

列出 3 種寫法，來將資料新增於資料列下方，並且包成為一組副程式

```
//建成一個刷新資料的副程式
function refreshUI(){
    $("#latestNews").empty();//清空舊有內容
    for(let i = 0; i < newsList.length ; i++){
        let newItem = newsList[i];
        //加入<li>
        //let litext = "<li class='list-group-
item'>"+newItem.title+"["+newItem.ymd+"]</li>";
        //``寫法，效果同上行，但下行較好維護
        //let litext =`<li class="list-group-
item"> ${newItem.title} [${newItem.ymd}] </li>`;
        //alert(litext);
        //showToast("test",litext);//120 行，傳入(標題，資料)

        //資料陣列出現在下方，但與上方資料不同，需增加美術樣式
        //$("#latestNews").append(litext);//append->既有內容不動，增
        加新資料在後方

        let litext = newItem.title+"["+newItem.ymd+"]";
        //傳入一個空元素，用 jQuery.text 增加陣列資料和.addClass 美術樣式
        進<li></li>中
        let $li = $("<li></li>");//jQuery 的 li
            .text(litext)
            .addClass("list-group-item")
            .append('<span class="pull-
right"><button class="btn btn-info btn-
xs editItem"><span class="glyphicon glyphicon-pencil" aria-
hidden="true"></span></button>&nbsp;<button class="btn btn-danger btn-
xs deleteItem"><span class="glyphicon glyphicon-remove" aria-
hidden="true"></span></button></span>');
        $li.appendTo("#latestNews");//一個學生參加這個班級
        //$("#latestNews").append($li);//一個班級 append 一個學生進來
    }
}
```

下載且重新整理資料

```
downloadAndRefresh();//呼叫此函數
function downloadAndRefresh(){
    $.ajax({//透過.ajax 方法
        type:"get",//用 get
        url:"/home/news"//向/home/news 撈資料
    }).then(function(e){//撈到資料 e
        newsList = JSON.parse(e);//字串轉成 JSON 格式陣列，存給
newsList
        refreshUI();
    })
}
```

呼叫對方框，點選背景圖示不會有反應

```
$("#NewItem").on("click",function(){//呼叫對話格
    $("#newsModal").modal({backdrop: "static"}/*此方法可以讓背景處無
作用*/ );//呼叫 modal 方法
```

新增一筆資料於資料表下方

```
$("#okButton").on("click",function(){
    $("#newsModal").modal("hide");//點選確定後對話框會隱藏
    let dataToServer = {
        //title 與 ymd 兩個輸入框的名字
        title:$("#titleTextBox").prop("value"),
        ymd:$("#ymdTextBox").val()
    }
    //alert(JSON.stringify(dataToServer));//確定有讀取輸入資料

    //往後端丟資料
    $.ajax({
        type:"post",
        url:"/home/news",
        contentType:"application/json",//指定網頁要送到 server 端的資料
型態
        data:JSON.stringify(dataToServer)
    }).then(function(){
        //alert("ok");//需手動重新整理
        //將重新整理包成一個函數呼叫
```

```

        downloadAndRefresh();
    })
})

```

更資料用 `currentindex` 來做判斷

```
let currentindex = -1; //-1->新增，0->編輯第一筆，2->編輯第三筆
```

更改資料

```

$(".editItem").on("click",function(){
    var index = $(this).closest("li").index();
    currentindex = index;

    //被指定要更新的資料未被替換，更新後資料出現在下方
    //將 okButton 倒過來描述
    title:$("#titleTextBox").prop("value",newsList[index].title
); //value 屬性設定為新的內容
    ymd:$("#ymdTextBox").val(newsList[index].ymd);
    $("#newsModal").modal({backdrop:"static"});
})

```

//新增/更改資料

```

$("#okButton").on("click",function(){
    $("#newsModal").modal("hide");//點選確定後對話框會隱藏
    if(currentindex<0){ //-1 時, new Item
        let dataToServer = {
            //title 與 ymd 兩個輸入框的名字
            title:$("#titleTextBox").prop("value"),
            ymd:$("#ymdTextBox").val()
        }
        //alert(JSON.stringify(dataToServer)); //確定有讀取輸入資料

        //往後端丟資料
        $.ajax({
            type:"post",
            url:"/home/news",
            contentType:"application/json", //指定網頁要送到 server 端
            data:JSON.stringify(dataToServer)

```

```
    }).then(function(){
        //alert("ok");//需手動重新整理
        //將重新整理包成一個函數呼叫
        downloadAndRefresh();
    })
}
else{//edit Item
    let dataToServer = {
        newsId:newsList[currentindex].newsId,
        //title 與 ymd 兩個輸入框的名字
        title:$("#titleTextBox").prop("value"),
        ymd:$("#ymdTextBox").val()
    }
    $.ajax({
        type:"put",
        url:"/home/news",
        data:dataToServer
    }).then(function(){
        //alert("ok");//需手動重新整理
        //將重新整理包成一個函數呼叫
        downloadAndRefresh();
    })
    //alert(JSON.stringify(dataToServer));//確定是否抓到資料
}

})
```