

91APP – 訂單管理後台

API路徑前綴:<https://localhost:5001/api/>

系統架構

- 我將該次作業需求，定義為訂單後台管理系統，目前針對訂單列表、完整訂單內容、產品資訊介紹，以上三種功能進行實作
- 在訂單列表中，賣家可查看所有針對自己商店下訂的紀錄，並且在完成商品出貨後，可點擊出貨按鍵，更改訂單狀態
- 在訂單列表中，也可以透過點擊訂單序號旁的查看更多，進入完整訂單內容
- 在完整訂單內容中，可進一步點擊產品名稱，進到產品資訊介紹頁面

資料庫設計說明

- 所有資料表都有Admin_Id、CreateTime、UpdateTime、Archive
 - Admin_Id，紀錄最後針對該筆資料的異動者
 - CreateTime，該筆資料的建立時間
 - UpdateTime，該筆資料的最後更新時間
 - Archive，DB 不執行DELETE 動作，若需刪除資料，以 Archive = 1 來實現。
- 透過 Order_Product_Relation，實現訂單與訂購產品一對多的關係
- 類別相關資訊以 ooxxType_Id 的方式進行寫入，提升後續的可維護性

備註

- 使用者只能操作自己的訂單內容
- 因 DEMO API 未實作 Token 之類的身分驗證機制，故暫以建構函式中的 `userId` 模擬從 Middleware 解出來的使用者資訊，`userId` 以信任資訊無誤的前提，在 `controller` 中進行操作
- 異動資料時，`Admin_Id`欄位會記錄下該次資料異動者的 `User_Id`，同時將 `UpdateTime` 更新為當下
- 刪除資料時，不會真的進行 `DELETE`動作，而是將 `Archive` 狀態調為1
- 對資料庫 `CRUD`，透過自行封裝的 `Function` 進行操作，增刪改的部分已開啟交易機制

查詢訂單GET:order/list

Request

```
{  
  
}
```

Response

```
{  
  "StatusCode": 0  
  "Message": 成功  
  "Data":  
  [  
    {  
      "Id": int,  
      "Buyer": string,  
      "PaymentType": string,  
      "Total": int,  
      "Address": string,  
      "CreateTime": datetime,  
      "OrderStatusType_Id": int,  
      "OrderStatusType": string  
    }, {...}, {...}, ...  
  ]  
}
```

目前該系統未實作身分驗證機制，故暫規劃以 **Request.Header** 中所攜帶的 **User_Id** 做判斷使用者身分的依據，而該值由後端寫死為 **1** 以因應第一階段程式需求，待進入第二階段加上登入機制後，即可完善上述規劃。

查詢訂單GET:order/detail/:id

Request

```
{  
}
```

Response

```
{  
  "StatusCode": 0  
  "Message": 成功  
  "Data":  
  {  
    "Id": int,  
    "Buyer": string,  
    "Seller": string,  
    "Products": [{"Name": string, "Price": int, "Quantity": int, "Total": int}, {...],  
    "PaymentType": string,  
    "Address": string,  
    "CreateTime": datetime,  
    "OrderStatusType": string,  
    "Total": int,  
    "Cost": int  
  }  
}
```

修改訂單狀態PUT:order/status

Request

```
{  
    "Ids": int [ ]  
}
```

Response

```
{  
    "StatusCode": 0  
    "Message": 成功  
    "Data": null  
}
```

執行修改動作前，先確認 DB 中的 OrderStatusType_Id 狀態，
只接受 OrderStatusType_Id 由 1 改為 2

修改訂單狀態PUT:order/status/:id

Request

```
{  
  "Id": int  
}
```

Response

```
{  
  "StatusCode": 0  
  "Message": 成功  
  "Data": null  
}
```

執行修改動作前，先確認 DB 中的 OrderStatusType_Id 狀態，
只接受 OrderStatusType_Id 由 1 改為 2

查詢產品GET:product/detail/:id

Request

```
{  
}
```

Response

```
{  
  "StatusCode": 0  
  "Message": 成功  
  "Data":  
  {  
    "Id": int,  
    "Name": string,  
    "Description": string,  
    "Cost": int,  
    "Price": int,  
    "CreateTime": datetime  
  }  
}
```