = req.body.itemID; //取得累別編號  var newData={  itemID:itemID, //類別編號  itemname: req.body.itemname, //取得類別名稱  }    item.update(newData).then(d => {  if (d>=0){  res.render('updateSuccess', {results:d}); //傳至成功頁面  }else{  res.render('updateFail'); //導向錯誤頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-47導至搜尋類別編號之頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-7 | 檔案名稱 | itemupdateno.js |
| 功能 | 導至搜尋類別編號之頁面 | | |
| var express = require('express');  var router = express.Router();  const item = require('./utility/item');  /\* GET home page. \*/  /\*  router.get('/', function(req, res, next) {  res.render('itemupdateno');  });  //匯出  module.exports = router;  \*/  //接收GET請求  router.get('/', function(req, res, next) {  item.getDropdownData().then(d => {  if (d!=[]){  res.render('itemupdateno', {result:d}); //轉至新增頁面  }else{  res.render('notFound'); //導向錯誤頁面  }  });  });  module.exports = router; | | | |

表 9-1-48 原有類別資料回傳至更新頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-8 | 檔案名稱 | itemupdateform.js |
| 功能 | 將原有類別資料回傳至更新頁面之欄位 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  var moment = require('moment');  const item = require('./utility/item');  //接收GET請求  router.get('/', function(req, res, next) {  var no = req.query.itemID;    item.query(no).then(d => {  if (d!=null && d!=-1){  var data = {  itemID: d.itemID,  itemname: d.itemname,  }  res.render('itemupdateform', {item:data}); //將資料傳給更新頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-49 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-9 | 檔案名稱 | item.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //------------------------------------------  //執行資料庫動作的函式-傳回所有類別資料  //------------------------------------------  var list = async function(){  var result=[];  console.log("查詢類別");  await sql('SELECT \* FROM item ORDER BY "itemID" ')  .then((data) => {  result = data.rows;  console.log(result) ;  }, (error) => {  result = null;  //console.log("除去錯誤") ;  });    return result;  }  //------------------------------------------  // 取出型態資料  //------------------------------------------  var getDropdownData = async function(){  //儲存下拉式選單資料  var item;    //取回protype資料  await sql('SELECT \* FROM item ORDER BY "itemID"')  .then((data) => {  item = data.rows;  }, (error) => {  result = [];  });    //設定回傳資料  var result = {};  result.item = item;  //回傳  return result;  }  //------------------------------------------  // 新增類別  //------------------------------------------  var add = async function(newData){  var result;  await sql('INSERT INTO item ("itemname") VALUES ($1)', [newData.itemname])  .then((data) => {  result = 0;  }, (error) => {  result = -1;  });    return result;  }  //----------------------------------  // 刪除類別  //----------------------------------  var remove = async function(itemID){  var result;  await sql('DELETE FROM item WHERE "itemID" = $1', [itemID])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });    return result;  }  //------------------------------------------  //執行資料庫動作的函式-取出單一類別  //------------------------------------------  var query = async function(itemID){  var result={};    await sql('SELECT \* FROM item WHERE "itemID" = $1', [itemID])  .then((data) => {  if(data.rows.length > 0){  result = data.rows[0];  }else{  result = -1;  }  }, (error) => {  result = null;  });    return result;  }  //----------------------------------  // 更新類別資料  //----------------------------------  var update = async function(newData){  var results;  console.log(newData)  await sql('UPDATE item SET "itemname"=$2 WHERE "itemID" = $1', [newData.itemID, newData.itemname])  .then((data) => {  results = data.rowCount;  }, (error) => {  results = -1;  });    return results;  }  module.exports = {list, add, remove, query, update, getDropdownData} | | | |

表 9-1-50 搜尋會員訂單之服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-6-1 | 檔案名稱 | orderlist.js |
| 功能 | 呼叫搜尋會員訂單之服務，並回傳顯示當日會員訂單 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const order = require('./utility/order');  var moment = require('moment');  //接收GET請求  router.get('/', function(req, res, next) {  order.list().then(data => {  if(data==null){  res.render('error'); //導向錯誤頁面  }else if(data.length > 0){  for(var i=0; i<data.length; i++){  data[i].ordtime=moment(data[i].ordtime).format("YYYY-MM-DD hh:mm:ss")  }  console.log(data);  res.render('orderlist', {items:data}); //將資料傳給顯示頁面  }else{  res.render('ordernotFound'); //導向找不到頁面  }  })  });  module.exports = router; | | | |

表 9-1-51 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-6-2 | 檔案名稱 | order.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //------------------------------------------  //執行資料庫動作的函式-傳回所有訂單資料  //------------------------------------------  var list = async function(){  var result=[];  console.log("查詢訂單");  await sql('SELECT \* FROM orderdetail WHERE date("ordtime") = date(now()) ORDER BY "ordtime" DESC')  .then((data) => {  result = data.rows;  console.log(result);  }, (error) => {  result = null;  //console.log("除去錯誤");  });    return result;  }  module.exports = {list} | | | |

表 9-1-52 搜尋會員結帳明細列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-1 | 檔案名稱 | checkoutlist.js |
| 功能 | 呼叫搜尋會員結帳明細之服務，並回傳顯示會員當日結帳明細 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const checkout = require('./utility/checkout');  var moment = require('moment');  //接收GET請求  router.get('/', function(req, res, next) {  checkout.list().then(data => {  if(data==null){  res.render('error'); //導向錯誤頁面  }else if(data.length > 0){  for(var i=0; i<data.length; i++){  data[i].billtime=moment(data[i].billtime).format("YYYY-MM-DD hh:mm:ss")  }  console.log(data);  res.render('checkoutlist', {items:data}); //將資料傳給顯示頁面  }else{  res.render('checknotFound'); //導向找不到頁面  }  })  });  module.exports = router; | | | |

表 9-1-53 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-2 | 檔案名稱 | checkout.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //------------------------------------------  //執行資料庫動作的函式-傳回所有結帳明細  //------------------------------------------  var list = async function(){  var result=[];  console.log("查詢結帳明細");  await sql('SELECT \* FROM checkout WHERE date("billtime") = date(now()) ORDER BY "billtime" DESC')  .then((data) => {  result = data.rows;  console.log(result);  }, (error) => {  result = null;  //console.log("除去錯誤") ;  });    return result;  }  module.exports = {list} | | | |

表 9-1-54 建立會員儲值資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-1 | 檔案名稱 | topupadd.js |
| 功能 | 建立會員儲值資料物件並呼叫新增服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const topupp = require('./utility/topupp');  //接收POST請求  router.post('/', function(req, res, next) {  var memberphone = req.body.memberphone;  var staffphone = req.session.staffphone;  var topup = Number(req.body.topup);  var topuppoints = Number(req.body.topuppoints);    console.log(staffphone)  // 建立一個新資料物件  var newData={  memberphone:memberphone,  staffphone:staffphone,  topup:topup,  topuppoints:topuppoints,    }  console.log(newData)  topupp.add(newData).then(d => {  if (d>0){  res.render('topupSuccess'); //傳至成功頁面  }else{  res.render('topupFail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-55導至會員儲值新增頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-2 | 檔案名稱 | topupaddform.js |
| 功能 | 導至會員儲值新增頁面 | | |
| var express = require('express');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  res.render('topupaddform');  });  module.exports = router; | | | |

表 9-1-56 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-3 | 檔案名稱 | topup.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  const session = require('express-session');  var add = async function(newData){  var result;  const current = new Date();  await sql('INSERT INTO topup ("memberphone" , "staffphone" , "topup", "topuppoints", "topuptime") VALUES ($1, $2, $3, $4, $5)', [newData.memberphone, newData.staffphone, newData.topup, newData.topuppoints, current])  .then((data) => {  result = 0;  }, (error) => {  return -1;  });  await sql('UPDATE member SET "points" = "points" + $2 WHERE "memberphone"=$1', [newData.memberphone, newData.topuppoints])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });    return result;  }  module.exports = {add} | | | |

表 9-1-57顯示登入中員工名稱列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-9-1 | 檔案名稱 | usershow.js |
| 功能 | 顯示登入中員工名稱 | | |
| var express = require('express');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  var name = req.session.name;  if(name==null || name==undefined){  name = '尚未登入';  }  res.render('usershow', { name: name });  });  module.exports = router; | | | |

表 9-1-58將員工登出

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-9-2 | 檔案名稱 | userlogout.js |
| 功能 | 將員工登出 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const user = require('./utility/user');  //接收POST請求  router.get('/', function(req, res, next) {  req.session.staffphone = null;  req.session.name = null;  res.render('usershow', {name:'已登出'}); //傳至登出  });  module.exports = router; | | | |

表 9-1-59判斷員工是否登入

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-9-3 | 檔案名稱 | userloginform.js |
| 功能 | 判斷員工是否登入 | | |
| var express = require('express');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  if(req.session.staffphone != null){  res.render('alreadylogin');  }else{  res.render('userloginform');  }  });  module.exports = router; | | | |

表 9-1-60 搜尋員工帳號密碼服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-9-4 | 檔案名稱 | userlogin.js |
| 功能 | 呼叫搜尋員工帳號密碼服務，並判斷是否存在 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const user = require('./utility/user');  //接收POST請求  router.post('/', function(req, res, next) {  var id = req.body.id; //取得帳號  var password = req.body.password; //取得密碼  user.login(id, password).then(d => {  if (d==null){  req.session.staffphone = null;  req.session.name = null;  res.render('loginFail'); //傳至登入失敗  }else{    req.session.staffphone = d.staffphone;  req.session.name = d.username;  res.render('usershow', {name:d.username}); //導向使用者  console.log('已登入');  console.log(d.staffphone)  console.log(d.username)  }  })  });  module.exports = router; | | | |

表 9-1-61檢查登入權限列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-9-5 | 檔案名稱 | checkAuth.js |
| 功能 | 檢查登入權限 | | |
| var express = require('express');  var router = express.Router();  //處理GET, POST, PUT, DELETE等所有請求  router.all('/', function(req, res, next) {  //檢查是否有session註記  var id = null;  try{  id = req.session.staffphone;  }catch(err){  id = null;  }  if(id===null || id===undefined){  res.render('unAuth'); //導向無權限畫面  }else{  next(); //執行在app.use()中, 串接在checkAuth之後的函式  }  });  module.exports = router; | | | |

表 9-1-62 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-9-6 | 檔案名稱 | user.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //---------------------------------------------  // 使用者登入  //---------------------------------------------  var login = async function(id, password){  var result;  //取得員工資料  await sql('SELECT \* FROM staff WHERE "staffphone"=$1 and "password"=$2', [id, password])  .then((data) => {  if(data.rows.length > 0){  result = data.rows[0];  }else{  result = null;  }  }, (error) => {  result = null;  });    //回傳物件  return result;  }  //匯出  module.exports = {login}; | | | |

**會員端**

表 9-1-63首頁功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-10-1 | 函式名稱 | index.js |
| 功能 | 顯示各功能之按鈕 | | |
| var express = require('express');  var router = express.Router();  /\* GET home page. \*/  router.get('/', function(req, res, next) {  res.render('index', { title: 'Express' });  });  module.exports = router; | | | |

表 9-1-64顯示登入中會員名稱列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-11-1 | 函式名稱 | user\_show.js |
| 功能 | 顯示登入中會員名稱 | | |
| var express = require('express');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  var name = req.session.name;  if(name==null || name==undefined){  name = '尚未登入';  }  res.render('user\_show', { name: name });  });  module.exports = router; | | | |

表 9-1-65將會員登出列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-11-2 | 函式名稱 | user\_logout.js |
| 功能 | 將會員登出 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const user = require('./utility/user');  //接收POST請求  router.get('/', function(req, res, next) {  req.session.memberphone = null;  req.session.name = null;  res.render('user\_show', {name:'已登出'}); //傳至登出  });  module.exports = router; | | | |

表 9-1-66判斷會員是否登入列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-11-3 | 函式名稱 | user­\_login\_form.js |
| 功能 | 判斷會員是否登入 | | |
| var express = require('express');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  if(req.session.memberphone != null){  res.render('login\_already');  }else{  res.render('user\_login\_form');  }  });  module.exports = router; | | | |

表 9-1-67搜尋會員帳號密碼服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-11-4 | 函式名稱 | user\_login.js |
| 功能 | 呼叫搜尋會員帳號密碼服務，並判斷是否存在 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const user = require('./utility/user');  //接收POST請求  router.post('/', function(req, res, next) {  var id = req.body.id; //取得帳號  var password = req.body.password; //取得密碼  user.login(id, password).then(d => {  if (d==null){  req.session.memberphone = null;  req.session.rname = null;  res.render('loginFail'); //傳至登入失敗  }else{  req.session.memberphone = d.memberphone;  req.session.name = d.membername;  res.render('user\_show', {name:d.membername}); //導向使用者  }  })  });  module.exports = router; | | | |

表 9-1-68檢查登入權限列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-11-5 | 函式名稱 | checkAuth.js |
| 功能 | 檢查登入權限 | | |
| var express = require('express');  var router = express.Router();  //處理GET, POST, PUT, DELETE等所有請求  router.all('/', function(req, res, next) {  //檢查是否有session註記  var id = null;  try{  id = req.session.memberphone;  }catch(err){  id = null;  }  if(id===null || id===undefined){  res.render('unAuth'); //導向無權限畫面  }else{  next(); //執行在app.use()中, 串接在checkAuth之後的函式  }  });  module.exports = router; | | | |

表 9-1-69 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-11-6 | 函式名稱 | user.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //---------------------------------------------  // 使用者登入  //---------------------------------------------  var login = async function(id, password){  var result;  //取得員工資料  await sql('SELECT \* FROM member WHERE memberphone=$1 and password=$2', [id, password])  .then((data) => {  if(data.rows.length > 0){  result = data.rows[0];  }else{  result = null;  }  }, (error) => {  result = null;  });    //回傳物件  return result;  }  //匯出  module.exports = {login}; | | | |

表 9-1-70 搜尋店家資料之服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-12-1 | 函式名稱 | storelist.js |
| 功能 | 呼叫搜尋店家資料之服務，並回傳及顯示店家資料 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const store = require('./utility/store');  //接收GET請求  router.get('/', function(req, res, next) {  store.list().then(data => {  if(data==null){  res.render('error'); //導向錯誤頁面  }else if(data.length > 0){  res.render('store\_list', {items:data}); //將資料傳給顯示頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  module.exports = router; | | | |

表 9-1-71 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-12-2 | 函式名稱 | store.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //------------------------------------------  //執行資料庫動作的函式-傳回所有店家資訊  //------------------------------------------  var list = async function(){  var result=[];  //console.log("查詢店家資訊");  await sql('SELECT \* FROM storeinformation')  .then((data) => {  result = data.rows;  console.log(result) ;  }, (error) => {  result = null;  //console.log("除去錯誤") ;  });    return result;  }  module.exports = {list} | | | |

表 9-1-72 會員的訂單明細列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-1 | 函式名稱 | orderdetail\_one.js |
| 功能 | 取得某一位會員的訂單明細 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  var moment = require('moment');  const orderdetail = require('./utility/orderdetail');  //接收GET請求  router.get('/', function(req, res, next) {  var memberphone = req.session.memberphone;  //console.log(memberphone)  orderdetail.list(memberphone).then(data => {  //console.log(data)  if(data==null){  res.render('error'); //導向錯誤頁面  }else if(data.length > 0){  res.render('orderdetail\_list', {items:data}); //將資料傳給顯示頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  module.exports = router; | | | |

表 9-1-73 建立訂單明細列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-2 | 函式名稱 | orderdetail\_add.js |
| 功能 | 建立訂單明細物件並呼叫新增服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const orderdetail = require('./utility/orderdetail');  //接收POST請求  router.post('/', function(req, res, next) {  console.log('222222222222222')  var memberphone=req.session.memberphone;  var k = req.body.foodname;  k = k.split(',');  var foodname= k[0];  var foodpoint= k[1];  var foodno= req.body.foodno;  var customized= req.body.customized;  var tableno=req.body.tableno;  // 建立一個新資料物件  var newData={  memberphone: memberphone,  foodname: foodname,  foodpoint: foodpoint,  foodno: foodno,  customized: customized,  tableno: tableno  }  console.log('333333333333333333')  console.log(newData)  orderdetail.add(newData).then(d => {  if (d==0){  res.render('addSuccess'); //傳至成功頁面  }else{  res.render('addFail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-74導至訂單明細新增頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-3 | 函式名稱 | orderdetail\_add\_form.js |
| 功能 | 導至訂單明細新增頁面 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const orderdetail = require('./utility/orderdetail');  //const e = require('express');  //接收GET請求  router.get('/', function(req, res, next) {  console.log("1111111111111")  orderdetail.getDropdownData().then(d => {  if (d!=[]){  console.log(d)  res.render('food\_list', {result:d}); //轉至新增頁面  }else{  res.render('addFail'); //導向錯誤頁面  }  });  });  module.exports = router; | | | |

表 9-1-75 取得欲刪除訂單明細之編號列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-4 | 函式名稱 | orderdetail\_remove.js |
| 功能 | 取得欲刪除訂單明細之編號並呼叫刪除服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const orderdetail = require('./utility/orderdetail');  //接收POST請求  router.post('/', function(req, res, next) {  //var memberphone = req.session.memberphone; //取得產品編號  var orderdetailid = req.body.orderdetailid;    orderdetail.remove(orderdetailid).then(d => {  if(d>=0){  res.render('removeSuccess', {result:d}); //傳至成功頁面  }else{  res.render('removeFail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-76導至訂單明細刪除頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-5 | 函式名稱 | orderdetail\_remove\_form.js |
| 功能 | 導至訂單明細刪除頁面 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const orderdetail = require('./utility/orderdetail');  //接收GET請求  router.get('/', function(req, res, next) {  var memberphone = req.session.memberphone; //取得產品編號  var orderdetailid = req.body.orderdetailid;  orderdetail.getDropdown(memberphone, orderdetailid).then(d => {  if (d!=[]){  res.render('orderdetail\_remove\_form', {result:d}); //轉至新增頁面  }else{  res.render('addFail'); //導向錯誤頁面  }  });  });  module.exports = router; | | | |

表 9-1-77 建立訂單明細更新資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-6 | 函式名稱 | orderdetail\_update.js |
| 功能 | 建立訂單明細更新資料物件並呼叫更新服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const orderdetail = require('./utility/orderdetail');  //接收POST請求  router.post('/', function(req, res, next) {  var orderdetailid = req.body.orderdetailid;  var newData={  orderdetailid:orderdetailid,  memberphone: req.body.memberphone,  foodname: req.body.foodname,  foodpoint: req.body.foodpoint,  foodno: req.body.foodno,  customized: req.body.customized,  tableno: req.body.tableno,  //ordtime: req.body.ordtime,    }  console.log(newData)  orderdetail.update(newData).then(d => {  if (d>=0){  res.render('updateSuccess', {result:d}); //傳至成功頁面  }else{  res.render('updateFail'); //導向錯誤頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-78導至搜尋訂單明細編號之頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-7 | 函式名稱 | orderdetail\_update\_no.js |
| 功能 | 導至搜尋訂單明細編號之頁面 | | |
| var express = require('express');  const orderdetail = require('./utility/orderdetail');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  var memberphone = req.session.memberphone; //取得產品編號  var orderdetailid = req.body.orderdetailid;  orderdetail.getDropdown(memberphone, orderdetailid).then(d => {  if (d!=[]){  res.render('orderdetail\_update\_no', {result:d}); //轉至新增頁面  }else{  res.render('notFound'); //導向錯誤頁面  }  });  });  //匯出  module.exports = router; | | | |

表 9-1-79 將原有訂單明細回傳至更新頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-8 | 函式名稱 | orderdetail\_update\_form.js |
| 功能 | 將原有訂單明細回傳至更新頁面之欄位 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  var moment = require('moment');  const orderdetail = require('./utility/orderdetail');  //接收GET請求  router.get('/', function(req, res, next) {  var no = req.query.orderdetailid;  var foodname = req.body.foodname;  var foodpoint = req.body.foodpoint;  orderdetail.query(no, foodname, foodpoint).then(d => {  if (d!=null && d!=-1){  res.render('orderdetail\_update\_form', {result:d}); //將資料傳給更新頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-80 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-9 | 函式名稱 | orderdetail.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  const session = require('express-session');  //------------------------------------------  //執行資料庫動作的函式-取出單一員工  //------------------------------------------  var list = async function(memberphone){  var result={};    await sql('SELECT \* FROM orderdetail WHERE memberphone = $1', [memberphone])  .then((data) => {  //console.log(data.rows.length)  console.log(data.rows)  if(data.rows.length > 0){  result = data.rows;  }else{  result = -1;  }  }, (error) => {  result = null;  });    return result;  }  //------------------------------------------  // 取出型態資料  //------------------------------------------  var getDropdownData = async function(){  //儲存下拉式選單資料    //取回protype資料  await sql('SELECT \* FROM food')  .then((data) => {  result.food = data.rows;  }, (error) => {  result.food = [];  });    //設定回傳資料  var result = {};  result.food = food;  //回傳  return result;  }  //------------------------------------------  // 取出型態資料  //------------------------------------------  var getDropdown = async function(memberphone){  //儲存下拉式選單資料  var orderdetail;  await sql('SELECT \* FROM orderdetail WHERE "memberphone" = $1 ORDER BY orderdetailid', [memberphone])  .then((data) => {  orderdetail = data.rows;  }, (error) => {  result = [];  //console.log(result)  });  //設定回傳資料  var result = {};  result.orderdetail = orderdetail;  //回傳  return result;  }  //------------------------------------------  //執行資料庫動作的函式-新增會員資料  //------------------------------------------  var add = async function(newData){  var result;    const current = new Date();  await sql('INSERT INTO orderdetail (foodname, foodpoint, foodno, customized, memberphone, tableno, ordtime) VALUES ($1, $2, $3, $4, $5, $6, $7)', [newData.foodname, newData.foodpoint, newData.foodno, newData.customized, newData.memberphone, newData.tableno, current])  .then((data) => {  result = 0;  }, (error) => {  result = -1;  });    return result;  }  //----------------------------------  // 刪除會員資料  //----------------------------------  var remove = async function(orderdetailid){  var result;  await sql('DELETE FROM orderdetail WHERE "orderdetailid" = $1', [orderdetailid])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });    return result;  }  //------------------------------------------  //執行資料庫動作的函式-取得一個會員資料  //------------------------------------------  var query = async function(orderdetailid){  var result={};  await sql('SELECT \* FROM orderdetail WHERE "orderdetailid" = $1', [orderdetailid])  .then((data) => {  if(data.rows.length > 0){  result.orderdetail = data.rows[0];  }else{  result.orderdetail = -1;  }  }, (error) => {  result.orderdetail = null;  });    await sql('SELECT \* FROM food ORDER BY foodname')  .then((data) => {  result.food = data.rows;  }, (error) => {  result.food = [];  });  //回傳  return result;  }  //----------------------------------  // 更新會員資料  //----------------------------------  var update = async function(newData){  var results;  const current = new Date();  await sql('UPDATE orderdetail SET "foodname"=$2, "foodpoint"=$3, "foodno"=$4, "customized"=$5, "memberphone"=$6, "tableno"=$7, "ordtime"=$8 WHERE "orderdetailid" = $1', [newData.orderdetailid, newData.foodname, newData.foodpoint, newData.foodno, newData.customized, newData.memberphone, newData.tableno, current])  .then((data) => {  results = data.rowCount;  }, (error) => {  results = -1;  });    return results;  }  module.exports = {list, add, getDropdownData, remove, query, update, getDropdown} | | | |

表 9-1-81取得會員的資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-1 | 函式名稱 | member\_one.js |
| 功能 | 取得某一位會員的資料 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  var moment = require('moment');  const member = require('./utility/member');  //接收GET請求  router.get('/', function(req, res, next) {  var memberphone = req.session.memberphone;    member.one(memberphone).then(data => {  if(data==null){  res.render('error'); //導向錯誤頁面  }else if(data==-1){  res.render('notFound'); //導向找不到頁面  }else{  res.render('member\_list', {items:data}); //將資料傳給顯示頁面  }  })  });  module.exports = router; | | | |

表 9-1-82建立會員資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-2 | 函式名稱 | member\_add.js |
| 功能 | 建立會員資料物件並呼叫新增服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const member = require('./utility/member');  //接收POST請求  router.post('/', function(req, res, next) {  var memberphone=req.body.memberphone; //會員手機號碼  var password=req.body.password; //密碼  var membername= req.body.membername; //會員名稱  var gender= req.body.gender; //性別  var birthday= req.body.birthday; //生日  // 建立一個新資料物件  var newData={  memberphone: memberphone, //會員手機號碼  password: password, //密碼  membername: membername, //會員名稱  gender:gender, //性別  birthday:birthday, //生日  }  console.log(newData)  member.add(newData).then(d => {  if (d==0){  res.render('registered\_success'); //傳至成功頁面  }else{  res.render('registered\_fail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-83導至會員資料新增頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-3 | 函式名稱 | member\_add\_form.js |
| 功能 | 導至會員資料新增頁面 | | |
| var express = require('express');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  res.render('member\_add\_form');  });  module.exports = router; | | | |

表 9-1-84 搜尋會員資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-4 | 函式名稱 | member\_list.js |
| 功能 | 呼叫搜尋會員資料之服務，並回傳及顯示會員資料 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const member = require('./utility/member');  //接收GET請求  router.get('/', function(req, res, next) {  member.list().then(data => {  if(data==null){  res.render('error'); //導向錯誤頁面  }else if(data.length > 0){  console.log(data);  res.render('member\_list', {items:data}); //將資料傳給顯示頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  module.exports = router; | | | |

表 9-1-85 取得欲刪除會員之帳號列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-5 | 函式名稱 | member\_remove.js |
| 功能 | 取得欲刪除會員之帳號並呼叫刪除服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const member = require('./utility/member');  //接收POST請求  router.post('/', function(req, res, next) {  var memberphone = req.body.memberphone; //取得產品編號    member.remove(memberphone).then(d => {  if(d>=0){  res.render('member\_remove\_Success', {results:d}); //傳至成功頁面  }else{  res.render('member\_remove\_Fail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-86導至會員資料刪除頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-6 | 函式名稱 | member\_remove\_form.js |
| 功能 | 導至會員資料刪除頁面 | | |
| var express = require('express');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  res.render('member\_remove\_form');  });  module.exports = router; | | | |

表 9-1-87建立會員更新資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-7 | 函式名稱 | member\_update.js |
| 功能 | 建立會員更新資料物件並呼叫更新服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const member = require('./utility/member');  //接收POST請求  router.post('/', function(req, res, next) {  var memberphone = req.body.memberphone; //取得會員手機號碼  var newData={  memberphone: memberphone, //會員手機號碼  membername: req.body.membername, //會員名稱  password: req.body.password, //密碼  gender: req.body.gender, //性別  birthday: req.body.birthday, //生日  //creationdate: req.body.creationdate //建立日期    }  console.log(newData)  member.update(newData).then(d => {  if (d>=0){  res.render('member\_update\_Success', {results:d}); //傳至成功頁面  }else{  res.render('member\_update\_Fail'); //導向錯誤頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-88導至搜尋會員帳號之頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-8 | 函式名稱 | member\_update\_no.js |
| 功能 | 導至搜尋會員帳號之頁面 | | |
| var express = require('express');  var router = express.Router();  /\* GET home page. \*/  router.get('/', function(req, res, next) {  res.render('member\_update\_no');  });  //匯出  module.exports = router; | | | |

表 9-1-89將原有會員資料回傳至更新頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-9 | 函式名稱 | member\_update\_form.js |
| 功能 | 將原有會員資料回傳至更新頁面之欄位 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  var moment = require('moment');  const member = require('./utility/member');  //接收GET請求  router.get('/', function(req, res, next) {  var no = req.query.memberphone;  member.query(no).then(d => {  if (d!=null && d!=-1){  var data = {  memberphone: d.memberphone,  membername: d.membername,  password: d.password,  gender: d.gender,  birthday: moment(d.birthday).format("YYYY-MM-DD"),  //creationdate: moment(d.creationdate).format("YYYY-MM-DD"),  }  res.render('member\_update\_form', {item:data}); //將資料傳給更新頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-90 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-10 | 函式名稱 | member.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //------------------------------------------  //執行資料庫動作的函式-傳回所有會員資料  //------------------------------------------  var list = async function(){  var result="";  //console.log("查看會員資訊");  await sql('SELECT \* FROM member')  .then((data) => {  result = data.rows;  //console.log(result) ;  }, (error) => {  result = null;  //console.log("除去錯誤") ;  });    return result;  }  //------------------------------------------  //執行資料庫動作的函式-取出單一會員  //------------------------------------------  var one = async function(memberphone){  var result={};    await sql('SELECT \* FROM member WHERE memberphone = $1', [memberphone])  .then((data) => {  if(data.rows.length > 0){  result = data.rows[0];  }else{  result = -1;  }  }, (error) => {  result = null;  });    return result;  }  //------------------------------------------  //執行資料庫動作的函式-新增會員資料  //------------------------------------------  var add = async function(newData){  var result;  console.log(newData)  console.log(newData.memberphone)  console.log(newData.password)  console.log(newData.membername)  console.log(newData.gender)  console.log(newData.birthday)  //console.log(newData.creationdate)  const current = new Date();  await sql('INSERT INTO member (memberphone, password, membername, gender, birthday, creationdate) VALUES ($1, $2, $3, $4, $5, $6)', [newData.memberphone, newData.password, newData.membername, newData.gender, newData.birthday, current])  .then((data) => {  result = 0;  }, (error) => {  result = -1;  });    return result;  }  //----------------------------------  // 刪除會員資料  //----------------------------------  var remove = async function(memberphone){  var result;  await sql('DELETE FROM member WHERE memberphone = $1', [memberphone])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });    return result;  }  //------------------------------------------  //執行資料庫動作的函式-取得一個會員資料  //------------------------------------------  var query = async function(memberphone){  var result={};    await sql('SELECT \* FROM member WHERE memberphone = $1', [memberphone])  .then((data) => {  if(data.rows.length > 0){  result = data.rows[0];  }else{  result = -1;  }  }, (error) => {  result = null;  });    return result;  }  //----------------------------------  // 更新會員資料  //----------------------------------  var update = async function(newData){  var results;  const current = new Date();  await sql('UPDATE member SET membername=$2, password=$3, gender=$4, birthday=$5, creationdate=$6 WHERE memberphone = $1', [newData.memberphone, newData.membername, newData.password, newData.gender, newData.birthday, current])  .then((data) => {  results = data.rowCount;  }, (error) => {  results = -1;  });    return results;  }  module.exports = {list, one, add, remove, query, update} | | | |

表 9-1-91 搜尋菜單列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-15-1 | 函式名稱 | food\_list.js |
| 功能 | 呼叫搜尋菜單之服務，並回傳及顯示菜單 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const food = require('./utility/food');  //接收GET請求  router.get('/', function(req, res, next) {  food.list().then(data => {  if(data==null){  res.render('error'); //導向錯誤頁面  }else if(data.length > 0){  console.log('\*\*\*\*\*\*\*\*\*')  console.log(data);  res.render('food\_list', {result:data}); //將資料傳給顯示頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  module.exports = router; | | | |

表 9-1-92 函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-15-2 | 函式名稱 | food.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //------------------------------------------  //執行資料庫動作的函式-傳回所有會員資料  //------------------------------------------  var list = async function(){  var result="";  //console.log("查看菜單");  await sql('SELECT \* FROM food')  .then((data) => {  result = data.rows;  console.log(result)  //console.log(result) ;  }, (error) => {  result = null;  //console.log("除去錯誤") ;  });    return result;  }  module.exports = {list} | | | |

表 9-1-93 建立計時資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-16-1 | 函式名稱 | caltime\_add.js |
| 功能 | 建立計時資料物件並呼叫新增服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const caltime = require('./utility/caltime');  //const { now } = require('moment');  //接收POST請求  router.post('/', function(req, res, next) {  var memberphone = req.session.memberphone; //取得到達時間    // 建立一個新資料物件  //var timestamp = new Date().getTime();  //var Timestamp = (new Date()).valueOf();    var newData={  memberphone:memberphone  }  console.log(newData)  caltime.add(newData).then(d => {  if (d==0){  res.render('caltime\_success'); //傳至成功頁面  }else{  res.render('caltime\_fail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-94導至計時資料新增頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-16-2 | 函式名稱 | caltime\_add\_form.js |
| 功能 | 導至計時資料新增頁面 | | |
| var express = require('express');  var router = express.Router();  //接收GET請求  router.get('/', function(req, res, next) {  res.render('caltime\_add\_form');  });  module.exports = router; | | | |

表 9-1-95 建立結帳資料列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-16-3 | 函式名稱 | caltime\_addend.js |
| 功能 | 建立結帳資料物件並呼叫新增服務 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const caltime = require('./utility/caltime');  //接收POST請求  router.post('/', function(req, res, next) {  var memberphone = req.session.memberphone; //取得到達時間  var newData={  memberphone:memberphone,  ttotal: req.body.ttotal,  sumtotal: req.body.sumtotal,  totalpoint: req.body.totalpoint,  confoirm: req.body.confoirm,  }  caltime.query(newData).then(d => {  if (d.rowCount>0){  res.render('caltime\_end\_success', {result:d}); //傳至成功頁面  }else{  res.render('caltime\_end\_fail'); //導向錯誤頁面  }  })  });  module.exports = router; | | | |

表 9-1-96導至結帳資料頁面列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-16-4 | 函式名稱 | caltime\_addend\_form.js |
| 功能 | 導至結帳資料頁面 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  var moment = require('moment');  const caltime = require('./utility/caltime');  //接收GET請求  router.get('/', function(req, res, next) {  var memberphone = req.session.memberphone;    caltime.addend(memberphone).then(d => {  //console.log(d)  if (d!=null && d!=-1){  res.render('caltime\_addend\_form'); //將資料傳給更新頁面  }else{  res.render('notFound'); //導向找不到頁面  }  })  });  //匯出  module.exports = router; | | | |

表 9-1-97函式呼叫服務列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-16-5 | 函式名稱 | caltime.js |
| 功能 | 回應上述函式呼叫服務 | | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  const { Client } = require('pg');  const session = require('express-session');  //------------------------------------------  // 新增開始時間  //------------------------------------------  var add = async function(newData){  var result;  const current = new Date();    await sql('INSERT INTO calculatingtime ( "memberphone", "arrivaltime") VALUES ($1, $2)', [newData.memberphone, current])  .then((data) => {  result = 0;  }, (error) => {  result = -1;  });    return result;  }  //------------------------------------------  // 新增結束時間  //------------------------------------------  var addend = async function(memberphone){  var results;  const current = new Date();    await sql('UPDATE calculatingtime SET endtime = $2 where "timeserNo" =(select max("timeserNo") from calculatingtime where memberphone =$1)', [memberphone, current])  .then((data) => {  if(data.rowCount > 0){  results = 1;  }else{  results = -1;  }  }, (error) => {  results = null;  });  await sql('UPDATE checkout SET billtime = $2 where "serNo" =(select max("serNo") from checkout where memberphone =$1)', [memberphone, current])  .then((data) => {  if(data.rowCount > 0){  results = 1;  }else{  results = -1;  }  }, (error) => {  results = null;  });    return results;  }  //------------------------------------------  //執行資料庫動作的函式-計算總花費時間  //------------------------------------------  var query = async function(data){  let results={};  let result={};  let minutes;  let mp=data.memberphone;  let t =data.ttotal;  let glo\_staymins;  let st =data.sumtotal;  let tp =data.totalpoint;  let glo\_points;  let glo\_confirm = data.confirm;  await sql('select \* from calculatingtime WHERE memberphone= $1 and date(arrivaltime) = current\_date and date(endtime) = current\_date', [data.memberphone])  .then((data) => {  result = data.rows[0];  var dt1 = new Date(result.arrivaltime);  var dt2 = new Date(result.endtime);  var diff =(dt2.getTime() - dt1.getTime()) / 60000;  minutes = Math.abs(Math.round(diff));  glo\_staymins =minutes;  }, (error) => {  result = -1;  });  await sql('update calculatingtime set staymins= $1 WHERE memberphone= $2 and date(arrivaltime) = current\_date and date(endtime) = current\_date', [minutes, mp])  .then((data) => {  result = data.rows;  }, (error) => {  result = -1;  });  //----------------------------------  // 計算遊玩所花費的金額  //----------------------------------  await sql('select \* from calculatingtime WHERE memberphone= $1 and date(arrivaltime) = current\_date and date(endtime) = current\_date', [mp])  .then((data) => {  results.ttotal = data.rows[0].ttotal;  }, (error) => {  result = -1;  });    await sql('select \* from storeinformation')  .then((data) => {  result = data.rows[0];  var atime = result.atime;  var apoint = result.apoint;  var lessatime = result.lessatime;  var addapoint = result.addapoint;  var h = parseInt(glo\_staymins/atime)  var r = glo\_staymins % atime  if(r>=lessatime){  h++;  r=0;  t = h \* apoint  }else{  t = h \* apoint + addapoint  }  console.log(t)  results.ttotal =t;    }, (error) => {  result = null;  });    await sql('update calculatingtime set ttotal= $1 WHERE memberphone= $2 and date(arrivaltime) = current\_date and date(endtime) = current\_date', [t, mp])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });  //----------------------------------  // 計算餐點費用  //----------------------------------  await sql('select memberphone, sum(foodpoint) as sumtotal from orderdetail where date(ordtime) = current\_date group by orderdetail.memberphone')  .then((data) => {  console.log(data.rows[0])  if(data.rowCount > 0){  st = data.rows[0].sumtotal;  }else{  st = 0;  }  results.sumtotal=st;  }, (error) => {  result = -1;  });  await sql('update orderdetail set sumtotal= $1 WHERE memberphone= $2 and date(ordtime) = current\_date', [st, mp])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });  //----------------------------------------  // 判斷是否點餐和計算總花費金額及轉換成點數  //----------------------------------------  await sql('select \* from checkout ')  .then((data) => {  result = data.rows[0];  if(st > 0){  tp = parseInt(t) + parseInt(st);  }else{  tp = t;  result = -1;  }  results.totalpoint=tp;    }, (error) => {  result = null;  console.log(tp)  });  await sql('update checkout set totalpoint= $1 WHERE memberphone= $2', [tp, mp])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });  //----------------------------------------  // 判斷點數是否足夠及計算剩餘點數  //----------------------------------------  await sql('select \* from member WHERE memberphone= $1', [mp])  .then((data) => {  result = data.rows[0];  glo\_points = result.points;  }, (error) => {  result = -1;  });  await sql('select \* from calculatingtime WHERE memberphone= $1 and date(endtime) = current\_date', [mp])  .then((data) => {  result = data.rows[0].glo\_confirm;  }, (error) => {  result = -1;  });  await sql('select \* from checkout WHERE memberphone= $1 and date(billtime) = current\_date', [mp])  .then((data) => {  result = data.rows[0];  if(glo\_points > tp){  glo\_points = glo\_points - tp;  glo\_confirm = '是';  }else{  glo\_confirm = '否';  }  results.glo\_confirm = glo\_confirm;  }, (error) => {  result = null;  });  await sql('update member set points= $1 WHERE memberphone= $2', [glo\_points, mp])  .then((data) => {  results.rowCount = data.rowCount;  }, (error) => {  result = -1;  });  await sql('update calculatingtime set confirm= $1 WHERE memberphone= $2 and date(endtime) = current\_date', [glo\_confirm, mp])  .then((data) => {  result = data.rowCount;    }, (error) => {  result = -1;  });  return results;  }  module.exports = {add, addend, query}; | | | |

表 9-1-98

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 函式名稱 |  |
| 功能 |  | | |
|  | | | |

表 9-1-99

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 函式名稱 |  |
| 功能 |  | | |
|  | | | |

表 9-1-100

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 函式名稱 |  |
| 功能 |  | | |
|  | | | |

表 9-1-101

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 函式名稱 |  |
| 功能 |  | | |
|  | | | |